

DOKUMEN TEKNIS FRONTEND SEALS

Author: Dimas Putra Anhar Wirjo Atmodjo

Team:

System Analyst	Adinda Wahyu Luchmansyahvira
UI UX Designer	Diah Putri Nofianti
Front End Developer	Dimas Putra Anhar Wirjo Atmodjo
Mobile Developer	Naresh Pratista
Back End Developer	Elang Putra Adam

Overview

SEALS adalah sebuah aplikasi pembelajaran Bahasa Inggris adaptif yang dirancang khusus untuk siswa SMK Pariwisata dan Bisnis. Aplikasi ini dikembangkan untuk menyediakan pengalaman belajar yang disesuaikan dengan kebutuhan individu siswa, memanfaatkan teknologi adaptive learning. Dengan SEALS, siswa dapat mengakses konten belajar yang relevan dengan industri pariwisata dan bisnis, melalui perangkat web maupun mobile.

Aplikasi SEALS memungkinkan siswa untuk belajar secara mandiri dengan memilih section dan topic yang sesuai dengan minat dan tingkat kemampuan mereka. Berdasarkan hasil pretest, sistem akan menempatkan siswa pada level yang tepat, dan secara otomatis menyesuaikan materi pembelajaran yang sesuai dengan hasil tersebut. SEALS juga menyediakan berbagai fitur interaktif seperti latihan soal dan ujian untuk mengukur kemajuan belajar siswa.

Aplikasi SEALS memiliki fungsi manajemen bagi admin dan guru untuk mengelola data siswa, materi pembelajaran, serta memantau kemajuan belajar siswa melalui laporan dalam bentuk tabel. Admin memiliki kontrol penuh atas sistem, termasuk kemampuan untuk mengelola data guru, siswa, section, topic, dan pengaturan rule adaptive learning.

SEALS diharapkan dapat meningkatkan keterlibatan siswa dalam belajar Bahasa Inggris dan membantu mereka mencapai kompetensi yang sesuai dengan kebutuhan industri pariwisata

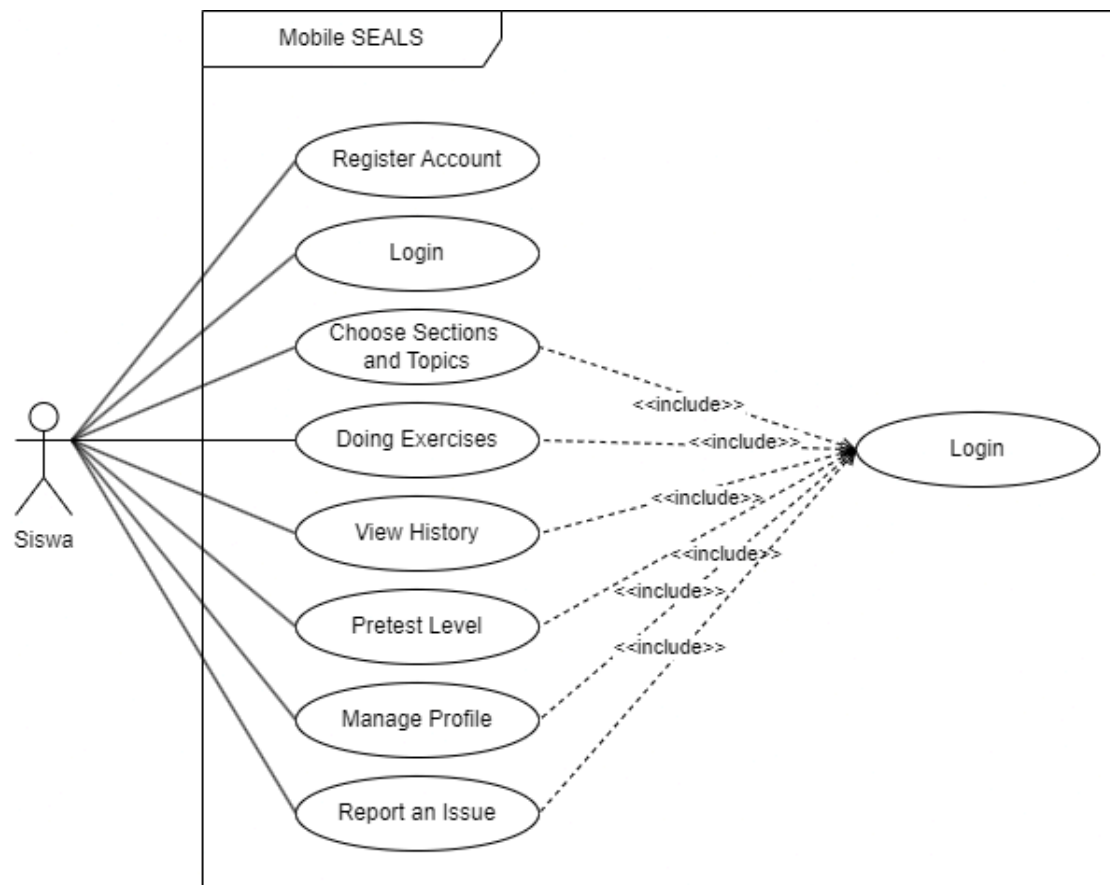
dan bisnis, dengan memberikan fleksibilitas dalam metode pembelajaran yang dapat diakses dimanapun dan kapanpun.

