

## **BAB II**

### **LANDASAN TEORI**

#### **4.3 Profil Perusahaan**

PT Berkah Energy Alam (PT BEA) merupakan perusahaan yang resmi didirikan pada tanggal 29 Februari 2024, berdasarkan Akta Notaris Raditya Eko Hartanto, S.H., M.Kn., dengan nomor akta pendirian 16. Sebagai perusahaan yang bergerak di bidang perdagangan berbasis perizinan risiko, PT BEA memiliki fokus utama pada penjualan bahan-bahan strategis, yaitu wood pellet, batu bara, dan gypsum. Perusahaan ini telah mengantongi Nomor Induk Berusaha (NIB) 1503240002337, yang memberikan legitimasi bagi operasionalnya di seluruh wilayah Republik Indonesia.

Berkantor pusat di Perumahan Gunawangsa Gresik Regency, Kebomas, Gresik, Jawa Timur, PT BEA hadir dengan visi besar untuk mendukung pemerataan kesejahteraan melalui penciptaan lapangan kerja di sektor produksi hingga fabrikasi. Dalam menjalankan aktivitas bisnisnya, perusahaan ini secara aktif memanfaatkan sumber daya manusia lokal sebagai bentuk kontribusi nyata terhadap peningkatan perekonomian masyarakat, khususnya di wilayah operasionalnya. Dengan demikian, PT BEA tidak hanya fokus pada aspek komersial, tetapi juga berperan dalam memberdayakan masyarakat lokal menuju kemandirian ekonomi.

Sebagai perusahaan dengan skala usaha mikro dan penanaman modal dalam negeri (PMDN), PT BEA dipimpin oleh Kun Mu'min sebagai Direktur dan Umar Faruq sebagai Komisaris. Manajemen perusahaan memiliki komitmen kuat untuk terus mengembangkan strategi bisnis yang inovatif dan berkelanjutan guna memberikan kontribusi nyata dalam sektor perdagangan. Dengan memanfaatkan potensi sumber daya alam dan pengelolaan yang efisien, PT BEA berupaya menjadi salah satu pelaku industri yang berpengaruh dalam mendukung pertumbuhan ekonomi lokal maupun nasional.

Lebih dari sekadar entitas bisnis, PT BEA juga menunjukkan perhatian besar terhadap aspek sosial dan lingkungan. Hal ini tercermin dari upayanya dalam memperluas lapangan pekerjaan dan memanfaatkan sumber daya lokal secara

optimal. Dengan pendekatan strategis dan inovasi yang berkelanjutan, perusahaan ini bertekad untuk terus meningkatkan daya saingnya di industri perdagangan sekaligus memberikan dampak positif bagi masyarakat dan lingkungan di sekitarnya.

### 4.3.3 Struktur Organisasi



Gambar 1 Struktur Organisasi Perusahaan PT BEA

Struktur organisasi PT Berkah Energy Alam dirancang untuk memastikan keberlangsungan operasional perusahaan secara efisien dan efektif. Adapun struktur tersebut terdiri dari beberapa jabatan penting dengan tugas dan tanggung jawab masing-masing sebagai berikut:

#### 3. Komisaris (Kun Mu'min)

Komisaris bertanggung jawab memberikan arahan strategis kepada perusahaan, mengawasi jalannya operasional, serta memastikan kebijakan yang diambil sesuai dengan visi dan misi perusahaan. Selain itu, Komisaris juga berperan dalam mengevaluasi kinerja direktur untuk mendukung keberhasilan perusahaan.

#### 2. Direktur Operasional (Umar Faruq)

Direktur Operasional mengelola aktivitas operasional perusahaan secara keseluruhan, khususnya dalam proses produksi dan distribusi. Tugasnya meliputi perencanaan, pengawasan, dan pengendalian operasional untuk memastikan efisiensi dan tercapainya target produksi.

3. Direktur Keuangan (Mariana Napolitani)

Direktur Keuangan bertugas mengelola keuangan perusahaan, termasuk perencanaan anggaran, pengelolaan kas, serta penyusunan laporan keuangan. Perannya juga mencakup pengawasan terhadap kepatuhan perusahaan terhadap aturan perpajakan dan pengelolaan dana secara efisien.

4. Manager Produksi (Andi Saputra)

Manager Produksi bertanggung jawab mengelola semua proses produksi wood pallet dan gypsum. Ia juga memastikan jadwal produksi berjalan sesuai target dan kualitas produk terjaga dengan baik.

5. Staff Produksi (Siti Rahmawati)

Staff Produksi membantu dalam operasional produksi harian, termasuk pengoperasian mesin, pengemasan, dan memastikan produk yang dihasilkan sesuai dengan standar perusahaan.

6. Manager Gudang (Dewi Kartika)

Manager Gudang mengatur dan mengelola aktivitas gudang, termasuk distribusi, penyimpanan, dan pengelolaan stok barang. Ia juga bertugas memastikan pengiriman barang tepat waktu dan stok barang tercatat dengan akurat.

7. Staff Gudang (Eko Prasetyo)

Staff Gudang bertugas membantu pengelolaan barang di gudang, seperti penerimaan barang, penyimpanan, dan pencatatan barang keluar. Ia juga bertanggung jawab melaporkan kondisi stok barang kepada Manager Gudang.

8. Auditor (Tina Suharto)

Auditor bertugas memeriksa laporan keuangan untuk memastikan keakuratan dan transparansi data. Selain itu, ia juga mengidentifikasi risiko keuangan serta memberikan rekomendasi untuk perbaikan.

