**DOKUMENTASI**

**SISTEM CANDRA**

1. **Deskripsi Sistem**

CANDRA (Scan & Tracking) adalah sistem manajemen dan pendataan proses produksi yang dirancang khusus untuk mengelola, melacak, dan menganalisis proses scan dokumen dalam perusahaan. Sistem ini memastikan setiap dokumen hasil produksi dapat dipantau secara akurat, termasuk jumlah lembar dan jumlah image yang di-scan, sehingga membantu dalam evaluasi dan optimalisasi proses produksi. Sistem ini dikembangkan dengan pendekatan backend dan frontend yang terpisah, menggunakan teknologi modern untuk memastikan performa tinggi, skalabilitas, dan kemudahan pengelolaan.

* **Backend**: Dibangun menggunakan Node.js, bertanggung jawab dalam mengelola API, autentikasi, dan komunikasi dengan database.
* **Frontend:** Dikembangkan menggunakan React.js, dengan desain yang responsif dan modern berkat Tailwind CSS, memberikan pengalaman pengguna yang optimal.

1. **Tujuan Pembuatan Sistem CANDRA**
2. **Manajemen Scan Dokumen**: Memastikan setiap dokumen hasil produksi dapat terdata dengan baik dalam sistem.
3. **Pelacakan Dokumen:** Memberikan fitur untuk melacak status dan histori dokumen secara real-time.
4. **Monitoring Kuantitas Scan:** Mencatat dan menampilkan jumlah lembar serta jumlah image yang telah di-scan untuk analisis lebih lanjut.
5. **Optimasi Workflow:** Mempercepat dan merapikan proses produksi dengan sistem yang terdigitalisasi.
6. **Keamanan** **Data:** Menyimpan dan mengelola data secara aman dengan autentikasi dan enkripsi yang sesuai.
7. **Aksesibilitas dan Kemudahan** **Penggunaan:** Menghadirkan antarmuka yang ramah pengguna dengan pengalaman yang responsif dan interaktif.
8. **Arsitektur Sistem**

Sebelumnya, sistem CANDRA (Scan & Tracking) dibangun menggunakan PHP Native. Namun, sistem mengalami berbagai kendala, seperti:

* **Performa yang Lambat:** Waktu eksekusi yang lama, terutama dalam pengolahan data scan dalam jumlah besar.
* **Source Code Tidak Terstruktur:** Sulit dikelola dan dikembangkan lebih lanjut.
* **Kurangnya Skalabilitas:** Tidak optimal untuk pengembangan fitur tambahan di masa depan.

Oleh karena itu, sistem ini dimigrasikan ke teknologi modern dengan spesifikasi sebagai berikut:

|  |  |
| --- | --- |
| **TEKNOLOGI** | **DESKRIPSI** |
| React Js | Frontend |
| Tailwind CSS | Styling |
| Node JS | Bakcend & REST API |
| Express JS | Routing Backend |
| Access (.mdb) | Database |
| Nginx | Deployment |

****

Keuntungan Migrasi ke Node.js & React.js

1. **Peningkatan Performa**

* Node.js lebih cepat dalam menangani request asinkron dibandingkan PHP Native.
* React.js memungkinkan rendering halaman yang lebih cepat dengan Virtual DOM.

1. **Struktur Kode yang Lebih Baik**

* Menggunakan arsitektur MVC (Model-View-Controller) pada backend untuk meningkatkan keteraturan kode.
* Pemisahan frontend dan backend memudahkan pengembangan dan pemeliharaan.

1. **Skalabilitas dan Fleksibilitas**

* Node.js dapat menangani lebih banyak permintaan secara efisien dengan event-driven architecture.
* React.js mendukung pengembangan fitur baru dengan lebih cepat dan modular.

1. **UI yang Lebih Modern dan Responsif**

* Tailwind CSS memberikan fleksibilitas dalam styling yang lebih rapi dan efisien.
* UI lebih adaptif untuk berbagai ukuran layar (desktop, tablet, mobile).

