

BAB II

KAJIAN LITERATUR

2.1 Fase Analisis Sistem

Analisis sistem merupakan proses dekonstruksi suatu sistem informasi secara menyeluruh menjadi komponen-komponen individunya. Tujuannya adalah untuk memperbaiki sistem, mengevaluasi, serta merumuskan permasalahan, peluang, hambatan, dan kebutuhan yang ada dalam sistem tersebut [7]. Analisis sistem penting dilakukan karena dapat melakukan pemecahan masalah, kebutuhan baru, penerapan ide dan teknologi baru, dan dapat meningkatkan kinerja sistem secara keseluruhan [8].

Fungsi dari analisis sistem adalah memecahkan suatu masalah dan mencari solusi untuk menyelesaikan masalah tersebut. Fase dalam analisis sistem yang perlu dilakukan sebagai berikut [9]:

1. Identifikasi (*Identify*)

Identifikasi diperlukan untuk mengidentifikasi masalah yang ada dalam sistem seperti penyebab masalah dan keputusan yang diambil.

2. Memahami (*Understand*)

Memahami diperlukan untuk memahami Cara kerja dari sistem yang sudah ada.

3. Analisis (*Analyze*)

Analisis diperlukan untuk menganalisis sistem dan dibagi menjadi 2 tahapan yaitu:

- a. Analisis Kebutuhan

Analisis kebutuhan adalah suatu proses yang bertujuan untuk menguraikan kebutuhan pengguna terkait dengan sistem yang dirancang. Proses ini melibatkan pengumpulan data melalui wawancara, observasi dan melakukan survei. Selama proses analisis kebutuhan sistem, permasalahan yang dapat diatasi oleh sistem yang dirancang, akan diidentifikasi. Selain itu, kebutuhan sistem dibagi menjadi dua jenis yaitu kebutuhan fungsional, yang menetapkan fitur-fitur yang dibutuhkan oleh pengguna, dan kebutuhan *non-fungsional*, yang mencakup identifikasi dan spesifikasi persyaratan sistem yang tidak terkait langsung dengan fitur fungsional [10].

b. Analisis Proses

Tahap analisis proses merupakan pemodelan proses yang dilakukan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan.

4. Laporan (*Report*)

Laporan diperlukan untuk membuat laporan hasil analisis.

Dengan kata lain, analisis sistem memerlukan pemahaman mendalam tentang bagaimana elemen-elemen ini berinteraksi dan bekerja Sama untuk mencapai tujuan yang telah ditetapkan.

2.2 Fase Perancangan Sistem

Perancangan Sistem merupakan langkah yang mengikuti analisis dalam siklus pengembangan sistem. Pada konsep perancangan sistem dilakukan definisi kebutuhan fungsional serta persiapan untuk implementasi yang bertujuan untuk menggambarkan bagaimana suatu sistem yang dibentuk [11]. Tujuan dari perancangan sistem adalah memberikan sebuah gambaran perancangan sistem yang dikembangkan dan memberikan pemahaman yang jelas mengenai aliran informasi dan proses yang terjadi dalam sistem tersebut [12].

Tahap perancangan sistem meliputi langkah yang mengikuti proses dari analisis sistem yang telah ada, dengan tujuan menentukan proses dan data yang diperlukan oleh sistem baru [13]. Hal yang perlu dilakukan dalam Fase perancangan sistem adalah:

1. Perancangan *User Interface* (UI)

Perancangan *User Interface* (UI) adalah proses desain yang memungkinkan pengguna berinteraksi dengan sistem, aplikasi, atau situs *Web* secara efektif dan efisien. Tujuan dari perancangan UI adalah menciptakan pengalaman pengguna yang positif dan mudah digunakan, sehingga pengguna dapat mencapai tujuan mereka tanpa kesulitan atau kebingungan. Proses perancangan UI melibatkan beberapa langkah, termasuk analisis kebutuhan pengguna dan pengembangan desain yang sesuai. Dalam proses ini, perancangan UI harus mempertimbangkan beberapa faktor seperti karakteristik pengguna, tujuan pengguna, dan keterbatasan teknis. Selain itu, perancangan ini juga harus mempertimbangkan bagaimana pengguna akan menggunakan sistem dan bagaimana sistem akan merespons interaksi pengguna [14].

