

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Pada penelitian terdahulu merupakan upaya untuk membandingkan dan menemukan inspirasi baru untuk penelitian ke depannya [38]. Pada penelitian terdahulu mencantumkan hasil dari penelitian terdahulu yang mendukung penelitian sekarang. Tabel 2.1 merupakan tabel penelitian terdahulu yang mendukung penelitian ini.

Tabel 2.1 Penelitian Terdahulu

No	Jurnal	Metode	Hasil	Kelebihan	Kekurangan
1.	Judul: Perancangan Basis data Sistem Informasi Akademik SMK Swakarya Palembang Tahun: 2019 Penulis: Meidyan Permata Putri, Evan Apriadi dan Dimas Budi Asmoro Jurnal: Jurnal Teknologi dan Informatika	<i>Database System Development Life Cycle</i>	Penelitian ini dilakukan untuk merancang basis data untuk sistem informasi SMK Swakarya Palembang yang dapat menyimpan dan menghasilkan data akademik siswa agar dapat mendukung kebutuhan informasi akademik sekolah.	- Tujuan dan kesimpulan yang dihasilkan sesuai - Hasil penelitian disertai dengan tabel dan gambar sehingga lebih mudah dimengerti	- Metode yang digunakan <i>Database System Development Life Cycle</i> tetapi tahapannya hanya menjelaskan 3 fase desain basis data (konseptual, logikal, fisik) - Tidak ada implementasi basis data yang dirancang - Tidak melakukan

No	Jurnal	Metode	Hasil	Kelebihan	Kekurangan
	(TEKNOMATIKA), Volume 9, Nomor 2				pengujian basis data
2.	<p>Judul: Perancangan Basis data Kawasan Suci Danau Tamblingan dengan Menerapkan Model Data Relasional</p> <p>Tahun: 2023</p> <p>Penulis: I Made Sugi Ardana dan Yan Mitha Djaksana</p> <p>Jurnal: Jurnal Syntax Admiration, Volume 4, Nomor 10</p>	Metode <i>Database System Development Life Cycle</i>	Penelitian ini dilakukan untuk merancang basis data yang dapat menyimpan informasi terkait kawasan suci yang terdapat pada Danau Tamblingan. Perancangan basis data dilakukan karena informasi kawasan suci ini masih disimpan dalam naskah fisik mengakibatkan informasi hilang dan naskah yang rusak.	- Tujuan dari penelitian jelas sehingga dapat dimengerti maksud dari dilakukannya penelitian ini	- Perancangan basis data hanya menjelaskan 3 fase desain basis data - Tidak ada hasil implementasi basis data yang dirancang
3.	<p>Judul: Perancangan Basis data Sistem Informasi Perwira</p>	Metode <i>Database System Development Life Cycle</i>	Penelitian ini dilakukan untuk mempermudah dan mendukung	- Tujuan dari penelitian jelas sehingga dapat dimengerti	- Tidak melakukan pengujian basis data

No	Jurnal	Metode	Hasil	Kelebihan	Kekurangan
	<p>Tugas Belajar (SIPATUBEL) pada Kementerian Pertahanan</p> <p>Tahun: 2020</p> <p>Penulis: Alvin Dwi Hardiansyah dan Catur Nugrahaeni Puspita Dewi</p> <p>Jurnal: Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA), Halaman 222 - 233</p>		<p>pengumpulan dan pelaporan data pada aplikasi SIPATUBEL dengan menggunakan metode <i>Database System Development Life Cycle</i>. SIPATUBEL dirancang untuk mendata perwira yang direkomendasi untuk belajar di luar lembaga Kemhan kemudian hasilnya diimplementasi ke dalam basis data menggunakan XAMPP dengan DBMS MySQL</p>	<p>tujuan penelitian</p> <ul style="list-style-type: none"> - Tahapan metodologi yang digunakan sesuai - Hasil penelitian dijelaskan sesuai dengan tahapan metodologi yang digunakan 	<ul style="list-style-type: none"> - Tidak melakukan <i>database design</i>
4.	<p>Judul: Perancangan Desain Basis data Sistem Informasi Geografis</p>	<p>Metode <i>Database System Development Life Cycle</i></p>	<p>Penelitian ini dilakukan untuk merancang basis data yang dapat</p>	<ul style="list-style-type: none"> - Tujuan dilakukannya penelitian jelas - Hasil penelitian 	<ul style="list-style-type: none"> - Metode perancangan basis data yang digunakan adalah