9. Analyst Keuangan (Margarita Perez)

Analyst Keuangan bertanggung jawab menganalisis data keuangan untuk mengevaluasi kinerja perusahaan. Ia memberikan rekomendasi kepada manajemen terkait strategi keuangan dan investasi.

10. Akuntan Pajak (Rachelle Beaudry)

Akuntan Pajak memastikan perusahaan memenuhi kewajiban perpajakan dengan tepat. Selain itu, ia juga menyusun laporan perpajakan dan mengoptimalkan strategi untuk efisiensi pajak.

11. Perancang Investasi (Hannah Morales)

Perancang Investasi mengidentifikasi peluang investasi yang potensial serta menyusun strategi investasi yang menguntungkan. Ia juga menganalisis risiko dan memberikan saran terkait rencana investasi perusahaan.

### 4.3.3 Visi dan Misi Perusahaan

#### Visi

Menjadi perusahaan terdepan di bidang perdagangan wood pallet dan gypsum yang terpercaya, berdaya saing global, dan berkontribusi pada keberlanjutan lingkungan serta kesejahteraan masyarakat.

#### Misi

1. Memberikan Produk dan Layanan Berkualitas Tinggi

Memastikan produk wood pallet dan gypsum memenuhi standar kualitas terbaik melalui inovasi dan efisiensi dalam proses produksi.

2. Membangun Kepercayaan Pelanggan

Menjalin hubungan jangka panjang dengan pelanggan melalui pelayanan yang profesional, transparan, dan berorientasi pada kepuasan pelanggan.

3. Mengembangkan Sumber Daya Manusia yang Kompeten

Memberikan pelatihan dan pengembangan karyawan secara berkelanjutan untuk meningkatkan profesionalisme dan kinerja perusahaan.

4. Menjalankan Operasi Berbasis Keberlanjutan

Mengimplementasikan praktik bisnis yang ramah lingkungan dengan memanfaatkan sumber daya secara bertanggung jawab.

5. Meningkatkan Daya Saing dan Inovasi

Beradaptasi dengan perkembangan teknologi dan kebutuhan pasar melalui pengembangan produk dan solusi inovatif.

6. Berkomitmen pada Pertumbuhan Ekonomi dan Sosial

Berkontribusi pada pembangunan ekonomi lokal dan nasional serta memberikan dampak positif bagi masyarakat melalui tanggung jawab sosial perusahaan.

#### 4.4 Teori

##### 4.3.3 Manajemen

Manajemen adalah suatu proses yang melibatkan perencanaan, pengorganisasian, pengarahan, dan pengendalian sumber daya organisasi untuk mencapai tujuan secara efektif dan efisien. Menurut George R. Terry, manajemen merupakan suatu proses yang khas yang terdiri dari beberapa tindakan, yaitu perencanaan, pengorganisasian, penggerakan, dan pengawasan. Hal ini menunjukkan bahwa dalam suatu organisasi, manajemen menjadi aspek penting dalam pengelolaan sumber daya agar dapat mencapai tujuan yang telah ditetapkan. (Terry, 2019)

Mary Parker Follett mendefinisikan manajemen sebagai seni dalam menyelesaikan pekerjaan melalui orang lain. Definisi ini menekankan bahwa seorang manajer memiliki peran utama dalam mengatur dan mengarahkan individu atau tim untuk mencapai hasil yang diharapkan. Oleh karena itu, manajemen tidak hanya berfokus pada pengelolaan sumber daya fisik tetapi juga mencakup pengelolaan sumber daya manusia sebagai elemen utama dalam pencapaian tujuan organisasi. (Follett, 2020)

Henry Fayol juga menyatakan bahwa manajemen adalah suatu proses yang mencakup perencanaan, pengorganisasian, pengoordinasian, dan pengawasan terhadap sumber daya yang ada agar dapat digunakan secara efektif dan efisien. Fayol menekankan bahwa setiap organisasi, baik besar maupun kecil, membutuhkan sistem manajemen yang baik untuk memastikan keberlangsungan

dan pengembangan usahanya. Konsep ini mendukung pentingnya peran manajemen dalam setiap sektor industri, termasuk dalam bidang perdagangan, manufaktur, dan jasa. (Fayol, 2018)

Secara umum, manajemen memiliki fungsi utama dalam mengoordinasikan dan mengawasi berbagai aktivitas dalam organisasi agar dapat berjalan sesuai dengan tujuan yang telah ditentukan. Proses manajemen yang baik akan membantu organisasi dalam mengoptimalkan penggunaan sumber daya, meningkatkan produktivitas, serta menghadapi persaingan bisnis dengan lebih kompetitif. Dengan memahami teori manajemen, perusahaan dapat mengembangkan strategi yang lebih baik dalam pengelolaan operasional serta peningkatan efisiensi dan efektivitas kerja. (Widharma, 2021)

### **4.3.3 Stok**

Stok merupakan persediaan barang yang disimpan oleh perusahaan untuk memenuhi permintaan pelanggan di masa mendatang. Stok memiliki peran penting dalam operasional bisnis, terutama dalam sektor perdagangan dan manufaktur, di mana ketersediaan barang yang memadai dapat memastikan kelancaran proses distribusi dan produksi. Menurut Heizer & Render (2019), stok adalah segala bentuk persediaan yang dimiliki oleh perusahaan untuk digunakan atau dijual dalam suatu periode tertentu. Persediaan ini dapat berupa bahan baku, barang dalam proses, maupun barang jadi yang siap untuk didistribusikan kepada pelanggan.

