

BAB 2

LANDASAN TEORI

2.1. Google Chrome Browser

Google Chrome adalah browser situs lintas platform yang dikembangkan oleh Google. Ini pertama kali dirilis pada 2008 untuk Microsoft Windows dan kemudian dikembangkan untuk Linux, macOS, iOS, dan Android. Adapun versi yang digunakan pada penelitian ini adalah Google Chrome 79.0. *Browser* Google Chrome memiliki beberapa produk yang dapat didistribusikan antara lain adalah *Extension*, *Chrome Apps* dan *Themes*. Pada penelitian ini.

2.1.1. Ekstensi Browser

Ekstensi Browser merupakan sebuah software module pada browser yang bertujuan untuk memperluas fungsionalitas yang dimiliki oleh sebuah web browser. Beberapa web browser besar yang menggunakan teknologi ekstensi browser adalah Google Chrome, Mozilla Firefox, dan Internet Explorer. Istilah lain yang digunakan untuk menyebut ekstensi browser adalah Add-on dan Plug-in. Untuk membangun perangkat lunak ekstensi browser ini data digunakan beberapa bahasa pemrograman diantaranya Javascript, HTML dan CSS [8]. Untuk memperluas fungsi dari ekstensi yang akan dibangun, Google Chrome menyediakan beberapa komponen yang dapat digunakan pengembang. Adapun komponen-komponen tersebut adalah sebagai berikut.

1. Komponen *Interface*

Komponen ini berfungsi untuk mengatur dan mendesain antarmuka yang akan disajikan oleh ekstensi yang akan dibangun, adapun komponen-komponen tersebut dapat dilihat pada Tabel 2.1.

Tabel 2.1 Komponen *Interface*

Nama Komponen	Keterangan
<i>Browser Actions</i>	Menambahkan <i>icon</i> , <i>tooltip</i> , <i>badge</i> dan <i>popup</i> pada <i>toolbar</i> Google Chrome
<i>Commands</i>	Menambahkan <i>shortcuts</i> keyboard yang akan men-trigger aksi.

<i>Context Menus</i>	Menambahkan item menu konteks pada Google Chrome.
<i>Omnibox</i>	Menambahkan fungsi <i>keyboard</i> pada bagian <i>address bar</i> .
<i>Override Pages</i>	Membuat halaman <i>Tab</i> baru, <i>Bookmark</i> , atau <i>Riwayat</i> .
<i>Page Actions</i>	Menampilkan <i>icon</i> secara dinamis.

2. Komponen *Utilities*

Komponen ini berfungsi untuk mendukung segala keperluan yang dibutuhkan untuk membangun ekstensi, adapun komponen-komponen tersebut dapat dilihat pada Tabel 2.2.

Tabel 2.2 Komponen *Utilities*

Nama Komponen	Keterangan
<i>Accessibility (a11y)</i>	Buat ekstensi dapat diakses oleh para penyandang disabilitas.
<i>Background Scripts</i>	Mendeteksi dan bereaksi ketika sesuatu terjadi.
<i>Internationalization</i>	Befungsi untuk mengatur bahasa
<i>Identify</i>	Untuk mengakses <i>OAuth2 access tokens</i> .
<i>Management</i>	Mengelola ekstensi yang diinstal dan dijalankan.
<i>Message Passing</i>	Berkomunikasi dari <i>Content Scripts</i> ke ekstensi induknya, atau sebaliknya.
<i>Options Pages</i>	Membuat pengguna agar dapat mengatur ekstensinya.
<i>Permissions</i>	Mengatur perizinan pada Google Chrome
<i>Storage</i>	Berfungsi menyimpan dan ambil data.

3. Komponen *Browser*

Komponen ini berfungsi untuk mengubah ataupun mengatur *browser* ketika ekstensi yang akan dibangun aktif, adapun komponen-komponen tersebut dapat dilihat pada Tabel 2.3.

Tabel 2.3 Komponen *Browser*

Nama Komponen	Keterangan
<i>Bookmarks</i>	Membuat, mengatur, dan memanipulasi perilaku <i>bookmark</i> .
<i>Browsing Data</i>	Menghapus data penjelajahan dari profil lokal pengguna.
<i>Downloads</i>	Secara terprogram memulai, memantau, memanipulasi, dan mencari unduhan.
<i>Font Settings</i>	Mengatur <i>font</i> dalam Chrome.
<i>History</i>	Berinteraksi dengan catatan halaman yang dikunjungi <i>browser</i> .
<i>Privacy</i>	Mengontrol fitur privasi pada Google Chrome.
<i>Proxy</i>	Mengontrol fitur <i>proxy</i> pada Google Chrome

<i>Sessions</i>	Mengatur <i>Query</i> dan <i>Restore tab</i> dan jendela dari sesi penjelajahan.
<i>Tabs</i>	Berfungsi untuk membuat, mengubah, dan mengatur ulang <i>tab</i> di browser.
<i>Top Sites</i>	Menampilkan situs yang sering dikunjungi
<i>Themes</i>	Ubah keseluruhan tampilan browser.
<i>Windows</i>	Berfungsi untuk membuat, mengubah, dan mengatur ulang jendela di browser.

