

BAB 2

TINJAUAN PUSTAKA

2.1 Tunanetra

Tunanetra adalah kondisi seseorang yang mengalami keterbatasan dalam penglihatan yang signifikan, baik total maupun parsial. Hal ini menyebabkan mereka tidak dapat melihat antarmuka secara visual dan mengandalkan cara lain untuk berinteraksi dengan sistem atau aplikasi. Oleh karena itu, tunanetra membutuhkan alat bantu atau teknologi asistif dalam mengakses aplikasi, seperti *talback*, *screen reader*, dan *keyboard* [1].

2.2 Tinjauan Tempat Penelitian

Penelitian ini dilakukan pada beberapa tempat penelitian, baik dari pihak Setoko, maupun Sentra Wyata Guna sebagai lembaga yang menaungi tunanetra.

2.2.1 Setoko Web

Setoko Web merupakan aplikasi yang berbasis mobile dan web memfasilitasi bisnis *online* di Indonesia, terutama UMKM. Setoko Web menawarkan layanan pembuatan *website* toko online yang mudah dan cepat, memberikan kesempatan kepada pelaku bisnis untuk memperluas jangkauan penjualan mereka di ranah digital. Dengan fitur-fitur seperti pembuatan *website* toko *online* gratis, beragam metode pembayaran (termasuk VA, *e-wallet*, dan CC), serta fleksibilitas dalam pengaturan pembayaran dan pengiriman dengan berbagai kurir ekspedisi terkemuka, Setoko Web menjadi solusi yang menjanjikan bagi pengusaha untuk meningkatkan efisiensi penjualan produk mereka dalam era digitalisasi yang terus berkembang di Indonesia [3].



Gambar 2.1 Setoko Web

Berikut penjelasan lebih rinci terkait layanan yang disajikan [4]:

a. Peningkatan Penjualan Usaha

Setoko menghadirkan *platform website* toko *online* yang dirancang untuk membantu para pengusaha, baik UMKM maupun individu, dalam mengembangkan bisnis mereka secara *online*. *Platform* ini menawarkan keamanan data transaksi, pengiriman terintegrasi ekspedisi, beragam metode pembayaran, hingga analisis data pelanggan berdasarkan transaksinya [3].

b. *Branding*

Setoko Setoko menawarkan solusi inovatif bagi para pengusaha yang ingin membangun *branding website* toko *online* tanpa perlu memiliki keahlian *coding* ataupun desain. *Platform* ini memungkinkan untuk mengkustomisasi tampilan katalog produk dan memilih berbagai tema yang disediakan [3].

c. Fleksibilitas Penjualan

Terdapat beragam *channel* penjualan yang dapat membantu *merchant* dalam memasarkan produknya melalui Whatsapp, Facebook, YouTube, hingga TikTok [3].

2.2.2 Sentra Wyata Guna

Sentra Wyata Guna (SWG) merupakan lembaga yang didirikan oleh Kementerian Sosial Republik Indonesia (Kemensos RI) pada tahun 1963. SWG

memiliki fokus utama dalam penyuluhan dan pemberdayaan bagi penyandang tunanetra di Indonesia [LAMPIRAN A].

a. Visi

Menjadi lembaga terdepan dalam rehabilitasi sosial dan pemberdayaan penyandang tunanetra yang unggul, mandiri, dan berdaya saing [LAMPIRAN A].

b. Misi

Sentra Wyata Guna berdedikasi untuk meningkatkan kehidupan tunanetra di Indonesia. Lembaga ini melaksanakan rehabilitasi sosial dan pemberdayaan secara komprehensif dan profesional, serta mengembangkan model-model inovatif dan adaptif untuk membantu mereka mencapai kemandirian. Berbagai program dan kegiatan dirancang untuk meningkatkan kualitas hidup mereka, dan juga aktif dalam advokasi dan kerjasama dengan berbagai pihak untuk mewujudkan kesetaraan akses bagi penyandang tunanetra [LAMPIRAN A].

c. Program

Sentra wyata guna menyediakan pelatihan keterampilan, seperti pijat, menjahit, memasak, komputer, musik, dan mobilitas. Disamping itu, terdapat penyuluhan dan pemberdayaan masyarakat, seperti rehabilitasi sosial, kesetaraan akses, hingga kemandirian tunanetra [LAMPIRAN A].

2.3 Teknologi Asistif

Teknologi asistif adalah suatu inovasi yang memberikan kesempatan kepada individu dengan kebutuhan khusus, seperti tunanetra, untuk mengeksplorasi dan berinteraksi dengan komponen digital secara mendalam. Melalui teknologi ini, mereka dapat menikmati pengalaman estetika yang terwujud dalam konten visual tersebut. Dengan bantuan alat-alat yang dirancang khusus dan pengembangan sistem yang memungkinkan deskripsi detail serta interaksi yang intuitif, teknologi asistif memungkinkan tunanetra untuk menikmati dunia visual dengan cara yang sebelumnya mungkin sulit atau tidak mungkin bagi mereka. Teknologi asistif terdiri dari, *talkback*, *screen reader*, dan *keyboard*. Alat ini akan membaca komponen yang ada pada aplikasi satu persatu dan akan direpresentasikan dengan suara [2].

2.4 Desain Interaksi

Desain interaksi adalah proses merancang interaksi yang mencapai tujuan dari produk atau penyedia layanan beserta penggunanya, sekaligus mendukung tujuan pengguna. Ini berarti tidak hanya menciptakan pengalaman yang memenuhi kebutuhan dan harapan pengguna, tetapi juga memastikan bahwa produk atau layanan tersebut secara efektif mencapai tujuan perusahaan. Desain interaksi yang baik mempertimbangkan kebutuhan dan preferensi pengguna sambil tetap memastikan bahwa tujuan bisnis juga tercapai. Dengan demikian, desain interaksi adalah tentang menyelaraskan kebutuhan dan tujuan antara pihak-pihak yang terlibat, sehingga menciptakan pengalaman yang bermanfaat dan memuaskan bagi semua pihak yang terlibat. Terdapat beberapa prinsip yang mendukung desain interaksi, yaitu [12]:

1. Pemahaman Pengguna

Desain interaksi dimulai dengan memahami siapa pengguna produk atau sistem yang sedang dirancang. Ini melibatkan mengumpulkan informasi tentang kebutuhan, preferensi, kemampuan, dan konteks pengguna [12].

2. Definisi Tujuan

Tentukan tujuan yang ingin dicapai dengan produk atau sistem yang sedang dirancang. Ini bisa mencakup tujuan bisnis, tujuan pengguna, atau keduanya. Desain interaksi harus mempertimbangkan bagaimana pengguna akan berinteraksi dengan sistem dan memfasilitasi tindakan yang diinginkan [12].

