

**UNIVERSITAS DIPONEGORO**

**PERANCANGAN SISTEM *DATABASE* DAN *SERVER* PADA SISTEM KEAMANAN KUNCI PINTU GEDUNG DENGAN *ACCESS CONTROL***

**TUGAS AKHIR**

**MUHAMMAD KHOIRIL WAFI**

**21060119140133**

**FAKULTAS TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO**

**PROGRAM STUDI SARJANA**

**SEMARANG**

**JUNI 2023**



**UNIVERSITAS DIPONEGORO**

**PERANCANGAN SISTEM *DATABASE* DAN *SERVER* PADA SISTEM KEAMANAN KUNCI PINTU GEDUNG DENGAN *ACCESS CONTROL***

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**MUHAMMAD KHOIRIL WAFI**

**21060119140133**

**FAKULTAS TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO**

**PROGRAM STUDI SARJANA**

**SEMARANG**

**JUNI 2023**

HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir ini adalah karya saya sendiri,**

**dan semua sumber baik yang dikutip maupun yang dirujuk**

**telah saya nyatakan dengan benar**

|  |  |  |
| --- | --- | --- |
| NAMA | : | MUHAMMAD KHOIRIL WAFI |
| NIM | : | 21060119140133 |
| Tanda Tangan | : |  |
| Tanggal | : | 23 Juni 2023 |

HALAMAN PENGESAHAN

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI**

**TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Diponegoro, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Khoiril Wafi

NIM : 21060119140133

Program Studi : SARJANA (S1)

Departemen : TEKNIK ELEKTRO

Fakultas : TEKNIK

Jenis Karya : TUGAS AKHIR

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right)* atas karya ilmiah saya yang berjudul: PERANCANGAN SISTEM *DATABASE* DAN *SERVER* PADA SISTEM KEAMANAN KUNCI PINTU GEDUNG DENGAN *ACCESS* *CONTROL.* Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada Tanggal : 23 Juni 2023

Yang menyatakan,

(Muhammad Khoiril Wafi)

NIM. 21060119140133

ABSTRAK

*Diisin nanti setelah perancangan selesai. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod posuere urna, in finibus nisi dignissim sed. Proin in justo eu turpis interdum condimentum. Quisque eu mauris sed massa feugiat dapibus vitae sit amet elit. Donec lacinia neque non magna pellentesque, sed venenatis enim accumsan. Fusce sit amet dui magna. Vivamus fringilla lobortis vulputate. Curabitur id consequat nisl. Aliquam erat volutpat. Nulla facilisi. Vestibulum scelerisque euismod tellus nec eleifend. Nulla eu quam elit. Vivamus non lacus ligula. Aenean eu congue tellus. In suscipit risus ac elit pulvinar, in congue ex pulvinar. Nullam in felis tincidunt, tincidunt ligula et, rutrum velit. Aenean iaculis lobortis nulla nec auctor. Sed luctus elementum purus id dictum.*

***Kata Kunci****: Lorem ipsum dolor sit amet, consectetur adipiscing elit*

ABSTRACT

*Diisin nanti setelah perancangan selesai. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod posuere urna, in finibus nisi dignissim sed. Proin in justo eu turpis interdum condimentum. Quisque eu mauris sed massa feugiat dapibus vitae sit amet elit. Donec lacinia neque non magna pellentesque, sed venenatis enim accumsan. Fusce sit amet dui magna. Vivamus fringilla lobortis vulputate. Curabitur id consequat nisl. Aliquam erat volutpat. Nulla facilisi. Vestibulum scelerisque euismod tellus nec eleifend. Nulla eu quam elit. Vivamus non lacus ligula. Aenean eu congue tellus. In suscipit risus ac elit pulvinar, in congue ex pulvinar. Nullam in felis tincidunt, tincidunt ligula et, rutrum velit. Aenean iaculis lobortis nulla nec auctor. Sed luctus elementum purus id dictum.*

***Key Words****:*

.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta’ala atas rahmat, karunia, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir dan penyusunan laporan ini. Tugas Akhir berjudul “PERANCANGAN SISTEM *DATABASE* DAN *SERVER* PADA SISTEM KEAMANAN KUNCI PINTU GEDUNG DENGAN *ACCESS CONTROL*” ini diajukan sebagai syarat akhir untuk menyelesaikan program Sarjana di Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Semarang. Penyusunan dan penyelesaian Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari semua pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, pada kesempatan kali ini penulis mengucapkan terima kasih kepada:

1. Bapak Aghus Sofwan, S.T., M.T., Ph.D., selaku Ketua Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang.
2. Bapak Dr. Munawar Agus Riyadi, S.T., M.T., selaku Ketua Program Studi S1 Teknik Elektro Universitas Diponegoro Semarang.
3. Bapak Yuli Christiyono, S.T., M.T., selaku Sekretaris Program Studi S1 Teknik Elektro Universitas Diponegoro.
4. Bapak M. Arfan, S. Kom., M. Eng., selaku Dosen Pembimbing Utama yang selalu memberikan bimbingan, arahan, dan motivasi dalam menyelesaikan tugas akhir ini.
5. Bapak Imam Santoso, S.T., M.T., selaku Dosen Pendamping yang selalu memberikan motivasi, bimbingan, serta arahan yang membangun kepada penulis.
6. Bapak Yuli Christiyono, S.T., M.T., selaku dosen wali penulis yang selalu memberikan semangat dan mendengarkan dengan baik permasalahan penulis selama menempuh studi di Departemen Teknik Elektro Universitas Diponegoro.
7. Orang tua dan keluarga besar penulis yang selalu memberikan kasih sayang, doa, motivasi, dan semangat kepada penulis.
8. Sahabat kelompok tugas akhir Henric Dhiki Wicaksono dan Novi Dianasari yang banyak membantu penulis dalam menyelesaikan tugas akhir ini.
9. Keluarga Eternity S1 Teknik Elektro Universitas Diponegoro Angkatan 2019 khususnya konsentrasi Teknologi Informasi, terimakasih atas semuanya selama penulis menimba ilmu di S1 Teknik Elektro Universitas Diponegoro.
10. Dan semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dari awal hingga akhir.

Penulis menyadari bahwa dalam penulisan laporan Tugas Akhir ini tentunya ada kekurangan. Oleh karena itu, kritik dan saran yang dapat membangun diperlukan demi kebaikan dan kesempurnaan penyusunan laporan di masa yang akan datang. Semoga Tugas Akhir ini dapat memberikan manfaat dan menambahkan pengetahuan bagi kita.

Semarang, 23 Juni 2023

(Muhammad Khoiril Wafi)

NIM. 21060119140133

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

# **BAB I**

**PENDAHULUAN**

1. Latar Belakang

Keamanan menjadi hal yang harus diperhatikan dalam sebuah gedung atau bangunan. Pada saat ini sistem penguncian masih banyak menggunakan penguncian tradisional dengan menggunakan kunci fisik yang tidak efisien mengingat jumlah ruangan yang banyak, kunci fisik juga mempunyai tingkat keamanan yang kurang dikarenakan kunci rentan untuk dicuri atau diduplikasi[1].

Mengelola sejumlah besar ruangan menjadi tantangan tersendiri. Seiring bertambahnya ruangan yang harus dikelola maka bertambah juga konpleksitas tugas yang harus dikerjakan seperti mengatur siapa saja yang dapat mengakses ruangan tersebut, bertambahnya ruangan juga meningkatkan resiko kunci hilang atau tertukar sehingga ruangan tidak bisa diakses[2].

Masalah keamanan ruangan dalam sebuah gedung dan efektifitas dapat diselesaikan dengan menggunakan sebuah sistem penguncian cerdas yang terorganisasi dan kerkoneksi ke sebuah sistem manajemen kunci pintu (*access control*) yang memiliki tingkat keamana yang lebih tinggi, adaptif dan fleksibel[3], [4].

Untuk mendukung kinerja dari sistem penguncian yang terorganisasi maka diperlukan sebuah *server* dan penyimpanan data. sebuah *server* akan menjalankan kode program yang bertugas untuk mengatur dan mengawasi semua aktifitas sistem penguncian dan sebuah penyimpanan data digunakan untuk menyimpan data-data seperti data pengguna, kunci, dan *backup*[5].

Dengan demikian, perancangan sistem *database* dan *server* pada sistem keamanan kunci pintu gedung dengan *access control* diharapkan dapat memberikan solusi yang efektif dalam meningkatkan keamanan kunci pintu gedung dan memberikan kenyamanan serta kemudahan dalam pengelolaannya.

1. Tujuan

Tujuan tugas akhir ini adalah untuk merancang sistem *database* dan *server* pada sistem keamanan kunci pintu gedung dengan *access control.*

1. Batasan Masalah

Dalam tugas akhir ini, telah ditentukan batasan-batasan masalah sebagai berikut:

1. Perancangan sistem *database* dan *server* akan difokuskan pada aplikasi *access control* untuk kunci pintu gedung. Hal ini meliputi perancangan struktur *database* untuk menyimpan informasi pengguna, akses pintu gedung, dan riwayat aktivitas akses serta perancangan server untuk menjalankan sistem penguncian.
2. Perancangan sistem *server* akan berfokus pada pemrosesan data dan komunikasi antara sistem *access control* dan perangkat keras.
3. Sistem keamanan kunci pintu gedung dengan *access control* akan mencakup teknologi seperti perangkat lunak *access control* untuk mengatur akses pengguna.
4. Sistem keamanan akan diimplementasikan pada gedung yang memiliki pintu masuk terbatas dengan akses terbatas pada karyawan atau pihak tertentu.
5. Penyimpanan data pengguna dan aktivitas akses akan disimpan dengan cara yang aman dan terenkripsi untuk memastikan keamanan informasi.
6. Pengembangan sistem *access control* harus memenuhi standar keamanan dan regulasi yang berlaku dalam industri atau lingkungan operasional gedung.
7. Sistem keamanan *access control* harus mudah dioperasikan dan dikelola oleh administrator gedung dengan tingkat keamanan dan kontrol yang optimal.
8. Sistematika Penulisan

Sistematika penulisan dalam laporan Tugas Akhir ini adalah sebagai berikut:

**BAB I PENDAHULUAN**

Bab ini berisi latar belakang, tujuan, batasan masalah, dan sistematika penulisan.