4. Komponen Situs

Komponen ini berfungsi untuk mengubah dan mengatur situs tertentu ketika ekstensi yang dibangun aktif, adapun komponen-komponen tersebut dapat dilihat pada Tabel 2.4.

Tabel 2.4 Komponen Mengubah Struktur Situs

Nama Komponen	Keterangan
<i>Active Tab</i>	Mengakses situs dengan aman dengan menghapus sebagian besar kebutuhan untuk izin <i>host <all_urls></i> .
<i>Content Settings</i>	Kustomisasi fitur situs seperti <i>cookie</i> , <i>JavaScript</i> , dan <i>plug-in</i> .
<i>Content Scripts</i>	Menjalankan kode <i>JavaScript</i> dalam konteks halaman situs.
<i>Cookies</i>	Menjelajahi dan modifikasi sistem <i>cookie</i> pada browser.
<i>Cross-Origin XHR</i>	Menggunakan <i>XMLHttpRequest</i> untuk mengirim dan menerima data dari <i>server</i> jarak jauh.
<i>Declarative Content</i>	Melakukan tindakan pada konten halaman tanpa memerlukan izin.
<i>Desktop Capture</i>	Mengabadikan konten layar, jendela atau tab tersendiri.
<i>Page Capture</i>	Menyimpan informasi sumber tab sebagai <i>MHTML</i> .
<i>Tab Capture</i>	Berinteraksi dengan aliran media tab.
<i>Situs Navigation</i>	Pembaruan status permintaan navigasi dalam penerbangan.
<i>Situs Request</i>	Mengamati dan analisis lalu lintas. Mencegat blok, atau mengubah permintaan dalam penerbangan.

2.2. Perangkat Lunak Pendukung

Perangkat lunak pendukung merupakan perangkat yang berupa aplikasi, bahasa pemrograman dan lainnya. Perangkat lunak ini digunakan sebagai pendukung dalam pembangunan perangkat lunak dalam penelitian ini.

Berikut merupakan beberapa perangkat lunak pendukung yang diperlukan pada penelitian ini.

2.2.1. JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON sendiri memiliki struktur data yang terdiri dari *json array* dan *json object*. Berikut ini adalah penjelasan dari kedua bagian tersebut.

1. JSON Array

JSON Array adalah kumpulan nilai yang terurutkan. Array dimulai dengan kurung kotak buka ([) dan diakhiri dengan kurung kotak tutup (]). Setiap nilai/*value* dipisahkan oleh tanda koma (,).

2. JSON Object

JSON Object adalah kumpulan pasangan dari string dan *value*. Objek dimulai dengan kurung kurawal buka ({) dan diakhiri dengan kurung kurawal tutup (}). Setiap pasangan string dan value diikuti oleh titik dua (:) dan diakhir setiap pasangan dipisahkan oleh tanda koma (,).

2.2.2. HTML

HTML (*Hypertext Markup Language*) adalah bahasa yang digunakan dalam menulis halaman situs. HTML merupakan pengembangan dari standar pemformatan dokumen teks, yaitu *Standard Generalized Markup Language* (SGML). HTML pada dasarnya merupakan dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu. Kegunaan Bahasa ini ialah untuk memanipulasi browser sehingga dapat menampilkan informasi yang dapat dibaca oleh pengguna komputer [9].

Kode html dibuat dalam file teks biasa yang disimpan dengan ekstensi “*.htm” atau “*.html”. 26 kode HTML terdiri dari tag-tag yang memiliki fungsi yang unik. Tag berarti penanda item baik yang akan ditampilkan oleh situs browser maupun tidak. Biasanya tag ini ditulis berpasangan dan mengapit item yang akan dijelaskan oleh tag tersebut. Tag tidak case sensitive, jadi bisa menggunakan huruf

kapital maupun huruf kecil atau keduanya dan akan menghasilkan output yang sama. Dokumen HTML dibagi menjadi tiga bagian utama yaitu tag HTML, Head, dan Body. Untuk memulai bekerja dengan HTML bisa menggunakan teks editor apa saja, seperti *Notepad*, *Notepad++*, *Sublime-text*, *Adobe Dreamweaver*, dan lainnya [9].

