

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Dalam penelitian ini peneliti mengkaji beberapa studi literatur beberapa jurnal penelitian sebelumnya. Jurnal-jurnal tersebut dipilih menjadi literatur berdasarkan topik, dan metode yang sesuai dan berkaitan dengan topik penelitian penulis tentang perancangan *website* pendataan burung pleci hasil konservasi Geong Bird Farm menggunakan metode *user centred design*.

Penelitian pertama ada penelitian tentang Perancangan UI/UX Menggunakan Metode UCD Pada Website PMKS Pendataan Anak Yatim Lombok Tengah. Aplikasi ini dirancang dengan menerapkan metode UCD dan melakukan wawancara langsung terhadap pengguna dan pengamat aplikasi yang sejenis. PKMS Pendataan anak yatim memiliki tujuan agar mempermudah akses mereka terhadap Pendidikan yang menyenangkan, dalam aplikasi terdiri dari filter data, grafik afikasi akun dan pendataan anak yatim disetiap berdasarkan desa, kecamatan dan prestasi mereka [8].

Penelitian kedua ada penelitian tentang Penerapan UCD (*User Centered Design*) Pada Perancangan Sistem Informasi Manajemen Aset TI Berbasis Web di Bid TIK Kepolisian Daerah Kepulauan Riau. Proses adminitrasi serta pendataan asset di bidang TIK masih dilakukan dengan menggunakan *Microsoft excel*, yang mengakibatkan sulitnya penelusuran dan pengelolaan asset yang ada. Menghambat proses ketika terjadi pergantian asset asset yang rusak menjadi kurang maksimal. Pengimplementasian aplikasi untuk pengelolaan asset agar tertib baik secara adminitrasi maupun dokumentasi sehingga ketika ada prosedur pengadaan, perubahan sampai penghapusan dapat terekam dengan jelas [9].

Penelitian ketiga ada penelitian tentang Perancangan Sistem Informasi Buku Penghubung Menggunakan Model *User Centered Design* (UCD) Berbasis Web. TK Saint josep Montessori school semarang menggunakan buku penghubung

sebagai pen jembatan komunikasi antar guru dan orang tua namun terdapat beberapa permasalahan seperti anak yang lupa membawa buku penghubung sehingga monitoring dan pencatatan terlewat atau tidak adanya pelaporan terhadap orang tua yang anaknya lupa membawa buku. Selain itu buku penghubung yang masih manual penulisan dilakukan satu persatu secara berulang sehingga memakan waktu yang lama. Pembuatan sistem informasi buku penghubung berbasis web menggunakan model pengembangan UCD (*User Centered Design*) bertujuan mempermudah pencatatan dan pelaporan kegiatan siswa TK [10].

Penelitian keempat ada penelitian tentang Rancang Bangun Sistem Informasi *E-Laundry* yang membantu menjalankan bisnis laundry dengan lebih efisien dan memberikan kemudahan bagi pelanggan untuk melakukan transaksi, pemilik lebih dapat mempermudah pengelolaan data dari pengguna *website* untuk mendata pelanggan yang baru saja melakukan pemesanan jasa dan dapat melakukan antar jemput. Tujuan dari penelitian ini untuk membuat sebuah sistem informasi e laundry berbasis website. *Waterfall* merupakan metode yang digunakan karena memiliki alur mulai perancangan, analisis, implementasi, pengujian. Perancangan sistem ini menggunakan Bahasa pemrograman *PHP Framework Laravel* dan *UML (Unified Modelling Language)*, database *MYSQL* [11].

Penelitian kelima ada penelitian tentang Sistem Informasi Mahasiswa untuk mengakses dan mengorganisir data dari mahasiswa yang memiliki prestasi di universitas palngkaraya. Hal tersebut terjadi karena proses pencatatan yang dilakukan oleh fakultas masih menggunakan media manual dengan media excel sehingga kurang efektif karena masih banyak mahasiswa yang belum tercatat prestasinya. Proses yang diperlukan juga relative lama karena pihak rektorat bidang Metodologi pengembangan menggunakan metode *waterfall* analisis, desain, implementasi, dan pengujian untuk membuat sistem informasi berbasis web ini. Diagram alur dikembangkan pada langkah analisis ini. *Data Flow Diagram (DFD)* dan *Entity Relationship Diagram (ERD)* dikembangkan selama tahap desain untuk memastikan bahwa pengunjung situs web dapat dengan mudah memahami aliran informasi di dalam sistem. Pendekatan pengujian *black box testing* digunakan dalam proses pengujian [12].

