



Analisis *Framework*, *Library Front-End* Populer : *Bootstrap*, *Tailwind CSS*, *React*, dan *Vue* Pada Mata Kuliah Perancangan *Web Design*

Baginda Harahap¹, Aripin Rambe², Muhammad Raihan Ramadhan³, Novri Kurniawan⁴

^{1,2,3}Informatika, Fakultas Teknologi, Universitas Battuta

profesionalbaginda@gmail.com

Abstract

Bootstrap and *Tailwind CSS* focus more on styling and require additional *JavaScript* for dynamic manipulation and form validation. *Bootstrap* provides visual classes that support *HTML5* validation, while *Tailwind CSS* requires manual validation. *React* and *Vue* make form validation easier through data binding and external library support. In conclusion, *React* and *Vue* are more suitable for applications with high interactivity and dynamic needs, while *Bootstrap* and *Tailwind CSS* are more appropriate for rapid prototyping and static design. This study provides insight for developers in choosing front-end technology according to project needs. The results of the analysis show that *React* and *Vue* excel in *DOM* and event management because they use virtual *DOM* and a reactive system that facilitates the development of interactive applications.

Keywords: *Bootstrap*, *Tailwind CSS*, *React*, *Vue*, *Front-end*.

Abstrak

Bootstrap dan *Tailwind CSS* lebih fokus pada penataan gaya dan membutuhkan *JavaScript* tambahan untuk manipulasi dinamis dan validasi formulir. *Bootstrap* menyediakan kelas visual yang mendukung validasi *HTML5*, sedangkan *Tailwind CSS* membutuhkan validasi manual. *React* dan *Vue* mempermudah validasi formulir melalui pengikatan data dan dukungan pustaka eksternal. Sebagai kesimpulan, *React* dan *Vue* lebih cocok untuk aplikasi dengan interaktivitas tinggi dan kebutuhan dinamis, sedangkan *Bootstrap* dan *Tailwind CSS* lebih sesuai untuk pembuatan prototipe cepat dan desain statis. Studi ini memberikan wawasan bagi pengembang dalam memilih teknologi *front-end* sesuai dengan kebutuhan proyek. Hasil analisis menunjukkan bahwa *React* dan *Vue* unggul dalam manajemen *DOM* dan acara karena mereka menggunakan *DOM* virtual dan sistem reaktif yang memfasilitasi pengembangan aplikasi interaktif.

Kata kunci: *Bootstrap*, *Tailwind CSS*, *React*, *Vue*, *Front-end*.

1. Pendahuluan

Perkembangan teknologi *web* yang pesat telah mendorong munculnya berbagai *framework* dan *library front-end* yang memudahkan pengembangan antarmuka pengguna (UI) yang responsif, efisien, dan terstruktur. Di antara berbagai pilihan yang tersedia, *Bootstrap*, *Tailwind CSS*, *React*, dan *Vue.js* menonjol sebagai solusi populer di kalangan pengembang *web modern*. Diperkenalkan oleh *Twitter* pada tahun 2011, *Bootstrap* adalah *framework CSS open-source* yang menyediakan kumpulan komponen siap pakai seperti tombol, *form*, dan navigasi. Dengan pendekatan berbasis *grid* dan desain responsif, *Bootstrap* memungkinkan pengembang membangun antarmuka yang konsisten dan

cepat. Menurut data dari *W3Techs*, pada tahun 2023, *Bootstrap* digunakan oleh sekitar 19,2% dari seluruh situs *web* di dunia, menjadikannya salah satu *framework CSS* paling populer.

Tailwind CSS, dirilis pada tahun 2019 oleh *Tailwind Labs*, mengadopsi pendekatan *utility-first*, di mana pengembang membangun UI dengan menggabungkan kelas-kelas utilitas kecil. Pendekatan ini memberikan fleksibilitas tinggi dalam desain dan memungkinkan pembuatan antarmuka yang sangat disesuaikan. *Tailwind CSS* telah mendapatkan popularitas yang signifikan, dengan lebih dari 81.000 bintang di *GitHub* pada Agustus 2024. *React* adalah *library JavaScript* yang dikembangkan oleh *Facebook* dan dirilis pada



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

tahun 2013. *React* memungkinkan pengembangan antarmuka pengguna yang dinamis melalui konsep komponen dan virtual DOM. Menurut *survei Stack Overflow 2023*, *React* digunakan oleh sekitar 40,58% pengembang, menjadikannya *library front-end* paling populer. *Vue.js*, dirilis pada tahun 2014 oleh Evan You, adalah *framework JavaScript* progresif yang fokus pada kemudahan integrasi dan fleksibilitas.

