

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

2.1.1. Sistem

System memiliki arti sebagai sebuah himpunan dari bagian yang bertautan untuk menjangkau suatu sasaran yang spesifik. Dibawah ini adalah beberapa teori sistem yang diambil dari penelitian yang berbeda dan ide-ide yang berpotensi beda:

1. *System* ialah kumpulan unsur yang berkorelasi dengan maksud menggapai *target* yang telah ditetapkan. Elemen-elemen yang ada pada sistem diartikan sebagai subsistem. Semua subsistem yang berkaitan melalui koneksi yang signifikan dapat beroperasi secara efisien dan efektif (Tukino, 2020).
2. Sistem yaitu himpunan dari faktor-faktor saling bergantung untuk membangun satu unsur terpadu yang dapat mencapai tujuan yang sudah ditetapkan. Maka dari itu sistem juga dapat disebut sebagai rangkaian objek bersambungan yang dapat dipandang bagaikan kesatuan yang diciptakan untuk memenuhi tujuan tertentu (Andrianof, 2018).
3. Pengertian sistem berhubungan dengan *component* ialah sekumpulan data yang bertautan untuk memenuhi suatu tujuan, sedangkan sistem berhubungan dengan *procedure* ialah sebuah jejaring yang bekerja terstruktur melalui tata cara saling bertautan buat menciptakan aktivitas yang telah memiliki tujuan (Silalahi, 2021).

Maka menurut pernyataan pengertian *system* yang telah diuraikan, dapat diambil kesimpulan bahwa *system* yaitu himpunan dari beberapa bagian yang saling terhubung untuk mendapatkan hasil dengan tujuan yang diinginkan.

2.1.2. Informasi

Informasi merupakan kelompok fakta yang berisi tentang sesuatu yang dapat digunakan untuk pengambilan suatu keputusan. Namun sebuah informasi yang diterima harus akurat dan tidak mengandung data yang menyesatkan agar dapat bermanfaat bagi pengguna.

Informasi menurut (Maydianto, 2021) adalah data yang relevan dan memiliki nilai yang tinggi bagi pengguna guna untuk memudahkan pengambilan suatu keputusan. Informasi ialah sesuatu yang mengandung arti yang sangat penting yang harus terbebas dari kesalahan dan harus menyimpan nilai penuh yakni *accuracy*, *timeliness* dan *relevance* untuk mempermudah dalam kegiatan mengambil suatu keputusan.

Data adalah salah satu komponen dari informasi. Data merupakan kejadian nyata yang mendefinisikan suatu peristiwa yang memiliki makna tersendiri. Maka informasi dapat diartikan sebagai data yang terolah menjadi lebih berbobot, sehingga memiliki sesuatu yang lebih bermanfaat bagi penggunanya (Andrianof, 2018).

Ada 3 hal yang mempengaruhi kualitas dari sebuah informasi:

1. *Accuracy*

Informasi harus dipastikan terbebas dari kesalahan yang bisa menyebabkan kesalahpahaman.

2. *Timeliness*

Informasi yang disampaikan kepada pengguna diharuskan sampai pada waktunya. Informasi merupakan suatu landasan dalam pemungutan sebuah keputusan.

3. *Relevance*

Informasi harus memiliki nilai yang bermanfaat bagi penggunanya.

2.1.3. Sistem Informasi

Information Systems ialah himpunan terdiri dari *hardware* dan *software* yang beroperasi bersama dengan tujuan untuk pemungutan sebuah ketetapan. Sistem informasi juga melibatkan manusia (*brainware*) untuk mendukung kegiatan operasional dan menghasilkan informasi yang bermanfaat.

Sistem informasi adalah perkumpulan dari beberapa elemen yang berinteraksi untuk menyimpan, mengelola, dan menyediakan informasi mengenai sebuah pengetahuan dan produk digital (Silalahi, 2021).

Menurut (Sihombing, 2018) *system* adalah unsur-unsur yang mempunyai suatu persatuan dan *information* ialah sesuatu yang bermanfaat sebagai alat penetapan keputusan yang terolah dari data dan dapat berguna pada saat ini dan kelak, maka *information systems* adalah sesuatu yang digunakan untuk menyampaikan informasi sehingga bermanfaat bagi penerimanya.