2. Perancangan *Database*

Perancangan *Database* ini melibatkan konversi model konseptual dari *Entity-Relationship Diagram* (ERD) ke dalam struktur tabel yang sesuai. Langkah-langkah utamanya mencakup identifikasi entitas, penentuan atribut untuk setiap entitas, penentuan *primary key* untuk mengidentifikasi setiap baris secara unik, dan menentukan hubungan antar entitas dengan menambahkan *foreign key*. Dengan mengikuti langkah-langkah ini, perancangan *Database* dapat menghasilkan struktur tabel yang memenuhi kebutuhan sistem dan ERD yang telah dirancang sebelumnya, sehingga memastikan integritas data dan kinerja sistem yang optimal [15].

2.3 UML (*Unified Model Language*)

Unified Modeling Language (UML) adalah suatu metode pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek [16]. UML memiliki peran yang penting dalam pengembangan perangkat lunak karena menjadi bahasa standar yang sering digunakan untuk merancang, mendokumentasikan, dan memvisualisasikan sistem perangkat lunak. UML menyediakan standar untuk merancang model sistem yang diperlukan dalam pembangunan perangkat lunak. UML juga dapat digunakan dalam berbagai perancangan sistem informasi [17].

UML memiliki fungsi untuk melakukan pemodelan seperti membantu menggambarkan hubungan interaksi antara sistem dan pengguna, menciptakan diagram yang menggambarkan aksi aktor dengan aksi dalam sistem, dan menjelaskan interaksi objek berdasarkan urutan waktu. Penggunaan UML tidak terikat pada satu metodologi tertentu, meskipun dalam praktiknya, UML sering kali digunakan dalam konteks metodologi berorientasi objek [18].

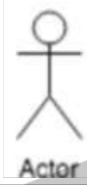




UML memiliki contoh diagram yang sering digunakan sebagai berikut [19]:

1. *Use Case Diagram*

Use Case Diagram adalah salah satu jenis dari diagram UML (*Unified Model Language*) dan sebuah modelisasi untuk menggambarkan perilaku sistem yang menjelaskan interaksi antara satu atau lebih aktor dengan informasi yang dihasilkan. Diagram ini digunakan untuk memahami bagaimana pengguna berinteraksi dengan suatu sistem. Selain itu, diagram ini juga merupakan bagian paling atas dalam fungsionalitas yang menggambarkan

bagaimana seseorang pengguna atau aktor yang menggunakan serta memanfaatkan sistem tersebut. Berikut penjelasan simbol yang digunakan pada *Use Case Diagram* pada tabel 2.1 [20].








Tabel 2. 1 Tabel Simbol *Use Case Diagram* [20]

| Notasi / Relasi | Nama | Keterangan |
|-------------------------------------------------------------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------|
|  | <i>Actor</i> | Untuk menggambarkan seseorang atau siapa saja yang berhubungan dengan sistem |
|  | <i>Use Case</i> | Untuk menggambarkan bagaimana seorang <i>actor</i> menggunakan sistem |
|  | Relasi Asosiasi | Untuk menggambarkan hubungan antara <i>actor</i> dan <i>Use Case</i> |
|  | Relasi <i>include</i> | Digunakan jika satu <i>Use Case</i> menggunakan fungsionalitas yang disediakan <i>Use Case</i> lain |
|  | Relasi <i>extends</i> | Digunakan jika satu <i>Use Case</i> menggunakan fungsionalitas yang disediakan <i>Use Case</i> lain |

2. *Activity Diagram*

Activity Diagram merupakan sebuah diagram aktivitas yang digunakan dalam *Unified Model Language* (UML) untuk menggambarkan berbagai proses dalam suatu sistem yang direncanakan. *Activity Diagram* memiliki fungsi dalam pemodelan proses dan analisis dan desain sistem [21]. Berikut Tabel 2.2 dibawah ini merupakan penjelasan simbol dari *Activity Diagram*:

Tabel 2. 2 Tabel Simbol *Activity Diagram* [20]

| Notasi / Relasi | Nama | Keterangan |
|-------------------------------------------------------------------------------------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Activity</i> | Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi dengan yang lainnya. |
|  | <i>Action</i> | Menunjukkan eksekusi dari suatu aksi |
|  | <i>Initial Node</i> | Awal dari <i>object</i> dibentuk |
|  | <i>Activity Final Node</i> | Akhir dari <i>object</i> dibentuk |
|  | <i>Decision</i> | Menggambarkan suatu keputusan atau tindakan pada kondisi tertentu |
|  | <i>Line Connector</i> | Digunakan untuk menghubungkan satu simbol dengan simbol lain. |
|  | <i>Database</i> | Menggambarkan interaksi proses aplikasi dengan data dalam basis data seperti memperbarui, mengambil, menyimpan, dan menghapus data |

