

## **BAB II STUDI PUSTAKA**

### **2.1 Landasan Teori**

#### **2.1.1 Sistem**

Menurut (Agustin, 2018) Sistem merupakan kumpulan elemen yang saling terhubung dan berkolaborasi untuk mencapai berbagai tujuan. Di samping itu, definisi lain dari sistem meliputi komponen-komponen dan elemen-elemen, termasuk input (masukan), proses (pemrosesan), dan output (keluaran).

#### **2.1.2 Informasi**

Informasi merupakan hasil pengolahan data yang telah diubah menjadi bentuk yang memiliki makna bagi penerima informasi tersebut, serta memberikan nilai dalam proses pengambilan keputusan, baik yang bersifat sekarang maupun di masa yang akan datang. (Agustin, 2018)

#### **2.1.3 Sistem Informasi**

Sistem informasi merupakan rangkaian sub-sistem yang memiliki keterkaitan satu sama lain agar dapat menghasilkan sebuah informasi yang bermanfaat (Gede & Brata, 2022).

#### **2.1.4 Akuntansi**

Menurut (Hanggara, 2019) Akuntansi adalah sebuah proses identifikasi, pencatatan, dan pelaporan data atau informasi yang bermanfaat sebagai penilaian dan pengambilan keputusan.

Dalam perancangannya sistem ini akan mengikuti alur dari siklus akuntansi sederhana. Menurut (Latuconsina, 2019) siklus akuntansi merupakan proses berulang untuk merekam setiap kegiatan akuntansi dalam sebuah perusahaan. Siklus akuntansi terdiri dari pencatatan jurnal akuntansi, pemosting ke buku besar, penghitungan neraca saldo dan diakhiri dengan penyusunan laporan laba rugi.

#### **2.1.4.1 Jurnal Akuntansi**

Menurut (Yatti & Rifa'i, 2019) jurnal akuntansi adalah buku harian yang mencatat semua transaksi yang terjadi berupa pendebetan dan pengkreditan, secara kronologis, serta dilengkapi dengan penjelasan yang diperlukan. Jurnal akuntansi berfungsi sebagai bukti dan dasar untuk memasukkan data ke dalam buku besar.

#### **2.1.4.2 Buku Besar**

Menurut (Yatti & Rifa'i, 2019) buku besar adalah kumpulan akun-akun yang digunakan dalam pencatatan jurnal akuntansi. Buku besar dapat digunakan untuk menyusun laporan keuangan seperti neraca, laporan laba rugi, dan laporan keuangan lainnya.

#### **2.1.4.3 Neraca Saldo**

Menurut (Yatti & Rifa'i, 2019) neraca saldo adalah ringkasan dari seluruh saldo akun-akun dalam buku besar pada suatu periode akuntansi. Neraca saldo menjadi dasar dalam menyusun laporan keuangan lain seperti laporan laba rugi.

#### **2.1.4.4 Laporan Laba Rugi**

Menurut (Yatti & Rifa'i, 2019) laporan laba rugi adalah laporan yang berisi kumpulan akun pendapatan dan beban. Laba rugi didapatkan dari perbandingan jumlah pendapatan dan beban. Jika jumlah pendapatan lebih besar dari jumlah beban maka perusahaan mendapatkan laba, begitu juga sebaliknya jika jumlah beban lebih besar maka perusahaan mengalami kerugian.

#### **2.1.5 Sistem Informasi Akuntansi**

Menurut (Nugraha, et al., 2022) Sistem informasi akuntansi adalah suatu sistem yang menghasilkan informasi tentang hasil dari beberapa kegiatan untuk mengumpulkan, menyimpan, menyimpan dan mengolah data yang digunakan untuk mengambil keputusan.

#### **2.1.6 Web / Website**

Menurut (Manullang, Aritonang, & Purba, 2021) *Website* adalah sebuah kumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital seperti teks, gambar, video, audio, dan animasi. Halaman-halaman ini dapat diakses melalui koneksi internet. *Website* juga dapat diartikan sebagai kumpulan halaman-halaman yang menampilkan informasi dalam bentuk teks, gambar diam atau gerak, animasi, suara, atau kombinasi dari semuanya. Halaman-halaman ini saling terhubung dan membentuk satu rangkaian bangunan yang terkait dan dapat diakses melalui jaringan halaman.