1. **Efisiensi dalam Manajemen Kegiatan Produksi**

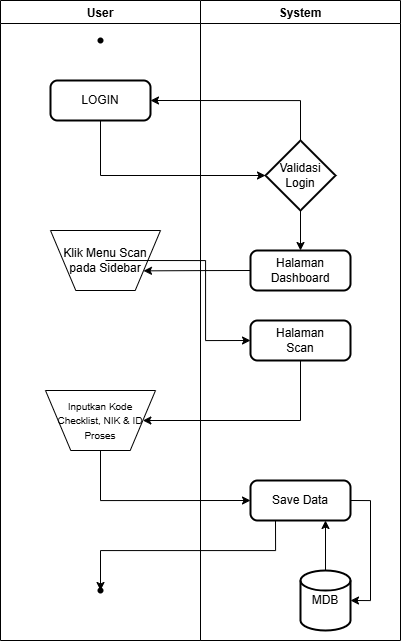
* Otomasi dalam pencatatan dan pelaporan jumlah scan dokumen per hari.
* Pemantauan produksi lebih akurat dengan dashboard interaktif dan fitur pelacakan data real-time.
* Reduksi pekerjaan manual yang berulang, sehingga meningkatkan produktivitas tim produksi.

1. **Peningkatan Manajemen Data Scan**

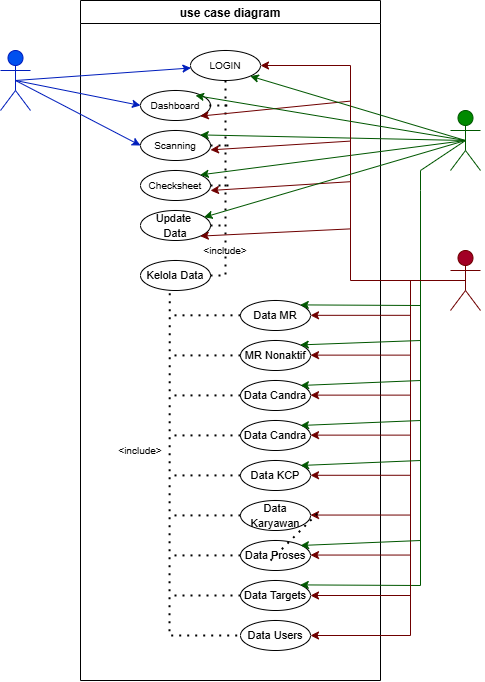
* Backend dapat menangani jumlah lembar dan image scan dengan lebih optimal.
* Query database lebih cepat dengan sistem caching dan optimasi query.

Dengan migrasi ini, sistem CANDRA menjadi lebih cepat, efisien, dan dapat mendukung manajemen produksi sehari-hari dengan lebih optimal.