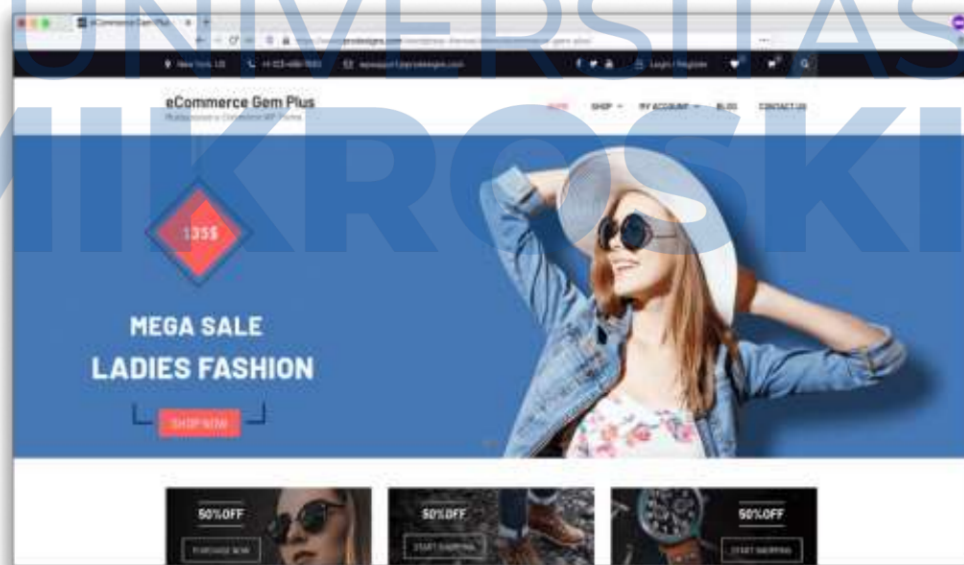
2.4 Aplikasi

Aplikasi adalah program komputer yang dirancang untuk melakukan fungsi tertentu dan menyediakan layanan kepada pengguna. Aplikasi dapat mencakup berbagai jenis, mulai dari aplikasi *desktop* hingga aplikasi *Mobile*, dan dari aplikasi *Web* hingga perangkat lunak khusus industri. Aplikasi dapat berguna untuk memenuhi berbagai kebutuhan, seperti pengolahan kata, pengeditan foto, akses ke *platform* media sosial, atau bahkan pengelolaan sistem informasi perusahaan. Dalam konteks teknologi modern, aplikasi sering kali menjadi langkah utama bagi pengguna untuk berinteraksi dengan perangkat lunak dan teknologi digital secara umum [22].

Dalam beberapa jurnal, aplikasi telah digunakan sebagai sarana untuk meningkatkan kualitas pendidikan dan pengajaran. Sebagai contoh, aplikasi yang digunakan untuk mengelola sistem operasi Windows dapat membantu dosen dalam mengajar sistem operasi kepada mahasiswa. Selain itu, aplikasi juga dapat digunakan untuk mengelola data, seperti aplikasi yang digunakan untuk mengelola absensi siswa. Dalam beberapa kasus, aplikasi juga dapat digunakan untuk mengelola jurnal, seperti aplikasi yang digunakan untuk membuat jurnal digital. Di sisi lain, aplikasi juga memainkan peran penting dalam mengatasi tantangan logistik dalam pembuatan sistem informasi dan manajemen logistik. Aplikasi khusus telah dikembangkan untuk memfasilitasi proses logistik, mulai dari manajemen persediaan hingga pelacakan pengiriman. Penggunaan aplikasi ini dapat meningkatkan efisiensi operasional dan meminimalkan kesalahan manusia dalam manajemen logistik [23].

