**LAPORAN**

**PROJECT WORK**

**“SISTEM INFORMASI INVENTARIS SARANA DAN PRASARANA SEKOLAH MENGGUNAKAN LARAVEL BERBASIS WEB”**

Diajukan untuk memenuhi salah satu syarat kelulusan dari SMK Negeri 1 Cimahi

OLEH:

NAMA : STEFANUS REVAN REGA REI

NO. INDUK : 181113886

TINGKAT : III (TIGA)

BIDANG KEAHLIAN : TEKNOLOGI INFORMASI DAN

KOMUNIKASI

KOMPETENSI KEAHLIAN : REKAYASA PERANGKAT LUNAK



**SEKOLAH MENENGAH KEJURUAN NEGERI 1**

**KOTA CIMAHI**

**2021**

LEMBAR PENGESAHAN DARI PIHAK SEKOLAH

**“SISTEM INFORMASI INVENTARIS SARANA DAN PRASARANA SEKOLAH MENGGUNAKAN LARAVEL BERBASIS WEB”**

Laporan ini telah disetujui oleh:

|  |  |  |
| --- | --- | --- |
| Ketua Kompetensi Keahlian |  | Pembimbing |
|  |  |  |
|  |  |  |
|  |  |  |
| **Agus Rahmawan, S.T**  **NIP. 197512022006041010** |  | **Yuli Pamungkas, S.P**  **NIP. 197007142008011007** |

|  |
| --- |
| Mengetahui: |
| Kepala SMK Negeri 1 Cimahi |
|  |
|  |
|  |

**Drs. Daud Saleh, M.M**

**NIP. 196307181989021001**

# KATA PENGANTAR

Segala puji dan syukur penulis ucapkan kehadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan seluruh rangkaian pelaksanaan kegiatan pengganti praktik kerja lapangan (PKL) berupa *project work* dengan lancar dan baik sebagai salah satu kewajiban siswa dan siswi tingkat III di SMK Negeri 1 Cimahi. Selain itu, penulis juga sangat bersyukur karena dapat menyelesaikan laporan akhir sebagai bukti dari pelaksanaan kegiatan pengganti PKL berupa *project work* yang berjudul **“Sistem Informasi Inventaris Sarana dan Prasarana Sekolah Menggunakan Laravel Berbasis Web”**.

Dalam penyusunan laporan akhir ini penulis banyak mendapatkan pengalaman dan tambahan ilmu, di samping itu penulis juga banyak mengalami kesulitan dan hambatan selama pelaksanaan kegiatan pengganti praktik kerja lapangan. Akan tetapi, berkat panduan, bimbingan, dan juga dorongan baik secara langsung maupun tidak langsung dari berbagai pihak, pelaksanaan kegiatan pengganti praktik kerja lapangan dan pembuatan laporan ini dapat diselesaikan, walaupun penulis sadari masih banyak kekurangan. Maka pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam pelaksanaan kegiatan pengganti praktik kerja lapangan dan penyusunan laporan ini, terutama kepada :

1. Drs. Daud Saleh, M.M, selaku Kepala Sekolah SMK Negeri 1 Cimahi.
2. Bapak Agus Rahmawan, ST, selaku Ketua Kompetensi Keahlian Rekayasa Perangkat Lunak SMK Negeri 1 Cimahi.
3. Bapak Yuli Pamungkas, SP, selaku pembimbing kegiatan pengganti PKL yang telah membimbing dan memberikan arahan penulis dalam pengerjaan laporan.
4. Bapak Mulyono Bano Syaidin selaku wali kelas XII Rekayasa Perangkat Lunak B yang telah membimbing dan memotivasi penulis dalam pengerjaan laporan.
5. Kepada semua guru pengajar di kompetensi keahlian Rekayasa Perangkat Lunak SMK Negeri 1 Cimahi.
6. Kepada semua pihak yang telah terlibat dan membantu penulis dalam melaksanakan kegiatan pengganti praktik kerja lapangan dan penyusunan laporan ini.

Semoga seluruh amal baik dari Bapak dan Ibu serta seluruh pihak yang telah membantu penulis dalam pelaksanaan kegiatan pengganti praktik kerja lapangan berupa *project work* dan penyusunan laporan ini mendapat balasan yang baik dari Tuhan Yang Maha Esa.

Penulis menyadari bahwa dalam penulisan laporan ini masih banyak terdapat kekurangan dikarenakan keterbatasan kemampuan yang dimiliki. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak untuk perbaikan dan masukkan kepada penulis di masa yang akan datang.

Akhir kata, penulis berharap semoga laporan ini dapat memberikan manfaat bagi penulis sendiri pada khususnya dan umunya untuk setiap pihak yang menggunakannya.

Cimahi, Januari 2021

Penulis

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc69591487)

[DAFTAR ISI iii](#_Toc69591488)

[DAFTAR GAMBAR vi](#_Toc69591489)

[DAFTAR TABEL vii](#_Toc69591490)

[BAB I PENDAHULUAN 1](#_Toc69591491)

[1.1. Latar Belakang 1](#_Toc69591492)

[1.2. Tujuan 2](#_Toc69591493)

[1.3. Pembatasan Masalah 2](#_Toc69591494)

[1.4. Sistematika Pembahasan 3](#_Toc69591495)

[BAB II LANDASAN TEORI 4](#_Toc69591496)

[2.1. Website 4](#_Toc69591497)

[2.1.1. Website Statis 4](#_Toc69591498)

[2.1.2. Website Dinamis 5](#_Toc69591499)

[2.2. Client Side Scripting 5](#_Toc69591500)

[2.2.1. Hyper Text Markup Language (HTML) 5](#_Toc69591501)

[2.2.2. Cascading Style Sheet (CSS) 6](#_Toc69591502)

[2.2.3. JavaScript 7](#_Toc69591503)

[2.2.4. jQuery 7](#_Toc69591504)

[2.3. Server Side Scripting 8](#_Toc69591505)

[2.3.1. PHP: Hypertext Preprocessor (PHP) 9](#_Toc69591506)

[2.4. Framework 10](#_Toc69591507)

[2.4.1. Laravel 10](#_Toc69591508)

[2.4.2. Bootstrap 11](#_Toc69591509)

[2.5. Basis Data 11](#_Toc69591510)

[2.5.1. Database Management System (DBMS) 12](#_Toc69591511)

[2.5.2. MySQL 13](#_Toc69591512)

[2.6. Composer 13](#_Toc69591513)

[2.7. Entity Relationship Diagram (ERD) 14](#_Toc69591514)

[2.8. Data Flow Diagram (DFD) 16](#_Toc69591515)

[2.9. Flow Map 17](#_Toc69591516)

[2.10. Implementasi Teori Pada Aplikasi 19](#_Toc69591517)

[BAB III ANALISA DAN PERANCANGAN 21](#_Toc69591518)

[3.1. Analisa Aplikasi 21](#_Toc69591519)

[3.2. Analisa Pengguna Aplikasi 22](#_Toc69591520)

[3.3. Analisa Kebutuhan Pengguna 22](#_Toc69591521)

[3.3.1. Analisa Kebutuhan Admin 22](#_Toc69591522)

[3.3.2. Analisa Kebutuhan Pegawai 23](#_Toc69591523)

[3.4. Perancangan Sistem 23](#_Toc69591524)

[3.4.1. Flowmap 23](#_Toc69591525)

[3.4.2. Data Flow Diagram 33](#_Toc69591526)

[3.4.3. Rancangan Basis Data 36](#_Toc69591527)

[3.4.4. Perancangan Layout Halaman 37](#_Toc69591528)

[BAB IV IMPLEMENTASI DAN PENGUJIAN 39](#_Toc69591529)

[4.1. Implementasi 39](#_Toc69591530)

[4.1.1. Desain Front End / Tampilan 39](#_Toc69591531)

[4.1.2. Struktur Tabel 46](#_Toc69591532)

[4.1.3. Relasi Tabel 50](#_Toc69591533)

[4.2. Pengujian 51](#_Toc69591534)

[BAB V PENUTUP 58](#_Toc69591535)

[5.1. Kesimpulan 58](#_Toc69591536)

[5.2. Saran 58](#_Toc69591537)

[DAFTAR PUSTAKA 60](#_Toc69591538)

# DAFTAR GAMBAR

[Gambar 3.1 Flowmap Sistem Login 25](#_Toc69589018)

[Gambar 3.2 Flowmap Sistem Tambah User 26](#_Toc69589019)

[Gambar 3.3 Flowmap Sistem Edit User 27](#_Toc69589020)

[Gambar 3.4 Flowmap Sistem Tambah Aset 28](#_Toc69589021)

[Gambar 3.5 Flowmap Sistem Edit Aset 29](#_Toc69589022)

[Gambar 3.6 Flowmap Sistem Tambah Ruangan 30](#_Toc69589023)

[Gambar 3.7 Flowmap Sistem Edit Ruangan 31](#_Toc69589024)

[Gambar 3.8 Flowmap Sistem Tambah Transaksi 32](#_Toc69589025)

[Gambar 3. 9 Flowmap Sistem Edit Transaksi 33](#_Toc69589026)

[Gambar 3.10 DFD Level 0 34](#_Toc69589027)

[Gambar 3.11 DFD Level 1 - Admin 35](file:///D:\aRevan\Kelas%20XII\Laporan%20Project%20Work.docx#_Toc69589028)

[Gambar 3.12 DFD Level 1 - Pegawai 36](#_Toc69589029)

[Gambar 3.13 Entity Relationship Diagram 37](#_Toc69589030)

[Gambar 3.14 Desain Layout Halaman Login 38](#_Toc69589031)

[Gambar 3.15 Desain Layout Halaman Dashboard 38](#_Toc69589032)

[Gambar 3.16 Desain Layout Halaman User 39](#_Toc69589033)

[Gambar 4.1 Halaman Login 40](#_Toc69589344)

[Gambar 4.2 Halaman Dashboard 41](#_Toc69589345)

[Gambar 4.3 Halaman User 42](#_Toc69589346)

[Gambar 4.4 Halaman KIB A 42](#_Toc69589347)

[Gambar 4.5 Halaman KIB B 43](#_Toc69589348)

[Gambar 4.6 Halaman KIB C 43](#_Toc69589349)

[Gambar 4.7 Halaman KIB D 44](#_Toc69589350)

[Gambar 4.8 Halaman KIB E 44](#_Toc69589351)

[Gambar 4.9 Halaman Data Aset 45](#_Toc69589352)

[Gambar 4.10 Halaman Ruangan 46](#_Toc69589353)

[Gambar 4.11 Halaman Change Password 46](#_Toc69589354)

[Gambar 4.12 Relasi Tabel 51](#_Toc69589355)

# DAFTAR TABEL

[Tabel 2.1 Simbol-Simbol ERD 16](#_Toc69589746)

[Tabel 2.2 Simbol-Simbol DFD 17](#_Toc69589747)

[Tabel 2.3 Simbol-Simbol *Flowmap* 18](#_Toc69589748)

[Tabel 2.4 Tabel Implementasi Teori Pada Aplikasi 20](#_Toc69589749)

[Tabel 4.1 Struktur Tabel Login 47](#_Toc69589827)

[Tabel 4.2 Struktur Tabel KIB A 47](#_Toc69589828)

[Tabel 4.3 Struktur Tabel KIB C 48](#_Toc69589829)

[Tabel 4. 4 Struktur Tabel KIB D 49](#_Toc69589830)

[Tabel 4.5 Struktur Tabel Aset 49](#_Toc69589831)

[Tabel 4.6 Struktur Tabel Ruangan 50](#_Toc69589832)

[Tabel 4.7 Pengujian Sistem Login 52](#_Toc69589833)

[Tabel 4.8 Pengujian Sistem Tambah User 53](#_Toc69589834)

[Tabel 4.9 Pengujian Edit User 53](#_Toc69589835)

[Tabel 4.10 Pengujian Hapus User 54](#_Toc69589836)

[Tabel 4.11 Pengujian Tambah Data Aset 54](#_Toc69589837)

[Tabel 4.12 Pengujian Edit Data Aset 55](#_Toc69589838)

[Tabel 4.13 Pengujian Hapus Data Aset 55](#_Toc69589839)

[Tabel 4.14 Pengujian Tambah Ruangan 56](#_Toc69589840)

[Tabel 4.15 Pengujian Edit Ruangan 56](#_Toc69589841)

[Tabel 4.16 Pengujian Hapus Ruangan 57](#_Toc69589842)

[Tabel 4.17 Pengujian Tambah Transaksi 57](#_Toc69589843)

[Tabel 4.18 Pengujian Edit Transaksi 58](#_Toc69589844)

[Tabel 4.19 Pengujian Hapus Transaksi 58](#_Toc69589845)

# BAB I PENDAHULUAN

## Latar Belakang

Perkembangan teknologi informasi dan komunisasi yang sangat pesat membuat kehadirannya mengakibatkan dampak yang cukup besar terhadap kehidupan umat manusia dalam berbagai aspek dan kalangan. Internet merupakan salah satu instrumen dalam era globalisasi yang telah menjadikan dunia ini menjadi transparan dan terhubungkan dengan sangat mudah dan cepat.

Di masa-masa mendatang, arus informasi akan semakin meningkatdan menuntut siapapun untuk beradaptasi dengan kecenderungan itu. Dengan kondisi demikian maka semua elemen masyarakat menggunakan keuntungan tersebut, salah satunya adalah dengan menggunakan teknologi tersebut untuk mempermudah pekerjaan mereka hamper untuk semua aspek.

Pemanfaatan teknologi dapat digunakan untuk memudahkan pekerjaan manusia sehingga banyak instasi dari berbagai bidang mulai menggunakan pemanfaatan teknologi dan informasi tersebut. Instansi pendidikan pun dapat memanfaatkan keuntungan dari teknologi untuk membuat suatu aplikasi yang dapat membantu pendataan dan pengelolaan data salah satunya pada bagian sarana dan prasarana sekolah karena saat ini kebanyakan sekolah masih memiliki kendala dalam melakukan pendataan dan pengelolaan data secara konvensional atau tradisional yang memungkinkan terjadinya banyak kesalahan. Diantaranya pendataan sarana ataupun prasarana yang dilakukan kurang efektif, memungkinkan kesalahan pada saat pendataan sarana ataupun prasarana seperti data yang kurang akurat atau terjadinya pengulangan data, dan apabila dokumen yang berisikan data penting hilang maka keseluruhan data yang disimpan pun akan hilang.