Vue memungkinkan pengembangan antarmuka pengguna yang reaktif dengan sintaks yang sederhana dan dokumentasi yang komprehensif. Dalam survei yang sama, *Vue* digunakan oleh sekitar 16,38% pengembang, menunjukkan pertumbuhan yang stabil dalam komunitas pengembang.

Perkembangan teknologi web yang sangat pesat telah mendorong perancang antarmuka (*UI/UX Designer*) dan pengembang *web* untuk menciptakan pengalaman pengguna yang lebih interaktif, efisien, dan dinamis. Dalam konteks ini, kemampuan untuk memanipulasi elemen-elemen halaman *web* secara dinamis menjadi sangat penting. Teknik ini memungkinkan perubahan tampilan dan perilaku antarmuka dilakukan secara langsung saat pengguna berinteraksi dengan aplikasi, tanpa perlu memuat ulang halaman secara keseluruhan.

Manipulasi elemen secara dinamis memungkinkan pengembang untuk menyesuaikan konten, mengatur ulang struktur layout, hingga menambahkan animasi atau efek transisi secara *real-time* berdasarkan input pengguna. Hal ini berperan besar dalam membangun web modern yang adaptif dan responsif terhadap berbagai perangkat serta kebutuhan pengguna.

Dengan memanfaatkan bahasa pemrograman seperti *JavaScript* dan dukungan dari berbagai *framework* seperti *jQuery*, *React*, atau *Vue.js*, pengembangan web kini semakin fleksibel dan efisien. Oleh karena itu, pemahaman terhadap teori dan praktik manipulasi elemen secara dinamis menjadi kompetensi dasar yang harus dimiliki oleh setiap pengembang *web* maupun mahasiswa yang menekuni bidang teknologi informasi dan desain *web*.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan deskriptif kualitatif dengan metode komparatif-analitik. Pendekatan ini dipilih untuk mengevaluasi dan membandingkan empat *framework* dan *library front-end* populer, yaitu *Bootstrap*, *Tailwind CSS*, *React*, dan *Vue*. Tujuan dari metodologi ini adalah untuk mengkaji karakteristik teknis, kemudahan penggunaan, efisiensi pengembangan, dan performa dari masing-masing teknologi, serta memberikan gambaran yang jelas mengenai keunggulan dan kekurangannya dalam konteks pengembangan antarmuka pengguna. Selain itu,

dilakukan eksperimen terbatas dalam bentuk studi kasus teknis dengan membangun antarmuka pengguna sederhana menggunakan keempat teknologi secara terpisah.

Dari eksperimen ini, dianalisis berbagai aspek seperti kemudahan instalasi dan konfigurasi awal, jumlah kode yang diperlukan untuk mencapai fungsi yang sama, waktu pengembangan, ukuran hasil *build*, dan performa *rendering* khususnya untuk *React* dan *Vue*. Penelitian ini juga mempertimbangkan aspek subjektif dengan menganalisis ulasan dan pengalaman para pengembang yang telah menggunakan keempat teknologi tersebut dalam proyek nyata. Meskipun wawancara langsung tidak dilakukan, informasi dari forum komunitas dan *blog* teknis digunakan sebagai rujukan untuk menambah perspektif terhadap hasil analisis.