No	Jurnal	Metode	Hasil	Kelebihan	Kekurangan
	<p>Tanah Penduduk dengan Menerapkan Model Data Relasional (Studi Kasus: Desa Tumbang Mantuhe Kabupaten Gunung Mas Provinsi Kalimantan Tengah)</p> <p>Tahun: 2021</p> <p>Penulis: Tri Amri Wijaya, Constantin Menteng, Adi Surya, Afis Julianto dan Ema Utami</p> <p>Jurnal: Jurnal Teknologi Informasi (JTI), Volume 15, Nomor 1</p>		<p>mendukung kegiatan operasional desa untuk menyimpan, mengubah dan menampilkan data informasi geografis dari tanah penduduk Desa Tumbang Mantuhe.</p>	<p>dijelaskan disertai dengan gambar dan tabel sehingga lebih mudah dimengeti</p>	<p><i>Database System Development Life Cycle</i> tetapi hanya menjelaskan 3 fase desain basis data (konseptual, logikal dan fisikal)</p>
5.	<p>Judul: Perancangan Basis data Penjualan</p>	<p>Metode <i>Database System Development</i></p>	<p>Penelitian ini dilakukan untuk merancang</p>	<p>- Tujuan dari penelitian jelas sehingga</p>	<p>- Tidak melakukan pengujian</p>

No	Jurnal	Metode	Hasil	Kelebihan	Kekurangan
	<p>dengan Metode Basis data <i>Lifecycle</i> pada Toko Lancar Elektronik</p> <p>Tahun: 2022</p> <p>Penulis: Yusuf Bahtiar dan Dene Herwanto</p> <p>Jurnal: Satuan Tulisan Riset dan Inovasi Teknologi (STRING), Volume 7, Nomor 2</p>	<i>ent Lifecycle</i>	<p>basis data untuk data penjualan, pemesanan dan stok barang Toko Lancar Elektrik sehingga dapat menyimpan data dengan mudah dan cepat.</p>	<p>dapat dimengerti maksud dari dilakukannya penelitian ini</p> <p>- Metodologi yang digunakan sesuai dan dijelaskan menggunakan bahasa yang mudah dipahami</p> <p>- Hasil penelitian dijelaskan sesuai dengan metodologi yang digunakan</p>	<p>pada basis data yang dirancang</p> <p>- Tidak melakukan normalisasi</p>
6.	<p>Judul: Sistem Informasi <i>Monitoring</i> Skripsi Berbasis Web (Studi Kasus: Prodi Akuntansi Universitas Mercu Buana)</p>	Metode <i>Rapid Application Development</i>	<p>Penelitian ini dilakukan untuk membuat sistem informasi untuk memantau aktivitas pengelolaan skripsi pada Universitas Mercu Buana</p>	<p>- Tujuan penelitian jelas sehingga dapat dimengerti dengan mudah</p> <p>- Pemodelan sistem digambarkan dengan jelas</p>	<p>- Rancangan basis data hanya digambarkan melalui ERD</p>

No	Jurnal	Metode	Hasil	Kelebihan	Kekurangan
	<p>Tahun: 2020</p> <p>Penulis: Dinar Ajeng Kristiyanti dan Ahmad Mulyana</p> <p>Jurnal: Jurnal Sistem Satuan Bisnis, Volume 10, Nomor 1</p>		<p>yang bertujuan untuk meningkatkan kinerja universitas dalam mengelola aktivitas skripsi mahasiswa.</p>	<p>menggunakan <i>use case</i> diagram, <i>activity</i> diagram, <i>component</i> diagram, <i>deployment</i> diagram dan ERD</p>	

Sumber: [13][22][9][15][16][39]