Hal-hal tersebut dapat mengakibatkan penanganan yang kurang maksimal, hasilnya data yang dihasilkan pun kurang maksimal, sehingga informasi yang ada tidak dapat tersalurkan dengan baik. Oleh karena itu masalah tersebut dapat diatasi dengan memanfaatkan teknologi. Salah satu solusinya adalah dengan dibuatnya suatu Aplikasi Sistem Informasi Pengelolaan Sarana dan Prasarana yang diharapkan dapat mempermudah pihak sekolah untuk melakukan pendataan dan penglolaan data sarana dan prasarana. Dengan demikian diharapkan masalah-masalah yang muncul pada proses pendataan dan pengelolaan data sarana dan prasarana sekolah terkait dapat dicegah dan teratasi.

## Tujuan

Tujuan yang dicapai oleh Sistem Informasi Sarana dan Prasarana SMK Negeri 1 Cimahi berbasis *website* adalah :

1. Mempermudah pendataan data sarana dan prasarana sekolah.
2. Mengoptimalkan pengelolaan data sarana dan prasarana sekolah.
3. Meningkatkan kualitas pelayanan sarana dan prasarana untuk warga sekolah.
4. Mencegah dan mengatasi permasalahan yang muncul dalam pendataan dan pengelolaan data sarana dan prasarana yang masih konvensional.
5. Mempermudah siswa untuk melakukan pengajuan dan peminjaman sarana.

## Pembatasan Masalah

Masalah yang disampaikan pada latar belakang perlu dibatasi agar sesuai dengan apa yang dibahas pada karya tulis ini. Berikut batasan-batasan masalahnya:

1. Pengaturan tampilan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi ini diatur menggunakan bahasa pemograman CSS dan *framework* Bootstrap.
2. Interaksi antara aplikasi dengan user menggunakan bahasa pemograman Javascript dan *framework* JQuery.
3. Aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi ini menggunakan framework Laravel 8
4. Aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi ini tidak membahas mengenai transaksi pembelian sarana yang digunakan pihak sekolah secara mendetail.
5. Aplikasi Inventaris Sarana dan Prasarana SMK Negeri 1 Cimahi ini memberikan laporan rekap data dengan format .xls.

## Sistematika Pembahasan

Agar Pembaca lebih mudah mempelajari dan memahami inti dari laporan yang dibuat penulis, maka penulis membaginya menjadi beberapa bab, yaitu :

**BAB 1 PENDAHULUAN**

Bab I Pendahuluan memuat latar belakang masalah, tujuan dari pembuatan aplikasi, pembatasan masalah, dan sistematika pembahasan.

**BAB II LANDASAN TEORI**

Bab II Landasan Teori berisi teori-teori yang menjadi acuan atau landasan dengan judul yang dibahas.

**BAB III ANALISA DAN PERANCANGAN**

Bab III Analisa dan Perancangan berisi uraian materi yang disesuaikan dengan laporan yang dibuat seperti analisa sistem, analisa kebutuhan data, kebutuhan aplikasi tersebut dan pembahasan tentang perancangan aplikasi.

**BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab III Implementasi dan Pengujian berisi tentang implementasi aplikasi dan beberapa pengujian untuk aplikasi yang dibuat.

**BAB V PENUTUP**

Bab IV Penutup berisi kesimpulan dari hasil analisis pada Bab III dan relevansinya dengan teori-teori pada Bab II serta saran-saran yang bersifat solusi, dan membangun terhadap aplikasi yang dibuat.

# BAB II LANDASAN TEORI

## Website

Website adalah kumpulan dari halaman-halaman situs yang terangkum dalam sebuah *domain* dan *subdomain*, yang tempatnya berada di World Wide Web (WWW) dalam internet. Sebuah halaman web biasanya berupa dokumen yang ditulis dalam format Hyper Text Markup Language (HTML), yang dapat diakses melalui Hyper Text Transfer Protocol (HTTP), yaitu sebuah protokol yang menyampaikan informasi dari *server* website untuk ditampilkan kepada para pemakai melalui *web browser*.

*Website* atau situs dapat juga diartikan sebagai 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 dimana masing-masing dihubungkan dengan jaringan-jaringan halaman atau disebut juga *hyperlink*.

Dalam pengembangan aplikasi Inventaris Sarana dan Prasarana SMK Negeri 1 Cimahi, penulis menggunakan *website* sebagai platform aplikasi.

Berdasarkan teknologinya, website terbagi menjadi dua, yaitu *web* statis dan *web* dinamis.

### Website Statis

*Web* statis adalah *website* yang mana pengguna tidak bisa mengubah konten dari web tersebut secara langsung menggunakan *browser*. Interaksi yang terjadi antara pengguna dengan *server* hanyalah seputar pemrosesan *link* saja. *Web* statis tidak berinteraksi dengan *database*, data dan informasi yang ada pada *web* statis tidak berubah-ubah kecuali diubah *sintaks*nya. Dokumen web yang dikirim kepada *client* akan sama isinya dengan apa yang ada di *web server*.

Contoh dari *web* statis adalah *web* yang hanya ada beberapa halaman saja dan kontennya hampir tidak pernah berubah karena konten langsung diletakan dalam *file* HTML saja.

### Website Dinamis

Dalam *web* dinamis, interaksi yang terjadi antara pengguna dengan *server* sangat kompleks. Seseorang bisa mengubah konten dari halaman tertentu dengan menggunakan *browser*. *Request* (permintaan) dari pengguna dapat diproses oleh *server* yang kemudian ditampilkan dalam isi yang berbeda-beda menurut alur programnya. *Web* dinamis dapat berinteraksi dengan *database*. *Web* dinamis, memiliki data dan informasi yang berbeda-beda tergantung input apa yang disampaikan *client*. Data yang sampai di *client* akan berbeda dengan data yang ada di *web server*. Contoh dari *web* dinamis adalah Sistem Informasi Sarana dan Prasarana ini sendiri.

## Client Side Scripting

**Client-side Scripting**adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh komputer pengguna/pengunjung. Jadi, ketika seseorang berkunjung ke sebuah web, komputernya akan mendownload data/*script* yang bersifat *client-side* di web tersebut. *Client-side script* meliputi kode-kode yang ditampilkan ketika anda mengklik kanan pada sebuah halaman web dan melihat sumber halaman (*View Page Source*).

### Hyper Text Markup Language (HTML)

Hyper Text Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser* internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan Standard Generalized Markup Language (SGML), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C).

HTML berupa kode-kode tag yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah *file* yang merupakan *file* HTML dapat dibuka dengan menggunakan *web browser*. Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, penulis menggunakan bahasa pemrograman HTML versi 5 untuk menjadi kerangka dasar aplikasi.

### Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Sama halnya *styles* dalam aplikasi pengolah kata seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya heading, sub bab, bodytext, footer, images dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas atau *file*. Pada umumnya CSS dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Dalam pengembangannya aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, penulis menggunakan bahasa pemrograman CSS versi 3 untuk mengatur tampilan aplikasi.

### JavaScript

JavaScript adalah sekumpulan perintah khusus yang digunakan untuk membuat sebuah halaman web yang lebih responsif dan interaktif. JavaScript merupakan bahasa *script* yang dicantumkan pada sebuah halaman web dan dijalankan pada *browser*. JavaScript terutama terkenal karena penggunaannya di halaman web yang memberikan kemampuan tambahan pada HTML dengan mengizinkan pengeksekusian perintah di sisi *user*, bukan di sisi *server web*.

JavaScript bergantung kepada browser yang memanggil halaman *web* yang berisi *script-script* dari JavaScript yang terdapat dalam dokumen HTML. *Syntax* JavaScript yang berada dalam file HTML harus dimasukkan di antara tag <script> dan </script>. JavaScript tidak memerlukan *compiler* atau program khusus untuk menjalankannya karena JavaScript sendiri sudah termasuk di dalam browser.

Secara fungsional, JavaScript digunakan untuk menyediakan akses script pada objek yang dibenamkan atau *embedded*. *Script* ini dapat membuka halaman, melakukan validasi pada *form* sebelum data dikirimkan ke *server*, mengubah kursor ketika melewati objek tertentu, berkomunikasi dengan *server* dan masih banyak lagi. Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, Penulis menggunakan bahasa pemrograman JavaScript untuk mempermudah interaksi antara *user* dan aplikasi.

### jQuery

JQuery adalah sebuah *library* JavaScript yang sangat ringkas dan sederhana untuk memanipulasi komponen di dokumen HTML, menangani event, animasi, efek dan memproses interaksi AJAX. JQuery dirancang sedemikian rupa agar membuat program menggunakan JavaScript menjadi relatif sangat mudah. JQuery memiliki ukuran yang cukup kecil, sehingga tidak memperlambat proses *loading* halaman *web* yang kita buat. JQuery juga kompatibel dengan CSS3 dan yang tidak kalah penting adalah JQuery bisa berjalan di semua *browser*.

JQuery adalah *library* JavaScript yang gratis dan *open source*. Oleh karena itu semua orang dapat menggunakan JQuery untuk keperluan pengembangan situs *web*. Tanpa *library* seperti JQuery, menerapkan JavaScript mungkin akan lebih sulit, terutama untuk pemula yang baru belajar JavaScript. *Plugin* tambahan seperti JQuery UI (User Interface) akan memudahkan pengembangan website yang cantik dan interaktif. Selain itu tersedia *plugin-plugin* lain yang makin memperkaya kemampuan JQuery.

Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, Penulis menggunakan JQuery untuk mempermudah dan mempercepat pembuatan fitur interaksi antara *user* dan aplikasi.

## Server Side Scripting

*Server-side scripting* adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh sisi server. Pengolahannya dilakukan di komputer server dan hasil pengolahannya kirimkan ke komputer klien/pengunjung dalam bentuk bahasa *markup* (HTML), sehingga pengunjung tidak mengetahui *script* yang telah ditulis oleh pembuatnya. Web tipe ini bersifat dinamis yaitu bisa dirubah kontennya dengan mudah tanpa merubah *script*-nya. Bisa digunakan untuk berbagai kebutuhan web seperti, Toko *Online*, forum, web sekolah, *blogging* dan berbagai kebutuhan lainnya.

*Server-side scripting* biasanya digunakan untuk menyediakan antarmuka ke *client* dan membatasi *client* untuk mengakses *database*, atau sumber informasi yang sifatnya rahasia. Script ini dapat menyesuaikan respon berdasarkan karakteristik, kebutuhan pengguna, hak akses, dan lainnya. Selain itu, *server-side scripting* juga memungkinkan pengelola server untuk membatasi akses ke *source code* dari *script* yang dijalankan. Karena proses pengolahan informasi data berlangsung di sisi *server*, maka metode *server-side scripting* banyak digunakan dalam pemrograman web yang memungkinkan server dapat menghasilkan halaman web dinamis.

Contoh bahasa *server-side scripting* yang banyak digunakan dalam pemrograman webdiantaranya adalah ASP, PHP, Python, Perl, Java Server Pages, dan Cold Fusion. Dengan kemampuan dalam menghasilkan halaman web dinamis, tentu saja memungkinkan sebuah webdapat menjadi lebih interaktif.

### PHP: Hypertext Preprocessor (PHP)

PHP **(**PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang termasuk kategori *server side programming* yaitu bahasa pemrograman yang nantinya sintak atau *script* program tersebut akan dijalankan oleh *server* dimana hasil pengolahan *script* program tersebut akan dikirim ke klien sebagai output. Selain *server side programming*, ada jenis yang lain yaitu *client side* *programming* yaitu script atau program akan diproses di dalam klien sendiri.

Pada prinsipnya, cara menggunakan PHP hanya perlu menyisipkan kode PHP ke dalam tag-tag HTML yang sudah ada di situs. Ketika ada yang mengakses web di halaman yang berisi kode PHP, *server* akan mengeksekusinya dan kemudian mengirimkan hasil eksekusinya ke web *server* untuk selanjutnya ditampilkan menggunakan kode HTML. Itulah sebabnya diperlukan server sendiri untuk mengetes kode PHP secara lokal. Dalam hal ini, *server* merupakan otaknya dan fungsi browser hanyalah untuk menampilkan hasil output *server* PHP ke PC klien.

Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, Penulis menggunakan bahasa pemrograman PHP versi 7.3.2 untuk membuat aplikasi menjadi dinamis.

## Framework

*Framework* merupakan kumpulan sintaks untuk berbagai *task* sehingga developer tidak perlu menjalankan perintah dan fungsi dasar. untuk memudahkan developer dalam mengembangkan sebuah aplikasi dengan sintaks yang telah disederhanakan. Sejatinya, *framework* adalah kerangka kerja untuk mengembangkan aplikasi agar lebih terstruktur dan tersusun rapi.

### Laravel

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi Massachusetts Institute of Technology (MIT), dibangun dengan konsep Model View Controller (MVC). Laravel merupakan *framework* pengembang website berbasis Model View Presenter (MVP) yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan megurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan *sintaks* yang ekspresif, jelas dan menghemat waktu.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti :

1. *Model*, *model* mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.

2. *View*, *view* adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman *web*.

3. *Controller*, *controller* merupakan bagian yang menjembatani *model* dan *view*.

Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, Penulis menggunakan *framework* laravel versi 8.12.3 untuk mempercepat dan mempermudah pengembangan aplikasi.

### Bootstrap

Bootstrap adalah sebuah *library framework* CSS yang dibuat khusus untuk bagian pengembangan *front-end* website. Bootstrap merupakan salah satu *framework* HTML, CSS dan JavaScript yang paling popular di kalangan *web developer*. Pada saat ini hampir semua *web developer* telah menggunakan bootstrap untuk membuat tampilan *front-end* menjadi lebih mudah dan sangat cepat. Karena developer hanya perlu menambahkan tag tertentu untuk membuatnya.

Bootstrap telah menyediakan kumpulan komponen class interface dasar yang telah dirancang sedemikian rupa untuk menciptakan tampilan yang menarik, bersih dan ringan. Selain komponen class interface, bootstrap juga memiliki fitur grid yang berfungsi mengatur layout pada halaman website yang bisa digunakan dengan sangat mudah dan cepat. Bootstrap memberi keleluasaan dalam mengembangkan tampilan *website* yaitu dengan cara mengubah tampilan bootstrap dengan class CSS sendiri.

Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, Penulis menggunakan *framework* bootstrap versi 4.4.1 untuk mengatur tampilan aplikasi menjadi lebih cepat dan mudah.

## Basis Data

Basis data atau *database* adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update yang rumit.

### Database Management System (DBMS)

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan *Database Management Sistem* (DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan *user* untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. Dengan kata lain semua akses ke basis data akan ditangani oleh DBMS.

Terdapat beberapa fungsi yang harus ditangani DBMS yaitu mengolah pendefinisian data, dapat menangani permintaan pemakai untuk mengakses data, memeriksa keamanan dan integritas data yang didefinisikan oleh Database Administrator (DBA), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun disk, dan menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama dari DBMS adalah untuk memberikan tinjauan abstrak data kepada *user*. Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara, dan tetap dapat diambil secara efisien. Pertimbangan efisien di sini adalah bagaimana merancang struktur data yang kompleks tetapi masih tetap bisa digunakan oleh pengguna tanpa mengetahui kompleksitas strukturnya.

DBMS memiliki berbagai macam jenis, diantaranya adalah Microsoft Access, Oracle, DB2, FireBird, SQL Server, Visual FoxPro, MySQL, dan lain-lain. Tentunya semua jenis dari DBMS ini memiliki kekurangan dan kelebihannya masing-masing, dan penggunaannya dapat menyesuaikan dengan kebutuhan aplikasi.

Adapun bahasa pemrograman khusus yang digunakan untuk memanajemen data dalam DBMS yaitu SQL. SQL (*Structured Query Language*) biasanya berupa perintah sederhana yang berisi instruksi-instruksi untuk manipulasi data. Perintah SQL ini sering juga disingkat dengan sebutan *query*.

### MySQL

MySQL adalah sebuah aplikasi DBMS SQL atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti apache, dimana perangkat lunak dikembangkan oleh komunitas umum dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya.

MySQL bersifat gratis dan *open source*, artinya setiap orang boleh menggunakan dan mengembangkan aplikasi ini. MySQL juga sebenarnya merupakan turunan dari salah satu konsep utama dalam *database* yaitu SQL.

Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, Penulis menggunakan MySQL sebagai *database server* aplikasi.

## Composer

Composer adalah *depedency* manager khusus PHP yang memiliki fungsionalitas seperti Gem (Ruby) atau Maven (Java). Composer dapat meng*install* suatu library dan composer juga akan secara otomatis meng*install* *library* lain yang dibutuhkan, tanpa perlu mengunduh satu persatu. Composer merupakan utilitas yang sangat membantu dalam mengelola proyek terhadap sebuah *library*.

PHP sudah memulai dengan membuat *repository* bernama packagist yang berisi *library* yang memungkinkan dipakai oleh semua *framework* dengan syarat *framework* tersebut harus dapat mengakses composer.

Dalam pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi, Penulis menggunakan composer untuk mengunduh *library* dan *plugin* yang dibutuhkan.

## Entity Relationship Diagram (ERD)

Diagram Hubungan Entitas atau entity relationship diagram merupakan model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpan. Model data sendiri merupakan sekumpulan cara, peralatan untuk mendeskripsikan data-data yang hubungannya satu sama lain, semantiknya, serta batasan konsistensi. Model data terdiri dari model hubungan entitas dan model relasional.

Diagram hubungan entitas ditemukan oleh Peter Chen dalam buku Entity Relational Model-Toward a Unified of Data. Chen mencoba merumuskan dasar-dasar model dan setelah itu dikembangkan dan dimodifikasi oleh Chen dan banyak pakar lainnya. Pada saat itu diagram hubungan entitas dibuat sebagai bagian dari perangkat lunak yang juga merupakan modifikasi khusus, karena tidak ada bentuk tunggal dan standar dari diagram hubungan entitas.

Diagram hubungan entitas digunakan untuk menkontruksikan model data konseptual, memodelkan struktur data dan hubungan antar data dan mengimplementasikan basis data secara logika maupun secara fisik dengan DBMS.

Adapun simbol-simbol yang terdapat dalam diagram hubungan entitas ini adalah :

Tabel 2. Simbol-Simbol ERD

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | Menyatakan himpunan entitas. |
|  | Menyatakan atribut. |
|  | Menyatakan himpunan relasi. |
|  | Menyatakan penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya. |

## Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) adalah sebuah diagram yang menggambarkan aliran data sebuah sistem dan kerja atau proses dilakukan dalam sistem tersebut.

Komponen utama pada DFD adalah sebagai berikut :

Tabel 2. Simbol-Simbol DFD

|  |  |  |
| --- | --- | --- |
| **DeMarco dan**  **Yourdan Symbols** | **Keterangan** | **Gane dan**  **Sarson Symbols** |
|  | *Source/Entity* mendefinisikan orang atau sebuah unit. |  |
|  | Proses adalah penyelenggaraan kerja atau jawaban, datangnya aliran data atau kondisinya. |  |
|  | *Data Flow* (Arus Data) merepresentasikan sebuah *input* data ke dalam sebuah proses atau *output* dari data pada sebuah proses. |  |
|  | *Data Store* (Penyimpanan Data). |  |

## Flow Map

*Flowmap* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowmap* berguna untuk membantu analisis dan *programmer* untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif pengoperasian. Biasanya *flowmap* mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Fungsi *flowmap* yaitu mendefinisikan hubungan antara bagian (pelaku proses), proses (manual / berbasis komputer) dan aliran data (dalam bentuk keluaran dan masukan).

Adapun simbol-simbol yang terdapat dalam *flowmap* adalah :

Tabel 2. Simbol-Simbol *Flowmap*

|  |  |  |
| --- | --- | --- |
| **SIMBOL** | **NAMA** | **FUNGSI** |
|  | Dokumen | Menunjukkan dokumen berupa dokumen input dan output proses berbasis computer. |
|  | Proses Manual | Menunjukkan proses yang dilakukan secara manual. |
|  | Penyimpan Magnetik | Menunjukkan media penyimpanan data/informasi file pada proses berbasis komputer. |

|  |  |  |
| --- | --- | --- |
| **SIMBOL** | **NAMA** | **FUNGSI** |
|  | Arah Alir Dokumen | Menunjukkan arah aliran dokumen antar bagian yang terkait pada suatu sistem. |
|  | Penghubung | Menunjukkan alir dokumen yang terputus atau terpisah pada halaman alur dokumen yang sama. |
|  | Proses Komputer | Menunjukkan proses yang dilakukan secara komputerisasi. |
|  | Pengarsipan | Menunjukkan simpanan data non komputer / informasi file pada proses manual. |
|  | Input Keyboard | Menunjukkan *input* yang dimasukkan melalui *keyboard*. |
|  | Penyimpanan Data | Menunjukkan media penyimpanan data. |

## Implementasi Teori Pada Aplikasi

Tabel 2. Tabel Implementasi Teori Pada Aplikasi

| **No** | **Nama** | **Penggunaan** | **Keterangan** | **Contoh** |
| --- | --- | --- | --- | --- |
| 1 | HTML | Sebagian | Sebagai dasar *layout* untuk halaman. | Penggunaan *tag-tag* “<div>” untuk membuat *layou*t di semua halaman, dan lain-lain. |
| 2 | CSS | Sebagian | Sebagai pengaturan tampilan berdasarkan *layout* yang dibuat menggunakan HTML. | Penggunaan atribut “class” dan “id” pada hampir seluruh tag yang ada dalam sebuah *file* untuk mengimplementasikan *style* yang diinginkan sesuai dengan *layout* yang sudah dibuat. |
| 3 | Javascript | Sebagian | Mengatur interaksi aplikasi. | Sebagai *alert* saat mengkonfirmasi pengolahan data. (Tambah, ubah dan hapus) |
| 4 | PHP | Semua *file* | Sebagai pengolah data pada sistem. | Penggunaan perintah PHP pada hampir seluruh file berekstensi “.php” seperti “echo”, “foreach”, dan sebagainya. |
| 5 | MySQL | Semua *file* | Sebagai pengolah data pada *database*. | Perintah MySQL seperti “Select”, “Insert”, “Update”, dan “Delete” digunakan untuk mengolah data yang digunakan pada aplikasi. |
| 6 | Laravel | Semua *file* | *Framework* untuk membuat sistem semakin mudah. | Perintah PHP juga MySQL yang digunakan di aplikasi ini menggunakan “*template*” yang sudah disediakan Laravel. |
| 7 | Bootstrap | Sebagian | *Framework* untuk membuat tampilan semakin mudah. | *Layout* pada aplikasi ini dibuat agar sesuai dengan keinginan menggunakan *style* yang sudah dibuat sebelumnya di CSS. |

# BAB III ANALISA DAN PERANCANGAN

## Analisa Aplikasi

Aplikasi Inventaris Sarana Prasarana adalah aplikasi berbasis website yang tengah dikembangkan oleh SMK Negeri 1 Cimahi. Seperti yang telah dijelaskan pada BAB I Pendahuluan, aplikasi ini dikembangkan menggunakan PHP *Framework* Laravel dan berfokus pada pengelolaan sarana dan prasarana sekolah khususnya untuk SMK Negeri 1 Cimahi.

Adapun metode yang digunakan dalam pengembangan aplikasi ini adalah metode *Rapid Application Development* (RAD) yaitu sebuah metode yang dibutuhkan untuk mendesain serta mengimplementasikan sistem sehingga dihasilkan siklus pengembangan yang sangat pendek. Metode RAD ini memiliki beberapa fase diantaranya yang pertama yaitu *business modelling* atau analisis yang bertujuan untuk menentukan aliran informasi dan bisnis dari proses aplikasi yang dibuat. Fase selanjutnya yaitu *data modelling* atau perancangan yang bertujuan untuk memilih data-data yang dibutuhkan untuk menopang proses pembuatan aplikasi. Selanjutnya yaitu fase *process modelling* yang bertujuan untuk mentransformasikan data pada *data modelling* agar dapat diimplementasikan pada proses bisnis yang telah dibuat pada proses *business modelling*.

Fase selanjutnya yang digunakan dalam metode RAD adalah *application generation* atau pengodean, pada metode RAD pengodean tentunya diharapkan dilaksanakan dengan waktu yang cepat. Maka dari itu pada metode RAD lebih banyak melakukan proses pengodean dengan menggunakan alat-alat bantu otomatis untuk memfasilitasi pembuatan aplikasi. Dalam hal ini penulis menggunakan PHP *framework* yaitu Laravel versi 8.12.3. Fase terakhir pada metode RAD yakni *testing and turnover* atau pengujian yang bertujuan untuk menguji aplikasi yang telah dibuat selain itu pada tahap ini menekankan pada *reuse* atau pemakaian kembali agar dapat mengurangi waktu pengujian.

## Analisa Pengguna Aplikasi

Aplikasi ini memberikan beberapa hak akses kepada para pengguna atau *user* untuk membatasi perilaku pengguna dalam pemakaian aplikasi ini.

* 1. **Admin**

*Admin* adalah pengguna yang memiliki hak tertinggi dengan kata lain *admin* dapat mengolah data yang dibutuhkan dalam sistem. Semua hal terkait dengan data master dapat diolah oleh *Admin*, termasuk di dalamnya data pengguna lainnya.

* 1. **Pegawai**

Pegawai adalah pengguna yang memiliki hak untuk mengolah data dalam sistem. Semua hal terkait dengan data master dapat diolah oleh Pegawai, namun Pegawai tidak dapat mengelola data *user* atau data pengguna lainnya seperti hak akses yang dimiliki oleh *Admin*.

## Analisa Kebutuhan Pengguna

Adapun sesuai dengan hak akses maka kebutuhan yang diperlukan berbeda-beda. Berikut merupakan analisa kebutuhan setiap hak akses.

### Analisa Kebutuhan Admin

Berikut adalah kebutuhan dari level *user* *admin* aplikasi Inventarisasi Sarana dan Prasarana SMK Negeri 1 Cimahi ini :

1. Sistem dapat mengolah data *user*.
2. Sistem dapat mengolah data aset.
3. Sistem dapat mengolah data ruangan.
4. Sistem dapat mengolah data sarana yang dipinjam.

### Analisa Kebutuhan Pegawai

Berikut adalah kebutuhan dari level *user* pegawai aplikasi Inventarisasi Sarana dan Prasarana SMK Negeri 1 Cimahi ini :

1. Sistem dapat mengolah data aset.
2. Sistem dapat mengolah data ruangan.
3. Sistem dapat mengolah data sarana yang dipinjam.

## Perancangan Sistem

Perancangan Aplikasi Inventarisasi Sarana dan Prasarana SMK Negeri 1 Cimahi menggunakan Laravel berbasis Web ini meliputi perancangan *flowmap*, *Data Flow Diagram*, perancangan basis data, serta *layout* aplikasi.

### Flowmap

*Flowmap* adalah penggambaran alur dari suatu program berupa langkah-langkah dan urutan prosedur penggunaan program yang ditampilkan dalam bentuk grafik. *Flowmap* memiliki bentuk dan fungsi yang mirip dengan *flowchart*, yaitu menjelaskan hubungan dari komponen utama sistem, yaitu antara pengguna, proses (berupa manual / pada sistem komputer), dan alur data yang terjadi.

