

# Rancang Bangun Web Optimasi Promosi Bimbingan Belajar Menggunakan Next.JS Dengan Pendekatan Agile

Irfan Rahmat Firmansyah<sup>1</sup>, I Gde Agung Sri Sidhimantra

Manajemen Informatika, Fakultas Vokasi, Universitas Negeri Surabaya  
Jl. Ketintang Gang II, Kec. Wonokromo, Kota Surabaya, Jawa Timur 60231

<sup>1</sup>irfan.21063@mhs.unesa.ac.id

<sup>2</sup>igdesidhimantra@unesa.ac.id

**Abstrak**— Keterbatasan media promosi digital pada platform sederhana menjadi tantangan utama dalam meningkatkan efektivitas promosi bimbingan belajar. Penelitian ini bertujuan merancang dan mengembangkan website promosi untuk cabang Bimbingan Belajar Master Prima di SMA Trimurti, Surabaya, melalui platform yang modern dan ramah pengguna. Proses penelitian diawali dengan wawancara dan observasi untuk mengidentifikasi kebutuhan promosi. Website dikembangkan menggunakan metodologi Agile dengan teknologi *Next.js* sebagai *frontend* dan *Supabase* sebagai *backend*. Sistem diuji melalui pengujian fungsionalitas dan pengujian penerimaan pengguna dengan melibatkan 27 responden. Hasil pengujian menunjukkan tingkat keberhasilan fungsi 100% dan skor kepuasan pengguna 4,6 dari 5. Penelitian ini menyimpulkan bahwa penerapan *Next.js* dan *Supabase* melalui pendekatan Agile efektif meningkatkan visibilitas digital, mempercepat iterasi pengembangan, dan mendukung keberlanjutan promosi pendidikan Master Prima.

**Kata kunci**—Agile, Next.js, Supabase, Promosi Digital, Website.

**Abstract**— Limitations in digital promotional media on simple platforms represent a primary challenge in enhancing the promotional effectiveness of tutoring services. The objective of this study is to design and build a promotional website for the Master Prima Tutoring branch at SMA Trimurti, Surabaya, utilizing a modern and user-friendly platform. The research process commenced with interviews and observations to identify promotional requirements. The website was developed employing Agile methodology with *Next.js* technology as the *frontend* and *Supabase* as the *backend*. System validation was conducted through functional testing and user acceptance evaluation involving 27 respondents. Testing outcomes revealed a 100% functional success rate and a user satisfaction score of 4.6 out of 5. This study concludes that implementing *Next.js* and *Supabase* through the Agile approach effectively enhances digital visibility, accelerates development iterations, and supports the sustainability of Master Prima's educational promotion.

**Keywords**—Agile, Next.js, Supabase, Digital Promotion, Website.

## I. PENDAHULUAN

Kemajuan teknologi digital telah merevolusi cara lembaga pendidikan, termasuk bimbabel, memasarkan layanannya. Pembelajaran daring membuat materi pelajaran dari berbagai penjuru dunia bisa diakses siapa saja dengan mudah, sehingga cakupan peserta didik menjadi lebih luas dan merata[1]. Website kini menjadi fasilitas teknologi informasi vital yang berfungsi sebagai media pemasaran dan komunikasi untuk membangun hubungan erat dengan pelanggan[2]. Selain itu, optimalisasi pemasaran digital melalui website dapat meningkatkan kuantitas dan kualitas trafik secara signifikan melalui mesin pencari[3].

Media sosial pun berperan penting sebagai sarana promosi yang ampuh karena memudahkan komunikasi dua arah dengan calon siswa sekaligus menjangkau khalayak lebih luas[4]. Namun di sisi lain, metode promosi konvensional seperti penyebaran brosur atau pemasangan spanduk mulai dinilai kurang efektif. Metode ini memiliki keterbatasan dalam hal jangkauan audiens dan risiko penyampaian informasi yang kurang mendetail, menjadikannya kurang relevan bagi target audiens yang menuntut akses informasi cepat dan transparan[5].

Sebagai solusi, penelitian ini mengembangkan website promosi mandiri menggunakan metodologi Agile. Agile dipilih karena menawarkan fleksibilitas yang lebih baik dibandingkan metode terstruktur seperti Waterfall, terutama dalam mengakomodasi perubahan kebutuhan pengguna selama pengembangan[6]. Pengembangan ini memanfaatkan *Next.js* untuk antarmuka yang responsif dan *Supabase* sebagai *backend* andal[7]. Pendekatan ini memastikan pengembangan berlangsung iteratif dan efisien melalui pembagian proyek dalam *sprint* kecil yang dievaluasi berkala[8].

## II. KAJIAN PUSTAKA

### A. Manajemen Proyek dengan Scrum

Penerapan kerangka kerja *Scrum* dalam metodologi Agile terbukti mampu meningkatkan produktivitas tim pengembang perangkat lunak. Penelitian [9] menunjukkan bahwa *Scrum* berdampak positif pada efisiensi kerja, komunikasi tim, dan kecepatan peluncuran produk. Selain itu, metode ini sangat relevan untuk tim yang heterogen maupun pengembang individu karena fleksibilitasnya dalam menghadapi perubahan dinamis selama siklus proyek[10]. Dalam konteks pengembangan mandiri, teknik *Solo Software Development Methodology* dapat diterapkan untuk menjaga kualitas dan keamanan kode tanpa mengorbankan kelincahan[11].

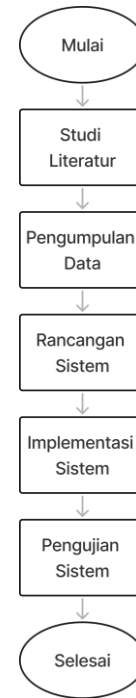
### B. Pengembangan Sistem Berbasis Web

Sistem informasi berbasis web yang dibangun dengan teknologi modern seperti *Next.js* mampu meningkatkan efisiensi manajemen data dan partisipasi pengguna. menegaskan bahwa penggunaan framework ini sangat efektif untuk mendigitalkan proses manajemen organisasi[12]. Hal ini sejalan dengan temuan[13], yang menyatakan bahwa sistem informasi manajemen berbasis web dapat memperbaiki pengambilan keputusan dan akses informasi pelanggan secara signifikan dibandingkan sistem manual.

