|  |  |
| --- | --- |
|  | UNIVERSITAS DIPONEGORO – FAKULTAS TEKNIK  DEPARTEMEN TEKNIK ELEKTRO  *Jl. Prof. H. Soedarto, SH, Tembalang, Semarang 50275*  *Telp/Faks. (024)-7460057 e-mail: departemen@elektro.undip.ac.id* |

**Dokumen Pengembangan Produk**

Lembar Sampul Dokumen

|  |  |  |
| --- | --- | --- |
| Judul Dokumen | **TUGAS AKHIR:**  **Rancang Bangun Sistem Keamanan Kunci Pintu GedungBerbasis *Internet of Things*** | |
|  |  | |
| Jenis Dokumen | **SPESIFIKASI** | |
|  | Catatan: Dokumen ini dikendalikan penyebarannya oleh Dept. Teknik Elektro Undip | |
| Nomor Dokumen | **B300-01-TA2223.2.19012** | |
|  |  | |
| Nomor Revisi | **01** | |
|  |  | |
| Nama File | **B300-2-TA2223** | |
|  |  | |
| Tanggal Penerbitan | **10 April 2023** | |
|  |  | |
| Unit Penerbit | **Departemen Teknik Elektro Undip** | |
|  |  | |
| Jumlah Halaman | **24** | (termasuk lembar sampul ini) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Pengusul | | | | |
| Pengusul | Nama  NIM | **Henric Dhiki Wicaksono**  21060119120011 | Jabatan  Tanda Tangan | Anggota |
|  | Nama  NIM | **Novi Dianasari**  21060119120039 | Jabatan  Tanda Tangan | Anggota |
|  | Nama  NIM | **Muhammad Khoiril Wafi**  21060119140133 | Jabatan  Tanda Tangan | Anggota |
| Pembimbing | Nama | **M. Arfan, S.Kom., M.Eng.** | Tanda Tangan |  |
| Utama | NIP | 198408172015041002 |  |  |
| Pendamping | Nama | **Imam Santoso, S.T., M.T.** | Tanda Tangan |  |
|  | NIP | 197012031997021001 |  |  |

**DAFTAR ISI**

[Catatan Sejarah Perbaikan Dokumen 3](#_Toc143345281)

[1. PENDAHULUAN 4](#_Toc143345282)

[1.1 Ringkasan Isi Dokumen 4](#_Toc143345283)

[1.2 Aplikasi Dokumen 4](#_Toc143345284)

[1.3 Referensi 4](#_Toc143345285)

[1.4 Daftar Singkatan 5](#_Toc143345286)

[2. SOLUSI DESAIN 6](#_Toc143345287)

[2.1 Solusi Desain 1 6](#_Toc143345288)

[2.2 Solusi Desain 2 6](#_Toc143345289)

[2.3 Solusi Desain 3 6](#_Toc143345290)

[3. PERANCANGAN 7](#_Toc143345291)

[3.1 Arsitektur Sistem 7](#_Toc143345292)

[3.2 Metode Verifikasi Akses 7](#_Toc143345293)

[3.3 Perangkat Penguncian 8](#_Toc143345294)

[3.4 Komunikasi Dua Arah 12](#_Toc143345295)

[3.5 Database 14](#_Toc143345296)

[3.6 Backend API 15](#_Toc143345297)

[3.7 Aplikasi Mobile 17](#_Toc143345298)

[3.8 Tampilan Website 20](#_Toc143345299)

[4. VERIFIKASI 23](#_Toc143345300)

[4.1 Komunikasi WiFi dan Bluetooth 23](#_Toc143345301)

[4.2 Backend API Laravel 23](#_Toc143345302)

[4.3 Kode QR 24](#_Toc143345303)

[5. PENUTUP 24](#_Toc143345304)

# Catatan Sejarah Perbaikan Dokumen

|  |  |
| --- | --- |
| **VERSI, TGL, OLEH** | **PERBAIKAN** |
| 01,  10 April 2023,  oleh Henric Dhiki Wicaksono, Novi Dianasari  dan Muhammad Khoiril Wafi. | *Draft* Dokumen B300 |

1. PENDAHULUAN

## **Ringkasan Isi Dokumen**

Dokumen ini berisi tentang uraian desain untuk merealisasikan produk berupa “Sistem Keamanan Kunci Pintu Gedung Berbasis IoT”. Uraian desain terdisi dari pemilihan solusi yang digunakan dalam menyelesaikan permasalahan serta gambaran cara kerja dari semua fitur yang telah dijabarkan. Dokumen ini juga akan menjelaskan desain dari subsistem yang ada yaitu perangkat kunci pintu, sistem komunikasi, database, server, website dan aplikasi *mobile*. Desain subsistem meliputi pemilihan kerangka kerja serta algoritma yang akan digunakan dalam mengimplementasikan sistem tersebut.

Dokumen ini selanjutnya akan digunakan sebagai acuan dalam proses implementasi “Sistem Keamanan Kunci Pintu Gedung Berbasis IoT” serta sebagai bahan evaluasi pada proses pengembangan selanjutnya.

## **Aplikasi Dokumen**

Dokumen ini digunakan dalam proses pengembangan produk “Rancang Bangun Sistem Keamanan Kunci Pintu Gedung Berbasis *Internet of Things*” untuk:

1. Menjelaskan pemilihan desain sesuai dengan subsistem yang ada.
2. Gambaran mengenai desain dan carakerja sistem.
3. Menjelaskan standar-standar yang dipergunakan.
4. Referensi komponen atau library yang digunakan.
5. Penjadi catatan proses pengerjaan dan revisi.

Dokumen B300 ini diajukan kepada dosen pembimbing tugas akhir dan tim tugas akhir Program Studi Sarjana Teknik Elektro Undip sebagai bahan penilaian tugas akhir.

## **Referensi**

[1] I. Hermawan, D. Arnaldy, P. Oktivasari, and D. A. Fachrudin, “Development of Intelligent Door Lock System for Room Management Using Multi Factor Authentication,” vol. 16, no. 1, pp. 1–14, 2023.