Sistem informasi merupakan kombinasi dari *software*, *hardware*, dan *brainware* dalam pengolahan data yang dapat menghasilkan informasi guna pengambilan keputusan. Sistem informasi juga dapat diartikan sebagai proses mengolah data menjadi informasi untuk mengambil sebuah keputusan (Tulodo & Solichin, 2019).

Dari beberapa teori yang telah diuraikan diatas, dapat disimpulkan sistem informasi yaitu gabungan elemen yang berinteraksi dan kerja bersama mengolah data menjadi informasi yang bermanfaat untuk mengambil sebuah keputusan.

2.1.4. Aplikasi

Application merupakan sebuah *software* komputer yang melakukan pekerjaan atau aktivitas secara spesifik yang dioperasikan oleh pengguna atau perangkat lunak lainnya berdasarkan fitur yang telah diberikan kedalam aplikasi. Aplikasi juga memiliki manfaat untuk berbagai macam bidang yang salah satunya adalah membuat pekerjaan menjadi lebih efisien.

Perangkat lunak aplikasi menurut (Fadhli, Islamadina, & Apriliansyah, 2019) adalah sebuah *software* yang menggunakan kecerdasan dari komputer untuk menjalankan perintah yang akan diberikan oleh pemakai. Salah satu contoh dari *software* aplikasi adalah aplikasi pengolah kata dan aplikasi pemutar media.

Application juga memiliki arti sebagai *software* yang memiliki tugas untuk menjalankan suatu tugas tertentu melalui perintah yang diberikan oleh pemakai aplikasi seperti penerapan dan penambahan data. Aplikasi juga diartikan sebagai bagian dari kelas *computer software* yang mengandalkan kekuatan *computer* untuk

menjalankan tugas yang diberikan oleh pengguna (Hartati, Kristiana Dewi, Puastuti, Muslihudin, & Setio Budi, 2017).

2.1.5. Aplikasi *Mobile*

Mobile application adalah *software* aplikasi yang dapat mempermudah mengerjakan berbagai aktifitas melalui *gadget* seperti telepon pintar atau *smartphone*. Keuntungan menggunakan aplikasi *mobile* yakni dapat mempermudah pengguna dalam melakukan aktivitas seperti *game*, mendapatkan informasi, menyelesaikan suatu pekerjaan, berselancar di internet, dan lainnya (Surahman & Setiawan, 2017).

Secara umum *mobile application* adalah sistem aplikasi yang pengoperasiannya menggunakan perangkat *mobile* contohnya ponsel pintar & tablet *PC* yang dibangun untuk mempermudah aktivitas serta meningkatkan fleksibilitas pengguna. Dalam mengembangkan aplikasi *mobile*, terdapat beberapa pertimbangan dasar terkait jenis aplikasi yang akan dikembangkan. Pertimbangan dasar tersebut dapat mempengaruhi proses pengembangan yang disebut dengan *mobile solutions* yang dibagi menjadi 3 kelompok yaitu aplikasi *mobile native* yang pengembangannya ditujukan untuk aplikasi *mobile* tertentu dengan bahasa program spesifik sesuai dengan sistem operasi yang digunakan, Kedua adalah aplikasi *mobile web* yang merupakan *mobile application* berbasis laman web yang dapat dibuka menggunakan *software browser* tanpa terbatas oleh sistem operasi apapun, Ketiga adalah *Hybrid Mobile Applications* yang merupakan aplikasi *mobile* gabungan antara aplikasi *Native* dan *Web* dan pengembangan aplikasi *Hybrid*

menggunakan bahasa pemrograman *mobile web* dan bahasa pemrograman *native* untuk fitur lainnya (Razi, Mutiaz, & Setiawan, 2018).

2.1.6. Pengembangan Sistem

Pengembangan sistem ialah teknik merancang dan membangun ulang sistem pengganti dengan tujuan menutup dan memperbaiki *problem* yang ada pada sistem saat ini. Dengan adanya pengembangan sistem, sistem yang telah dikembangkan akan mendapatkan manfaat positif baik dari segi internal maupun eksternal.

Menurut (Binardo, 2021) pengembangan sistem adalah proses menyusun sistem baru yang dapat menggantikan sistem yang sedang berjalan secara keseluruhan dengan tujuan memperbarui sistem lama. Sebuah sistem yang terdapat pada sebuah perusahaan digunakan untuk menyelesaikan masalah-masalah yang sering terjadi. Maka pengembangan sistem sangat dibutuhkan untuk mengantisipasi munculnya permasalahan lainnya.