Menurut Nasution (2020), stok dapat dikategorikan ke dalam beberapa jenis, yaitu stok bahan baku, stok barang setengah jadi, stok barang jadi, dan stok suku cadang. Stok bahan baku adalah persediaan yang digunakan dalam proses produksi, sementara stok barang setengah jadi adalah barang yang sedang dalam tahap produksi dan belum menjadi produk akhir. Stok barang jadi merupakan barang yang telah selesai diproduksi dan siap untuk dijual atau dikirim kepada pelanggan, sedangkan stok suku cadang merupakan komponen cadangan yang digunakan untuk perawatan atau perbaikan produk.

Pengelolaan stok yang baik sangat penting untuk meningkatkan efisiensi operasional perusahaan. Menurut Wibowo (2021), manajemen stok yang tidak terorganisir dapat menyebabkan berbagai masalah, seperti kelebihan stok

(overstock), kekurangan stok (stockout), serta meningkatnya biaya penyimpanan dan kerugian akibat barang yang rusak atau kedaluwarsa. Oleh karena itu, perusahaan perlu menerapkan sistem manajemen stok yang efektif, seperti metode *First In First Out* (FIFO) atau *Just In Time* (JIT), untuk memastikan bahwa stok selalu tersedia dalam jumlah yang optimal dan sesuai dengan permintaan pasar.

Dalam konteks bisnis modern, sistem informasi manajemen stok berbasis teknologi sangat membantu dalam pencatatan dan pemantauan stok secara real-time. Hal ini memungkinkan perusahaan untuk mengoptimalkan pengelolaan stok, mengurangi risiko kesalahan pencatatan, serta meningkatkan efisiensi dalam distribusi barang. Dengan adanya sistem yang terintegrasi, perusahaan dapat mengambil keputusan yang lebih cepat dan tepat dalam mengelola persediaan mereka.

### **4.3.3 Manajemen Stok**

Manajemen stok merupakan proses perencanaan, pengendalian, dan pengawasan terhadap persediaan barang yang dimiliki oleh perusahaan guna memastikan ketersediaan stok yang optimal sesuai dengan kebutuhan operasional. Menurut Heizer & Render (2019), manajemen stok adalah aktivitas yang bertujuan untuk menentukan jumlah dan waktu pemesanan persediaan guna meminimalkan biaya penyimpanan sekaligus memastikan kelancaran distribusi barang. Dalam dunia bisnis, manajemen stok yang baik sangat berperan dalam meningkatkan efisiensi operasional serta mengurangi risiko kelebihan atau kekurangan stok (*overstock* dan *stockout*).

Menurut Nasution (2021), terdapat beberapa metode dalam manajemen stok yang sering digunakan oleh perusahaan, di antaranya:

- 1. Metode First In First Out (FIFO)**

Metode ini mengutamakan penggunaan barang yang lebih dahulu masuk ke gudang untuk dikeluarkan atau dijual terlebih dahulu. FIFO umumnya digunakan dalam industri dengan produk yang memiliki masa kedaluwarsa, seperti makanan dan farmasi, guna mencegah barang menjadi rusak atau tidak layak jual.

- 2. Metode Last In First Out (LIFO)**

Dalam metode ini, barang yang terakhir masuk akan lebih dahulu digunakan atau dijual. LIFO sering digunakan dalam industri yang menghadapi kenaikan harga bahan baku karena dapat membantu mengurangi beban pajak dengan menghitung biaya produksi yang lebih tinggi.

### 3. **Metode Just In Time (JIT)**

JIT adalah strategi manajemen stok yang bertujuan untuk mengurangi jumlah persediaan dengan hanya menyimpan barang sesuai dengan permintaan. Metode ini membantu perusahaan menghemat biaya penyimpanan, tetapi memiliki risiko tinggi jika terjadi keterlambatan dalam rantai pasokan.

### 4. **Economic Order Quantity (EOQ)**

EOQ adalah metode yang digunakan untuk menentukan jumlah pemesanan optimal guna meminimalkan biaya total persediaan, termasuk biaya pemesanan dan penyimpanan. EOQ membantu perusahaan dalam menetapkan kebijakan pengadaan stok yang lebih efisien.

Manajemen stok yang baik memiliki beberapa manfaat utama, seperti meningkatkan efisiensi operasional, mengurangi biaya penyimpanan, memastikan kelancaran proses produksi dan distribusi, serta meningkatkan kepuasan pelanggan dengan ketersediaan barang yang selalu siap sesuai permintaan.

Seiring perkembangan teknologi, perusahaan semakin banyak yang mengadopsi sistem informasi manajemen stok berbasis digital untuk mengoptimalkan pencatatan dan pemantauan stok secara real-time. Sistem ini memungkinkan perusahaan untuk memiliki data yang lebih akurat, mengurangi risiko human error dalam pencatatan, serta meningkatkan transparansi dalam rantai pasokan.

#### **4.3.3 Barang**

Barang adalah segala sesuatu yang berwujud dan dapat diperdagangkan, digunakan, atau dikonsumsi oleh manusia untuk memenuhi kebutuhan atau keinginan mereka. Dalam konteks ekonomi dan bisnis, barang merupakan komponen utama dalam aktivitas perdagangan dan produksi. Menurut Kotler dan

Armstrong (2018), barang adalah objek fisik yang dapat ditawarkan kepada pasar untuk memenuhi keinginan atau kebutuhan konsumen.

