

PENGEMBANGAN APLIKASI WEB MANAJEMEN IKLAN RADIO DAN VIDEOTRON DI KOTA SALATIGA MENGGUNAKAN MERN STACK

Tiara Syah Indra Putri^{*1)}, Evangs Mailoa²⁾

1. Teknik Informatika, Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia
2. Teknik Informatika, Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia

Article Info

Kata Kunci: Aplikasi Web; Javascript; MERN; MongoDB; Express; ReactJS; NodeJS

Keywords: Web Application; Javascript; MERN; MongoDB; Express; ReactJS; NodeJS

Article history:

Received 21 August 2024

Revised 11 September 2024

Accepted 5 October 2024

Available online 1 September 2025

DOI :

<https://doi.org/10.29100/jipi.v10i3.6496>

* Corresponding author.

Corresponding Author

E-mail address:

*tiarasyahip@gmail.com

evangs.mailoa@uksw.edu

ABSTRAK

Iklan radio dan videotron memainkan peran penting dalam periklanan modern, termasuk di Kota Salatiga, di mana kedua media ini turut berkontribusi dalam membentuk identitas kota dan mendukung strategi bisnis. Seiring dengan meningkatnya penggunaan media ini, muncul permasalahan dalam manajemen data iklan yang dilakukan secara manual menggunakan spreadsheet dan pencatatan manual, yang mengakibatkan ketidaksesuaian data. Untuk mengatasi masalah ini, penelitian ini bertujuan untuk mengembangkan aplikasi web manajemen iklan radio dan videotron dengan memanfaatkan teknologi *MERN Stack*, yang terdiri dari *MongoDB*, *Express.js*, *React.js*, dan *Node.js*. *MERN Stack* dipilih karena kemampuannya dalam membangun aplikasi dinamis dan skalabel dengan menggunakan JavaScript. Dengan menerapkan arsitektur *MVC (Model, View, Controller)*, aplikasi ini diharapkan dapat meningkatkan efisiensi pengelolaan data iklan, mengotomatisasi proses manajemen, dan memperbaiki efektivitas sistem yang ada. Penelitian ini juga meninjau relevansi penggunaan *MERN Stack* dalam konteks aplikasi manajemen data lainnya dan membandingkannya dengan pendekatan yang telah ada sebelumnya.

ABSTRACT

Radio and videotron advertisements play a crucial role in modern advertising, including in Salatiga City, where these media contribute to shaping the city's identity and supporting business strategies. With the increasing use of these media, problems have arisen in managing advertisement data, which has been done manually using spreadsheets and manual records, leading to data discrepancies. To address these issues, this study aims to develop a web-based advertisement management application for radio and videotron by leveraging the *MERN Stack* technology, which consists of *MongoDB*, *Express.js*, *React.js*, and *Node.js*. The *MERN Stack* is chosen for its ability to build dynamic and scalable applications using JavaScript. By implementing the *MVC (Model, View, Controller)* architecture, this application is expected to enhance the efficiency of advertisement data management, automate management processes, and improve the effectiveness of the existing system. This study also reviews the relevance of using the *MERN Stack* in the context of other data management applications and compares it with existing approaches.

I. PENDAHULUAN

Iklan radio dan videotron memiliki peran yang sangat penting dalam dunia periklanan modern termasuk di Kota Salatiga[1]. Kota Salatiga, sebagai salah satu kota yang berkembang di Indonesia, turut merasakan dampak positif dari strategi periklanan yang menggunakan iklan radio dan videotron[2][3]. Di sisi lain, videotron atau *billboard* elektronik juga memainkan peran penting dalam pemandangan kota Salatiga. Melalui layar besar yang menarik perhatian, iklan-iklan yang ditampilkan dapat dengan cepat mencapai khalayak yang berlalu-lalang di jalan-jalan utama kota[4]. Keberadaan iklan radio dan videotron di Kota Salatiga tidak hanya memberikan manfaat bagi pelaku bisnis dan industri periklanan, tetapi juga memberikan kontribusi positif dalam membentuk identitas kota. Dalam konteks globalisasi dan persaingan bisnis yang semakin ketat, pemanfaatan iklan radio dan videotron di Kota Salatiga menjadi strategi yang tak terhindarkan bagi pelaku bisnis yang ingin tetap relevan dan bersaing secara efektif.

Dalam beberapa tahun terakhir, teknologi berbasis web telah mengalami perkembangan pesat dengan munculnya berbagai stack teknologi yang mempermudah pengembangan aplikasi manajemen data yang kompleks. *MERN Stack* (*MongoDB*, *Express.js*, *React.js*, dan *Node.js*) telah menjadi salah satu solusi populer dalam pengembangan aplikasi web modern karena kemampuannya untuk menyediakan pengembangan *full-stack* berbasis *JavaScript*, yang mendukung efisiensi dan kecepatan dalam pembuatan aplikasi web. *MERN Stack*, yang terdiri dari *MongoDB*, *Express.js*, *React.js*, dan *Node.js*, merupakan pilihan populer untuk pengembangan aplikasi web modern karena sifatnya yang *full-stack* dan berbasis *JavaScript*[5][6]. Dibandingkan dengan teknologi lain seperti *Django* dan *Ruby on Rails*, *MERN Stack* menawarkan keunggulan dalam hal kecepatan pengembangan dan performa, terutama untuk aplikasi *single-page* yang membutuhkan interaktivitas tinggi. *Django*, yang berbasis *Python*, dikenal dengan arsitektur yang kuat dan skalabilitas besar, namun sering kali memerlukan lebih banyak konfigurasi dan tidak seefisien *MERN* dalam hal *real-time updates*. *Ruby on Rails* menawarkan kemudahan dalam pengembangan dengan konvensi kuat dan produktivitas tinggi, tetapi cenderung memiliki performa yang lebih lambat dibandingkan *Node.js*, terutama dalam aplikasi yang membutuhkan kecepatan dan skalabilitas tinggi. *MERN Stack* unggul dalam aplikasi yang memerlukan interaksi pengguna intensif, sementara *Django* lebih unggul dalam aplikasi yang memerlukan keamanan tingkat tinggi dan kompleksitas *backend* yang signifikan[7]. Sedangkan *Ruby on Rails* menawarkan pengembangan yang lebih cepat untuk prototipe, tetapi memiliki keterbatasan dalam skala besar dibandingkan *MERN Stack*[8]. Keunggulan *MERN Stack* dalam pengembangan *end-to-end* menggunakan satu bahasa pemrograman, yang memudahkan integrasi antara *front-end* dan *back-end*[9]. Di sisi lain, *Django* lebih diandalkan untuk aplikasi enterprise karena kemampuannya dalam menangani transaksi dan integritas data yang kompleks[10]. Pilihan teknologi harus disesuaikan dengan kebutuhan spesifik proyek, dengan *MERN Stack* lebih cocok untuk aplikasi yang membutuhkan skalabilitas tinggi dan interaksi pengguna, sementara *Django* dan *Ruby on Rails* lebih sesuai untuk aplikasi yang memerlukan struktur backend yang kuat dan manajemen data yang kompleks[11].

Topik penelitian ini diambil berdasarkan kebutuhan yang diidentifikasi oleh Diskominfo Kota Salatiga. Adanya permintaan untuk mengatasi permasalahan pengelolaan iklan radio dan videotron telah mendorong pengembangan solusi berbasis teknologi yang lebih efisien. Seiring dengan semakin meluasnya penggunaan jasa pengiklanan melalui radio dan videotron oleh pelaku bisnis maupun instansi pemerintahan di Kota Salatiga, timbul permasalahan signifikan terkait manajemen data iklan yang masuk. Permasalahan ini mencakup proses pengumpulan, pengelolaan, dan pelaporan data iklan, termasuk pemasukan data dari pengiklan, pengelolaan biaya yang terkait dengan iklan, serta penyediaan laporan yang jelas untuk kebutuhan akuntansi.