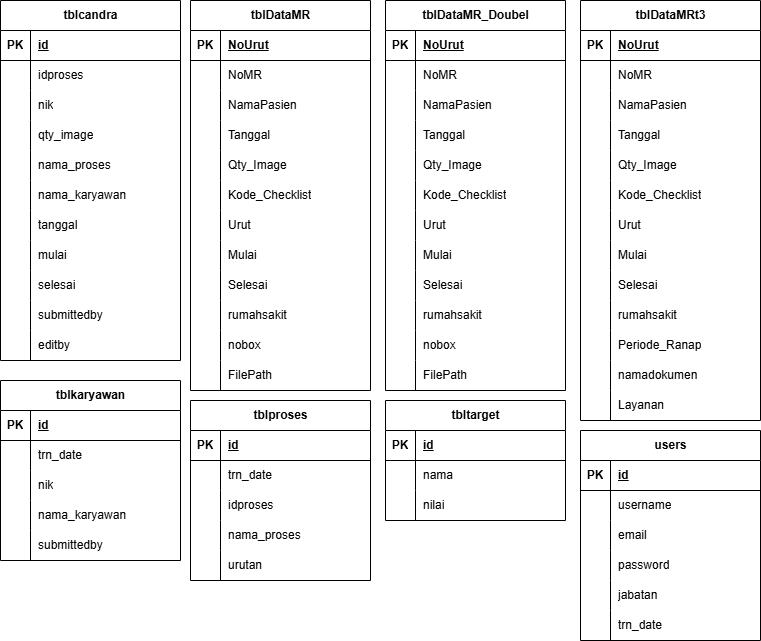
* 1. **Diagram Alur Sistem CANDRA**



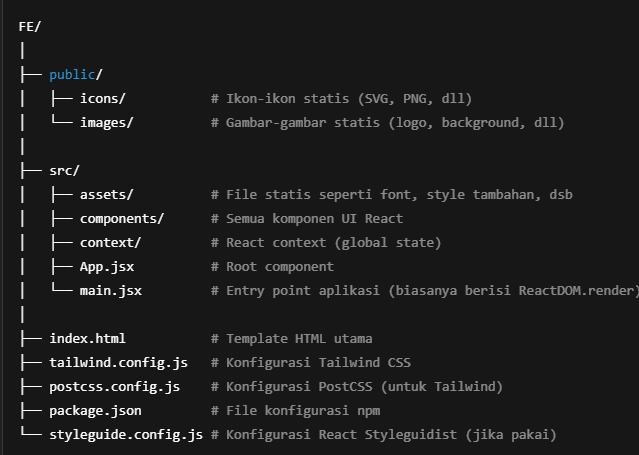
* 1. **Use Case Diagram**



* 1. **Design Database**



1. **Struktur Direktori**
   1. **Frontend**



* 1. **Backend**

1. **Installasi & Setup**
2. Persyaratan yang perlu di install:

* Node JS v22.13.1

<https://nodejs.org/id/download>

* MySQL

<https://www.oracle.com/mysql/technologies/mysql-enterprise-edition-downloads.html#windows>

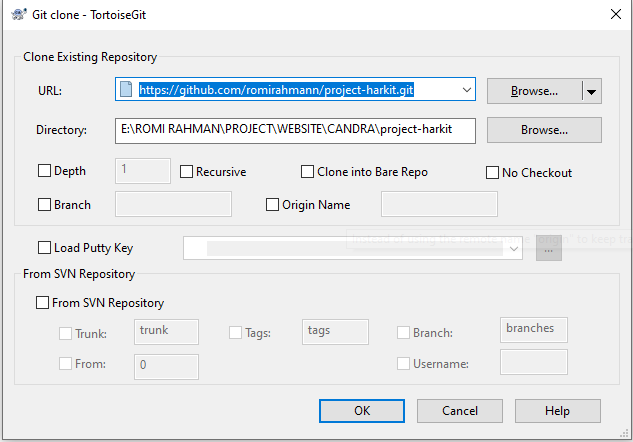
* Git

Link git: <https://git-scm.com/downloads/win>

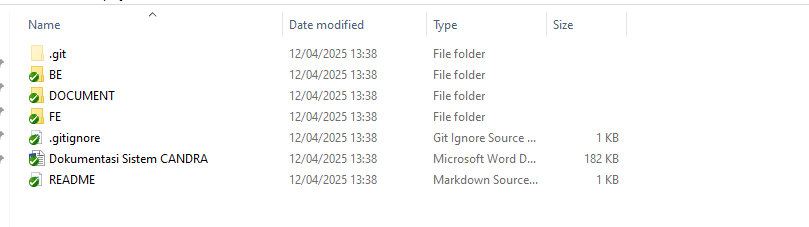
Link GUI git: <https://tortoisegit.org/download/>