### C. Pengujian Perangkat Lunak

Untuk menjamin keberhasilan implementasi, pengujian sistem mutlak diperlukan. Pengujian *User Acceptance Testing (UAT)* menjadi metode krusial untuk memvalidasi kualitas produk dan memastikan kepuasan pengguna akhir terhadap fitur yang ditawarkan[14]. Aplikasi yang telah melalui tahap *UAT* dengan hasil positif menunjukkan tingkat penerimaan yang baik dan kemudahan penggunaan yang tinggi[15]. Selain itu, evaluasi melalui metode *Black-box* juga penting untuk memastikan aplikasi berjalan efektif dan *user-friendly* sebelum diluncurkan ke publik [16].

## III. METODE PENELITIAN



Gambar 1. Alur Penelitian

Alur penelitian yang ditampilkan pada gambar di atas menggambarkan tahapan-tahapan yang digunakan dalam proses perancangan website. Penjelasan dari alur tersebut disajikan sebagai berikut:

- *Identifikasi Masalah*  
Tahap identifikasi masalah dilakukan melalui pengamatan langsung dan wawancara mendalam dengan pemilik serta tim "Master Prima" untuk memahami kebutuhan dan tantangan yang dihadapi. Metode ini bertujuan mengumpulkan data akurat guna merancang solusi yang tepat dan relevan dengan kebutuhan yang ada.
- *Studi Literatur*  
Kajian literatur dilakukan untuk membangun landasan teoretis penelitian melalui pencarian dan pengumpulan berbagai referensi yang relevan dengan topik penelitian, seperti jurnal ilmiah, artikel, dan tesis, yang kemudian dijadikan dasar dalam merumuskan strategi pelaksanaan penelitian..
- *Pengumpulan Data*  
Pengumpulan data dilaksanakan untuk memenuhi kebutuhan dalam proses pengembangan sistem. Pada tahap ini, peneliti menerapkan dua metode pengumpulan data, yaitu:

1. Observasi

Pengamatan langsung dilakukan terhadap aktivitas operasional di cabang SMA Trimurti Surabaya untuk memahami alur kerja promosi yang sedang berjalan dan mengidentifikasi kebutuhan spesifik yang belum terpenuhi oleh media promosi yang tersedia saat ini.

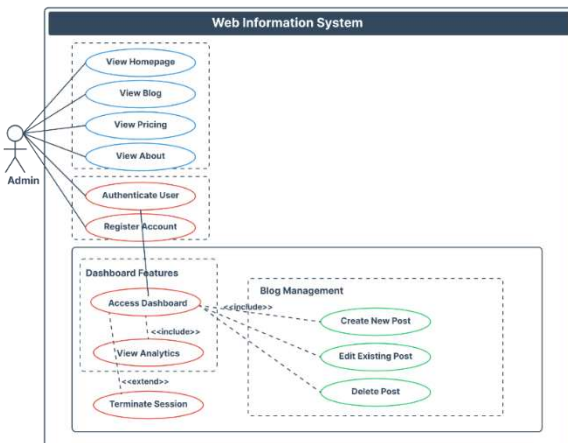
2. Wawancara

Dilakukan melalui sesi tanya jawab langsung dengan tim pemasaran dan pemilik Bimbingan Belajar "Master Prima". Hasil wawancara mengidentifikasi kendala utama, yaitu penggunaan *platform Blogspot* yang dinilai kurang profesional, sulit dikelola, dan memiliki tampilan yang tidak rapi, sehingga menghambat efektivitas promosi cabang Surabaya.

• Rancangan Sistem

Perancangan sistem dilakukan menggunakan *Unified Modeling Language (UML)* guna memvisualisasikan struktur dan perilaku sistem sebelum memasuki tahap implementasi.

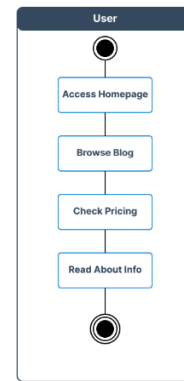
1. Diagram Use Case



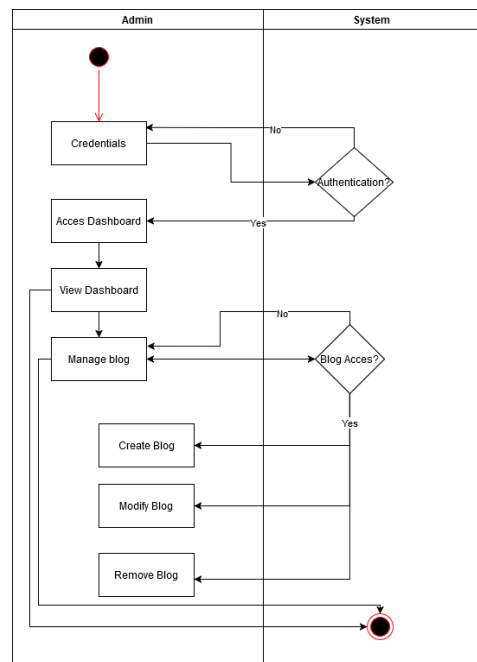
Gambar 2. Diagram Use Case

Sistem dirancang melibatkan dua aktor utama, yaitu *Admin* dan *User* (Pengguna Umum). *User* memiliki akses ke fitur publik seperti melihat beranda, membaca blog, dan mengecek harga paket. Sementara itu, *Admin* memiliki hak akses khusus untuk melakukan autentikasi, mengakses *dashboard*, serta mengelola konten blog (membuat, mengedit, dan menghapus artikel).