Pada awalnya, manajemen data ini dilakukan secara manual menggunakan *spreadsheet Excel* dan pencatatan struk secara manual. Pendekatan manual ini telah menyebabkan beberapa ketidaksesuaian data saat melakukan perhitungan akhir. Oleh karena itu, diperlukan suatu solusi yang lebih efisien dan terotomatisasi dalam pengelolaan data iklan, dan pengembangan aplikasi web manajemen iklan radio dan videotron menggunakan *MERN stack* menjadi alternatif yang relevan dan dapat memperbaiki efektivitas manajemen data yang ada.

Penelitian ini menawarkan kebaruan dalam pengembangan aplikasi manajemen iklan yang tidak hanya mengintegrasikan pengelolaan iklan radio dan videotron dalam satu *platform*, tetapi juga memanfaatkan keunggulan *MERN Stack* dalam hal efisiensi, keamanan, dan skalabilitas. Dengan fokus pada Kota Salatiga, penelitian ini juga memperkenalkan pendekatan yang spesifik terhadap kebutuhan lokal, yang dapat memberikan solusi yang lebih terukur dan relevan dibandingkan dengan aplikasi serupa yang ada. Selain itu, penerapan arsitektur *MVC* (*Model*, *View*, *Controller*) pada *MERN Stack* untuk konteks ini menawarkan struktur kode yang lebih mudah dikelola, memungkinkan penambahan fitur atau modifikasi sistem di masa depan tanpa mengorbankan stabilitas aplikasi.

Penelitian terkait menunjukkan relevansi dan penerapan teknologi *MERN Stack* dalam berbagai konteks. Misalnya, penelitian berjudul "Penerapan Teknologi Stack *MERN* pada Aplikasi Manajemen Layanan Bengkel Berbasis Web" bertujuan untuk mengatasi permasalahan yang dihadapi oleh bengkel XYZ Motor dalam mengelola proses bisnis mereka yang masih dilakukan secara manual. Penelitian ini menghasilkan sebuah aplikasi manajemen layanan berbasis web dengan menggunakan teknologi *MERN Stack* (*MongoDB*, *Express.js*, *React.js*, dan *Node.js*) untuk mengotomatisasi proses bisnis bengkel. Penulis menggunakan metode *WDLC* (*Web Development Life Cycle*) untuk menentukan ruang lingkup sistem yang akan dibuat dan mempersiapkan jadwal agar sistem dikembangkan secara terorganisir. Penelitian ini mencakup pemanfaatan *MERN Stack* dalam membangun aplikasi berbasis web dengan memanfaatkan fungsi-fungsi *Express* seperti pengambilan data, penerimaan data, dan pengaturan route[12]. Penggunaan *MERN Stack* dengan bahasa pemrograman *JavaScript* mempermudah penulis dalam mengimplementasikan sistem tersebut[13].

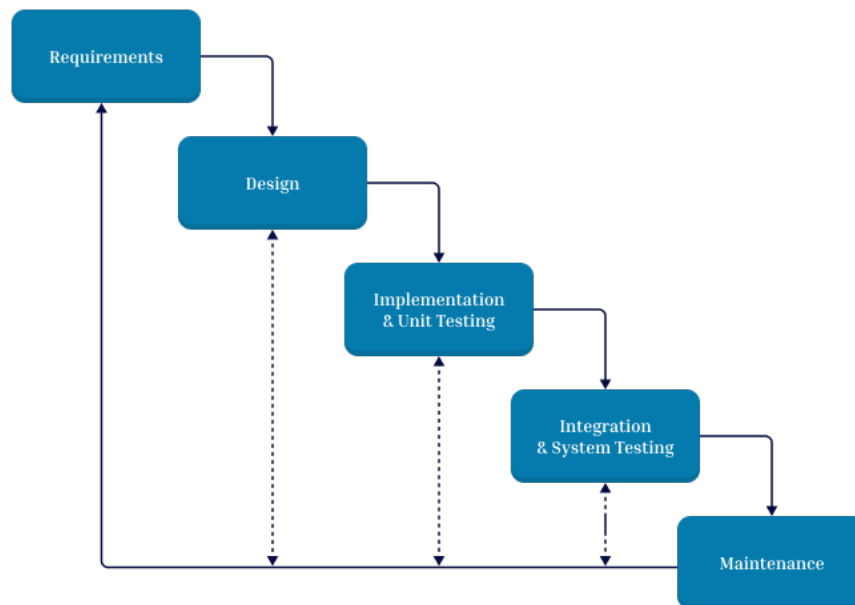
Penelitian lainnya, berjudul “Perancangan Aplikasi Sistem Penjualan *Top-Up Game Online* Berbasis Website Memanfaatkan Fitur *MERN stack* Menggunakan Metode *Waterfall*,” bertujuan untuk mengatasi permasalahan dalam pembelian voucher game online dengan menciptakan aplikasi web yang memfasilitasi proses pembelian. Penelitian ini menggunakan teknologi *MERN Stack* dan metode *waterfall* dalam pengembangan aplikasi, menunjukkan kesamaan pendekatan dengan penelitian penulis. Kontribusi penelitian ini adalah memperjelas penerapan *MERN Stack* dan metode *waterfall* dalam meningkatkan efisiensi proses bisnis online, terutama dalam manajemen data[14].

Penelitian lain bertujuan membantu masyarakat menemukan tempat menginap yang mudah diakses saat berlibur. Penggunaan aplikasi web dan mobile yang semakin meningkat sesuai hasil survei menunjukkan kebutuhan yang semakin besar. Website *Staycation* dibuat menggunakan teknologi *MERN Stack* untuk pengembangan *full-stack* dengan kecepatan dan efisiensi pengolahan data. Metode *Extreme Programming* memungkinkan adaptasi cepat dengan kebutuhan pengguna. Fitur lengkap website ini termasuk integrasi sistem pembayaran online untuk meningkatkan keamanan dan kepercayaan pengguna. Pengujian menggunakan metode *black box* untuk memastikan kualitasnya. Kesamaan dengan jurnal yang disebutkan adalah penggunaan teknologi *MERN Stack* dan metode *waterfall* dalam pengembangan aplikasi, serta fokus pada manajemen data untuk meningkatkan kinerja aplikasi[15].

Penelitian berjudul “Perancangan Website Iklan Radio Kota Salatiga menggunakan Framework *Laravel*” bertujuan membangun sebuah sistem pencatatan iklan radio menggunakan bahasa pemrograman *PHP* dan memanfaatkan *Framework Laravel*. Penulis menggunakan metode *prototype*. Hasil penelitian ini berupa suatu aplikasi pencatatan iklan radio untuk memantau data iklan yang masuk. Penelitian ini memiliki hubungan kasus yang sama yaitu membahas mengenai aplikasi pencatatan iklan. Pada penelitian tersebut hanya membahas mengenai iklan radio, sehingga pada penelitian ini akan dibuat sistem manajemen data iklan radio dan videotron yang lebih rinci[16].

II. METODE PENELITIAN

Metode penelitian yang dipakai dalam membangun Pengembangan Aplikasi Web Manajemen Iklan Radio dan Videotron di Kota Salatiga adalah *Waterfall Model*. *Model Waterfall* adalah salah satu metode dalam *Software Development Life Cycle (SDLC)* yang sering diterapkan dalam pengembangan perangkat lunak [17]. Metode *Waterfall* adalah salah satu jenis pengembangan aplikasi dengan penekanan pada fase yang sistematis dan berurutan. Namun, model ini memiliki keterbatasan fleksibilitas karena sulit untuk mengakomodasi perubahan kebutuhan atau persyaratan yang muncul di tengah proses pengembangan. Setelah suatu tahap selesai, sulit untuk kembali ke tahap sebelumnya tanpa mengganggu alur proyek secara keseluruhan. *Model Waterfall* adalah model yang umum digunakan dalam pengembangan perangkat lunak atau rekayasa perangkat lunak dan termasuk dalam kategori model generik dalam bidang tersebut[18][19]. Selain itu, risiko yang dihadapi dalam model ini cukup signifikan, terutama terkait kesulitan dalam mendeteksi kesalahan pada tahap akhir. Hal ini disebabkan oleh sifatnya yang linier, di mana pengujian dilakukan setelah semua tahap selesai, sehingga kesalahan yang terakumulasi pada tahap awal mungkin baru terdeteksi di tahap akhir, yang bisa meningkatkan biaya dan waktu untuk perbaikan. *Model Waterfall* memiliki beberapa kelebihan yang bisa didapatkan dalam proses implementasinya, yaitu proses pengembangan dilakukan secara satu per satu sehingga dapat meminimalisir kesalahan yang mungkin akan terjadi, setiap proses dilakukan secara sistematis dan bertahap sehingga alur kerja menjadi jelas dan terukur, lebih hemat biaya karena tidak banyak membutuhkan sumber daya, dokumentasi yang baik, dan cocok untuk pengembangan perangkat lunak berskala besar seperti Pengembangan Aplikasi Web Manajemen Iklan Radio dan Videotron di Kota Salatiga[20].