[2] K. Y. Sun, Y. Pernando, and M. I. Safari, “Perancangan Sistem IoT pada Smart Door Lock Menggunakan Aplikasi BLYNK,” *JUTSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 1, no. 3, pp. 289–296, 2021, doi: 10.33330/jutsi.v1i3.1360.

[3] C. Asiminidis, G. Kokkonis, and S. Kontogiannis, “Database Systems Performance Evaluation for IoT Applications,” *SSRN Electron. J.*, no. November 2019, 2019, doi: 10.2139/ssrn.3360886.

[4] N. A. Amir Hamzah, M. R. Abu Saad, W. Z. Wan Ismai, T. Bhunaeswari, and N. Z. Abd Rahman, “Development of A Prototype of An IoT Based Smart Home with Security System Flutter Mobile,” *J. Eng. Technol. Appl. Phys.*, vol. 1, no. 2, pp. 34–41, 2019, doi: 10.33093/jetap.2019.1.2.70.

## **Daftar Singkatan**

**Tabel 1.1** Daftar Singkatan

| Singkatan | Arti |
| --- | --- |
| IoT | *Internet of Things* |
| WiFi | *Wireless Fidelity* |
| JSON | *Javascript Object Notation* |
| ACL | *Access Control List* |
| HTTPS | *Hypertext Transfer Protocol Secure* |
| IOS | *iPhone Operating System* |
| PC | *Personal Computer* |
| IDE | *Integrated Development Environment* |
| PCB | *Printed Circuit Board* |
| UI | *User Interface* |
| QR-*Code* | *Quick Response Code* |
| SSL/TLS | *Secure Sockets Layer/Transport Layer Security* |

1. SOLUSI DESAIN

Sesuai dengan uraian pada dokumen B200 sebelumnya yaitu dibutuhkan sebuah sistem yang dapat digunakan untuk mengelola kunci pintu gedung secara aman dan efisien dengan berbagai fitur yang dimiliki seperti pemantauan, pengaturan hak akses, kendali jarak jauh dan lain sebagainya maka diperlukan desain sistem yang dapat menjalankan semua fitur tersebut. Pemilihan desain didasarkan pada kebutuhan serta keamanan dari sistem untuk menjalankan semua fitur yang ada dengan menggunakan ilmu rekayasa terkait sehingga menghasilkan desain sistem yang aman dan efisien.

1. Solusi Desain 1

Solusi desain yang pertama yaitu menggunakan konsep IoT untuk mengelola semua kunci pintu yang sebelumnya berupa kunci fisik menjadi kunci digital. Pada setiap pintu dalam satu gedung akan terpasang sebuah perangkat kunci digital yang dapat dipantau serta dapat dikendalikan oleh semua pengguna gedung tersebut. Dengan menggunakan perangkat penguncian secara digital memungkinkan untuk melakukan pengelolaan kunci dengan lebih aman dan efektif.

1. Solusi Desain 2

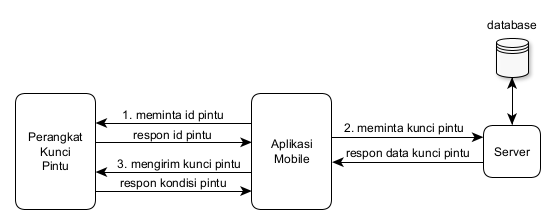
Solusi desain yang kedua yaitu membangun sebuah layanan untuk mengendalikan sistem yang sudah dibangun oleh perangkat kunci digital. Sebuah server dapat menyediakan layanan untuk melakukan pemantauan serta memberikan perintah ke perangkat kunci pintu secara otomatis. Setiap perangkat kunci pintu akan terhubung ke server dengan menggunakan koneksi internet. Server juga dapat menentukan siapa saja yang diijinkan untuk mengakses pintu berdasarkan data yang tersimpan didalamnya.

1. Solusi Desain 3

Solusi desain yang ketiga yaitu membuat sebuah tampilan sebagai media yang digunakan oleh pengguna gedung untuk berinteraksi dengan perangkat kunci pintu seperti membuka kunci atau melihat kondisi pintu secara jarak jauh. Sebuah aplikasi mobile dan website dapat digunakan sebagai antarmuka pengguna untuk berinteraksi dengan sistem, aplikasi mobile dapat digunakan untuk membuka atau memantau kondisi pintu dengan karakteritiknya yang ringkas dan mudah sedangkan website digunakan untuk melakukan pengaturan sistem.

1. PERANCANGAN
2. Arsitektur Sistem
3. Metode Verifikasi Akses

Setiap pintu pada gedung akan dikunci menggunakan perangkat kunci pintu digital, hal tersebut memungkinkan untuk melakukan verifikasi akses kepada semua pengguna yang ingin mengakses ruangan tersebut. Proses verifikasi akses dapat dilaksanakan dengan mengunakan berbagai macam metode, penelitian yang dilakukan oleh Indra Hermawan dkk [1] tentang autentikasi pada kunci pintu pintar, RFID dapat digunakan untuk melakukan autentikasi kunci pintu, namun metode ini juga tidak menyediakan solusi yang sesuai untuk menyelesaikan permasalahan dikarenakan RFID hanya akan menggantikan tempat dari kunci fisik. Oleh karena itu pada penelitian ini dikembangkan metode lain untuk melakukan autentikasi akses. Pada penelitian ini menggunakan *access* *control* *list* untuk menentukan apakah pengguna diijinkan untuk masuk ke suatu ruangan. Diagram dari metode verifikasi akses dapat dilihat pada Gambar 3.1 dibawah.



Gambar 3.1 Metode Autentikasi

Pada Gambar 3.1 terlihat proses verifikasi akses melibatkan 3 bagian yaitu perangkat kunci pintu, aplikasi *mobile* dan *sever* menggunakan 3 langkah dengan penjelasan sebagai berikut :

1. Meminta identitas pintu

Untuk membuka kunci pintu tentunya perlu mengetahui pintu mana yang akan dibuka, identitas pintu akan membedakan pintu satu dengan pintu lainnya sehingga setiap pintu akan memiliki identitas yang unik. Identitas pintu dapat berupa kode QR sehingga memudahkan aplikasi mobile untuk mengenali pintu tersebut dengan memindai kode QR tersebut.