**BAB II DASAR TEORI**

Bab ini berisi teori yang melandasi dalam perancangan.

**BAB III PERANCANGAN SISTEM**

Bab ini berisi tentang perancangan sistem *database* dan *server* serta sistem keamanan kunci pintu gedung dengan *access control*.

**BAB IV PENGUJIAN DAN ANALISIS**

Pada bab ini dilakukan pengujian dan analisis hasil pengujian.

**BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran terhadap hasil pengujian dan analisis yang telah dibuat.

**BAB II**

**DASAR TEORI**

1. Kendali Akses (Access Control)

Kendali akses atau *access control* merupakan sebuah mekanisme pengaturan kebijakan yang digunakan untuk membatasi dan mengatur hak akses pengguna terhadap suatu sumber dayaatau fasilitas tertentu. Kendali akses dapat diaplikasikan secara *hardware* maupun *software* yang akan menjadi pintu masuk pengguna sebelum mengakses suatu fasilitas. Dengan menggunakan kendali akses kita bisa mengatur dan membatasi akses pengguna sehingga hanya pengguna tertentu yang diijinkan yang bisa mengakses sumber daya yang dilindungi[6].

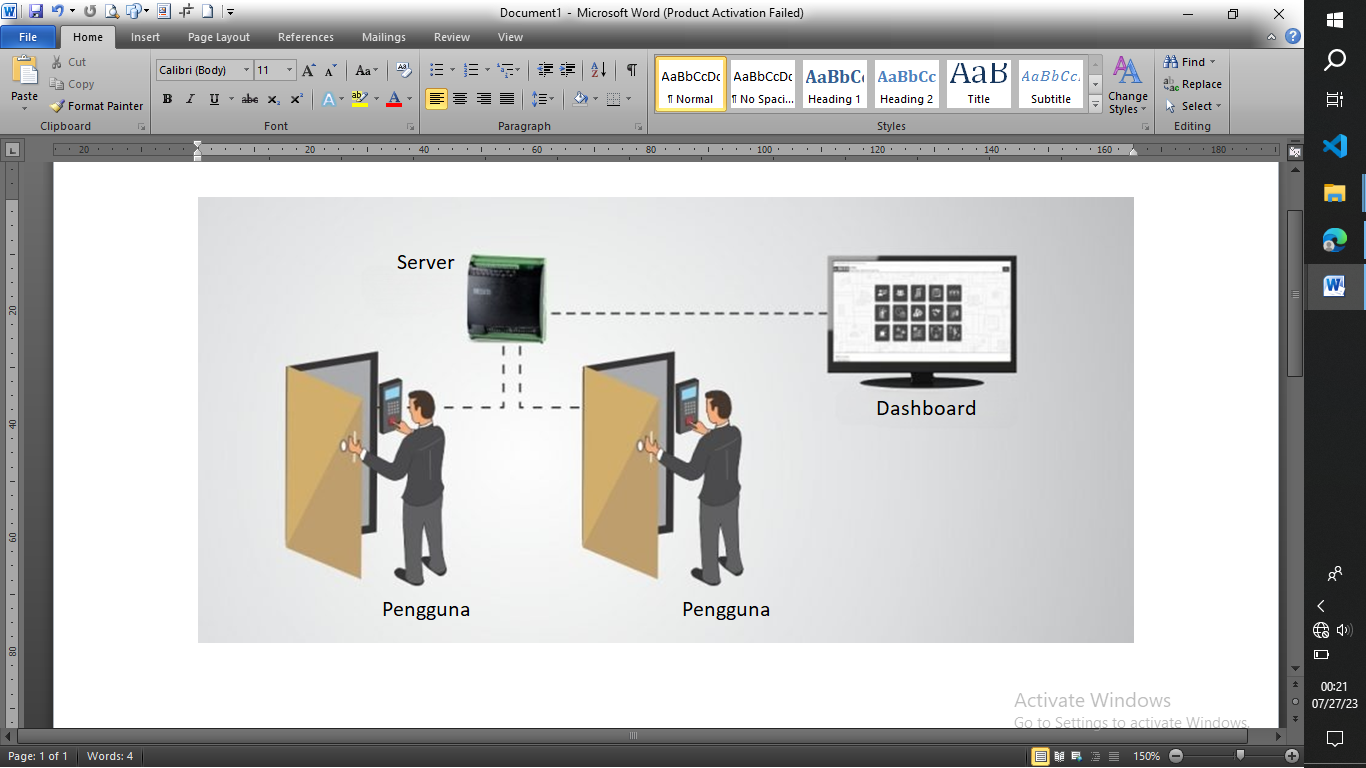
Terdapat beberapa pendekatan yang biasanya digunakan untuk membatasi akses pengguna untuk mengakses suatu sumber daya yang dimiliki, diantaranya yaitu :