3. Konsistensi

Konsistensi dalam desain antarmuka memainkan peran penting dalam membangun pengalaman yang baik bagi pengguna. Ini mencakup konsistensi dalam penggunaan simbol, terminologi, tata letak, dan pola interaksi di seluruh produk atau sistem [12].

4. Perancangan Konseptual

Ide-ide konseptual untuk antarmuka dan interaksi perlu dibuat. Ini melibatkan penentuan struktur informasi, navigasi, dan aliran kerja yang akan digunakan dalam produk atau sistem. [12].

5. Prototipe

Prototipe perlu diuji pada pengguna secara *real*. Prototipe dapat berupa sketsa kasar, *wireframe*, atau prototipe interaktif yang lebih canggih, tergantung pada tingkat kedalaman yang diinginkan [12].

6. Evaluasi

Kumpulkan umpan balik tentang pengalaman mereka. Ini dapat dilakukan melalui sesi pengujian pengguna formal, wawancara, atau observasi langsung [12].

7. Iterasi atau Perbaikan

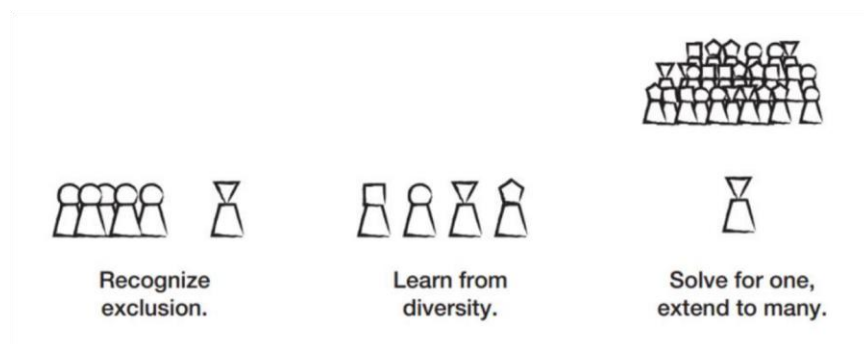
Perbaiki masalah yang ditemukan dan meningkatkan pengalaman pengguna. Proses ini dapat melibatkan beberapa iterasi hingga mencapai desain yang memuaskan [12].

8. Implementasi dan Pemeliharaan

Implementasikan desain interaksi dalam produk atau sistem yang sebenarnya. Ini melibatkan penerapan desain antarmuka, pengkodean, dan pengujian akhir sebelum peluncuran. Kemudian, pantau pengalaman pengguna untuk perbaikan berkelanjutan [12].

2.5 Desain Inklusif

Desain inklusif merupakan pendekatan yang mengutamakan pemahaman mendalam terhadap kebutuhan beragam pengguna dan upaya untuk mengurangi hambatan aksesibilitas. Melalui empati, kolaborasi lintas disiplin, dan penerapan prinsip *Universal Design*, desain inklusif bertujuan untuk menciptakan pengalaman yang tidak hanya dapat diakses secara fisik, tetapi juga secara emosional, mempromosikan rasa inklusi bagi semua individu. Terdapat tiga langkah desain inklusif secara fundamental, yaitu [10]:



Gambar 2. 2 Metode Desain Inklusif

1. Recognize Exclusion

Mengidentifikasi interaksi yang mengakibatkan kelompok masyarakat tertentu terpinggirkan, sehingga mereka tidak dapat menikmati pengalaman interaksi secara optimal [10].

2. *Learn from Human Diversity*

Mengidentifikasi interaksi yang mengakibatkan kelompok masyarakat tertentu terpinggirkan, sehingga mereka tidak dapat menikmati pengalaman interaksi secara optimal [10].

3. *Solve for One, Extend to Many*

Memberikan solusi yang memperhitungkan kebutuhan paling umum atau paling banyak dialami sehingga dapat dinikmati secara luas oleh banyak orang [10].

Konsep "The Cycle of Exclusion" yang dijelaskan dalam buku "*Mismatch: How Inclusion Shapes Design*" oleh Kat Holmes menggambarkan bagaimana kekurangan desain yang mengabaikan kebutuhan beragam pengguna dapat menciptakan lingkungan yang eksklusif. Dengan menerapkan konsep ini dalam merancang ide untuk aplikasi Setoko, kita dapat memecah siklus eksklusi yang terjadi bagi pengguna tunanetra. Berikut adalah analisis keadaan berdasarkan konsep tersebut:

1. Penjelasan Siklus Eksklusi

Identifikasi bagaimana aplikasi Setoko saat ini menciptakan siklus eksklusi bagi pengguna tunanetra dengan kurangnya aksesibilitas dan perhatian terhadap kebutuhan mereka.

2. Penciptaan Kesempatan Inklusi

Rancang ide untuk mengubah siklus eksklusi tersebut dengan menciptakan kesempatan inklusi melalui perubahan desain yang memperhatikan keberagaman pengguna. Misalnya, dengan memperbaiki label dan tata letak yang tidak terstruktur, serta meningkatkan dukungan untuk talkback dalam seluruh fitur aplikasi. Hal ini dapat dicapai melalui penerapan standar aksesibilitas web, seperti WCAG dan WAI-ARIA.

3. Penerapan Prinsip Desain Inklusif:

Terapkan prinsip-prinsip desain inklusif dalam merancang antarmuka aplikasi Setoko, seperti fleksibilitas, kesederhanaan, dan keberagaman. Hal ini dilakukan dengan memastikan desain aplikasi dapat diakses dan digunakan oleh semua pengguna, termasuk mereka yang memiliki disabilitas penglihatan.

4. Evaluasi Dampak dan Perbaikan Berkelanjutan:

Evaluasi dampak dari perubahan desain yang diterapkan dalam merancang ide tersebut terhadap pengalaman pengguna tunanetra. Lakukan perbaikan berkelanjutan berdasarkan umpan balik dari pengguna dan observasi langsung untuk memastikan inklusi yang berkelanjutan dalam pengembangan aplikasi.

Dengan menerapkan konsep "The Cycle of Exclusion" secara praktis dalam merancang ide untuk aplikasi Setoko, kita dapat mengubah siklus eksklusi menjadi siklus inklusi yang memperhatikan kebutuhan semua pengguna. Hal ini akan menciptakan lingkungan yang lebih ramah dan inklusif bagi semua pengguna, memastikan bahwa aplikasi Setoko dapat diakses dan digunakan dengan mudah oleh semua orang, tanpa terkecuali.

