

## BAB 4. KEGIATAN KHUSUS DAN PEMBAHASAN

### 4.1 Tinjauan Pustaka

#### 4.1.1 Website

*Website* merupakan kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait di mana masing-masing dihubungkan dengan jaringan-jaringan halaman.

Menurut (Hasibuan & Handoko, 2023), *website* didefinisikan sebagai kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan halaman. Lebih lanjut, keberadaan *website* telah menjadi instrumen vital dalam penyebaran informasi global karena kemampuannya untuk diakses tanpa batasan waktu dan tempat selama terhubung dengan internet.

#### 4.1.2 FrontEnd

Dalam arsitektur pengembangan web, *FrontEnd* merujuk pada bagian dari aplikasi web yang berinteraksi langsung dengan pengguna (*Client side*). Pengembangan *FrontEnd* berfokus pada apa yang dilihat oleh pengguna pada layar mereka.

*FrontEnd* adalah aplikasi web yang dapat berinteraksi dengan pengguna secara langsung. Bagian ini bertanggung jawab atas antarmuka pengguna (*User Interface*) dan pengalaman pengguna (Emirzaki dkk., 2022). Tujuan utama dari pengembangan *front-end* adalah memastikan bahwa ketika pengguna membuka situs, informasi dapat dibaca dengan format yang mudah dipahami dan relevan. Hal ini mencakup struktur desain, perilaku elemen, dan konten yang ditampilkan pada peramban (*browser*).

#### 4.1.3 JavaScript

JavaScript didefinisikan sebagai bahasa pemrograman yang digunakan untuk membuat program agar dokumen HTML yang ditampilkan dalam browser menjadi lebih interaktif dan tidak sekadar statis (Khoirurizal dkk., 2024). JavaScript memungkinkan pengembang untuk mengontrol elemen halaman web secara *real-time*, menangani *event* dari pengguna (seperti klik tombol), dan melakukan komunikasi asinkron dengan server tanpa perlu memuat ulang halaman (*reload*).

#### 4.1.3 Vue.js

Vue.js adalah kerangka kerja (*framework*) JavaScript yang bersifat progresif dan digunakan untuk membangun antarmuka pengguna (*User Interface*). Berbeda dengan *framework* monolitik lainnya, Vue dirancang dari dasar untuk dapat diadopsi secara bertahap.

Kerangka kerja Vue.js menawarkan keunggulan dalam kecepatan pengembangan, kemudahan penggunaan, dan fleksibilitas. Vue memfokuskan dirinya pada *view layer* saja, yang membuatnya sangat mudah diintegrasikan dengan perpustakaan lain atau proyek yang sudah ada (Aryasta, 2022). Vue menggunakan konsep Virtual DOM dan *two-way data binding* (pengikatan data dua arah) yang mempermudah sinkronisasi antara model data dan tampilan (UI).

#### 4.1.4 React.js

React.js adalah pustaka (*library*) JavaScript yang bersifat open-source yang dikembangkan oleh Facebook (Meta) untuk membangun antarmuka pengguna, khususnya untuk aplikasi satu halaman (*Single Page Application*).

Menurut (Iswari, 2021), React.js memiliki pendekatan berbasis komponen (*component-based*) yang memungkinkan pengembang untuk memecah UI yang kompleks menjadi komponen-komponen kecil yang dapat digunakan kembali (*reusable*). React menggunakan mekanisme Virtual DOM untuk meningkatkan performa aplikasi dengan cara meminimalkan manipulasi langsung pada DOM browser yang lambat (Iswari, 2021). Berbeda dengan Vue, React umumnya menerapkan *one-way data flow* (aliran data satu arah) yang membuat logika aplikasi lebih mudah diprediksi dalam skala besar.

#### 4.1.5 Tailwind CSS

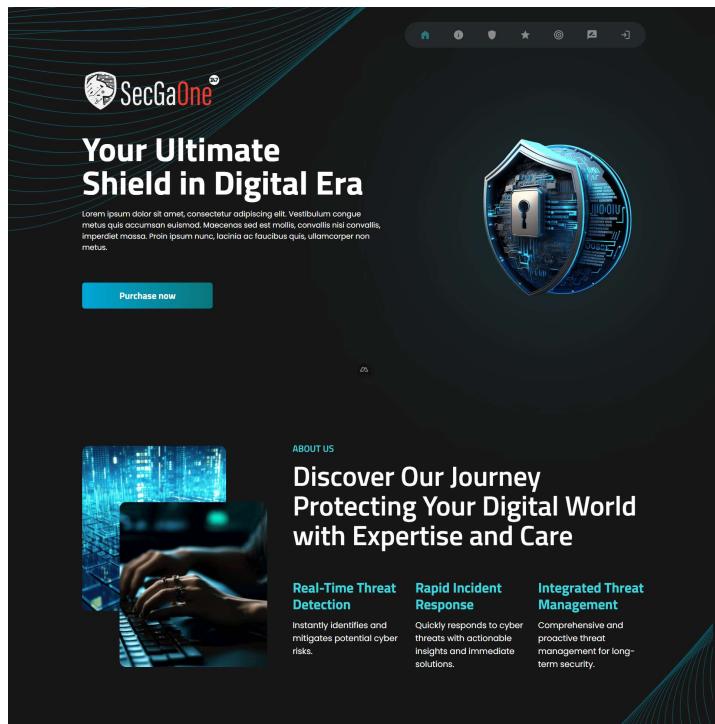
Tailwind CSS adalah kerangka kerja (*framework*) CSS yang mengusung konsep *utility-first*, yang dirancang untuk mempercepat proses pembuatan antarmuka pengguna (*User Interface*) dengan menyediakan sekumpulan kelas utilitas tingkat rendah.