2.2.3. Javascript

Javascript adalah bahasa script berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML. Di mana objek tersebut dapat berupa suatu window, frame, URL, dokumen, *form*, *button* atau item yang lain. Yang mana item tersebut memiliki sebuah property yang saling berhubungan dengannya, dan masing-masing memiliki nama, lokasi, warna nilai, dan atribut lain [9].

Kode javascript dapat disisipkan pada dokumen html “*.htm” atau dapat pula disimpan pada file terpisah dengan ekstensi “*.js” lalu dipanggil dari dokumen html dengan tujuan agar kode javascript dapat lebih reusable. Kelebihan menggunakan client side scripting adalah dapat digunakan untuk membangun sebuah tampilan yang dinamis dan lebih atraktif. Namun, kekurangannya adalah karena merupakan *client-side scripting*, maka kode program dapat di lihat oleh browser pengguna, hal ini dapat menimbulkan celah keamanan jika salah dalam penggunaannya.

2.2.4. CSS

CSS (*Cascading Style Sheet*) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu situs, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk memformat halaman situs yang ditulis dengan HTML atau XHTML [9].

Kode css itu sendiri dapat kita sisipkan langsung pada file html, dan bisa juga kita tuliskan pada file terpisah yang berekstensi “*.css”. Pada dasarnya tidak ada ketentuan dalam penulisan kode stylesheet di file html ataupun terpisah di file css. Namun bila memiliki banyak file html yang harus di kelola, maka menuliskan

kode *stylesheet* pada file css terpisah merupakan pilihan yang lebih baik dalam menghindari penulisan kode yang sama berulang kali dan membuat kode *stylesheet* menjadi lebih *reusable*.

2.2.5. JQuery

JQuery adalah *library JavaScript* multiplatform yang dirancang untuk memudahkan penyusunan *client-side* script pada file HTML. Sintaks *jQuery* dirancang sedemikian rupa untuk memudahkan pengembang situs dalam menavigasi dokumen, menyeleksi elemen-elemen dari *Document Object Model*, menerapkan animasi, mengaplikasikan events, serta membangun aplikasi AJAX [10].

Fitur-fitur inti *jQuery* – meliputi penyeleksian, traversal, dan manipulasi elemen-elemen dari *Document Object Model* – dimampukan oleh sebuah *selector engine* (dinamai *Sizzle* mulai versi 1.3), telah mewujudkan suatu gaya pemrograman baru yang memadukan antara algoritma dan struktur data *Document Object Model*. Gaya ini telah mempengaruhi arsitektur dari framework JavaScript lainnya seperti YUI v3 dan Dojo, dan di kemudian menstimulasi pengembangan *Selectors API* standar [10].

2.3. URL

URL (*Uniform Resource Locator*) adalah cara mengidentifikasi lokasi file di internet. Jadi dalam garis besar, URL adalah sebuah cara bagaimana kita menemukan sebuah lokasi file yang berada di internet, Baik berupa situs, gambar, video, program perangkat lunak, ataupun jenis file lainnya yang di-host di server. Adapun fungsi url adalah sebagai berikut :