2.6 Accessibility

Accessibility merujuk pada kemampuan untuk mengakses dan menggunakan produk atau layanan secara efektif oleh semua individu, termasuk mereka yang memiliki berbagai jenis kebutuhan atau keterbatasan. Aksesibilitas dalam desain digital melibatkan pembuatan antarmuka, konten, dan fitur yang dapat diakses dengan mudah oleh semua pengguna, tanpa memandang latar belakang, kemampuan, atau kondisi mereka. Hal ini melibatkan berbagai strategi dan teknik desain, termasuk penggunaan struktur informasi yang jelas, navigasi yang intuitif, kontras yang baik antara teks dan latar belakang, serta dukungan untuk pembaca layar dan perangkat bantu aksesibilitas lainnya. Prinsip-prinsip aksesibilitas dalam desain digital bertujuan untuk memastikan bahwa semua individu dapat mengalami dan berinteraksi dengan produk atau layanan secara sepenuhnya, tanpa menghadapi hambatan yang tidak perlu [2].

2.7 UX Audit Heuristics Evaluation

Heuristics Evaluation adalah aturan praktis atau pedoman yang digunakan untuk menilai kualitas desain antarmuka pengguna berdasarkan prinsip-prinsip

desain yang terbukti secara empiris. Dalam konteks *UX Audit Heuristics Evaluation*, seorang penilai menggunakan seperangkat prinsip atau aturan praktis ini untuk menilai berbagai aspek pengalaman pengguna, seperti kegunaan, aksesibilitas, konsistensi, efisiensi, dan kepuasan pengguna [6].

Proses evaluasi biasanya melibatkan peninjauan secara mendalam terhadap antarmuka atau pengalaman pengguna yang ada, di mana penilai mencatat kelemahan dan kekuatan berdasarkan prinsip-prinsip heuristik yang telah ditetapkan sebelumnya. Setelah evaluasi selesai, hasilnya direkomendasikan untuk perbaikan desain atau pengembangan yang lebih baik. Terdapat 10 prinsip yang perlu dipenuhi, di antaranya [6]:

1. *Visibility of System Status*

Sistem harus memberikan umpan balik yang jelas kepada pengguna tentang status operasionalnya saat ini, seperti proses yang sedang berlangsung, kesalahan yang terjadi, hingga keberhasilan aksi yang dilakukan pengguna [6].

2. *Match Between System and The Real World*

Istilah, konsep, dan representasi yang digunakan dalam antarmuka harus mencerminkan dunia nyata sehingga mudah dipahami oleh pengguna [6].

3. *User Control and Freedom*

Pengguna harus diberikan kontrol atas sistem, termasuk kemampuan untuk membatalkan atau menghindari aksi yang tidak diinginkan [6].

4. *Consistency and Standards*

Antarmuka harus konsisten dalam seluruh aplikasi atau produk, serta mengikuti standar desain yang sudah ada untuk meningkatkan kegunaan dan meminimalkan kebingungan pengguna [6].

5. *Error Prevention*

Sistem harus dirancang untuk mencegah terjadinya kesalahan pengguna atau menyediakan mekanisme untuk mengoreksi kesalahan dengan mudah [6].

6. *Recognition Rather than Recall*

Informasi penting, opsi, dan aksi harus jelas dan mudah diakses sehingga pengguna tidak perlu mengandalkan ingatan mereka sendiri [6].

7. *Flexibility and Efficiency of Use*

Sistem harus memberikan opsi dan pintasan untuk pengguna berpengalaman agar dapat menyelesaikan tugas dengan cepat dan efisien, tanpa mengorbankan kegunaan bagi pengguna baru [6].

8. *Aesthetic and Minimalist Design*

Antarmuka harus dirancang dengan estetika yang menarik dan menggunakan elemen desain yang sederhana untuk mengurangi kebingungan dan gangguan [6].

9. *Help Users Recognize, Diagnose, and Recover from Errors*

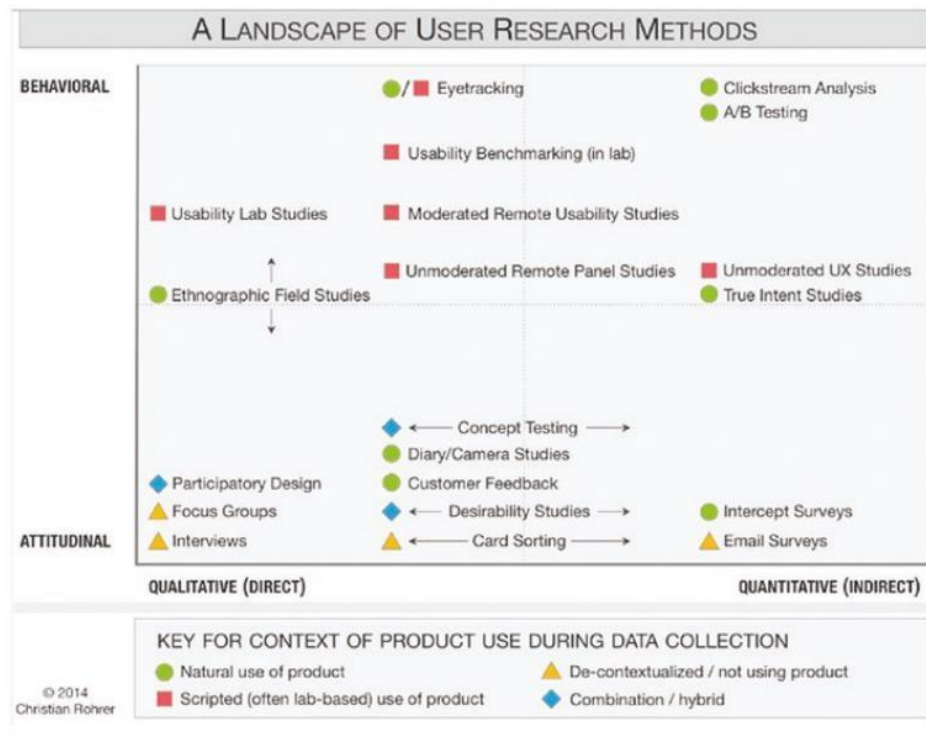
Sistem harus menyediakan pesan kesalahan yang jelas dan bermanfaat, serta panduan yang mudah dipahami untuk membantu pengguna menyelesaikan masalah [6].

10. *Help and Documentation*

Sistem harus menyediakan bantuan dan dokumentasi yang mudah diakses ketika diperlukan, tetapi desain yang baik seharusnya meminimalkan ketergantungan pada dokumentasi eksternal [6].

2.8 UX Research

UX Research adalah proses memahami kebutuhan, preferensi, dan perilaku pengguna. Ini adalah langkah penting dalam desain yang inklusif, karena membantu memastikan bahwa produk dan layanan dapat diakses dan digunakan oleh semua orang, termasuk penyandang disabilitas [2].



