

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Guna mengetahui kekurangan penelitian sebelumnya, peneliti melakukan studi pustaka terhadap penelitian terdahulu yang relevan dengan penelitian ini. Penelitian terdahulu yang dilakukan oleh Elmayati dengan masalah yang terjadi pada perguruan tinggi STMIK-MURA perihal pengajuan beasiswa masih menggunakan cara manual belum menggunakan teknologi internet. Berdasarkan permasalahan tersebut dirancanglah sebuah sistem informasi pengajuan beasiswa berbasis web dengan metode *Analytical Hierarchy Process* dan untuk pengembangan sistemnya menggunakan metode *Rapid Application Development*. Dihasilkanlah sistem informasi yang dapat melakukan pengajuan beasiswa dan mengatur data pendaftar secara online serta melakukan perhitungan AHP untuk dapat menghasilkan hasil dari penyaringan pendaftar apakah diterima atau tidak [5]. Pada penelitian ini tidak disediakan dashboard pendaftar untuk dapat mengatur pengajuan beasiswa.

Kemudian pada penelitian Supriyatna, mengangkat masalah mengenai proses pengajuan klaim untuk Pemberi Pelayanan Kesehatan masih dilakukan secara manual dan pengelolaan data yang masih menggunakan *Microsoft Excel*. Untuk mengatasi masalah tersebut dirancanglah sebuah sistem informasi berbasis website yang dapat digunakan untuk melakukan pengecekan dokumen klaim secara online, mempermudah pengajuan klaim dengan mengunggah dokumen pada website, dan mengurangi resiko kehilangan data. Sistem informasi yang telah dirancang adalah sistem informasi pengajuan klaim jaminan pemeliharaan kesehatan masyarakat dengan menggunakan metode *Rapid Application Development*. Dihasilkan sebuah sistem informasi pengajuan klaim yang dapat melakukan pengajuan klaim jaminan kesehatan bagi pihak pemohon dan bagi pihak perusahaan

dapat melakukan pengecekan data serta pengorganisasian data yang terstruktur [9]. Kekurangan dari penelitian ini tidak dijelaskannya hak akses dari pengguna yang menggunakan sistem informasi dan tidak dilakukannya pengujian sistem apakah layak atau tidak.

Sedangkan pada penelitian yang dilakukan oleh Rini *et al.* mengangkat masalah mengenai teknis pengambilan data pada Badan Kesatuan Bangsa dan Politik Kota Palembang yaitu pengolahan data yang dikelola admin masih sulit diorganisir dan pemohon belum mengetahui mekanisme serta ketentuan umum dalam pengambilan data penelitian. Untuk mengatasi masalah tersebut dibuatlah sistem pengajuan pengambilan data penelitian Bankesbangpol berbasis *website* dengan metode pengembangan sistemnya *Rapid Application Development*. Hasilnya berupa sistem informasi yang dapat digunakan pemohon untuk melakukan pengajuan pengambilan data secara sistematis dan bagi admin untuk melakukan pengaturan data [3]. Adapun kekurangan dari penelitian ini adalah tidak tersedianya manajemen dashboard pemohon.

Penelitian yang dilakukan oleh Susanto dan Ramadhan mengangkat permasalahan di Dinas Kesehatan Kota Metro tentang pelayanan perizinan praktik tenaga kesehatan yang masih secara *manual* menggunakan *Microsoft Excel* dan *Microsoft Word* dalam pendataan permohonannya. Masalah tersebut disebabkan karena proses perizinan yang belum tersusun sehingga diperlukan sebuah sistem informasi yang menangani hal tersebut. Sistem informasi yang dibuat adalah sistem informasi perizinan praktik tenaga kesehatan berbasis web pada Dinas Kesehatan Kota Metro. Metode pengembangan sistem yang digunakan adalah metode *Extreme Programming* dengan metode pengujian sistemnya menggunakan *black box testing*. Hasilnya adalah sistem informasi yang dibuat berhasil mempermudah dan mempercepat proses perizinan tenaga kesehatan dan hasil dari pengujian pada tiap-tiap aktor secara keseluruhan fungsional dari sistem berjalan dengan semestinya [10]. Adapun kekurangan dari penelitian ini adalah tidak tersedianya manajemen dashboard pemohon.