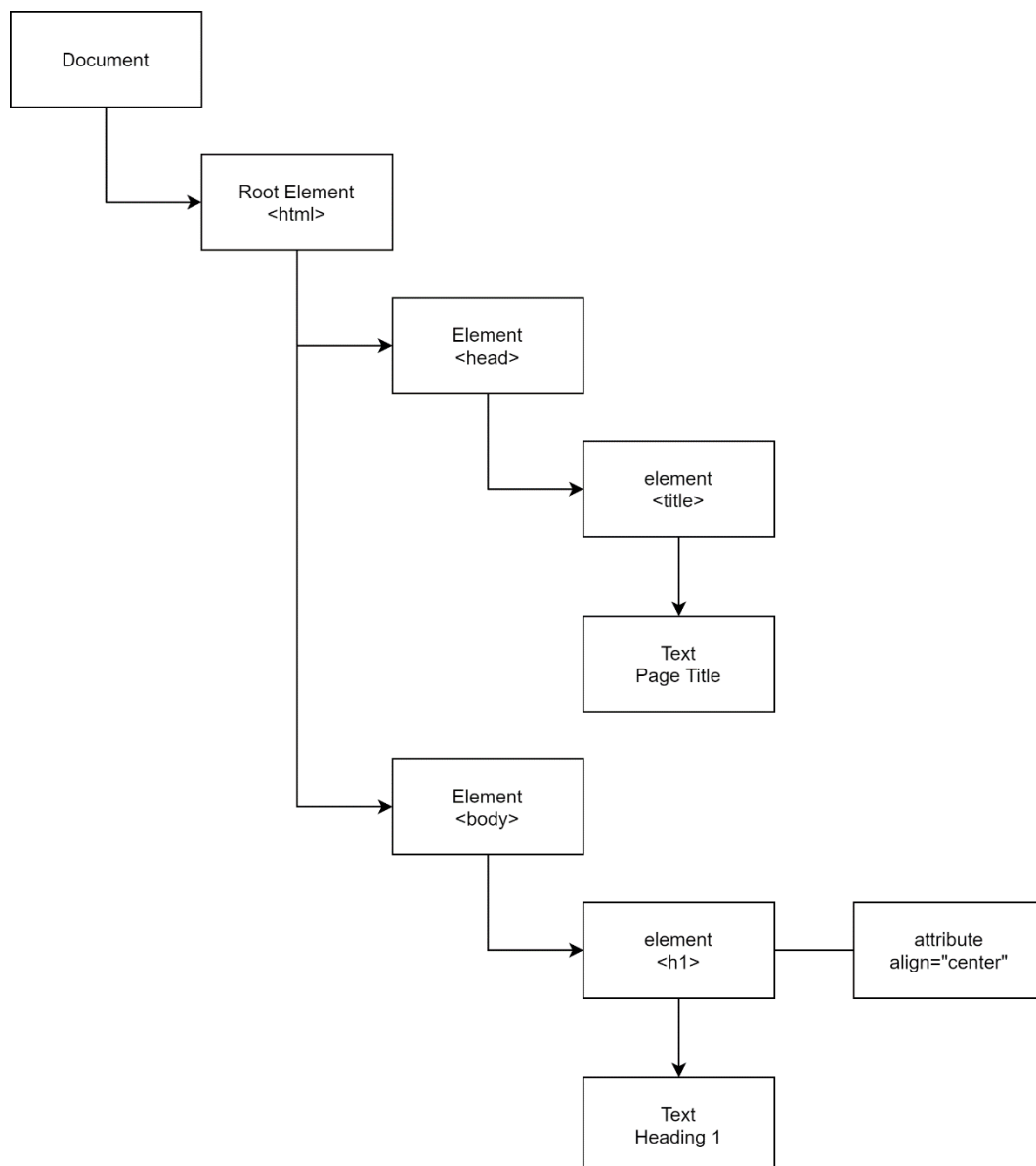
1. Penamaan file yang tersimpan dalam situs, Kamu bisa menggunakan URL sebagai penamaan file pada situs. Hal tersebut sesuai dengan makna URL sesuai penjelasan di atas.
2. Mempermudah *user* mengakses situs, Dengan URL *user* lebih mudah dalam mengakses situs dan mempermudah *user* dalam mengingat alamat dokumen yang pernah diakses.

3. Mempermudah Identifikasi *File*, Kamu dapat melakukan Identifikasi dokumen, gambar, dan *file* dalam bentuk lainnya dengan mudah.
4. Pemberian alamat dokumen, Kamu dapat mempergunakan pemberian alamat pada setiap situs yang berisikan dokumen atau *file*.

2.4. DOM

DOM atau *Document Object Model* merupakan sebuah ketentuan yang dikembangkan oleh W3C untuk berinteraksi dengan objek-objek yang ada di dalam HTML, XML, maupun XHTML. DOM bersifat *crossplatform* dan *language-independent*. DOM dapat digunakan dengan bahasa pemrograman apapun dan pada sistem operasi manapun. Pengembang bahasa pemrograman atau sistem operasi harus mengimplementasikan antarmuka DOM terlebih dahulu sebelum dapat digunakan pada aplikasi[11].

Pada model DOM ini, setiap elemen html dipandang sebagai sebuah objek. Setiap objek bisa terdiri dari objek-objek lain, sama halnya dengan dokumen html yang terdiri dari elemen *root* (elemen `<html>`), elemen *root* terdiri dari elemen `<head>` dan elemen `<body>`, elemen `<body>` boleh jadi terdiri dari elemen `<a>`, `<h1>`, `<p>`, dst. Elemen-elemen pada dokumen html membentuk sebuah object document yang merupakan object dari dokumen html itu sendiri.



Gambar 2.1 Document Object Model Tree

2.5. Interaksi Manusia dan Komputer

Interaksi manusia dan komputer merupakan disiplin ilmu yang mempelajari hubungan antara manusia dan komputer yang meliputi perancangan, evaluasi, dan implementasi antarmuka pengguna komputer agar mudah digunakan oleh manusia. Ilmu ini berusaha menemukan cara yang paling efisien untuk merancang pesan elektronik.