Gambar 1. Metode Waterfall

Tahapan penelitian yang ditunjukkan pada Gambar 1 adalah sebagai berikut:

A. TAHAP REQUIREMENT ANALYSIS

Analisis kebutuhan merupakan tahapan pertama dalam *model Waterfall*. Analisis kebutuhan dilakukan untuk mengumpulkan seluruh informasi mengenai perangkat lunak seperti kegunaan dan batasan perangkat lunak. Analisis kebutuhan sistem dapat dibagi menjadi dua jenis, yaitu analisis kebutuhan fungsional dan non-fungsional. Hal ini dapat dijadikan sebagai acuan untuk menentukan fitur-fitur atau layanan yang akan dikembangkan. Informasi yang didapatkan bisa bersumber dari hasil diskusi, wawancara, dan survei. Dalam Pengembangan Aplikasi Web Manajemen Iklan Radio dan Videotron di Kota Salatiga, tahapan pertama yang dilakukan adalah identifikasi masalah. Dengan adanya kendala dalam pengelolaan data iklan yang sebelumnya dilakukan secara manual dengan menggunakan Excel membuat proses pengecekan data iklan oleh pihak pengelola menjadi kurang efisien. Maka, dengan adanya permasalahan tersebut, dibangunlah sebuah sistem yang dapat mampu melakukan pengecekan data Iklan Radio dan Videotron. Pengumpulan data untuk pengembangan aplikasi dapat diperoleh dari data pengelola iklan kota salatiga. Semua informasi tersebut kemudian dianalisis dan diproses untuk mendapatkan gambaran lengkap tentang spesifikasi kebutuhan pengguna dan perangkat lunak yang akan dikembangkan. Pemilihan teknologi *MERN Stack* untuk proyek ini dipertimbangkan karena keunggulannya dalam efisiensi, skalabilitas, dan keamanan. Teknologi lain seperti *PHP* atau *.NET* juga memiliki keunggulan, namun *MERN Stack* memberikan fleksibilitas penuh dengan satu bahasa pemrograman, yaitu JavaScript, yang mengurangi kompleksitas pengembangan dan memungkinkan integrasi yang lebih baik antara *front-end* dan *back-end*. Selain itu, informasi mengenai kebutuhan pengguna dapat dikumpulkan secara menyeluruh, termasuk penentuan platform perangkat lunak, bahasa pemrograman, *database*, dan teknologi yang akan digunakan dalam proses pengembangan perangkat lunak. Aplikasi Web Manajemen Iklan Radio dan Videotron di Kota Salatiga akan berbasis website dan dikembangkan menggunakan teknologi *MERN Stack*.

B. TAHAP DESIGN

Tahapan desain adalah proses perancangan dan pengembangan sistem *software* berdasarkan informasi kebutuhan pengguna. Perancangan desain dilakukan untuk mempermudah proses pengerjaan dengan memberikan gambaran menyeluruh tentang apa yang harus dikerjakan dan memperoleh detail lengkap dari sistem yang akan dikembangkan. Tahapan desain pada *model Waterfall* berfungsi untuk membantu mengidentifikasi kebutuhan perangkat keras/*hardware* dan sistem yang diperlukan untuk keseluruhan proses pengembangan. Perancangan sistem mencakup alokasi kebutuhan perangkat keras dan perangkat lunak dengan menyusun arsitektur sistem secara menyeluruh. Dalam Pengembangan Aplikasi Web Manajemen Iklan Radio dan Videotron di Kota Salatiga, perancangan sistem dibuat dengan menggunakan *Unified Modelling Language (UML)* yang digambarkan dengan *Use Case Diagram* dan *Activity Diagram*.

C. TAHAP IMPLEMENTATION AND UNIT TESTING

Tahapan yang ketiga adalah *Implementation and Unit Testing*. Tahapan ini adalah tahapan implementasi yang mengarah pada proses pemrograman atau penulisan code program. Pada tahapan *implementation*, pembuatan perangkat lunak atau *software* akan dibagi menjadi modul-modul kecil yang nantinya akan digabungkan pada tahapan selanjutnya dalam *model Waterfall*. Pada tahapan *Unit Testing*, dilakukan pemeriksaan dan pengujian lebih dalam terhadap modul yang sudah dibuat dengan tujuan memastikan modul yang dikerjakan memenuhi kriteria dan fungsi yang sesuai dengan standar. Dalam Pengembangan Aplikasi Web Manajemen Iklan Radio dan Videotron di Kota Salatiga, rancangan yang telah dibuat pada tahap sebelumnya diimplementasikan menggunakan *ReactJS* untuk membuat tampilan antarmuka aplikasi pada sisi *front-end*. Sementara itu, pada sisi *back-end*, implementasi dilakukan dengan menggunakan kerangka kerja aplikasi *Express* dan *database* menggunakan *MongoDB*. Tantangan utama dalam mengimplementasikan *MERN Stack* pada proyek ini termasuk adaptasi dengan lingkungan teknologi yang sudah ada, serta kebutuhan untuk memastikan bahwa integrasi antara berbagai komponen stack berjalan lancar. Untuk mengatasi tantangan ini, dilakukan pengujian berkala dan penggunaan alat-alat pengembangan modern seperti *Visual Studio Code* untuk memantau kesalahan secara *real-time* dan memastikan pengembangan berjalan sesuai rencana. Proses implementasi pembangunan aplikasi menggunakan *Code Editor Visual Studio Code*, yang dapat membantu para pengembang dengan ekstensi dan mendukung banyak bahasa pemrograman, termasuk bahasa pemrograman *JavaScript*.

D. TAHAP INTEGRATION AND SYSTEM TESTING

Tahap keempat adalah *Integration and System Testing* yang mengacu kepada pengintegrasian dari setiap modul. Setelah proses pengintegrasian modul selesai, langkah selanjutnya adalah melakukan pengujian dan pemeriksaan terhadap sistem untuk mengecek jalannya fungsi sistem secara keseluruhan. Pengujian dan pemeriksaan secara keseluruhan dilakukan untuk mengidentifikasi kemungkinan adanya kesalahan dan kegagalan sistem. Risiko yang muncul pada tahap ini adalah kemungkinan adanya kesalahan sistem yang baru terdeteksi pada tahap akhir, yang bisa menyebabkan penundaan dalam peluncuran aplikasi. Untuk memitigasi risiko ini, pengujian dilakukan secara menyeluruh dengan pendekatan *black box*, yang memungkinkan tim untuk mendeteksi kesalahan tanpa melihat kode internal. Pengujian *black box* dilakukan dari sisi fungsionalitasnya agar dapat memenuhi fungsi sebagaimana mestinya. Proses pengujian nantinya akan dibuat sebagai laporan, apabila terdapat *bug* atau *error* dalam aplikasi yang dikembangkan, maka aplikasi tersebut akan dikembalikan kepada *developer* untuk diperbaiki kesalahannya. Apabila aplikasi sudah sesuai dengan perancangan maka akan dilanjutkan ke tahapan atau proses selanjutnya.

E. TAHAP MAINTENANCE

Tahap *Maintenance* merupakan tahapan terakhir dari *model Waterfall*. Tahapan ini dilakukan agar sistem terpelihara dan memastikan sistem tetap berjalan baik sesuai dengan fungsinya. Tahapan ini meliputi beberapa proses, antara lain perbaikan implementasi unit sistem, peningkatan performa sistem sesuai dengan kebutuhan, dan perbaikan error.

III. HASIL PEMBAHASAN

HASIL PENELITIAN

Penelitian ini menghasilkan manajemen data berbasis website pada aplikasi manajemen Iklan Radio dan Videotron yang dibangun menggunakan *framework MERN*.