1. **Flowmap Login**

*Flowmap* dibawah ini merupakan gambaran proses *login* yang terdapat dalam aplikasi ini. Baik itu sebagai admin, ataupun sebagai pegawai.

|  |  |
| --- | --- |
| Login | |
| User | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Login

1. **Flowmap Tambah User**

*Flowmap* dibawah ini merupakan gambaran proses penambahan *user* yang terdapat dalam aplikasi ini. Proses penambahan data user yang hanya dapat dilakukan oleh Admin.

|  |  |
| --- | --- |
| Penambahan User | |
| Admin | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Tambah User

1. **Flowmap Edit User**

*Flowmap* dibawah ini merupakan gambaran proses pengubahan *user* yang terdapat dalam aplikasi ini. Proses pengubahan data user hanya dapat dilakukan oleh Admin.

|  |  |
| --- | --- |
| Edit User | |
| Admin | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Edit User

1. **Flowmap Tambah Aset**

*Flowmap* dibawah ini merupakan gambaran proses penambahan asetyang terdapat dalam aplikasi ini. Proses penambahan aset dapat dilakukan oleh Admin dan Pegawai.

|  |  |
| --- | --- |
| Tambah Aset | |
| Admin / Pegawai | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Tambah Aset

1. **Flowmap Edit Aset**

*Flowmap* dibawah ini merupakan gambaran proses pengubahan data asetyang terdapat dalam aplikasi ini. Proses pengubahan data aset dapat dilakukan oleh Admin dan Pegawai.

|  |  |
| --- | --- |
| Edit Aset | |
| Admin / Pegawai | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Edit Aset

1. **Flowmap Tambah Ruangan**

*Flowmap* dibawah ini merupakan gambaran proses penambahan ruanganyang terdapat dalam aplikasi ini. Proses penambahan ruangan dapat dilakukan oleh Admin dan Pegawai.

|  |  |
| --- | --- |
| Tambah Ruangan | |
| Admin / Pegawai | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Tambah Ruangan

1. **Flowmap Edit Ruangan**

*Flowmap* dibawah ini merupakan gambaran proses pengubahan data ruanganyang terdapat dalam aplikasi ini. Proses pengubahan data ruangan dapat dilakukan oleh Admin dan Pegawai.

|  |  |
| --- | --- |
| Edit Ruangan | |
| Admin / Pegawai | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Edit Ruangan

1. **Flowmap Tambah Transaksi**

*Flowmap* dibawah ini merupakan gambaran proses penambahan transaksiyang terdapat dalam aplikasi ini. Proses penambahan transaksi dapat dilakukan oleh Admin dan Pegawai.

|  |  |
| --- | --- |
| Tambah Transaksi | |
| Admin / Pegawai | Sistem |
|  |  |

Gambar 3. Flowmap Sistem Tambah Transaksi

1. **Flowmap Edit Transaksi**

*Flowmap* dibawah ini merupakan gambaran proses pengubahan data transaksiyang terdapat dalam aplikasi ini. Proses pengubahan data transaksi dapat dilakukan oleh Admin dan Pegawai.

|  |  |
| --- | --- |
| Edit Transaksi | |
| Admin / Pegawai | Sistem |
|  |  |

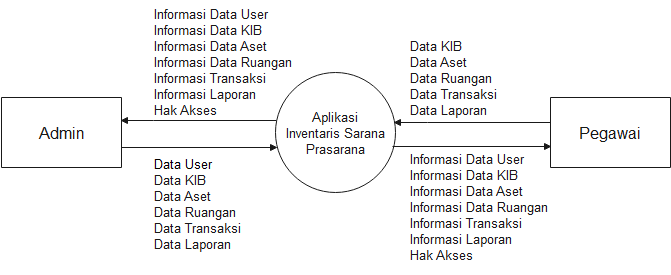
Gambar 3. Flowmap Sistem Edit Transaksi

### Data Flow Diagram

*Data Flow Diagram* dibawah ini menggambarkan beberapa proses arus data yang terdapat dalam Aplikasi Inventarisasi Sarana dan Prasarana menggunakan Laravel berbasis Web.

1. **DFD Level 0 (Diagram Konteks)**

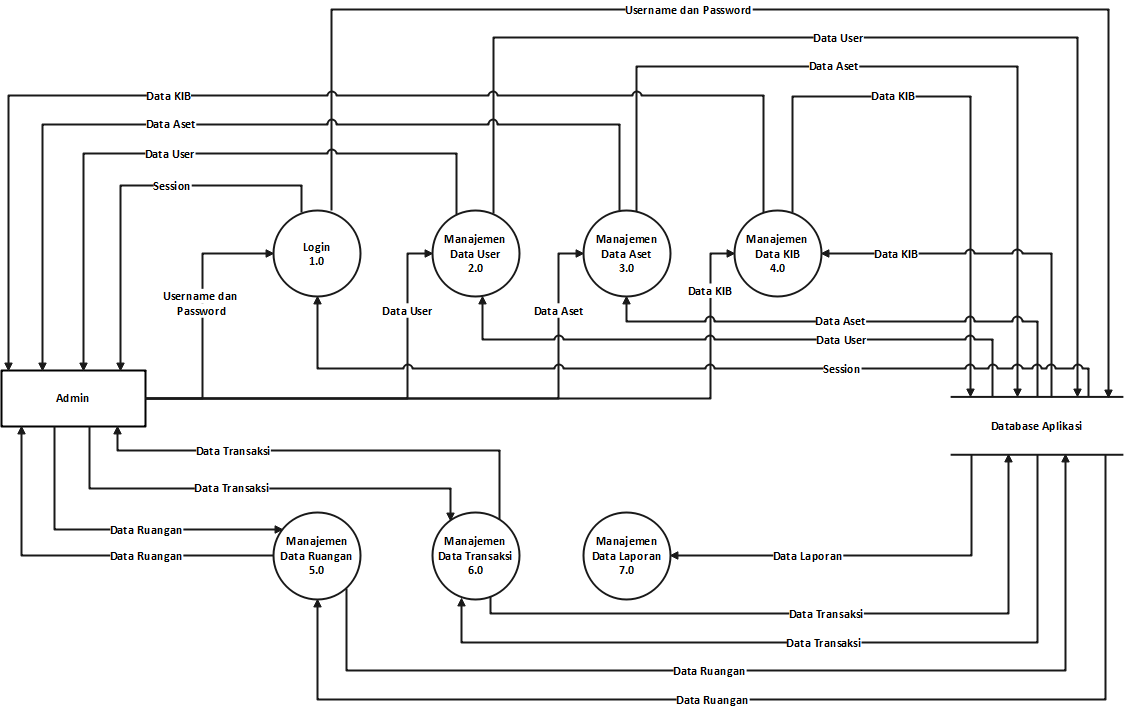
*Data Flow Diagram* atau diagram konteks dibawah ini menggambarkan alur data yang terdapat dalam aplikasi dalam satu sistem tunggal.



Gambar 3. DFD Level 0

1. **DFD Level 1 – Admin**

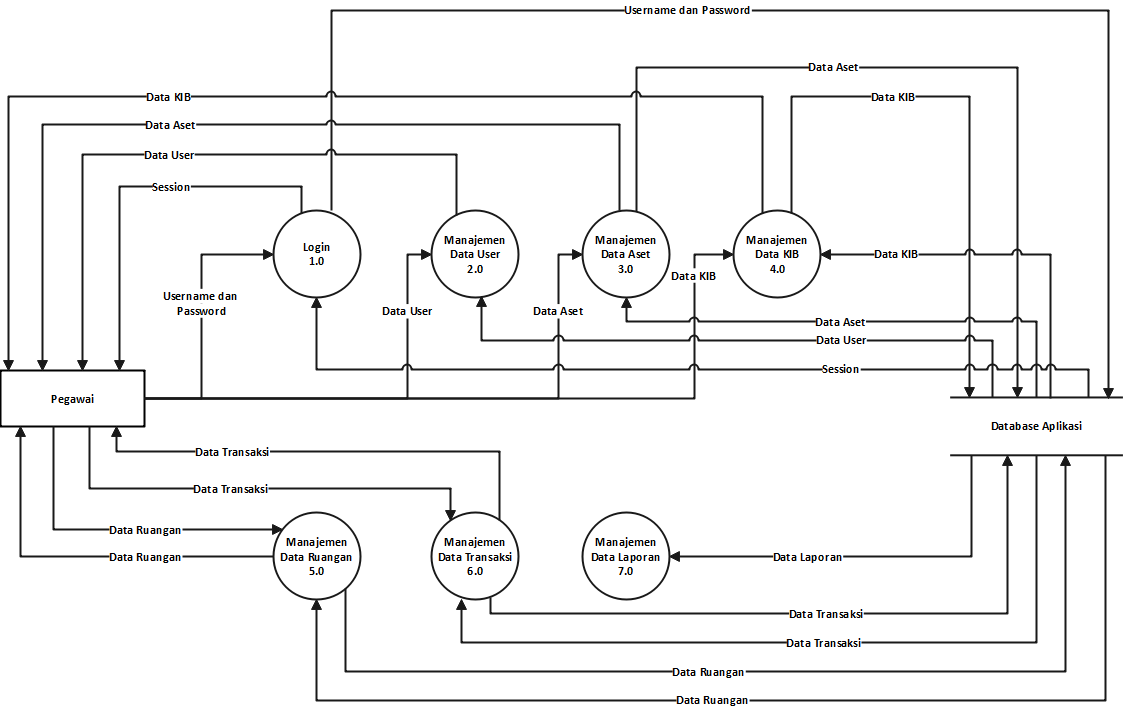
Diagram level 1 untuk Admin ini merupakan diagram yang akan menjabarkan hal-hal apa saja yang terdapat dalam DFD level 0 atau Diagram Konteks secara lebih mendetail.



Gambar 3. DFD Level 1 - Admin

1. **Diagram Level 1 - Pegawai**

Diagram level 1 untuk Pegawai ini merupakan diagram yang akan menjabarkan hal-hal apa saja yang terdapat dalam DFD level 0 atau Diagram Konteks secara lebih mendetail.



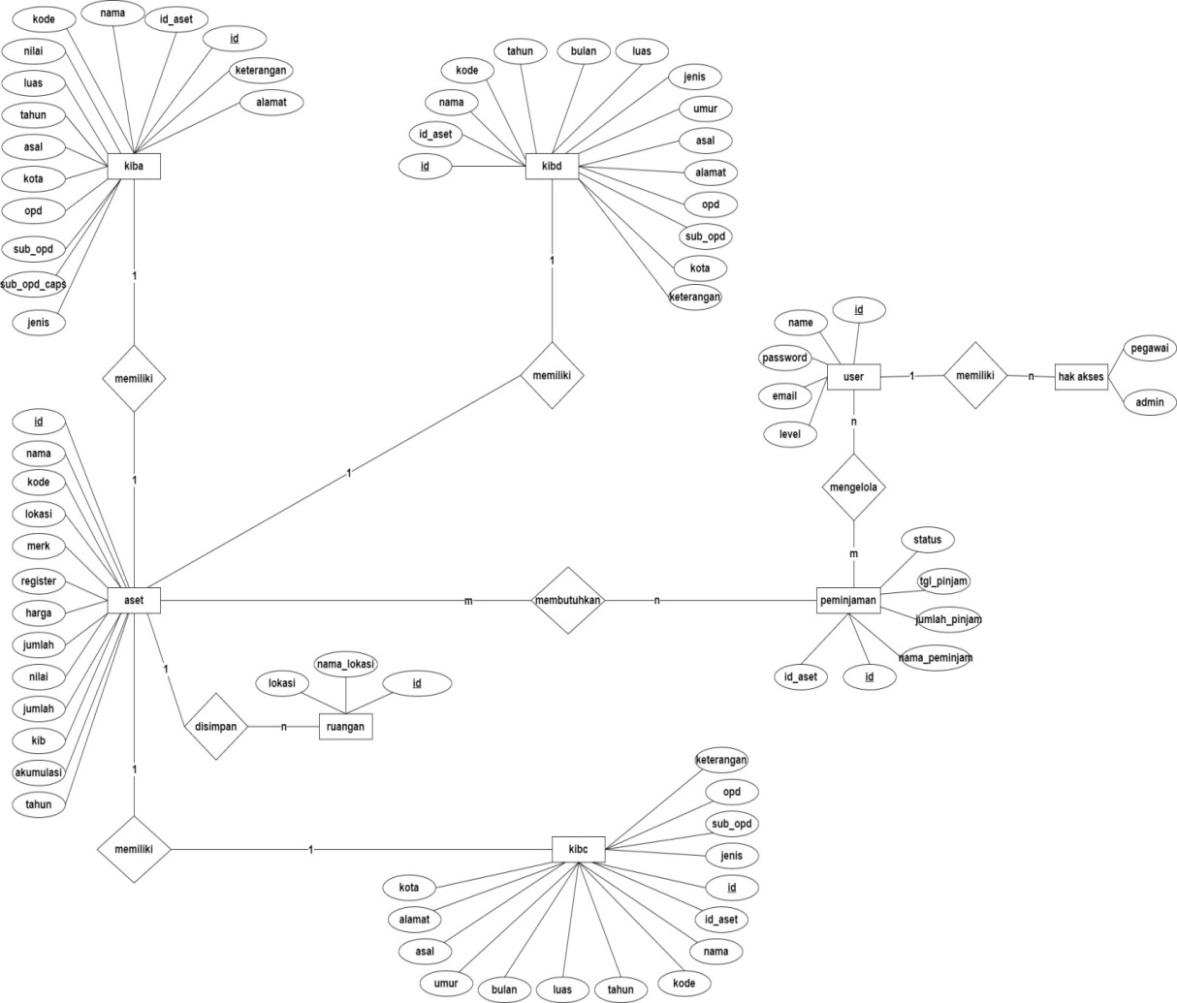
Gambar 3. DFD Level 1 - Pegawai

### Rancangan Basis Data

Perancangan basis data sangat diperlukan dalam setiap pembuatan aplikasi. Dengan menggunakan basis data (*database*) membuat penyimpanan data pada aplikasi yang digunakan akan semakin efektif dan efisien sehingga dapat menghindari data yang redudan atau nilai yang kosong.

1. **Entity Relationship Diagram**

*Entity Relationship Diagram* dibawah ini merupakan pemodelan kebutuhan data dari Aplikasi Inventarisasi Sarana dan Prasarana menggunakan Laravel berbasis Web.