Use Case Diagram

Website SEALS mendukung tiga role user yang di antaranya adalah admin, pengajar, dan siswa sesuai yang sudah digambarkan pada diagram di bawah ini. Admin dapat mengelola berbagai data, mulai dari user yaitu pengajar dan siswa, serta menambahkan bahan pembelajaran dan latihan. Pengajar bisa menambah siswa ke kelas, memantau progres belajar siswa, dan memberikan umpan balik. Sedangkan siswa dapat belajar sesuai yang diinginkan, mengerjakan soal, melihat history pengerjaan latihan, dan juga melaporkan masalah.



Sementara itu, pada mobile hanya role siswa yang akan bisa menggunakan aplikasi tersebut. Jadi, aktivitasnya kurang lebih sama seperti yang telah tercantum di use case diagram website untuk role siswa.



Arsitektur Umum

Aplikasi SEALS dibangun dengan menggunakan pendekatan SPA (Single Page Application) dengan ReactJS sebagai framework utama. Arsitektur ini memungkinkan pengalaman pengguna yang mulus tanpa perlu memuat ulang halaman secara penuh. Aplikasi berinteraksi dengan backend melalui API yang disediakan untuk mendapatkan dan mengirim data.

Komponen Utama

- **Dashboard Siswa**

Siswa bisa melihat section dan topic serta memilih level, di setiap level terdapat material dan exercise.

- **Exercise Siswa,**

Siswa bisa mendapatkan 3 jenis soal, yaitu multiple choice, true/false question, dan matching pairs.

- **Dashboard Guru**

Guru dapat mengelola kelas, monitoring progress siswa, dan memberikan feedback kepada siswa. tidak dapat mengelola section, topic, material, exercise(soal).

- **Dashboard Admin**

Admin adalah aktor yang menegelola user(siswa), guru, dan mengelola section, topic, material, exercise(soal).

Teknologi dan Framework

1. HTML5, CSS3, JavaScript

Teknologi dasar yang digunakan untuk membangun aplikasi ini adalah HTML5, CSS3, dan JavaScript. HTML5 digunakan untuk struktur halaman, CSS3 untuk styling dan layout, serta JavaScript untuk interaktivitas halaman.

2. Framework Front-End Web

ReactJS: Framework utama yang digunakan untuk membangun antarmuka pengguna. ReactJS memfasilitasi pembuatan UI yang efisien dan dapat diperbarui secara dinamis tanpa memuat ulang halaman secara keseluruhan.