2.5.1. *User Interface (UI)*

Menurut Wilbert O Galitz pada buku yang berjudul *The Essential Guide to User Interface Design An Introduction to GUI Design Principles And Techniques*, *User interface* adalah bagian dari komputer dan perangkat lunak yang dapat dilihat, didengar, disentuh, diajak bicara, dan yang dapat dimengerti secara langsung oleh manusia [12]. Adapun prinsip-prinsip dasar dari *User Interface* menurut Wilbert O Galitz adalah sebagai berikut :

- a. *Aesthetically Pleasing*
- b. *Clarity*
- c. *Compability*
- d. *Comprehensibility*
- e. *Configurality*
- f. *Consistency*
- g. *Control*
- h. *Directness*
- i. *Efficiency*
- j. *Familiarity*
- k. *Flexibility*
- l. *Forgiveness*
- m. *Predictability*
- n. *Recovery*
- o. *Responsiveness*
- p. *Simplicity*
- q. *Transparency*
- r. *Trade-offs*

2.5.2. *User Experince (UX)*

User Experience memiliki banyak definisi, namun pada buku *A Project Guide to UX Design For user experience designers in the field or in the Making*, *User Experience* adalah penciptaan dan sinkronisasi elemen-elemen yang memengaruhi pengalaman pengguna dengan produk tertentu, dengan maksud

memengaruhi persepsi dan perilaku mereka [13]. Adapun factor-faktor yang dapat mempengaruhi *User Experience* adalah sebagai berikut :

- a. *Usable*
- b. *Useful*
- c. *Desirable*
- d. *Findable*
- e. *Accessible*
- f. *Credible*

2.6. Aksesibilitas Website

Aksesibilitas *web* dapat diartikan sebagai pengguna dapat melihat, memahami, menavigasi dan berinteraksi dengan situs *web*, tidak hanya pengguna normal dan pengguna dengan disabilitas, tapi juga orang dewasa yang mengalami penurunan kemampuan. Aksesibilitas situs *web* mencakup semua jenis disabilitas visual (buta, low vision, buta warna), auditori (tuli, pendengaran terganggu), motorik (ketidakmampuan menggunakan mouse, waktu respons lambat, keterbatasan kemampuan motorik lain) dan kognitif (mudah terganggu, tidak mudah untuk fokus terhadap informasi yang cukup banyak dan ketergangguan pada kemampuan belajar/mencerna informasi)[14].

2.6.1. World wide web Consortium (W3C)

World wide web Consortium (W3C) adalah suatu konsorsium yang bekerja untuk mengembangkan standar-standar untuk *World Wide Web*. Spesifikasi teknologi-teknologi utama yang dipakai sebagai basis utama situs, seperti URL (*Uniform Resource Locator*), HTTP (*Hypertext Transfer Protocol*), dan HTML (*HyperText Markup Language*) dikembangkan dan diatur oleh badan ini [15].

W3C dibuat pada 20 Oktober 1994 oleh Tim Berners-Lee, didirikan oleh *Massachusetts Institue of Tekchnology* (MIT). W3C bekerja dengan komunitas global untuk membuat standard internasional client dan server yang memungkinkan perdagangan dan komunikasi *online* melalui internet. W3C juga menghasilkan *software* acuan. W3C *Netscape Communications Corporation* adalah salah satu

anggota pendiri. Konsorsium ini dijalankan oleh MIT LCS, INIRA *Institute national the Recherche en Informatique* sebuah lembaga penelitian ilmu komputer perancis, bekerja sama dengan CERN (*Conseil Européen pour le Recherche Nucleaire*), tempat lahirnya Situs. W3C didanai oleh industri yang menjadi anggotanya, tetapi produknya tersedia gratis. Direktur W3C adalah tim Berners-Lee yang menemukan *world wide web* di CERN. W3C (*World wide web Consortium*), standar dari berbagai macam penyedia jasa untuk pembangunan dari teknologi yang berhubungan dengan Situs, seperti HTML. W3C bekerja dengan tujuan umum membuat Situs dapat diakses oleh semua user (lepas dari batasan budaya, pendidikan, keahlian, lokasi, keadaan lingkungan dan psikososial) [16].