Tabel 2. 1 Ringkasan penelitian terdahulu

| No | Judul   | Perbandingan  | Tujuan  | Kritikan   | Persamaan   | Kesimpulan  |
|----|---|---|---|--|---|---|
| 1. | Perancangan UI/UX Menggunakan Metode UCD Pada <i>Website</i> PMKS Pendataan Anak Yatim Lombok Tengah [8].   | -Menerapkan metode <i>user centred design</i>   | Membuat aplikasi PMKS untuk pendataan anak yatim di Lombok Tengah.      | Tidak terdapat testing untuk sistem yang dibuat.   | Sistem yang dibuat sama sama berbasis web manajemen pendataan serta menerapkan metode <i>UCD</i> .  | Hanya masih berupa rancangan desain.  |
| 2. | Penerapan UCD ( <i>User Centered Design</i> ) Pada Perancangan Sistem Informasi Manajemen Aset TI Berbasis Web di Bid TIK Kepolisian Daerah Kepulauan Riau [9]. | -Menggunakan metode <i>user centred design</i> .<br>- Perancangan dengan analisis UML.<br>-Pengujian dilakukan dengan <i>Black Box</i> .<br>-Menggunakan <i>MySQL</i> untuk database. | Membuat sistem untuk pendataan asset bidang TIK Kepolisian daerah Riau. | Peran Admin, Operator, dan <i>User</i> hanya dijabarkan sekilas tidak dijelaskan secara detail perbedaan akses mereka terhadap sistem. | Sistem yang dibuat sama sama berbasis web sebagai platform untuk manajemen pendataan dengan menerapkan <i>UCD</i> , <i>UML</i> , dan <i>Black Box Testing</i> . | Aplikasi yang dibangun dapat digunakan untuk mengelola asset yang terdapat pada Bidang TIK dengan baik. |

| No | Judul   | Perbandingan   | Tujuan  | Kritikan   | Persamaan  | Kesimpulan   |
|----|---|--|---|--|--|--|
| 3. | Perancangan Sistem Informasi Buku Penghubung Menggunakan Model <i>User Centered Design (UCD)</i> Berbasis Web [10]. | -Menggunakan metode <i>user centred design</i> .<br>-Perancangan dengan analisis UML.<br>-Pengujian dilakukan dengan <i>Black Box</i> .  | perancangan aplikasi sistem buku penghubung untuk mealporkan kegiatan anak TK | Diagram UML yang dipakai sedikit.                        | Sistem yang dibuat sama sama menerapkan <i>UCD, UML</i> dan <i>Black Box</i> .                             | Dibandingkan dengan pencatatan manual sistem dapat melakukan lebih cepat |
| 4. | Rancang Bangun Sistem Informasi <i>E-Laundry</i> dengan Implementasi Berbasis Web (Analysis) [11].                  | -Menerapkan metode <i>waterfall</i> .<br>-Perancangan dengan analisis UML.   | Membuat aplikasi web untuk manajemen laundry.                                 | Hanya sampai perancangan UML desain , testing tidak ada. | Sama sama sistem manajemen berbasis Web dan Menerapkan UML   | Hanya sampai perancangan UML.  |
| 5. | Sistem Informasi Mahasiswa Berprestasi Universitas Palangka Raya Berbasis <i>Website</i> [12].                      | -Perancangan dengan analisis <i>UML</i> .<br>-Menggunakan <i>MySQL</i> untuk database.<br>-Pengujian dilakukan dengan <i>Black Box</i> . | Membuat sistem informasi data mahasiswa.                                      | Tidak dijelaskan tahap desain dan scenario pengujianya.  | Sama sama sistem manajemen berbasis Web dan Menerapkan <i>UML, Black Box</i> , dan database <i>MySQL</i> . | Memperbaiki desain dari sistem yang sudah ada .                          |

## 2.2 Landasan Teori

Dalam penelitian ini peneliti mengkaji beberapa teori sebagai acuan pedoman dalam menyusun penelitian diantaranya adalah sebagai berikut :