Gambar 2.3 Gambaran UX Research

Terdapat beberapa metode penelitian pengguna yang dapat digunakan, terdiri dari metode kualitatif dan kuantitatif [2]:

a. Kuantitatif

Mengumpulkan data numerik dan statistik untuk mengukur tren, pola, dan preferensi pengguna. Hal ini bertujuan untuk memberikan gambaran umum tentang bagaimana banyak pengguna berinteraksi dengan produk atau layanan, mengidentifikasi area perbaikan di mana mayoritas pengguna mengalami kesulitan. Dalam metode kuantitatif, terdapat beberapa kegiatan yang dapat dilakukan, di antaranya [2]:

1. Survei

Melibatkan meminta pengguna untuk menjawab pertanyaan tentang produk atau layanan. Ini dapat membantu mengumpulkan data kuantitatif tentang pengalaman pengguna [2].

2. Analisis Data

Menganalisis data penggunaan, seperti *log* dan data pelacakan mata. Ini dapat membantu memahami bagaimana pengguna berinteraksi dengan produk atau layanan [2].

3. *Usability Testing*

Pengamatan pengguna saat mereka mencoba menggunakan produk atau layanan. Ini dapat membantu mengidentifikasi masalah kegunaan dan area di mana desain dapat ditingkatkan [2].

4. Kualitatif

Memahami pengalaman, motivasi, dan tantangan pengguna melalui pemindaian, wawancara, dan pengamatan. Hal ini bertujuan untuk memberikan pemahaman yang kaya tentang perspektif pengguna, termasuk pengguna penyandang disabilitas, mengungkap motivasi dan hambatan mereka, serta menghasilkan ide-ide desain yang lebih inklusif. Dalam metode kualitatif, terdapat beberapa kegiatan yang dapat dilakukan, di antaranya [2]:

1. Wawancara

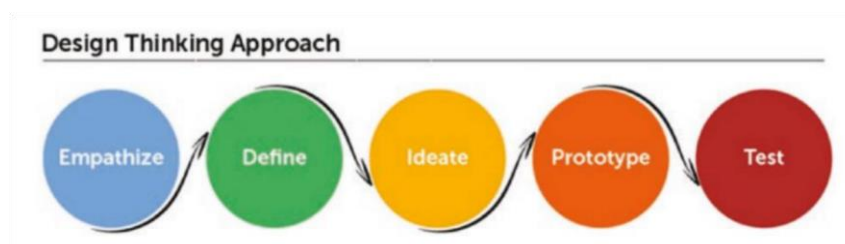
Berbicara dengan pengguna tentang pengalaman mereka dengan produk atau layanan. Ini dapat membantu memahami kebutuhan dan preferensi mereka, serta frustrasi dan tantangan mereka [2].

2. Etnografi

Memahami budaya dan masyarakat dari sudut pandang orang-orang yang hidup di dalamnya. Etnografi dilakukan dengan cara observasi partisipan, di mana peneliti menghabiskan waktu yang lama di lapangan untuk mengamati dan berinteraksi dengan orang-orang yang ditelitinya [2].

2.9 Design Thinking

Design Thinking adalah pendekatan atau metodologi iteratif yang digunakan dalam proses perancangan produk, layanan, atau solusi untuk masalah yang kompleks. Ini melibatkan serangkaian langkah-langkah yang berpusat pada pengguna dan berfokus pada memahami masalah, menghasilkan ide, dan menciptakan solusi yang inovatif dan efektif. Pendekatan ini menekankan empat tahap utama, di antaranya [2]:



Gambar 2.4 Metode Design Thinking

1. *Emphatize*

Memahami pengalaman, kebutuhan, dan tantangan pengguna dengan mendengarkan secara aktif dan mengamati [2].

2. *Define*

Merumuskan masalah dengan jelas dan spesifik berdasarkan wawasan dari tahap empati [2].

3. *Ideate*

Menghasilkan beragam gagasan dan solusi kreatif untuk menyelesaikan masalah yang telah ditentukan [2].

4. *Prototype*

Membangun model atau prototipe dari ide-ide yang dihasilkan untuk diuji dengan cepat dan murah [2].

5. *Test*

Mengujikan prototipe dengan pengguna untuk mengumpulkan umpan balik dan mengevaluasi solusi [2].

2.10 Web Content Accessibility Guideline (WCAG)

WCAG adalah seperangkat pedoman yang dikembangkan oleh *World Wide Web Consortium (W3C)* untuk membantu mengoptimalkan aksesibilitas konten web bagi semua individu, termasuk mereka yang memiliki berbagai kebutuhan atau keterbatasan. Pedoman ini mencakup berbagai aspek desain dan pengembangan web, termasuk teks, gambar, multimedia, navigasi, dan interaktivitas. Terdapat 4 prinsip (POUR) yang perlu dipenuhi, di antaranya [7]:

1. *Perceptible*

Memastikan bahwa semua informasi dan antarmuka dapat dipersepsikan oleh pengguna, termasuk mereka yang memiliki gangguan penglihatan atau pendengaran [7].

2. *Operable*

Memastikan bahwa antarmuka dan konten dapat dioperasikan oleh berbagai jenis perangkat input, termasuk *keyboard*, *mouse*, atau perangkat bantu aksesibilitas lainnya [7].

3. *Understandable*

Memastikan bahwa informasi dan operasi antarmuka dapat dipahami dengan mudah oleh pengguna, termasuk mereka yang memiliki keterbatasan kognitif atau bahasa [7].

4. *Robust*

Memastikan bahwa konten web dapat diakses dengan baik di berbagai perangkat dan platform, serta kompatibel dengan teknologi bantu aksesibilitas [7].

Terdapat klasifikasi penilaian standar WCAG yang digunakan sesuai dengan tujuan yang ingin dicapai. Pedoman Aksesibilitas Konten Web (WCAG) menjelaskan cara membuat konten web lebih mudah diakses oleh orang-orang dengan disabilitas. WCAG mencakup situs web, aplikasi, dan konten digital lainnya. Ini dikembangkan oleh Inisiatif Aksesibilitas Web (WAI) dari World Wide Web Consortium (W3C). WCAG adalah standar internasional. Dapat dilihat, sebagai berikut:

1. Level A

Tingkat dasar yang menyediakan aksesibilitas dasar bagi sebagian besar pengguna. Memenuhi kriteria pada tingkat ini memberikan aksesibilitas minimal [7].

2. Level AA

Tingkat yang lebih tinggi yang menyediakan aksesibilitas yang lebih baik, biasanya mencakup lebih banyak kriteria daripada tingkat "A". Memenuhi kriteria pada tingkat ini menghasilkan tingkat aksesibilitas yang lebih tinggi.[7].