Karena Situs begitu penting (meliputi segala aspek), tidak ada satu organisasipun yang dapat berdiri sendiri mengembangkan setiap bagian teknologi itu, W3C memberikan suatu wadah untuk bisa mengembangkan secara bersama. Perusahaan anggota utama dari organisasi itu adalah : IBM, Microsoft, America Online, Apple, Adobe Macromedia, Sun Microsystems selain anggota-anggota tersebut masih banyak lagi lembaga-lembaga baik pemerintahan maupun swasta yang turut andil dalam perkembangan W3C. Satu hal penting yang dilakukan oleh W3C adalah membangun spesifikasi pembangunan Situs (yang disebut dengan "Rekomendasi W3C") yang mana mendefinisikan protokol komunikasi (seperti HTML dan XML) serta masih banyak lagi. Dengan adanya rekomendasi baru ini maka para developer perangkat internet seperti situs browser harus menyesuaikan agar nantinya fasilitas baru itu dapat digunakan dalam browser mereka.

2.6.2. Web Content Accessibility Guidelines (WCAG)

Web Content Accessibility Guidelines atau WCAG merupakan pedoman yang digunakan untuk membuat website agar lebih mudah diakses. Dengan mengikuti pedoman ini, konten dalam website akan lebih mudah diakses oleh berbagai macam pengguna, terutama pengguna dengan disabilitas[17].

WCAG dibuat oleh World Wide Web Consortium (W3C). Dalam WCAG ada 4 prinsip yang harus dipatuhi untuk kemudahan akses dalam website, yaitu: Perceivable (Dapat dipahami), Operable (Dapat dioperasikan), Undersstandable,

dan Robust (Kokoh). Selain itu Kriteria Sukses di WCAG dibagi menjadi 3 yaitu: A (Bagus), AA (Lebih Bagus) dan AAA (Sangat Bagus) [17].

2.6.3. Prinsip Aksesibilitas

Aksesibilitas merupakan topik luas yang berkisar dari putusan hukum abstrak hingga pedoman pengkodean khusus. Pedoman standar untuk aksesibilitas pada konten situs disusun berdasarkan empat prinsip: *Perceivable*, *Operable*, *Understandable*, dan *Robust* (POUR). Untuk membantu memahami dasar-dasar aksesibilitas, penjelasan tentang POUR antara lain sebagai berikut.

1. Perceivable

Suatu situs dikatakan *perceivable* berarti pengguna dapat memproses informasi di situs tersebut, ini berarti menyediakan teks untuk mereka yang tidak bisa mendengar, dan audio untuk mereka yang tidak bisa melihat. Ini tidak berarti membuat audio untuk semua teks, tetapi konten harus dapat dibaca oleh pembaca layar dan teknologi bantuan lainnya. Situs situs dan aplikasi yang memerlukan penglihatan atau pendengaran tidak akan lulus uji *Perceivability*.

2. Operable

Operable berarti bahwa pengguna dapat menggunakan kontrol, tombol, navigasi, dan elemen interaktif lain yang diperlukan. Bagi banyak pengguna, ini berarti mengidentifikasi antarmuka secara *visual*, dan kemudian mengklik, mengetuk, atau menggeser. Untuk pengguna lain, menggunakan *keyboard* komputer atau perintah suara mungkin merupakan satu-satunya cara mereka dapat mengoperasikan dan mengontrol antarmuka.

3. Understandable

Understandable berarti pengguna harus dapat memahami atau mengerti bagaimana sebuah situs atau aplikasi bekerja. Contohnya seperti situs-situs yang menggunakan bahasa yang jelas, ringkas dan menawarkan fungsionalitas yang mudah dipahami. Jika pengguna mengambil tindakan, koneksi antara tindakan dan hasilnya harus jelas. Navigasi harus digunakan secara konsisten di seluruh situs. Formulir harus mengikuti alur logis dan memberikan label yang jelas. Jika pengguna harus melalui proses seperti *checkout*, panduan yang memadai harus

disediakan. Jika ini terasa seperti kegunaan dan bukan aksesibilitas, itu karena situs situs yang dapat digunakan secara inheren lebih mudah diakses.

4. Robust

Pengguna memilih teknologi mereka gunakan sendiri. Dalam batas tertentu, situs situs harus bekerja dengan cukup baik di seluruh *platform*, *browser*, dan perangkat untuk memperhitungkan pilihan pribadi dan kebutuhan pengguna. Salah satu cara terbaik untuk memenuhi prinsip *Robust* adalah dengan mengikuti standar yang sudah ada. *Clean Code* umumnya lebih baik dan dapat digunakan untuk lintas platform.