### 2.2.1 Rancang bangun

Kata kerja "merancang" adalah asal mula istilah "desain", yang berarti mengatur sesuatu (sebelum bertindak, melakukan, atau merencanakan). Sedangkan Perancangan merupakan suatu proses untuk membuat baru atau mendesain pengembangan sistem dapat berupa penggambaran, pembuatan sketsa, perencanaan maupun pengaturan dari berbagai elemen yang terpisah ke dalam kesatuan yang utuh sehingga dan berfungsi, dapat menyangkut konfigurasi dari komponen baik perangkat lunak maupun perangkat keras dari suatu system [13]. Sedangkan pengertian dari bangun maupun pembangunan sistem merupakan rangkaian aktivitas menciptakan sebuah sistem baru memperbaiki sistem yang sebelumnya ada baik secara keseluruhan maupun sebagian. Untuk rancang sendiri merupakan aktivitas dari penerjemahan hasil analisis perancangan menjadi bentuk perangkat lunak kemudian menciptakan sistem tersebut [14].

### 2.2.2 Website

*Website* merupakan sekumpulan beberapa halaman berfungsi sebagai media penyajian informasi dalam bentuk teks, foto, suara, animasi, maupun gabungan dari semuanya, yang membentuk rangkain bangunan terkait dihubungkan dengan *hyperlink* atau jaringan halaman dapat diakses melalui perangkat lunak *browser*. *Browser* atau perambah merupakan aplikasi yang mampu menjalankan informasi dokumen web dengan cara diterjemahkan oleh komponen *web engine* sehingga semua dokumen dapat diterjemahkan lalu ditampilkan. *Website* memiliki 2 sifat yaitu *statis* dan *dinamis* [15]. Berikut ini merupakan penjelasan dari jenis *website* berdasarkan sifatnya :

#### 1. *Website statis*

*Website statis* merupakan *website* dimana sifat dari *website* ini ialah tidak mudah diubah kontennya oleh pengguna. Dilakukanya perubahan

perubahan isi atau konten, pengguna harus merubahnya dengan cara *coding* atau mengubah struktur kodenya [16].

## 2. *Website dinamis*

*Website dinamis* merupakan *website* dimana sifat dari *website* ini ialah mudah beradaptasi dan otomatis menyesuaikan perubahan konten secara langsung tanpa harus merubah struktur kode *website* [16]. Contoh dari *website* dinamis adalah *website-ecommerce* dimana untuk datanya dapat berubah-ubah dalam kurun waktu tertentu [17].

### 2.2.3 Pendataan

Data adalah realitas suatu kejadian atau peristiwa hasil dari kegiatan organisasi disebut data organisasi. Data dipandang sebagai informasi mentah dalam konteks sistem informasi. adalah data yang telah diolah sesuai dengan tuntutan. Pendataan adalah proses pencatatan informasi yang benar dan nyata tentang sesuatu orang, benda, lingkungan, atau peristiwa tertentu sebagai dokumentasi atau arsip yang dapat digunakan untuk keperluan masa depan [18].

### 2.2.4 Konservasi

Konservasi berasal dari bahasa Inggris *conservation*, yang memiliki makna arti pelestarian atau perlindungan sumber daya alam untuk dikelola dengan teratur dengan tujuan untuk mencegah kerusakan dan kemusnahan. Dengan tujuan menyelamatkan dan melindungi lingkungan serta sumber daya alam untuk memperpanjang jangka waktu manfaatnya. Tujuan ini dikaitkan dengan gagasan pembangunan berkelanjutan, yang menjaga keseimbangan dan melindungi sumber daya alam sepanjang pembangunan. Artinya segala bentuk aktivitas pengelolaan sumber daya alam harus dilakukan secara proporsional dan tidak melakukan eksploitasi secara berlebihan [19].