1. *Access Control List*

ACL atau *Access Control List*  merupakan sebuah daftar yang berisi aturan-aturan atau daftar pengguna yang diijinkan untuk mengakses sebuah sumber daya yang dilindungi[7]. Dengan menggunakan ACL kita dapat mambatasi hak akses dengan hanya mengijinkan pengguna yang terdapat didalam daftar untuk mengakses sumber daya atau fasilitas tertentu. ACL akan memberikan pembatasan akses pada setiap pengguna secara ketat dan hanya mengijinkan akses jika pengguna memiliki hak akses yang tersimpan didalam daftar akses.

1. *Role-based Access Control*

RBAC atau *Role-based Access Control* merupakan pendekatan kendali akses yang tidak memberikan hak akses langsung ke pengguna melainkan memberikan hak akses berdasarkan peran yang dimiliki oleh pengguna seperti *admin*, operator, *supervisor*, dan lain sebagainya. Metode pengendalian akses berbasis peran akan memberikan aturan-aturan terkait akses berdasarkan peran pengguna tanpa mengetahui identitas pengguna secara spesifik dan hanya menggunakan peran pengguna sebagai parameter yang digunakan untuk melakukan pengendalian akses sumber daya.

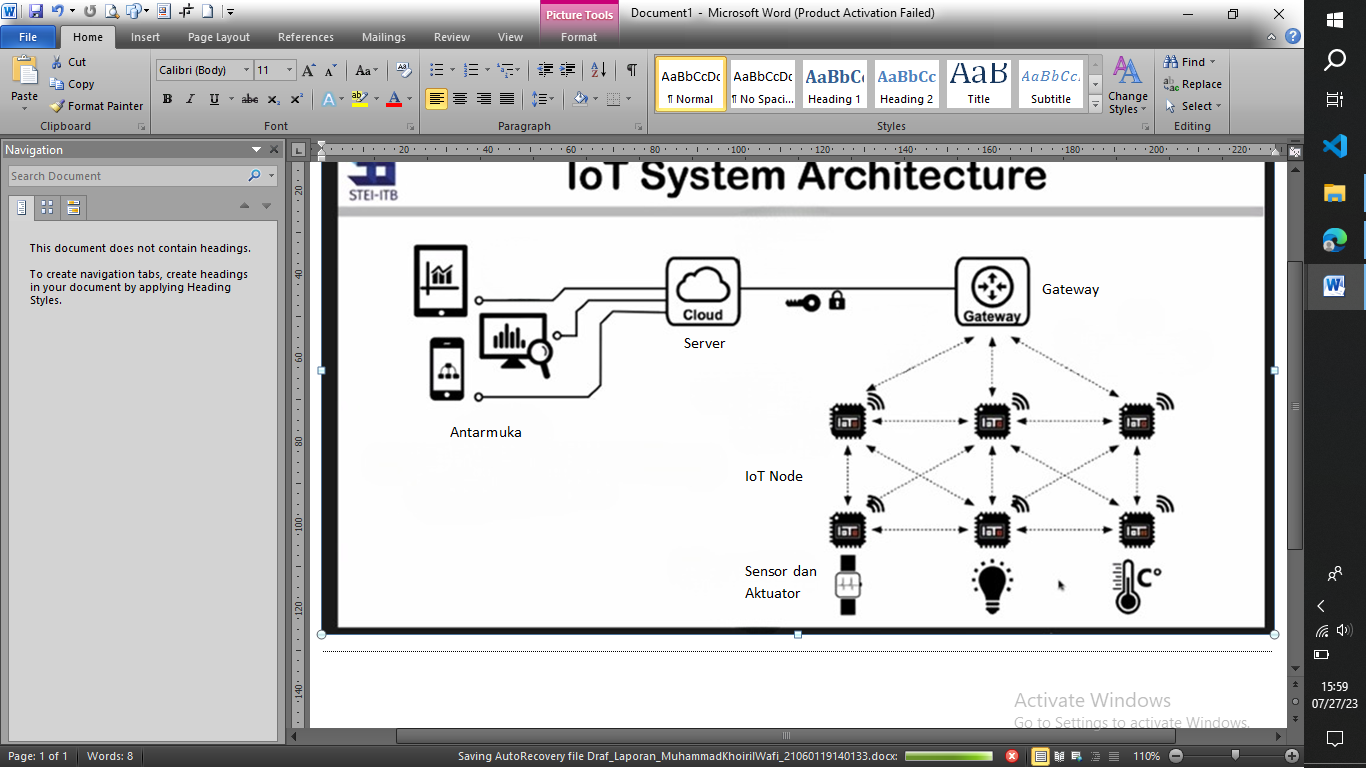


**Gambar 2.1** Contoh Pengendalian Akses

Gambar 2.1 merupakan contoh pengendalian akses pada pintu untuk membatasi akses pengguna ke suatu ruangan. Setiap pengguna memiliki identitas unik yang digunakan untuk menentukan tindakan apakah pengguna tersebut diijinkan atau tidak, proses perijinan dilakukan secara otomatis dan dikendalikan oleh sebuah *server* sebagai pusat kendali dan pusat data. Pada sistem pengendalian akses juga terdapat sebuah *sofware* yang digunakan sebagai antar muka untuk melakukan pengelolaan hak akses seperti mengijinkan pengguna baru atau menghapus ijin pengguna dan juga melihat riwayat akses pengguna.

1. Internet of Things