Tailwind memberikan keleluasaan kepada pengembang untuk membangun desain yang sepenuhnya *custom* tanpa harus meninggalkan berkas HTML atau menulis kode CSS dari awal (Mardiana & Junaeti, 2024). Pendekatan ini memungkinkan pengembang untuk memanggil kelas-kelas utilitas langsung pada elemen HTML untuk mengatur tata letak dan gaya.

## 4.2 Hasil Kegiatan

#### 4.2.1 Dashboard Marketing

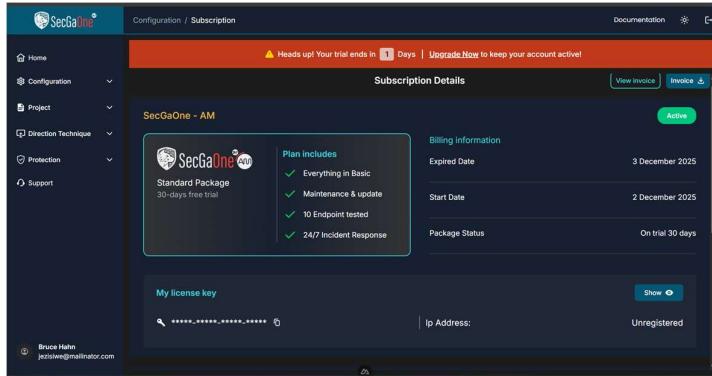
Dashboard Marketing merupakan aplikasi yang dirancang untuk menampilkan informasi terkait berbagai produk yang tersedia dalam platform keamanan siber. Dashboard ini dirancang untuk memudahkan proses registrasi akun, pengajuan akses uji coba (trial), pemantauan status langganan produk, hingga melihat dokumentasi yang diperbarui oleh admin. Dalam pengembangannya, aplikasi ini dibangun menggunakan Nuxt.js dan dipadukan dengan TailwindCSS untuk membangun antarmuka yang responsif, konsisten, dan mudah dikustomisasi.



Gambar 4. 1 Halaman Utama Dashboard Marketing

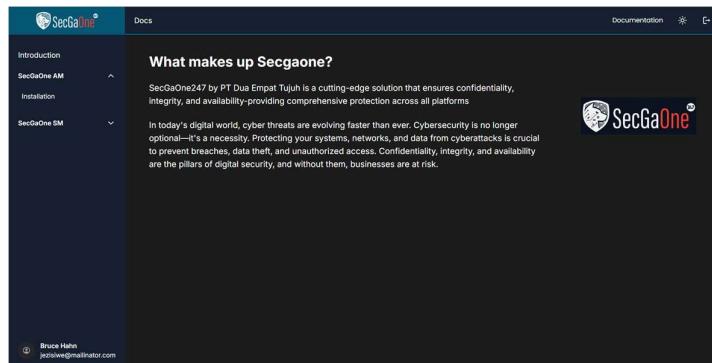
Selain itu, penulis juga mengimplementasikan tampilan pada halaman status langganan sesuai desain yang telah disiapkan oleh tim UI/UX. Halaman ini menampilkan ringkasan informasi langganan secara menyeluruh dalam satu tampilan yang rapi dan mudah dipahami oleh pengguna. Implementasi dilakukan dengan menyesuaikan struktur komponen dan penyajian elemen visual, serta

memastikan bahwa data yang ditampilkan telah terhubung dengan API sehingga informasi langganan dapat tersaji secara konsisten dan sesuai dengan respons dari server.



Gambar 4. 2 Halaman Status Langganan

Halaman dokumentasi produk juga menjadi salah satu bagian yang diimplementasikan selama proses pengembangan. Bagian ini berfungsi untuk menampilkan panduan penggunaan, deskripsi fitur, serta informasi teknis yang ditulis dan diperbarui oleh admin. Pada implementasinya, penulis memastikan bahwa konten dokumentasi yang diambil dari API dapat ditampilkan secara konsisten dan sesuai dengan desain antarmuka yang telah ditetapkan oleh tim UI/UX.



Gambar 4. 3 Halaman Dokumentasi Produk