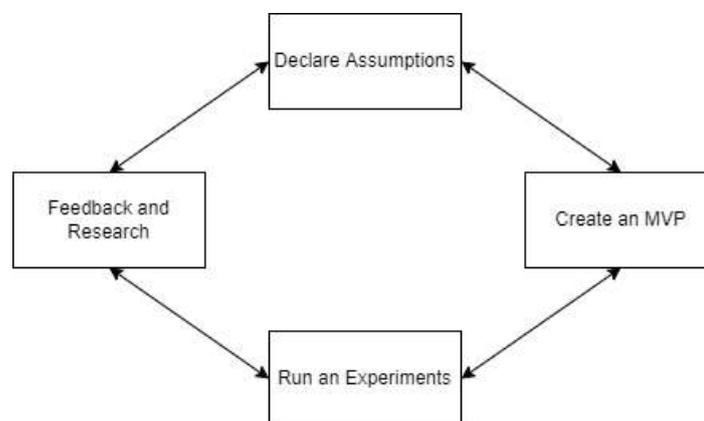
2.1.7. Lean UX

Lean UX menurut (Adhipratama, Sagirani, & Kartikasari, 2018) adalah sebuah metode yang mengutamakan kecepatan proses dan sangat berfokus kepada kebutuhan pengguna aplikasi. Metode *Lean UX* memberikan sifat asli dari desain antarmuka pengguna untuk tingkat keberhasilan yang lebih cepat, kolaboratif dan lintas fungsi dengan meminimalkan penekanan pada dokumentasi dan mengurangi proses yang tidak perlu selama pengembangan.

Sedangkan menurut (Saputra, Sulistiowati, & Lemantara, 2018) *Lean UX* adalah metode perancangan UI/UX yang berpusat pada pengalaman nyata yang

akan di desain serta disertai dengan teknik dokumentasi yang konvensional untuk mendapatkan informasi yang hanya diperlukan saat pengembangan. Sehingga metode ini dapat menghindari proses yang tidak dibutuhkan dan membuat pekerjaan menjadi lebih cepat.

Terdapat 4 siklus tahapan pada metode *Lean UX*, yaitu mendeklarasi asumsi, membuat MVP, melakukan eksperimen, serta umpan balik dan penelitian.



Gambar 2.1. Siklus Tahapan Lean UX

1. Mendeklarasi Asumsi (*Declare Assumptions*)

Pada tahap mendeklarasi asumsi dilakukan pendeklarasian asumsi berdasarkan informasi yang didapatkan melalui wawancara dan menyebarkan kuesioner mengenai cara kerja dan permasalahan yang dihadapi pengguna selama menggunakan aplikasi serta melakukan observasi pada aplikasi yang akan dikembangkan.

2. Membuat MVP (*Create an MVP*)

Pada tahap membuat MVP akan dilakukan pembuatan MVP (*Minimum Viable Product*) berupa *prototype* untuk merancang ulang antarmuka aplikasi berdasarkan penyelesaian permasalahan yang dihadapi.

3. Melakukan Eksperimen (*Run an Experiment*)

Tahap melakukan eksperimen adalah tahap menguji MVP yang telah dirancang ditahap sebelumnya, eksperimen dapat dikerjakan dengan menggunakan tim yang ada atau dilakukan secara mandiri dengan tujuan mengetahui apakah *prototype* telah layak dan sesuai seperti yang dibutuhkan serta dapat berjalan dengan baik.

4. Umpan Balik dan Penelitian (*Feedback and Research*)

Tahap Umpan balik dan Penelitian yaitu melakukan pengujian *prototype* yang sudah dibuat kepada pengguna aplikasi untuk mendapatkan masukan mengenai tampilan antarmuka yang telah dikembangkan.