1. Meminta kunci pintu

Setelah mendapatkan identitas pintu selanjutnya aplikasi *mobile* akan meminta kode kunci pintu dengan mengirimkan permintaan ke *server* disertai dengan identitas pengguna. *Server* akan memeriksa data akses pengguna dengan menggunakan data identitas pintu dan identitas pengguna, jika aksesnya cocok maka *server* akan mengembalikan kode kunci pintu yang digunakan untuk membuka pintu.

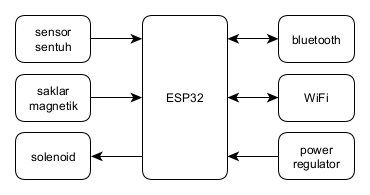
1. Mengirim kunci pintu

Setelah mendapatkan kode kunci pintu selanjutnya aplikasi mobile akan mengirimkan kode kunci tersebut melalui koneksi bluetooth untuk membuka kunci pintu.

Berdasarkan penjelasan metode autentikasi diatas, proses membuka kunci pintu menggukan koneksi bluetooth hal dikarenakan untuk mendapatkan identitas pintu menggunakan kode QR statis sehingga meungkinkan pengguna untuk menduplikasi kode QR tersebut, dengan adanya kekurangan tersebut maka penggunaan bluetooth akan memastikan bahwa pengguna benar-benar memindai kode QR diarea atau didepan pintu persis karena karakteristik bluetooth yang memiliki batas jarak koneksi yang dekat. Untuk membuka kunci pintu maka pengguna menggunakan kode kunci dinamis, kode tersebut akan berubah seiring dengan perubahan kondisi pintu sehingga proses autentikasi benar-benar aman dengan menggunakan kunci yang dinamis.

1. Perangkat Penguncian

Untuk menjalankan proses penguncian secara digital pada setiap pintu tentunya membutuhkan sebuah alat yang dapat digunakan untuk mengunci dan memantau kondisi pintu. Pada penilitian yang dilakukan oleh Kaleb Yefune Sun dkk [2] tentang sistem kunci pintu menggunakan BLYNK, ESP32 dapat digunakan untuk mengendalikan kunci pintu serta melakukan pemantauan kondisi pintu. diagram perangkat penguncian dapat dilihat pada Gambar 3.2 dibawah.

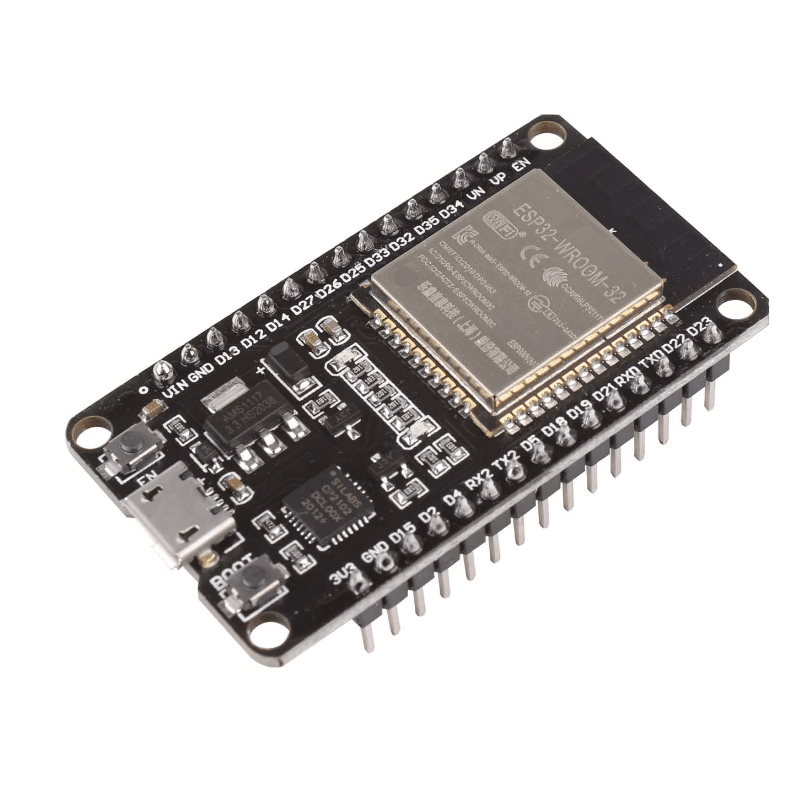


Gambar 3.2 Diagram Perangkat Penguncian

Pada Gambar 3.2 terlihat perangkat penguncian menggunakan ESP32 sebagai komponen utama dengan beberapa komponen tambahan sebagai berikut :

1. ESP32

ESP32 menjadi komponen utama didalam perangkat penguncian, ESP32 merupakan sebuah mikrokontroler sehingga dapat digunakan untuk mengatur dan memantau kondisi pintu dengan mengolah data yang diterima dari sensor dan aktuator.

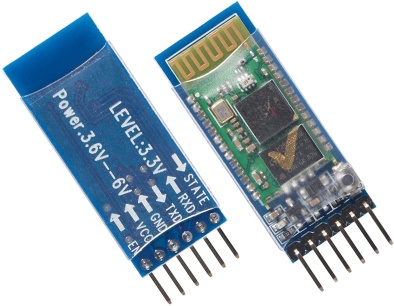


Gambar 3.3 ESP32 *Development Board*

Gambar 3.3 merupakan sebuah papan pengembangan ESP32, dimana pada papan pengembangan tersebut terdapat pin-pin digital yang dapat digunakan untuk membaca sensor atau menggerakkan aktuaror, pada papan pengembangan juga terdapat modul programmer yang digunakan untuk menjalankan program pada proses pembuatan kode program.

1. Bluetooth

Proses autentikasi akses pengguna akan menggunakan koneksi bluetooth untuk mengirim kode kunci. Oleh karena itu perangkat kunci pintu harus mempunyai mobul komunikasi bluetooth sebagai jalur komunikasi untuk berkomunikasi dengan aplikasi *mobile*. Modul bluetooth yang digunakan harus dapat berkuminukasi dengan ESP32 sebagai kontroler utama maka pada perangkat penguncian menggunakan modul HC-05 sebagai modul bluetooth dengan komunikasi serial yang didukung oleh ESP32.