Berdasarkan Tabel 2.1 dari penelitian terdahulu, ditemukan bahwa perancangan basis data merupakan salah satu tahapan penting dalam menerapkan teknologi informasi karena dapat mencari informasi lebih cepat, mengurangi adanya duplikasi atau perulangan pada data dan dapat menyimpan data dengan aman [22]. Adanya teknologi informasi, dapat menyimpan dan memudahkan untuk mempelajari informasi yang ada dalam basis data [39]. Basis data juga sudah menjadi bagian penting bagi perusahaan maupun organisasi karena dalam penggunaan sistem informasi, basis data digunakan untuk menyimpan, mengelola dan menggunakan data dalam jumlah yang besar [9][16]. Metode yang digunakan penelitian terdahulu dalam perancangan basis data adalah metode *Database System Development Life Cycle*. Berdasarkan penelitian terdahulu, metode *Database System Development Life Cycle* dapat membantu untuk memberikan gambaran dari suatu sistem yang akan dibangun sehingga, dapat memberikan manfaat untuk *developer* [15][13]. Namun dari penelitian terdahulu tersebut, belum ada yang melakukan perancangan basis data untuk pengangkutan dan pemusnahan limbah B3 dan sebagian besar tidak melakukan pengujian basis data. Oleh karena itu, penelitian ini dilakukan untuk merancang basis data logistik limbah B3 pada PT Gurindam Sinar Berkah dengan menggunakan metode *Database System*

Development Life Cycle dan basis data yang telah dirancang akan diuji untuk menemukan kesalahan dalam proses perancangan.

2.2 Basis data

Basis data atau basis data adalah susunan kumpulan data perusahaan atau organisasi yang lengkap, terorganisir dan disimpan ke dalam komputer secara terintegrasi sehingga dapat memenuhi kebutuhan informasi penggunanya [6]. Basis data merupakan struktur komputer terintegrasi yang menyimpan kumpulan data yang saling berhubungan [7]. Tujuan dari adanya basis data adalah untuk mengatur data, menjaga konsistensi data, data terintegrasi, data selalu *up to date*, menyediakan struktur informasi yang mudah dipahami dan untuk memenuhi kebutuhan informasi dari pengguna dan aplikasi tertentu [8].

2.1.1 Sistem Basis data

Sistem basis data merupakan suatu sistem yang berisikan kumpulan tabel data yang berhubungan dimana pengguna dapat memanipulasi dan mengakses tabel data yang ada. Sistem basis data memiliki beberapa komponen, yaitu [8]:

1. *Hardware* (Perangkat Keras), perangkat keras atau *hardware* yang dimaksud merupakan perangkat komputer yang mendukung proses operasi pengolahan data.
2. *Operating System* (Sistem Operasi), merupakan perangkat lunak yang mengendalikan sistem dan seluruh sumber daya komputer.
3. *Database* (basis data), suatu sistem basis data dapat berisi beberapa basis data dan setiap basis data dapat berisi sejumlah objek basis data seperti tabel, *file* dan indeks.
4. *Management System* (DBMS), dalam memasukkan dan mengambil data dari basis data diperlukan perangkat lunak seperti Basis data *Management System* (DBMS).
5. Pengguna (*User*), pengguna berinteraksi dengan sistem basis data untuk dapat memanipulasi data yang terdapat dalam basis data.