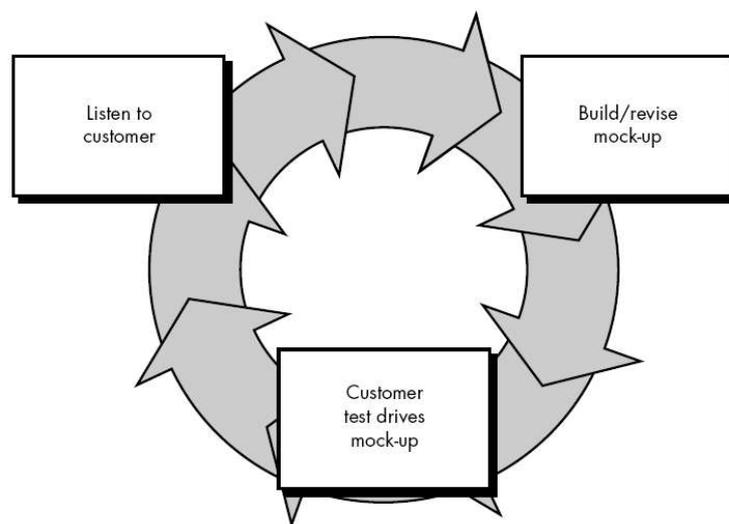
2.1.8. *Prototype*

Prototipe merupakan model pengerjaan untuk pengembangan *software*. *Prototype* jika diuraikan dalam bahasa latin *proto* yang memiliki arti asli dan *typus* memiliki arti model. Tujuan pembuatan *prototype* sebagai model presentasi atau sebagai alat dari proses pengembangan suatu sistem. Pada bidang desain, *prototype* dapat disebut sebagai bentuk awal atau contoh dari sebuah entitas (Muhyidin, Sulhan, & Sevtiana, 2020).

Menurut (Siswidiyanto, Munif, Wijayanti, & Haryadi, 2021) *Prototype* adalah sebuah produk yang secara langsung dapat mendemonstrasikan bagaimana

suatu sistem informasi akan beroperasi sebelum diimplementasikan ke dalam sistem yang sebenarnya.

Metode *prototyping* adalah metode yang digunakan untuk membangun sebuah program secara cepat dan bertahap sehingga pengguna dapat mengevaluasinya dengan segera (Perwitasari, 2019).



Gambar 2.2. Tahapan *Prototyping*

Tahapan-tahapan yang dilakukan pada metode *prototype* yaitu:

1. *Listen to customer*

Pada tahap mendengarkan pelanggan/*user* akan dilakukan pengumpulan data terhadap cara kerja dan permasalahan apa saja yang sedang dihadapi pada sistem yang akan dikembangkan.

2. *Build/revise mock-up*

Tahap membuat *prototype* ialah tahap melakukan perancangan/pengembangan desain antarmuka aplikasi sesuai dengan kebutuhan.

3. *Customer test drives mock-up*

Tahap pengujian akan dilakukan pengujian dengan sistem yang sudah dikembangkan untuk memastikan bahwa sistem dapat berjalan secara fungsional tanpa adanya kesalahan dan menghasilkan keluaran sesuai dengan keinginan.

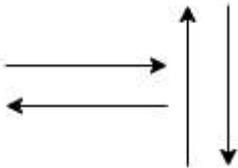
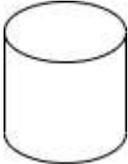
2.1.9. Aliran Sistem Informasi

Aliran sistem informasi ialah gambaran dari suatu alur sistem yang menjelaskan proses dari mula sebuah program sampai akhir sebuah program (Silalahi, 2021).

Menurut (Maydianto, 2021) aliran sistem informasi adalah tabel yang menjelaskan sebuah alur kerja dari awal sistem dikerjakan sampai akhir sistem diselesaikan atau aliran sistem informasi juga dapat diartikan sebagai alat perancang gambaran alur pemrograman dari awal hingga selesai.

Menurut berbagai pernyataan yang telah disampaikan, maka bisa disimpulkan bahwa aliran sistem informasi penting digunakan pada proses perancangan suatu sistem. Permasalahan yang dihadapi pada sistem dapat dilihat dari proses dari sistem informasi tersebut, sehingga dapat menentukan apakah sistem tersebut dapat beroperasi dengan baik atau tidak.