Gambar 3.4 Modul Bluetooth HC-05

Gambar 3.4 merupakan modul bluetootnh HC-05, modul bluetooth tersebut dapat berkomuniasi dengan perangkat *smartphone* melalui koneksi tanpa kabel. Pada modul bluetooth tersebut juga terdapat pin-pin yang digunakan untuk berkomunikasi dengan mikrokontroler melalui koneksi serial.

1. WiFi

Perankat penguncian perlu terhubung dengan server untuk mengirimkan kondisi pintu atau menerima perintah dari jarak jauh, oleh karena itu ESP32 membutuhkan modul komunikasi yang dapat terhubung dengan internet. Pada ESP32 sudah tersedia modul WiFI yang dapat digunakan untuk terhubung ke internet sehingga tidak perlu menambah komponen WiFi eksternal.

1. Saklar magnetik

Perangkat penguncian harus dapat memantau kondisi pintu untuk selanjutnya dilaporkan ke *server*, sebuah sensor diperlukan untuk melakukan tugas penatauan ini. Sebuah sensor magnetik dapat digunakan untuk mendeteksi kondisi pintu terbuka atau tidak dengan memanfaatkan medan magnet.



Gambar 3.5 Saklar Magnetik

Gambar 3.5 merupakan sebuah saklar magnetik, sebuah saklar magnetik terdiri dari dua bagian yaitu bagian magnet dan saklar, pada saat keduanya berdekatan maka saklar akan tertutup dan jika keduanya terpisah maka saklar aka terbuka. Tengan memanfaatkan fenomena ini maka menungkinkan untuk mendeketsi kondisi pintu sedang terbuka atau tertutup.

1. Solenoid

Sebuah aktuator diperlukan untuk mengunci pintu secara fisik, solenoid dapat digunakan untuk mengunci pintu fisik dengan kendali berupa arus listrik. Dengan menggunakan solenoid memungkinkan kendali penguncian dilakukan secara digital dengan mengirimkan arus listrik.



Gambar 3.6 Solenoid

Gambar 3.6 merupakan sebuah solenoid, solenoid menggunakan prinsip elektromagnetik untuk menarik pin kunci sehingga dapat digunakan untuk mengunci pintu dengan memberikan arus listrik yang sesuai.

1. Sensor sentuh

Dalam kondisi normal sebuah solenoid akan selalu mengunci pintu dan solenoid hanya dapat membuka kunci pintu dalam waktu yang relatif singkat atau tidak dapat membuka kunci pintu terus menerus, oleh karena itu diperlukan mekanisme untuk mendeteksi pengguna yang membuka pintu untuk mengaktifkan solenoid. ESP32 menyediakan sebuah sensor sentuh yang dapat digunakan untuk mendeteksi sentuhan pada gagang pintu sehingga informasi tersebut dapat digunakan untuk mengaktifkan solenoid.

1. Power regulator

Untuk dapat beroperasi tentunya perangkat penguncian memerlukan arus listrik, setiap komponen yang digunakan pada perangkat penguncian mungkin memerlukan voltase yang berbeda sehingga diperlukan untuk menyediakan voltase yang sesuai dengan kebutuhan setiap komponen.

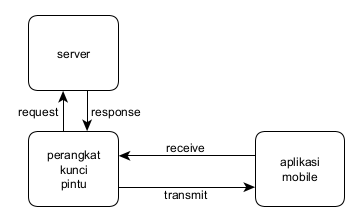


Gambar 3.7 Regulator 5 Volt

Gambar 3.7 merupakan sebuah regulator untuk menurunkan tegangan masukan menjadi 5 volt. Perangkat penguncian menggunakan tegangan masukan sebesar 12 volt hal ini sesuai dengan voltase kerja dari solenoid sedangkan ESP32 bekerja pada voltase 5 volt sehingga diperlukan sebuah regulator untuk menurunkan tegangan 12 volt menjadi 5 volt.

1. Komunikasi Dua Arah

Perangkat penguncian perlu berkomunikasi dengan perangkat lain seperti *smartphone* dan *server* baik itu untuk menerima perintah atau mengirimkan respon oleh karena itu perangkat penguncian membutuhkan komunikasi dua arah untuk mengirim dan menerima data yang akan digunakan pada proses selanjutnya.



Gambar 3.8 Komunikasi Dua Arah

Gambar 3.8 manampilkan diagram komunikasi yang dilakukan oleh perangkat kunci pintu dengan server dan aplikasi mobile. Komunikasi yang dilakukan oleh perangkat kunci pintu terbagi menjadi dua yaitu komunikasi dengan server dan komunikasi dengn apliaksi mobile dengan penjelasan sebagai berikut :

1. Komunikasi dengan server

Perangkat kunci pintu perlu berkomunikasi dengan server untuk menerima perintah secara jarak jauh dan untuk melaporkan kondisi pintu secara realtime. Untuk menjalin komunikasi tersebut perangkat kunci pintu menggunakan protokol http dengan menggunakan metode API. Perangkat kunci pintu akan menggunakan API yang telah disediakan oleh server untuk melaporkan kondisi pintu.