(Wandira, 2022) *Web* sendiri terbagi menjadi beberapa jenis diantaranya adalah:

#### **2.1.6.1 Web Statis**

Adalah situs *web* dalam keadaan yang tidak berubah atau statis dan hanya dapat diubah secara manual dengan mengedit struktur kode yang membangun situs. Contoh *web* statis adalah pada website profil perusahaan (*company profile*) yang hanya menampilkan 1 halaman saja yang disebut *landing page*.

#### **2.1.6.2 Web Dinamis**

Situs *web* yang bertujuan untuk diperbarui sesering mungkin. Web ini terstruktur dan memiliki halaman *back-end* yang berfungsi untuk memodifikasi konten situs *web*. Contoh *web* dinamis adalah pada blog pribadi, toko online.

#### **2.1.6.3 Web Interaktif**

Merupakan situs *web* yang dapat berinteraksi dengan penggunanya ataupun antar pengguna. Contoh web interaktif adalah media sosial yang kita gunakan seperti *Facebook*, *Instagram*, dan lain-lain.

#### **2.1.7 Pemrograman Web**

Pemrograman *web* melibatkan penulisan, *markup*, dan pengkodean yang berkaitan dengan pengembangan *web*. Ini mencakup pengaturan konten *web*, skrip *server* dan klien web, serta keamanan jaringan. Dalam pemrograman *web*, klien dan *server* saling terhubung melalui kode yang memungkinkan pertukaran data, akses informasi, pembuatan antarmuka pengguna, serta langkah-langkah

keamanan yang diperlukan (Wandira, 2022). Bahasa pemrograman yang digunakan antara lain adalah:

#### **2.1.7.1 PHP**

*PHP (Hypertext PreProcessor)* adalah bahasa pemrograman skrip yang dirancang untuk membuat *website* dan bersifat *server-side* dan dinamis (Novendri, Saputra, & Firman, 2019).

*PHP* digunakan untuk membuat konten dinamis dan berinteraksi dengan basis data, menghasilkan halaman *web* berdasarkan permintaan dari pengguna. Seperti pengertiannya Pemrosesan *PHP* terjadi pada sisi *server*, artinya kode *PHP* dieksekusi di *server web* sebelum halaman *web* dikirimkan ke *browser* pengguna.

Fungsi utama *PHP* adalah untuk mengambil data dari basis data, melakukan operasi matematika, memproses formulir *web*, mengelola sesi pengguna, dan banyak lagi.

Hasil dari kode *PHP* yang dieksekusi akan menghasilkan *HTML*, yang kemudian akan ditampilkan di *browser* pengguna. *PHP* dapat ditempatkan di dalam kode *HTML* menggunakan *tag* khusus, seperti `<?php ... ?>`.

#### **2.1.7.2 Javascript**

*Javascript* adalah Bahasa pemrograman yang dapat membuat tampilan *web* lebih interaktif. Bahasa ini mendukung elemen berorientasi objek dan bersifat *client-side* (Wandira, 2022).

Berbeda dengan *PHP* yang dieksekusi di sisi *server*, *JavaScript* dieksekusi di sisi pengguna, yaitu di *browser web* pengguna. Ini memungkinkan pengembang untuk melakukan berbagai tindakan dan manipulasi di *browser* tanpa harus mengirim permintaan kembali ke *server*.

*JavaScript* digunakan untuk mengatur perilaku elemen-elemen pada halaman *web*, mengendalikan animasi, mendeteksi tindakan pengguna.

Kode *JavaScript* dapat ditempatkan langsung di dalam kode *HTML* atau dalam file terpisah yang dihubungkan dengan halaman web.