3. Hasil dan Pembahasan

a. Dasar DOM dan Event

Document Object Model (DOM) adalah representasi struktur dokumen HTML dalam bentuk pohon objek yang dapat dimanipulasi melalui *JavaScript*. Setiap *framework* dan *library* yang dibahas memiliki pendekatan berbeda terhadap interaksi dengan DOM dan penanganan *event*. *Bootstrap* dan *Tailwind CSS* sebagai *framework* berbasis CSS tidak menyediakan antarmuka langsung untuk manipulasi DOM atau *event*. Keduanya bergantung pada *JavaScript* eksternal atau integrasi dengan *library* seperti *jQuery* atau *framework* seperti *React* dan *Vue* untuk menangani interaktivitas. *React* menggunakan pendekatan virtual DOM, yang merupakan representasi ringan dari DOM nyata. Virtual DOM memungkinkan *React* untuk melakukan pembaruan antarmuka secara efisien, hanya pada bagian yang berubah. *React* menangani *event* dengan sistem *event* sintesis (*SyntheticEvent*), yang menyeragamkan perilaku *event* lintas *browser*. Penulisan *event* di *React* biasanya dilakukan langsung pada elemen JSX, seperti `onClick={handleClick}`. *Vue.js* juga menggunakan virtual DOM, tetapi dengan sintaks yang lebih sederhana. *Event* di *Vue* ditangani menggunakan direktif seperti *v-on* atau *shorthand @*, misalnya `@click="handleClick"`.

Vue mendukung binding dua arah (*two-way binding*) melalui *v-model*, yang memudahkan sinkronisasi data dan DOM. Dengan demikian, untuk kebutuhan manipulasi DOM dan *event handling*, *React* dan *Vue* memberikan pendekatan yang lebih terstruktur dan efisien dibandingkan *Bootstrap* dan *Tailwind CSS*, yang lebih difokuskan pada gaya visual.

Perbandingan penanganan *event* antara empat teknologi menunjukkan bahwa *React* dan *Vue* menyediakan struktur native untuk *event handling*, sedangkan

Bootstrap dan *Tailwind CSS* membutuhkan *JavaScript* tambahan.

Contoh event click pada masing-masing:
React:
function App() {
 const handleClick = () => { alert("Tombol diklik!");
 };

 return <button onClick={handleClick}>Klik Saya</button>;
}

Vue:
<template>
<button @click="handleClick">Klik Saya</button>
</template>
<script>

export default { methods: { handleClick() {
 alert("Tombol diklik!");
 }
 }
 }
</script>

Bootstrap (dengan JavaScript):
<button id="btn">Klik Saya</button>
<script>
 document.getElementById("btn").addEventListener("click", function () {

b. Manipulasi Elemen Secara Dinamis

Manipulasi elemen secara dinamis merujuk pada kemampuan untuk menambahkan, menghapus, atau mengubah elemen HTML berdasarkan interaksi pengguna atau perubahan data. React memanfaatkan *state* dan *props* untuk mengatur perubahan elemen secara deklaratif.

Dengan menggunakan *useState* atau *setState*, komponen dapat memperbarui tampilannya secara otomatis ketika data berubah. Misalnya, menampilkan daftar item dari *array* dapat dilakukan dengan metode *map()* yang merender ulang setiap kali *array* diperbarui. *Vue* memiliki konsep reaktif yang serupa, dengan penggunaan data, methods, dan komputer. Ketika data dalam *data()* berubah, DOM yang terkait secara otomatis diperbarui. *Vue* juga menawarkan sistem *directive* seperti *v-for* untuk iterasi dinamis, dan *v-if* untuk kontrol kondisi. *Bootstrap* dan *Tailwind CSS* sendiri tidak memiliki dukungan native terhadap manipulasi dinamis. *Bootstrap*, jika digabungkan dengan *jQuery*, memungkinkan manipulasi DOM secara *imperative*. *Tailwind*, yang berbasis *utility class*, tetap membutuhkan

JavaScript eksternal atau integrasi dengan *framework* lain untuk menampilkan perubahan secara dinamis.

Oleh karena itu, manipulasi elemen secara dinamis paling efisien dan terkontrol dilakukan menggunakan *React* atau *Vue*, karena mereka mengintegrasikan logika data. Teori Manipulasi Elemen Secara Dinamis merujuk pada prinsip-prinsip dan teknik untuk mengubah, menambah, menghapus, atau memperbarui elemen-elemen dalam antarmuka (UI) atau struktur data secara langsung saat program berjalan (*runtime*), tanpa perlu memuat ulang halaman atau sistem. Manipulasi Elemen Secara Dinamis mengacu pada proses mengubah atau mengatur elemen-elemen dalam suatu sistem (terutama antarmuka pengguna atau struktur data) secara langsung saat program sedang berjalan (*runtime*), tanpa harus *reload* halaman atau aplikasi. Dalam konteks pengembangan perangkat lunak dan web, manipulasi elemen secara dinamis.