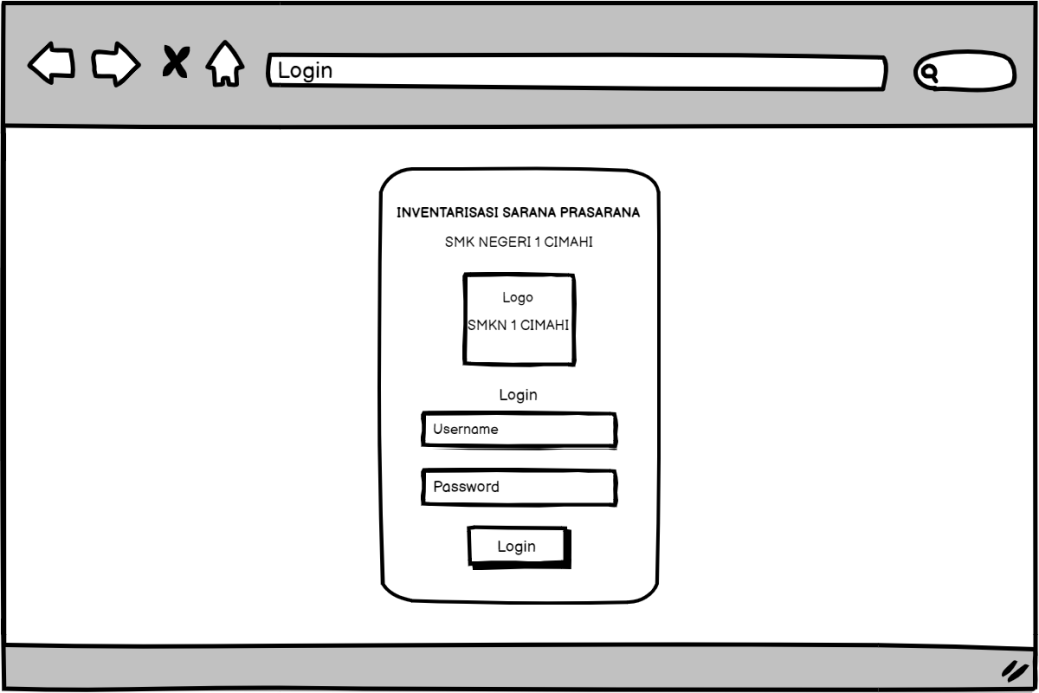


Gambar 3. Entity Relationship Diagram

### Perancangan Layout Halaman

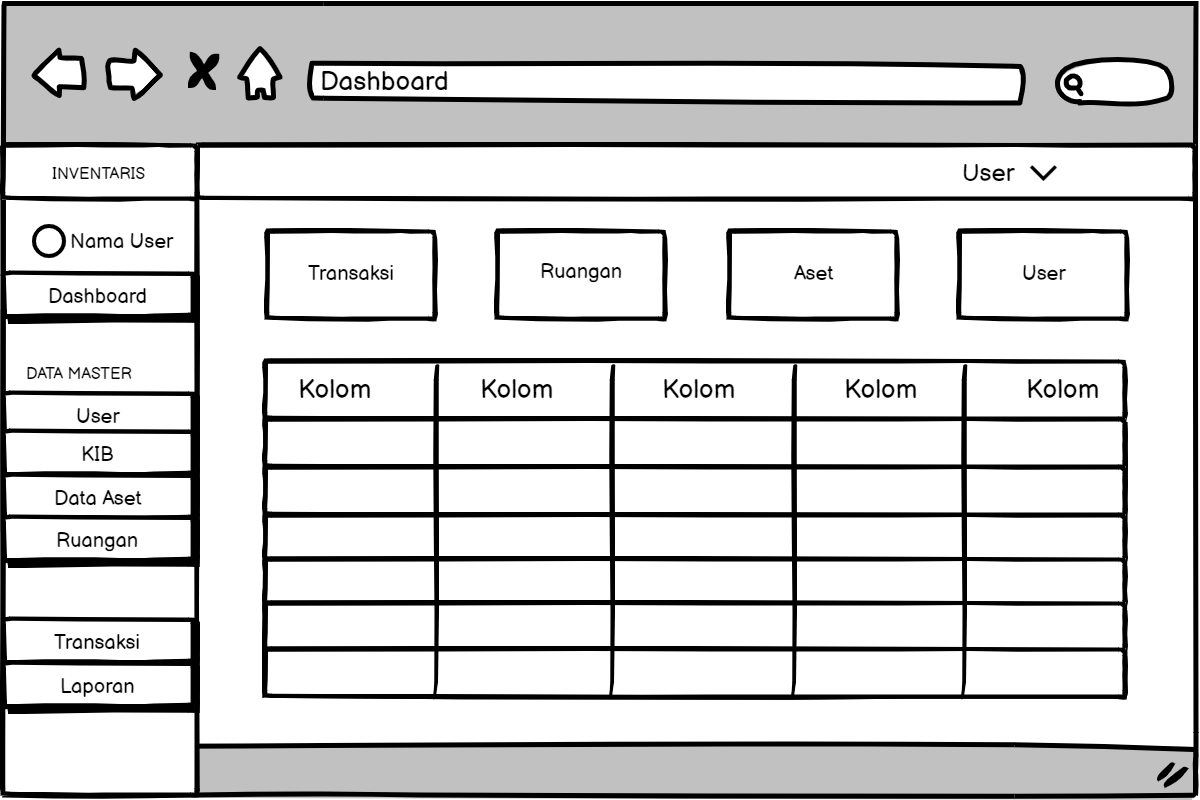
Desain *layout* dibawah ini menggambarkan letak dan posisi konten yang terdapat dalam aplikasi.

1. **Halaman Login**

****

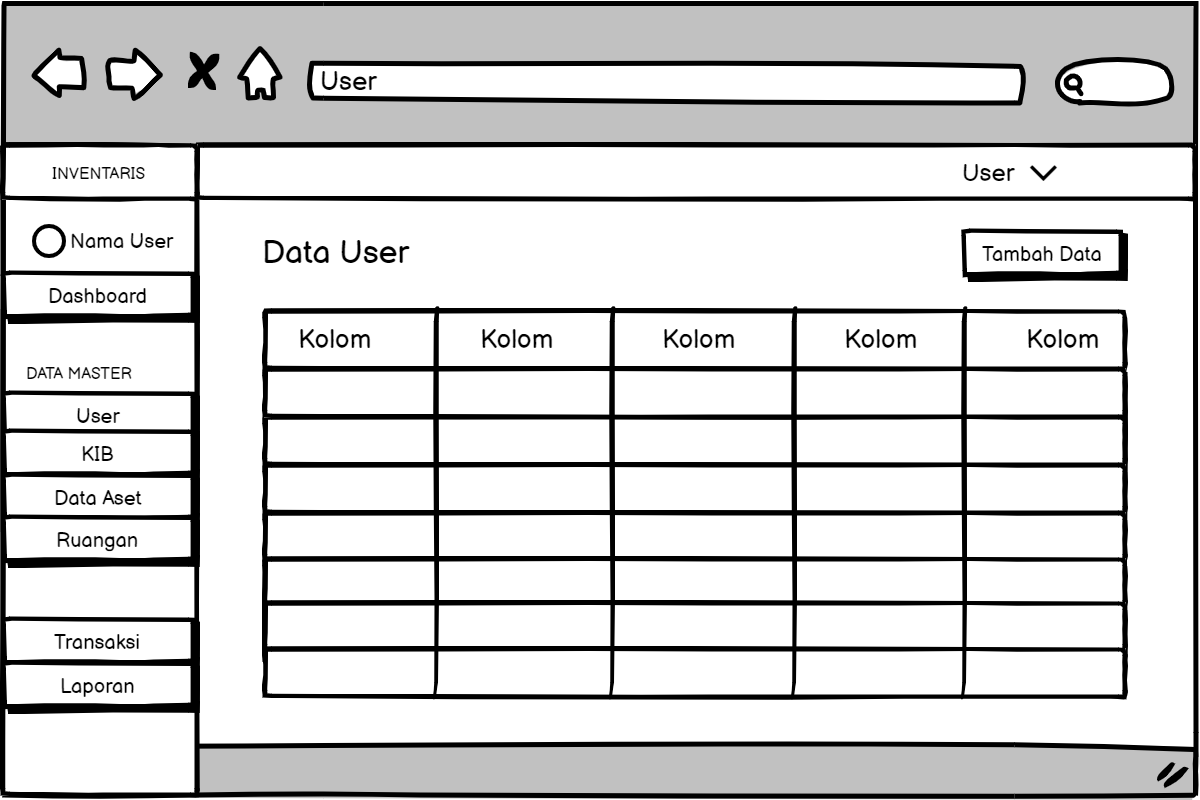
Gambar 3. Desain Layout Halaman Login

1. **Halaman Dashboard**

****

Gambar 3. Desain Layout Halaman Dashboard

1. **Halaman User**

****

Gambar 3. Desain Layout Halaman User

# BAB IV IMPLEMENTASI DAN PENGUJIAN

## Implementasi

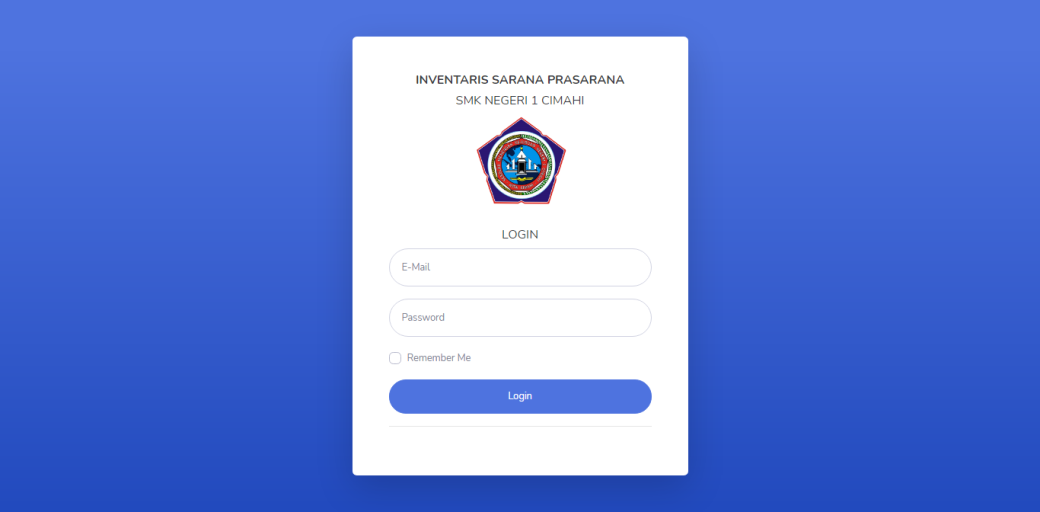
Setelah proses pembuatan aplikasi selesai, aplikasi ini dengan sistem *database*-nya dapat diimplementasikan, dan diharapkan akan memenuhi tujuan pembuatan aplikasi.

### Desain Front End / Tampilan

Berikut beberapa tampilan antarmuka yang ada dalam aplikasi Inventaris Sarana dan Prasarana SMK Negeri 1 Cimahi.

1. **Halaman Login**

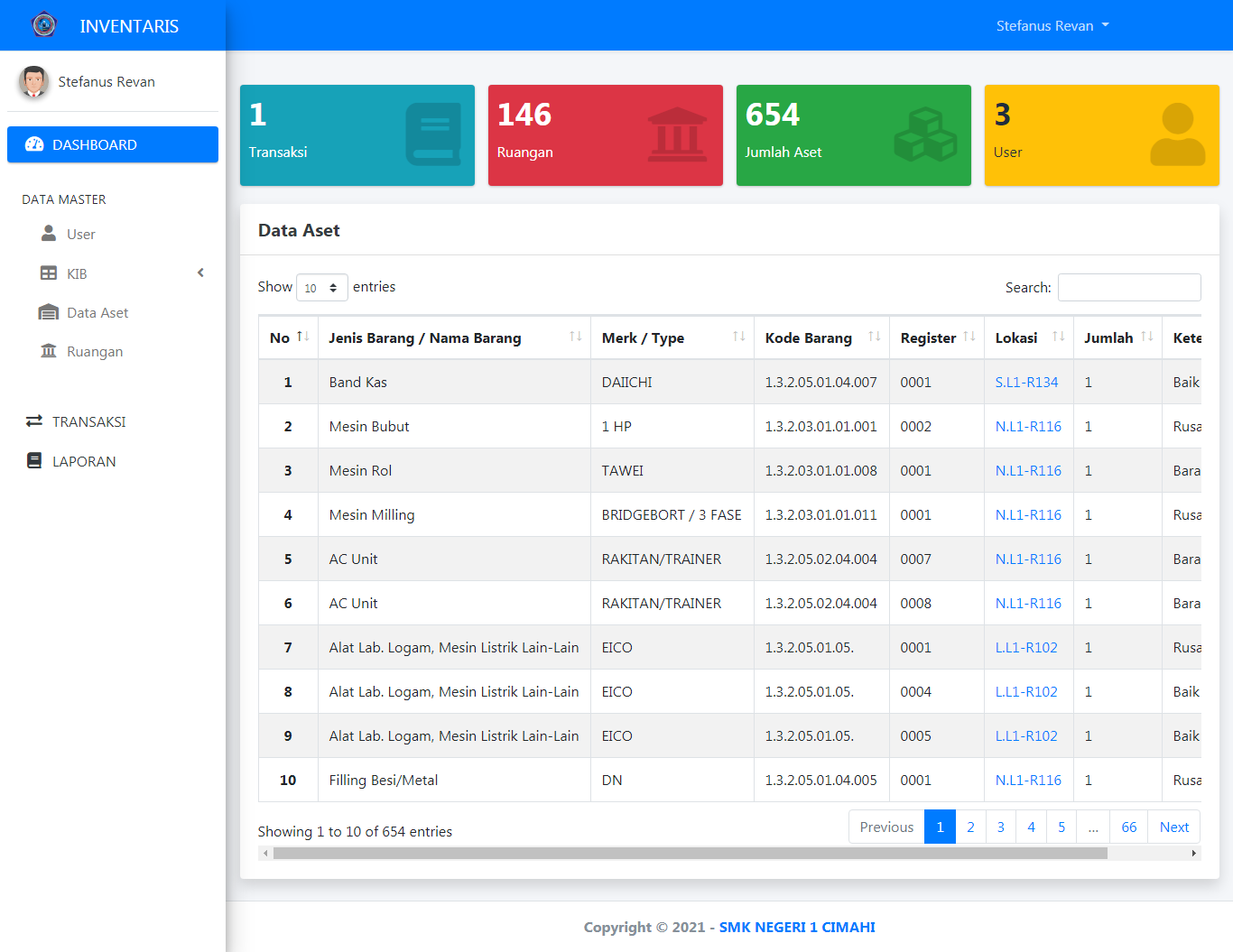
Halaman *login* terdiri atas judul aplikasi dan *form* untuk melakukan proses *login*. *Form* ini memiliki tiga komponen, yaitu *username*, *password*, dan juga tombol *submit*.