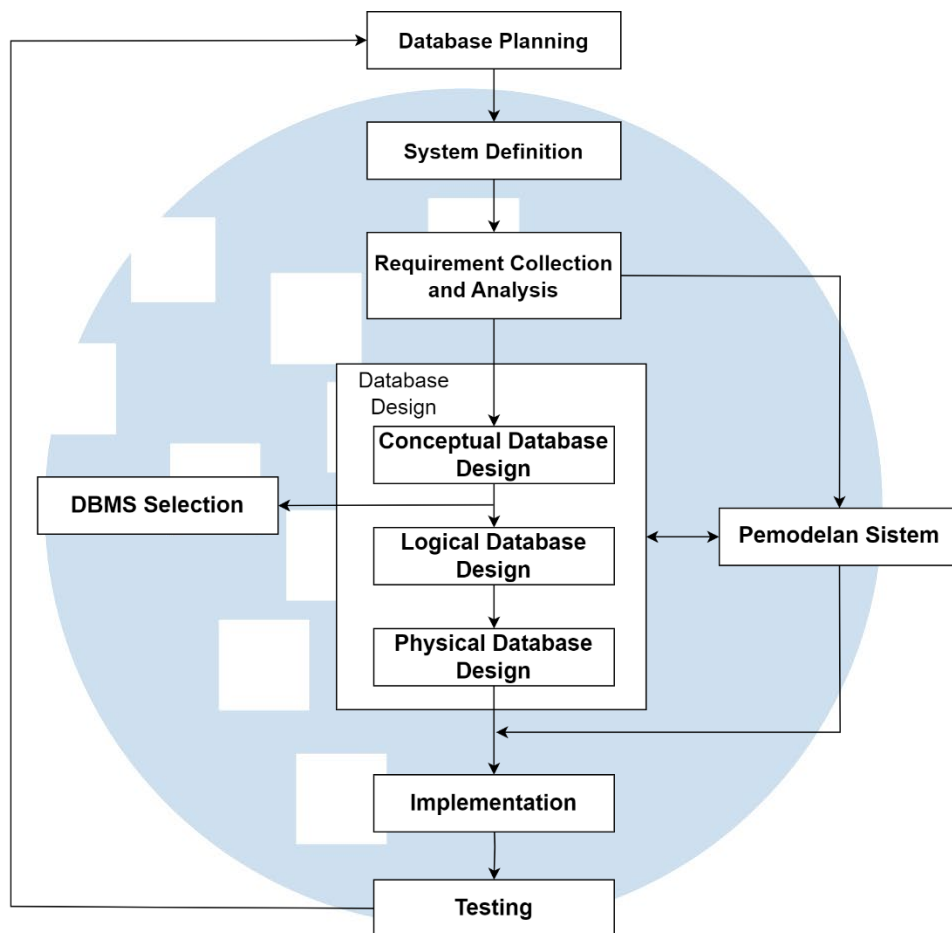
2.3 Limbah Bahan Berbahaya dan Beracun (B3)

Limbah adalah bahan-bahan pembuangan yang sudah tidak terpakai sehingga berdampak negatif bagi masyarakat jika sisa-sisa pembuangan tersebut tidak dikelola dengan baik [18]. Pada umumnya karakteristik dari limbah adalah berukuran mikro, penyebarannya memiliki dampak yang luas dan berjangka panjang pada masyarakat. Berdasarkan jenis dari karakteristik limbah terbagi menjadi 3 yaitu, karakteristik kimia (bahan organik, *Biological Oxygen Demand*, *Dissolved Oxygen*, *Chemical Oxygen Demand*), fisik (bau, suhu, warna) dan biologi (kualitas air) [18].

Limbah B3 adalah zat, energi atau komponen lain yang memiliki sifat, konsentrasi dan jumlah baik itu secara langsung atau tidak langsung yang mencemarkan, membahayakan, merusak lingkungan hidup, kesehatan serta keberlangsungan hidup manusia dan makhluk hidup lainnya [19]. Limbah dapat diidentifikasi menjadi 3 kategori yaitu, limbah B3 kategori 1, limbah B3 kategori 2 dan limbah nonB3. Limbah B3 memiliki karakteristik yang mudah meledak, mudah menyala, reaktif, infeksius, korosif dan beracun [20].

2.4 Database System Development Life Cycle (DBSDLC)

Database System Development Life Cycle dari basis data merupakan sebuah siklus yang menelusuri basis data dalam suatu sistem informasi [7]. DBSDLC merupakan salah satu siklus dari *System Development Life Cycle* yang dikhususkan untuk pengembangan *software* pada sistem basis data [9]. Berdasarkan Gambar 2.1 terdapat 7 tahapan dalam merancang basis data yaitu, tahapan *database planning*, tahapan *system definition*, tahapan *requirement collection and analysis*, tahapan *database design* yang terdiri dari *conceptual database design*, *logical database design* dan *physical database design* kemudian tahapan *DBMS selection*, tahapan *implementation* dan tahapan *testing*.