Barang dapat diklasifikasikan ke dalam beberapa kategori berdasarkan karakteristik dan tujuan penggunaannya. Menurut Stanton (2016), klasifikasi barang meliputi:

1. **Barang Konsumsi**

Barang yang langsung digunakan oleh konsumen akhir untuk memenuhi kebutuhan sehari-hari, seperti makanan, pakaian, dan peralatan rumah tangga.

2. **Barang Industri**

Barang yang digunakan dalam proses produksi untuk menghasilkan produk lain atau untuk operasional bisnis, seperti mesin, bahan baku, dan peralatan produksi.

3. **Barang Tahan Lama**

Barang yang memiliki masa pakai panjang dan tidak habis dalam sekali penggunaan, seperti kendaraan, perabotan, dan elektronik.

4. **Barang Tidak Tahan Lama**

Barang yang habis atau rusak setelah satu atau beberapa kali penggunaan, seperti makanan, minuman, dan produk kebersihan.

Pemahaman yang baik mengenai definisi dan klasifikasi barang sangat penting bagi perusahaan dalam mengelola operasional dan strategi pemasaran mereka. Dengan mengetahui jenis-jenis barang yang diproduksi atau diperdagangkan, perusahaan dapat menentukan pendekatan yang tepat dalam distribusi, promosi, dan penetapan harga untuk memenuhi kebutuhan pasar secara efektif.

### 4.3.3 Perancangan

Perancangan sistem adalah proses yang bertujuan untuk menciptakan rancangan suatu sistem baru yang mampu menyelesaikan permasalahan dalam suatu organisasi atau perusahaan. Proses ini dilakukan dengan mempertimbangkan berbagai alternatif solusi yang paling efektif dan efisien agar sistem yang dihasilkan dapat berfungsi secara optimal (Wahidah et al., 2021).

Dalam perancangan sistem, langkah-langkah yang dilakukan berfokus pada penyusunan model sistem berdasarkan hasil analisis yang telah dilakukan sebelumnya. Model ini mencerminkan bagaimana sistem baru akan beroperasi serta bagaimana elemen-elemen sistem akan saling berinteraksi. Berikut adalah tahapan dalam perancangan sistem:

1. **Perancangan Output**

Merancang keluaran sistem berupa laporan, tampilan, atau data yang akan digunakan oleh pengguna sebagai hasil dari proses sistem.

2. **Perancangan Input**

Menentukan data yang akan dimasukkan ke dalam sistem, termasuk format, metode pengumpulan, dan validasi untuk memastikan akurasi data.

3. **Perancangan Proses Sistem**

Mendefinisikan bagaimana data akan diproses dalam sistem, termasuk algoritma dan logika bisnis yang diterapkan untuk menghasilkan output yang sesuai.

4. **Perancangan Database**

Membuat struktur penyimpanan data yang efisien dan aman, termasuk perancangan tabel, hubungan antar data, serta mekanisme pengelolaan data.

5. **Perancangan Kontrol**

Menyusun mekanisme pengamanan dan kontrol sistem untuk memastikan bahwa sistem berjalan sesuai dengan yang diharapkan dan terhindar dari kesalahan atau penyalahgunaan.

#### **4.3.3 Tujuan Perancangan Sistem**

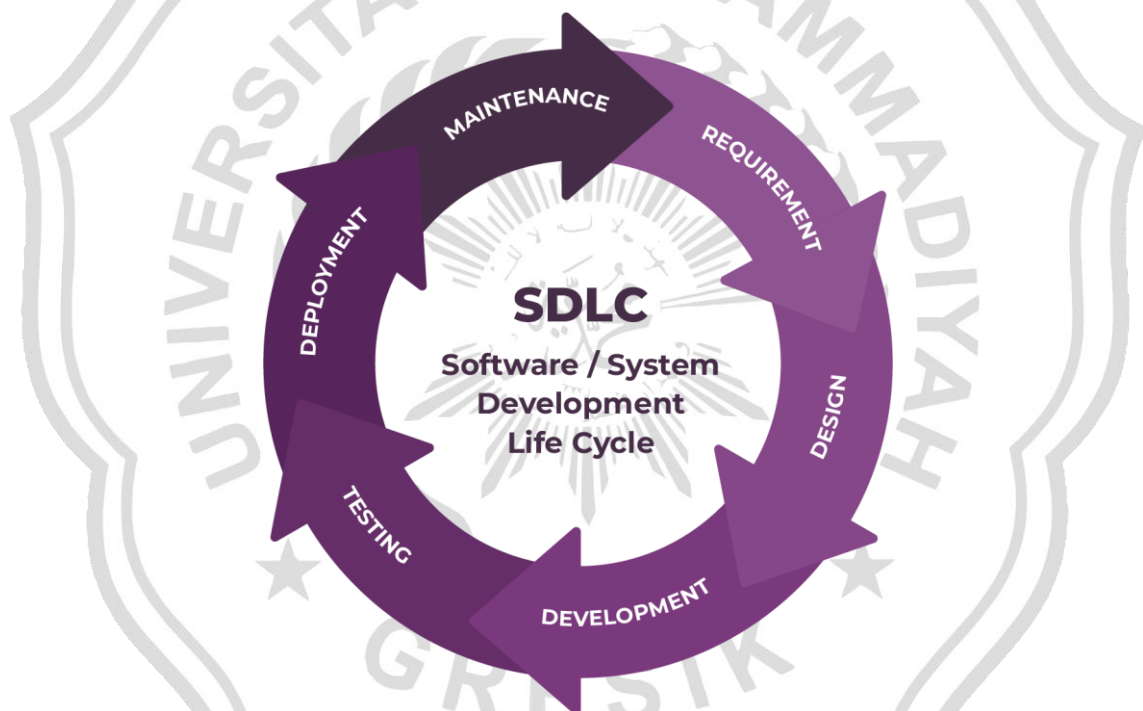
Tujuan utama dari perancangan sistem adalah untuk menghasilkan sistem yang efektif dan sesuai dengan kebutuhan pengguna. Berikut beberapa tujuan spesifiknya:

1. Memastikan bahwa sistem yang dirancang dapat memenuhi kebutuhan pengguna, baik yang bersifat eksplisit maupun implisit, dengan mempertimbangkan efisiensi sumber daya yang digunakan.