A. ANALISA KEBUTUHAN SISTEM

Dalam pembangunan aplikasi web manajemen Iklan Radio dan Videotron memerlukan beberapa kebutuhan perangkat lunak dan perangkat keras yang dipergunakan untuk mendukung pembangunan aplikasi ini, antara lain:

Kebutuhan perangkat keras:

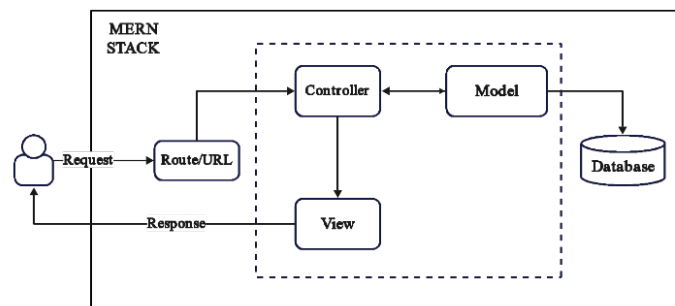
1. *Processor* x86 atau ARM minimal 2 GHz dual-core
2. *RAM* minimum 4GB
3. *Storage* minimal SSD dengan 20 GB ruang yang tersedia (tergantung pada ukuran proyek dan *database*).
4. *Operating System*:
 - Windows 10 atau lebih baru.
 - macOS 10.13 atau lebih baru.
 - Linux modern seperti Ubuntu 18.04 atau lebih baru.

Kebutuhan perangkat Lunak:

1. Terinstal *Node Js*
2. *Database MongoDB*
3. *Text Editor* atau *IDE*

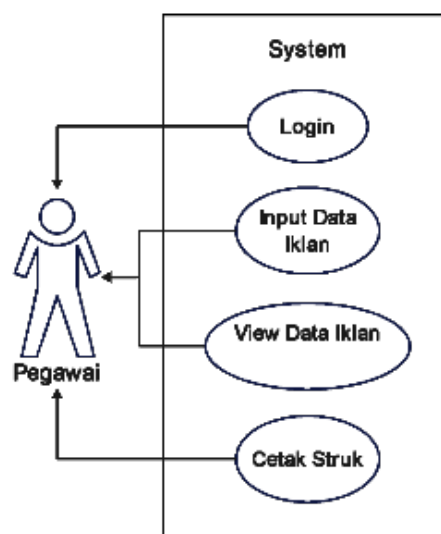
B. ARSITEKTUR SISTEM

Arsitektur sistem yang dibuat pada awal perancangan sistem digunakan untuk menjelaskan alur data dari system. Arsitektur yang digunakan dalam membangun aplikasi Web Manajemen Iklan Radio dan Videotron adalah arsitektur *Model View Controller (MVC)*. Dalam pembuatan website, *MVC* merupakan sebuah pola desain arsitektur yang memisahkan kode menjadi tiga bagian. Ketiga bagian tersebut adalah *model*, *view*, dan *controller*. *Model* merupakan bagian yang bertugas untuk mengatur dan mengelola data serta berhubungan langsung dengan *database*. Di dalam aplikasi web Manajemen Iklan model diberi nama sebagai *User* dan *Receipt*. Pada bagian *User* dan *Receipt* berisikan bagian kode yang mengatur proses *database*. *View* merupakan bagian yang bertugas untuk menyajikan desain tampilan dan informasi yang akan berinteraksi langsung dengan *user* dalam bentuk *Graphical User Interface (GUI)*. Untuk tampilan *User interface* aplikasi Manajemen Iklan ini menggunakan *ReactJs*, dengan menggunakan *ReactJs* membantu memudahkan pekerjaan dalam hal tampilan. *Controller* adalah komponen yang bertugas untuk mengatur dan menghubungkan model dan view dalam setiap proses permintaan dari pengguna, sehingga keduanya dapat saling terhubung. Di dalam Aplikasi Web Manajemen Iklan, *controller* merupakan bagian yang menampung semua *route* proses yang menghubungkan antara user dengan tampilan maupun tampilan dengan *database*.



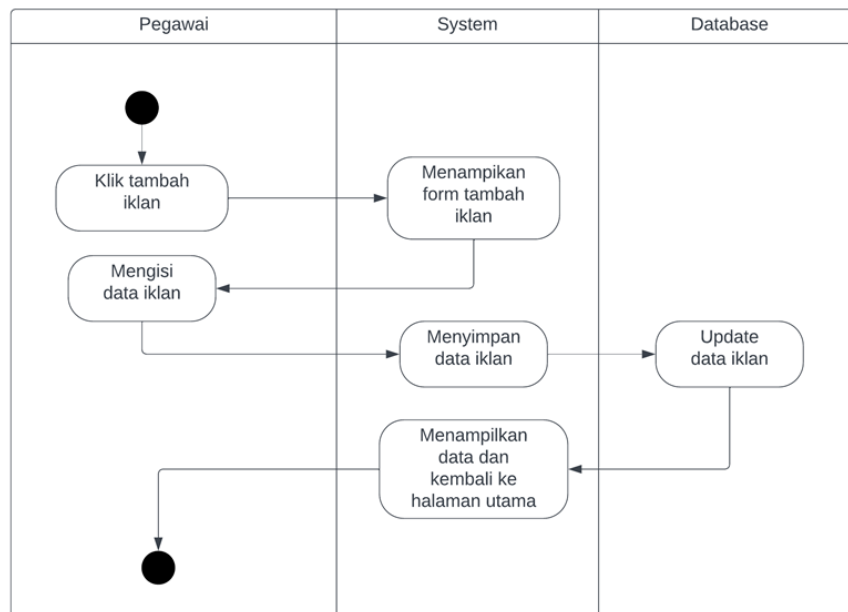
Gambar 2. Arsitektur Sistem

Berdasarkan Gambar 2, dalam proses penginputan data iklan, hal pertama yang dilakukan adalah *user* melakukan *request* melalui *URL*. Request tersebut diterima oleh *controller* sehingga akan menampilkan halaman tambah data. Pembuatan laporan atau struk Iklan akan dilakukan oleh *user* setelah selesai memasukkan data iklan pada halaman tersebut. Setelah semua data tervalidasi, system akan melakukan *request* kepada *controller*. *Controller* akan menerima permintaan tersebut dan akan dikirimkan ke bagian *model* untuk diproses. Kemudian, *model* akan mencari dan memproses data yang diminta dalam *database*. Setelah data ditemukan dan diproses, *model* akan mengirimkan data tersebut ke *controller*. *Controller* mengambil hasil olahan tersebut dan membuatkan file yang akan dikirimkan di bagian *view* untuk di tampilan dan diunduh kepada *user*.



Gambar 2. Use Case Diagram

Gambar 3 menunjukkan *Use Case Diagram* yang digunakan dalam penelitian ini, di mana aktor dalam *Use Case* tersebut adalah *User*. *User* merupakan *Pegawai* yang menjalankan aplikasi iklan ini. Pengguna dapat melakukan *login* ke aplikasi untuk menginput data Iklan yang masuk. Setelah menginputkan data iklan akan ditampilkan data yang sudah diinputkan tadi dan dapat mencetak struk Iklan tersebut.



Gambar 4. Activity Diagram

Dapat dilihat pada gambar 4 merupakan *activity* diagram yang menggambarkan aktivitas yang dilakukan *user* pada saat menambahkan data iklan. Untuk menambahkan data iklan *user* memilih icon tambah iklan yang ada di halaman utama. *User* diwajibkan mengisi semua data iklan dengan benar. Apabila semua data terisi dan benar maka sistem akan meneruskan proses ke *database* untuk di *update*. Setelah data sudah ditambahkan ke *database* sistem aplikasi akan kembali lagi ke halaman utama.

C. IMPLEMENTASI

Bahasa pemrograman *JavaScript* dan *MERN Stack* digunakan untuk mengimplementasikan hasil dari perancangan sistem yang telah dibuat. Hasil dari penelitian yang dilakukan adalah aplikasi berbasis website yang digunakan untuk manajemen data iklan yang masuk.