Tabel 2.1. Aliran Sistem Informasi *Symbols*

<i>Symbols</i>	Penjelasan
	<p>Proses</p> <p>Sebagai pemrosesan data yang dilakukan secara komputerisasi.</p>
	<p>Dokumen</p> <p>Sebagai entitas merekam data input dan output, bisa digunakan untuk pemrosesan secara manual atau komputer.</p>
	<p>Garis Alir</p> <p>Untuk arah pengaliran data proses</p>
	<p>Proses Manual</p> <p>Untuk proses pengolahan data secara manual</p>
	<p>Basis Data</p> <p>Untuk media penyimpanan secara komputerisasi</p>

2.1.10. UML

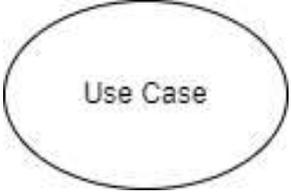
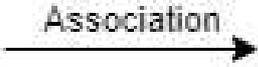
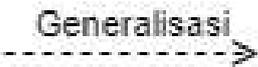
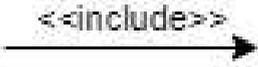
Dalam pengembangan bahasa *object-oriented programming*, terdapat bahasa pemodelan standar untuk pengembangan perangkat lunak yang dirancang menggunakan teknologi *object-oriented programming* yaitu *Unified Modeling Language* (UML). Pemodelan ini umumnya digunakan untuk mendeskripsikan dan membangun sistem *software* (Andrianof, 2018).

UML merupakan bahasa perancangan pada program yang memiliki tujuan umum dalam pengembangan sistem dengan memberikan visual desain dasar yang ada pada sistem (Silalahi, 2021).

Menurut (Maydianto, 2021) Bahasa Pemodelan Terpadu (UML) merupakan suatu *language* perancangan visual yg sering dipakai didalam dunia kerja sebagai pembuat analisis desain dan penggambaran arsitektur berorientasi objek atau *Unified Modeling Language* (UML) juga dapat diartikan sebagai kumpulan diagram yang telah memiliki masing-masing kriteria untuk merancang suatu perangkat lunak berbasis objek. Untuk membangun perangkat lunak berbasis objek, dapat menggunakan diagram pada UML diantaranya adalah: Diagram *Use Case*, Diagram *Sequence*, dan Diagram *Activity*.

1. Diagram *Use Case* dapat diartikan sebagai gambaran pada sistem yang akan dibangun. Pada umumnya diagram *Use Case* dipakai sebagai alat yang dapat memberikan gambaran fungsi dari sistem yang akan dibangun serta hak akses dalam penggunaan sistem tersebut. Diagram *Use Case* memiliki simbol yang dapat digunakan, berikut adalah simbol yang digunakan:

Tabel 2.2. *Use Case Diagram Symbols*

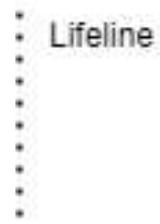
Symbols	Penjelasan
	<p>Utilitas yang tersedia untuk sistem sebagai kumpulan unit yang bertukar <i>message</i> satu sama lain</p>
	<p>Aktor atau utilitas lainnya yang dapat berkorelasi dengan <i>system</i> yang akan dirancang, meskipun simbol yang digunakan adalah gambar orang, aktor jelas bukan manusia</p>
	<p>Penggunaan hubungan antara aktor dan <i>use case</i></p>
	<p>Penggunaan hubungan generalisasi antar dua <i>use case</i> yang memiliki fungsi umum dari <i>use case</i> lainnya</p>
	<p>Untuk menambah <i>relation</i> antar <i>use case</i></p>

Tabel 2.2. Lanjutan

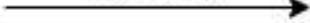
 <p>The diagram shows a horizontal arrow pointing to the right. Above the arrow, the text '<<extend>>' is written in a monospace font.</p>	<p>Untuk menambah relasi dari satu <i>use case</i> ke <i>use case</i> lainnya yang independen</p>
---	---

2. Diagram *Sequence* merupakan penggambaran aktivitas dari objek yang terdapat pada *use case* dengan menjelaskan waktu kejadian serta laporan yang terkirim dan diterima sesama *object*.