*Internet of Things* atau IoT merupakan sebuah konsep yang digunakan untuk mengembangkan koneksifitas internet, IoT juga memberikan gambaran mengenai kemampuan dari berbagai perangkat elektronik yang saling terhubung dengan membentuk sebuah jaringan komunikasi baik melalui internet maupun komunikasi lainnya seperti bluetooth. Dengan menggunakan konsep IoT kita dapat menghubungkan peralatan seperti sensor dan aktuator yang terhubung ke ke sebuah jaringan menjadi sebuah kesatuan sistem yang dapat dikendalikan secara efektif dan efisien dengan tingkat kerumitan yang rendah[8].



**Gambar 2.2** Arsitektur Sistem IoT

Gambar 2.2 menjelaskan arsitektur dari sistem IoT, sebuah sistem IoT dibangun dari beberapa bagian yang membentuk suatu sistem yang dapat berkomunikasi antara perangkat IoT dan juga server. Bagian-bagian dari sistem IoT diantaranya :

* 1. Sensor dan Aktuator

Sensor dan aktuator akan menjadi penghubung yang menjembatani interaksi antara sistem dengan dunia luar. Sebuah sensor akan melakukan akuisisi data yang merupakan sumber data dari sebuah sistem IoT sedangkan aktuator akan menerima data dan memberikan keluaran mekanis seperti menyalakan lampu, membuka kunci dan lain sebagainya.

* 1. IoT *Node*

IoT *node* bertindak sebagai penghubung antara perangkat keras seperti mikrokontroler kedalam sebuah jaringan IoT dengan adanya IoT *node* memungkinkan antar perangkat IoT untuk saling berkomunikasi satu dengan lainnya secara independen[8].

* 1. *Gateway*

Pada sistem IoT umumnya akan terhubung ke sebuah server sebagai pusat kendali dan pusat data. *Gateway* menjadi penghubung antara IoT *node* dengan jaringan internet global, sebuah *gateway* misalnya *router* dan modem akan menyediakan koneksi internet yang dapat digunakan oleh perangkat IoT untuk terhubung ke *server* melalui koneksi internet[8].

* 1. *Server*

*Server* akan menjadi pusat kendali dan penyimpanan data terpusat pada sistem IoT. Dengan adanya sebuah *server* memungkinkan perangkat IoT untuk saling saling berkoordinasi dengan *server* sebagai pengaturnya, dengan adanya *server* juga memungkinkan perangkat IoT melakukan penyimpanan data dan informasi secara *online* pada *database*[8].