2. Memberikan rancangan yang jelas dan detail kepada tim pengembang, termasuk programmer dan teknisi, agar proses implementasi dapat berjalan dengan lancar.
3. Menyediakan solusi yang mampu menyelesaikan masalah yang menjadi fokus pengembangan sistem, sehingga meningkatkan kinerja dan produktivitas organisasi atau perusahaan.

Dengan tahapan dan tujuan yang terstruktur, perancangan sistem menjadi langkah krusial dalam pengembangan sistem informasi yang andal dan berdaya guna.

#### 4.3.3 System Development Life Cycle (SDLC)



Gambar 2 Process SDLC

SDLC (System Development Life Cycle) adalah sebuah proses yang digunakan untuk mendesain, mengembangkan, menguji, dan memelihara perangkat lunak. SDLC bertujuan untuk memberikan panduan yang jelas mengenai langkah-langkah yang harus dilakukan dalam setiap fase proyek pengembangan perangkat lunak. Dengan mengikuti SDLC, tim pengembang dapat meminimalkan risiko, mengelola waktu dengan lebih baik, dan menghasilkan perangkat lunak yang berkualitas. SDLC memastikan bahwa perangkat lunak yang dihasilkan berkualitas

tinggi, sesuai dengan kebutuhan pengguna, dan dapat dioperasikan dengan efektif serta efisien (Pressman, 2014).

Tahapan-tahapan dalam SDLC meliputi perencanaan, analisis kebutuhan, desain, pengembangan, pengujian, implementasi, dan pemeliharaan. Setiap tahap memiliki tujuan dan proses spesifik untuk memastikan perangkat lunak yang dikembangkan memenuhi kebutuhan dan ekspektasi pengguna (Sommerville, 2011).

Tahapan-tahapan dalam SDLC adalah sebagai berikut:

1. **Planning (Perencanaan):** Pada tahap perencanaan, tim proyek mendefinisikan tujuan sistem yang akan dikembangkan. Perencanaan mencakup identifikasi kebutuhan perangkat keras dan perangkat lunak, anggaran, serta jadwal pelaksanaan proyek. Selain itu, risiko-risiko yang mungkin terjadi selama pengembangan juga dianalisis dan disiapkan langkah mitigasinya (Pressman, 2014).
2. **Requirements Analysis (Analisa Kebutuhan):** Di fase ini, dilakukan pengumpulan dan analisis kebutuhan dari pengguna atau pemangku kepentingan. Informasi yang diperoleh dari analisis ini akan digunakan untuk mendefinisikan spesifikasi fungsional dan non-fungsional dari sistem yang akan dikembangkan. Dokumentasi kebutuhan menjadi acuan utama bagi pengembang dalam merancang dan membangun sistem (Sommerville, 2011).
3. **Desain (Design):** Setelah kebutuhan dianalisis, tim pengembang mulai merancang sistem yang sesuai. Fase desain terdiri dari dua jenis desain: desain arsitektur dan desain antarmuka. Desain ini harus memperhatikan efisiensi, skalabilitas, dan kemudahan pemeliharaan. Pemilihan teknologi juga dilakukan pada tahap ini (Boehm, 1988).
4. **Implementation (Implementation):** Pada fase ini, pengembang mulai menulis kode sesuai dengan desain yang telah dibuat. Pengembangan ini biasanya dilakukan dalam beberapa iterasi atau sprint, tergantung pada metodologi yang digunakan (misalnya, Agile). Setiap bagian dari sistem dibangun dan diuji secara terpisah dalam pengujian unit (McConnell, 1993).

5. **Pengujian & Integration (Pengujian & Integrasi):** Pengujian dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan kebutuhan yang telah ditentukan. Pengujian ini dapat berupa pengujian fungsional untuk mengecek kinerja perangkat lunak, pengujian beban untuk mengukur kemampuan sistem dalam menangani volume tinggi, dan pengujian kompatibilitas dengan perangkat lain (Beizer, 1990).
6. **Pemeliharaan (Maintenance):** Fase terakhir dari SDLC adalah pemeliharaan. Setelah perangkat lunak diluncurkan, fase pemeliharaan mencakup perbaikan bug, peningkatan sistem, dan penyesuaian sistem dengan perubahan kebutuhan atau teknologi (Sommerville, 2011). Pemeliharaan memastikan perangkat lunak tetap relevan dan berfungsi dengan baik dalam jangka panjang.