Selanjutnya penelitian yang dilakukan oleh Fatoni *et al.* mengambil sebuah masalah yang berkaitan dengan proses pengajuan cuti atau izin masuk kerja bagi karyawan, dosen dan pimpinan masih menggunakan cara tertulis di Universitas Bina Darma. Sistem informasi pengajuan cuti dan izin dibuat berdasarkan masalah tersebut dengan menggunakan metode *Rapid Application Development*. Dan menghasilkan sistem informasi pengajuan cuti dan izin berbasis *website* pemohon dapat melakukan pengajuan hanya dengan mengisi form kemudian menunggu hasil *review* diterima atau tidak, serta dilakukan pengujian sistem informasi menggunakan teknik *black box* [4]. Dalam penelitian ini tidak dibuatkannya *Unified Modeling Language (UML)* sehingga alur yang dibuat pada sistem ini sulit untuk dimengerti.

Penelitian yang dilakukan oleh Putri *et al.*, mengangkat masalah yang terjadi di PT. Bank Negara Indonesia masih menyimpan berkas pengajuan pembuatan kartu kredit secara tidak terstruktur, masih menggunakan formulir kertas, dan pendataanya menggunakan *Microsoft Excel*. Berdasarkan dari permasalahan tersebut dirancang sebuah sistem informasi berbasis web yang dapat melakukan pengajuan kartu kredit menggunakan sistem informasi dan pendataan yang tersusun secara terstruktur dengan menggunakan *database*. Sistem informasi yang dirancang bernama sistem informasi pendataan pengajuan pembuatan kartu kredit berbasis web dengan studi kasus kantor wilayah PT. Bank Negara Indonesia (Persero) Tbk. Dengan menggunakan metode pengembangan sistem *Rapid Application Development*. Penelitian ini menghasilkan sistem informasi yang dapat melakukan pengorganisasian data dari pengajuan kartu kredit secara terstruktur [11]. Kekurangan dari penelitian ini adalah metode pengumpulan data yang dilakukan tidak dijelaskan dan tidak menyantumkan hasil pengujian sistem.

Berikut merupakan ringkasan dari penelitian terdahulu tersaji pada Tabel 2.1.

Tabel 2. 1 Ringkasan Penelitian Sebelumnya

No	Judul	Peneliti	Masalah	Metode	Hasil	Kekurangan	Perbandingan
1	Aplikasi Sistem Informasi Pengajuan Beasiswa Berbasis Web Pada Sekolah Tinggi Manajemen Dan Ilmu Komputer MUSI RAWAS (STMIK-MURA) Kota Lubuklinggau [5]	Elmayati (2016)	Pengajuan beasiswa masih menggunakan cara manual belum menggunakan teknologi internet.	<i>Rapid Application Development & Analytical Hierarchy Process</i>	Sistem informasi yang dapat melakukan pengajuan beasiswa dan mengatur data pendaftar secara online serta melakukan perhitungan AHP untuk dapat menghasilkan hasil dari penyaringan pendaftar apakah diterima atau tidak.	Tidak disediakan dashboard pendaftar untuk dapat mengatur pengajuan beasiswa.	Pada penelitian yang akan dilakukan akan membuat dashboard bagi pemohon dan dilakukan pengujian sistem.
2	Sistem Informasi Pengajuan Klaim Jaminan Pemeliharaan	Adi Supriyatna (2016)	Proses pengajuan klaim masih dilakukan secara manual dan pengolahan	<i>Rapid Application Development</i>	Sistem informasi pengajuan klaim yang dapat	Tidak dijelaskannya hak akses dari pengguna yang menggunakan	Akan dibuatkan desain sistem dengan menggunakan Unified