2.4.1 Aplikasi *Web*

Web adalah sebuah sistem informasi yang dapat diakses melalui internet dan berisi informasi, konten, dan aplikasi yang dapat diakses secara online. *Web* dapat diakses melalui perangkat lunak *Web browser*, seperti *Google Chrome*, *Mozilla Firefox*, dan *Microsoft Edge*. *Web* dapat dikelola oleh individu, organisasi, atau pemerintah, dan dapat berisi informasi tentang berbagai macam topik, seperti informasi umum, informasi berita, informasi produk dan layanan, informasi pendidikan, dan lain sebagainya [24].



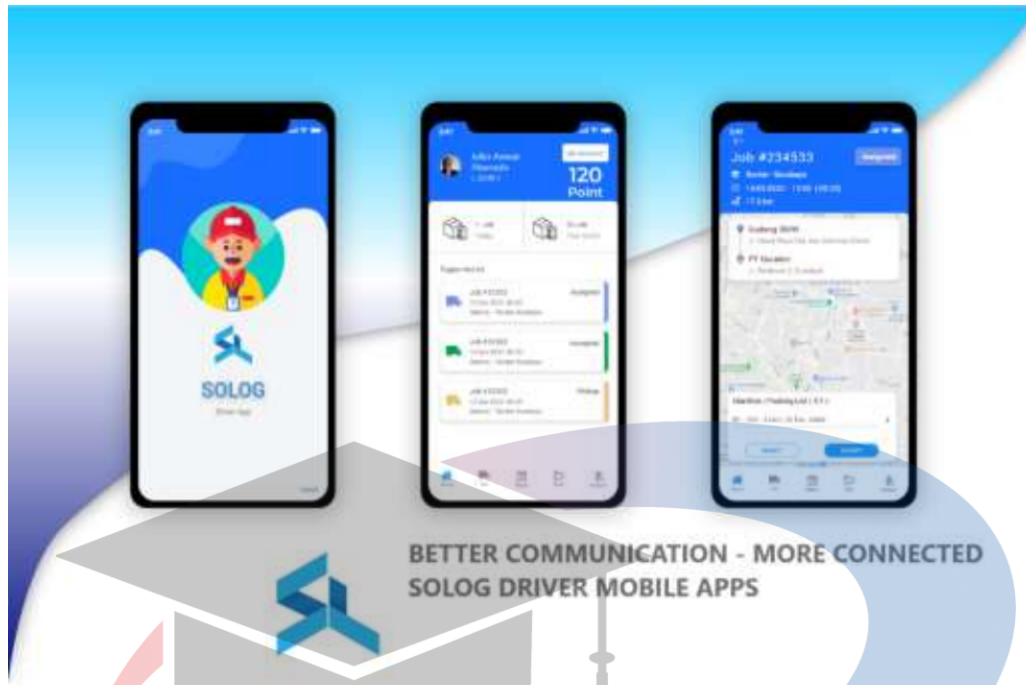
Gambar 2. 1 Gambar Aplikasi *Web* [25]

Aplikasi *Web* adalah program komputer yang dapat diakses melalui *browser Web* tanpa memerlukan instalasi pada perangkat. Dengan aksesibilitasnya melalui koneksi internet, Aplikasi *Web* memungkinkan pengguna untuk mengaksesnya dari berbagai lokasi. Karakteristiknya mencakup sifat berbasis *Web* yang diakses melalui *browser*, serta mendukung *multi-platform* untuk perangkat komputer, *smartphone*, dan tablet. Aplikasi ini interaktif dengan penggunaan teknologi seperti *JavaScript* dan *HTML5* untuk memberikan pengalaman yang dinamis kepada pengguna, sementara kontennya yang dapat diperbarui tanpa perlu melakukan instalasi ulang aplikasi [25].

Aplikasi *Website* dapat beroperasi melalui jaringan lokal (LAN) maupun internet. Fitur utamanya, seperti data yang terpusat dan akses yang mudah, menjadikan aplikasi *Web* semakin populer dan mudah diterapkan dalam berbagai aspek kehidupan. Aplikasi *Website* juga memungkinkan pengembang untuk memperbarui konten dan fitur secara terpusat tanpa memerlukan intervensi dari pengguna akhir. Dengan demikian, aplikasi *Website* selalu memiliki versi terbaru dan paling aman, serta memastikan bahwa pengguna selalu memiliki akses ke fitur terbaru dan terbaik. Selain itu, aplikasi *Website* dapat diintegrasikan dengan berbagai sistem lain, seperti sistem manajemen *Database*, sistem keamanan, dan sistem analisis data, untuk meningkatkan efisiensi operasional dan kualitas layanan kepada pengguna [26].