1. Clone repository

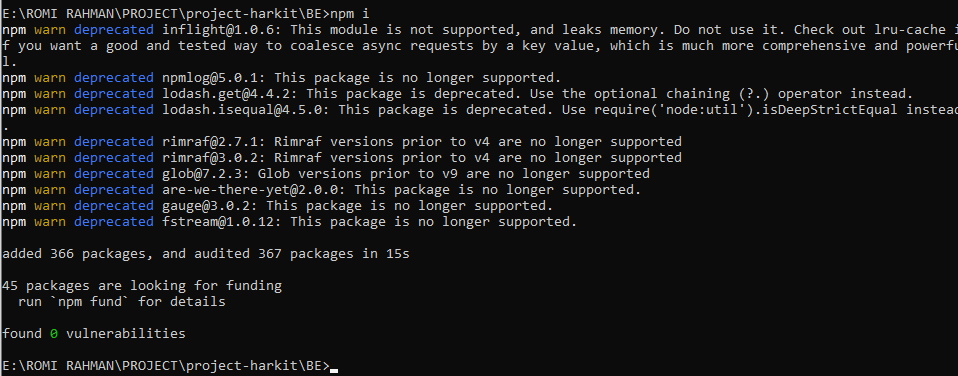
Link repo: <https://github.com/romirahmann/project-harkit.git>



1. Install dependensi bagian Backend dan Frontend dengan:

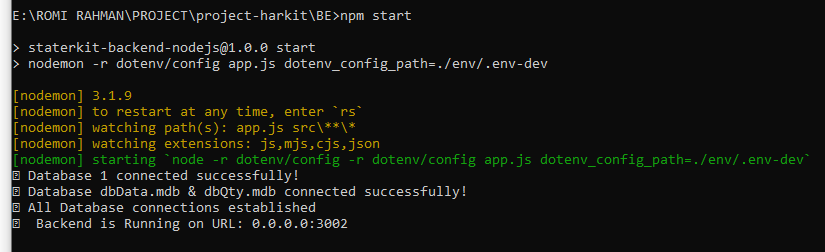


* Masuk folder BE
* Kemudian masuk cmd, jalankan perintah *npm i*



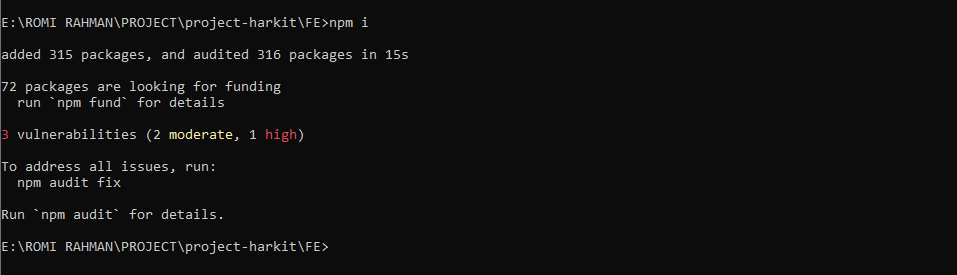
* Jalankan server nodeJs di cmd dengan perintah

“*npm start*”



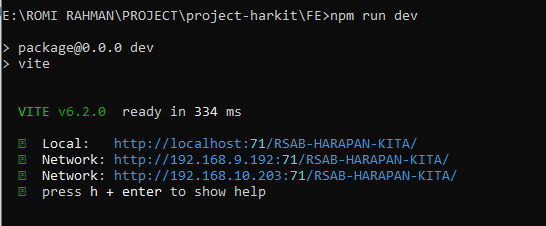
Server sudah dapat diakses dengan *port 3002*