### **2.1.7.3 HTML-CSS**

Menurut (Wandira, 2022) *HTML (HyperText Markup Language)* adalah bahasa markup yang berfungsi untuk membangun *website*. Disebut Bahasa *markup* karena menggunakan *Tag Markup (<...>)* dalam penulisannya. Sedangkan *CSS (Cascading Style Sheets)* adalah kumpulan atribut yang berfungsi untuk mengatur format tampilan *website* dan dapat mengontrol banyak tampilan dokumen secara bersamaan. (Novendri, Saputra, & Firman, 2019)

### **2.1.8 Laravel**

Menurut (Hermanto, Yusman, & Nagara, 2019) *Laravel* adalah sebuah kerangka kerja (*framework*) untuk *PHP* yang dibangun dengan konsep *MVC (model view controller)*. *Laravel* dirancang untuk mempermudah dan mempercepat pengembangan aplikasi *web*. Dengan menggunakan *Laravel*, pengembang dapat membuat aplikasi *web* yang kuat, efisien, dan mudah diatur dengan lebih cepat daripada membangun semuanya dari awal.

### 2.1.9 Database

Menurut (Manullang, Aritonang, & Purba, 2021) *database* atau Basis data merupakan kumpulan data yang saling terkait yang diatur sedemikian rupa sehingga memungkinkan penggunaannya dapat menggunakan data tersebut dengan cepat dan mudah.

Menurut (Novendri, Saputra, & Firman, 2019) *MYSQL*, yang juga dikenal sebagai *SQL*, merupakan singkatan dari *Structured Query Language*. *SQL* adalah sebuah bahasa terstruktur yang secara khusus digunakan untuk memanipulasi *database*. *SQL* pertama kali didefinisikan oleh *American National Standards Institute (ANSI)* pada tahun 1986. Sementara itu, *MYSQL* adalah sebuah sistem manajemen database yang bersifat *open source*.

### 2.1.10 Software Development Life Cycle (SDLC)

*SDLC* yang dalam bahasa Indonesia berarti Siklus hidup pengembangan perangkat lunak merupakan sebuah proses untuk mengembangkan atau mengubah perangkat lunak yang melibatkan penggunaan model-model dan metodologi tertentu. Salah satu metode *SDLC* yang sering digunakan oleh para pengembang adalah metode *waterfall* (Voutama & Novalia, 2022).

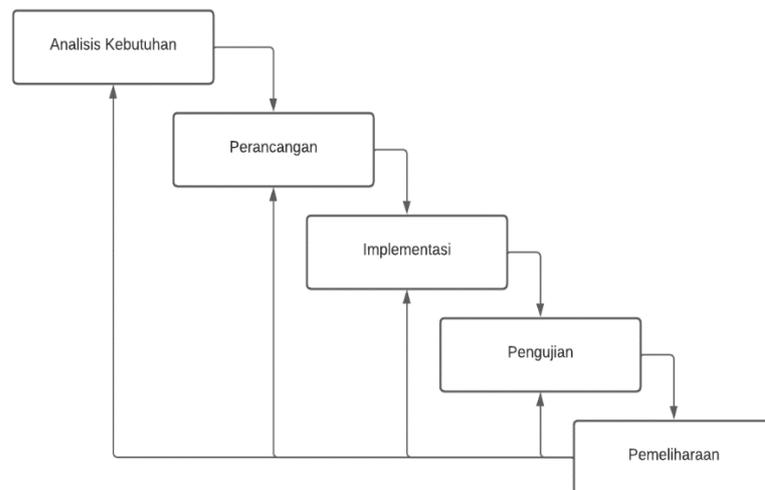
Kelebihan dari jenis metode ini adalah antara lain Menyediakan struktur yang terorganisir untuk pengembangan perangkat lunak, mendorong pembuatan dokumentasi yang baik, cocok untuk proyek besar dan terstruktur, memiliki prediktabilitas yang lebih tinggi dalam hal waktu dan biaya.

Sedangkan kekurangan dari jenis metode ini adalah Kurang fleksibel terhadap perubahan, memakan waktu lama untuk menghasilkan hasil yang

terlihat, tidak cocok untuk proyek yang dinamis atau memerlukan tanggapan cepat terhadap perubahan, risiko perubahan terlambat dan biaya tinggi jika kesalahan terdeteksi di tahap awal.

### 2.1.11 Metode Waterfall

Metode *Waterfall* atau disebut model air terjun adalah sebuah metode dalam mengembangkan sebuah *software* yang dilakukan secara berurutan setiap tahapnya. Setiap tahap harus selesai sebelum melanjutkan ke tahap berikutnya untuk mencegah pengulangan atau kembali ke tahapan sebelumnya. Dengan menggunakan model ini, pengembangan sistem dapat mencapai hasil yang diinginkan dengan lebih terorganisir dan terstruktur (Cahyadi & Susanto, 2020).