2.4.2 Aplikasi *Mobile*

Mobile adalah istilah untuk perangkat portabel, kecil, dan nirkabel yang mendukung komunikasi. Perangkat ini memiliki kemampuan untuk bergerak dan berpindah, dan aplikasi *Mobile* adalah istilah yang digunakan untuk merujuk pada aplikasi yang berjalan di perangkat *Mobile*. *Mobile* bukan lagi sebatas alat komunikasi, tetapi telah berkembang menjadi Gaya hidup dan menjadi bagian integral dari kehidupan modern. Perkembangan *Mobile* terus mengantarkan kita menuju era konektivitas yang semakin canggih dan membuka peluang-peluang baru yang tak terbayangkan sebelumnya [27].



Gambar 2. 2 Gambar Aplikasi *Mobile* [28]

Aplikasi *Mobile* adalah perangkat lunak yang berguna untuk memudahkan pengguna dalam berbagai aktivitas, dapat dioperasikan melalui perangkat bergerak seperti *smartphone*, tablet, dan sejenisnya. Aplikasi *Mobile* sangat diminati oleh masyarakat karena berbagai kemudahan yang ditawarkan, mulai dari keterjangkauan informasi hingga kemampuan untuk menyelesaikan tugas-tugas sehari-hari dengan cepat dan efisien. Hal ini menciptakan terobosan baru dalam berbagai jenis aplikasi yang digunakan dalam kehidupan sehari-hari, mencakup berbagai bidang seperti komunikasi, belanja, kesehatan, pendidikan, dan hiburan. Dengan demikian, aplikasi *Mobile* telah menjadi bagian integral dari gaya hidup modern, mengubah cara manusia berinteraksi dengan teknologi dan lingkungan sekitarnya [28].

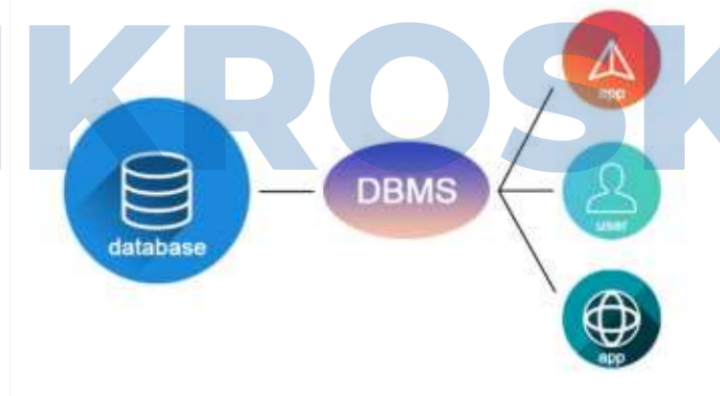
Aplikasi *Mobile* memiliki beberapa karakteristik penting yang meliputi berbagai aspek. Dari segi *hardware*, aplikasi *Mobile* dipengaruhi oleh ukuran layar, kebutuhan daya, interaksi dengan pengguna, dan sensor. Dari segi *software*, aplikasi ini mengutamakan interaksi yang intuitif, desain UI/UX yang menarik, menu aplikasi yang mudah dinavigasi, serta kemampuan untuk berinteraksi dengan sumber data atau informasi. Dalam hal komunikasi, aplikasi *Mobile* harus memiliki konektivitas jaringan yang handal. Dari sudut pandang pengembangan, fokus pada konsep dan implementasi ide-ide baru yang bernilai sangatlah penting. Terakhir, dari perspektif

pengguna, aplikasi *Mobile* harus responsif, dapat dipersonalisasi, dan menyediakan notifikasi atau umpan balik yang efektif [29].

2.5 Basis Data (*Database*)

Basis Data atau *Database* merujuk pada kumpulan data yang diatur sesuai dengan peraturan tertentu atau struktur tertentu sehingga memudahkan pengelolaan dan akses informasi. *Database* terdiri dari berbagai file, tabel, atau koleksi yang saling terhubung dan disimpan dalam format digital [30]. *Database* memberikan manfaat dalam menyimpan data karena formatnya yang bersifat mandiri dan dapat disesuaikan dengan fleksibilitas [31].