Kode Program 1. `./bin/www`

```

const serverInstance = http.createServer(app);
serverInstance.listen(PORT, HOST.toString());

serverInstance.on("error", (error) => {
  if (error["syscall"] !== "listen") {
    throw error;
  }
  const bind = typeof PORT === "string" ? `PIPE ${PORT}` : `PORT ${PORT}`;

  if (error.code === "EACCES") {
    Logger.error(`${chalk.bold(bind)} requires elevated privileges`);
    process.exit(1);
  } else if (error.code === "EADDRINUSE") {
    Logger.error(`${chalk.bold(bind)} is already in use`);
    process.exit(1);
  } else {
    throw error;
  }
});

serverInstance.on("listening", async () => {
  await connect({
    stopFailed: true,
    logger: true,
  }).then(async () => {
    await importJSON(["./documents/categories.json"]);
  });
  const addr = serverInstance.address();
  const bind =
    process.env.NODE_ENV === "DEV"
      ? `http://localhost:${addr.port}/`
      : typeof addr === "string"
        ? "PIPE " + addr
        : "PORT " + addr.port;
  Logger.info(`Server is running on: ${chalk.blue.underline(bind)}`);
});
    
```

Pada kode program 1 dalam pembuatan web Manajemen Iklan, *Server HTTP* dari aplikasi dijalankan pada file `.bin/www` yang mengatur error, mengelola koneksi ke layanan tambahan, dan memberi tahu *user* Ketika *server* berhasil dijalankan.

Kode Program 2. `app.js`

```
const app = express();

app.use(requestTimeout);
app.use(checkHeader);
app.use(express.json());
app.use("/api", require("./routes/api"));
app.all("/*", (_, res) => res.sendStatus(404)); app.use(errorResponder);
```

Kode program 2 menginisialisasi *Express* dengan *Middleware* untuk menangani batas waktu *request*, memeriksa *header*, dan *parsing JSON*. *Rute API* diarahkan ke file rute terpisah dengan prefix `/api`, sementara rute yang tidak dikenali akan mendapatkan status 404. *Middleware* penanganan kesalahan digunakan untuk menangani kesalahan yang terjadi.

Kode Program 3. `ReceiptController.js`

```
const checkReceipt = await Receipt.findOne({
  receipt_number: req.body.receipt_number,
}).select("-_id -__v");
if (checkReceipt) {
  const message = Locale.format("ERR_GENERIC_409", [
    "receipt_number",
  ]);
  throwHttpError(409, message, {
    matched_document: checkReceipt,
  });
}

const allowedCategories = await Category.arrayCategories();

const categoryRules = Joi.object({
  ads_category: Joi.string().valid(...allowedCategories),
});

await validate(categoryRules, matchData, {
  allowUnknown: true,
}).catch((error) => {
  throwHttpError(...error);
});

const newReceipt = new Receipt(matchData);
await newReceipt.save();

let receipt = newReceipt.toObject({
  versionKey: false,
  transform: true,
});

receipt.period = Object.fromEntries(receipt.period);
delete receipt._id;

return res.status(201).json({
  success: true,
  message: "Receipt created successfully",
  created_document: receipt,
});
```

Kode program 3 `'ReceiptController'` mengelola operasi *CRUD* untuk *Receipt* dengan menggunakan validasi data, penanganan kesalahan, dan operasi *database*. *Code line* pertama memeriksa apakah yang diberikan sudah ada dalam basis data. Jika nomor struk sudah ada, maka akan mengirimkan pesan kesalahan dengan status HTTP 409 (*Conflict*), menandakan bahwa data yang dimasukkan sudah ada. Validasi kategori iklan mengambil daftar kategori iklan yang diperbolehkan dari `'Category.arrayCategories()'` dan memvalidasi `'ads_category'` di dalam *request body* untuk memastikan bahwa nilainya sesuai dengan kategori yang diizinkan. Setelah Validasi berhasil, membuat objek *Receipt* baru dan menyimpannya ke *database*. Data `'Receipt'` kemudian dikonversi menjadi objek JavaScript dengan beberapa transformasi, seperti menghapus `'_id'` dan mengubah field *period*. Setelah itu mengirimkan respons HTTP dengan status 201 (*Created*) yang menyertakan pesan sukses dan data dokumen yang baru dibuat.

IklanApp

Silahkan masuk untuk melanjutkan

Email Address

Password

☐ Remember me [Forgot password?](#)

Sign in

Gambar 5 Tampilan Login

Gambar 5 merupakan tampilan dari halaman login dalam aplikasi web manajemen iklan. *User* memasukan *Email* dan *Password* yang benar untuk bisa masuk ke dalam halaman utama website.

Kode Program 5. Kode AuthController.js

```
async POST_AUTHORIZE(req, res, next) {
  try {
    const validationRules =
      AuthSchema.POST_AUTHORIZATION_BODY();
    const matchData = ValidationService.validate(validationRules,
      req.body);

    const { email, password } = matchData;
    let user = await User.findOne({ email }, "-__v");

    if (!user) return res.status(400).json({ message: "Invalid
      email, please check again" });

    const isMatch = await user.matchPassword(password);
    if (!isMatch) return res.status(400).json({ message: "Invalid
      password" });

    const staleRefreshToken = req.signedCookies[process.env.APP_COOKIE_REFRESH_TOKEN_NAME];
    if (staleRefreshToken) await AuthService.revokeToken-
      JWT(staleRefreshToken);

    const accToken = await AuthService.generateTokenJWT("ACCESS",
      user);
    const refToken = await AuthService.generateToken-
      JWT("REFRESH", user, { hash: true, save: true });

    const cookieExpires = {
      accToken: convertToMillisec-
        onds(AuthService.TOKEN.expiresIn),
      refToken: convertToMillisec-
        onds(AuthService.TOKEN_REFRESH.expiresIn),
    };

    res.cookie("acc_token", accToken, {
      sameSite: "strict",
      signed: true,
      secure: productionEnvironment,
      httpOnly: true,
      maxAge: cookieExpires.accessToken,
    });

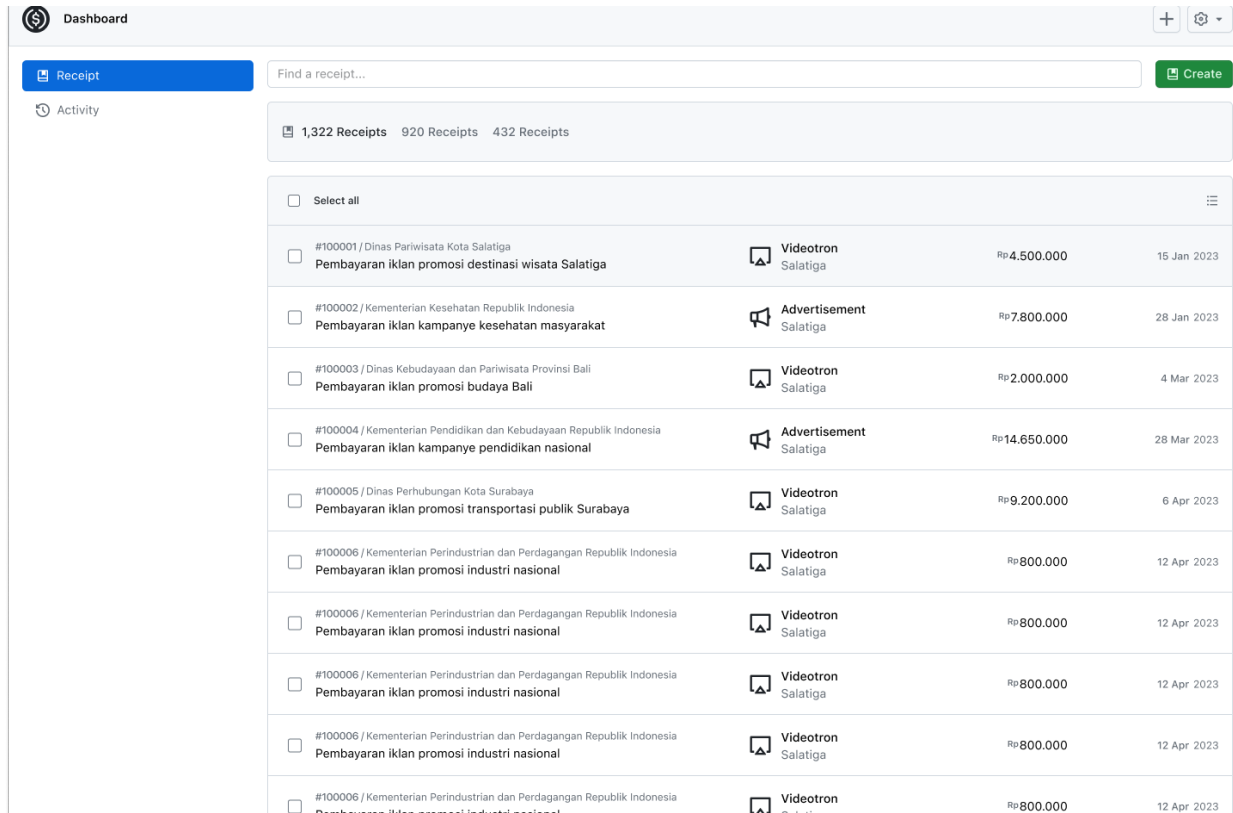
    res.cookie("ref_token", refToken, {
      sameSite: "strict",
      signed: true,
      secure: productionEnvironment,
      httpOnly: true,
      maxAge: cookieExpires.refToken,
    });
  }
}
```

```

        return res.sendStatus(201);
    } catch (error) {
        return next(error);
    }
}