No	Judul	Peneliti	Masalah	Metode	Hasil	Kekurangan	Perbandingan
	Kesehatan Masyarakat Berbasis Web [9]		data klaim yang masih menggunakan <i>Microsoft Excel</i> .		melakukan pengajuan klaim jaminan kesehatan bagi pihak pemohon dan bagi pihak perusahaan dapat melakukan pengecekan data serta pengorganisir data yang terstruktur.	sistem informasi dan tidak dilakukannya pengujian sistem apakah layak atau tidak.	Modeling Sistem sehingga pembuatan sistem dapat dengan mudah dimengerti dan juga dilakukan pengujian sistem.
3	Penerapan Metode RAD Pada Sistem Pengajuan Pengambilan Data Penelitian Bankesbangpol Kota Palembang [3]	Arsia Rini, Fatmariyani (2017)	Pengolahan data yang dikelola admin masih sulit diorganisir dan pemohon belum mengetahui mekanisme serta ketentuan umum dalam pengambilan data penelitian.	<i>Rapid Application Development</i>	Sistem informasi yang dapat digunakan pemohon untuk melakukan pengajuan pengambilan data secara sistematis dan bagi admin untuk	Tidak tersedianya manajemen dashboard pemohon.	Pada penelitian yang akan dilakukan akan membuat dashboard bagi pemohon dan dilakukan pengujian sistem.

No	Judul	Peneliti	Masalah	Metode	Hasil	Kekurangan	Perbandingan
					melakukan pengaturan data. Dan juga dilakukan pengujian <i>black box testing</i> dengan hasil uji sesuai semua.		
4	Rancang Bangun Aplikasi Berbasis Web Perizinan Praktik Tenaga Kesehatan Menggunakan Framework Codeigniter Pada Dinas Kesehatan Kota Metro [10]	Erliyan Redy Susanto dan Ferdy Ramadhan (2017)	Pelayanan perizinan praktik tenaga kesehatan masih menggunakan cara <i>manual</i> menggunakan <i>Microsoft Excel</i> dan <i>Microsoft Word</i> dalam melakukan pendataan permohonannya.	<i>Extreme Programming</i>	Sistem informasi perizinan praktik tenaga kesehatan yang dapat digunakan oleh staff perizinan, sekretariat, dan tenaga kesehatan. Pengujian <i>black box</i> dilakukan pada tiap-tiap aktor menghasilkan keseluruhan fungsi berjalan	Tidak tersedianya manajemen dashboard pemohon.	Pada penelitian yang akan dilakukan akan membuat dashboard bagi pemohon.

No	Judul	Peneliti	Masalah	Metode	Hasil	Kekurangan	Perbandingan
					dengan semestinya.		
5	Sistem Informasi Pengajuan Cuti dan Izin Berbasis Web [4]	Fatoni, Dony Wahyu Isprananda, dan Ahmad Syazili (2020)	Proses pengajuan cuti atau izin masuk kerja bagi karyawan, dosen dan pimpinan masih menggunakan cara tertulis di Universitas Bina Darma.	<i>Rapid Application Development</i>	Sistem informasi pengajuan cuti dan izin berbasis <i>website</i> pemohon dapat melakukan pengajuan hanya dengan mengisi form, serta dilakukan pengujian sistem informasi menggunakan teknik <i>black box</i> dengan hasil uji yang telah sesuai.	Tidak dibuatkannya <i>Unified Modeling Language (UML)</i> sehingga alur yang dibuat pada sistem ini sulit untuk dimengerti.	Akan dibuatkan desain sistem menggunakan Unified Modeling Language sehingga perancangan sistem dapat dengan mudah dimengerti dan dilakukan pengujian sistem.
6	Sistem Informasi Pendataan Pengajuan	Eunike Gracia Putri, Maniah,	Proses pendataan pengajuan pembuatan	<i>Rapid Application Development</i>	Sistem informasi yang dapat melakukan	Metode pengumpulan data yang dilakukan tidak	Memperjelas bagaimana proses pengumpulan

No	Judul	Peneliti	Masalah	Metode	Hasil	Kekurangan	Perbandingan
	Pembuatan Kartu Kredit Berbasis Web (Studi Kasus: Kantor Wilayah PT. Bank Negara Indonesia (Persero) Tbk.) [11]	dan Mubassiran (2020)	masih menggunakan formulir kertas dan pendataan masih menggunakan <i>Microsoft Excel</i>		pengorganisasian data dari pengajuan kartu kredit secara terstruktur.	dijelaskan dan tidak menyantumkan hasil pengujian sistem.	data dan mengidentifikasi kebutuhan serta dilakukan pengujian sistem.