Beberapa contoh platform *Database* seperti *MySQL*, *Sql Server*, *Ms.Access*, *Oracle*, dan *PostgreSQL*. *Database* memiliki fungsi untuk mempermudah identifikasi data, meminimalisir suatu data ganda, mempermudah pengguna dalam penginputan data baru, melakukan penyimpanan secara digital, menjadi cara lain dalam masalah penyimpanan ruang dalam suatu aplikasi. *Database* memiliki beberapa jenis seperti *Operational Database*, yang digunakan untuk melakukan penyimpanan data yang sangat detail agar dapat mudah digunakan, *Relational Database* yang digunakan untuk mencari informasi dalam tabel yang beragam, *Distributed Database* yang digunakan untuk menyebarkan data secara merata sambil menjaga keterkaitannya dan memungkinkan akses bersama. *External Database* yang dimanfaatkan untuk tujuan bisnis karena kemudahannya dalam akses yang ditujukan untuk penggunaan publik [32]. Berikut merupakan gambar *Database Management system (DBMS)* sebagai berikut:



Gambar 2. 3 Gambar *Database Management System (DBMS)* [33]

DBMS (*Database Management System*) adalah perangkat lunak yang dirancang untuk menyimpan, mengelola, dan mengakses *Database* secara efisien, aman, dan terorganisasi. DBMS

juga memungkinkan untuk membuat, membaca, memperbarui, dan menghapus data dalam *Database* secara terstruktur dan dapat mendukung akses pengendalian untuk memastikan hanya pengguna tertentu yang dapat mengakses data. Fungsi utama pada DBMS yaitu menyederhanakan pengelolaan data yang terlalu kompleks, mendukung aplikasi *Web*, layanan *cloud* dan sebagainya [33].

2.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang digunakan untuk merancang sebuah basis data dan digunakan untuk mewakili hubungan atau relasi antara entitas dan objek yang terlihat beserta atributnya [34]. ERD memiliki fungsi sebagai alat untuk membantu dalam pembuatan basis data dan memberikan gambaran bagaimana basis data atau *Database* dapat beroperasi. Pada ERD terdapat tiga konsep utama yaitu [35]:

1. Entitas

Entitas adalah objek yang menjadi fokus dalam sebuah basis data. Entitas bisa berupa manusia, tempat, benda, atau kondisi yang berkaitan dengan data yang diperlukan. Simbol entitas biasanya berbentuk persegi panjang.

2. Atribut


Atribut adalah garis yang menghubungkan atribut satu dengan yang lain untuk menunjukkan hubungan antar entitas dalam ERD. Selain itu, garis juga dapat menunjukkan alur atau aliran data dalam ERD.

3. Relasi

Relasi merupakan hubungan antara dua atau lebih entitas yang berasal dari himpunan yang berbeda dan saling berkaitan. Relasi terbagi menjadi beberapa jenis seperti Relasi *one*, relasi *many*, relasi *Zero or many*, relasi *one and only one*, dan relasi *one or many*. Berikut penjelasan

Komponen-komponen pembentuk ERD dapat dilihat pada tabel 2.3 dibawah ini [36]:

Tabel 2. 3 Tabel Komponen ERD [36]