```

Kode program 5 Metode POST_AUTHORIZE ini adalah bagian penting dalam proses otorisasi pengguna di aplikasi web. Fungsinya mencakup validasi kredensial pengguna, pengelolaan token *JWT*, dan pengaturan *cookie* yang aman untuk sesi pengguna. Jika otorisasi berhasil, pengguna mendapatkan *access* token dan *refresh* token yang memungkinkan mereka untuk tetap terotentikasi di aplikasi. Jika ada masalah dengan kredensial atau proses otorisasi, metode ini akan mengembalikan pesan kesalahan yang sesuai. Keamanan adalah perhatian utama, dengan berbagai pengamanan yang diterapkan pada pengelolaan token dan *cookie*.



The screenshot shows a dashboard with a sidebar menu containing 'Receipt' and 'Activity'. The main area displays a table of receipts with columns for selection, receipt number, description, type, amount, and date. A 'Create' button is visible in the top right corner.

	Find a receipt...	Create
1,322 Receipts 920 Receipts 432 Receipts		
☐ Select all		
☐ #100001 / Dinas Pariwisata Kota Salatiga Pembayaran iklan promosi destinasi wisata Salatiga	Videotron Salatiga	Rp4,500.000 15 Jan 2023
☐ #100002 / Kementerian Kesehatan Republik Indonesia Pembayaran iklan kampanye kesehatan masyarakat	Advertisement Salatiga	Rp7,800.000 28 Jan 2023
☐ #100003 / Dinas Kebudayaan dan Pariwisata Provinsi Bali Pembayaran iklan promosi budaya Bali	Videotron Salatiga	Rp2,000.000 4 Mar 2023
☐ #100004 / Kementerian Pendidikan dan Kebudayaan Republik Indonesia Pembayaran iklan kampanye pendidikan nasional	Advertisement Salatiga	Rp14,650.000 28 Mar 2023
☐ #100005 / Dinas Perhubungan Kota Surabaya Pembayaran iklan promosi transportasi publik Surabaya	Videotron Salatiga	Rp9,200.000 6 Apr 2023
☐ #100006 / Kementerian Perindustrian dan Perdagangan Republik Indonesia Pembayaran iklan promosi industri nasional	Videotron Salatiga	Rp800.000 12 Apr 2023
☐ #100006 / Kementerian Perindustrian dan Perdagangan Republik Indonesia Pembayaran iklan promosi industri nasional	Videotron Salatiga	Rp800.000 12 Apr 2023
☐ #100006 / Kementerian Perindustrian dan Perdagangan Republik Indonesia Pembayaran iklan promosi industri nasional	Videotron Salatiga	Rp800.000 12 Apr 2023
☐ #100006 / Kementerian Perindustrian dan Perdagangan Republik Indonesia Pembayaran iklan promosi industri nasional	Videotron Salatiga	Rp800.000 12 Apr 2023
☐ #100006 / Kementerian Perindustrian dan Perdagangan Republik Indonesia Pembayaran iklan promosi industri nasional	Videotron Salatiga	Rp800.000 12 Apr 2023

Gambar 6 Halaman Utama

Pada Gambar 6 merupakan tampilan antarmuka dari sebuah *dashboard* aplikasi manajemen iklan, yang menampilkan daftar kuitansi terkait pembayaran iklan radio dan videotron. Di sebelah kiri terdapat menu untuk mengakses "*Receipts*" dan "*Activity*", serta kolom pencarian yang memungkinkan pengguna mencari kuitansi berdasarkan kata kunci atau nomor tertentu. Bagian utama layar menampilkan ringkasan dari 1,322 kuitansi yang terdaftar, dengan rincian masing-masing kuitansi termasuk nomor, instansi terkait, jenis iklan (Videotron atau Advertisement), lokasi penayangan, jumlah pembayaran, dan tanggal transaksi. Tombol "*Create*" di pojok kanan atas memungkinkan pengguna untuk menambahkan kuitansi baru.

Membuat catatan kuitansi baru

Silakan isi formulir di bawah ini untuk menyimpan catatan kuitansi baru. Kami akan menyimpan informasi berdasarkan data yang Anda masukkan.

Kolom yang wajib diisi ditandai dengan tanda asterisk (*)

No Kuitansi *

Nomor kuitansi harus unik. Untuk mempercepat proses, klik tombol  untuk menghasilkan nomor kuitansi sesuai dengan urutan data.

Instansi *

Terbilang (Jumlah) *

Guna Membayar *

Tanggal Mulai *

Date: 05 Month: December Year: 2023

Tanggal Selesai

Date: 05 Month: December Year: 2023

☒  Radio

Kuitansi ini diterbitkan untuk tujuan pembayaran iklan yang telah diiklankan.

☐  Videotron

Kuitansi ini diterbitkan untuk layanan penayangan konten di videotron.

☐ Pastikan data yang Anda masukkan di atas benar. Jika terdapat kesalahan, Anda dapat memperbaruinya nanti.

Tambahkan catatan

Gambar 7 Halaman Tambah data iklan

Pada Gambar 7 merupakan tampilan dari halaman tambah data iklan dalam aplikasi web manajemen iklan. *User* memasukan nama instansi yang akan menggunakan jasa iklan. Memasukkan jumlah uang yang dibayarkan untuk iklan. Masukkan keterangan konten iklan apa yang akan diiklankan nanti. Memilih periode berapa lama waktu untuk pengiklanan, dan memilih kategori iklan videotron atau iklan radio yang akan digunakan.

Kode Program 4. Penggunaan AJAX

```
const result = fetch("https://localhost:3000/api/receipts?page=1&offset=1")
  .then((response) => response.json())
  .then((data) => render(data))
  .catch((error) => alert(error));
```

Kode program 4 ini bertujuan untuk mengambil data dari *API* dengan menggunakan *HTTP GET*. Setelah data diterima dan dikonversi dari *JSON* ke objek *JavaScript*, data tersebut diproses oleh fungsi `render`. Jika terjadi kesalahan selama permintaan atau pemrosesan data, kesalahan tersebut akan ditangani dengan menampilkan pesan kesalahan kepada *user*.

KOTA SALATIGA

Jalan Merpati No. 123, RT 02/RW 05, Kelurahan Sukamaju, Kecamatan Cipayung
Nomor telepon / kontak: 0812-3456-7890
Email: iklansalatiga@mail.com

Kuitansi

No. 02122001-000001

Instansi : Dinas Perhubungan

Terbilang : Rp 1.000.000

Guna Membayar : Iklan Sosialisasi

Periode : 01-05-2020 s/d 01-05-2021

Kategori : Videotron

Uang Sebanyak : Satu Juta Rupiah

Salatiga, 20 September 2021

Gambar 8 Contoh Struk Iklan

Pada Gambar 8 menampilkan Struk Iklan yang telah dibuat oleh *user*. Data Struk tersebut diberikan kepada instansi yang menggunakan layanan iklan ini, dan data yang masuk ke dalam *database* aplikasi memudahkan pegawai dalam melakukan pengecekan data yang masuk tanpa harus menggunakan cara yang lama.