2.2. Dasar Teori

2.2.1 Pendidikan, Penelitian dan Pengembangan (Diklitbang)

Pendidikan, penelitian dan pengembangan memiliki fungsi sebagai lahan praktek bagi institusi pendidikan khususnya pendidikan kesehatan serta menyelenggarakan pendidikan dan penelitian untuk pegawai dan pihak luar. Tugas utama dari diklitbang adalah melakukan penyiapan bahan perumusan kebijakan teknis di seksi pendidikan dan penelitian yang meliputi pelaksanaan pendidikan, penelitian dan pengembangan profesi kesehatan, dan pengelolaan pendidikan serta pelatihan [12].

2.2.2 Praktik Kerja Lapangan (PKL)

Praktik kerja lapangan atau praktik kerja industri merupakan kegiatan pendidikan, pelatihan dan pembelajaran di dunia industri. Kegiatan ini memiliki fungsi agar siswa atau mahasiswa dapat mendapatkan pengalaman dunia kerja secara langsung dan meningkatkan keahlian. Selain itu kegiatan PKL dapat mempererat hubungan antara institusi pendidikan dengan dunia industri yang berjalan pada saat PKL [13].

2.2.3 Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan - laporan yang dibutuhkan [14].

Sistem informasi dapat dirancang menyesuaikan dengan kebutuhan. Untuk dapat menerapkan sistem yang efektif dan efisien dibutuhkan perencanaan, pelaksanaan, pengaturan, dan evaluasi [15].

Dari pernyataan sebelumnya dapat diambil kesimpulan bahwa sistem informasi dapat dikatakan sebagai sistem yang mengelola data -

data sehingga dapat dengan mudah dipahami oleh pengguna sistem yang dirancang berdasarkan suatu kebutuhan.

2.2.4 Website

Pada [16] *website* merupakan rangkaian dari beberapa halaman yang menjadi satu kesatuan dengan tiap-tiap halaman yang berisi informasi berupa teks, gambar, video, audio dan animasi serta terhubung dengan internet. *Website* statis merupakan *website* yang berjalan statis dengan tidak ada perubahan informasi didalamnya. *Website* dinamis adalah *website* yang dapat berinteraksi dengan pengguna dalam pertukaran informasi yang dinamis serta berhubungan dengan *database* [17].

2.2.5 PHP (*Hypertext Preprocessor*)

PHP singkatan dari *Hypertext Preprocessor* yang merupakan bahasa pemrograman yang berbasis *server-side programming* yaitu bahasa pemrograman yang berjalan di sisi belakang server. Kegunaan dari PHP adalah melaksanakan pengolahan data yang berasal dari *database*. Data website akan disimpan ke penyimpanan data dengan beberapa cara yaitu tambah, edit, hapus, dan tampilkan pada website yang diatur oleh website [16].

2.2.6 Framework Laravel

Laravel merupakan sebuah framework yang dibuat oleh Taylor Otwell dan laravel ini pertama dibuat pada April 2011 [18]. Laravel juga salah satu framework aplikasi web yang memiliki penulisan coding yang *expressive* dan *elegant*. *Framework* laravel saat ini aktif dikembangkan setiap 6 bulan sekali rilis update baru sehingga menjadikannya pilihan yang tepat bagi pengembang aplikasi web karena memiliki arsitektur *framework* yang terbaru [19].

Laravel adalah salah satu framework PHP yang dibangun dengan konsep MVC (*Model View Controller*) dan memiliki lisensi MIT. MVC

merupakan metode pendekatan dalam pembuatan perangkat lunak yang memisahkan *application layer logic* dan *presentation* [18].

a. *Model*

Model dalam laravel memiliki beberapa fungsi yang membantu dalam mengelola basis data, seperti *insert data*, *update data*, *delete data*, *get data* [18].

b. *View*

View adalah salah satu konsep dalam MVC yang mengatur tampilan pengguna [18].

c. *Controller*

Controller merupakan konsep MVC yang menjembatani antara *View* dengan *Model* yang bertugas melakukan pengolahan data [18].