#### 4.3.3 Unified Modeling language (UML)

Unified Modeling Language (UML) adalah sebuah standar visualisasi yang digunakan dalam rekayasa perangkat lunak untuk merancang, mendokumentasikan, dan memahami sistem perangkat lunak secara lebih jelas. UML menyediakan serangkaian diagram yang membantu dalam merepresentasikan berbagai aspek dari sistem, baik dari segi struktur maupun perilaku (Booch et al., 2005). UML digunakan dalam berbagai tahap pengembangan perangkat lunak, mulai dari analisis kebutuhan, desain sistem, hingga dokumentasi. Dengan menggunakan UML, pengembang dapat lebih mudah mengomunikasikan ide dan konsep kepada tim proyek serta pemangku kepentingan lainnya (Rumbaugh et al., 2004).

1. Diagram Kelas (Class Diagram) menunjukkan struktur kelas dalam sistem, termasuk atribut dan metode yang dimiliki.
2. Diagram Objek (Object Diagram) memvisualisasikan instance dari kelas dalam suatu titik waktu tertentu.
3. Diagram Komponen (Component Diagram) menggambarkan bagaimana komponen dalam sistem saling berhubungan.
4. Diagram Deployment (Deployment Diagram) menunjukkan bagaimana perangkat lunak didistribusikan pada perangkat keras.

5. Diagram Paket (Package Diagram) mengelompokkan elemen-elemen dalam sistem ke dalam paket yang lebih mudah dikelola.
6. Diagram Use Case (Use Case Diagram) menggambarkan bagaimana pengguna berinteraksi dengan sistem.
7. Diagram Aktivitas (Activity Diagram) menunjukkan alur kerja atau proses dalam sistem.
8. Diagram Urutan (Sequence Diagram) memvisualisasikan interaksi antara objek dalam sistem seiring waktu.
9. Diagram State Machine (State Machine Diagram) menunjukkan transisi keadaan dalam suatu objek berdasarkan aksi yang diterima.

UML memiliki banyak manfaat, di antaranya membantu dalam analisis dan desain sistem perangkat lunak, memudahkan komunikasi antara anggota tim pengembang dan pemangku kepentingan, menyediakan dokumentasi yang jelas dan terstruktur untuk pengembangan serta pemeliharaan perangkat lunak, serta dapat digunakan dengan berbagai metodologi pengembangan perangkat lunak seperti Agile atau Waterfall. UML telah menjadi standar dalam industri perangkat lunak dan banyak digunakan dalam berbagai proyek pengembangan sistem, baik yang berskala kecil maupun besar (Fowler, 2003).

UML telah menjadi standar dalam industri perangkat lunak dan banyak digunakan dalam berbagai proyek pengembangan sistem, baik yang berskala kecil maupun besar (Fowler, 2003).

### 4.3.3 Use Case Diagram





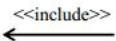
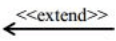
Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Table 1 Use Case Diagram

Use Case Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara aktor (pengguna atau sistem lain) dengan sistem yang dikembangkan. Diagram ini berfungsi untuk menjelaskan bagaimana suatu sistem digunakan dari perspektif pengguna dan bagaimana sistem merespons terhadap input yang diberikan (Booch et al., 2005).

Use Case Diagram membantu dalam memahami kebutuhan fungsional dari sebuah sistem dengan menampilkan skenario penggunaan utama. Diagram ini terdiri dari beberapa elemen utama, yaitu:

1. **Aktor (Actor)** adalah entitas eksternal yang berinteraksi dengan sistem, seperti pengguna, perangkat lunak lain, atau perangkat keras. Aktor bisa berupa manusia atau sistem lain yang menggunakan layanan dari sistem yang dikembangkan (Rumbaugh et al., 2004).
2. **Use Case** merepresentasikan fungsi atau layanan yang disediakan oleh sistem kepada aktor. Use case biasanya digambarkan dalam bentuk oval

dengan nama yang mendeskripsikan aktivitas yang dapat dilakukan pengguna dalam sistem.

3. **Hubungan (Association)** menunjukkan interaksi antara aktor dan use case dalam sistem. Hubungan ini digambarkan sebagai garis yang menghubungkan aktor ke use case yang relevan.
4. **Include Relationship** adalah hubungan antara use case di mana satu use case selalu memanggil use case lainnya sebagai bagian dari prosesnya. Ini digunakan untuk menghindari duplikasi fungsi dalam diagram.
5. **Extend Relationship** menunjukkan hubungan opsional di mana satu use case dapat memperluas atau menambahkan perilaku ke use case lainnya jika kondisi tertentu terpenuhi.
6. **Generalisasi (Generalisation)** hubungan hierarkis antara aktor atau antara use case. Dalam aktor, generalisasi digunakan untuk menunjukkan bahwa satu aktor adalah turunan dari aktor lain, yang berarti aktor tersebut mewarisi semua interaksi yang dimiliki oleh aktor induknya. Dalam use case, generalisasi digunakan untuk menunjukkan bahwa satu use case adalah versi lebih spesifik dari use case lain (Fowler, 2003).

Use Case Diagram banyak digunakan dalam tahap analisis dan desain perangkat lunak karena memberikan gambaran yang jelas tentang bagaimana sistem digunakan dan bagaimana interaksi antar komponen berlangsung. Dengan adanya use case diagram, pengembang dan pemangku kepentingan dapat memahami kebutuhan pengguna dengan lebih baik serta mengidentifikasi fitur utama dari sistem sebelum implementasi dimulai (Fowler, 2003).

### 4.3.3 Activity Diagram







Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Table 2 Activity Diagram

Activity Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk memodelkan alur proses atau aktivitas dalam suatu sistem. Diagram ini membantu menggambarkan bagaimana proses dalam sistem berjalan, bagaimana aliran aktivitas terjadi, dan bagaimana kondisi tertentu memengaruhi jalannya proses (Booch et al., 2005).