2.6.4. Kriteria Sukses

Setelah menjelaskan lapisan prinsip, ada juga pedoman umum aksesibilitas. Pada pedoman ini ada 12 ketentuan yang merupakan sasaran dasar yang harus dipenuhi dalam penyusunan konten, agar konten tersebut lebih mudah diakses pengguna dengan bermacam-macam disabilitas. Ketentuan-ketentuan ini tidak dapat diuji, melainkan berfungsi sebagai kerangka kerja dan tujuan menyeluruh dalam membantu penyusun konten memahami kriteria sukses dan bagaimana menerapkan teknik yang dianjurkan dengan lebih baik. Adapun 12 ketentuan tersebut meliputi :

- a. *Text Alternatives*
- b. *Time Based Media*
- c. *Adaptable*
- d. *Distinguishable*
- e. *Keyboard Accessible*
- f. *Enough Time*
- g. *Seizures Time*
- h. *Navigable*
- i. *Readable*
- j. *Predictable*
- k. *Input Assistance*
- l. *Compatible*

2.6.5. Level

Agar WCAG mudah diterapkan, setiap pedoman dilengkapi dengan level yang bisa diuji, hal ini bermanfaat ketika persyaratan dan pengujian kepatuhan wajib dilaksanakan, misalnya dalam kasus spesifikasi desain, pembelian, penyusunan aturan hukum, dan kesepakatan kontrak. Agar dapat memenuhi kebutuhan dari berbagai kalangan dan macam-macam situasi, ada tiga level kepatuhan yang dijabarkan: A (paling rendah), AA (pertengahan), dan AAA (paling tinggi).

Adapun pada penelitian ini hanya menggunakan kriteria sukses A dan AA karena menurut Dokumen WCAG tidak merekomendasikan bahwa kesesuaian Level AAA diperlukan sebagai kebijakan umum untuk seluruh situs karena tidak mungkin memenuhi semua kriteria kesuksesan Level AAA untuk beberapa konten.

2.7. *Web Accessibility Evaluation Tools*

Web Accessibility Evaluation Tools adalah program perangkat lunak atau layanan online yang membantu menemukan masalah-masalah aksesibilitas pada konten situs berdasarkan pedoman aksesibilitas. Penyedia layanan biasanya menyediakan fitur filter atau spesifikasi evaluasi untuk menemukan hasil yang sesuai dengan kebutuhan.

2.7.1. Tawdis

Tawdis adalah alat otomatis untuk menganalisis aksesibilitas situs situs. Dibuat dengan referensi teknis Pedoman Aksesibilitas Situs (WCAG 2.0) dari W3C.

Tujuan TAW adalah untuk memeriksa tingkat aksesibilitas yang dicapai dalam desain dan pengembangan halaman situs untuk akses ke semua orang terlepas dari karakteristik mereka. Layanan dari Tawdis ditujukan untuk pengguna tanpa pengalaman yang ingin mengetahui tingkat aksesibilitas situs mereka serta untuk para profesional lapangan seperti *webmaster*, Pengembang, desainer *website* dan lain-lain.

Hasil dari pengujian Tawdis memiliki 5 jenis kategori yaitu *No problem found*, *Problem found*, *Warning (A human review is recommended)* dan *Unable to perform automatic review*.

2.8. Object Oriented Anaysis Design

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelaskelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem [18].

2.8.1. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain[19].

2.8.2. Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

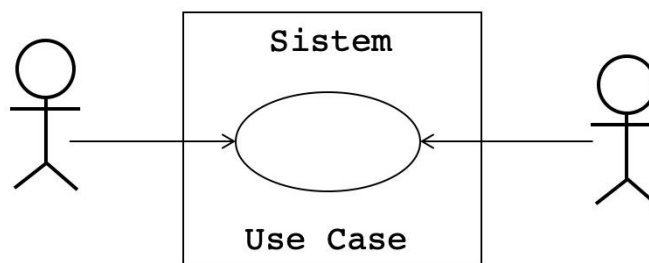
Use Case diagram digunakan untuk menggambarkan analisis kebutuhan dari sistem dari *level* atas melalui fungsionalitas dari sistem dan interaksi diantara para aktor. Aktor adalah sesuatu yang berinteraksi dengan sistem.

Secara umum, tujuan dari *use case diagram* adalah sebagai berikut [19].

1. Digunakan untuk mengumpulkan kebutuhan dari sebuah sistem
2. Untuk mendapatkan pandangan dari luar sistem
3. Untuk mengidentifikasi factor yang mempengaruhi sistem baik internal maupun eksternal

Untuk menunjukkan interaksi dari para actor dari sistem

Ilustrasi dari *actor*, *use case* dan *boundary* dapat dilihat pada Gambar 1.3



Gambar 2.2 Use Case Model

2.8.3. Activity Diagram

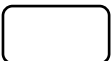
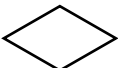
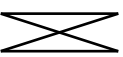
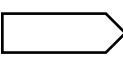
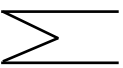
Activity diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika procedural, proses bisnis, dan aliran kerja suatu bisnis bias dengan mudah dideskripsikan dalam *activity diagram*. *Activity diagram* memiliki peran seperti halnya *flowchart*, akan tetapi perbedaannya adalah *activity diagram* bisa mendukung perilaku paralel.