Gambar 2.1.1 Tahapan Metode Waterfall

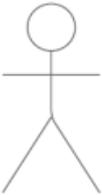
### 2.1.12 UML (Unified Modelling Language)

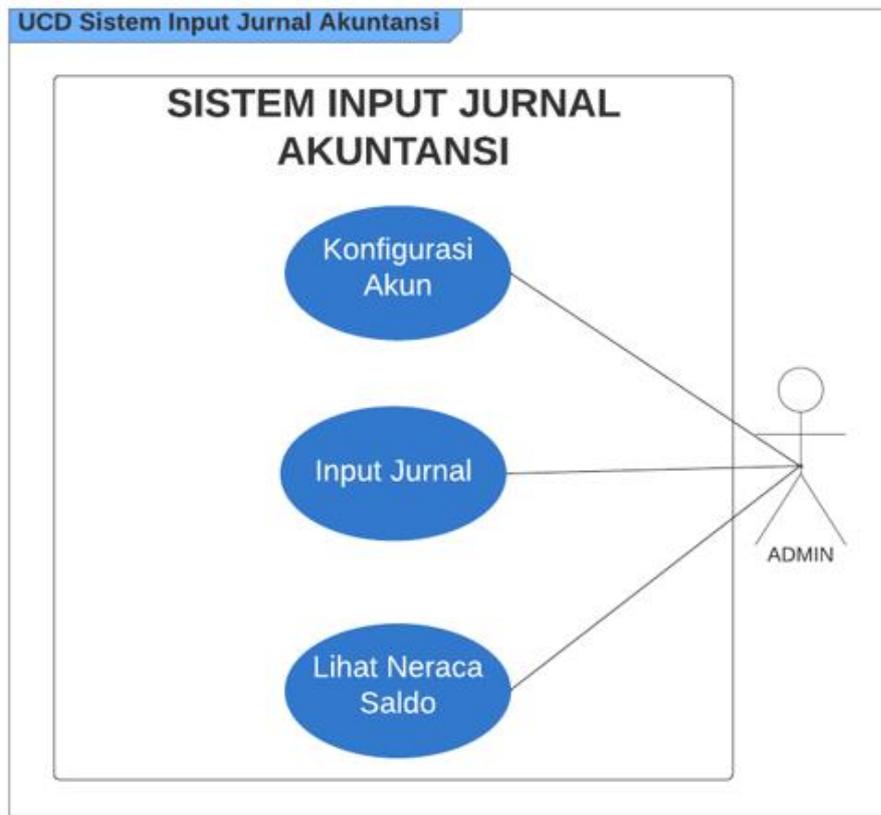
Menurut (Ciccozzi, Malavolta, & Selic, 2019) *UML* adalah sebuah alat bantu pemodelan visual yang digunakan dalam perancangan dan pengembangan perangkat lunak yang berorientasi objek. Beberapa diagram dari *UML* yang sering dan umum digunakan, yaitu:

### 2.1.12.1 Use Case Diagram

*Use case* adalah representasi fungsionalitas yang diinginkan dari suatu sistem, yang menggambarkan interaksi antara aktor (manusia atau sistem) dengan sistem yang terlibat. *Use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan tugas atau pekerjaan tertentu (Voutama & Novalia, 2022). Berikut adalah simbol-simbol yang ada pada *use case diagram* dan contoh diagramnya:

Tabel 2.1.1 Tabel Simbol Use Case Diagram

No	Simbol	Penjelasan
1		<b>Actor</b> <i>Actor</i> adalah pihak yang berinteraksi dengan sistem untuk mencapai suatu tujuan. <i>Actor</i> bisa berupa pengguna manusia, perangkat, atau sistem lainnya yang berhubungan dengan sistem yang sedang dianalisis.
2		<b>Use Case</b> <i>Use case</i> merupakan representasi dari suatu skenario atau alur kerja tertentu yang menjelaskan interaksi antara aktor dengan sistem
3		<b>Association</b> Merupakan garis lurus yang menghubungkan aktor dengan <i>use case</i> . Garis ini menunjukkan bahwa aktor tersebut terlibat dalam atau berperan dalam <i>use case</i> tersebut.
4		<b>Include Association</b> Merupakan penanda bahwa suatu <i>use case</i> adalah bagian dari fungsionalitas <i>use case</i> lainnya.
5		<b>Extend Association</b> Merupakan penanda bahwa suatu <i>use case</i> adalah tambahan fungsionalitas dari <i>use case</i> lainnya.