3. State Management untuk Web

State aplikasi dikelola menggunakan React Context API, yang memungkinkan berbagi data secara global di antara berbagai komponen tanpa harus melakukan props drilling.

Pengembangan Aplikasi Web

1. Struktur Proyek Web

```
SEALS/  
├─ node_modules  
├─ src/  
│  ├─ assets/  
│  │  ├─ images/  
│  │  │  └─ illustration  
│  │  └─ styles/  
│  │     ├─ admin.css  
│  │     ├─ index.css  
│  │     ├─ teacher.css  
│  │     └─ user.css  
│  └─ components/
```

```
| | | └─ layout/
| | |   └─ admin
| | |   └─ guest
| | |   └─ teacher
| | |   └─ user
| | └─ ui/
| |   └─ adminMessageModal
| |   └─ TablePaginate.jsx
| └─ roles/
|   └─ admin/
|     ├── dashboard
|     ├── manage_classes
|     ├── manage_exercise
|     ├── manage_materials
|     ├── manage_progress
|     ├── manage_reports
|     ├── manage_section
|     ├── manage_students
|     ├── manage_teachers
|     ├── manage_topics
|     ├── setting
|     ├── AdminRoutes.jsx
|     └─ NotFound.jsx
|   └─ guest/
|     ├── auth
|     ├── landing_page
|     ├── NotFound.jsx
|     └─ PublicRoutes.jsx
|   └─ teacher/
|     ├── dashboard
|     ├── feedback
|     ├── manage_classes
|     ├── manage_students
|     ├── monitoring
|     ├── setting
|     ├── NotFound.jsx
|     └─ TeacherRoutes.jsx
|   └─ user/
|     ├── dashboard
|     ├── exercise
|     ├── history
|     ├── learning
|     ├── level
|     ├── material
|     ├── report
|     ├── review
|     ├── setting
|     ├── topic
|     ├── NotFound.jsx
|     └─ userRoutes.jsx
| └─ utils/
```

```

|   |   | — axiosInstance.jsx
|   |   | — Constant.jsx
|   |   | — ProtectedRoute.jsx
|   |   | — SlugContext.jsx
|   | — App.jsx
|   | — main.jsx
| — index.html
| — package.json