3. Level AAA

tingkat tertinggi dalam hal aksesibilitas. Memenuhi kriteria pada tingkat ini menghasilkan tingkat aksesibilitas yang optimal atau paling tinggi [7].

Standar WCAG yang digunakan pada penelitian ini, di antaranya:

1. *WCAG Guidelines 1.4.5 – Images of Text (AA)*

Hindari penggunaan gambar untuk menampilkan informasi penting yang dibutuhkan pengguna untuk menyelesaikan suatu aktivitas. Teks dalam gambar

tidak bisa disesuaikan ukurannya dan akan terlihat buram saat diperbesar, sehingga bisa membingungkan dan mengganggu pengguna [7].

Pada standar ini, terdapat beberapa rekomendasi, yaitu:

Gambar yang mengandung informasi penting wajib dilengkapi dengan *tag alt* deskriptif dan jelas. Jangan hanya mengharapkan *developer* untuk menulis teks *alt text* karena mereka bertanggung jawab pada kode, bukan konten.

2. WCAG Guidelines 2.4.6 – *Heading and Labels* (AA)

Hindari penggunaan gambar untuk menampilkan informasi penting yang dibutuhkan pengguna untuk menyelesaikan suatu aktivitas. Teks dalam gambar tidak bisa disesuaikan ukurannya dan akan terlihat buram saat diperbesar, sehingga bisa membingungkan dan mengganggu pengguna [7].

Pada standar ini, terdapat beberapa rekomendasi, yaitu:

- Teks yang jelas dan tidak ambigu membantu pengguna dengan kesulitan membaca memahami topik dan tujuan konten. Ini juga membantu pengguna pembaca layar untuk menavigasi ke bagian konten yang relevan di halaman.
- Saat merancang tata letak, perlu dipertimbangkan bagaimana teks dibagi dan bagaimana formulir diberi label.
- Hindari penyajian *state* hanya dengan warna saja, perlu diberikan deskripsi secara tertulis
- Berikan label dan informasi pada setiap komponen visual, seperti gambar, grafik, dll.

3. WCAG Guidelines 1.3.2 – *Meaningful Sequence* (A)

Atur konten dalam desain untuk memastikan urutan membaca dan navigasi menjadi logis dan intuitif [7].

Pada standar ini, terdapat beberapa rekomendasi, yaitu:

- Jika urutan konten penting, pengguna harus dapat mengaksesnya dalam urutan tersebut. Ini berlaku meskipun gaya visualnya dinonaktifkan, seperti pengguna yang menonaktifkan CSS atau mengandalkan pembaca layar.
- Pelajari tentang penanda semantik pada halaman *web*. Mengatur konten dengan cermat akan memastikan struktur yang benar pada perangkat pengguna. Ini akan menghemat waktu dan kerumitan selama pengembangan.

4. WCAG Guidelines 3.2.3 – Consistent Navigation (AA)

Menjaga keseragaman menu dan struktur halaman antarmuka di semua halaman. Bayangkan seperti peta - jika tata letaknya berubah setiap halaman, pengguna akan bingung dan tersesat [7].

Pada standar ini, terdapat rekomendasi, yaitu:

Jika produk memiliki menu navigasi atau *breadcrumb* yang digunakan di banyak halaman, pastikan tampilan dan fungsinya selalu sama di setiap halaman.

5. WCAG Guidelines 1.3.3 – Sensory Characteristics (A)

Jangan hanya menggunakan bentuk, ukuran, atau posisi untuk menjelaskan sesuatu. Ingat, tidak semua orang bisa melihat dengan baik, dan beberapa memiliki kesulitan memproses informasi visual [7].

Pada standar ini, terdapat beberapa rekomendasi, yaitu:

Hindari instruksi seperti "klik tombol merah untuk melanjutkan" atau "instruksi ada di kolom kanan".

6. WCAG Guidelines 3.2.4 – Consistent Identification (AA)

Ketika fitur yang sama muncul di berbagai bagian aplikasi atau website, pastikan pengguna bisa mengenalinya dengan mudah. Artinya, gunakan istilah, ikon, atau penanda yang sama persis setiap kali fitur tersebut muncul [7].

Pada standar ini, terdapat beberapa rekomendasi, yaitu:

- Label tombol dan tautan harus konsisten sepanjang perjalanan pengguna, misalnya, "Lanjut" atau "Teruskan".
- Jika bidang formulir memiliki tujuan yang sama, mereka harus diberi label dengan konsisten di mana pun digunakan.
- Jaga agar ikon tetap konsisten jika digunakan untuk tujuan tertentu.

7. WCAG Guidelines 4.1.2 – Name, Role, Value (A)

Ketika Kriteria ini memastikan bahwa untuk semua komponen antarmuka pengguna (seperti elemen *form*, *link*, dan komponen antarmuka yang dibuat oleh *script*).

- Nama (*Name*): Komponen harus memiliki nama yang dapat diprogram sehingga teknologi bantu (seperti pembaca layar) dapat mengidentifikasinya.

- Peran (*Role*): Komponen harus memiliki peran yang dapat diprogram sehingga teknologi bantu dapat memahami fungsi komponen tersebut.
- Nilai (*Value*): Jika komponen dapat ditetapkan nilai oleh pengguna, maka nilai tersebut harus dapat diprogram dan diperbarui ketika terjadi perubahan.

8. WCAG Guidelines 4.1.3 – Status Messages (AA)

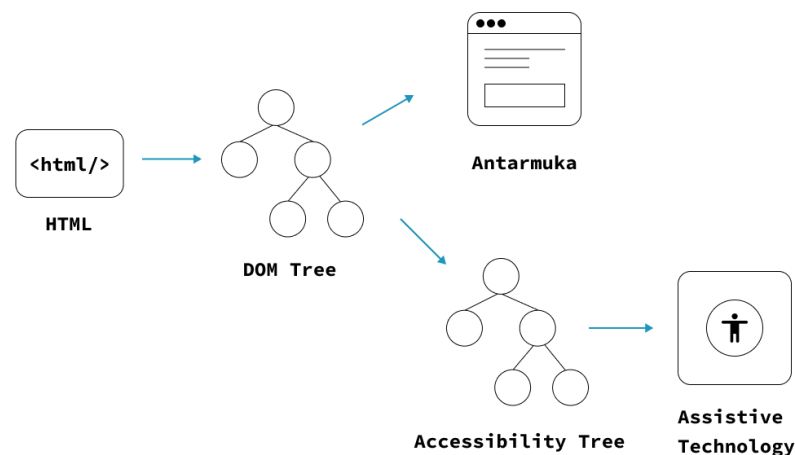
Kriteria ini memastikan bahwa pesan status yang tidak menyebabkan perubahan dalam konteks (seperti hasil pencarian, pesan kesalahan, atau notifikasi keberhasilan) dapat diprogram sehingga dapat diidentifikasi oleh teknologi bantu tanpa menerima fokus.