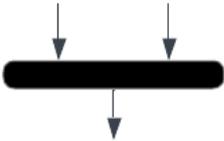
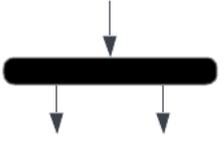


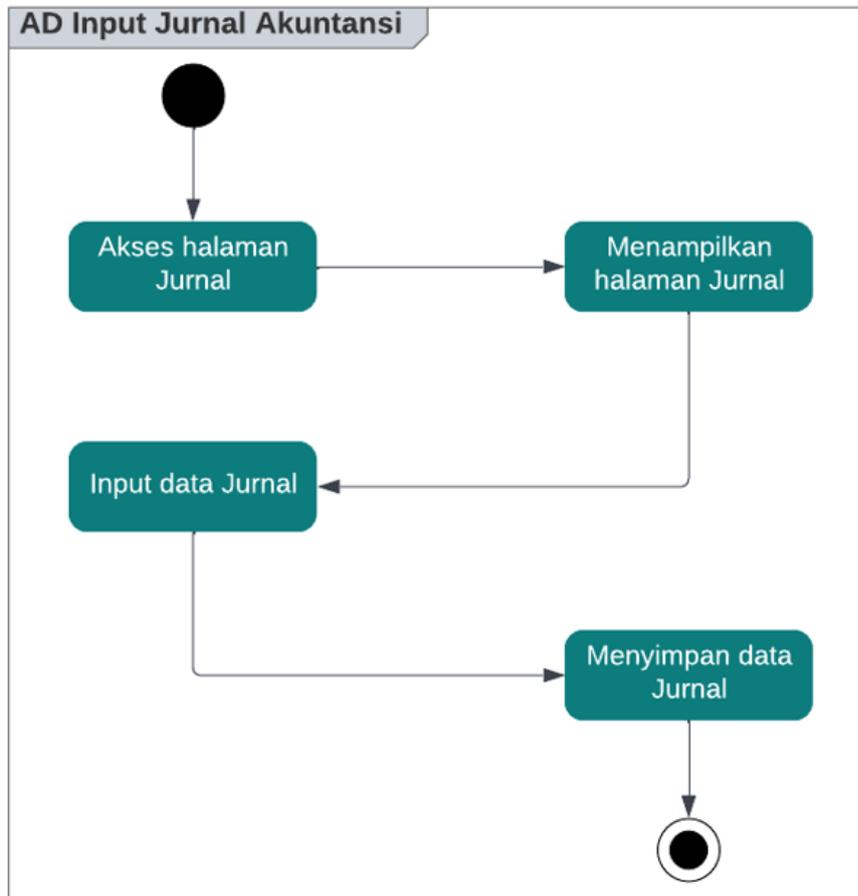
Gambar 2.1.2 Contoh Use Case Diagram

### 2.1.12.2 Activity Diagram

Menurut (Voutama & Novalia, 2022) *activity diagram* adalah representasi visual dari aktivitas-aktivitas yang terjadi dalam sistem, *Diagram* ini menggambarkan urutan langkah-langkah atau tindakan yang dilakukan oleh aktor atau sistem dalam menjalankan proses atau aktivitas tertentu. Berikut adalah simbol-simbol yang ada pada *activity diagram* dan contoh diagramnya:

Tabel 2.1.2 Tabel Simbol Activity Diagram

No	Simbol	Penjelasan
1		<b>Start Point</b> Menggambarkan awal dari aktivitas
2		<b>End Point</b> Menggambarkan akhir dari aktivitas
3		<b>Activity</b> Menggambarkan suatu proses atau kegiatan bisnis.
4		<b>Decision</b> Menggambarkan keputusan atau pilihan.
5		<b>Transition</b> Menggambarkan hubungan antara aktivitas
6		<b>Join</b> Menggambarkan penggabungan dari beberapa aktivitas menjadi satu
7		<b>Fork</b> Menggambarkan pemecahan dari satu aktivitas menjadi dua atau lebih aktivitas.