* Selanjutnya masuk ke folder FE, jalankan perintah *npm i*



* Jalankan server react js di cmd dengan perintah

*“npm run dev”*



Server reactJS sudah dijalankan dan dapat diakses melalui

➜ Local: <http://localhost:71/RSAB-HARAPAN-KITA/>

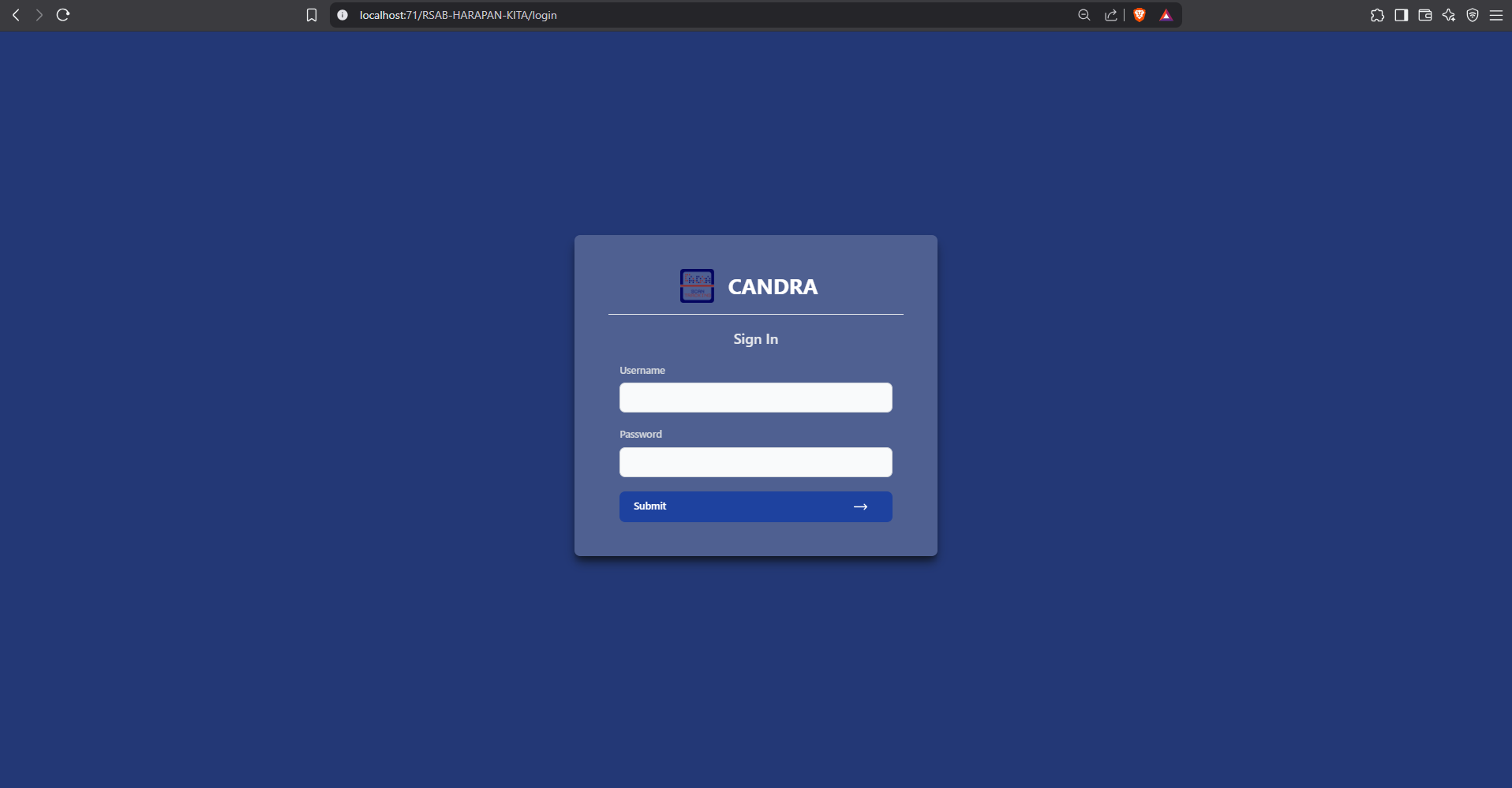
➜ Network: <http://192.168.9.192:71/RSAB-HARAPAN-KITA/>

➜ Network: <http://192.168.10.203:71/RSAB-HARAPAN-KITA/>

**\*NOTE\***

*url untuk server reactJS akan menyesuaikan dengan IP computer yang digunakan*

* Setelah semua server telah berjalan, akses sistem dengan menjalankan salah satu url reactJS diatas



1. **DOKUMENTASI SOURCE CODE**

Ada beberapa file yang perlu dikonfigurasi sebelum dijalankan, baik dari folder backend maupun frontend. Berikut ini beberapa file yang perlu di konfigurasi:

**BACKEND**

1. File .*env*
2. File db.config.js pada folder src/database/