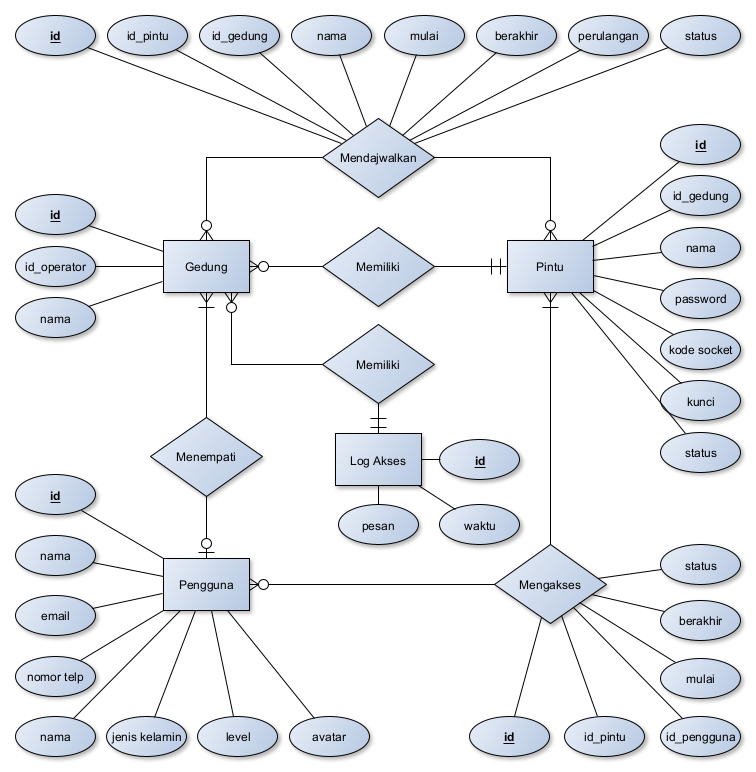
Untuk menerima perintah dari server metode yang digunakan sedikit berbeda dikarenakan perangkat kunci pintu bertindak sebagai client, untuk menerima perintah dari server perangkat kunci pintu menggunakan protokol websocket, websocket menungkinkan untuk menjalin koneksi dengan server dan koneksi tersebut akan terus terjalin, dengan menggunakan koenksi ini prangkat kunci dapat menerima event dari server meskipun perangkat kunci pintu tidak melaukan permintaan data. dengan menggunakan metode ini memungkinkan perintah dikirimkan secara langsung tanpa menunggu polling yang dilakukan oleh perangkat kunci pintu.

1. Komunikasi dengan aplikasi mobile

Untuk berkomunikasi dengan aplikasi mobile, perangkat kunci pintu menggunakan koneksi bluetooth yang umumnya disediakan oleh semua perangkat smartphone. Dengan menggunakan bluetooth maka proses autentikasi hanya bisa dilakukan didekat perangkat kunci pintu dan juga memberikan keuntungan bahwa perangkat kunci pintu tetap dapat bekerja walaupun tanpa koneksi intenet.

1. Database

*Database* digunakan untuk menyimpan data yang digunakan dalam sistem penguncian pintu gedung seperti data pengguna, data pintu, data akses dan lain sebagainya. *Database* yang digunakan harus dapat menunjang kinerja sistem dengan karakteristik respon yang cepat dan pengelolaan data terstruktur. Berdasarkan penelitian [3] yang membandingkan kinerja dari berbagai tipe dan jenis *database* didapatkan hasil penggunaan MySQL menunjukkan hasil kinerja yang bagus dalam hal waktu eksekusi permintaan, dengan sistem penyimpanan data bersifat relasional dan terstruktur maka MySQL dapat diterapkan pada sistem penguncian pintu gedung ini.

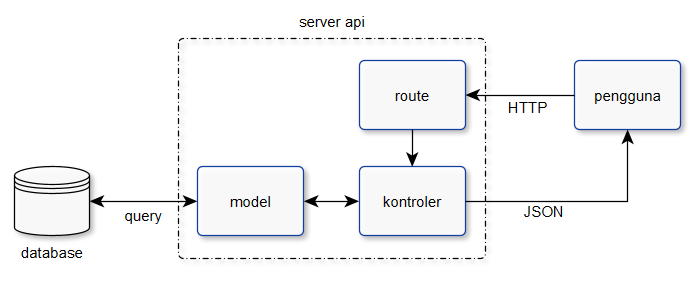


Gambar 3.9 ERD *Database*

Pada Gambar 3.9 terlihat relasi antar entitas didalam *database* sistem penguncian pintu gedung. Pada database sistem penguncian pintu gedung terdapat beberapa entitas seperti pengguna, pintu, gedung dan *log* akses, entitas tersebut digunakan untuk meyimpan data yang akan digunakan didalam pengelolaan sistem dengan isi data sesuai dengan atribut dari masing-masing entitas. Dialam sistem database juga terdapat relasi, relasi menunjukkan hubungan antar entitas seperti menempati, memiliki, menjadwalkan dan mengakses, khusus untuk relasi menjadwalkan dan mengakses pada Gambar 3.9 terlihat bahwa relasi tersebut memiliki aktribut yang menandakan bahwa relasi tersebut juga bertindak sebagai entitas yang menjadi penghubung antara 2 entitas yang berbeda, pada umumnya kondisi seperti ini berarti relasi tersebut menunjukkan kardinalitas *many-to-many* dimana relasi tersebut bertindak sebagai entitas pivot untuk menghubungkan relasi *many-to-many* dari 2 entitas lainnya.

1. Backend API

Sebuah API dibutuhkan sebagai layanan komunikasi yang digunakan oleh perangkat kunci pintu dan aplikasi *mobile* untuk berkomunikasi dengan server. Selain terdapat *website* yang digunakan sebagai antarmuka utama dalam mengelola sistem ini juga terdapat aplikasi berbasis *mobile* yang digunakan untuk menunjang kinerja dari sistem terutama pada bagian yang membutuhkan teknologi yang belum disediakan oleh *browser* seperti koneksi *bluetooth* dan pindai kode QR menggunakan kamera.



Gambar 3.10 Diagram *Backend* API

Pada Gambar 3.10 terlihat diagram dari backend API. Backend API dibangun menggunakan konsep MVC yang disediakan oleh laravel. Laravel merupakan sebuah kerangka kerja pengembangan website dan API dengan bahasa pemrograman PHP dengan menggunakan konsep MVC. Dengan menggunakan laravel proses pengembangan API dapat dilakukan dengan cepat dengan menggunakan berbagai modul-modul yang telah tersedia seperti autentikasi, koneksi *database*, *query*, token dan lain sebagainya. API akan menyediakan layanan berupa *enpoint* yang dapat diakses oleh perangkat kunci pintu maupun aplikasi mobile, *endpoint* yang tersedia dapat dilihat pada tabel Tabel 3.1 dibawah.