D. PENGUJIAN

Pada penelitian ini, pengujian sistem dilakukan menggunakan metode *black-box*. Uji ini dilakukan pada tahap akhir pengembangan perangkat lunak untuk memastikan apakah aplikasi berfungsi sesuai dengan rencana yang telah ditentukan. Sehingga, dalam pengujian ini yang akan mengelola dapat mengetahui dan menentukan apakah sistem informasi yang dibuat telah berjalan baik atau tidak.

1. Pengujian Keamanan

Pengujian keamanan XSS (*Cross-Site Scripting*) melalui metode *Black-box* dilakukan untuk mendeteksi dan mencegah penyuntikan skrip jahat ke dalam aplikasi web, guna melindungi data pengguna dan mencegah eksekusi skrip berbahaya di *browser*. Hasil pengujian dapat dilihat pada Tabel 1.

TABEL 1
 PENGUJIAN XSS MELALUI METODE BLACK BOX

No	Modul	Fungsi yang Diuji	Skenario Uji	Data Uji	Hasil yang Diharapkan	Status
1	Receipt Management	Input Data Struk Iklan	Pengguna mengirimkan input yang mengandung script melalui form tambah iklan.	title: "<script>alert('XSS')</script>", content: "Normal content"	Script XSS harus di-sanitasi, tidak dieksekusi, dan disimpan dalam bentuk teks.	Passed
2	Receipt Management	Input Data Struk Iklan	Pengguna mengirimkan input XSS melalui API untuk menambah struk iklan baru.	title: "", content: "Normal content"	Input XSS harus di-sanitasi oleh backend sebelum disimpan ke database.	Passed
3	Receipt Management	Menampilkan Data Struk Iklan	Pengguna melihat data struk iklan yang sebelumnya diinput dengan script XSS.	Data iklan dengan script XSS yang disimpan sebelumnya.	Script XSS tidak dieksekusi pada halaman web, ditampilkan sebagai teks biasa.	Passed
4	User Management	Input Data Pengguna	Pengguna mencoba memasukkan script XSS melalui form pendaftaran atau update profil.	email: "<script>alert('XSS')</script>", name: "Normal Name"	Input script XSS harus di-sanitasi dan ditampilkan sebagai teks biasa di UI.	Passed
5	Dashboard	Menampilkan Data Iklan di Dashboard	Pengguna mengakses dashboard setelah login dan mencoba script XSS di UI.	Iklan yang mengandung script XSS di bagian title atau content.	Script XSS tidak boleh dieksekusi di dashboard, tampil sebagai teks biasa.	Passed
6	Search Functionality	Fungsi Pencarian di Dashboard	Pengguna melakukan pencarian yang menghasilkan iklan dengan script XSS.	Pencarian dengan hasil mengandung script XSS.	Hasil pencarian tidak mengeksekusi script XSS, tampil sebagai teks biasa.	Passed

2. Pengujian Integrasi

Pengujian integrasi dengan metode *black box* adalah teknik pengujian di mana komponen atau sistem yang diintegrasikan diuji berdasarkan input dan output tanpa mengetahui detail internal atau implementasi kode, sehingga fokusnya pada bagaimana komponen bekerja bersama dalam skenario penggunaan sebenarnya. Hasil pengujian dapat dilihat pada Tabel 2.

TABEL 2
PENGUJIAN INTEGRASI MELALUI METODE BLACK BOX

No	Modul	Fungsi yang Diuji	Skenario Uji	Data Uji	Hasil yang Diharapkan	Status
1	Authentication	Login Pengguna	Pengguna mengirimkan data login melalui endpoint API /auth.authorize.	email: user@example.com, password: 12345678	Login berhasil, respons berisi token JWT.	Passed
2	Dashboard	Menampilkan Data Struk Iklan	Pengguna mengakses dashboard setelah login.	JWT token valid	Data struk iklan ditampilkan sesuai dengan data yang tersimpan di database.	Passed
3	Receipt Management	Menambahkan Struk Iklan Baru	Pengguna mengirimkan data struk iklan melalui endpoint API /receipts.	Instansi: "Dinas Pariwisata Kota Salatiga" Jumlah: 4.500.000 Deskripsi: "Pembayaran iklan promosi destinasi wisata salatiga" Periode Iklan: 15/01/2024 - 31/08/2024 Kategori: "Videotron"	Struk iklan baru berhasil ditambahkan ke database, respons berisi data iklan baru.	Passed
4	Receipt Management	Mengedit Struk Iklan yang Ada	Pengguna mengirimkan data edit struk iklan melalui endpoint API /receipts/:id.	id: 20220112-000001, Deskripsi: "Berita kehilangan dompet"	Struk iklan berhasil diperbarui di database.	Passed
5	Receipt Management	Menghapus Struk Iklan yang Ada	Pengguna mengirimkan permintaan hapus melalui endpoint API /receipts/:id.	id: 20220112-000001	Struk iklan berhasil dihapus dari database.	Passed
6	Reporting	Menghasilkan Laporan Struk Iklan	Pengguna mengakses endpoint API /reports/ads.	id: 20220112-000005	Laporan struk iklan berhasil dihasilkan dan dikirimkan sebagai file PDF.	Passed

3. Pengujian *System Usability Scale*

System Usability Scale (SUS) adalah metode kuesioner yang dirancang untuk menilai kemudahan penggunaan suatu sistem dari sudut pandang subjektif pengguna. Pengujian *usability* ini melibatkan sepuluh skala untuk memberikan gambaran umum mengenai pengalaman penggunaan sistem tersebut. Untuk hasil pengujian dapat dilihat pada Tabel 3 dan juga Tabel 4.

TABEL 3
Skor Asli SUS

Resp.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	4	3	4	2	3	2	4	2	4	2
2	3	2	4	3	4	2	3	2	4	2
3	5	2	4	3	4	1	5	2	4	1
4	4	2	3	1	4	2	3	1	5	2
5	4	2	5	2	5	2	5	1	4	2
6	5	1	4	2	5	1	4	1	3	2
7	3	3	3	3	4	1	5	3	5	1
8	5	1	4	1	3	2	3	1	5	3

TABEL 4
Skor Hasil Hitung SUS

Resp.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai (jumlah x2,5)
1	3	2	3	3	2	3	3	3	3	3	28	70
2	2	3	3	2	3	3	2	3	3	3	27	67.5
3	4	3	3	2	3	4	4	3	3	4	33	82.5
4	3	3	2	4	3	3	2	4	4	3	31	77.5

5	3	3	4	3	4	3	4	4	3	3	34	85
6	4	4	3	3	4	4	3	4	2	3	34	85
7	2	2	2	2	3	4	4	2	4	4	29	72.5
8	4	4	3	4	2	3	2	4	4	2	32	80
Rata-Rata												77.5

Presentase kelayakan dari uji usability adalah 77,5%, sehingga dapat disimpulkan pada aspek usability Website Manajemen Iklan Radio dan Videotron yang dibuat mempunyai nilai “Cukup Baik”.

PEMBAHASAN

Aplikasi yang dikembangkan ini menawarkan inovasi berupa integrasi manajemen iklan radio dan videotron yang lebih rinci dan terstruktur dibandingkan solusi manual yang sebelumnya digunakan. Dengan memanfaatkan teknologi *MERN Stack* (*MongoDB*, *Express.js*, *React.js*, dan *Node.js*), aplikasi ini memungkinkan otomatisasi pengelolaan data iklan yang sebelumnya dilakukan secara manual menggunakan spreadsheet, yang sering kali menyebabkan ketidaksesuaian data. Penerapan arsitektur *Model-View-Controller* (*MVC*) dalam aplikasi ini juga memberikan fleksibilitas yang lebih tinggi dalam pengelolaan data dan pengembangan fitur baru di masa depan.