Gambar 4. Halaman Login

1. **Halaman *Dashboard***

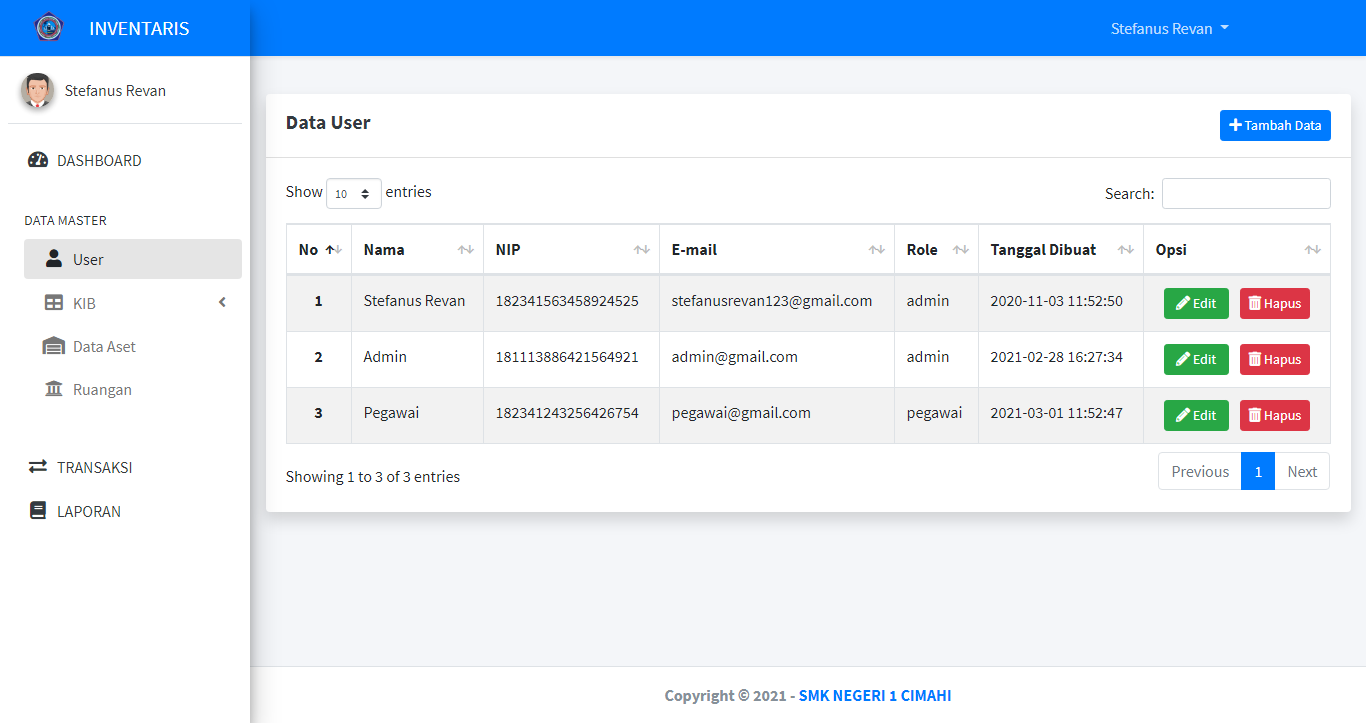
Halaman *dashboard* menampilkan total data transaksi, data ruangan, data aset, dan data *user*. Gambar dibawah ini merupakan halaman *dashboard* admin dan pegawai.



Gambar 4. Halaman Dashboard

1. **Halaman *User***

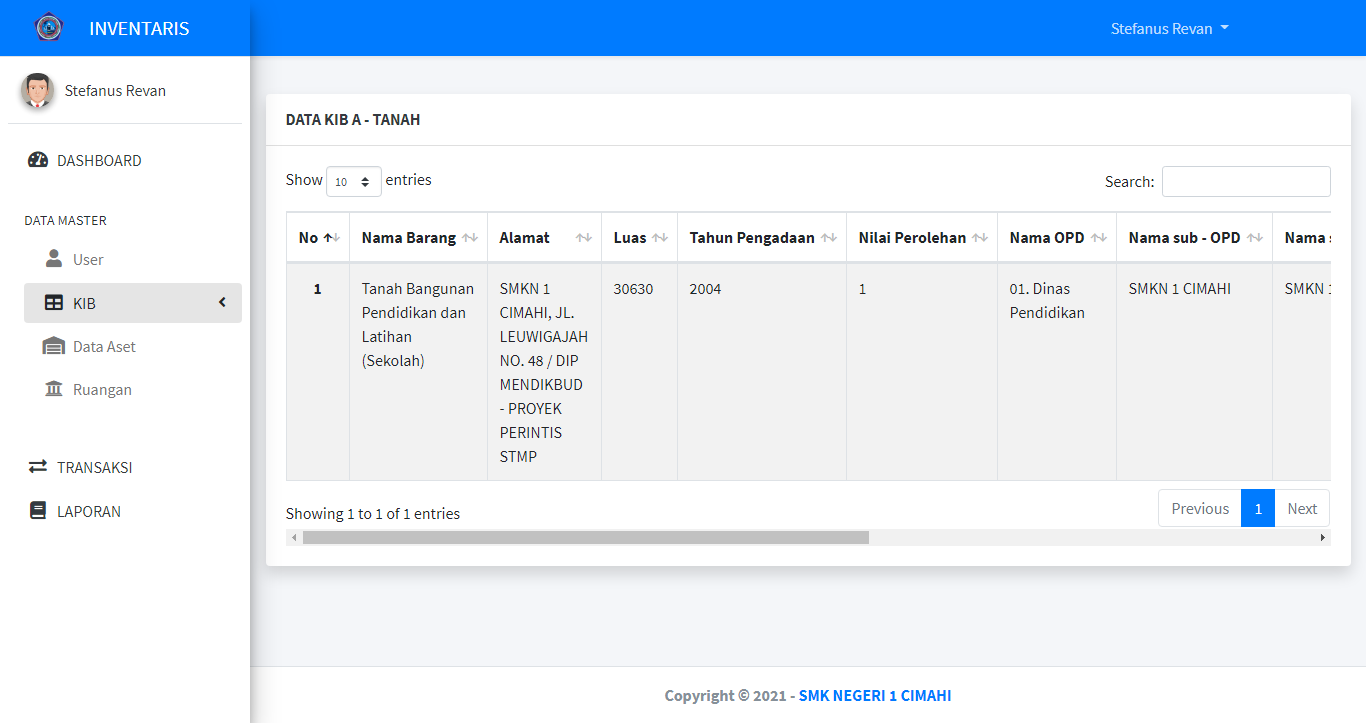
Halaman *user* ini berisikan data-data *user* yang sudah terdapat di dalam database aplikasi ini, *user* yang dimaksud merupakan daftar admin ataupun pegawai. Di halaman ini juga terdapat tombol tambah data yang akan menuju halaman form tambah data untuk menambah data *user* dan juga terdapat tombol *edit* dan hapus data. Gambar dibawah ini merupakan halaman *user* admin.

****

Gambar 4. Halaman User

1. **Halaman KIB A**

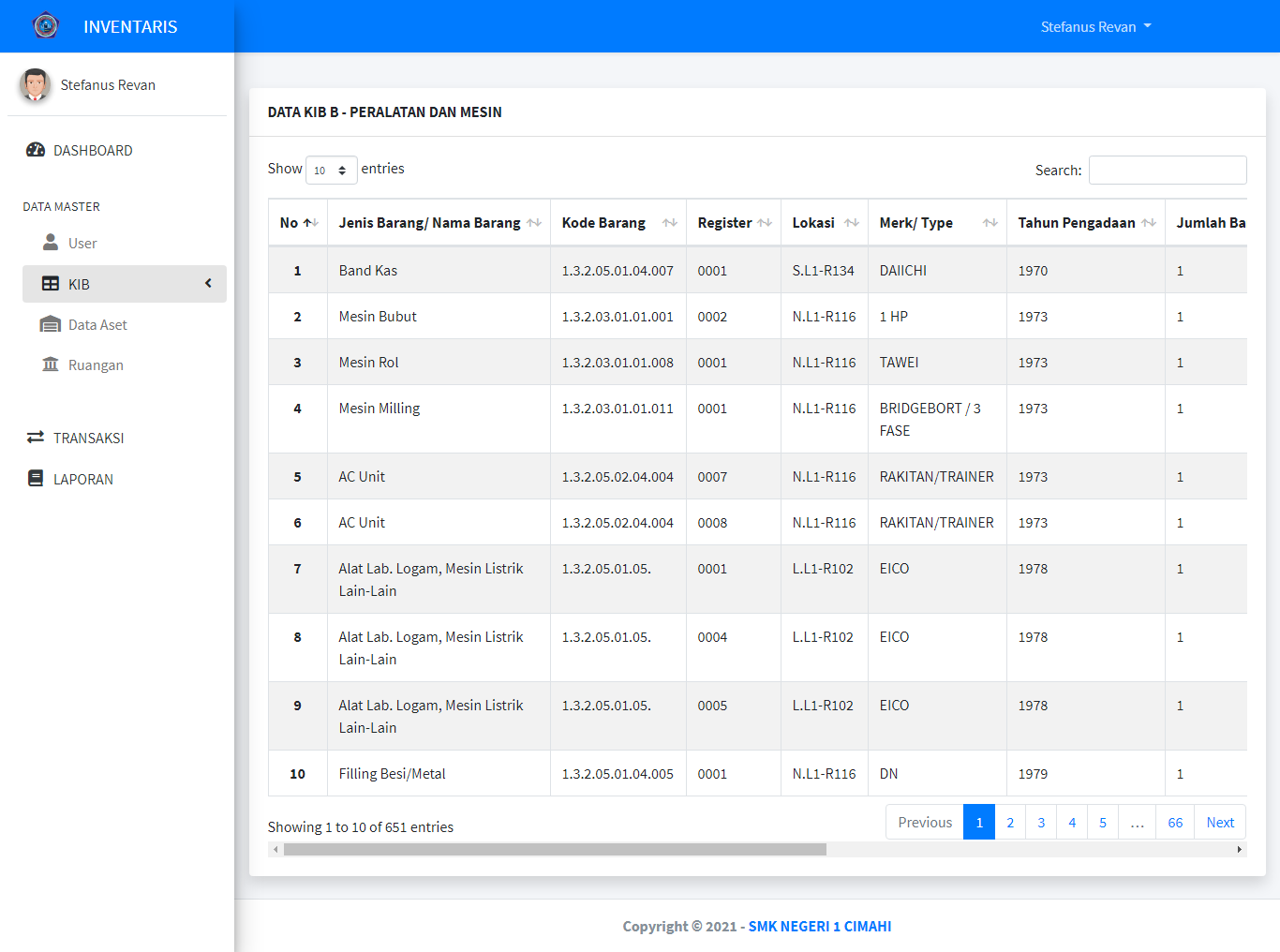
Halaman KIB A atau Kategori Inventaris Barang Aini berisikan data-data tanah yang terdapat di dalam database aplikasi ini. Gambar dibawah ini merupakan tampilan dari halaman KIB A.



Gambar 4. Halaman KIB A

1. **Halaman KIB B**

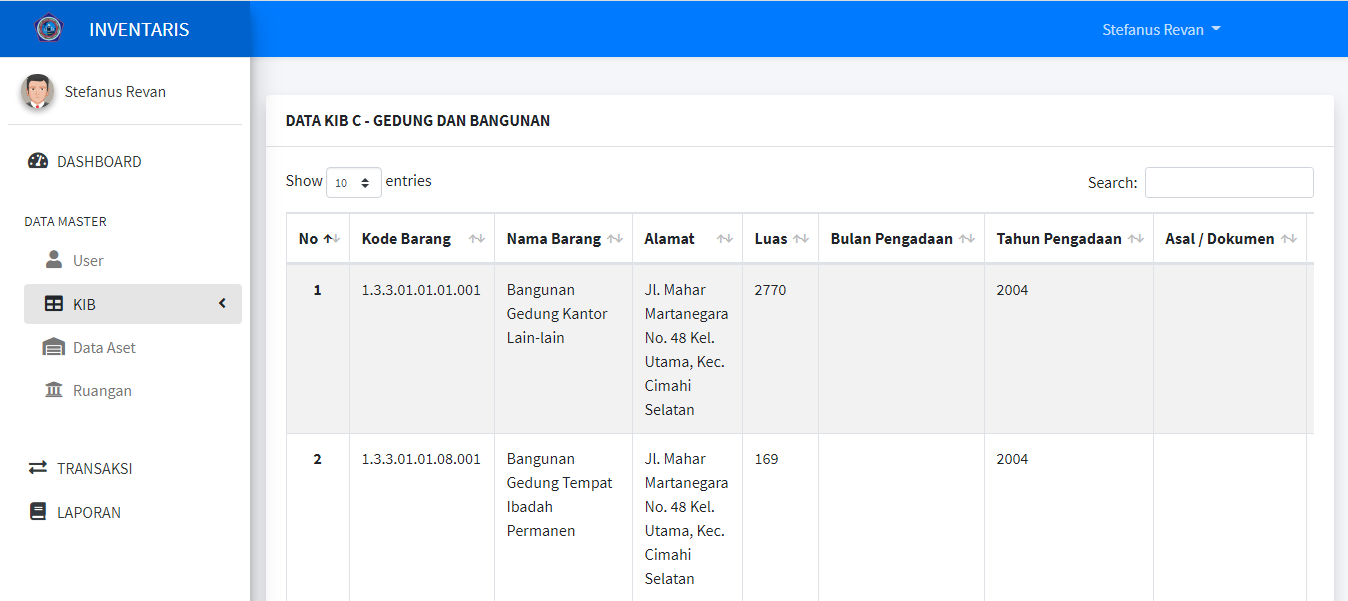
Halaman KIB B atau Kategori Inventaris Barang Bini berisikan data-data peralatan dan mesin yang terdapat di dalam database aplikasi ini. Gambar dibawah merupakan tampilan dari halaman KIB B.