2. Activity Diagram



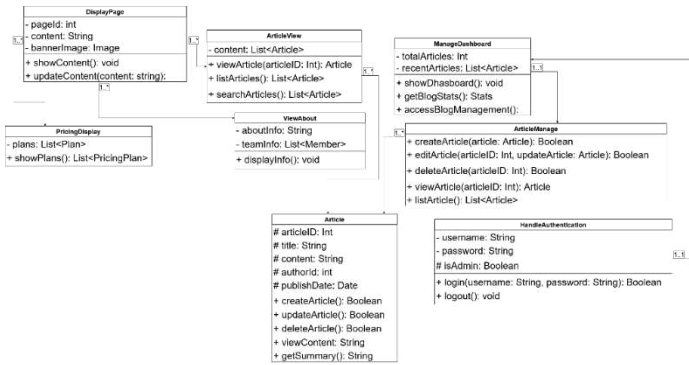
Gambar 3. Activity Diagram User



Gambar 4. Diagram Aktivitas Admin

Alur aktivitas sistem dipetakan secara mendetail. Untuk *User*, aktivitas dimulai dari mengakses beranda hingga mencari informasi layanan. Untuk *Admin*, alur dimulai dengan validasi kredensial pada halaman *login*; jika berhasil, *Admin* diarahkan ke *dashboard* untuk melakukan manajemen data sistem.

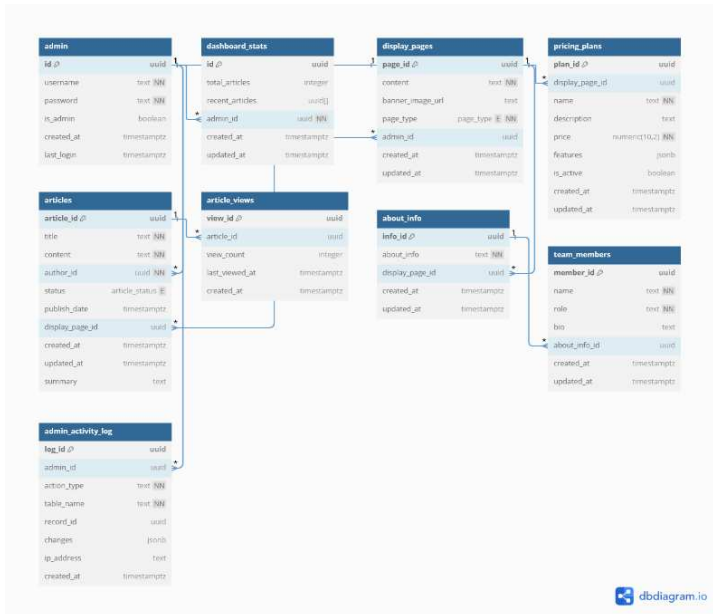
### 3. Diagram kelas



Gambar 5. Class Diagram

Struktur sistem dibangun atas beberapa kelas utama yang saling berinteraksi. Kelas *DisplayPage* dan *ArticleView* menangani tampilan konten kepada pengguna, sedangkan kelas *ManageDashboard* dan *ArticleManage* berfungsi sebagai pengendali logika di sisi admin. Keamanan akses diatur oleh kelas *HandleAuthentication*.

### 4. Skema Database



Gambar 6. Skema Database

Penyimpanan data dirancang menggunakan skema relasional yang komprehensif. Tabel utama meliputi *admin* untuk data pengguna otoritatif, *articles* dan *article\_views* untuk manajemen konten blog, serta *pricing\_plans* untuk menyimpan informasi paket layanan. Integritas data dijaga melalui penggunaan

*Foreign Key* dan tipe data yang relevan seperti *UUID* dan *Timestamp*.

#### • Implementasi Sistem

Implementasi website "Master Prima" menggunakan Next.js sebagai frontend untuk menghasilkan antarmuka responsif dan modern, serta Supabase sebagai backend untuk mengelola basis data secara andal. Metodologi Agile diterapkan dalam pengembangan untuk memastikan website adaptif dan fleksibel sesuai kebutuhan pengguna melalui proses iteratif yang memungkinkan perbaikan berkelanjutan di setiap tahap pengembangan.

#### • Pengujian Sistem

Pada tahap pengujian sistem, peneliti menerapkan dua metode untuk memastikan kualitas dan keandalan aplikasi web. Metode pertama adalah *User Acceptance Testing (UAT)* melalui *Google Form* yang disebarluaskan kepada mahasiswa untuk memvalidasi kesesuaian sistem dengan kebutuhan pengguna dan mendapatkan umpan balik mengenai kemudahan navigasi, autentikasi, pencarian, dan aksesibilitas blog. Metode kedua adalah *black-box testing* melalui *Google Form* yang ditujukan kepada pemangku kepentingan untuk mengevaluasi fungsionalitas sistem sesuai spesifikasi yang ditetapkan. Hasil evaluasi dari kedua metode dianalisis untuk mengidentifikasi area perbaikan, mengoptimalkan kinerja sistem, dan memastikan aplikasi web memenuhi standar kualitas serta kebutuhan pengguna secara efektif dan efisien.

## IV. HASIL DAN PEMBAHASAN

### A. Tahap Persiapan Implementasi Sistem

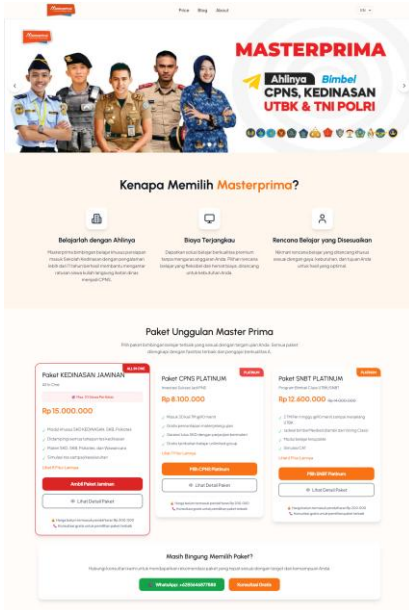
Tahap awal implementasi diawali dengan persiapan lingkungan pengembangan yang memadai untuk mendukung performa aplikasi optimal. Spesifikasi perangkat keras yang digunakan dalam penelitian ini meliputi prosesor Intel Core i7-13650HX, memori RAM 12 GB DDR5, dan penyimpanan SSD 512 GB untuk menjamin kecepatan komputasi yang tinggi. Pada aspek perangkat lunak, sistem operasi Windows 11 digunakan bersama runtime Node.js v20.x dan kerangka kerja Next.js v14.x. Pengelolaan kode dilakukan menggunakan Visual Studio Code sebagai editor dan Git untuk kontrol versi. Kombinasi spesifikasi ini dipilih untuk memastikan kompatibilitas teknologi dan efisiensi selama proses pengembangan hingga tahap deployment.