Tabel 3.1 *Endpoint* API

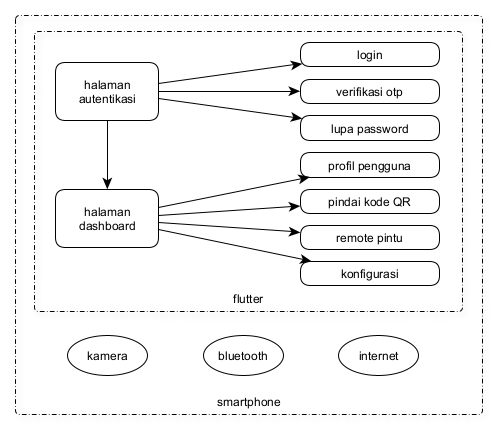
|  |  |  |  |
| --- | --- | --- | --- |
| **Tipe** | ***Endpoint*** | ***Auth*** | **Keterangan** |
| POST | /api/login | - | login pengguna dan operator |
| POST | /api/reset-password | - | reset password menggunakan email |
| POST | /api/verify-email | sanctum | verifikasi email |
| GET | /api/logout | sanctum | logout pengguna dan operator |
| POST | /api/update-profile | sanctum, verified | update profile nama, email, dan lainnya |
| GET | /api/avatar | sanctum, verified | mengambil gambar avatar |
| POST | update-avatar | sanctum, verified | mengubah gambar avatar |
| POST | /api/change-password | sanctum, verified | mengubah password pengguna atau operator |
| GET | /api/my-access | sanctum, verified | mengambil daftar akses |
| GET | /api/get-doors | sanctum, verified | mengambil daftar pintu |
| GET | /api/my-history | Sanctum, verified | mengambil daftar riwayat akses |
| GET | /api/verify-access/{door\_id} | sanctum, verified | verifikasi akses dari kode QR pintu |
| POST | /api/remote-access | sanctum, verified | membuka atau mengunci pintu jarak jauh |
| POST | /door/login | - | login perangkat kunci pintu |
| POST | /door/register | - | menambahkan perangkat kunci pintu baru |
| GET | /door/logout | sanctum | logout perangkat kunci pintu |
| POST | /door/get-signature | sanctum | mengambil kode signature channel pusher |
| POST | /door/update-status | sanctum | update status pintu |
| POST | /door/alert | sanctum | peringatan pada pintu |

Dapat dilihat pada Tabel 3.1 diatas, API terbagi menjadi 2 bagian yang ditandai dengan awalan (*prefix*) “/api” dan “/door”. Awalan “/api” merupakan *endpoint* API yang khusus ditujukan untuk penggunaan aplikasi mobile pengguna dan operator sedangkan awalan “/door” merupakan API yang ditujukan untuk perangkat kunci IoT sehingga dengan menggunakan API yang terpisah maka API akan semakin terorganisir dengan baik.

Pada Tabel 3.1 juga terlihat bahwa beberapa *endpoint* memerlukan autentikasi sanctum dan *verified*. Autentikasi sanctum merupakan sebuah metode autentikasi berbasis token yang digunakan untuk mengamankan sumber daya API dari *client* dengan hanya mengijinkan pengguna yang sudah terautentikasi yang dapat mengakses API tersebut. Sedangkan *verified* merupakan autentikasi tambahan yang digunakan untuk memastikan bahwa client (pengguna dan operator) sudah melakukan verifikasi email sehingga dapat meningkatkan keamanan API.

1. Aplikasi Mobile

Sebuah aplikasi *mobile* diperlukan untuk membuka kunci pintu dengan menggunakan kode QR. Aplikasi *mobile* akan menyediakan fitur koneksi bluetooth yang belum tersedia pada sebuah *website* sehingga proses verifikasi akses dapat berjalan dengan baik. Untuk menjalankan proses verifikasi akses dan fitur lainnya maka aplikasi mobile yang dikembangkan harus memiliki beberapa kemampuan yang mendukung kinerja dari sistem penguncian seperti koneksi internet, koneksi bluetooth, serta pemindai kode QR. Pada penelitian yang dilakukan oleh Nur Asyiqin Bt. Amir Hamzah dkk [4] tentang perancangan rumah pintar berbasi IoT, kerangka kerja Flutter dapat digunakan dalam pengembangan aplikasi mobile. Flutter merupakan sebuah kerangka kerja untuk membuat aplikasi mobil, flutter merupakan kerangka kerja cross-platform yang artinya aplikasi yang dihasailkan dapat dijalankan pada sistem operasi yang berbeda misalnya Android dan IOS dengan menggunakan satu stuktuk program yang sama. Secara umum aplikasi yang akan dikembangkan digunakan untuk berinteraksi dengan perangkat IoT misalnya untuk membuka kunci, melihat konsidi pintu dan melakukan pengaturan.



Gambar 3.11 Struktur Aplikasi Mobile

Gambar 3.11 memperlihatkan struktur dari aplikasi mobile yang akan dikembangkan, pada aplikasi mobile akan terdapat beberapa layar yang dapat digunakan untuk berintraksi dengan sistem kunci pintu, penjelasan mengenai setiap layat dapat dilihat pada Tabel 3.2 dibawah.

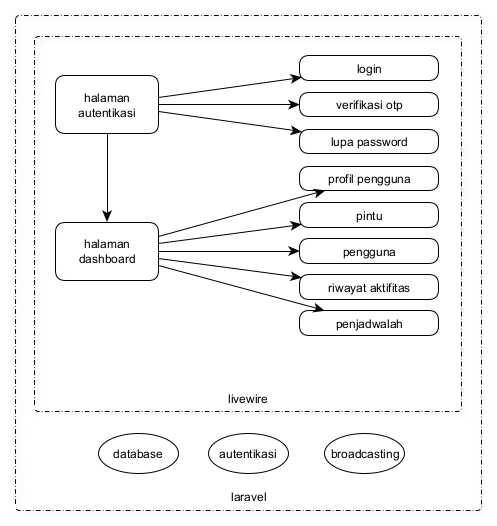
Tabel 3.2 Pembagian Layar Aplikasi Mobile