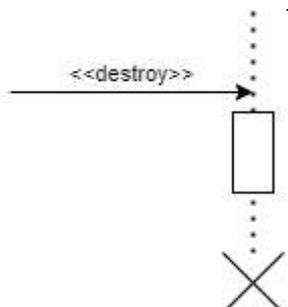
Tabel 2.3. *Sequence Diagram Symbols*

Symbols	Penjelasan
 <p>The diagram shows a stick figure symbol representing an actor. It consists of a circle for a head, a horizontal line for shoulders, and two diagonal lines for legs. Below the figure, the word 'Actor' is written.</p>	<p>Aktor atau utilitas lainnya yang dapat berkorelasi dengan <i>system</i> yang akan dirancang, meskipun simbol yang digunakan adalah gambar orang, aktor jelas bukan manusia</p>
 <p>The diagram shows a vertical dashed line representing a lifeline. To the right of the line, the word 'Lifeline' is written.</p>	<p>Menggambarkan kehidupan pada objek yang berada pada garis tersebut</p>

Tabel 2.3. Lanjutan

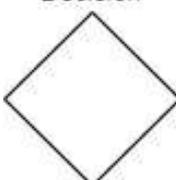
<p style="text-align: center;">Objek</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 80%;"> <p style="text-align: center;">Nama Objek : Nama Kelas</p> </div>	<p>Menggambarkan <i>object</i> yang berkorelasi <i>message</i></p>
<p style="text-align: center;">Waktu Aktif</p> <div style="background-color: black; width: 20px; height: 40px; margin: 10px auto;"></div>	<p>Memiliki gambaran mengenai objek yang dapat melakukan interaksi dan aktif</p>
<p style="text-align: center;"><<create>></p> <div style="text-align: center;">  </div>	<p>Menunjukkan <i>object</i> yang dapat menciptakan <i>object</i> lainnya</p>
<p style="text-align: center;">1:nama_metode()</p> <div style="text-align: center;">  </div>	<p>Menunjukkan <i>object</i> yang memanggil keterangan yang terdapat pada <i>object</i> lainnya atau sendiri</p>
<p style="text-align: center;">1:masukan</p> <div style="text-align: center;">  </div>	<p>Menunjukkan kalau <i>object</i> memberikan informasi kepada <i>object</i> lainnya</p>
<p style="text-align: center;">1:keluaran</p> <div style="text-align: center;">  </div>	<p>Menunjukkan suatu kembalian yang dihasilkan dari <i>object</i> setelah melakukan suatu pekerjaan</p>

Tabel 2.3. Lanjutan

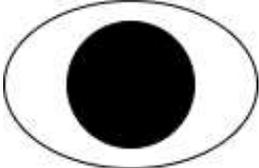
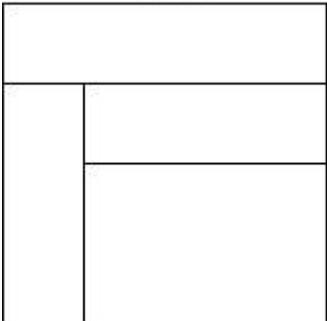
	<p>Menggambarkan <i>object</i> yang memusnahkan <i>object</i> lainnya.</p>
---	--

3. *Activity Diagram* merupakan pemodelan alur kerja atau aktivitas yang hanya dilakukan pada sistem tidak termasuk dengan aktivitas yang dikerjakan oleh *actor*.

Tabel 2.4. *Activity Diagram Symbols*

Symbols	Penjelasan
<p>Status Awal</p> 	<p>Status awal dari <i>activity</i> yang akan dilakukan pada <i>system</i></p>
	<p>Suatu aktivitas pengerjaannya dilakukan oleh sistem</p>
<p>Decision</p> 	<p>Suatu <i>symbol</i> pengambilan keputusan yang terdiri dari dua atau lebih pilihan</p>

Tabel 2.4. Lanjutan

<p style="text-align: center;"><i>Join</i></p> 	<p>Kombinasi dari aktivitas yang lebih dari satu</p>
<p style="text-align: center;">Status Akhir</p> 	<p>Status akhir aktivitas dari sistem</p>
<p style="text-align: center;"><i>Swimlane</i></p> 	<p>Kelompok yang memiliki tanggung jawab pada kerjaan yang dilakukan.</p>

2.2. Tinjauan Teori Khusus

2.2.1. *User Interface*

Berdasarkan penelitian yang dilakukan oleh (Muhyidin et al., 2020) antarmuka pengguna adalah tampilan pada aplikasi yang mencakup tombol, teks, gambar, animasi, dan hal lain yang memiliki kegunaan untuk berinteraksi dengan pengguna. Aplikasi dengan antarmuka pengguna yang baik dapat menambah keinginan pengguna untuk menggunakan aplikasi tersebut. Untuk meningkatkan loyalitas pengguna, perancang antarmuka pengguna dapat menentukan skema

warna yang cocok dengan tema aplikasi, memilih bentuk tombol yang sesuai, dan jenis *font* yang akan digunakan untuk teks.