Contoh skrip untuk menampilkan/menyembunyikan Konten

```
<button onclick="tampilSembunyi()">Tampil / Sembunyi</button>  
<div id="kotak" style="display:none; background:lightblue; padding:10px;">  
  Ini konten dinamis.  
</div>  
<script>  
  function tampilSembunyi() {  
    let kotak = document.getElementById("kotak");  
    kotak.style.display = kotak.style.display ===  
    "none" ? "block" : "none";  
  }  
</script>
```

c. Validasi Form Sederhana

Validasi form adalah *fitur* penting dalam antarmuka pengguna untuk memastikan data yang dimasukkan sesuai dengan aturan tertentu sebelum diproses lebih lanjut. *Bootstrap* menyediakan komponen validasi berbasis *CSS* yang memungkinkan pemberian umpan balik visual melalui kelas seperti *is-invalid* atau *is-valid*. Namun, logika validasi tetap ditulis menggunakan *JavaScript* eksternal. *Bootstrap 5* memiliki contoh penggunaan validasi dengan HTML5 built-in seperti *required*, tetapi untuk kebutuhan validasi kompleks, perlu integrasi tambahan. *Tailwind CSS* tidak menyertakan komponen validasi. Namun, dengan *utility classes* seperti *border-red-500* atau *text-sm text-red-600*, pengembang dapat membangun umpan balik visual secara manual. Validasi tetap memerlukan *JavaScript* murni atau *framework* lain untuk logika dan pengendalian perilaku *form*. *React* memungkinkan validasi *form* secara deklaratif dengan pengelolaan *state*.

Validasi dapat diintegrasikan secara langsung dalam komponen dengan pengecekan nilai input dan pengaturan pesan kesalahan.

Validasi form adalah proses untuk memastikan bahwa data yang dimasukkan pengguna ke dalam formulir telah sesuai dengan aturan atau kriteria tertentu sebelum dikirim ke server.

- Validasi sederhana biasanya mencakup:
- Mengecek apakah input kosong.
- Memastikan format email benar.
- Memastikan panjang kata sandi mencukupi.
- Validasi angka (misal: usia harus angka dan > 0).

Banyak juga *library* validasi untuk *React* seperti *Formik* atau *React Hook Form* yang mempermudah proses validasi. *Vue* juga menyediakan validasi *form* dengan pendekatan deklaratif. Dengan binding *v-model* dan direktif seperti *v-if*, validasi dapat dilakukan secara reaktif berdasarkan input pengguna. *Vue* juga memiliki plugin seperti *Vuelidate* atau *vee-validate* untuk validasi *form* yang lebih kompleks dan terstruktur. Secara keseluruhan, untuk validasi *form* sederhana hingga kompleks, *React* dan *Vue* menawarkan fleksibilitas dan efisiensi tinggi, sementara *Bootstrap* dan *Tailwind* hanya berperan sebagai pendukung tampilan visual validasi. Validasi *form* paling mudah dan reaktif dilakukan di *React* dan *Vue*. *Bootstrap* memberi gaya siap pakai, sedangkan *Tailwind* memerlukan pemodelan ulang.

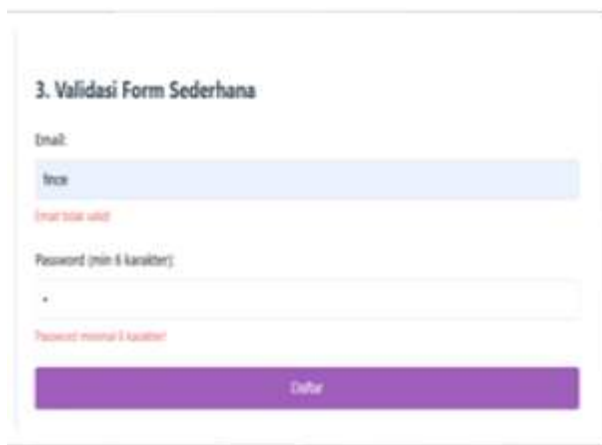
React – validasi input kosong:

```
function Form() {
  const [name, setName] = React.useState("");
  const handleSubmit = (e) => {
    e.preventDefault();
    if (!name) alert("Nama harus diisi!");
  };
  return (
    <form onSubmit={handleSubmit}>
      <input value={name} onChange={(e) =>
        setName(e.target.value)} />
      <button type="submit">Submit</button>
    </form>
  );
}
```