2.2.7 Basis Data (*Database*)

Basis data atau *database* adalah kumpulan data terstruktur. Sehingga dapat menambahkan, mengakses, serta memproses data yang tersimpan dalam *database* komputer, diperlukan sistem manajemen basis data [20]. Kemudian pada [21], *database* dapat dikatakan sebagai kumpulan dari tabel dan file dengan setiap tabel terdiri dari *record* yang disusun atas *field-field* yang ada di dalamnya.

2.2.8 MySQL *Database*

MySQL merupakan RDBMS (*Relational Database Management System*) yang dapat melakukan penanganan data secara banyak sekaligus dan mampu mengirim dan menerima data secara cepat serta multi user [22]. MySQL merupakan *database server* yang dikembangkan oleh MySQL AB Swedia dengan lisensi *Open Source* dengan menyediakan fitur yang lengkap dan mudah untuk digunakan [23].

2.2.9 Rapid Application Development (RAD)

Rapid Application Development adalah suatu metode pengembangan sistem perangkat lunak yang tergolong dalam teknik incremental. Pada metode ini menggunakan teknik yang berorientasi objek didalamnya mencakup pengembangan serta perangkat-perangkat lunak. Dengan menggunakan metode ini dapat menghasilkan produk berkualitas tinggi dengan menggunakan strategi seperti metode *Prototype* [24]. Metode *Rapid Application Development* diperkenalkan oleh James Martin yang mengacu pada suatu pengembangan siklus hidup untuk memproduksi sistem secara cepat tanpa mengurangi kualitas [25].

Dalam metode *Rapid Application Development* memiliki 3 tahapan agar tercapainya suatu sistem antara lain:



Gambar 2. 1 Metode RAD [24]

a. *Requirement Planning* (Rencana Kebutuhan)

User dan *analyst* melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak [24].

b. *Design System* (Desain Sistem)

Pada tahap ini keaktifan *user* yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain

dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Seorang *user* dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain [24].

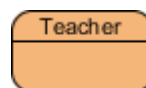
c. *Implementation* (Implementasi)

Tahapan ini adalah tahapan programmer yang mengembangkan desain suatu program yang telah disetujui oleh user dan analyst. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini user biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut [24].

2.2.10 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram adalah diagram yang menunjukkan desain konseptual dari model konseptual suatu database relasional. Selain itu, ERD juga merupakan gambar yang menghubungkan satu objek di dunia nyata dengan objek lain, lebih dikenal sebagai hubungan antar entitas [26]. Berikut merupakan notasi-notasi dalam *entity relationship diagram*.

a. *Entity* (Entitas)

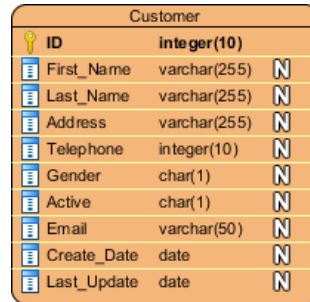


Gambar 2. 2 Entitas ERD

Entity atau entitas merupakan sesuatu yang ditentukan dalam sebuah sistem seperti aktor, objek, konsep, atau kejadian yang berbentuk tabel. Dalam ERD, entitas digambarkan sebagai bentuk

kotak persegi panjang dengan nama entitas dibagian atas dan atribut pada bagian bawah serta penamaan entitas harus unik [27].

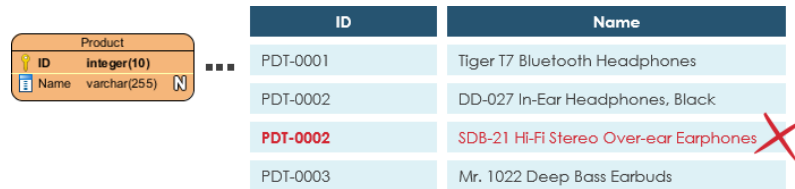
b. *Entity Attributes* (Atribut Entitas)



Gambar 2. 3 Entitas Atribut

Atribut entitas merupakan karakteristik atau properti dari suatu entitas menyesuaikan dengan namanya. Atribut diberi nama menyesuaikan dengan karakteristik entitas dengan tipe datanya seperti *string*, *integer*, atau yang lainnya [27].