* 1. Antarmuka

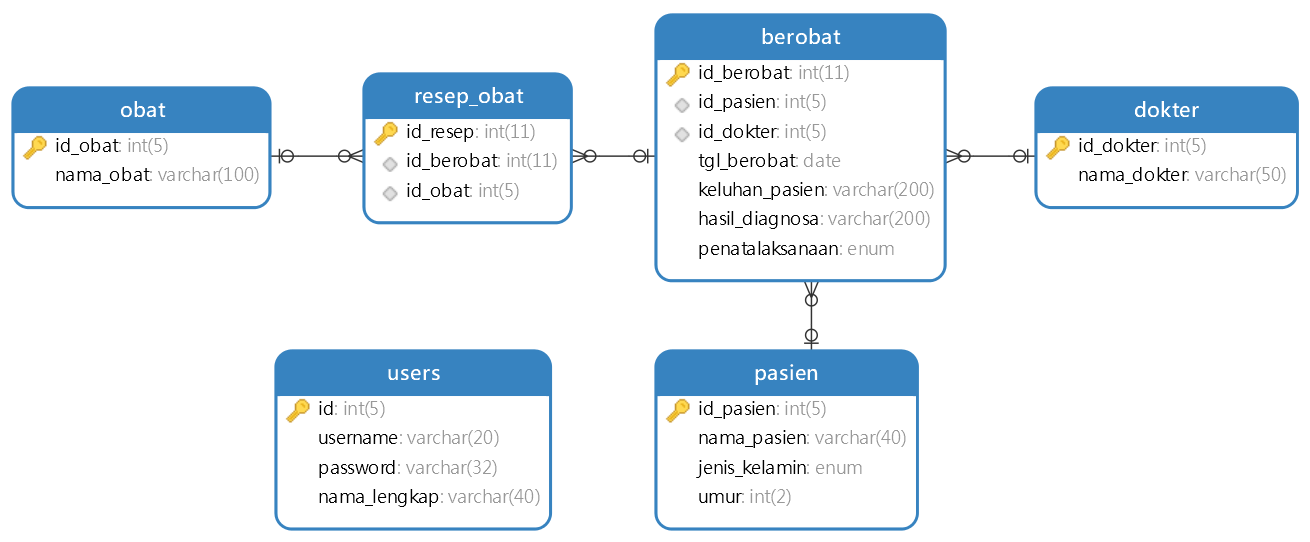
Antarmuka pada sistem IoT digunakan untuk melakukan proses pemantauan aktifitas dan data yang terjadi didalam sistem IoT. Antarmuka akan memberikan akses kepada pengguna untuk berinteraksi dengan sistem IoT seperti melihat data dan memberikan perintah ke perangkat IoT.

1. *Database* MySQL

*Database* atau basisdata merupakan sekumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dicari, diambil, ditambahkan, dan diolah dengan tepat[6]. Selain berisikan data, *database* juga berisikan informasi lainnya seperti tipe data, nama-nama kolom dan baris. Adapun fungsi lain dari *database* yaitu[9] :

1. Mempermudah identifikasi data dengan cara pengelompokkan data, salah satu contohnya dengan pembuatan beberapa tabel atau *field* yang berbeda-beda.
2. Meminimalisir suatu data ganda.
3. Penyimpanan secara digital.
4. Menjadi alternatif lain terkait masalah penyimpanan ruang dalam suatu aplikasi.

MySQL merupakan singkatan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. MySQL merupakan sistem manajemen *database* yang bersifat relational. Artinya, data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat[10].