Gambar 4. Halaman KIB B

1. **Halaman KIB C**

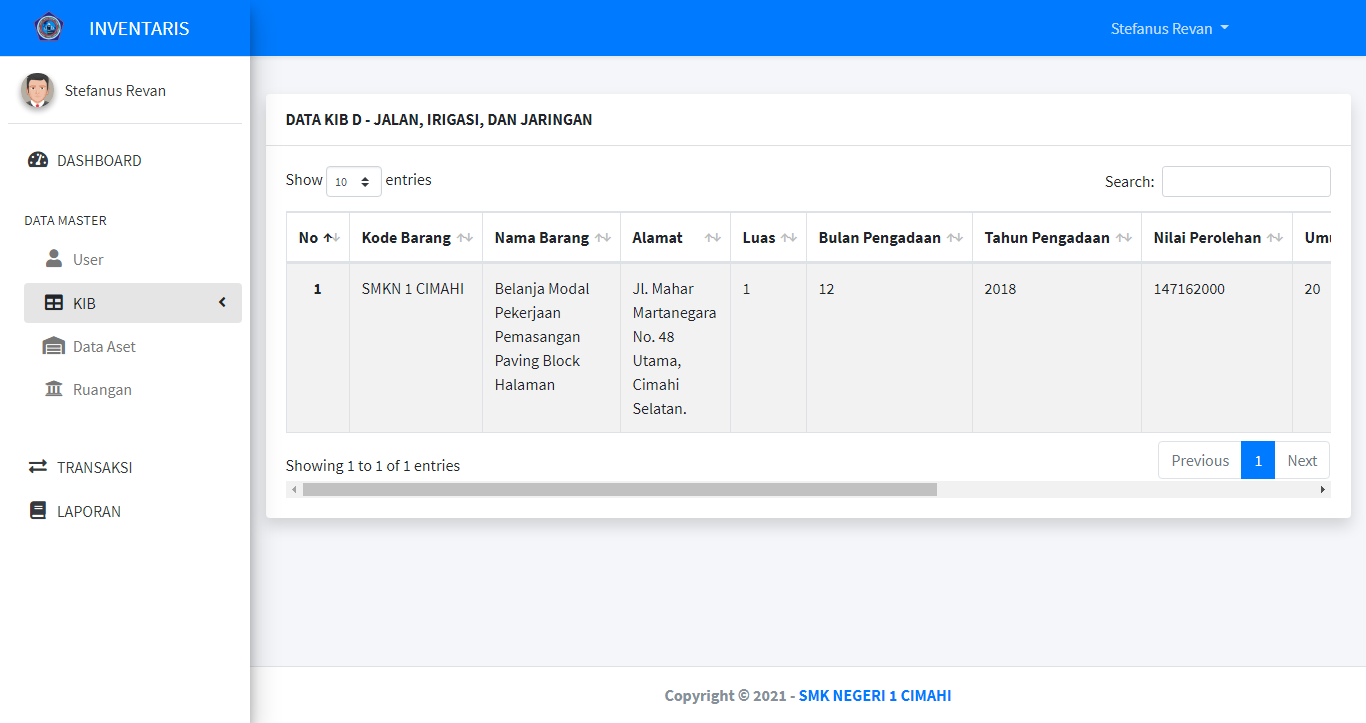
Halaman KIB C atau Kategori Inventaris Barang C ini berisikan data-data gedung dan bangunan yang terdapat di dalam database aplikasi ini. Gambar dibawah ini merupakan tampilan dari halaman KIB C.



Gambar 4. Halaman KIB C

1. **Halaman KIB D**

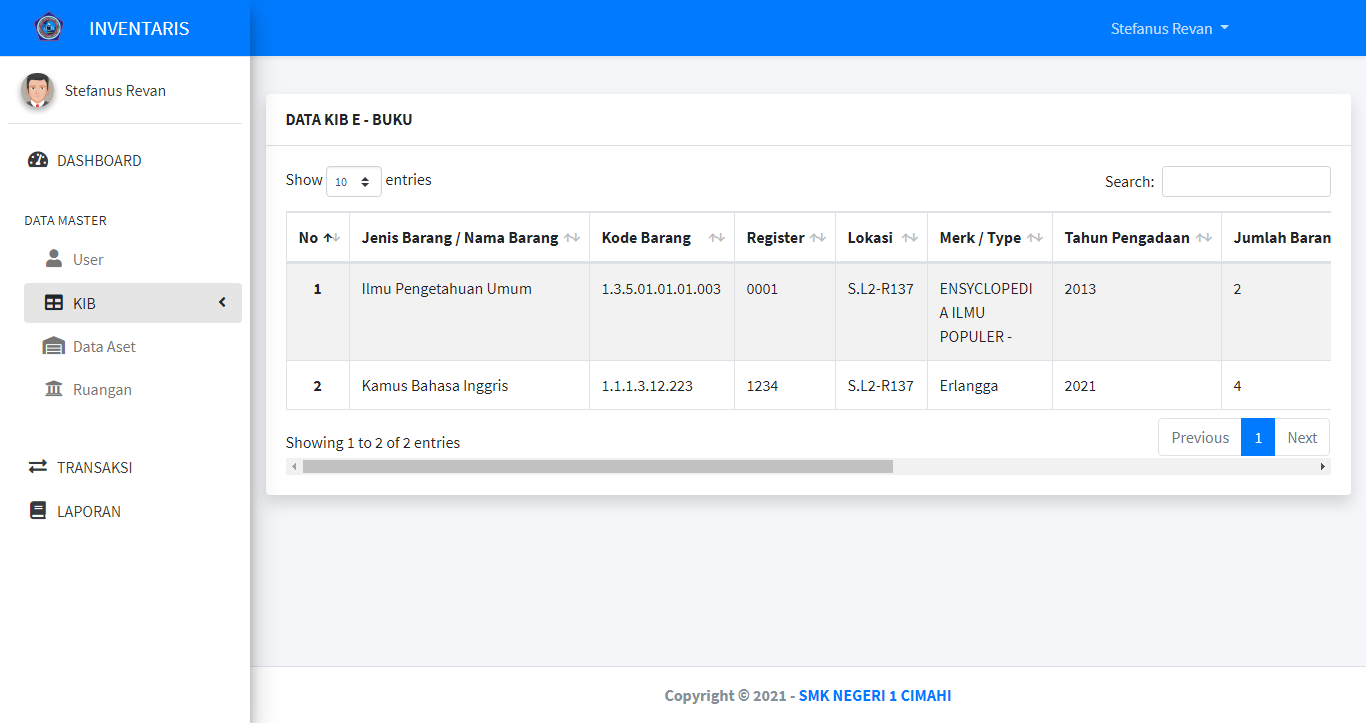
Halaman KIB D atau Kategori Inventaris Barang D ini berisikan data-data jalan, irigasi dan jaringan yang terdapat di dalam database aplikasi ini. Gambar dibawah ini merupakan tampilan dari halaman KIB D.

****

Gambar 4. Halaman KIB D

1. **Halaman KIB E**

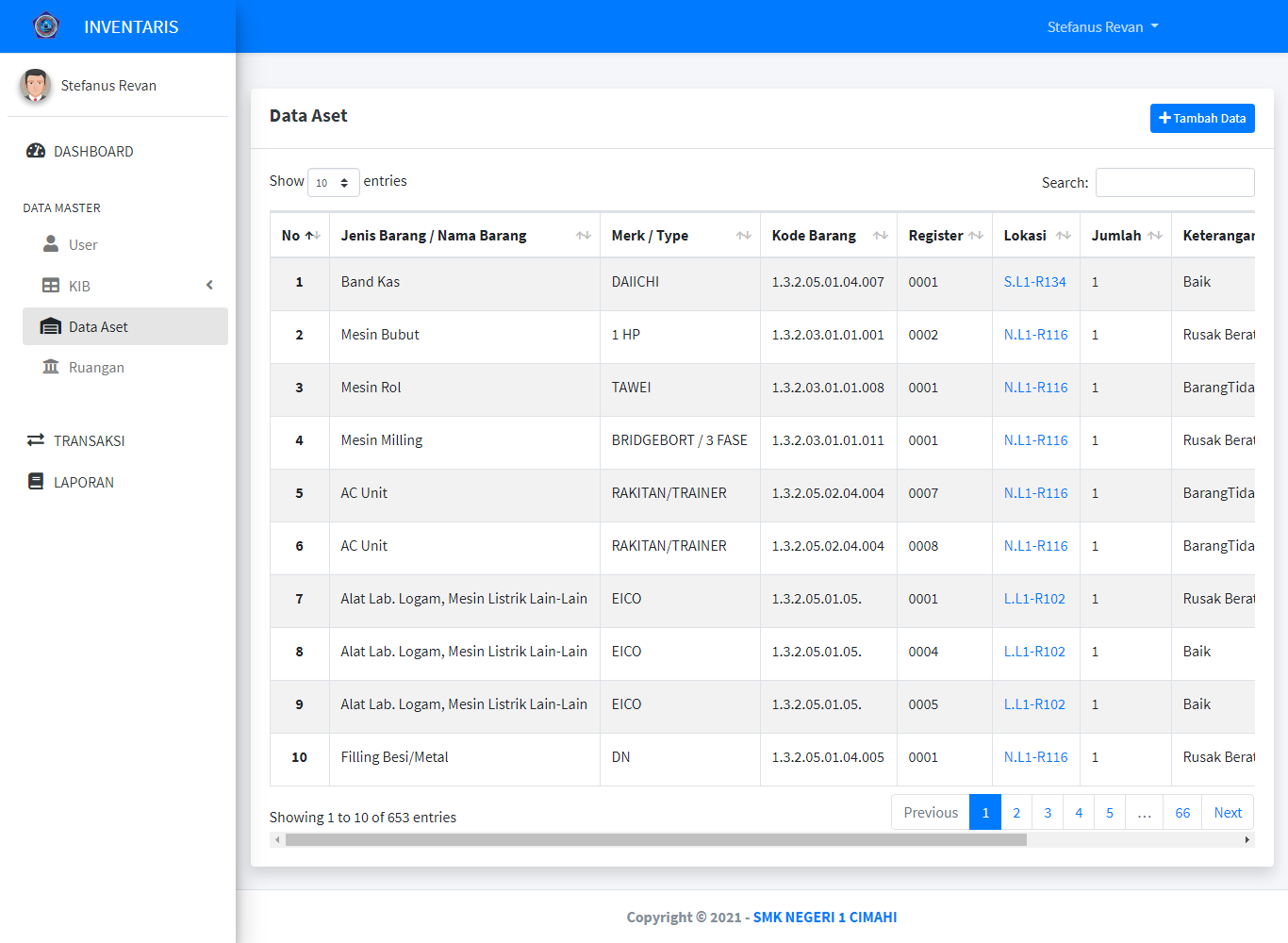
Halaman KIB E atau Kategori Inventaris Barang E ini berisikan data-data buku yang terdapat di dalam database aplikasi ini. Gambar dibawah ini merupakan tampilan dari halaman KIB E.



Gambar 4. Halaman KIB E

1. **Halaman Data Aset**

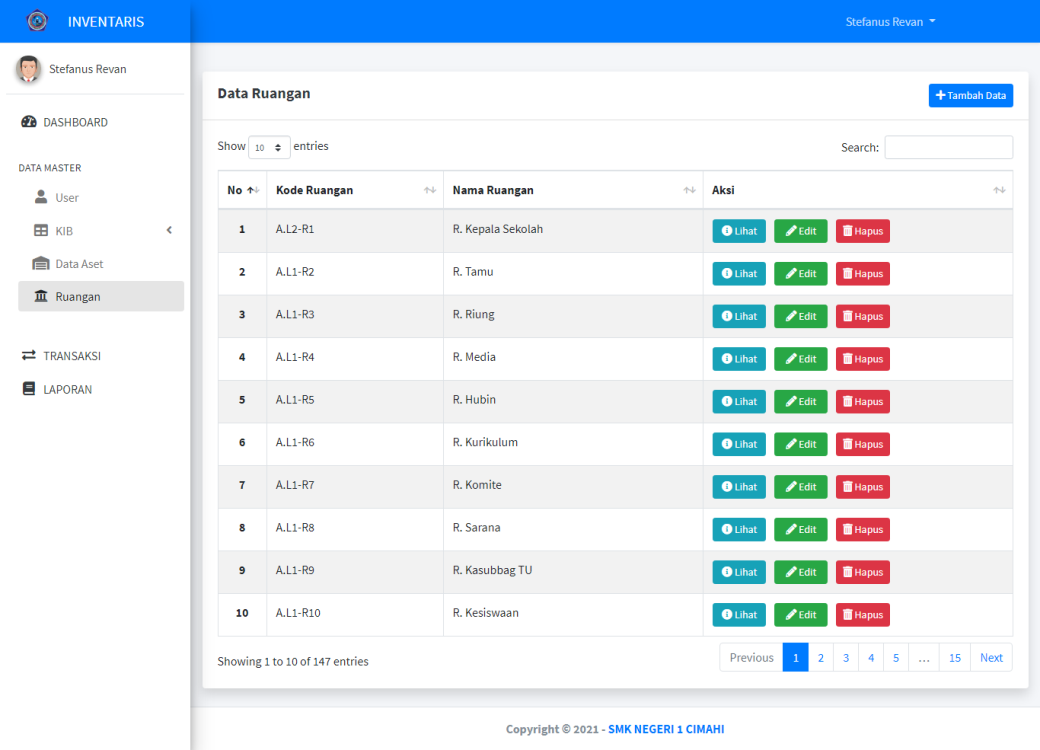
Halaman data aset ini berisikan keseluruhan data aset pada aplikasi ini. Pada halaman ini juga terdapat tombol tambah data, detail, edit dan hapus yang memungkinkan user untuk mengolah data aset. Gambar dibawah ini merupakan tampilan dari halaman data aset.



Gambar 4. Halaman Data Aset

1. **Halaman Ruangan**

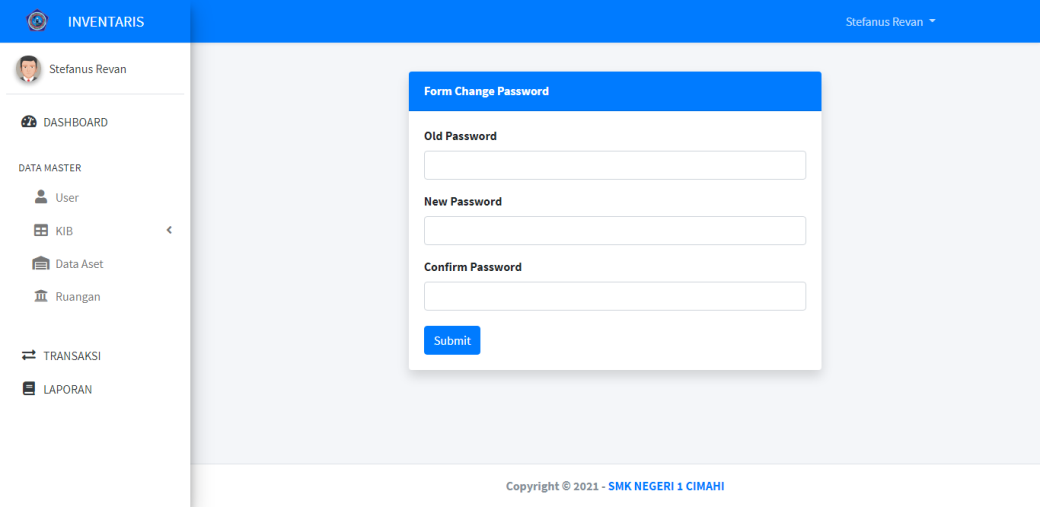
Halaman ruangan ini berisikan keseluruhan data ruangan pada aplikasi ini. Pada halaman ini juga terdapat tombol tambah data, lihat, edit dan hapus yang memungkinkan user untuk mengolah data ruangan. Gambar dibawah ini merupakan tampilan dari halaman ruangan.