### 2.2.5 Figma

Figma merupakan design berbasis *website* dan bisa juga berupa software aplikasi yang berjalan di system operasi *windows*, *linux* atau *mac OS* terhubung dengan *cloud* sehingga bisa digunakan kapanpun dan dimanapun asalkan kita tersambung dengan internet yang biasa digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website* dan lain-lain [20]. Figma dipilih karena kemampuannya yang memungkinkan beberapa pengguna berkolaborasi dalam tugas yang sama dari berbagai lokasi. Kita mungkin menggambarkan hal ini sebagai kerja sama kolektif. Banyak desainer UI/UX menggunakan Figma karena fitur-fiturnya, yang memungkinkan mereka menghasilkan prototipe untuk situs web atau aplikasi dengan cepat dan efisien. [21].

### 2.2.6 Visual studio code

*Visual studio code* merupakan sebuah code editor buatan Microsoft dapat digunakan oleh semua Bahasa pemrograman dan bisa digunakan di *Linux*, *macOS*, dan *Windows*. *VS Code* memiliki fitur-fitur yang sangat berguna bagi para pengembang, seperti kemampuan untuk melakukan debugging, integrasi dengan *Git*, dan dukungan untuk berbagai bahasa pemrograman. Selain itu, *VS Code* juga memiliki ekstensi yang dapat diunduh dan dipasang untuk menambahkan fitur-fitur tambahan.

### 2.2.7 User centered design

*User centered design* merupakan perancangan dengan filosofi pengembangan sistem, tujuan, serta sifat dalam lingkungan sistem menjadikan pengguna fokus atau pusat utama pada pengalaman pengguna *atau experience*. Perhatian lebih diberikan pada setiap tahapan proses perancangan dengan memperhatikan kebutuhan, keinginan, dan keterbatasan dari pengguna akhir [22].

Dalam proses pada metode UCD, terdapat empat langkah yang dilakukan secara berulang berikut ini merupakan penjelesan keempat langkah tersebut:

1. *Specify the context of use.*

Melakukan proses sambil memutuskan identifikasi siapa yang akan menggunakan sistem tersebut. menggambarkan produk dan keadaan di mana mereka ingin menggunakannya [23]. User stories dibutuhkan untuk menjelaskan peran dari pengguna masing-masing dari actor memiliki peran masing masing untuk menjalankan sistem [24].

2. *Specify the user and organizational requirements.*

Melakukan proses ini untuk identifikasi kebutuhan dari pengguna. wawancara maupun menggunakan kuesioner bisa dilakukan untuk memperoleh informasi kebutuhan *fungsiional* serta *non-fungsiional* yang nanti akan diterapkan pada aplikasi [23]. Pada titik ini, peneliti akan menggunakan UML untuk mendefinisikan permintaan pengguna. Diawali dengan use case diagram, dibuatlah diagram aktivitas berdasarkan use case yang teridentifikasi, skenario use case, dan terakhir sequence diagram [25].

3. *Product design solution*

Pada tahap ini langkah peneliti adalah mengembangkan desain berdasarkan temuan analisis kebutuhan. Hal ini akan memfasilitasi pembuatan wireframe dan memungkinkan penjelasan luas tentang fungsionalitas sistem yang dimaksud. [23].

4. *Evaluate design*

Setelah tahap evaluasi desain tercapai dan diselesaikan, pengguna akan menerima temuan untuk penilaian tambahan guna menentukan apakah desain memenuhi kebutuhan dan spesifikasi pengguna atau tidak. Jika tidak sesuai, peneliti akan berupaya menyempurnakan desain sistem dengan memasukkan hasil evaluasi ke dalam rekomendasi yang selanjutnya akan diberikan oleh calon pengguna. [23].

### 2.2.8 Unified modeling language

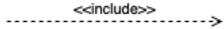
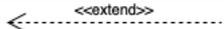
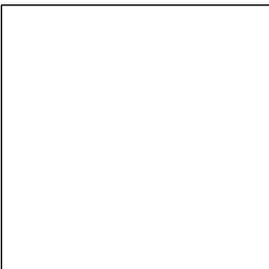
Berikut ini merupakan penjelasan dan notasi dari UML atau *Unified Modeling Language* yang merupakan sebuah *tools* atau pemodelan dalam perancangan pengembangan *software* dengan berbasis *object-oriented*. Dengan penggunaan penulisan standar, *Unified Modeling Language* sendiri dapat menghasilkan rancangan sistem yang mencakup skema database, ide proses bisnis, instruksi penulisan dalam bahasa pemrograman tertentu, dan komponen sistem perangkat lunak. [26].