```

setiap page pada roles memiliki strungtur service, hooks, views

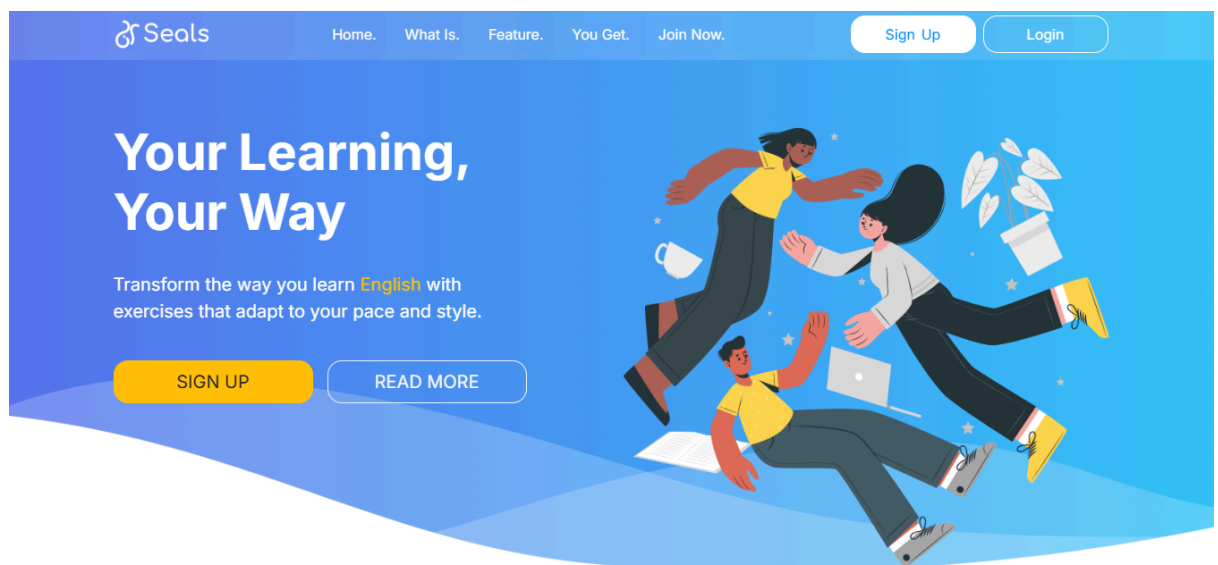
```

roles/
  | — page/
  |   | — services
  |   | — hooks
  |   | — views

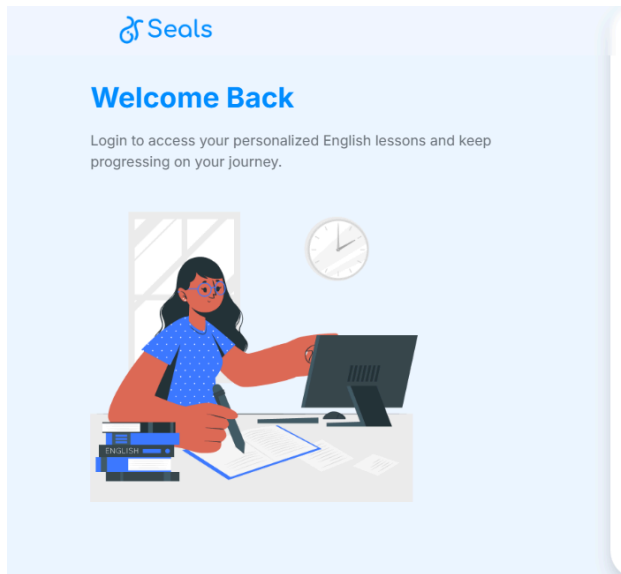
```

2. Komponen UI Web

2.1 guest



landing Page



Login

Let's continue your learning journey.

Email Address *

Password *

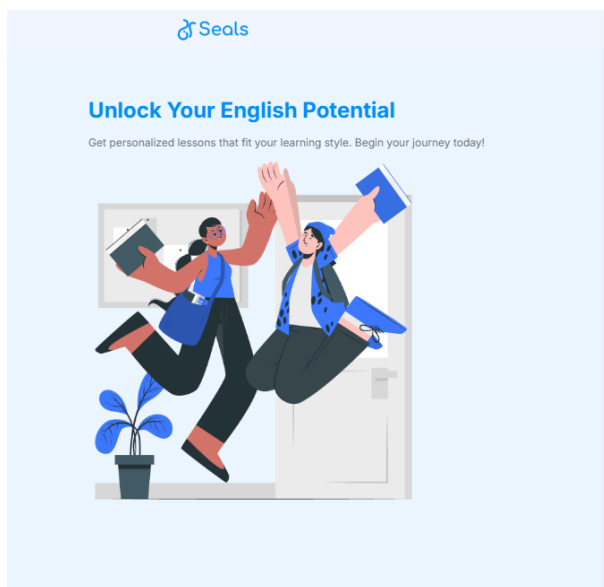
☐ Remember me

[Forgot password?](#)

Log In

Haven't joined us yet? [Sign Up](#) first

Login Page



Sign Up

Let's start by creating your account first.