Gambar 4. Halaman Ruangan

1. **Halaman Change Password**

Halaman *change password* ini dapat digunakan jika *user* ingin merubah password akunnya. Gambar dibawah ini merupakan tampilan dari halaman *change password*.



Gambar 4. Halaman Change Password

### Struktur Tabel

Struktur tabel dari aplikasi ini disusun menurut rancangan yang telah dibuat pada bab sebelumnya.

1. **Tabel User**

Tabel 4. Struktur Tabel Login

|  |  |  |
| --- | --- | --- |
| **Atribut / Kolom** | **Tipe Data** | **Constraint** |
| id | bigint (20) | Primary Key |
| name | varchar (255) | - |
| email | varchar (190) | - |
| email\_verified\_at | timestamp | - |
| password | varchar (255) | - |
| level | enum (‘admin’, ‘pegawai’) | - |
| remember\_token | varchar (100) | - |
| created\_at | timestamp | - |
| updated\_at | timestamp | - |

1. **Tabel KIB A**

Tabel 4. Struktur Tabel KIB A

|  |  |  |
| --- | --- | --- |
| **Atribut / Kolom** | **Tipe Data** | **Constraint** |
| id | bigint (20) | Primary Key |
| nama | varchar (255) | - |
| alamat | varchar (255) | - |
| luas | varchar (255) | - |
| tahun | char (4) | - |
| nilai | bigint (20) | - |
| opd | varchar (255) | - |
| sub\_opd | varchar (255) | - |
| sub\_opd\_caps | varchar (255) | - |
| kota | varchar (255) | - |
| asal | varchar (255) | - |
| keterangan | varchar (255) | - |
| kode | char (18) | - |
| jenis | varchar (255) |  |
| created\_at | timestamp | - |
| updated\_at | timestamp | - |

1. **Tabel KIB C**

Tabel 4. Struktur Tabel KIB C

|  |  |  |
| --- | --- | --- |
| **Atribut / Kolom** | **Tipe Data** | **Constraint** |
| id | bigint (20) | Primary Key |
| kode | char (18) | - |
| nama | char (255) | - |
| alamat | char (255) | - |
| luas | char (255) | - |
| bulan | char (255) | - |
| tahun | char (255) | - |
| asal | char (255) | - |
| umur | char (255) | - |
| opd | char (255) | - |
| sub\_opd | char (255) | - |
| kota | char (255) | - |
| keterangan | char (255) | - |
| created\_at | timestamp | - |
| updated\_at | timestamp | - |

1. **Tabel KIB D**

Tabel 4. Struktur Tabel KIB D

|  |  |  |
| --- | --- | --- |
| **Atribut / Kolom** | **Tipe Data** | **Constraint** |
| id | bigint (20) | Primary Key |
| kode | char (255) | - |
| nama | char (255) | - |
| alamat | char (255) | - |
| luas | char (255) | - |
| bulan | char (255) | - |
| tahun | char (255) | - |
| nilai | bigint (255) | - |
| umur | char (255) | - |
| opd | char (255) | - |
| sub\_opd | char (255) | - |
| kota | char (255) | - |
| keterangan | char (255) | - |
| created\_at | timestamp | - |
| updated\_at | timestamp | - |

1. **Tabel Aset**

Tabel 4. Struktur Tabel Aset

|  |  |  |
| --- | --- | --- |
| **Atribut / Kolom** | **Tipe Data** | **Constraint** |
| id | bigint (20) | Primary Key |
| nama | varchar (255) | - |
| kode | char (18) | - |
| register | char (4) | - |
| kib | enum (‘KIB B’, ‘KIB E’) | - |
| lokasi | varchar (190) | Foreign Key |
| merk | varchar (255) | - |
| tahun | char (4) | - |
| jumlah | int (11) | - |
| harga | bigint (20) | - |
| nilai | bigint (20) | - |
| akumulasi | int (11) | - |
| keterangan | varchar (255) | - |
| created\_at | timestamp | - |
| updated\_at | timestamp | - |

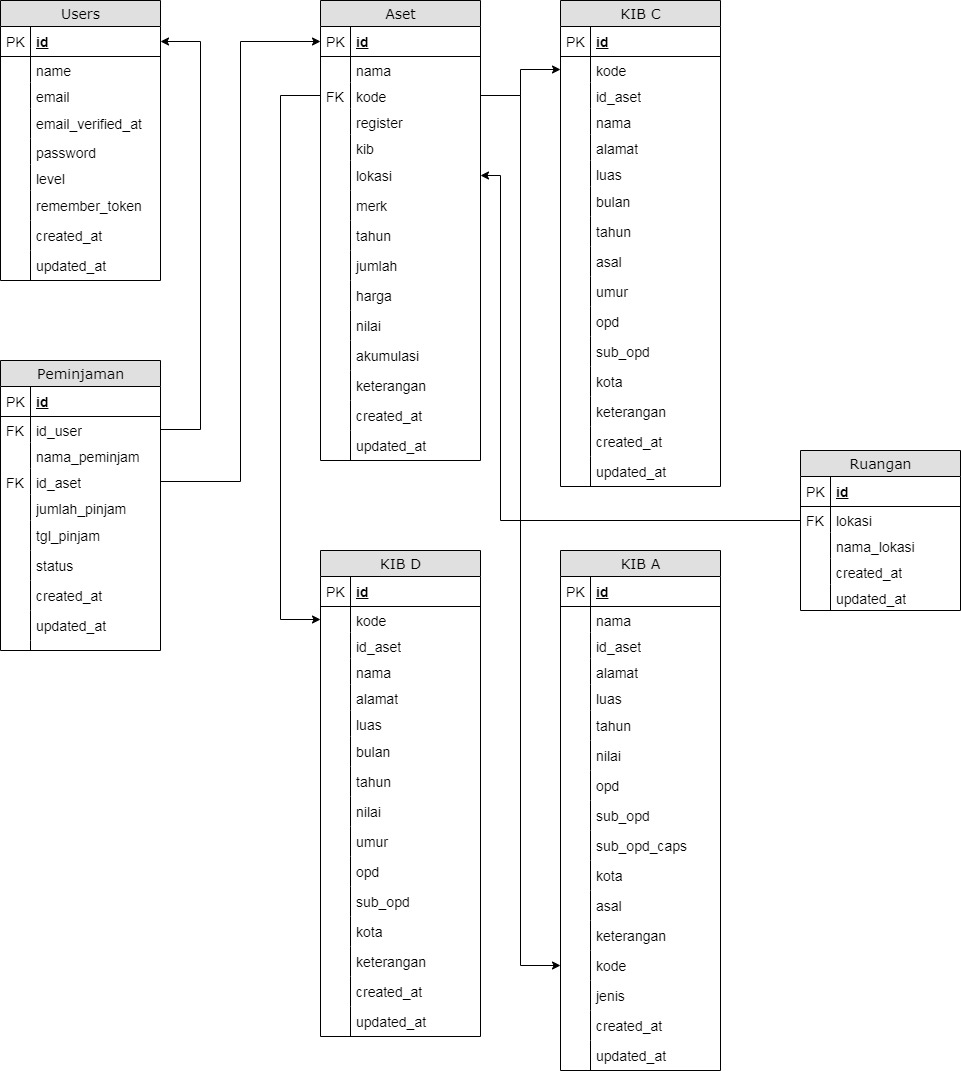
1. **Tabel Ruangan**

Tabel 4. Struktur Tabel Ruangan

|  |  |  |
| --- | --- | --- |
| **Atribut / Kolom** | **Tipe Data** | **Constraint** |
| id | int (20) | Primary Key |
| lokasi | varchar (190) | Foreign Key |
| nama\_lokasi | varchar (255) | - |
| created\_at | timestamp | - |
| updated\_at | timestamp | - |

### Relasi Tabel

Berikut adalah relasi tabel aplikasi ini setelah mengetahui selurush struktur *database* yang sudah diimplementasikan.

****

Gambar 4. Relasi Tabel

## Pengujian

Pada tahap pengujian aplikasi ini, sistem akan dicek keseluruhan fungsinya supaya sistem dapat berjalan dengan baik. Tujuan dari pengujian adalah untuk menjamin aplikasi yang dibuat dapat berjalan sesuai dengan apa yang diinginkan.

1. **Login**

Tabel 4. Pengujian Sistem Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses *login* tidak berjalan | Proses *login* tidak dilaksanakan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *username* atau *password* yang dimasukkan salah | Tetap berada di halaman *login* | Tetap pada halaman tersebut, ditampilkan pemberitahuan bahwa *username* atau password salah | Sesuai |
| Apabila *form* diisi dengan benar | Proses *login* dilakukan dan *user* diarahkan menuju ke halaman *dashboard* | Proses *login* berhasil dan muncul halaman *dashboard* | Sesuai |

1. **Tambah User**

Tabel 4. Pengujian Sistem Tambah User

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses tambah *user* tidak berjalan | Proses tambah *user* tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses tambah *user* berjalan | Proses tambah *user* berhasil dijalankan | Sesuai |

1. **Edit User**

Tabel 4. Pengujian Edit User

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses edit *user* tidak berjalan | Proses edit *user* tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses edit *user* berjalan | Proses edit *user* berhasil dijalankan dan data *user* terpilih berubah | Sesuai |

1. **Hapus User**

Tabel 4. Pengujian Hapus User

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *button* hapusdiklik | Proses hapus *user* dijalankan | Proses hapus *user* berhasil dijalankan | Sesuai |

1. **Tambah Data Aset**

Tabel 4. Pengujian Tambah Data Aset

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses tambah data asettidak berjalan | Proses tambahdata aset tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses tambah data aset berjalan | Proses tambah data asetberhasil dijalankan | Sesuai |

1. **Edit Data Aset**

Tabel 4. Pengujian Edit Data Aset

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses edit data asettidak berjalan | Proses editdata aset tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses edit data aset berjalan | Proses edit data aset berhasil dijalankan dan data aset terpilih berubah | Sesuai |

1. **Hapus Data Aset**

Tabel 4. Pengujian Hapus Data Aset

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *button* hapusdiklik | Proses hapus data aset dijalankan | Proses hapusdata aset berhasil dijalankan | Sesuai |

1. **Tambah Ruangan**

Tabel 4. Pengujian Tambah Ruangan

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses tambah ruangantidak berjalan | Proses tambahruangan tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses tambah ruangan berjalan | Proses tambah ruanganberhasil dijalankan | Sesuai |

1. **Edit Ruangan**

Tabel 4. Pengujian Edit Ruangan

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses edit ruangantidak berjalan | Proses editruangan tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses edit ruangan berjalan | Proses edit ruangan berhasil dijalankan dan ruangan terpilih berubah | Sesuai |

1. **Hapus Ruangan**

Tabel 4. Pengujian Hapus Ruangan

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *button* hapusdiklik | Proses hapus ruangan dijalankan | Proses hapusruangan berhasil dijalankan | Sesuai |

1. **Tambah Transaksi**

Tabel 4. Pengujian Tambah Transaksi

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses tambah transaksitidak berjalan | Proses tambahtransaksi tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses tambah transaksi berjalan | Proses tambah transaksiberhasil dijalankan | Sesuai |

1. **Edit Transaksi**

Tabel 4. Pengujian Edit Transaksi

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *form* tidak diisi | Proses edit transaksitidak berjalan | Proses edittransaksi tidak dijalankan, tetap pada halaman tersebut, dengan pemberitahuan bahwa *form* harus diisi | Sesuai |
| Apabila *form* diisi dengan benar | Proses edit transaksi berjalan | Proses edit transaksi berhasil dijalankan dan transaksi terpilih berubah | Sesuai |

1. **Hapus Transaksi**

Tabel 4. Pengujian Hapus Transaksi

|  |  |  |  |
| --- | --- | --- | --- |
| **Poin Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesesuaian Hasil** |
| Apabila *button* hapusdiklik | Proses hapus transaksi dijalankan | Proses hapustransaksi berhasil dijalankan | Sesuai |

# BAB V PENUTUP

## Kesimpulan

Dari uraian pembahasan keseluruhan materi pada bab-bab sebelumnya dalam rangka mengakhiri pembahasan mengenai aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi ini, dapat diambil beberapa kesimpulan, yaitu :

Berikut merupakan kesimpulan dari proses pengujian dan pengembangan aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahi :

1. Dengan menggunakan aplikasi ini pihak sekolah menjadi lebih mudah untuk melakukan pendataan data sarana dan prasarana sekolah.
2. Kualitas pelayanan sarana dan prasarana menjadi lebih baik terutama untuk warga sekolah yang terlibat.
3. Dengan menggunakan aplikasi ini dapat meminimalisir masalah yang muncul dalam pendataan dan pengelolaan data sarana dan prasarana yang masih dilakukan secara konvensional.
4. Pengajuan peminjaman dan pengajuan sarana yang dilakukan oleh siswa menjadi lebih mudah karena dengan adanya aplikasi ini penanganan permintaan tersebut dapat diproses lebih cepat.

## Saran

Aplikasi Inventaris Sarana Prasarana SMK Negeri 1 Cimahiini tentunya masih memiliki berbagai kekurangan yang dapat diperbaiki dengan pengembangan yang lebih lanjut agar dapat lebih bermanfaat di masa yang akan datang. Berikut merupakan hal-hal yang dapat dikembangkan lebih lanjut diantaranya :

# DAFTAR PUSTAKA