### B. Implementasi Antarmuka Pengguna

Desain antarmuka dikembangkan menggunakan kerangka kerja *Next.js* dengan dukungan *Tailwind CSS* untuk memastikan tampilan yang modern, responsif, dan

ramah pengguna. Implementasi ini menghasilkan beberapa halaman utama yang strategis:

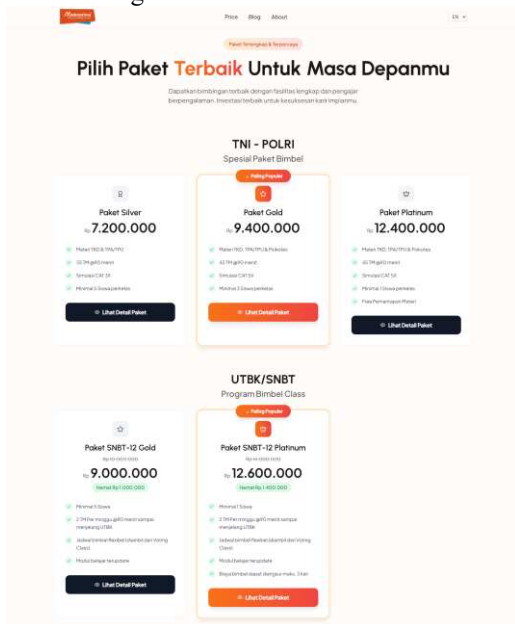
- Halaman Utama



Gambar 7. Halaman Utama

Halaman ini berfungsi sebagai gerbang informasi utama yang menampilkan elemen visual seperti banner promosi, keunggulan lembaga, dan testimoni untuk memberikan gambaran menyeluruh mengenai layanan yang ditawarkan.

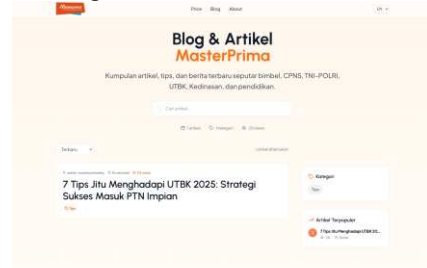
- Halaman Harga



Gambar 8. Halaman Price

Halaman harga menyajikan daftar paket bimbingan secara komprehensif untuk memberikan transparansi layanan kepada calon siswa mengenai pilihan paket dan biaya yang tersedia.

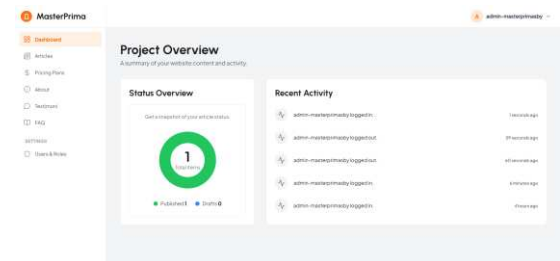
- Halaman Blog & Artikel



Gambar 9. Halaman Blog

Halaman Blog menyajikan daftar artikel terstruktur yang menampilkan konten edukatif untuk mendukung strategi pemasaran konten dan meningkatkan keterlibatan pengguna dengan informasi relevan seputar layanan bimbingan belajar.

- Halaman Panel Admin



Gambar 10. Halaman Dashboard Admin

Halaman Panel merupakan pusat manajemen sistem, dilindungi oleh sistem login yang aman, menawarkan visualisasi data statistik dan log aktivitas terkini. Bilah navigasi yang dirancang intuitif memungkinkan akses mudah ke berbagai menu manajemen konten.

### C. Implementasi Metodologi Agile

Penerapan metodologi Agile dalam proyek ini menggunakan pendekatan kerangka kerja *Solo Scrum*, yang dirancang untuk menjaga fleksibilitas dan efisiensi pengembangan. Proses ini dimulai dengan tahap perencanaan yang matang melalui estimasi beban kerja dan pembagian *sprint*.

#### 1. Story Point

Story Points	Effort Required	Complexity	Risk & Uncertainty
1	Low	Low	Low
2	Medium-Low	Low	Low-Medium
3	Medium	Medium	Medium
5	Medium-High	Medium-High	High
8	High	High	High
13	Very High	High	Medium-High
21	Extremely High	Very High	Extremely High

Gambar 11. Story Point

Sebelum pengembangan dimulai, seluruh kebutuhan sistem yang tercatat dalam *product backlog* diestimasi menggunakan metode *Story Points* dengan skala *Fibonacci* (1, 2, 3, 5, 8, 13, 21) sebagaimana ditunjukkan pada gambar di atas. Penggunaan skala ini bertujuan untuk merepresentasikan tingkat kompleksitas, usaha, dan risiko dari setiap fitur secara relatif. Angka yang semakin besar dalam skala ini mencerminkan ketidakpastian yang lebih tinggi pada tugas yang kompleks, sehingga membantu pengembang dalam mengukur kapasitas kerja yang realistis untuk setiap iterasi *sprint*.

#### 2. Product Backlog

Sprint	Item
Sprint 1	Halaman utama Beranda utama
	Halaman Price Tampilan harga
	Pricing Plan Update Update harga/paket
	Halaman Dashboard Dashboard admin
	Manage Blog CRUD blog
	Halaman About/Tentang Informasi
	Halaman Blog Menulis halaman blog
	Halaman Login Admin Login admin
Sprint 2	Fix: Price/Plan Data Not Displayed on User Page
	Feature: Dynamic Testimonials Management via Admin Dashboard
	Feature: Dynamic Profile & Performance Metrics Management
	Feature: FAQ Management System for Admin
	Enhancement: Admin Dashboard Content Management UI
Backlog	Validation: Display Dynamic Content on Homepage
	Feature: Admin Management System
	Integration: Checkout and Payment Gateway

Gambar 12. Product Backlog

Berdasarkan estimasi tersebut, fitur-fitur dibagi ke dalam dua sprint utama sebagaimana ditunjukkan pada gambar *Product Backlog* di atas.

#### - Sprint 1

*Sprint* pertama difokuskan pada pembangunan fondasi sistem atau *Minimum Viable Product (MVP)* yang meliputi pengembangan halaman utama, sistem login admin, manajemen blog dasar dengan fitur CRUD, serta halaman informasi statis seperti *About* dan *Pricing*.