Vue – validasi input kosong:

```
<template>
  <form @submit.prevent="validate">
    <input v-model="name" />
    <button type="submit">Submit</button>
  </form>
</template>
<script>
  export default { data() {
    return { name: "" };
  },
```

```
  methods: { validate() {
    if (!this.name) alert("Nama harus diisi!");
  }
}
</script>
Bootstrap – validasi visual (HTML5):
<form>
  <input type="text" class="form-control" required
  />
  <div class="invalid-feedback">Nama wajib
  diisi.</div>
  <button class="btn btn-primary">Submit</button>
</form>
Tailwind CSS – validasi manual:
<form onSubmit="return validateForm()">
  <input type="text" id="name" class="border p-2"
  />
  <p id="error" class="text-red-500 text-sm
  hidden">Nama wajib diisi.</p>
  <button class="bg-blue-500 text-white px-4 py-
  2">Submit</button>
</form>
<script>
  function validateForm() {
    const name =
    document.getElementById("name").value; const error
    = document.getElementById("error");
    if (!name) { error.classList.remove("hidden");
    return false;
    }
    return true;
  }
</script>
```



Gambar 1. Hasil Tag dari Form Sederhana

Aspek	Bootstrap	Tailwind CSS	React	Vue
DOM & Event	Mengandalkan JavaScript/jQuery eksternal	Tidak menangani DOM secara langsung	Menggunakan Virtual DOM dan Synthetic Event	Virtual DOM dengan binding event via @
Manipulasi Dinamis	Menggunakan JavaScript manual	Memerlukan JavaScript manual	Declarative, berbasis state dan props	Reactive, berbasis data() dan methods
Validasi Form	Dukung validasi HTML5 & styling visual	Memerlukan validasi manual & styling	Validasi mudah via state atau library	Validasi mudah via v-model atau plugin
Kemudahan Penggunaan	Sangat mudah untuk pemula	Mudah, fleksibel, butuh pemahaman lanjut	Perlu belajar JSX dan konsep komponen	Lebih ringan & mudah dari React untuk pemula
Library Tambahan	Perlu JS eksternal untuk interaksi	Perlu framework JS tambahan untuk logika	Bisa berdiri sendiri (SPA) atau hybrid	Bisa standalone atau integrasi fleksibel

Gambar 2. Perbandingan Berdasarkan DOM, Manipulasi Dinamis, dan Validasi

Sebaliknya, *Bootstrap* dan *Tailwind CSS* tidak menyediakan kemampuan langsung untuk pengelolaan *event* atau DOM dan lebih bergantung pada *JavaScript* tambahan, seperti *jQuery* atau *native JavaScript*, untuk menciptakan interaktivitas. Dalam hal manipulasi elemen secara dinamis, *React* dan *Vue* menawarkan pendekatan yang jauh lebih efisien dan terstruktur. Melalui sistem reaktivitas dan manajemen *state*, kedua *framework* ini memungkinkan pembaruan antarmuka secara otomatis saat data berubah. Sementara itu, *Bootstrap* dan *Tailwind CSS*, yang lebih fokus pada styling, memerlukan pendekatan imperatif dan penulisan kode *JavaScript* tambahan untuk menciptakan perubahan dinamis di dalam antarmuka.

Untuk validasi *form*, *React* dan *Vue* juga menunjukkan keunggulan melalui pendekatan deklaratif dan dukungan terhadap berbagai *library* pihak ketiga yang memperkaya fungsionalitas validasi. *Bootstrap* memang menyediakan gaya visual untuk validasi dan kompatibel dengan fitur *HTML5*, namun tidak menyediakan sistem logika validasi tersendiri. *Tailwind CSS*, meskipun sangat fleksibel dalam pengaturan tampilan, tetap membutuhkan logika validasi yang ditulis secara manual.