#### A. Use case diagram

*Use case diagram* atau sering juga disebut dengan diagram *use case* sendiri merupakan model suatu perilaku yang menggambarkan interaksi satu atau beberapa aktor terhadap sistem yang pertama kali dibuat. [27]. interaksi disini adalah antara user sebuah sistem dengan sistemnya sendiri dengan menggunakan cerita bagaimana sistem itu dipakai [28]. *Use case* digunakan untuk memahami fungsi apa saja yang saat ini tersedia dalam sistem dan siapa yang mempunyai wewenang untuk menggunakan fungsi tersebut. [29]. Berikut ini merupakan notasi *use case diagram* beserta serta penjelasannya dapat kita lihat pada Tabel 2.2

Tabel 2. 2 Notasi *use case diagram*

| No | Nama Notasi     | Simbol  | Keterangan   |
|----|-----------------|---|--|
| 1. | <i>Actor</i>    | <br>Actor    | Mewakili, dalam berinteraksi dengan peran orang, dengan sistem lain. |
| 2. | <i>Use Case</i> | <br>Use Case | <i>Use case</i> abstraksi dan interaksi antara sistem dan aktor      |
| 3. | <i>Asosiasi</i> |              | Relasi antara aktor dengan <i>use case</i> .                         |

| No | Nama Notasi    | Simbol  | Keterangan  |
|----|----------------|---|---|
| 4. | <i>Include</i> |    | Simbol untuk menjelaskan suatu <i>use case</i> B adalah fungsionalitas dari <i>use case</i> A         |
| 5. | <i>Extend</i>  |    | Simbol yang menjelaskan bahwa <i>use case</i> B adalah tambahan fungsionalitas dari <i>use case</i> A |
| 6. | <i>Sytem</i>   |  | Simbol yang menyatakan batasan antara sistem dengan <i>actor</i> yang menggunakannya.                 |

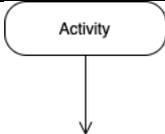
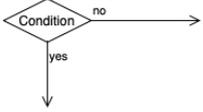
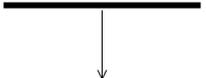
### B. Use Case Secenario

Skenario merupakan urutan spesifik dari aksi dan interaksi antara aktor dengan sistem sistem [30]. *Use case skenario* merupakan suatu *flow of event* dari use case utama dan dapat memberikan gambaran urutan dari interaksi aktor ke *use case* tersebut dari awal mulai aktor berinteraksi sampai selesai [31].

### C. Activity Diagram

*Activity diagram* atau diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem [32]. Berikut ini merupakan *notasi Activity diagram* beserta penjelasannya dapat kita lihat pada Tabel 2.3

Tabel 2. 3 Notasi *activity diagram*

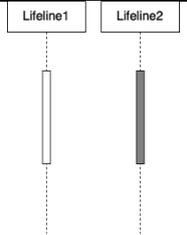
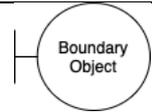
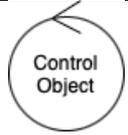
| No | Nama Notasi                      | Simbol   | Keterangan  |
|----|----------------------------------|--|---|
| 1. | Status awal                      |     | Simbol yang digunakan untuk memulai <i>activity</i>   |
| 2. | Aktivitas                        |    | Simbol untuk <i>activity</i> yang dilakukan pengguna atau sistem. Umumnya diawali dengan kata kerja |
| 3. | Percabangan /<br><i>Decision</i> |  | Simbol yang digunakan ketika terdapat pilihan aktivitas lebih dari satu                             |
| 4. | Penggabungan/ <i>Join</i>        |  | Simbol untuk menggabungkan beberapa aktivitas menjadi satu  |
| 5. | Status Akhir                     |   | Simbol untuk menyatakan proses telah berakhir   |
| 6. | <i>Swimlane</i>                  |  | Berfungsi untuk memisahkan kelompok <i>activity</i>   |