1. **Status Awal** adalah titik awal dalam activity diagram yang menunjukkan di mana proses dimulai. Status awal biasanya digambarkan sebagai lingkaran hitam kecil yang menunjukkan bahwa aktivitas akan segera dimulai dari titik tersebut.
2. **Aktivitas** adalah tindakan atau proses yang terjadi dalam sistem. Aktivitas digambarkan dengan bentuk persegi panjang dengan sudut membulat dan berisi deskripsi singkat tentang langkah-langkah yang dilakukan. Aktivitas dapat berupa proses manual, otomatis, atau kombinasi dari keduanya (Rumbaugh et al., 2004).
3. **Percabangan/Decision** digunakan untuk menggambarkan titik keputusan dalam suatu proses. Percabangan ini ditandai dengan bentuk belah ketupat

dan memiliki beberapa cabang keluar yang menunjukkan berbagai kemungkinan jalur yang dapat diambil berdasarkan kondisi tertentu. Setiap cabang biasanya memiliki label kondisi untuk menentukan jalur mana yang harus diambil.

4. **Penggabungan/Join** adalah titik dalam activity diagram di mana beberapa alur aktivitas bergabung kembali menjadi satu jalur sebelum proses berlanjut. Join biasanya digunakan setelah proses paralel atau ketika ada beberapa kondisi yang mengarah pada hasil yang sama (Fowler, 2003).
5. **Status Akhir** adalah titik di mana aktivitas dalam diagram selesai. Status akhir biasanya digambarkan sebagai lingkaran hitam kecil yang dikelilingi oleh lingkaran putih lebih besar, menunjukkan bahwa proses atau aktivitas telah berakhir.
6. **Swimlane** digunakan untuk membagi diagram ke dalam bagian-bagian yang merepresentasikan entitas atau peran yang bertanggung jawab atas aktivitas tertentu. Swimlane membantu dalam memahami bagaimana tugas dan tanggung jawab dibagi di antara aktor atau departemen dalam sistem yang lebih kompleks.

Activity Diagram sangat berguna dalam pengembangan perangkat lunak dan analisis proses bisnis karena memberikan gambaran visual yang jelas tentang bagaimana suatu proses bekerja dari awal hingga akhir. Diagram ini sering digunakan untuk memahami alur kerja sistem, mengidentifikasi potensi perbaikan, dan merancang otomatisasi proses dalam pengembangan perangkat lunak (Fowler, 2003).

#### 4.3.3 Entity-Relationship Diagram(ERD)

Representasi visual dari struktur data dalam suatu sistem basis data. ERD digunakan untuk memodelkan hubungan antar entitas, atribut, dan bagaimana data saling berinteraksi dalam suatu sistem. Diagram ini sering digunakan dalam tahap perancangan basis data untuk memastikan struktur yang efisien sebelum implementasi (Elmasri & Navathe, 2016).Komponen utama dalam erd sebagai berikut:

##### 1. Entitas (Entity)

Entitas adalah objek atau konsep dalam sistem yang dapat direpresentasikan dalam basis data. Entitas biasanya digambarkan sebagai persegi panjang dan dapat berupa **entitas kuat** (memiliki atribut kunci) atau **entitas lemah** (tidak memiliki atribut kunci sendiri dan bergantung pada entitas lain).

## 2. Atribut (Attribute)

Atribut adalah karakteristik atau informasi yang dimiliki oleh suatu entitas. Atribut digambarkan dalam bentuk oval dan terhubung ke entitas yang bersangkutan. Jenis-jenis atribut meliputi:

- A. **Atribut Kunci (Key Attribute):** Atribut unik yang digunakan untuk mengidentifikasi setiap instance entitas.
- B. **Atribut Multinilai (Multivalued Attribute):** Atribut yang dapat memiliki lebih dari satu nilai untuk satu entitas.
- C. **Atribut Komposit (Composite Attribute):** Atribut yang dapat dipecah menjadi sub-atribut.
- D. **Atribut Derivatif (Derived Attribute):** Atribut yang nilainya dapat dihitung dari atribut lain.

## 3. Hubungan (Relationship)

Hubungan menggambarkan bagaimana entitas-entitas dalam sistem saling berinteraksi. Hubungan digambarkan sebagai bentuk belah ketupat dan dihubungkan dengan garis ke entitas terkait. Jenis hubungan dalam ERD meliputi:

- A. **One-to-One (1:1):** Satu entitas berhubungan dengan satu entitas lain.
- B. **One-to-Many (1:M):** Satu entitas dapat berhubungan dengan banyak entitas lain.
- C. **Many-to-Many (M:N):** Banyak entitas dapat berhubungan dengan banyak entitas lain.

## 4. Kardinalitas

Kardinalitas menunjukkan jumlah maksimum hubungan yang dapat terjadi antara entitas. Kardinalitas biasanya ditulis dalam notasi (1,1), (0,1), (1,N), atau (0,N) untuk menggambarkan batasan hubungan antar entitas (Connolly & Begg, 2015).

## 5. Generalisasi, Spesialisasi, dan Agregasi

- A. **Generalisasi** adalah proses menggabungkan beberapa entitas ke dalam satu entitas induk yang lebih umum.
- B. **Spesialisasi** adalah kebalikan dari generalisasi, yaitu memecah entitas induk menjadi entitas yang lebih spesifik.
- C. **Agregasi** adalah hubungan antara hubungan, di mana suatu hubungan dapat bertindak sebagai entitas dalam hubungan lain.