Keseluruhan, dapat disimpulkan bahwa *React* dan *Vue* merupakan pilihan terbaik untuk membangun antarmuka pengguna yang dinamis, kompleks, dan membutuhkan interaktivitas tinggi. Sementara itu, *Bootstrap* dan *Tailwind CSS* sangat cocok digunakan dalam pengembangan cepat atau untuk kebutuhan antarmuka yang lebih statis dan sederhana. Pemilihan teknologi sebaiknya disesuaikan dengan kebutuhan proyek dan tingkat kompleksitas yang dihadapi. Penelitian ini diharapkan dapat menjadi referensi bagi pengembang, mahasiswa, maupun akademisi dalam menentukan pilihan teknologi yang tepat dalam pengembangan *front-end* aplikasi web.

4. Kesimpulan

Penelitian ini telah membahas dan menganalisis empat teknologi *front-end* populer :

1. *Bootstrap*, *Tailwind CSS*, *React*, dan *Vue*, dengan fokus pada tiga aspek utama: dasar DOM dan *event*, manipulasi elemen secara dinamis, serta validasi *form* sederhana. Berdasarkan pendekatan deskriptif kualitatif yang digunakan, penelitian ini menggabungkan studi literatur dari berbagai sumber terpercaya dan eksperimen terbatas dalam bentuk implementasi kode sebagai studi kasus.
2. Dari hasil analisis, terlihat bahwa *React* dan *Vue* memiliki keunggulan yang signifikan dalam hal pengelolaan DOM dan *event*. Keduanya menggunakan pendekatan virtual DOM yang efisien dan menawarkan cara kerja deklaratif yang mempermudah pengembangan antarmuka dinamis. *Event* di *React* ditangani melalui sistem *Synthetic Event*, sedangkan *Vue* menggunakan binding langsung melalui direktif seperti *@click*.
3. Validasi form sederhana adalah teknik penting untuk menjaga kualitas input pengguna. Meskipun bisa dilakukan dengan *HTML5*, validasi berbasis *JavaScript* memberikan fleksibilitas lebih untuk kontrol dan interaktivitas. Validasi yang baik tidak hanya memastikan keamanan data, tetapi juga meningkatkan kepuasan pengguna dalam mengisi formulir.

Daftar Rujukan

- [1] Al-Khattat, S. H. K., Habeeb, R. R., & Mohammed, A. R. (2019). An ASSURE-Model Instructional Design Based on Active Learning Strategies and its Effect for 1st Intermediate Student's Higher Order Thinking Skills in Teaching Science Text Book. *Psihologija*, 52(5).
- [2] Al Huda, S., Fatirul, A. N., & Hartono, H. H. (2022). Development of assure based digital learning modules in English subject at SMP Islam Al Azhar 13 Surabaya. *Intensive Journal*, 5(2), 92–103.
- [3] Analisis Teknik Pengambilan Vidio Cinematic dan Proses Editing Menggunakan Filmora Pada Organisasi Pramuka SMAN 14 Medan. (n.d.).
- [4] Apriono, D., Wardhono, A., & Ahtia, D. (n.d.). *Validity of Google Sites-Based Learning Media to Increase Students' Learning Motivation in Class 5 PPKN Lessons In Elementary Schools*.
- [5] Asrowi, H. A. R. (2018). PAUD Students Based on Prezi Media Using Assure Model. *PROCEEDING ICTESS (Internasional Conference on Technology, Education and Social Sciences)*.
- [6] Baginda, B. H., Singarimbun, R. N., & Hasibuan, E. H. (2022). Penggunaan Sosial Media Interaktif Di Masa Pandemi Guna Untuk Meningkatkan Kreativitas Siswa SMK Halongonan Kelas X. *Mejujua: Jurnal Pengabdian Pada*