#### D. Sequence Diagram

Pada titik ini, *Sequence Diagram* menawarkan pemahaman menyeluruh tentang langkah-langkah yang digunakan oleh sistem untuk mencapai tujuan *use case*. [29]. *Sequence diagram* sendiri adalah

kelakuan objek pada use case dan pendeskripsian waktu hidup objek serta pengiriman pesan dan yang diterima dari beberapa objek. Gambaran *sequence diagram* ketika dibuat mempunyai jumlah yang sama dengan pendefinisian use case, proses paling penting semua use case mempunyai definisi interaksi jalannya pesan sudah mencakup *sequence diagram* dengan jumlah *sequence diagram* disesuaikan dengan jumlah use case harus dibuat sama banyak [28]. Berikut ini merupakan notasi *Sequence diagram* beserta penjelasannya dapat kita lihat pada Tabel 2.4

Tabel 2. 4 Notasi *sequence diagram*

| No | Nama Notasi     | Simbol  | Keterangan  |
|----|-----------------|---|---|
| 1. | <i>Lifline</i>  |   | Mewakili keberadaan suatu objek.  |
| 2. | <i>Actor</i>    |  | Mewakili fungsi alat, sistem lain, atau manusia dalam komunikasi sistem.                |
| 3. | <i>Boundary</i> |  | Untuk berkomunikasi dengan sistem lain, baik melalui antarmuka pengguna atau cara lain. |
| 4. | <i>Control</i>  |  | Untuk mengatur arus informasi yang menghasilkan skenario atau rencana                   |
| 5. | <i>Entity</i>   |  | Untuk menunjukkan keberadaan penyimpanan data atau informasi.                           |

| No | Nama Notasi    | Simbol  | Keterangan  |
|----|----------------|---|---|
| 6. | <i>Message</i> |  | Simbol yang menggambarkan komunikasi antar objek. |

### 2.2.9 Entity relationship diagram

*Entity Relationship Diagram* merupakan diagram berbentuk notasi grafis. ERD berfungsi sebagai alat untuk membantu pembuatan database dan memberikan gambaran umum mengenai pengoperasian *database* yang akan dikembangkan [33]. Di dalam ERD terdapat 3 elemen dasar, yaitu entitas, atribut, serta relasi

#### a) *Entitas*

Entitas merupakan objek yang akan menjadi perhatian dalam suatu database. Entitas dapat berupa orang, lokasi, benda, atau keadaan yang berkaitan dengan informasi yang diperlukan. Simbol entitas berbentuk persegi panjang [33].

#### b) *Atribut*

Atribut merupakan informasi yang ada dalam *entitas*. Suatu entitas memiliki *primary key* atau ciri khas *entitas* dan *atribut deskriptif*. *Atribut* biasanya terletak dalam tabel entitas atau dapat juga terpisah dari tabel. Simbol dari atribut berbentuk elips [33].

#### c) *Relasi*

Relasi dalam ERD merupakan hubungan antara dua atau lebih entitas. Simbol dari relasi berbentuk belah ketupat. Relasi yang dapat dimiliki oleh ERD ada beberapa macam. Pertama *One to One* satu anggota *entitas* dapat berelasi dengan satu anggota entitas lain. Kedua *One to Many* Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lain. Ketiga *Many to Many* Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain [33].

### 2.2.10 User experience questionnaire

Metode pengukuran *User Experience Questionnaire* merupakan kemampuan mengukur pengalaman pengguna aplikasi melalui kuesioner yang terdiri dari 26 item pertanyaan yang disusun menjadi 6 skala pengukuran. Pendekatan UEQ memiliki keunggulan dibandingkan teknik pengukuran lainnya karena mudah diterapkan, memungkinkan penghitungan cepat, dan memberikan hasil yang komprehensif.

Tiga komponen utama yang terdiri dari *Attractiveness*, *Pragmatic Quality*, dan *Hedonic Quality*. Aspek *Pragmatic Quality* aspek kualitas pragmatis menggambarkan bagaimana pengguna mempersepsikan suatu objek dari segi masalah teknis (berorientasi tugas), sedangkan aspek *Hedonic Quality* menggambarkan bagaimana pengguna mempersepsikan suatu objek dari segi masalah non-teknis (tidak berorientasi tugas). Ketiga karakteristik tersebut dipecah menjadi enam skala pengukuran: kebaruan (newness), ketergantungan (accuracy), daya tarik (attractiveness), kejelasan (clarity), dan efisiensi (efficiency) [34]. 26 item pertanyaan kuesioner dikelompokkan menjadi enam skala pengukuran. Panduan UEQ mengatur secara lengkap bagaimana instrumen soal disusun. Item pertanyaan dapat dilihat pada gambar 2.1