Full Name*

Email Address*

Role*

☒ Student

☐ Teacher

NISN *

Unique 10-digits ID

Password*

Minimum 8 characters

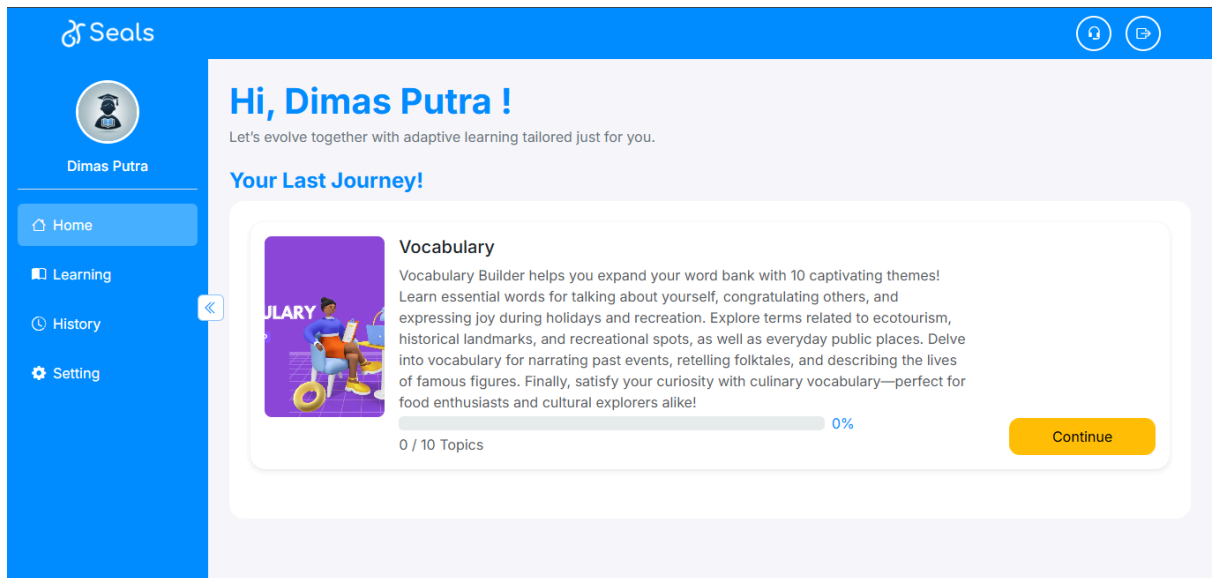
Confirm Password*

Sign Up

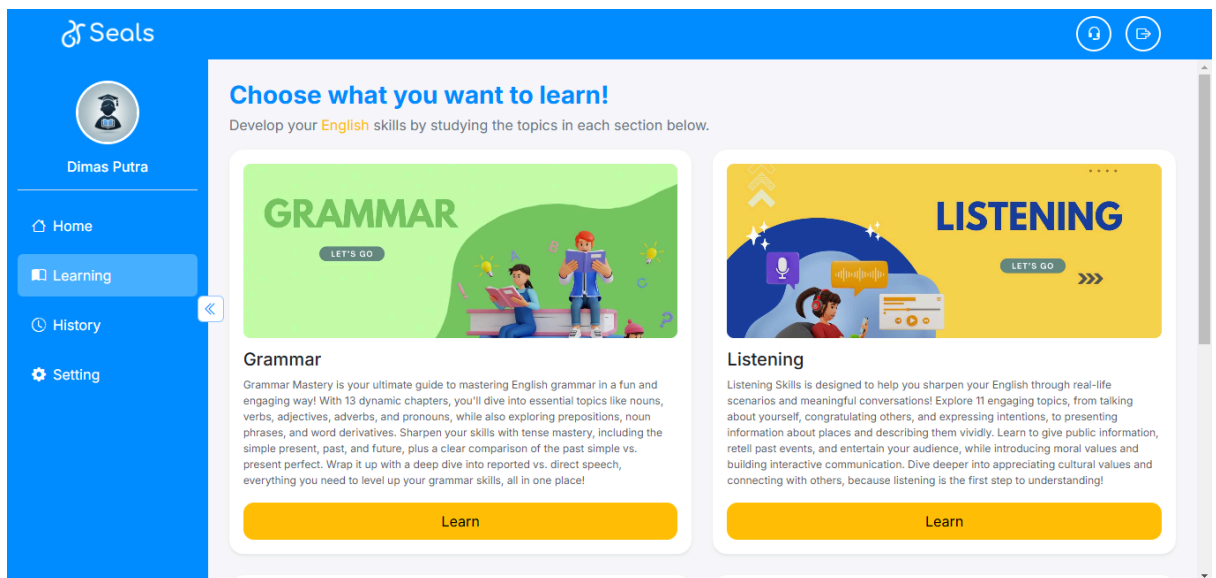
Already have an account? [Log In](#) instead