Setelah mengobservasi elemen antarmuka berdasarkan prinsip WCAG, dilakukan perhitungan tingkat kepatuhan untuk mengukur seberapa baik situs web atau aplikasi web memenuhi kriteria aksesibilitas yang ditetapkan oleh W3C (*World Wide Web Consortium*). Perhitungan ini melibatkan jumlah halaman yang sesuai dan tidak sesuai dengan standar WCAG. Kemudian, jumlah tersebut dihitung persentasenya dengan persamaan berikut [19]:

$$\text{Inaccessibility score} = \frac{\text{Number of violated Success Criteria}}{\text{Total number of Success Criteria}} \times 100\% \quad (2.1)$$

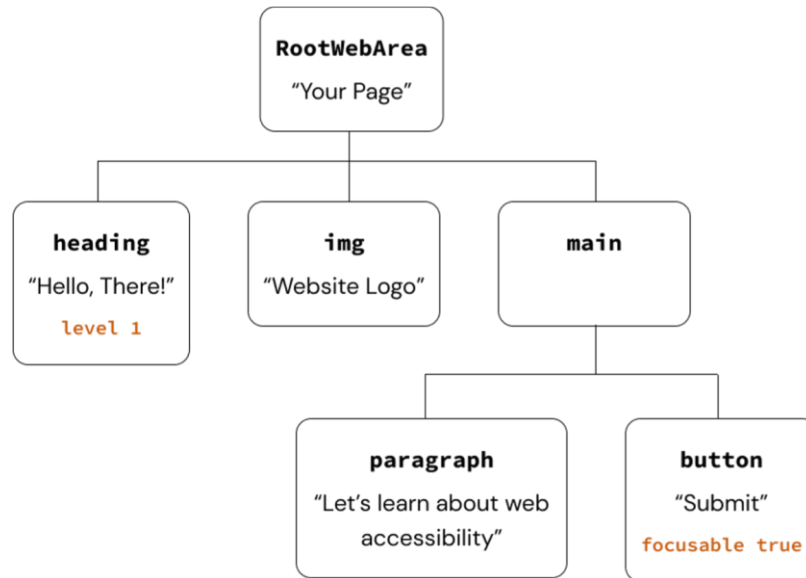
2.11 Accessible Rich Internet Applications (ARIA)

Accessible Rich Internet Applications (ARIA) adalah seperangkat atribut yang dapat ditambahkan ke kode HTML untuk membuat konten web lebih mudah diakses oleh pengguna dengan disabilitas. ARIA memungkinkan pengembang web untuk memberikan informasi tambahan tentang struktur dan makna elemen web, yang dapat digunakan oleh teknologi asistif, seperti *talkback*, *screen reader* untuk mengakses *website* menggunakan suara [8].



Gambar 2. 5 Accessible Rich Internet Application (ARIA)

Hal ini dapat direpresentasikan secara visual dari bagaimana elemen-elemen di dalam antarmuka pengguna (UI) dan konten web disusun dan dihubungkan, melalui *accessibility tree*. Dapat dilihat pada gambar di bawah ini [8]:



Gambar 2. 6 Accessibility Tree

Terdapat konsep ARIA yang dapat diimplementasikan, di antaranya:

2.11.1 Semantik

Seiring dengan meningkatnya kompleksitas situs web, penggunaan *tag* HTML semakin berkembang. Teknologi seperti pembaca layar akan membaca situs web sesuai dengan struktur HTML [8].

```

<html>
  <head></head>
  <body>
    <header></header>
    <nav></nav>
    <main>
      <article></article>
      <button></button>
    </main>
    <footer></footer>
  </body>
</html>
  
```

The image shows a code editor window with a light blue background and a title bar with red, yellow, and green buttons. It displays the following HTML code:

Gambar 2. 7 Semantik HTML

Penataan *tag* HTML yang baik dan sesuai dengan hirarki fungsionalitasnya akan sangat membantu teknologi ini. Berikut ini contoh semantik HTML yang digunakan [8]:

1. Struktur Utama

`<header>` untuk menampung informasi header seperti logo, judul situs, dan navigasi utama.

`<main>` untuk menampung konten utama situs web.

`<footer>` untuk menampung informasi footer seperti informasi kontak, *copyright*, dan tautan ke media sosial.

2. Konten Sekunder

`<aside>` untuk menampung konten sekunder seperti *sidebar*, *widget*, dan iklan.

`<section>` untuk menampung bagian konten yang terhubung secara logis, seperti artikel, *blog post*, atau produk.

3. Navigasi

`<nav>` untuk menampung elemen navigasi seperti menu dan *breadcrumb*.

4. Heading

`<h1>` - `<h6>` untuk menampung judul dengan hirarki tertentu

2.11.2 Polifil

Polifil adalah kode *JavaScript* yang meniru fungsionalitas fitur browser yang belum didukung secara *native*. Dalam konteks aksesibilitas, polifil dapat digunakan untuk menambahkan dukungan ARIA ke browser yang lebih lama dan meningkatkan fungsionalitas teknologi bantu. Berikut adalah beberapa contoh penggunaan ARIA [8]:

Tabel 2. 1 Gambaran Polifil

Polifil	Tujuan
<code>aria-label="Gambar produk"</code>	Menambahkan label teks ke gambar
Polifil	Tujuan
<code>aria-role="button"</code>	Menjelaskan fungsi tombol
<code>aria-disabled="true"</code>	Menunjukkan status elemen formulir

2.11.3 Tengarane (Landmark)

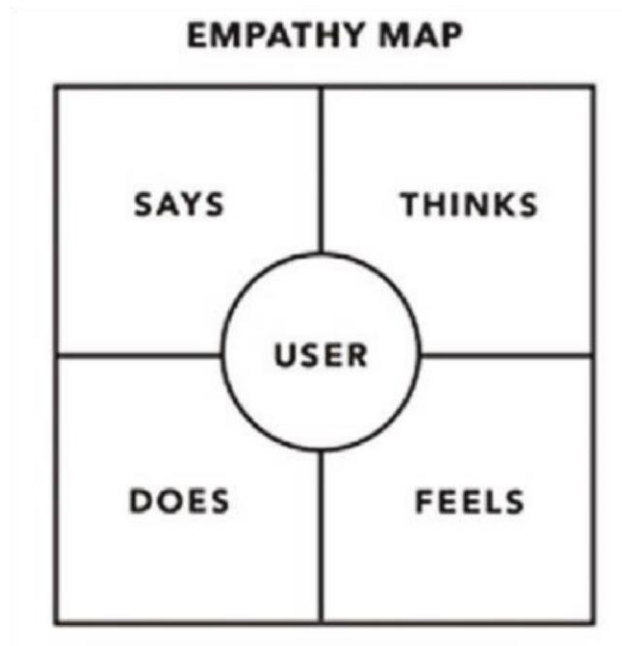
Tengaran dalam situs *web*, layaknya *landmark* dalam dunia nyata, berfungsi sebagai penunjuk arah yang membantu pengguna menemukan informasi yang mereka cari dengan mudah. Tengaran *web* membantu pengguna menavigasi dan memahami struktur situs web dengan lebih baik, terutama bagi teknologi asistif [8]:.