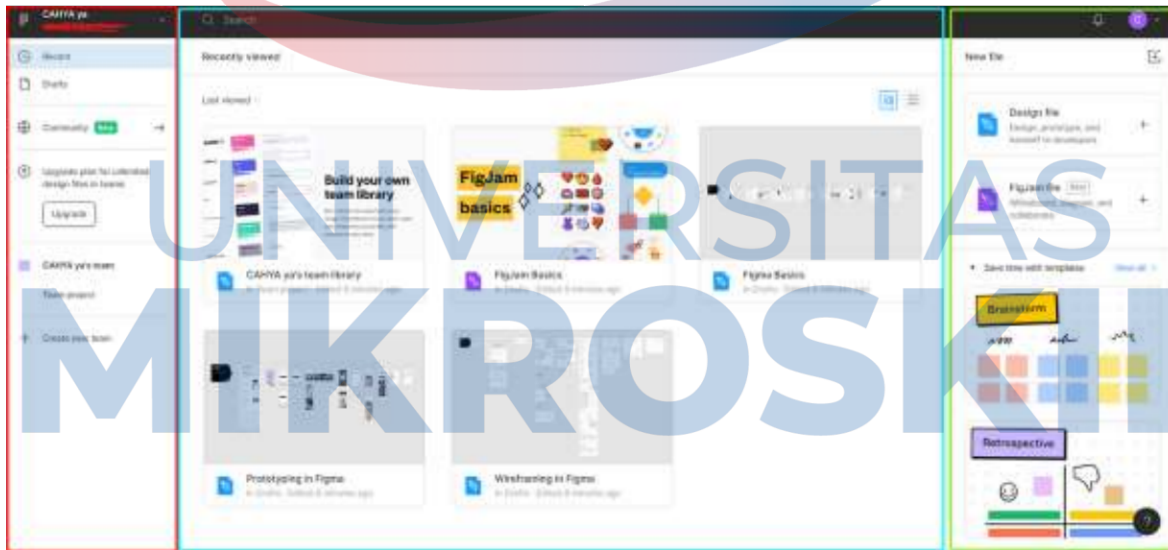
| Notasi | Komponen | Keterangan |
|-------------------------------------------------------------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Entitas | Segala sesuatu yang dapat digambarkan oleh data dan objek yang dapat diidentifikasi secara unik dan berbeda dengan yang lain. Entitas dapat memiliki beberapa atribut yang menjelaskan karakteristik entitas tersebut. |
|  | Atribut | Elemen yang mendeskripsikan karakteristik dari entitas. Atribut yang menjadi kunci entitas atau <i>key</i> diberi garis bawah. |
|  | Relasi | Komponen yang menghubungkan antara beberapa entitas. |
|  | Relasi One | Setiap entitas dalam satu tabel berhubungan dengan tepat satu entitas dalam tabel lain. |
|  | Relasi Many | Setiap entitas dalam satu tabel dapat berhubungan dengan banyak entitas dalam tabel lain. |
|  | Relasi Zero Or Many | Setiap entitas dalam satu tabel dapat tidak memiliki hubungan atau dapat berhubungan dengan banyak entitas dalam tabel lain. |
|  | Relasi One and Only One | Setiap entitas dalam satu tabel berhubungan dengan tepat satu entitas dalam tabel lain tanpa pengecualian |
|  | Relasi One Or Many | Setiap entitas dalam satu tabel harus berhubungan dengan setidaknya satu entitas dalam tabel lain dan bisa berhubungan dengan lebih dari satu entitas. |

2.7 Figma

Figma merupakan salah satu *platform* perangkat lunak *design tool* dan alat *prototyping* yang digunakan untuk membuat produk digital dan juga digunakan untuk membuat desain aplikasi berbasis *Mobile*, *Web*, dan juga *desktop*. Figma memungkinkan pengguna dapat berkolaborasi

secara *real-time* dan bekerja Sama dimanapun dan kapanpun melalui internet [37]. Dalam konteks Figma, alat desain berbasis *Web* ini memiliki fitur *prototyping* yang memungkinkan pengguna untuk melihat model dari sistem yang telah dibuat. *Prototype* adalah contoh atau model awal dari sebuah sistem atau aplikasi yang digunakan sebagai referensi untuk pengembangan lebih lanjut. Dalam desain, *prototipe* dibuat sebelum dikembangkan secara lebih lanjut atau sebelum diproduksi secara massal. Tujuan utama dari *prototipe* adalah untuk mendapatkan *Feedback* dan memahami bagaimana sistem tersebut berfungsi, sehingga dapat dikembangkan lebih efektif [38].

Figma juga menonjol karena kemampuannya untuk terintegrasi dengan berbagai alat dan *platform* lain, yang membuat alur kerja desain menjadi lebih efisien. Integrasi dengan alat seperti *Slack*, *Microsoft Teams*, dan *Asana* yang memungkinkan tim menjaga komunikasi dan manajemen proyek tetap terkoordinasi. Selain itu, Figma menyediakan API yang memungkinkan pengembang membuat *plugin* dan alat kustom yang dapat disesuaikan dengan kebutuhan spesifik proyek. Dengan komunitas pengguna yang aktif dan sumber daya edukasi yang melimpah, Figma terus berkembang dan berinovasi, menjadikannya salah satu pilihan utama bagi desainer profesional di berbagai industri [39]. Berikut merupakan contoh gambar dari aplikasi *Figma*:



Gambar 2. 4 Gambar Aplikasi Figma [40]

Figma memiliki fitur utama yang tersedia pada aplikasinya seperti [40]:

1. *Design* yang digunakan untuk memudahkan pengguna dalam mendesain tampilan *Web* atau aplikasi, beberapa fitur *design* yaitu *modern pen tool*, *OpenType*, *Auto Layout*, dan *Plugins/Widgets*.
2. *Prototyping* yang memungkinkan pengguna untuk melihat prototype desain atau model dari sistem yang telah dibuat, termasuk *Interactions*, *Mobile-viewing*, *Dynamic Overlays*, dan *Animated GIFs*.
3. *Design System* yang dapat memudahkan pengguna dalam aksesibilitas dan elemen pendukung lainnya.
4. *Library dan Aset* yang dapat menyediakan aksesibilitas dan elemen pendukung lainnya kepada pengguna.

2.8 Manajemen Logistik

Logistik adalah suatu proses yang terencana dan sistematis untuk mengatur aliran barang dan informasi dari titik awal atau sumber hingga ke konsumen akhir. Tujuan utama dari logistik adalah mencapai efisiensi dalam pengurangan biaya dan waktu pengantaran, memastikan barang yang tepat sampai pada waktu yang tepat, menjaga ketersediaan barang, serta meningkatkan kepuasan pelanggan melalui layanan logistik yang baik [41]. Proses dalam logistik meliputi aktivitas-aktivitas seperti pergerakan, pengadaan, dan penyimpanan barang atau material, serta ketersediaan produk yang terhubung dalam satu alur informasi yang saling terkait. Seluruh proses ini melintasi organisasi atau perusahaan dengan tujuan memaksimalkan keuntungan melalui pengeluaran yang efektif dengan memenuhi pesanan yang ada [42]. Sedangkan, manajemen adalah sebuah proses untuk mengatur sesuatu yang dilakukan oleh sekelompok orang atau organisasi untuk mencapai tujuan organisasi dengan bekerja sama memanfaatkan sumber daya yang dimiliki. Proses ini melibatkan beberapa langkah penting, yaitu: perencanaan, pengorganisasian, pengarahan, dan pengendalian. Perencanaan melibatkan penetapan tujuan yang ingin dicapai dan merumuskan strategi untuk mencapainya. Pengorganisasian melibatkan penyusunan struktur organisasi dan pembagian tugas kepada anggota organisasi. Pengarahan melibatkan pemberian instruksi dan motivasi kepada anggota organisasi agar dapat bekerja dengan baik. Pengendalian melibatkan pemantauan kinerja organisasi dan memastikan bahwa organisasi mencapai tujuannya [43].

Manajemen logistik merupakan serangkaian fungsional yang bertujuan untuk mengatur material, termasuk perencanaan dan penentuan kebutuhan, alokasi anggaran untuk pengadaan, penyimpanan, distribusi, pemeliharaan, penghapusan, dan pengendalian. Tujuan utama dari manajemen logistik adalah untuk menyediakan barang dan material dalam jumlah yang sesuai, tepat waktu, dalam kondisi yang layak dan di lokasi yang dibutuhkan, dengan biaya total yang minimal. Manajemen logistik juga memiliki fungsi utama untuk mencapai efisiensi maksimal dalam rantai pasok dan memastikan kepuasan pelanggan yang melibatkan pengelolaan persediaan dengan tepat, transportasi yang efisien dan pengiriman yang tepat waktu [44].

Indikator keberhasilan dalam manajemen logistik mencakup perencanaan yang baik, pengadaan bahan baku yang tepat waktu, penyimpanan yang efektif, dan pendistribusian yang efisien. Selain itu, manajemen logistik memerlukan penggunaan teknologi yang tepat untuk memantau dan mengelola aliran bahan baku, produk, dan jasa. Teknologi ini memungkinkan perusahaan untuk meningkatkan visibilitas terhadap inventaris, mengoptimalkan rute distribusi, dan mengurangi biaya operasional. Penggunaan teknologi seperti sistem manajemen gudang (*Warehouse Management System/WMS*) dan sistem pelacakan pengiriman sangat penting dalam manajemen logistik. Implementasi teknologi ini membantu perusahaan dalam memantau dan mengendalikan aliran bahan baku, produk, dan jasa. Dengan visibilitas yang lebih baik terhadap inventaris dan kemampuan untuk mengoptimalkan rute distribusi, perusahaan dapat mengurangi biaya operasional melalui otomatisasi dan efisiensi proses [45].

UNIVERSITAS
MIKROSKIL