Penelitian ini memiliki keunggulan dibandingkan dengan penelitian sebelumnya yang hanya fokus pada manajemen iklan radio. Penelitian sebelumnya seperti yang dilakukan pada aplikasi pencatatan iklan radio menggunakan *framework Laravel*, meskipun efektif, masih terbatas pada satu media periklanan dan tidak menggunakan pendekatan *full-stack* yang terintegrasi seperti *MERN Stack*[16].

Inovasi lain dalam penelitian ini adalah kemampuan sistem untuk mengakomodasi perubahan kebutuhan pengguna di masa depan melalui struktur kode yang lebih mudah dikelola, sesuatu yang tidak dijelaskan secara rinci dalam penelitian-penelitian sebelumnya.

Aplikasi ini juga dibandingkan dengan penelitian terkait yang menggunakan *MERN Stack* dalam konteks lain, seperti aplikasi manajemen layanan bengkel dan penjualan *top-up game online*. Hasil penelitian menunjukkan bahwa penerapan teknologi *MERN Stack* tidak hanya meningkatkan efisiensi pengelolaan data, tetapi juga memberikan pengalaman pengguna yang lebih responsif dan adaptif terhadap perubahan kebutuhan, yang mana hal ini juga tercermin dalam penelitian ini[13].

IV. KESIMPULAN

Penggunaan iklan radio dan videotron di Kota Salatiga memiliki peran penting dalam mendukung strategi periklanan modern, memberikan manfaat signifikan bagi pelaku bisnis dan industri periklanan, serta berkontribusi dalam membentuk identitas kota. Namun, manajemen data iklan yang masih dilakukan secara manual telah menimbulkan masalah, seperti ketidaksesuaian data dan ketidakefisienan dalam pengelolaan informasi. Untuk mengatasi permasalahan tersebut, pengembangan aplikasi web manajemen iklan radio dan videotron menggunakan teknologi *MERN Stack* menjadi solusi yang relevan. Teknologi ini menawarkan keunggulan dalam hal keamanan, skalabilitas, dan kemudahan pengelolaan kode, yang dapat meningkatkan efektivitas manajemen data iklan di Kota Salatiga. Pengembangan selanjutnya diharapkan dapat memperhatikan pada bagian responsivitas tampilan aplikasi agar lebih baik lagi kedepannya. Selain itu, pengamanan sistem perlu ditingkatkan dengan menerapkan enkripsi data yang lebih kuat, memastikan perlindungan terhadap serangan seperti *cross-site scripting* (*XSS*), serta melakukan audit keamanan secara berkala untuk mendeteksi dan mengatasi potensi celah keamanan. Penggunaan autentikasi multifaktor juga bisa dipertimbangkan untuk memberikan lapisan keamanan tambahan bagi pengguna.

V. DAFTAR PUSTAKA

- [1] P. Saniscara, “Media Luar Ruang Serta Dampaknya Terhadap Kualitas Dan Lingkungan Hidup Masyarakat,” *J. Dimens. DKV Seni Rupa dan Desain*, vol. 7, no. 2, pp. 163–174, 2022, doi: 10.25105/jdd.v7i2.15168.
- [2] RILA SUCI ANDHARA, *Skripsi Gabungan*, no. 3747. 2019.
- [3] P. Studi, I. Komunikasi, F. Psikologi, D. A. N. Ilmu, S. Budaya, and U. I. Indonesia, *Strategi Komunikasi Publik Dinas Komunikasi dan Informatika (Diskominfo) Kabupaten Kudus dalam Penanganan Covid-19 Lensa Khoirul Jannah Sonida*, no. 18321177. 2022.
- [4] S. R. Wicaksono, E. S. Krisnadiva, and A. B. Yuliermawan, “Sistem Pendukung Keputusan Pengadaan Perangkat Multimedia Kodam Iv/Dip Semarang Dengan Metode Simple Additive Weighting (Saw),” *J. Ilm. Manajemen, Bisnis dan Kewirausahaan*, vol. 3, no. 2, pp. 240–247, 2023, doi: 10.55606/jurimbik.v3i2.479.
- [5] P. Porter, S. Yang, and X. Xi, “The Design and Implementation of a RESTful IoT Service Using the MERN Stack,” in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, 2019, pp. 140–145. doi: 10.1109/MASSW.2019.00035.
- [6] D. Gunawan *et al.*, “Implementasi MERN Stack pada Pengembangan Sistem Penerimaan Peserta Didik Baru,” *Swabumi*, vol. 11, no. 2, pp. 102–110, 2023, doi: 10.31294/swabumi.v11i2.15965.
- [7] J. V. Sanchit Aggarwal, “Comparative analysis of MEAN stack and MERN stack,” *Int. J. Recent Res. Asp.*, 2018.
- [8] M. Bawane, “A Review on Technologies used in MERN stack,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 1, pp. 479–488, 2022, doi: 10.22214/ijraset.2022.39868.
- [9] V. Raj, “E-Commerce Website Using Mern Stack,” *Interantional J. Sci. Res. Eng. Manag.*, vol. 08, no. 05, pp. 1–5, 2024, doi: 10.55041/ijrsrem34876.
- [10] E. Santosh, Kumar & Shukla, & Dubey, Shivam & Rastogi, Tarun & Srivastava, Nikita & Ijmtst, “Application using MERN Stack,” *Int. J. Mod.*

- Trends Sci. Technol.*, 2022.
- [11] A. N. -, J. S. -, A. V. -, J. J. -, and P. K. -, "Learn-It : An E-Learning Web Application Using MERN Stack," *Int. J. Multidiscip. Res.*, vol. 6, no. 1, pp. 1–37, 2024, doi: 10.36948/ijfmr.2024.v06i01.9125.
 - [12] R. Triandy and N. Santoso, "Pengembangan Aplikasi Web Reservasi Paket Wisata menggunakan MERN Stack (Studi Kasus: Zona Tamasya Tour Organizer)," ... *Teknol. Inf. dan Ilmu Komput. e ...*, vol. 4, no. 6, pp. 1616–1624, 2020, [Online]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/7308/3519>
 - [13] M. A. Maulana, H. Haryoko, B. Santoso, and L. Lukman, "Penerapan Teknologi Stack MERN pada Aplikasi Service Manajemen Bengkel Berbasis Web," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1536, 2022, doi: 10.30865/mib.v6i3.4147.
 - [14] R. Wati, Jeprianto, Sudewi, and A. Maseleno, "Perancangan Aplikasi Sistem Penjualan Top-Up Game Online Berbasis Website Memanfaatkan Fitur Mern Fullstack Menggunakan Metode Waterfall," *J. Tek.*, vol. 18, no. 1, pp. 93–104, 2024.
 - [15] S. M. Afif and I. K. D. Nuryana, "Rancang Bangun Sistem Informasi Staycation Berbasis Web Dengan Implementasi Teknologi Mern Stack," *Angew. Chemie Int. Ed. 6(11)*, 951–952., pp. 1–12, 2021, [Online]. Available: <https://ejournal.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/41882>
 - [16] L. Candra, G. Wicaksono, and E. Sedyono, "3442-9937-1-Pb," vol. 11, no. 1, pp. 1–11, 2023.
 - [17] R. Hermiati, A. Asnawati, and I. Kanedi, "Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql," *J. Media Infotama*, vol. 17, no. 1, pp. 54–66, 2021, doi: 10.37676/jmi.v17i1.1317.
 - [18] M. Stauffer, *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. 2023. [Online]. Available: <https://books.google.co.id/books?id=GcqPDwAAQBAJ>
 - [19] Sitanggang Rianto, Urian Dachi Teddy, and Manurung H G Immanuel, "Rancang Bangun Sistem Penjualan Tanaman Hiasberbasis Web Menggunakan Php Dan Mysql," *Tekesnos*, vol. 4, no. 1, pp. 84–90, 2022.
 - [20] D. Rahmawati, A. S. Prabowo, and R. Purwanto, "Implementasi Model Waterfall pada Pengembangan Sistem Informasi Monitoring Prestasi Mahasiswa," *J. Innov. Inf. Technol. Appl.*, vol. 3, no. 1, pp. 82–93, 2021, doi: 10.35970/jinita.v3i1.678.