Tabel 2. 2 Gambaran Tengaran

Tengaran	Tujuan
aria-label	Pelabelan komponen tampilan agar dikenali dan dapat dibaca oleh teknologi asistif
aria-live	Digunakan untuk komponen yang membutuhkan pembaruan konten, seperti <i>pop-up</i>
aria-required	Menentukan komponen yang wajib diisi. Hal ini akan dibaca oleh teknologi asistif
aria-hidden	Membuat komponen yang tidak perlu dibaca oleh teknologi asistif
tabIndex	Membuat komponen dapat diakses oleh “ <i>tab</i> ” pada <i>keyboard</i>
Atribut lang	Menentukan bahasa yang akan dibaca oleh teknologi asistif
aria-labelledby	Pelabelan didasarkan pada komponen dengan <i>id</i> tertentu
aria-controls	Membuat suatu komponen yang bergantung pada komponen lain
region	Membuat suatu daerah yang berisi komponen di dalamnya

2.12 Emphaty Map

Dalam melakukan desain, diperlukan nilai empati untuk mendukung ide yang akan dihasilkan. Akan tetapi, pentingnya melampaui pemahaman emosional semata, tetapi juga mencakup empati kognitif, yang melibatkan perspektif orang lain dan memahami pengalaman mereka [10]. *Empathy Map* merupakan alat visual yang digunakan untuk membantu desainer memahami kebutuhan, perasaan, dan motivasi pengguna. Dalam konteks desain inklusif, peta empati dapat membantu desainer memahami pengalaman pengguna penyandang disabilitas saat menggunakan produk atau layanan digital. [2].



Gambar 2. 8 Empathy Map

Terdapat 4 komponen *empathy map* yang perlu didefinisikan berdasarkan hasil *user research*, di antaranya [2]:

1. *Says*

Apa yang pengguna katakan secara verbal maupun non-verbal? Keluhan, komentar, pendapat, dan pujian apa yang mereka sampaikan? [2]

2. *Does*

Apa yang pengguna lakukan secara nyata? Tindakan dan perilaku apa yang mereka tunjukkan saat menggunakan produk atau layanan? Setelah merasakan emosi tertentu, apa tindakan selanjutnya yang dilakukan pengguna? Apakah mereka terus menggunakan produk, meninggalkannya, atau mencari solusi lain? [2]

3. *Thinks*

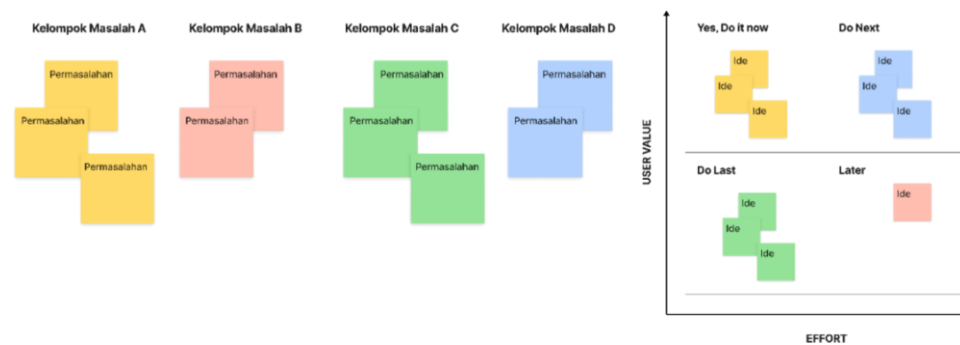
Apa yang pengguna pikirkan dan rasakan di dalam hati mereka? Apa harapan, kekhawatiran, dan keyakinan mereka? [2]

4. *Feels*

Apa emosi yang dirasakan pengguna? Kegembiraan, frustrasi, kebingungan, atau kekecewaan? [2]

2.13 Affinity Diagram

Affinity Diagram adalah alat atau teknik yang digunakan dalam proses pemikiran kreatif atau manajemen pemecahan masalah untuk mengorganisir dan menyusun gagasan, informasi, atau temuan menjadi kelompok yang terkait atau memiliki kesamaan tema atau pola. Gambaran *affinity diagram* dapat dilihat pada gambar di bawah ini [9]:

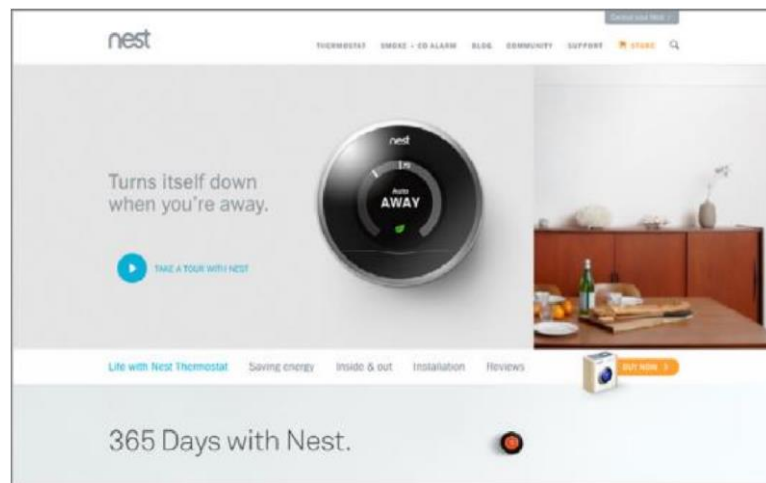


Gambar 2. 9 Affinity Diagram

Ide dipilih menurut urutan kepentingannya. Pengelompokkan prioritas terbagi dalam empat kategori, yang didasarkan pada kebutuhan pengguna dan upaya yang diperlukan untuk mewujudkannya. Semakin besar nilai bagi pengguna dan semakin sedikit upaya yang diperlukan, maka prioritasnya akan semakin tinggi. Sebaliknya, semakin kecil nilai bagi pengguna dan semakin besar upaya yang diperlukan, maka prioritasnya akan semakin rendah [9].