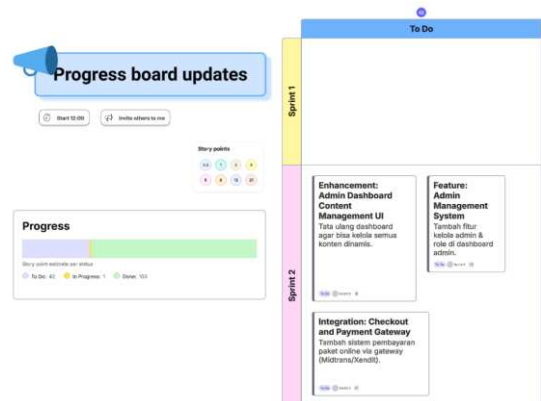
#### - Sprint 2

*Sprint* kedua ditujukan untuk peningkatan fungsionalitas dan pengalaman pengguna melalui pengembangan fitur-fitur yang lebih dinamis seperti manajemen testimoni, pengelolaan *FAQ*, serta validasi konten untuk memastikan seluruh data tampil secara dinamis dari *backend*.

#### 3. Visualisasi dan Pengendalian Progres

Untuk menjaga transparansi dan disiplin kerja, alur penyelesaian tugas dikelola menggunakan papan *Kanban* yang memetakan status pengerjaan fitur secara *real-time*.

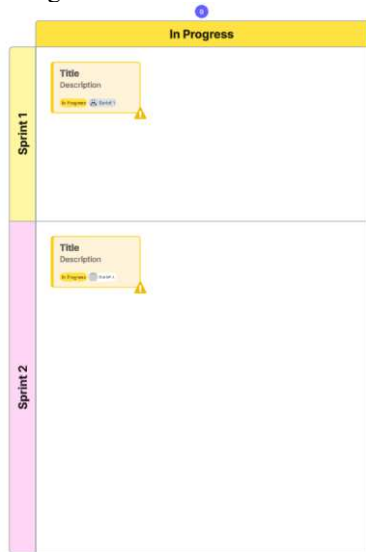
#### - Status To Do



Gambar 13. To Do

Kolom *To Do* memuat daftar antrian tugas yang telah diprioritaskan berdasarkan urgensi bisnis, dengan tugas di posisi teratas merupakan fitur prioritas yang harus diselesaikan terlebih dahulu sebelum mengerjakan fitur pendukung lainnya, sebagaimana ditunjukkan pada gambar di atas.

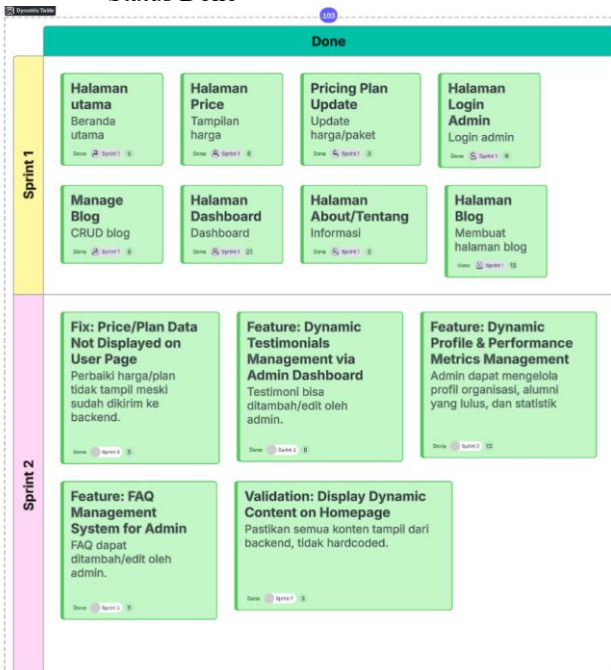
- Status In Progress



Gambar 14. In Progress

Fitur yang sedang dikerjakan dipindahkan ke kolom *In Progress* sebagaimana ditunjukkan pada gambar di atas. Dalam konteks pengembangan individu (*Solo Scrum*), kolom ini berfungsi membatasi pekerjaan aktif (*Work In Progress*) untuk menjaga fokus pengembang, di mana satu tugas harus diselesaikan sepenuhnya sebelum beralih ke tugas lain guna mencegah inefisiensi akibat perpindahan konteks (*context switching*).

- Status Done



Gambar 15. Done

Kolom *Done* menampung fitur-fitur yang telah selesai dikembangkan dan lolos pengujian unit sebagaimana ditunjukkan pada gambar di atas. Pada akhir *Sprint 1* dan *2*, seluruh item dalam *backlog* seperti fitur manajemen blog dan integrasi halaman utama berhasil dipindahkan ke status ini, menandakan bahwa fitur tersebut telah memenuhi kriteria "selesai" (*definition of done*) dan siap digunakan.

D. Implementasi Database



Gambar 16. Database

Infrastruktur penyimpanan data sistem dikembangkan menggunakan layanan *Supabase* sebagai *Backend as a Service (BaaS)* berbasis *PostgreSQL*. Desain skema basis data dirancang secara relasional untuk mendukung pengelolaan konten dinamis yang terstruktur dan aman.

Sebagaimana ditunjukkan pada gambar di atas, arsitektur data terdiri dari beberapa entitas utama yang saling terintegrasi:

- Manajemen Pengguna & Keamanan
  - Tabel *admin* berfungsi sebagai pusat autentikasi dan otorisasi dengan relasi *one-to-many* terhadap tabel *admin\_activity\_log*, yang mencatat setiap aktivitas (*action\_type*), tabel yang dimodifikasi (*table\_name*), dan perubahan data (*changes*) dalam format *JSONB*. Mekanisme ini menciptakan jejak audit (*audit trail*) yang transparan untuk keamanan sistem.
- Manajemen Konten Blog
  - Tabel *articles* menyimpan konten edukatif dan terhubung dengan tabel *admin* sebagai penulis (*author*) serta tabel *article\_views* untuk melacak statistik pengunjung berdasarkan alamat *IP* dan *user agent*.
- Manajemen Layanan & Harga
  - Informasi paket bimbingan dikelola dalam tabel *pricing\_plans* yang memiliki relasi dengan

*pricing\_categories* untuk pengelompokan paket, serta mencatat admin yang melakukan pembaruan terakhir (*updated\_at*) untuk memastikan akuntabilitas data.