SignUp Page

2.1 User (siswa)



Home Page



Section Page

Seals

🏠

🔍

🏠

Topic list of Grammar

1. Noun

2. Adjectives

3. Verb

4. Adverb

5. Pronoun

6. Adverb of Time

7. Noun Phrases

8. Derivatives Noun from Verbs

9. Simple Present

10. Simple Past

11. Simple Future

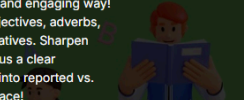
12. Simple Past vs Present Perfect

13. Reported Speech vs Direct Speech

GRAMMAR

📖 Learning / Grammar

Grammar Mastery is your ultimate guide to mastering English grammar in a fun and engaging way! With 13 dynamic chapters, you'll dive into essential topics like nouns, verbs, adjectives, adverbs, and pronouns, while also exploring prepositions, noun phrases, and word derivatives. Sharpen your skills with tense mastery, including the simple present, past, and future, plus a clear comparison of the past simple vs. present perfect. Wrap it up with a deep dive into reported vs. direct speech, everything you need to level up your grammar skills, all in one place!





Topic 1: Noun

Nouns are words that name people, places, things, or ideas, serving as the subjects or objects in a sentence. They are essential for constructing meaningful sentences, as they provide the foundation for describing who or what is involved in an action or state.

Start Topic

 Seals



 Learning / Grammar / Noun

PRETEST

Score 0/100

PRETEST: Introduction to Topic

Learn Now

LEVEL 1

Score 0/100

Beginner

Not Allowed

LEVEL 2

Score 0/100

Advanced Concepts

Not Allowed

LEVEL 3

Score 0/100

LEVEL 4

Score 0/100

Seals

Talking About Self

Learning / Vocabulary / Talking About Self / Pretest

Take test

1. Family

The word "family" refers to a group of people who are related to each other, usually by blood or law. A typical family includes parents and their children, but it can also include extended members like grandparents, aunts, uncles, and cousins. Families can be formed by marriage, adoption, or other close relationships. The family unit is often seen as a source of emotional support and security. Understanding the importance of family in society can help you describe different family dynamics and relationships. Families are typically expected to care for each other, support each other emotionally and financially, and share responsibilities within the household.

2. Describing Family Members

In the sentence "My family is very ____ and supportive," the best word to complete the sentence is "loving." This word describes a family that shows care and affection toward each other. A "loving" family is warm, kind, and expresses love regularly, whether through words, actions, or gestures. Other adjectives that could describe family members include "caring," "close," or "nurturing." Adjectives like "pretty" or "hate" do not make sense in this context, as "pretty" describes physical appearance, and "hate" has a negative meaning that contradicts the idea of being supportive.