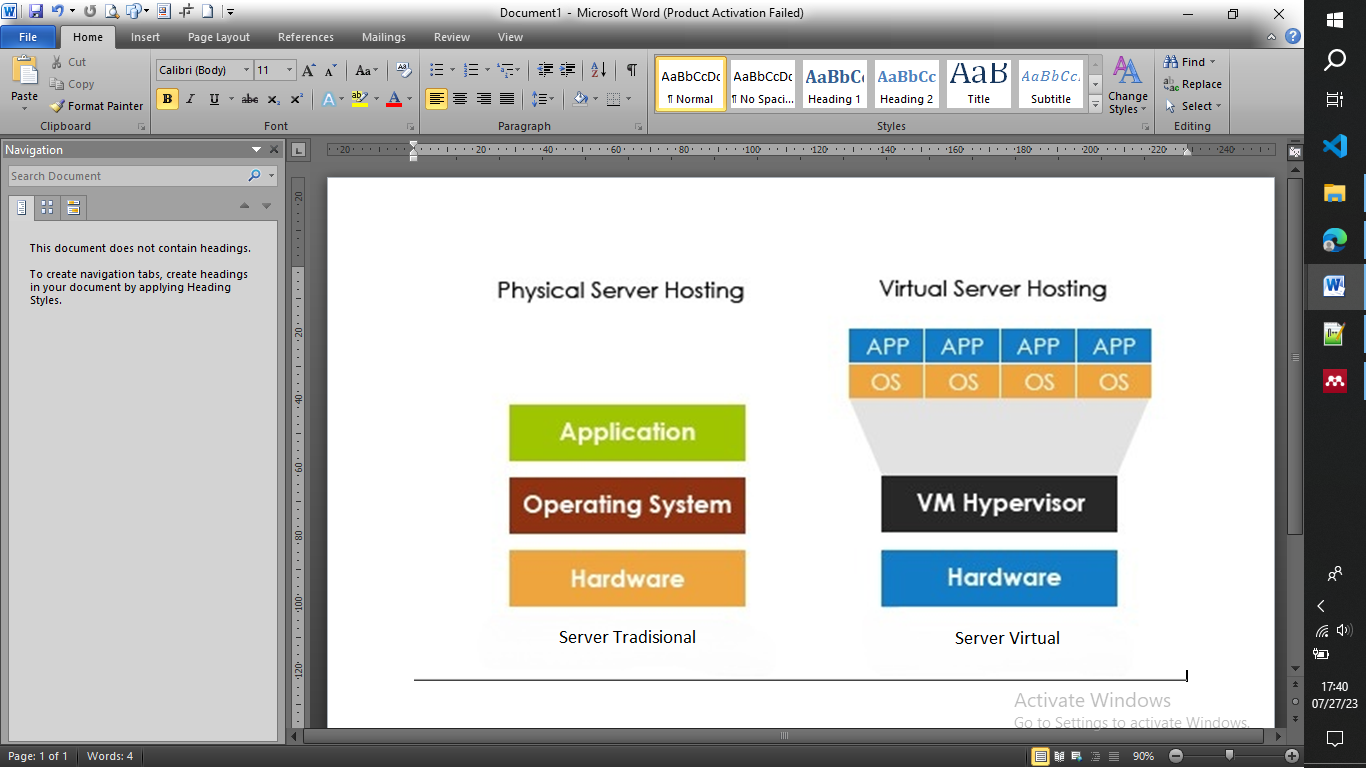


**Gambar 2.3** Struktur Tabel MySQL

MySQL merupakan sistem database relasional yang berarti pada MySQL data akan disimpan dalam tabel-tabel yang terdiri dari beberapa kolom, gambar 2.3 menjelaskan strukur data yang tersimpan didalam *database* MySQL, data yang disimpan dapat berupa teks, angka, *boolean*, waktu, hari dan lain sebagainya tergantung dengan tipe data pada setiap kolomnya. Dalam MySQL terdapat relasi antar tabel yang memberikan informasi hubungan antara tabel satu dengan tabel lainnya serta perilakunya terhadap perubahan yang terjadi pada tabel induknya.

1. *Virtual Private Server*

*Virtual Private Server* atau VPS adalah jenis server yang secara eksklusif diperuntukkan bagi satu pengguna, sehingga seluruh sumber daya yang ada di dalamnya tidak dipengaruhi atau dibagi dengan pengguna lain. Dengan menggunakan teknologi VPS, sebuah mesin fisik dapat menjalankan beberapa sistem operasi secara bersamaan. Pengguna VPS memiliki kendali penuh untuk mengatur seluruh konfigurasi sesuai kebutuhan. Teknologi yang digunakan dalam VPS adalah virtualisasi hardware pada server fisik yang memungkinkan pembagian sumber daya menjadi beberapa bagian yang berbeda, sehingga setiap VPS berfungsi seperti server pribadi yang terisolasi dari pengguna lainnya[11].

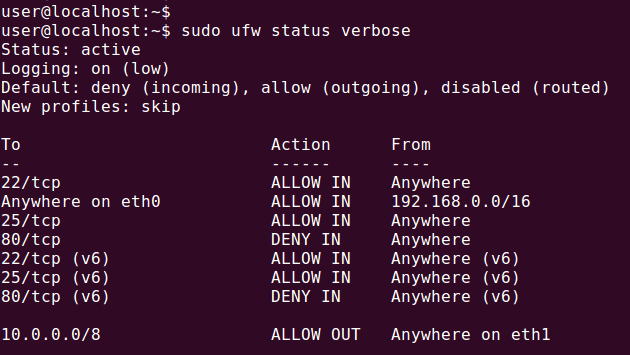


**Gambar 2.4** Perbedaan Server Tradisional dan Virtual

Gambar 2.4 memperlihatkan perbedaan antara *server* tradisional dengan *server* virtual, pada *server* tradisional satu *server* fisik hanya bisa digunakan untuk menjalankan satu sistem operasi dan satu aplikasi sedangkan pada *server* virtual memungkinkan menjalankan beberapa sistem operasi dan aplikasi yang banyak didalam satu *server* fisik. Perbedaan utama yaitu adanya virtualisasi pada server virtual yang bertugas untuk membagi sumber daya *server* fisik menjadi beberapa bagian untuk menjalankan sistem operasi. Proses pembagian ini dilakukan menggunakan perangkat lunak, sehingga satu server fisik dapat menampung beberapa VPS yang berjalan secara terpisah. Setiap VPS memiliki akses penuh (*Full Root Access*) untuk mengatur sistem operasi dan konfigurasi sesuai keinginannya, termasuk pengaturan *init script*, pengguna, pemrosesan data, sistem file, serta sumber daya server seperti CPU dan RAM yang beroperasi secara independen.