- Konten Dinamis Lainnya
  - Sistem juga mencakup tabel *testimonials* untuk ulasan siswa, *team\_members* untuk profil pengajar, *faqs* untuk pertanyaan umum, dan *site\_settings* yang menyimpan konfigurasi global situs seperti kontak *WhatsApp*, email, dan visi-misi agar dapat dimodifikasi tanpa mengubah kode program.

Seluruh tabel menggunakan UUID sebagai *primary key* untuk menjamin keamanan dan skalabilitas, serta dilengkapi dengan timestamp (*created\_at*, *updated\_at*) untuk memastikan integritas data historis. Relasi antar tabel dijaga menggunakan *Foreign Key* yang ketat untuk mencegah terjadinya inkonsistensi data.

### E. Integrasi Pembayaran

```
export async function POST(req: NextRequest) {
  try {
    const { items, customer } = (await req.json()) as {
      items: CartItem[];
      customer?: Customer;
    };

    // Validasi input
    if (!Array.isArray(items) || items.length === 0) {
      return NextResponse.json({ error: "Items required" }, { status: 400 });
    }

    // (opsional) sanitasi qty minimal 1
    const safeItems = items.map(it => ({
      ...it,
      qty: typeof it.qty === "number" && it.qty > 0 ? it.qty : 1,
    }));

    // Buat invoice di Xendit
    const invoice = await createInvoice(safeItems, customer);

    // Upsert transaksi PENDING agar langsung tampil di Admin
    // (kalau gagal, log saja; status final tetap akan dikirim via webhook)
    try {
      await supabaseAdmin.from("transactions").upsert(
        {
          id: invoice.id, // invoice_id Xendit sebagai PK
          external_id: invoice.external_id,
          amount: Number(invoice.amount ?? 0),
          status: "PENDING",
          items: safeItems,
          customer_details: customer ?? null,
          created_at: new Date().toISOString(),
          updated_at: new Date().toISOString(),
        },
        { onConflict: "id" }
      );
    } catch (dbErr) {
      console.error("[create-invoice] upsert PENDING failed:", dbErr);
      // jangan melempar error; biarkan invoice tetap dikirim ke client
    }

    // Kembalikan URL untuk redirect ke halaman pembayaran Xendit
    return NextResponse.json({
      invoice_url: invoice.invoice_url,
      id: invoice.id,
      external_id: invoice.external_id,
    });
  } catch (e: any) {
    console.error("[create-invoice] error:", e);
    return NextResponse.json(
      { error: e?.message || "Create invoice failed" },
      { status: 500 }
    );
  }
}
```

Gambar 17. Potongan kode integrasi pembayaran

Untuk memfasilitasi transaksi digital yang aman dan otomatis, sistem diintegrasikan dengan gerbang pembayaran (*payment gateway*) Xendit sebagaimana ditunjukkan pada gambar di atas. Integrasi ini memungkinkan sistem untuk menerbitkan tagihan (*invoice*), menerima pembayaran melalui berbagai metode seperti transfer bank dan dompet digital, serta memverifikasi status pembayaran secara otomatis tanpa intervensi manual.

Proses teknis pembuatan tagihan diimplementasikan melalui fungsi *createInvoice* pada sisi *backend* dengan alur logika sebagai berikut:

1. Sistem menerima data pesanan dan informasi pelanggan dari frontend
2. Fungsi memproses permintaan ke API Xendit untuk menghasilkan tautan pembayaran yang unik
3. Data transaksi awal disimpan ke dalam tabel *transactions* di Supabase dengan status default 'PENDING'
4. Tautan pembayaran dikembalikan kepada pengguna untuk diselesaikan

Setelah pengguna menyelesaikan pembayaran, sistem memanfaatkan mekanisme *webhook* dari Xendit untuk menerima konfirmasi secara otomatis. Mekanisme ini memicu pembaruan status transaksi di basis data menjadi 'PAID' secara *real-time*, sehingga akses layanan dapat segera diberikan kepada pengguna.

### F. Implementasi Teknis dan Backend

Secara teknis, sistem ini memiliki arsitektur modular. Komponen antarmuka dibagi menjadi bagian-bagian yang lebih kecil dan dapat digunakan kembali, seperti bilah navigasi, footer, dan tab artikel.

#### 1. Manajemen API & Autentikasi

```
potongan code api route article TypeScript
1 import { NextRequest, NextResponse } from 'next/server';
2 import { supabaseAdmin } from '@lib/supabase/admin';
3 import { verifySessionPayload } from '@lib/auth/utils';
4
5 export async function GET(request: NextRequest) {
6   const sessionToken = request.cookies.get('admin-session')?.value;
7   if (!sessionToken) return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
8
9   const currentAdmin = verifySessionPayload(sessionToken);
10  if (!currentAdmin?.is_admin) return NextResponse.json({ error: 'Forbidden' }, { status: 403 });
11
12  const { data, error } = await supabaseAdmin
13    .from('articles')
14    .select('article_id, title, updated_at')
15    .order('updated_at', { ascending: false });
16
17  if (error) throw new Error('Failed to fetch articles.');
```

Gambar 18. Potongan kode article API routes

Komunikasi data dalam sistem ditangani melalui API Routes *Next.js* sebagaimana ditunjukkan pada gambar di atas. Implementasi endpoint untuk manajemen artikel menggunakan metode *GET* dan *POST* yang dilindungi oleh sistem autentikasi berbasis sesi. Token sesi disimpan dalam *HTTP-only cookies* dan divalidasi menggunakan fungsi *verifySessionPayload* sebelum data dari basis data Supabase dapat diakses, memastikan keamanan dan integritas setiap transaksi data yang dilakukan.