|  |  |  |
| --- | --- | --- |
| **Nama** | **Autentikasi** | **Penjelasan** |
| Login | - | Halaman autentikasi untuk login pengguna dan operator. Proses login menggunakan alamat email dan password yang sudah terdaftar. |
| Verifikasi OTP | - | Jika email belum terverifikasi maka akan diarahkan ke halaman verifikasi, verifikasi dengan menggunakan kode OTP yang dikirimkan ke email yang dimasukkan pada saat login. |
| Lupa password | Login | Pengguna aplikasi dapat memanfaatkan fitur ini jika pengguna kehiangan password dengan cara memasukkan alamat email yang sesuai untuk menerima petunjuka lebih lanjut. |
| Profil pengguna | Login | Proses pengaturan pengguna dilakukan pada halaman profil pengguna, pengguna aplikasi dapat menyesuaikan informasi diri seperti nama, nomor hp dan foto profil. |
| Pindai kode QR | Login | Pengguna dapat menggunakan kode QR pada pintu untuk membuka kunci pintu, pada halaman ini pengguna memindai kode tersebut dan aplikasi akan meminta akses ke server secara otomatis. |
| Remote pintu | Login, Operator | Operator dapat membuka atau mengunci pintu secara jarak jauh melalui koneksi internet, pada halaman ini juga menampilkan semua pintu dan kondisinya. |
| konfigurasi | Login, Operator | Halaman ini digunakan untuk melakukan kofigurasi pada perangkat kunci pintu seperti mengatur kredensial WiFi atau mendaftarkan perangkat baru ke server. |

Pada Gambar 3.11 juga terhihat bahwa aplikasi membutuhkan dukungan dari kamera, internet dan bluetooth. Pada kerangka kerja flutter penggunaan peripheral pada smartphone dapat dilakukan dengan menggunakan beberapa modul yang telah disediakan seperti contohnya dapat menggunakan modul http untuk menjalankan fungsi yang berkaitan dengan internet, modul bluetooth\_serial untuk menjalankan komunikasi bluetooth sederhana dan qr\_scanner untuk menajalankan fundsi penindaian menggunakan kamera.

1. Tampilan Website

Sistem keamanan kunci pintu gedung ini juga memerlukan sebuah website yang digunakan untuk melakukan pengaturan pada sistem tersebut. Dengan menggunakan website maka proses pengaturan menjadi lebih fleksibel, proses pengaturan yang dilakukan meliputi pengelolaan pintu, pengguna, akses, penjadwalan dan riwayat aktifitas. Website tersebut hanya disa diakses oleh pengelola gedung sebagai operator yang menjadi penanggung jawab terhadap sistem keamanan yang dijalankan.



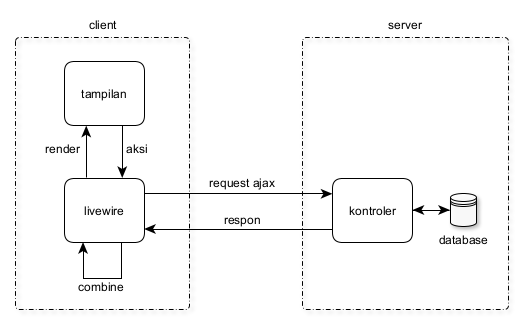
Gambar 3.12 Struktur Website

Pada Gambar 3.12 memperlihatkan struktur dari website yang akan dikembangkan, sama seperti pada backend API website dikembangkan dengan menggunakan kerangka kerja Laravel dengan memanfaatkan modul-modul yang telah disediakan seperti koneksi database, autentikasi dan broadcasting. Seperti pada apliksi mobile, website memiliki beberapa layar yang digunakan oleh operator untuk mengatur sistem seperti menambahkan pintu baru, membuat jadwal, menambahkan akses, menambahkan pengguna dan lainn sebagainya. Penjelasan fungsi setiap layar pada website dapat dilihat pada x dibawah.

Tabel 3.3 Pembagian Layar Website

|  |  |  |
| --- | --- | --- |
| **Nama** | **Autentikasi** | **Penjelasan** |
| Login | - | Halaman autentikasi untuk login pengguna dan operator. Proses login menggunakan alamat email dan password yang sudah terdaftar. |
| Verifikasi OTP | - | Jika email belum terverifikasi maka akan diarahkan ke halaman verifikasi, verifikasi dengan menggunakan kode OTP yang dikirimkan ke email yang dimasukkan pada saat login. |
| Lupa password | Login | Pengguna aplikasi dapat memanfaatkan fitur ini jika pengguna kehiangan password dengan cara memasukkan alamat email yang sesuai untuk menerima petunjuka lebih lanjut. |
| Profil pengguna | Login, Operator | Proses pengaturan pengguna dilakukan pada halaman profil pengguna, pengguna dapat menyesuaikan informasi diri seperti nama, nomor hp dan foto profil. |
| Pintu | Login, Operator | Halaman pintu akan menampilkan daftar semua pintu yang ada disertai dengan kondisinya pada halaman ini juga tersedia tombol untuk membuka atau mengunci pintu. Pada halaman ini juga tesedia menu untuk menambahkan akses pengguna pada pintu tertentu. |
| Pengguna | Login, Operator | Halama pengguna digunakan untuk menampilkan semua pengguna yang terdaftar disertai dengan akses ayng dimiliki, halaman ini juga menyediakan menu untuk menambah dan menghapus pengguna. |
| Penjadwalan | Login, Operator | Halaman penjadwalan digunakan untuk mengatur jadwal seperti menambahkan, memperbarui dan menghapus jadwal. |
| Riwayat akses | Login, Operator | Halaman ini digunakan untuk menampilkan semua aktifitas pengguna pada sistem keamanan kunci pintu gedung ini. |

Untuk meningkatkan kenyamanan dan performa dari website maka pembuatan setiap tampilan akan menggunakan livewire. Livewire merupakan sebuah library pada laravel yang digunakan untuk menerapkan konsep Single Page Application (SPA) pada laravel, laravel sendiri merupakan kerangka kerja yang bersifat server-side yaitu semua pengolahan data akan dilakukan didalam server dan kemudian hasilnya akan dikirimkan kembali ke client berupa tampilan dalam format http, hal ini menyebabkan setiap ada interaksi pengguna maka seluruh halaman akan diperbarui sehingga menjadikan tampilan kurang nyaman dan dapat menurunkan performa website karena harus memuat seluruh halaman dari awal.