User Interface merupakan komponen yang tidak terpisahkan dengan aplikasi untuk mempermudah pengguna dalam berinteraksi dengan aplikasi yang sedang digunakan. *User Interface* yang baik juga dapat mempengaruhi kenyamanan serta minat dari pengguna saat berinteraksi dengan aplikasi (Anggara et al., 2021).

User Interface menurut (Fernando, 2020) adalah masukan dan keluaran yang langsung melibatkan pengguna akhir. Tujuan sebuah *user interface* adalah mengkomunikasikan fitur yang tersedia pada sistem dengan tujuan membuat pengguna mengerti dan dapat mengoperasikan sistem tersebut.

User Interface memiliki fungsi untuk menghubungkan suatu sistem kepada *user*, sehingga *user* dapat memperoleh informasi yang mereka butuhkan. *User interface* yang baik juga harus mudah digunakan atau biasa disebut dengan *user friendly*. *User friendly* adalah kemampuan antarmuka pengguna yang mudah digunakan dan dapat memberikan kemudahan sehingga pengguna tidak merasa kesulitan dan kewalahan dalam menggunakan aplikasi (Windarto & Marfiah, 2020).

2.2.2. *User Experience*

User experience adalah proses yang berkaitan dengan pengalaman pengguna saat menggunakan aplikasi, kemudahan penggunaan aplikasi, persepsi pengguna menggunakan aplikasi dan bagaimana pengguna mencapai tujuan yang diinginkan saat menggunakan sebuah aplikasi. Jika sebuah aplikasi memiliki desain antarmuka

pengguna yang buruk, maka hal itu dapat mempengaruhi pengalaman penggunanya (*user experience*) (Razi et al., 2018).

Menurut (Adhipratama et al., 2018) pengalaman pengguna bukan dinilai dari cara kerja yang dilakukan oleh suatu sistem, melainkan interaksi antara pengguna dengan sistem, apakah sistem mudah digunakan, dan bagaimana pengalaman pengguna dalam memahami, menyerap dan mendapatkan informasi yang disediakan.

Tujuan dari *user experience* adalah untuk meningkatkan interaksi antara pengguna dan aplikasi yang dimana kondisi internal dan emosional dari pengguna dapat berubah selama atau setelah menggunakan sebuah aplikasi (S. L. Ramadhan, 2021).

Menurut (Fernando, 2020) *User experience* merupakan persepsi pengguna dalam menggunakan sebuah aplikasi. Seberapa bergunanya fungsi yang diberikan aplikasi, kemudahan saat pengoperasian dan kepuasan pengguna terhadap aplikasi juga dinilai melalui *user experience*. Pengalaman pengguna tidak dapat dinilai hanya melalui aplikasi pada perangkat yang sama, seperti halnya saat mengakses aplikasi pada *smartphone* dengan *desktop* tentu saja memiliki pengalaman pengguna yang berbeda. Oleh karena itu, perlu memahami fungsi dan kebergunaan aplikasi yang akan dibuat.

2.2.3. Figma

Figma merupakan *software editor* dan alat *prototyping* yang berbasis *website* yang dapat di akses melalui *browser* dan aplikasi *desktop* untuk sistem operasi Mac

OS dan Windows. Aplikasi Figma dapat mempermudah dalam mendesain *User Interface* aplikasi, *website*, dan antarmuka lainnya. Figma juga memiliki fitur *Cloud* yang memungkinkan para pengguna dari aplikasi ini dapat mengakses *file* dan data yang tersimpan dimana saja melalui *browser*.

Berdasarkan penelitian yang dilakukan oleh (Muhyidin et al., 2020) Figma merupakan *software* komputer yang dipakai untuk merancang *interface* aplikasi *mobile*, halaman web, dan sebagainya. Pada umumnya aplikasi Figma lebih umum dipakai pengguna yang beroperasi di bidang Desainer *User Interface*, desainer antarmuka *website*, dan lainnya dikarenakan aplikasi Figma mempunyai fitur yang berguna yakni dapat mengerjakan suatu pekerjaan *desain* lebih dari satu orang secara bersama-sama walaupun berada di tempat yang berbeda.