3. Hobbies

A "hobby" is an activity that someone does for fun, relaxation, or personal enjoyment during their free time. It is something you do because it interests you, not because you are paid to do it. Common hobbies include reading, painting, playing sports, gardening, and collecting items like stamps or coins. Hobbies are an important part of life because they allow individuals to relax, learn new skills, and meet people with similar interests. It is important to distinguish between hobbies and jobs. While a job is usually something you do to earn money, a hobby is

Material Page

Seals

reading : the-legend-of-naresh - pretest

Pretest

1

2

3

4

5

6

7

8

9

10

Questions 3 of 10

Next Questions →

Match the supporting characters to their roles in the story:

1 Village Elder

2 Merchant

3 Animal Guide

4 Blacksmith

5 Sorceress

1 Teaches Naresh magical skills

2 Provides a key to Naresh's journey

3 Warns Naresh of dangers

4 Guides Naresh with wisdom

5 Sells Naresh important supplies

Exercise Page

3. Routing Web

3.1 Guest

```

<Routes>
  <Route path="/" element={<NotFound/>} />
  <Route path="/unauthorized" element={<Unauthorized />} />

  <Route path="/" element={<HomePage />} />
  <Route path="login" element={<LoginPage />} />
  <Route path="signup" element={<RegisterPage />} />
  <Route path="forgot" element={<ForgotPage />} />
  <Route path="validate-email" element={<ValidateEmail />} />
  <Route path="resetPassword/:token" element={<ResetPass />} />
</Routes>

```

3.2 User

```
<Routes>
  <Route element={<ProtectedRoute role="student" />}>
    <Route path="*" element={<NotFound />} />

    <Route path="/" element={<Navigate to="home" replace />} />
    <Route path="home" element={<Dashboard />} />
    <Route path="module" element={<Learning />} />
    <Route path="setting" element={<Setting />} />
    <Route path="history" element={<ExerciseHistory />} />
    <Route path="review/stdLearning" element={<Review />} />

    <Route path="module/:section" element={<Topic />} />
    <Route path="module/:section/:topic" element={<Level />} />
    <Route path="module/:section/:topic/:level" element={<Navigate to="material" replace />} />
    <Route path="module/:section/:topic/:level/material" element={<Material />} />
    <Route path="module/:section/:topic/:level/exercise" element={<Exercise />} />
    <Route path="module/:section/:topic/:level/result" element={<ExerciseResult />} />
  </Route>
</Routes>
```

3.3 Teacher

```
<Routes>
  <Route element={<ProtectedRoute role="teacher" />}>
    <Route path="*" element={<NotFound />} />

    <Route path="/" element={<Navigate to="home" replace />} />
    <Route path="home" element={<Dashboard />} />
    <Route path="class" element={<ManageClasses />} />
    <Route path="class/d/:classId" element={<ClassDetail />} />
    <Route path="student" element={<ManageStudents />} />

    <Route path="monitoring" element={<Monitoring />} />
    <Route path="monitoring/s/:stdLearning" element={<Review />} />
    <Route path="monitoring/c/:classId" element={<ClassProgress />} />

    <Route path="feedback" element={<Feedback />} />
    <Route path="feedback/s/:progressId" element={<StudentFeedback />} />
    <Route path="feedback/c/:progressId" element={<ClassFeedback />} />
    <Route path="setting" element={<Setting />} />
  </Route>
</Routes>
```

3.4 Admin

```
<Route element={<ProtectedRoute role="admin" />}>
  <Route path="*" element={<NotFound />} />
  <Route path="/" element={<Navigate to="dashboard" replace />} />
  <Route path="dashboard" element={<Dashboard />} />
  <Route path="student" element={<ManageStudents />} />
  <Route path="teacher" element={<ManageTeachers />} />
  <Route path="class" element={<ManageClasses />} />
  <Route path="class/class-detail" element={<ClassDetail />} />
  <Route path="class/class-detail/:classId" element={<ClassDetail />} />
  <Route path="section" element={<ManageSections />} />
  <Route path="topic" element={<ManageTopics />} />
  <Route path="material" element={<ManageMaterials />} />
  <Route path="material/update-material/:materialId" element={<EditorMaterial />} />
  <Route path="exercise" element={<ManageExercises />} />
  <Route path="exercise/old" element={<OldManageExercises />} />
  <Route path="exercise/exercise-detail/:levelId" element={<ExerciseDetail />} />
  <Route path="exercise/update-exercise/:levelId" element={<UpdateExercises />} />
  <Route path="learning-progress" element={<ManageProgress />} />
  <Route path="learning-progress/s/:progressId" element={<StudentProgress />} />
  <Route path="learning-progress/c/:progressId" element={<ClassProgress />} />
  <Route path="report" element={<ManageReports />} />
  <Route path="profile" element={<Setting />} />
</Route>
```