Gambar 2.1 Database System Development Life Cycle [41]

Berdasarkan Gambar 2.1, DBSDLC terdiri dari beberapa tahapan, yaitu [41]:

1. *Database planning*, tahapan ini merupakan tahapan perencanaan yang menentukan *mission statement* dan *mission objective* dalam perancangan basis data.
2. *System Definition*, tahapan ini merupakan tahapan yang menjelaskan batasan dari sistem basis data dan *user view*.
3. *Requirement Collection and Analysis*, tahapan ini merupakan tahapan yang mengumpulkan dan menganalisis informasi yang dibutuhkan dalam merancang sistem basis data.
4. *Database design*, merupakan tahapan pembuatan desain yang mendukung *mission statement* dan *mission objective* pada basis data yang dirancang. Pada *database design* terdapat 3 fase yaitu [32],

- a. *Conceptual Database Design*, dilakukan untuk membuat desain basis data berdasarkan informasi yang digunakan perusahaan tanpa bergantung pada perencanaan fisik.
 - b. *Logical Database Design*, dilakukan membangun model data berdasarkan model tertentu tetapi tidak bergantung pada DBMS tertentu dan perencanaan fisik.
 - c. *Physical Database Design*, dilakukan untuk mendeskripsikan implementasi struktur data dan indeks pada basis data.
5. *DBMS Selection*, merupakan tahapan dalam menentukan DBMS yang sesuai untuk mendukung sistem basis data yang dirancang. Pemilihan DBMS dapat dilakukan antara fase *conceptual* dan *logical database design*. Namun, pemilihan DBMS dapat dilakukan kapan saja sebelum fase *logical database design* asalkan informasi memenuhi persyaratan sistem.
6. Pemodelan Sistem, merupakan tahapan untuk desain *user interface* pada aplikasi yang digunakan untuk memproses basis data. Tahapan ini merupakan aktivitas paralel dari DBSDLC karena sering kali desain aplikasi tidak dapat diselesaikan sebelum desain basis data itu sendiri selesai dibuat.
7. *Implementation*, tahapan ini merupakan tahapan realisasi fisik dari basis data yang dirancang. Realisasi basis data dengan DBMS dapat dilakukan dengan *SQL statement* untuk membuat struktur dari basis data seperti *Data Definition Language (DDL)*, *Data Manipulation Language (DML)* dan *Data Control Language (DCL)*.
8. *Testing*, merupakan tahapan pengujian basis data yang telah dibangun untuk menemukan kesalahan dalam basis data. Melakukan pengujian pada basis data akan memastikan stabilitas, kinerja dan keamanan basis data kemudian dapat menguji operasi umum basis data seperti operasi *insert*, *update*, *delete* dan *query* untuk memastikan semua berfungsi sebagaimana mestinya.

2.5 ERD, UML, Prisma, DBEaver, MySQL

2.4.1 ERD

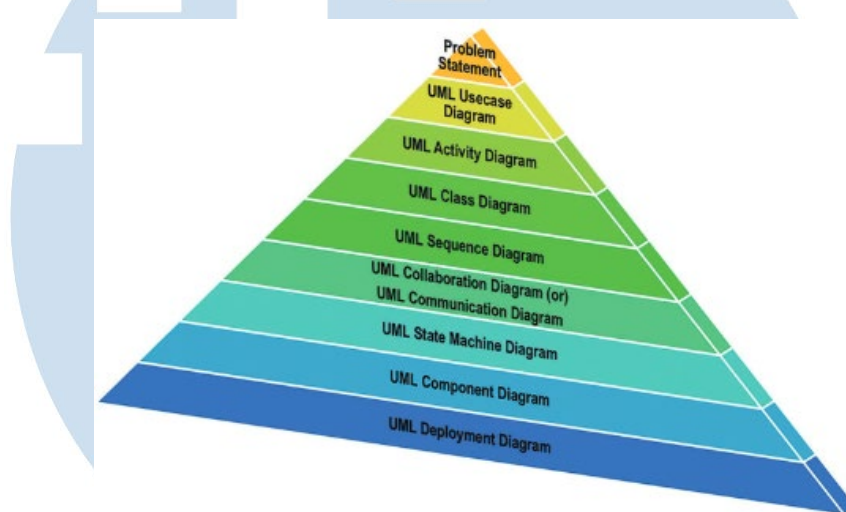
Entity Relationship Diagram (ERD) merupakan sebuah diagram yang berbentuk notasi atau sistem lambang (tanda) grafis yang digunakan dalam membuat basis data untuk menghubungkan satu data dengan data lainnya [10]. ERD juga digunakan untuk menggambarkan hubungan antar tabel yang terdapat dalam basis data [11]. Pada ERD terdapat 3 komponen [7], yaitu:

1. Entitas, dapat didefinisikan sebagai segala sesuatu yang datanya akan dikumpulkan dan disimpan. Biasanya juga dapat didefinisikan sebagai objek yang menjadi perhatian dalam basis data. Entitas dalam ERD berbentuk persegi panjang.
2. Atribut, atribut merupakan informasi yang ada dalam sebuah entitas. Setiap entitas terdiri dari sekumpulan atribut yang menggambarkan karakteristik dari entitas. Atribut dalam ERD berbentuk elips atau oval.
3. Relasi, relasi pada ERD menggambarkan hubungan antara satu entitas dengan lainnya atau lebih. Ada beberapa macam relasi yang ada pada suatu ERD [30], antara lain:
 - a. *One to one* (1:1), satu anggota entitas berelasi dengan satu anggota entitas lainnya.
 - b. *One to many* (1:M), satu anggota entitas berelasi dengan banyak anggota entitas lainnya
 - c. *Many to many* (M:M), beberapa anggota entitas berelasi dengan beberapa anggota entitas lainnya.

2.4.2 UML

Unified Modeling Language (UML) adalah salah satu pemodelan visual yang digunakan untuk memvisualisasikan, membangun, menentukan, mendokumentasikan dan menggambarkan sebuah *blueprint* dari suatu sistem perangkat lunak atau *software*. *Unified Modeling Language* (UML) dapat digunakan untuk membantu dalam memahami sebuah *software* atau perangkat lunak lebih baik lagi sehingga dapat dikembangkan oleh *developer*.

Berdasarkan Gambar 2.2, *Unified Modeling Language* (UML) terdiri dari beberapa jenis diagram, antara lain: UML *Use Case* Diagram, UML *Activity* Diagram, UML *Class* Diagram, UML *Sequence* Diagram, UML *Collaboration* Diagram, UML *State Machine* Diagram, UML *Component* Diagram dan UML *Deployment* Diagram [25].



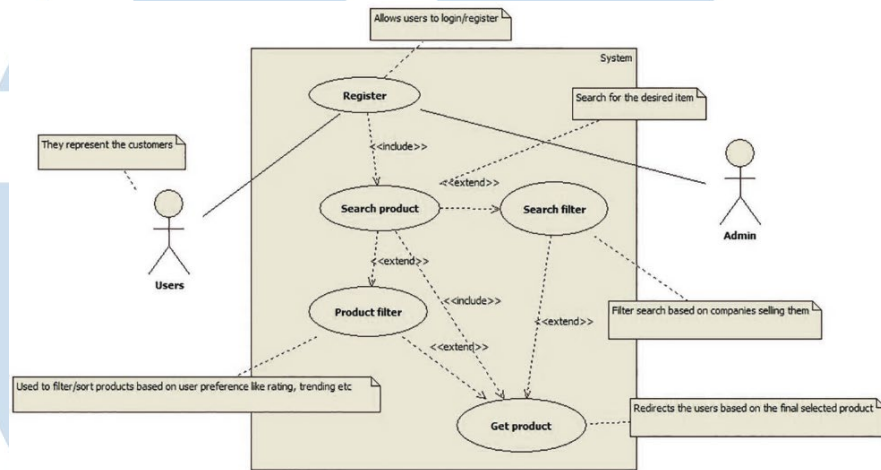
Gambar 2.2 Piramida UML Diagram [25]

1. Use Case Diagram

Unified Modeling Language (UML) *Use Case* Diagram merupakan sebuah diagram yang mendeskripsikan interaksi antara *user* dengan sistem melalui bagaimana sistem tersebut digunakan [26]. *Usecase* diagram sendiri terdiri dari 3 komponen, antara lain *system boundary*, *actor* dan *usecase*. *System boundary* menggambarkan ruang lingkup atau batasan dari suatu sistem dan merangkum fungsi dari sebuah sistem; *Actor* merupakan interaksi *user* dengan sebuah sistem dimana *actor* dapat berupa organisasi, individu atau sistem luar yang berinteraksi dengan sebuah sistem;