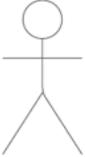


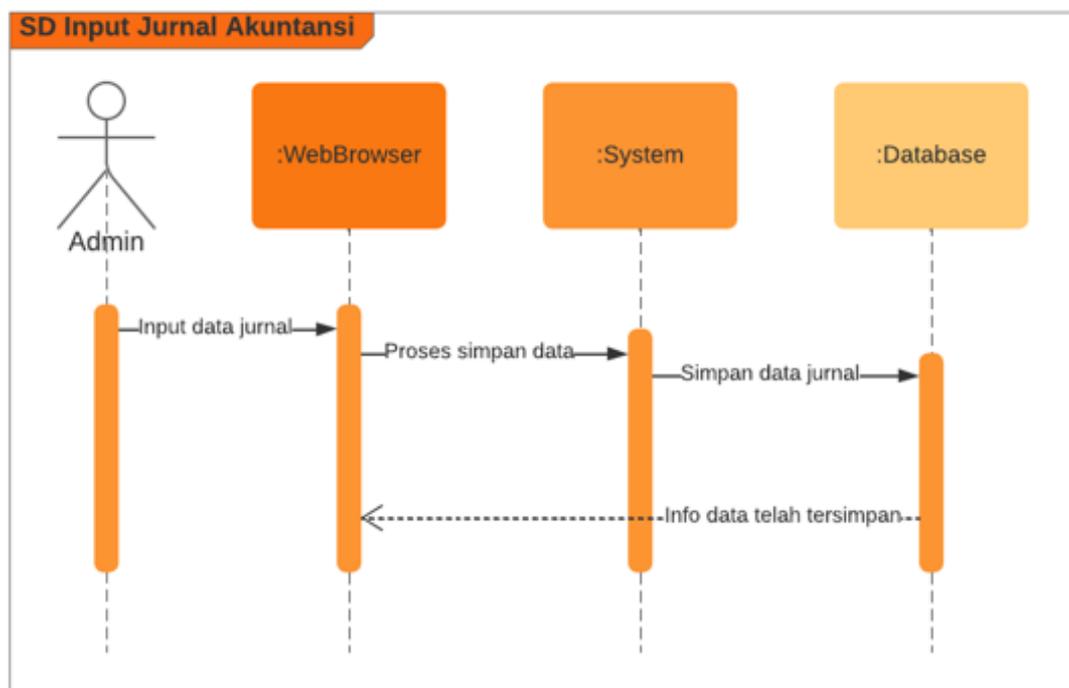
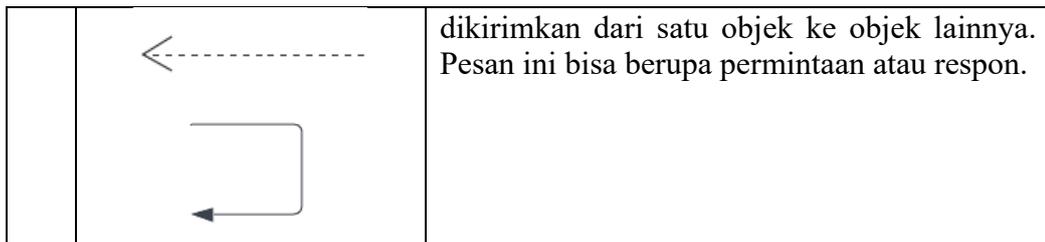
Gambar 2.1.3 Contoh Activity Diagram

### 2.1.12.3 Sequence Diagram

Menurut (Prasetya, Sintia, & Putri, 2022) *sequence diagram* adalah gambaran yang digunakan untuk menggambarkan dan menampilkan interaksi rinci antara objek-objek dalam suatu sistem. *Diagram* ini memberikan penjelasan mengenai urutan langkah-langkah atau pesan yang dikirim antara objek-objek tersebut selama proses interaksi. Berikut adalah simbol-simbol yang ada pada *sequence diagram* dan contoh diagramnya:

Tabel 2.1.3 Tabel Simbol Sequence Diagram

No	Simbol	Penjelasan
1		<b>Actor</b> <i>Actor</i> adalah entitas eksternal yang berinteraksi dengan sistem. <i>Actor</i> bisa berupa pengguna, sistem, atau entitas eksternal lainnya.
2		<b>Lifeline</b> Adalah garis vertikal putus-putus yang menggambarkan periode waktu ketika sebuah objek melakukan sebuah aktivitas atau memproses suatu pesan.
3		<b>Object</b> Objek adalah sebuah kotak yang mewakili suatu entitas atau objek dalam sistem. Objek-objek ini merupakan bagian dari sistem yang berinteraksi satu sama lain dalam proses yang sedang dianalisis.
4		<b>Activation</b> Adalah penanda sebuah objek mulai berpartisipasi didalam <i>sequence diagram</i> yang menunjukkan kapan sebuah objek mengirim atau menerima objek.
5		<b>Message</b> Adalah garis panah yang menghubungkan dua objek, menunjukkan pesan atau metode yang

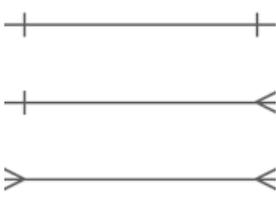


Gambar 2.1.4 Contoh Sequence Diagram

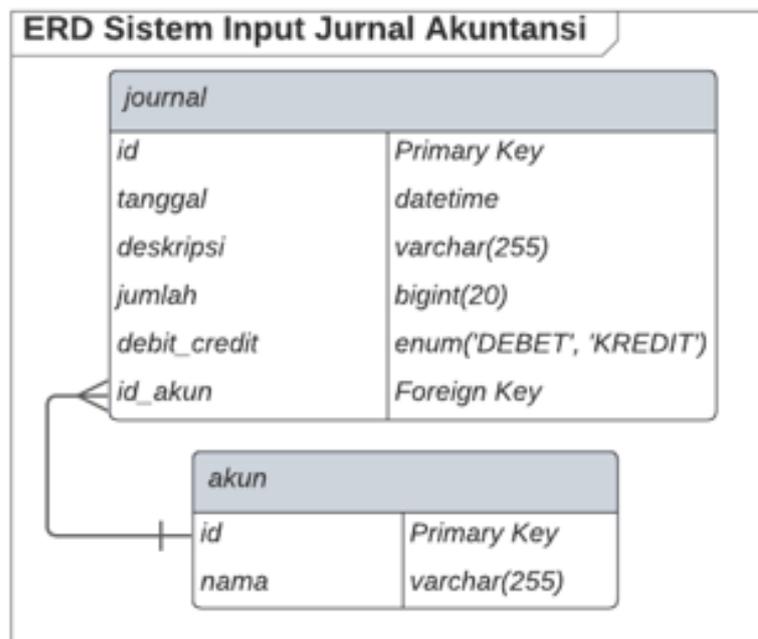
### 2.1.13 ERD (Entity Relationship Diagram)

*ERD* adalah gambaran tentang entitas, proses dan hubungan tabel-tabel didalam *database* (Putri, Voutama, & Heryana, 2023). Dengan menggunakan *ERD*, kita dapat menguji model dengan fokus pada struktur data dan mengabaikan proses yang terlibat. Berikut adalah simbol-simbol yang ada pada *sequence diagram* dan contoh diagramnya :

Tabel 2.1.4 Tabel Simbol Entity Relationship Diagram

No	Simbol	Penjelasan
1		<p><b>Entity</b></p> <p>Entitas merupakan objek atau konsep dalam dunia nyata yang ingin dimodelkan dalam basis data.</p> <p><b>Attribute</b></p> <p>Setiap entitas memiliki atribut-atribut yang menggambarkan karakteristik atau informasi yang terkait dengan entitas tersebut.</p> <p><b>Type</b></p> <p>Merupakan tipe data dari attribute-attribut terkait.</p>
2		<p><b>Relation</b></p> <p>Relasi adalah garis yang memiliki simbol disetiap ujungnya yang berfungsi untuk menghubungkan antara satu entitas dengan entitas lainnya. Simbol-simbol di setiap ujung relasi berfungsi untuk mengetahui jenis hubungan antara 2 entitas yang terhubung.</p>
3		<p><b>Relation One to One</b></p> <p>Adalah jenis relasi dimana satu entitas hanya dapat berelasi dengan satu entitas lain dan sebaliknya.</p>
4		<p><b>Relation One to Many</b></p>