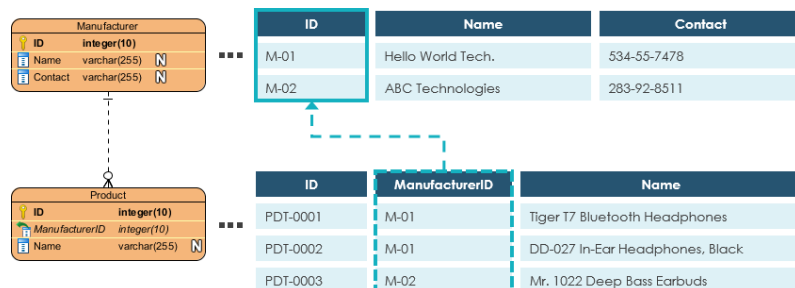
c. *Primary Key*



Gambar 2. 4 *Primary Key* ERD

Primary key merupakan sebuah atribut yang memiliki sifat unik dalam suatu entitas. Ditandai dengan tanda kunci pada salah satu atribut. Pada *primary key*, hasil data tidak bisa lebih dari satu karena bersifat unik [27].

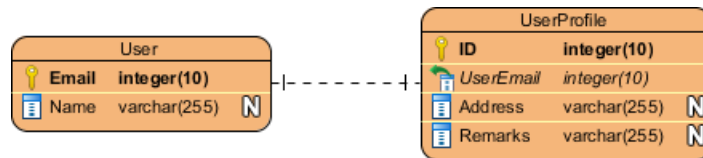
d. *Foreign Key*



Gambar 2. 5 *Foreign Key* ERD

Foreign key merupakan sebuah atribut yang berperan sebagai referensi dari entitas yang lain. Referensi ini digunakan untuk merelasikan satu entitas dengan entitas lainnya [27].

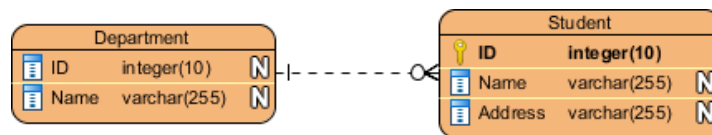
e. *One-to-One Cardinality*



Gambar 2. 6 *One-to-One Cardinality*

One-to-One cardinality merupakan relasi dalam ERD dimana ketergantungan entitas tersebut hanya memiliki satu data pada entitas relasinya. Contoh entitas pengguna bergantung pada entitas profil, dimana pada entitas profil memiliki atribut lanjutan dari entitas pengguna [27].

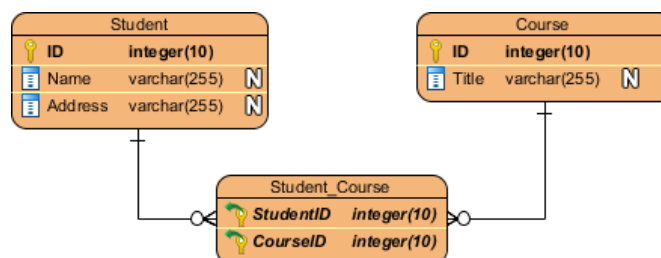
f. *One-to-Many Cardinality*



Gambar 2. 7 *One-to-Many Cardinality*

One-to-Many cardinality merupakan relasi dalam ERD dimana ketergantungan entitas tersebut memiliki banyak atau lebih dari satu data pada entitas lainnya. Contoh entitas program studi dengan entitas mahasiswa, dimana mahasiswa hanya memiliki satu program studi sedangkan program studi memiliki banyak mahasiswa [27].

g. *Many-to-Many Cardinality*



Gambar 2. 8 *Many-to-Many Cardinality*

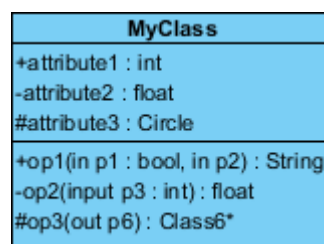
Many-to-Many cardinality merupakan relasi dalam ERD dimana ketergantungan entitas tersebut memiliki banyak data antara entitas satu dengan entitas relasi. Contoh entitas mahasiswa dengan entitas mata kuliah, dimana mahasiswa memiliki banyak mata kuliah dan mata kuliah memiliki banyak mahasiswa [27].