Use case menggambarkan visual dari fungsionalitas dari sebuah bisnis yang berbeda dalam suatu sistem. Pada *use case* diagram juga terdapat 4 relasi, antara lain *association*, *directed association*, *include* dan *extend*. *Association* merupakan relasi yang menggambarkan hubungan antar *actor* dan *use case*; *Directed association* merupakan relasi yang menggambarkan hubungan satu arah (*one-way*) dimana

actor bertanggungjawab pada *use case*; *Include* merupakan relasi yang menggambarkan sebuah situasi dimana satu *use case* mencakup fungsionalitas dari satu *use case* lainnya [25]. Gambar 2.3 merupakan salah satu contoh *use case diagram*

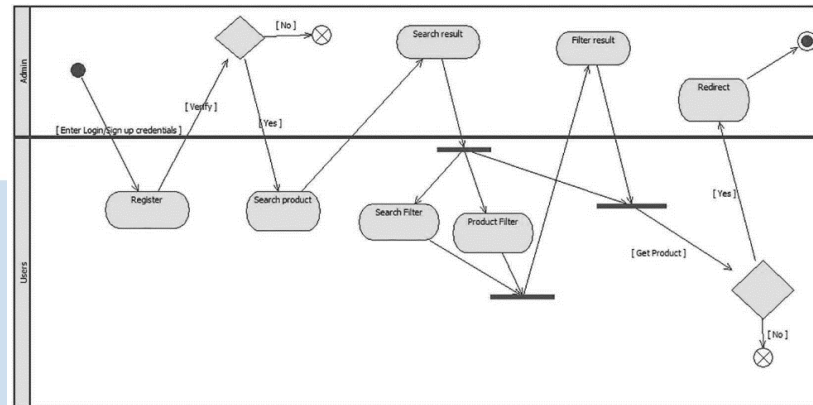


Gambar 2.3 Contoh *Use Case Diagram* [25]

2. Activity Diagram

Unified Modeling Language (UML) Activity Diagram merupakan sebuah diagram yang mendeskripsikan alur atau perilaku aktivitas pada sistem yang sedang dirancang [26]. *Activity diagram* sendiri terjadi beberapa komponen, antara lain *initial state*, *final state*, *action state*, *object*, *decision*, *merge*, *flow final*, *transition* dan *self-transition*. *Initial state* merupakan keadaan awal dari sebuah sistem; *Final state* merupakan yang mengakhiri sebuah sistem; *Action state* merupakan keadaan yang menggambarkan aktivitas atau operasi bisnis dari sebuah sistem; *Object* merupakan entitas data dari *action state*; *Decision* merupakan kondisi dari satu atau dua *input* atau banyak *output* dimana jika suatu kondisi terpenuhi maka proses akan dilanjutkan; *Merge* merupakan penggabungan *decision path* berbeda dengan menggunakan *decision node*; *Flow final* menggambarkan alur tidak normal yang dihentikan dalam *activity diagram*; *Transition* merupakan panah yang menggambarkan alur dari *source activity date* ke *target activity state*;

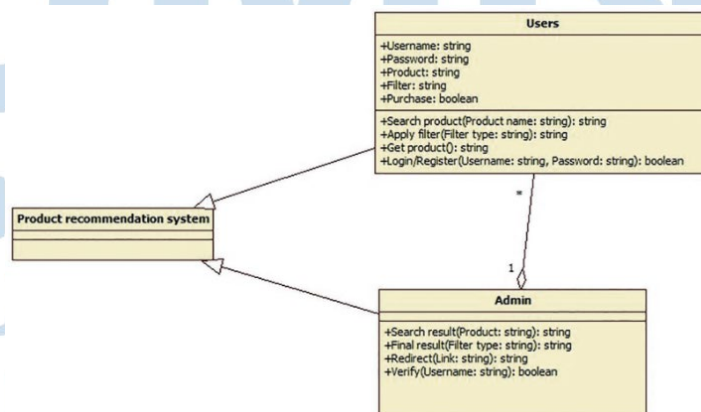
Self-transition menggambarkan alur *transition* ke *action state* itu sendiri [25]. Gambar 2.4 merupakan salah satu contoh *activity diagram*.