4. Manajemen State Web

Manajemen state dilakukan menggunakan React Context API. Data global seperti status login pengguna, data materi pembelajaran, dan hasil ujian siswa dikelola melalui context yang dapat diakses oleh seluruh aplikasi.

5. Optimasi Performa Web

Untuk mengoptimalkan performa aplikasi, beberapa teknik yang diterapkan meliputi:

- Lazy loading untuk memuat komponen hanya ketika diperlukan.
- Memoization untuk menghindari re-rendering yang tidak perlu.
- Menggunakan `React.PureComponent` untuk komponen yang tidak memerlukan pembaruan state.
-

6. Keamanan Front-End Web

Keamanan aplikasi dijaga dengan menggunakan HTTPS untuk enkripsi data yang dikirimkan antara frontend dan backend. Selain itu, data sensitif seperti token autentikasi disimpan dengan aman menggunakan mekanisme penyimpanan lokal (`localStorage`) dengan teknik penyandian.

Integrasi dan Komunikasi Data

1. API Integration

Aplikasi ini berkomunikasi dengan backend menggunakan HTTP API yang dikonsumsi melalui Axios. API ini digunakan untuk mendapatkan data pengguna, materi pembelajaran, soal ujian, dan hasil ujian siswa.

2. Manajemen State Global

State global aplikasi dikelola menggunakan React Context API, yang memungkinkan data dapat dibagikan di antara berbagai komponen tanpa harus diteruskan melalui props.

3. Sinkronisasi Data antara Web

Data disinkronkan secara real-time antara frontend dan backend menggunakan Axios. Untuk mempercepat pengalaman pengguna, data yang sering diakses disimpan secara lokal menggunakan `localStorage`.

Manajemen Aset

1. Gambar dan Ikon

Gambar dan ikon yang digunakan dalam aplikasi disimpan di direktori [assets/](#) dan dioptimalkan menggunakan teknik kompresi untuk mempercepat waktu pemuatan halaman. Ikon disini menggunakan library ikon dari [bootstrap-icon](#)

2. Font

Aplikasi menggunakan font dari Google Fonts berjenis "[Inter, sans-serif](#)" untuk memastikan tampilan yang konsisten dan estetis di berbagai perangkat.

3. Lokalisasi dan Internasionalisasi

Untuk mendukung penggunaan di berbagai wilayah, aplikasi dilengkapi dengan dukungan untuk lokalisasi dan internasionalisasi, menggunakan pustaka [react-intl](#) untuk menerjemahkan konten sesuai dengan preferensi pengguna.

Keamanan

1. Autentikasi dan Otorisasi

Autentikasi dilakukan dengan menggunakan JSON Web Token (JWT), yang disimpan di [localStorage](#). Hanya pengguna yang telah terautentikasi yang dapat mengakses fitur-fitur tertentu dalam aplikasi.

2. Perlindungan Data

Data yang dikirimkan antara frontend dan backend selalu dienkripsi menggunakan protokol HTTPS untuk melindungi kerahasiaan data pengguna.

3. Keamanan Komunikasi (HTTPS, SSL)

Aplikasi ini menggunakan HTTPS untuk mengamankan komunikasi antara frontend dan backend. SSL diterapkan untuk memastikan integritas dan keamanan data yang dikirimkan melalui jaringan.