2.2.11 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan sebuah teknik dalam pengembangan sistem yang menggunakan bahasa visual yang bertujuan untuk melakukan pencatatan spesifikasi dalam membuat sistem [28]. UML memiliki beberapa kegunaan ketika diaplikasikan yaitu untuk merancang perangkat lunak, sarana komunikasi antara perangkat lunak dengan proses bisnis, menjelaskan sistem secara rinci untuk analisis dan kebutuhan sistem, dan mendokumentasikan sistem dari proses dan organisasinya [29].

a. Diagram Kelas (*Class Diagram*)

Diagram kelas merupakan diagram yang bersifat statis biasa terdapat di pemodelan sistem berbasis objek dengan memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi [29].



Gambar 2. 9 Contoh *Class Diagram*

Gambar 2. 9 menunjukkan komponen dari *class diagram*, terdapat komponen penyusun yang antara lain.

- *Class Name* (Komponen Atas)

Komponen ini berisi nama kelas, pada tiap kelas memiliki nama kelas yang berbeda atau unik [30].

- *Class Attributes* (Komponen Tengah)

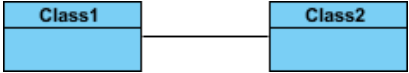
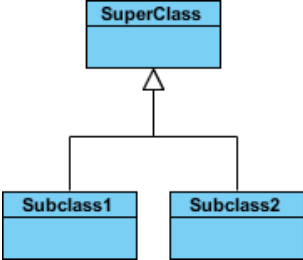
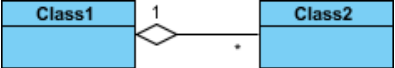
Komponen ini berisikan atribut kelas, pada tiap atribut terdapat tipe data setelah titik dua. Komponen ini digunakan untuk menjelaskan seberapa penting kelas tersebut [30].

- *Class Methods* (Komponen Bawah)

Komponen ini berisi fungsi atau kegiatan dari kelas berdasarkan atribut. Pada tiap operasi atau *method* memiliki tipe data yang menandakan tipe data pada akhir operasi [30].

Dalam *class diagram* terdapat aturan dalam relasi antar kelasnya, berikut adalah relasi dari *class diagram*.

Tabel 2. 2 Notasi *Class Diagram*

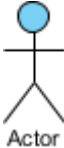

Notasi	Representasi Visual
<p><i>Association</i></p> <p><i>Association</i> merupakan relasi antar kelas yang bersifat statis. Relasi asosiasi terhubungan dengan garis penuh [30].</p>	 <pre> classDiagram Class1 --- Class2 </pre>
<p><i>Inheritance</i></p> <p><i>Inheritance</i> merupakan relasi antar kelas yang memiliki kemampuan mewarisi atribut dan atau metode dari <i>super class</i>. Relasi <i>inheritance</i> dihubungkan dengan garis penuh dan panah ke <i>super class</i> [30].</p>	 <pre> classDiagram SuperClass < -- Subclass1 SuperClass < -- Subclass2 </pre>
<p><i>Aggregation</i></p> <p><i>Aggregation</i> merupakan relasi antar kelas yang memiliki keterkaitan satu sama lain</p>	 <pre> classDiagram Class1 o-- "*" Class2 </pre>


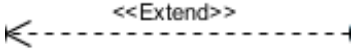
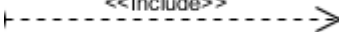
Notasi	Representasi Visual
namun tiap kelas tersebut dapat berdiri sendiri. Relasi agregasi dihubungkan dengan garis penuh dan panah berbentuk wajik [30].	

b. Diagram *Use Case*

Diagram *use case* merupakan diagram yang bersifat statis bertujuan untuk mengorganisasi dan memodelkan perilaku dari sistem. Diagram ini akan memperlihatkan himpunan *use case* dan aktor-aktor [29]. Berikut merupakan notasi-notasi dalam *use case diagram*.