Gambar 2.4 Contoh *Activity Diagram* [25]

3. Class Diagram

Unified Modeling Language (UML) Class Diagram merupakan sebuah diagram yang digunakan untuk menggambarkan struktur *class* yang digunakan dari sebuah sistem dan dapat memperlihatkan hubungan dari antar *class* [24]. *Class diagram* sendiri terdiri dari beberapa komponen, antara lain *class*, *class relationship* dan *association class*. *Class* terdiri dari nama *class*, atribut (menggambarkan keadaan objek) dan *operations* (menggambarkan fungsi atau tindakan yang dapat dilakukan oleh *class*); *Class relationship* menggambarkan hubungan (*association*, *dependency*, *aggregation*, *composition*, *generalization* dan *realization*) antar *class* [25]. Gambar 2.5 merupakan salah satu contoh *class diagram*.



Gambar 2.5 Contoh *Class Diagram* [25]

2.4.3 Prisma

Prisma merupakan *open source Object Relational Mapping (ORM)* yang memungkinkan untuk mengelola dan berinteraksi dengan basis data dengan mudah. ORM merupakan pemetaan tabel yang ada pada basis data relasional dengan *class* pada bahasa pemrograman yang orientasinya adalah objek [40]. Prisma juga dapat melakukan sistem migrasi dan menyediakan GUI (*Graphical User Interfaces*) untuk melihat dan mengedit data pada basis data. Setiap *project* yang menggunakan *tools* Prisma dimulai dengan *file* skema Prisma yang memungkinkan *developer* untuk menentukan model basis data dan relasi dengan menggunakan bahasa skema Prisma. Prisma dikenal banyak oleh *developer* karena Prisma menyediakan API yang mudah digunakan dalam mengakses dan memanipulasi data dalam basis, menyediakan layanan basis data *cache query* dan pengelolaan koneksi basis data sehingga dapat meningkatkan performa sistem dan mengurangi beban server basis data, Prisma memiliki komunitas untuk *developer* berbagi pengalaman, pengetahuan dan saran serta dapat mendukung berbagai basis data seperti MySQL, PostgreSQL, SQLite dan SQL Server [21].

2.4.4 DBeaver

DBeaver merupakan *tools platform* basis data gratis untuk *developer*, basis data *administrator*, data analis dan semua orang yang bekerja dengan data [27]. DBeaver merupakan aplikasi *Graphical User Interface (GUI)* untuk mengakses dan mengelola berbagai basis data seperti MySQL, Oracle Basis data, PostgreSQL, Microsoft SQL Server, MongoDB dan lain-lain. DBeaver dapat terhubung dengan banyak basis data yang berbeda dengan menggunakan GUI yang sama, menyediakan editor SQL, dapat melakukan *import export* berbagai format *file* seperti CSV, SQL *dump*, Excel dan lain-lain [28].

2.4.5 MySQL

MySQL merupakan Basis data *Management System (DBMS)* yang bersifat *open source* yang memiliki 2 lisensi yaitu perangkat lunak bebas (*free software*) dan perangkat lunak yang penggunaannya terbatas (*shareware*). MySQL juga merupakan server basis data gratis yang memiliki lisensi GNU

General Public License (GPL) sehingga, dapat digunakan untuk kepentingan pribadi atau komersial tanpa perlu membayar lisensi [29]. MySQL masuk ke dalam jenis *Relational Basis data Management Systems (RDBMS)* yang digunakan untuk menyimpan dan mengelola data. *Relational Basis data Management System (RDBMS)* memungkinkan untuk mengimplementasikan sebuah basis data dengan tabel, kolom dan indeks. Pada awalnya MySQL dibuat oleh perusahaan Swedia dan didirikan oleh David Axmark, Allan Larsson dan Finland Swede Michael Widenis pada tahun 1994 yang kemudian diambil alih oleh Oracle pada tahun 2010. Sebuah *relational* basis data seperti MySQL mengatur data ke dalam satu tabel atau lebih dimana tipe data berhubungan secara langsung satu sama lain untuk membantu proses *structure data* dengan benar. Selain digunakan untuk mengedit, membuat, mengekstrak data dan mengelola akses *user* ke basis data, MySQL juga menggunakan sistem operasi untuk dapat mengeksekusi *relational* basis data dalam sistem penyimpanan komputer dan mengelola *user*, memberikan akses jaringan, memastikan *testing* dari integritas basis data serta membuat *backups* [12].