Gambar 3.13 Cara Kerja Livewire

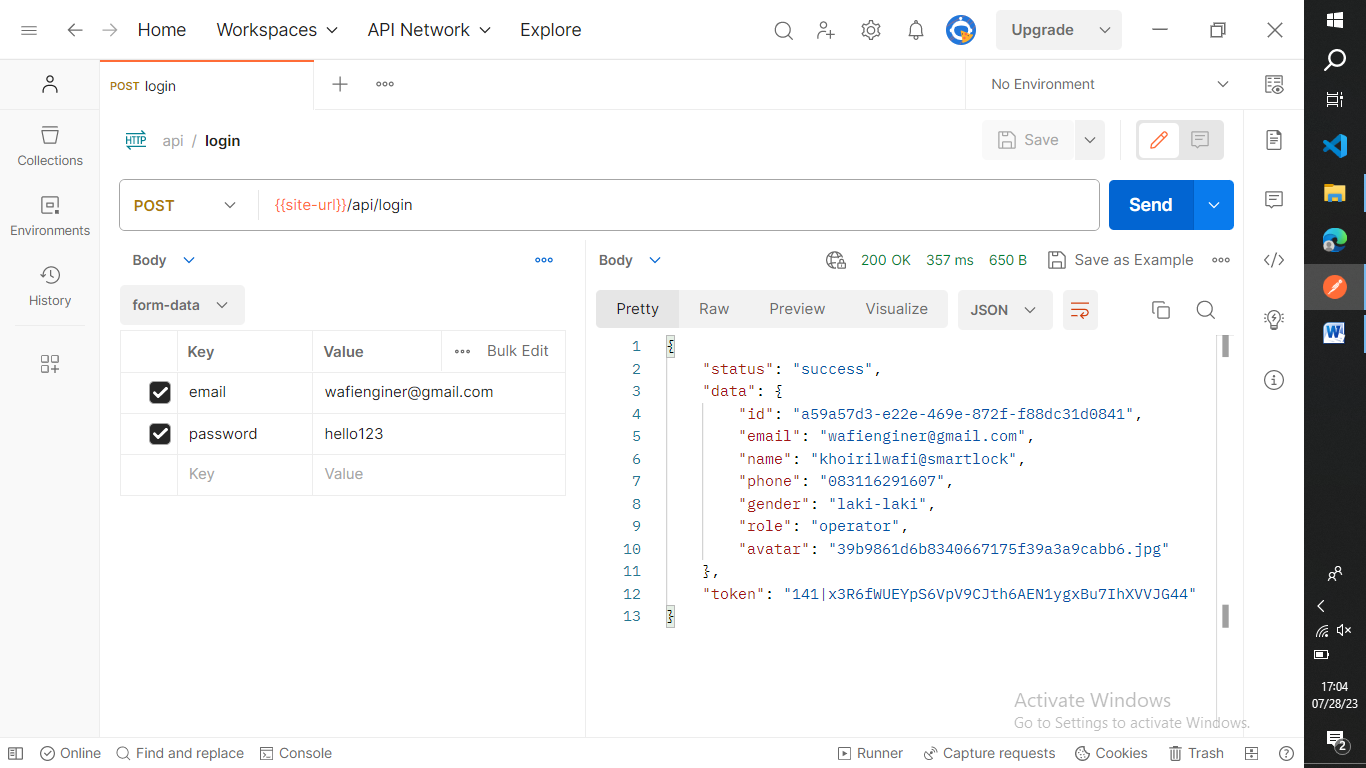
Pada Gambar 3.13 terlihat cara kerja dari library livewire. Livewire bekerja dengan menggunakan sebuah request AJAX untuk memanggil sebuah metode pada kontroler. Jika pada tampilan pengguna melakukan sebuah aksi seperti menekan tombol maka livewire akan mengirimkan sebuah perintaan ke server dengan menggunakan AJAX, permintaan dikirimkan ke sebuah endpoint API yang dibuat secara otomatis oleh livewire untuk menjalankan method-method yang ada pada kontroler. Kontroler akan mengirimkan respon berupa DOM hasil dari perubahan yang dilakukan, setelah diterima oleh client maka DOM tersebut akan dibantingkan dan digadungkan dengan DOM yang sedang tampil dan kemudian dirender. Dengan menggunakan metode ini maka proses pembaharuan tampilan hanya dilakukan pada bagian yang berubah dengan menggunakan manipulasi DOM tanpa memuat seluruh tampilan dari awal.

1. VERIFIKASI
2. Komunikasi WiFi dan Bluetooth

\*butuh snapshoot tampilan serial monitor arduino

1. Backend API Laravel

Proses pengembangan backend API dilakukan dengan menggunakan kerangka kerja Laravel, laravel menyediakan modul-modul yang dapat langsung digunakan untuk membuat sebuah API.



Gambar 4. API Laravel

Gambar 4.1 memperlihatkan contoh hasil request API dengan menggunakan postman, request yang dikirimkan ke endpoint “/api/login” merupakan contoh proses login melalui API yang akan diimplementasikan pada aplikasi mobile. Dapat dilihat pada Gambar 4.1 postman berhasil melakukan login dengan mendapatkan respon dengan kode 200 dan status sukses sehingga penggunaan Laravel untuk membuat sebuah API dapat dilaksanakan pada proses selanjutnya.

1. Kode QR

\*butuh snapshoot aplikasi mobile

1. PENUTUP

Dokumen B300 menjelaskan desain perancangan sistem keamanan kunci pintu gedung berbasis *internet of things* yang meliputi perancangan subsistem komunikasi data dua arah, perangkat penguncian, *database*, *serve*r, metode untuk melakukan verifikasi akses*,* dan aplikasi *mobile*. Hasil perancangan desain sistem keamanan kunci pintu gedung berbasis *internet of things* akan menjadi acuan dalam proses implementasi.