- Masyarakat*, 1(3), 1–6.
- [7] Ellis, J. T. P. (2009). *Assessing the development of high school chemistry students' conceptual and visual understanding of dimensional analysis via supplemental use of a proprietary interactive software program*. Louisiana State University and Agricultural & Mechanical College.
- [8] Fitria, R. D., & Umam, M. B. (2025). Analysis of Environmental Literacy in Students of the Social Studies Study Program UIN Kiai Haji Achmad Siddiq Jember: Analisis Literasi Lingkungan Pada Mahasiswa Program Studi Tadris IPS UIN Kiai Haji Achmad Siddiq Jember. *Chalim Journal of Teaching and Learning*, 5(1), 18–27.
- [9] Gasmil, N., Afifah, U., Kasturi, R., Ananda, R., & Tadjuddin, N. (2025). Implementation of the Quizizz-Based ASSURE Learning Model in Islamic Religious Education. *Journal of Islamic Education Research*, 6(2), 163–178.
- [10] Harahap, B. (2022). The Effect of Image Media on Learning Outcomes of Class V MIS Students in Siboruangin Village in Information Communication Technology (ICT) Subjects. *Jurnal Informatika*, 1(1), 12–19.
- [11] Illah, R., & Susetyarini, E. (2024). A systematic literature review on the development of STEM-based module. *Research and Development in Education (RaDEn)*, 4(1), 582–591.
- [12] Jamain, R. R., Putro, H. Y. S., Hairunisa, H., & Ariliani, T. (2023). Storybook as a Preventive Measure Against the "Sexual Harassment" of Children in the Elementary School. *International Journal of Asian Education*, 4(4), 247–255.
- [13] Marthidayah, N. B. (n.d.). *Pengembangan Multimedia Life-Like pada Pokok Bahasan Kingdom Plantae untuk Media Pembelajaran Biologi SMA*.
- [14] Muhamad, R., & Surat, S. (2024). KNOWLEDGE, ATTITUDES, AND PRACTICES OF SOCIALEMOTIONAL LEARNING (SEL) AMONG SCHOOL ADMINISTRATORS. *Special Education [SE]*, 2(1), e0030--e0030.
- [15] Nasution, A. A., Harahap, B., Harahap, R. A., Wahdi, N., & others. (2022). Socialization of the Use of Multimedia as a Learning Tool to Improve the Skills of MAS Darul Ilmi Students. *IJCS: International Journal of Community Service*, 1(1), 48–61.
- [16] Nurhalisa, N., Rizal, R., Aqil, M., Lagandesa, Y. R., & Fasli, M. (2025). Pengaruh Model Problem Based Learning (PBL) dengan berbantuan Media Wordwall terhadap Hasil Belajar Siswa pada Mata Pelajaran Bahasa Indonesia. *Attadrib: Jurnal Pendidikan Guru Madrasah Ibtidaiyah*, 8(1), 151–159.
- [17] Prastawa, S. (2025). Determinant of Success English Reading Skills Among University Students: Remedial and Challenge Strategies: Faktor Penentu Keberhasilan Kemampuan Membaca Bahasa Inggris di Kalangan Mahasiswa: Strategi Remedial dan Tantangan. *Chalim Journal of Teaching and Learning*, 5(1), 1–10.
- [18] Rahmah Nur. 2016. Pengembangan Media Pembelajaran Interaktif Berbasis *Assure* Pokok Bahasan Virus Kelas X Sma Negeri 11 Makassar. Makassar. Fakultas Tarbiyah Dan Keguruan Universitas Islam Negeri (UIN) Alauddin Makassar.
- [19] Rahmandhani, H. N., & Utami, E. (2022). Comparative analysis of ADDIE and ASSURE models in designing learning media applications. *Jurnal Educative: Journal of Educational Studies*, 7(2), 123–138.
- [20] Rofiq, M. H., Shah, A. A., Alai, A., & Istiqomah, I. (2025). Enhancing Learning Motivation in Islamic Religious Education: The Role of Sahabat Anak Muslim Comics among Elementary School Students. *Attadrib: Jurnal Pendidikan Guru Madrasah Ibtidaiyah*, 8(1), 255–268.
- [21] Tarsono, T., Nurlaila, W. N., Najiha, I., Ningsih, N. W., Rifqi, A., Agustin, U., & Tauviqillah, A. (2024). Assure-Based Multimedia Learning Design in Islamic Education Module at Hikmah Teladan Junior High School. *Tadrib: Jurnal Pendidikan Agama Islam*, 10(1), 735–741.
- [22] Utami, E. (2018). Hasna Nirfya Rahmandhani. *Learning*, 7, 212–221.
- [23] Zulkifli, Z. A., Sharip, A. A., Zain, S. M. M., Rashid, N., Saidi, R. M., Rashid, N. A. M., & Geigiana, A. (2022). Integration of Instructional Models and Learning Styles for Open and Distance Learning Environment. *World Journal of English Language*, 12(2), 226–238.