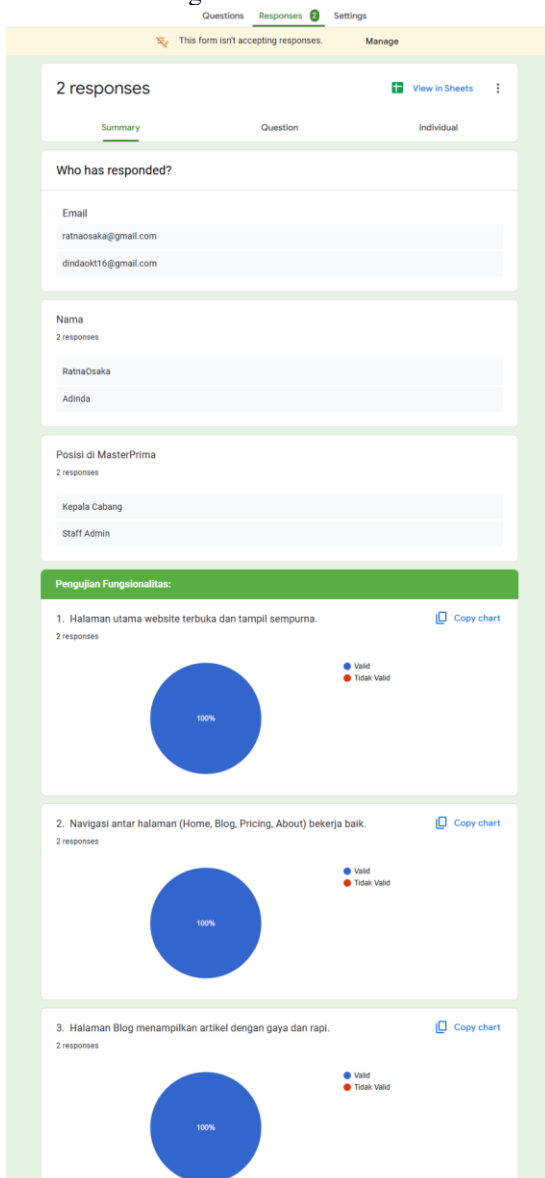
## 2. Keamanan Data

Selain validasi di level aplikasi, keamanan data diperkuat dengan penerapan *Row-Level Security* (RLS) pada Supabase. Fitur ini membatasi akses ke tabel-tabel sensitif seperti *admin* dan *transactions* langsung di tingkat basis data, memastikan hanya pengguna dengan hak akses yang sah yang dapat membaca atau memodifikasi data.

## G. Hasil Uji Sistem

Tahap akhir implementasi adalah validasi kualitas sistem melalui dua metode pengujian utama untuk menjamin fungsi teknis dan penerimaan pengguna.

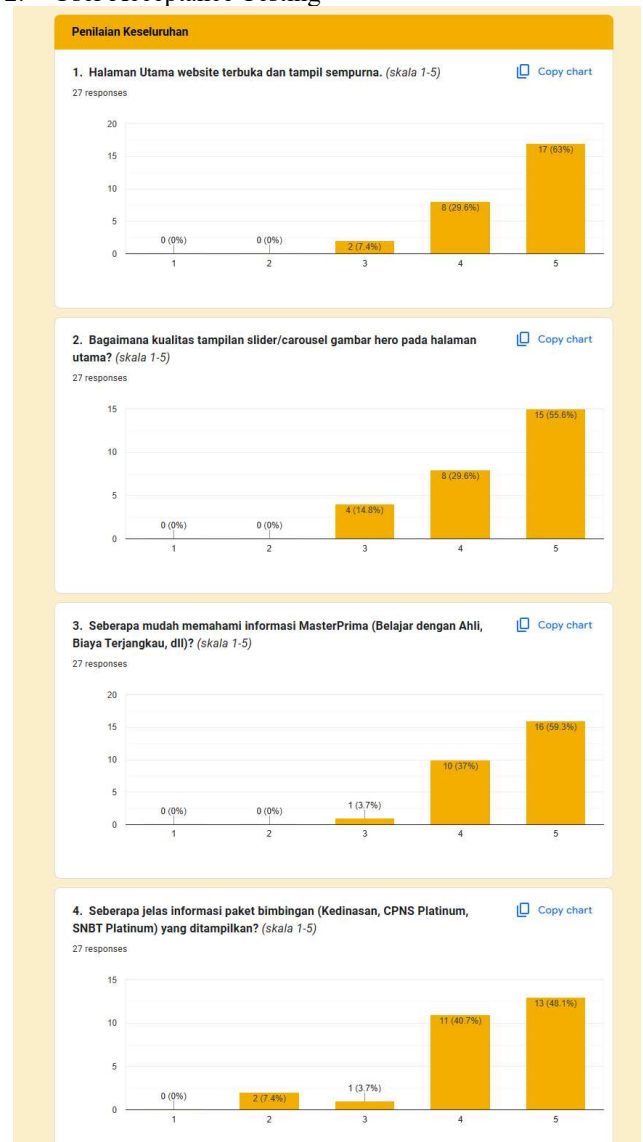
### 1. Black-box Testing



Gambar 19. Black-box Testing

*Black-box* testing berfokus pada validasi fungsionalitas fitur tanpa melihat struktur kode internal. Berdasarkan hasil pengujian sebagaimana ditunjukkan pada gambar di atas, seluruh skenario uji terhadap fitur krusial seperti navigasi halaman, formulir login, manajemen CRUD artikel, dan integrasi pembayaran berhasil dijalankan dengan status 100% Valid. Tidak ditemukan kegagalan fungsi (*error*) pada alur utama sistem yang diuji, menunjukkan bahwa seluruh fitur berfungsi sesuai dengan spesifikasi yang telah ditetapkan.

### 2. User Acceptance Testing



Gambar 20. User Acceptance Testing

Pengujian penerimaan pengguna (*UAT*) dilakukan terhadap 27 responden untuk menilai kegunaan dan pengalaman pengguna menggunakan kuesioner skala Likert (1-5). Hasil ringkasan, yang disajikan pada grafik di atas, menunjukkan bahwa mayoritas responden memberikan penilaian sangat

positif, dengan rata-rata penilaian berkisar antara 4,4 hingga 4,7. Hal ini menunjukkan bahwa antarmuka visual situs web, kemudahan navigasi, dan kejelasan informasi sangat memuaskan dan diterima dengan baik oleh pengguna target.