		Adalah jenis relasi dimana satu entitas dapat berelasi dengan banyak entitas lainnya.
5		<b>Relation Many to Many</b> Adalah jenis relasi dimana satu entitas dapat berelasi dengan banyak entitas lain dan sebaliknya.



Gambar 2.1.5 Contoh Entity Relationship Diagram

## 2.2 Penelitian Terkait

Berikut adalah beberapa penelitian yang memiliki kaitan atau topik yang mirip dengan penelitian peneliti saat ini :

Tabel 2.2.1 Tabel Penelitian Terkait

No	Peneliti	Judul Penelitian	Hasil Penelitian
1	Andi Ilham Rahmansyah, Dedi Darwis	SISTEM INFORMASI AKUNTANSI PENGENDALIAN INTERNAL TERHADAP PENJUALAN (STUDI KASUS : CV. ANUGRAH PS	Hasil yang dicapai adalah sistem informasi akuntansi keuangan berbasis <i>web</i> yang memungkinkan karyawan dan pimpinan terhubung tanpa perlu mengirimkan laporan fisik. Sistem ini dapat mengelola data keuangan, termasuk penerimaan dan pengeluaran kas, serta menghasilkan laporan yang diperlukan oleh CV. Anugrah PS.
2	Eva Nurfitriana, Widya Apriliah. Herlina Ferliyanti. Hasan Basri. Ratnawati	IMPLEMENTASI MODEL WATERFALL DALAM SISTEM INFORMASI AKUNTANSI PIUTANG JASA PENYEWAAN KENDARAAN PADA PT. TRICIPTA SWADAYA KARAWANG	Dengan adanya sistem informasi yang terkomputerisasi, pemesanan kendaraan menjadi lebih efektif, efisien, dan mendukung kinerja yang lebih baik di PT. Tricipta Swadaya Karawang.
3	Tri Haryati, Ririn Mulyani, Siti Nurwahyuni	Penerapan Metode Waterfall Pada Sistem Informasi Akuntansi Penjualan GRC JAYABRIX Toko Bagunan Bangun Istana Sukses Cilamaya	Sistem informasi akuntansi terkomputerisasi memperbaiki kelemahan dan mempermudah proses penjualan. Pengolahan transaksi penjualan menjadi lebih cepat, mengurangi kesalahan data, dan meningkatkan kontrol dalam penyimpanan data. Selain itu, data dan bukti transaksi secara otomatis disimpan dalam <i>database</i> ,

			memudahkan proses transaksi dan pencarian data, serta meningkatkan keamanan dan keandalan.
4	Fitri Wahyuni	PERANCANGAN SISTEM INFORMASI KAS BERBASIS WEB DENGAN MENGGUNAKAN METODE WATERFALL	<i>Website</i> yang dihasilkan dalam sistem informasi akuntansi terkomputerisasi dapat meminimalkan kesalahan dalam pemrosesan data dan pengelolaan kas manual. Proses pengolahan data dan penyajian laporan menjadi lebih mudah dan cepat karena sistem yang melakukan semua transaksi dan penghitungan saldo tanpa campur tangan manusia. Selain itu, aplikasi ini juga dapat mengurangi potensi kesalahpahaman antar divisi.
5	Linda Yunita, Neneng, Auliya Rahman Isnain, Prita Dellia	Analisis Perancangan Sistem Informasi Akuntansi Pencatatan Dan Pengelolaan Keuangan Pada Yayasan Panti Asuhan Harapan Karomah	Hasil dari penelitian ini adalah sebuah sistem informasi yang dapat digunakan untuk mengolah data dan laporan data keuangan. Berdasarkan hasil pengujian, sistem yang dikembangkan dapat berjalan sesuai dengan fungsinya.