2.14 Website

Sebuah situs *web* adalah kumpulan halaman web terkait dan aset digital lainnya yang disajikan kepada pengguna. Situs *web* dapat digunakan untuk berbagai tujuan, termasuk menyediakan informasi, menjual barang atau jasa, dan memfasilitasi komunikasi. Pengguna dapat berinteraksi dengan komponen-komponen tampilan yang disediakan. Contoh *website* sebagai berikut [12]:



Gambar 2. 10 Contoh Website

Situs web biasanya terdiri dari kode *HTML (HyperText Markup Language)*, yang menentukan struktur dan konten halaman [20]. *CSS (Cascading Style Sheets)* digunakan untuk mengatur tampilan halaman, dan *JavaScript* digunakan untuk menambahkan interaktivitas [12]. Terdapat beberapa teknologi tambahan yang digunakan pada rise ini, di antaranya:

1. *Tailwind CSS*

Kerangka kerja (*framework*) untuk membangun antarmuka (*interface*) *website* dengan cepat dan mudah. Hal ini menggunakan pendekatan kelas utilitas kecil disediakan untuk mengatur berbagai aspek tampilan seperti warna, *padding*, *margin*, *font*, *layout*, dan lain-lain [13].

2. *NextJS*

NextJS adalah kerangka kerja yang dirancang untuk membangun *website* yang cepat, efisien, dan berperforma tinggi. Kerangka kerja ini menawarkan berbagai fitur canggih seperti *server-side rendering (SSR)* dan *static site generation (SSG)*, *routing built-in*, *API routes*, dan banyak lagi. Hal ini untuk meningkatkan performa dan kecepatan kerja dalam merancang *website* [14].

2.15 Usability Testing

Usability Testing adalah proses untuk mengevaluasi seberapa mudah dan efisien pengguna dapat menggunakan suatu produk. Dalam konteks desain *web*, pengujian *usability* membantu memastikan bahwa situs *web* mudah dinavigasi, dipahami, dan digunakan oleh semua orang, termasuk orang disabilitas. Hal ini merupakan bagian penting dari proses desain *web* yang inklusif. Dengan melakukan pengujian *usability*, desainer dan pengembang dapat memastikan bahwa situs *web*

mereka dapat diakses dan digunakan oleh semua orang [3]. Terdapat dua metode *usability testing* yang dapat dilakukan:

1. *User Acceptance Testing (UAT)*

UAT adalah proses formal untuk memvalidasi apakah suatu sistem atau aplikasi memenuhi kebutuhan dan persyaratan pengguna. Pengujian ini dilakukan oleh pengguna akhir atau perwakilan mereka untuk memastikan bahwa sistem atau aplikasi dapat diterima dan digunakan secara efektif. Langkah-langkah UAT, sebagai berikut [11]:



Gambar 2. 11 Metodologi UAT

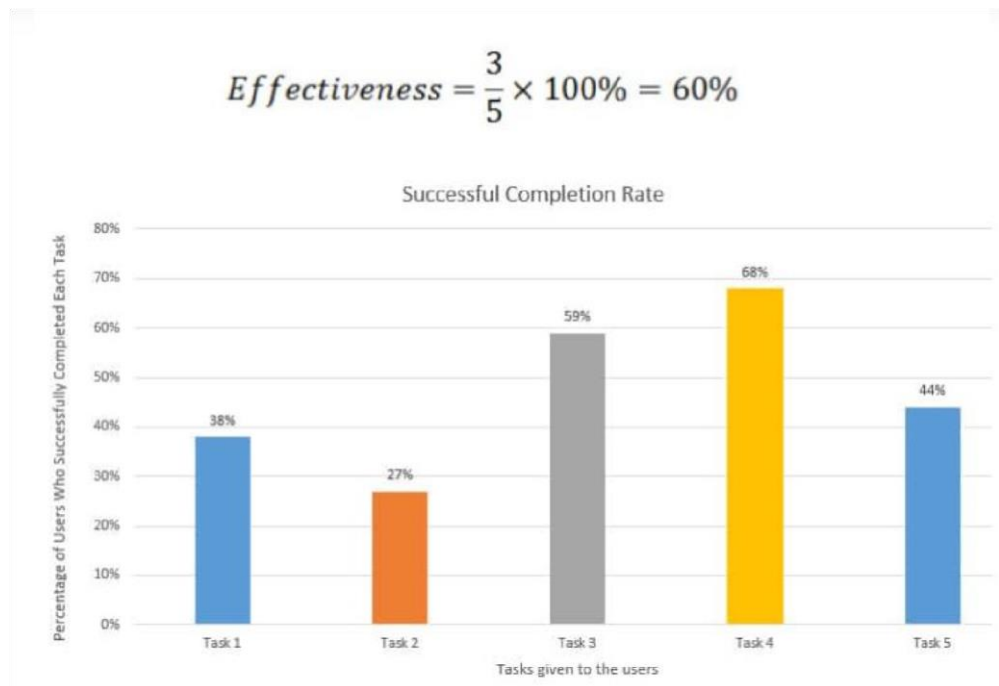
2. *UX Metrics Effectiveness*

Perhitungan efektivitas mengacu pada kemampuan atau tingkat keberhasilan dalam mencapai tujuan dalam jangka waktu tertentu. Ini berkaitan dengan bagaimana efisien dan optimal waktu yang tersedia untuk menyelesaikan suatu tugas atau proyek. Perhitungan dapat ditulis dalam rumus berikut [7]:

$$Effectiveness = \frac{Number\ of\ tasks\ completed\ successfully}{Total\ number\ of\ tasks\ undertaken} \times 100\% \quad (2.2)$$

Perhitungan tersebut dihitung berdasarkan hasil *usability testing* yang dilakukan secara langsung berdasarkan tugas-tugas tertentu. Kemudian, hitung jumlah tugas yang berhasil diselesaikan oleh pengguna dibagi total jumlah tugas. Contoh perhitungannya, sebagai berikut [5]:

Lima pengguna mengerjakan tugas yang sama menggunakan sistem yang sama. Di akhir sesi pengujian, 3 pengguna berhasil mencapai tujuan tugas, sedangkan 2 pengguna lainnya tidak. Untuk menghitung keefektifan keseluruhan sistem berdasarkan pengguna, maka digunakan persamaan berikut [5]:



Gambar 2. 12 Contoh Perhitungan UX Metrics Effectiveness

Standar minimum *time effectiveness* yang ditetapkan adalah 78%. Ini berarti bahwa dalam pengujian pengguna, tugas yang diberikan kepada mereka harus diselesaikan oleh setidaknya 78% dalam jumlah tugas yang telah ditentukan [5]. Dengan memiliki standar minimum yang jelas seperti ini, perancang antarmuka dapat menilai seberapa efektif desain mereka. Dalam perhitungan di atas, terlihat bahwa antarmuka masih berada pada batas minimum keefektifan, yaitu hanya 60%.