## V. KESIMPULAN

### A. Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan sistem aplikasi web promosi untuk Bimbingan Belajar "Master Prima" cabang SMA Trimurti Surabaya menggunakan *Next.js* sebagai *frontend* dan *Supabase* sebagai *backend*. Penerapan metodologi Agile dengan pendekatan *Solo Scrum* dalam dua *sprint* memungkinkan pengembangan yang fleksibel dan responsif terhadap kebutuhan fungsional. Hasil pengujian *black-box* menunjukkan validitas 100% untuk seluruh fitur utama tanpa kendala teknis, sementara *User Acceptance Testing (UAT)* mengonfirmasi tingkat penerimaan pengguna yang tinggi dengan penilaian sangat memuaskan terhadap kemudahan penggunaan dan tampilan visual. Secara keseluruhan, sistem ini layak dan siap digunakan sebagai media promosi digital mandiri yang efisien bagi institusi pendidikan.

### B. Saran

Untuk pengembangan lebih lanjut, disarankan agar sistem dilengkapi dengan fitur pendaftaran *online*, integrasi pembayaran digital yang komprehensif, dan *live chat* untuk meningkatkan interaksi dengan calon siswa. Dari aspek teknis, perlu dilakukan pengujian keamanan mendalam (*penetration testing*) pada halaman admin serta penerapan sistem manajemen konten (*CMS*) internal yang lebih canggih agar staf pemasaran dapat mengelola informasi secara mandiri tanpa ketergantungan teknis yang tinggi.

## REFERENSI

- [1] A. Ridow, J. Metekohy, J. I. Sihotang, and J. Simarmata, "Pengantar Teknologi Digital: Web dan Mobile Teknologi," 2024. [Online]. Available: <https://www.researchgate.net/publication/381291236>
- [2] Moch. A. Karyawan, M. Nurhadi, A. C. Puspitaningrum, and M. S. Prasetya, "Structural Model Kekuatan Kualitas Website dalam Membangun Citra Perguruan Tinggi," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 3, pp. 224–233, Jul. 2023, doi: 10.47233/jteksis.v5i3.857.
- [3] R. Ivandi, Y. T. Mursityo, and W. H. N. Putra, "Implementasi Marketing Automation Menggunakan Metode Search Engine Optimization (SEO) On Page dan Off Page pada Website Berbasis Content Management System (CMS) pada PT. Eltama Prima Indo," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 1, pp. 77–88, Feb. 2023, doi: 10.25126/jtiik.20231015120.
- [4] Harry Saptarianto, Shelvi Deviani, Syamas Isti Anah, and Indah Noviyanti, "Menghadapi Tantangan Era Digital, Strategi Integrasi Media Sosial, Literasi Digital dan Inovasi Bisnis," *Jurnal Manuhara : Pusat Penelitian Ilmu Manajemen dan Bisnis*, vol. 2, no. 3, pp. 128–139, Jun. 2024, doi: 10.61132/manuhara.v2i3.955.
- [5] K. Wahyudi and Badrus Soleh, "ONLINE-BASED PROMOTION MODEL IN RECRUITING OUTSTANDING NEW STUDENTS," *re-JIEM (Research Journal of Islamic Education Management)*, vol. 6, no. 1, pp. 86–97, May 2023, doi: 10.19105/re-jiem.v6i1.8797.
- [6] A. Mishra and Y. I. Alzoubi, "Structured software development versus agile software development: a comparative analysis," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 4, pp. 1504–1522, Aug. 2023, doi: 10.1007/s13198-023-01958-5.
- [7] I. Made, P. Muliada, A. A. Istri, I. Paramitha, N. Purnama, and P. Korespondensi, "Jurnal Restikom : Riset Teknik Informatika dan Komputer PENGEMBANGAN SISTEM BOOKING ENGINE PADA GRIPASTUDIO A B S T R A K," vol. 6, no. 2, pp. 261–270, 2024, [Online]. Available: <https://restikom.nusaputra.ac.id>
- [8] S. Suwarno and W. S. Jaya, "Design and Development of Software Project Management System using Scrum," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 5, no. 2, pp. 483–493, Jan. 2022, doi: 10.31289/jite.v5i2.6412.
- [9] O. C. Resmi Rachmawati, Deyana Kusuma Wardani, Wifda Muna Fatihia, Arna Fariza, and Hestiasari Rante, "Implementing Agile Scrum Methodology in The Development of SICITRA Mobile Application," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, pp. 41–50, Feb. 2023, doi: 10.29207/resti.v7i1.4688.
- [10] I. Tahyudin and Zidni Iman Sholihati, "Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 169–176, Feb. 2022, doi: 10.29207/resti.v6i1.3793.
- [11] S. Moyo and E. Mnkandla, "A Novel Lightweight Solo Software Development Methodology with Optimum Security Practices," *IEEE Access*, vol. 8, pp. 33735–33747, 2020, doi: 10.1109/ACCESS.2020.2971000.
- [12] F. G. P. Susanto, N. I. Y. Fadlan, and P. Haryani, "Design of Web-Based Management Information System for Student Organizations in Kendal Regency Using Next.js Framework," *Compiler*, vol. 12, no. 1, p. 9, May 2023, doi: 10.28989/compiler.v12i1.1616.
- [13] S. Famy and T. Tukino, "Design and Build a Web-based Management Information System at CV Sanjaya Abadi Baru," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 5, no. 2, pp. 371–383, Jan. 2022, doi: 10.31289/jite.v5i2.6184.
- [14] D. Febrianti, A. Hadi, Y. Hendriyani, G. Farell, J. Hamka Kampus UNP, and A. Tawar Padang, "Jurnal Vocational Teknik Elektronika dan Informatika," vol. 12, no. 2, 2024, [Online]. Available: <http://ejournal.unp.ac.id/index.php/voteknika/>
- [15] M. A. Bastari, D. Darmansah, and D. P. Rakhmadani, "Sistem Informasi Jasa Cuci Interior Rumah dan Mobil Menggunakan Metode User Acceptance Test," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, p. 305, Apr. 2022, doi: 10.30865/jurikom.v9i2.3926.
- [16] A. Wandu, S. #1, H. Priyanto, and H. Sujaini, "Rancang Bangun Aplikasi Media Pembelajaran Tes CPNS Berbasis Web," vol. 6, no. 1, pp. 14–25, 2018.