ERD membantu dalam memahami struktur data dengan lebih baik sebelum implementasi basis data, memastikan efisiensi dalam penyimpanan dan pengambilan data, serta meminimalkan redundansi data (Elmasri & Navathe, 2016).

### 4.3.3 Wireframe

Representasi visual sederhana dari tata letak (layout) suatu halaman atau antarmuka pengguna dalam pengembangan aplikasi atau situs web. Wireframe digunakan untuk merancang struktur dan hierarki informasi sebelum masuk ke tahap desain visual dan pengembangan fungsional. Wireframe umumnya dibuat dalam bentuk sketsa atau menggunakan alat desain seperti Figma, Balsamiq, atau Adobe XD (Krug, 2014).

#### Komponen Utama dalam Wireframe

##### 1. Header

Bagian atas halaman yang biasanya berisi logo, judul, navigasi utama, atau elemen pencarian.

##### 2. Navigasi (Navigation Bar)

Menampilkan menu utama atau tautan yang memudahkan pengguna dalam berpindah antar halaman.

##### 3. Konten Utama (Main Content)

Bagian utama yang berisi informasi atau fitur utama dari halaman, seperti teks, gambar, video, atau elemen interaktif lainnya.

##### 4. Sidebar

Kolom tambahan yang biasanya digunakan untuk menampilkan tautan terkait, kategori, atau fitur tambahan.

##### 5. Footer

Bagian bawah halaman yang sering kali berisi informasi kontak, tautan ke halaman penting, hak cipta, dan ikon media sosial.

#### **6. Tombol & Formulir (Buttons & Forms)**

Elemen interaktif seperti tombol "Login", "Sign Up", atau formulir input untuk pencarian dan pendaftaran pengguna.

#### **7. Placeholder Gambar & Teks Dummy**

Digunakan untuk menandai posisi elemen visual seperti gambar atau paragraf yang nantinya akan diisi dengan konten sebenarnya.

### **Jenis Wireframe**

#### **1. Low-Fidelity Wireframe**

- A. Sketsa kasar atau diagram sederhana tanpa banyak detail desain.
- B. Biasanya dibuat di kertas atau alat digital sederhana.

#### **2. Mid-Fidelity Wireframe**

- A. Lebih terstruktur dengan elemen UI yang lebih jelas, tetapi masih tanpa warna atau detail grafis.
- B. Menggunakan teks dan ikon dasar.

#### **3. High-Fidelity Wireframe**

- A. Hampir menyerupai tampilan final dengan lebih banyak detail seperti font, ikon, dan elemen desain lainnya.
- B. Biasanya digunakan dalam tahap pengujian antarmuka pengguna sebelum pengembangan dimulai (Tidwell, 2020).

Wireframe sangat penting dalam proses desain karena membantu tim pengembang dan desainer memahami struktur halaman sebelum masuk ke tahap implementasi. Dengan adanya wireframe, kesalahan desain dapat diminimalkan sejak awal sehingga mempercepat proses pengembangan aplikasi atau website.

### **4.3.3 User Interface (UI)**

UI (User Interface) atau antarmuka pengguna adalah titik interaksi antara pengguna dan suatu sistem, baik itu aplikasi, website, atau perangkat lunak lainnya. UI mencakup semua elemen visual dan fungsional yang memungkinkan pengguna berinteraksi dengan sistem, seperti tombol, menu, ikon, teks, gambar, dan tata letak. Tujuan utama UI adalah menyediakan pengalaman yang intuitif, mudah dipahami,

dan efisien bagi pengguna, sehingga mereka dapat mencapai tujuan mereka dengan sedikit usaha (Norman, 1988).

UI yang baik dirancang dengan mempertimbangkan prinsip-prinsip desain seperti konsistensi, kesederhanaan, dan umpan balik visual. Konsistensi memastikan bahwa elemen-elemen antarmuka berperilaku serupa di seluruh sistem, sementara kesederhanaan membantu mengurangi beban kognitif pengguna. Umpan balik visual, seperti animasi atau perubahan warna saat tombol diklik, memberikan konfirmasi bahwa tindakan pengguna telah diterima oleh sistem (Nielsen, 1994).

UI sering kali dikaitkan dengan UX (User Experience), yang mencakup aspek yang lebih luas seperti emosi, persepsi, dan kepuasan pengguna saat menggunakan sistem. Meskipun UI fokus pada tampilan dan interaksi, UX melibatkan keseluruhan pengalaman pengguna dari awal hingga akhir. Namun, UI yang dirancang dengan baik merupakan fondasi penting untuk menciptakan UX yang positif (Garrett, 2010).

Dalam pengembangan modern, UI dirancang menggunakan alat seperti Figma, Adobe XD, atau Sketch, yang memungkinkan desainer membuat prototipe interaktif dan menguji usability sebelum implementasi. Dengan perkembangan teknologi, UI juga semakin adaptif, seperti desain responsif untuk berbagai perangkat (desktop, tablet, smartphone) dan integrasi elemen interaktif seperti animasi mikro (micro-interactions) untuk meningkatkan keterlibatan pengguna (Tidwell, 2010).

Secara keseluruhan, UI adalah komponen kritis dalam pengembangan produk digital karena berperan sebagai jembatan antara pengguna dan sistem. Desain UI yang efektif tidak hanya meningkatkan kepuasan pengguna tetapi juga mendorong penggunaan yang lebih luas dan loyalitas terhadap produk.

*(Halaman sengaja dikosongkan)*