Tabel 2. 3 Notasi *Use Case Diagram*

Notasi	Representasi Visual
<p>Aktor</p> <p>Aktor adalah suatu notasi yang menggambarkan peran pengguna dalam suatu bisnis proses. Dapat memiliki peran yang berbeda menyesuaikan proses bisnis dari suatu sistem dimana aktor akan berinteraksi dengan use case atau fungsi sistem [31].</p>	 <p>Actor</p>
<p>Use Case</p> <p><i>Use case</i> merupakan fungsi atau kegiatan yang dilakukan oleh aktor dalam suatu bisnis proses. <i>Use case</i> harus</p>	 <p>Use Case</p>




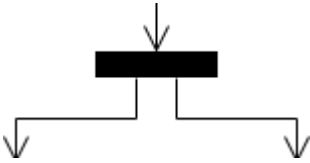
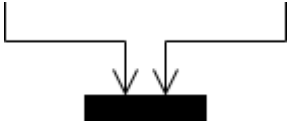
Notasi	Representasi Visual
dikaitkan dengan suatu aktor dan beberapa terdapat <i>use case</i> yang dapat berdiri sendiri [31].	
<p style="text-align: center;"><i>Communication Link</i></p> <p><i>Communication link</i> merupakan garis penghubung antara aktor dengan <i>use case</i> yang menunjukkan bahwa aktor dan <i>use case</i> berkomunikasi satu sama lain menggunakan pesan [31].</p>	
<p style="text-align: center;"><i>Extend</i></p> <p><i>Extend</i> merupakan notasi relasi suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi [31].</p>	
<p style="text-align: center;"><i>Include</i></p> <p><i>Include</i> merupakan notasi relasi suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya [31].</p>	

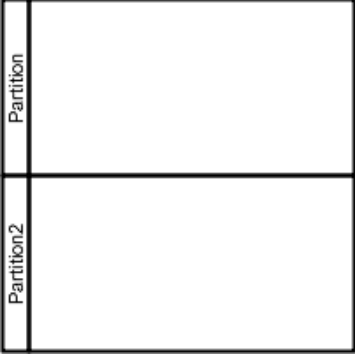
c. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas merupakan diagram yang bersifat dinamis dengan fungsi menjelaskan alur dari suatu aktivitas ke aktivitas

lainnya dalam sebuah sistem [29]. Berikut merupakan notasi dari *activity diagram* pada Tabel 2. 4.

Tabel 2. 4 Notasi *Activity Diagram*

Notasi	Representasi Visual
<p><i>Initial Node</i> <i>Initial node</i> atau node awal merupakan simbol uml yang memiliki arti awal dari suatu aktivitas diagram [32].</p>	
<p><i>Activity Final Node</i> <i>Activity final node</i> atau node akhir memiliki fungsi sebagai penutup dari suatu aktivitas diagram [32].</p>	
<p><i>Action</i> <i>Action</i> atau aksi merupakan sebuah kegiatan yang dilakukan [32].</p>	
<p><i>Fork Node</i> <i>Fork node</i> atau node cabang merupakan percabangan dari satu alur kegiatan hingga banyak alur kegiatan [32].</p>	
<p><i>Join Node</i> <i>Join node</i> atau node gabung merupakan penggabung dari banyak alur kegiatan menjadi satu alur kegiatan [32].</p>	
<p><i>Swimlane</i></p>	

Notasi	Representasi Visual
<p><i>Swimlane</i> berfungsi untuk memisahkan grup aktivitas berdasarkan dengan organisasi bisnisnya [32].</p>	

2.2.12 *Black Box Testing*

Pengujian *black box* adalah salah satu teknik pengujian perangkat lunak dengan menguji fungsional dari perangkat lunak [8]. Adapun untuk cara pengujiannya dilakukan dengan menjalankan unit atau modul, setelah itu dilihat hasil dari pengujian unit apakah sesuai dengan proses bisnis yang diinginkan [33]. Percobaan tersebut dapat dilakukan dengan mencoba inputan acak yang mungkin membuat perangkat lunak gagal berfungsi. Inputan acak dapat berupa angka, string ataupun karakter khusus yang diinputkan [8]. Pengujian sistem dilakukan dengan responden yang memiliki keahlian atau *expert* dalam bidang *software engineering* [34].