|                      | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     |                           |    |
|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------------|----|
| menyusahkan          | <input type="radio"/> | menyenangkan              | 1  |
| tak dapat dipahami   | <input type="radio"/> | dapat dipahami            | 2  |
| kreatif              | <input type="radio"/> | monoton                   | 3  |
| mudah dipelajari     | <input type="radio"/> | sulit dipelajari          | 4  |
| bermanfaat           | <input type="radio"/> | kurang bermanfaat         | 5  |
| membosankan          | <input type="radio"/> | mengasyikkan              | 6  |
| tidak menarik        | <input type="radio"/> | menarik                   | 7  |
| tak dapat diprediksi | <input type="radio"/> | dapat diprediksi          | 8  |
| cepat                | <input type="radio"/> | lambat                    | 9  |
| berdaya cipta        | <input type="radio"/> | konvensional              | 10 |
| menghalangi          | <input type="radio"/> | mendukung                 | 11 |
| baik                 | <input type="radio"/> | buruk                     | 12 |
| rumit                | <input type="radio"/> | sederhana                 | 13 |
| tidak disukai        | <input type="radio"/> | menggembirakan            | 14 |
| lazim                | <input type="radio"/> | terdepan                  | 15 |
| tidak nyaman         | <input type="radio"/> | nyaman                    | 16 |
| aman                 | <input type="radio"/> | tidak aman                | 17 |
| memotivasi           | <input type="radio"/> | tidak memotivasi          | 18 |
| memenuhi ekspektasi  | <input type="radio"/> | tidak memenuhi ekspektasi | 19 |
| tidak efisien        | <input type="radio"/> | efisien                   | 20 |
| jelas                | <input type="radio"/> | membingungkan             | 21 |
| tidak praktis        | <input type="radio"/> | praktis                   | 22 |
| terorganisasi        | <input type="radio"/> | berantakan                | 23 |
| atraktif             | <input type="radio"/> | tidak atraktif            | 24 |
| ramah pengguna       | <input type="radio"/> | tidak ramah pengguna      | 25 |
| konservatif          | <input type="radio"/> | inovatif                  | 26 |

Gambar 2. 1 Pertanyaan *user experience questionnaire*

### 2.2.11 Black box testing

*Black box testing* adalah teknik pengujian perangkat lunak yang berfokus pada perilaku eksternal suatu sistem tanpa mempertimbangkan struktur internal atau detail implementasinya. Dalam *black box testing*, penguji memperlakukan sistem sebagai "kotak hitam" dan menguji fungsionalitasnya berdasarkan spesifikasi, persyaratan, dan keluaran yang diharapkan dari sistem tersebut. Tujuan dari *black box testing* adalah untuk memvalidasi apakah sistem memenuhi persyaratan yang ditentukan dan berfungsi dengan benar dari sudut pandang pengguna. *Black box testing* terutama berhubungan dengan pengujian input dan output sistem serta memverifikasi apakah sistem berperilaku sesuai harapan. metode pengujian menggunakan kotak hitam (black box testing). suatu teknik untuk menguji perangkat lunak yang berkonsentrasi pada fungsionalitas, khususnya dalam menentukan apakah masukan dan keluaran aplikasi sesuai dengan harapan atau tidak[35].

### 2.2.12 Analisis deskriptif

Analisis deskriptif berfungsi untuk menghasilkan nilai presentase dari responden pengujian[36]. Salah satunya digunakan dalam pengujian fungsionalitas sebuah sistem dengan formalitas perhitungannya sebagai berikut yang sesuai dengan rumus pada gambar 2.2 beserta penentuan interpretasi dari tabel presentasi kelayakan yang tertera pada Tabel 2.5

$$\text{Presentase kelayakan} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

Gambar 2. 2 Presentase kelayakan

Tabel 2. 5 Tabel Presentase kelayakan

| Presentase | Kelayakan          |
|------------|--------------------|
| 81%-100%   | Sangat Layak       |
| 61%-80%    | Layak              |
| 41%-60%    | Cukup              |
| 21%-40%    | Tidak Layak        |
| ≤ 20%      | Sangat Tidak Layak |