Tujuan dari *activity diagram* adalah untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya. Secara umum tujuan *activity diagram* adalah sebagai berikut [19].

1. Menggambarkan aliran aktivitas dari sistem
2. Menggambarkan urutan aktifitas dari satu aktifitas ke aktifitas lainnya
3. Menggambarkan paralelisme, percabangan dan aliran konkuren dari sistem

Berikut daftar simbol-simbol dapat dilihat pada Gambar 2.5.

Tabel 2.5 Simbol - simbol yang sering dipakai pada *activity diagram*

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan (percabangan)
	Fork ; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	Rake ; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>Flow Final</i>)

2.8.4. Class Diagram

Class Diagram adalah diagram statis yang mewakili pandangan statis dari suatu aplikasi. Class diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak. Class diagram menunjukkan koleksi kelas, antarmuka, asosiasi, kolaborasi, dan constraint. Class diagram juga dikenal sebagai diagram struktural. Tujuan dari class diagram adalah untuk memodelkan pandangan statis suatu aplikasi. Secara lebih rinci tujuannya adalah sebagai berikut

1. Analisis dan desain pandangan statis aplikasi
2. Menjelaskan tanggung jawab suatu sistem
3. Basis untuk diagram komponen dan penyebaran (*deployment*)

4. *Forward and reverse engineering*

2.8.5. ***Sequence Diagram***

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case.

Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

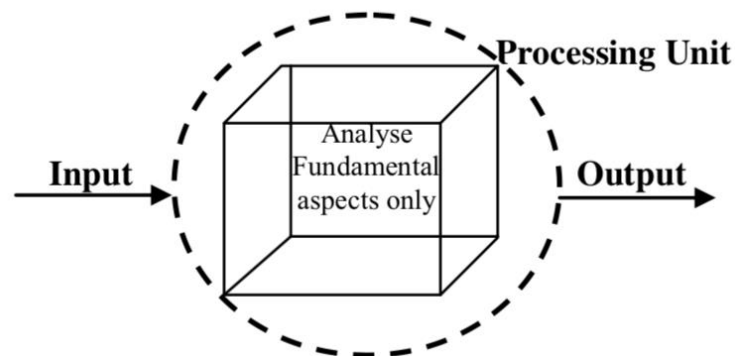
2.9. **Metode Pengujian**

Metode pengujian merupakan metode – metode yang digunakan dalam pengujian perangkat lunak yang dibangun. Pengujian dilakukan untuk memberikan evaluasi atas perangkat lunak yang dibangun, apakah telah sesuai dengan kebutuhan yang menjadi masalah atau belum sesuai.

Berikut merupakan beberapa metode pengujian yang dilakukan pada penelitian ini.

2.9.1. ***Metode Blackbox***

Metode *Black Box* adalah teknik pengujian tanpa memiliki pengetahuan tentang kerja internal dari aplikasi. Hanya mengkaji aspek fundamental dari sistem dan tidak memiliki atau relevansi kecil dengan struktur logis internal sistem. Saat melakukan uji kotak hitam, tester harus mengetahui arsitektur sistem dan tidak akan memiliki akses ke kode sumber [20].



Gambar 2.3 Ilustrasi Pengujian *Blackbox*

Metode *Black Box* memiliki keuntungan dan kekurangan diantaranya [25] :

A. Keuntungan

1. Efisien diterapkan pada segmentasi kode yang besar.
2. Persepsi yang harus dimiliki *tester* sederhana.
3. Perspektif pengguna dipisahkan dari perspektif pengembang (*programmer* dan *tester* independen satu sama lain).
4. Pengembangan kasus uji relatif cepat.

B. Kekurangan

1. Hanya sejumlah skenario yang dilakukan dan dipilih dalam pengujian. Akibatnya, cakupan pada pengujian terbatas.
2. Tanpa spesifikasi yang jelas sehingga kasus uji sulit untuk dirancang.
3. Pengujian tidak efisien.

2.9.2. *Usability Testing*

Pengujian kegunaan atau *Usability testing* merupakan sebuah tahap pengujian software yang dilakukan oleh seseorang dengan melibatkan secara langsung pengguna. Tujuan dari *usability testing* adalah mencari permasalahan kegunaan, mengumpulkan data kualitatif dan kuantitatif, serta menentukan kepuasan pengguna terhadap produk tersebut. *Usability testing* merupakan tahap yang penting untuk dilakukan sebelum sebuah produk dirilis[21].