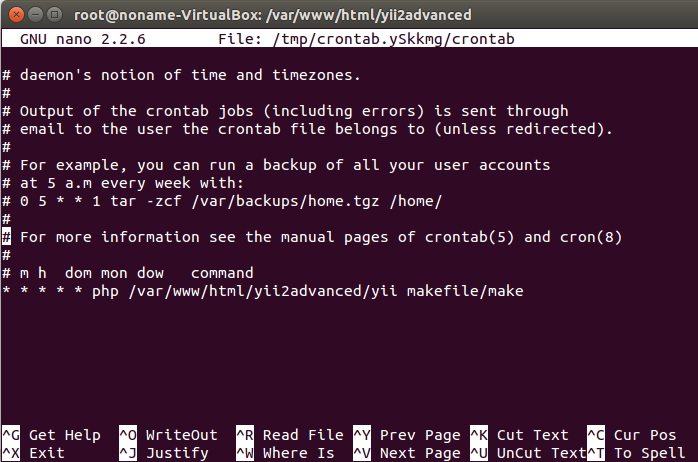
1. Sistem Operasi Ubuntu

Ubuntu adalah salah satu distribusi Linux yang didasarkan pada Debian. Distro ini menyediakan sistem operasi berbasis Debian dengan jadwal rilis yang teratur dan dukungan yang luas, baik untuk pengguna perorangan maupun perusahaan. Selain digunakan sebagai sistem operasi desktop, Ubuntu juga populer digunakan sebagai sistem operasi pada *server*, karena memiliki karakteristik yang ringan dan dapat diandalkan dalam menangani berbagai tugas *server*[12]. Ubuntu menyediakan fitur yang cukup lengkap untuk diaplikasikan pada *server* seperti adanya *firewall* dan penjadwalan menggunakan cronjob.



**Gambar 2.5** Contoh Konfigurasi Firewall Ubuntu

Gambar 2.5 merupakan contoh konfigurasi *firewall* pada sistem operasi ubuntu. Di dalam *firewall* semua komunikasi yang keluar dan masuk akan difilter atau dikontrol, *port*-*port* yang rentan servisnya dapat di tutup atau diblokir, sehingga hanya pihak yang diizinkan saja yang boleh lewat. Cara ini merupakan salah satu pengamanan jaringan yang sering digunakan[13]. Selain *firewall* ubuntu juga menyediakann sistem penjadwalan dengan menggunakan cronjob.

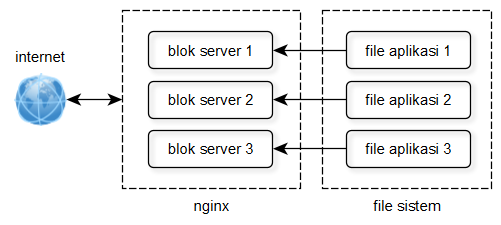


**Gambar 2.6** Contoh Konfigurasi Cronjob

Gambar 2.6 merupakan contoh konfigurasi conjob pada sistem operasi ubuntu, dengan menggunakan cronjob maka dimungkinkan untuk mengeksekusi perintah pada terminal secara berulang setiap rentang waktu tertentu seperti setiap 1 minggu sekali atau 1 jam sekali.

1. Nginx

Nginx merupakan perangkat lunak *open-source* yang memiliki performa tinggi sebagai server HTTP dan *reverse proxy*. Nginx memiliki keunggulan dalam memberikan konten statis secara cepat, sambil menggunakan sumber daya sistem dengan efisien. Selain itu, Nginx dapat mendistribusikan konten HTTP dinamis melalui jaringan menggunakan FastCGI *handler* untuk menjalankan *script*[14]. Nginx dibangun secara modular, dengan demikian nginx mampu mendukung berbagai fitur seperti *Load Balancing* dan *Reverse Proxying*, *Virtual hosts*.

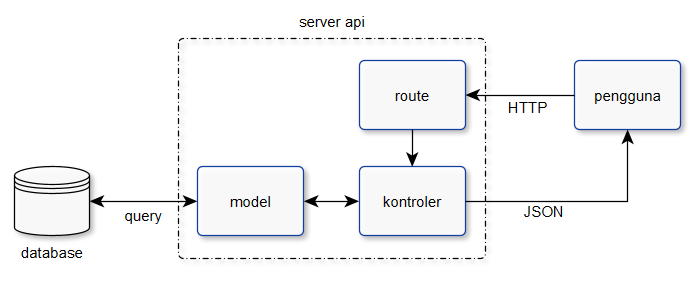


**Gambar 2.7** Virtual Host pada Nginx

Gambar 2.7 menjelaskan penggunaan virtual host pada nginx, dengan menggunakan virtual host nginx dapat menangani beberapa aplikasi yang sudah dikonfigurasi pada masing-masing *port*-nya. Nginx menggunakan pendekatan *asynchronous-event* untuk menangani permintaan sehingga nginx tidak sepenuhnya bergantung pada ‘*thread*’ untuk menangani permintaan (*request*). Arsitektur ini memberikan hasil kinerja nginx pada saat dibebani hanya memerlukan sedikit memori dalam jumlah yang bisa diprediksikan[14].

1. *Application Programming Interface*

*Application Programming Interface* atau API merupakan sebuah protokol yang digunakan untuk menampilkan layanan atau data yang disediakan oleh sebuah aplikasi melalui sumber daya yang telah ditentukan. Dengan menggunakan API, aplikasi lain dapat mengakses layanan atau data tanpa harus mengimplementasikan prosedur atau objek yang mendasarinya[15].



**Gambar 2.8** Cara Kerja API

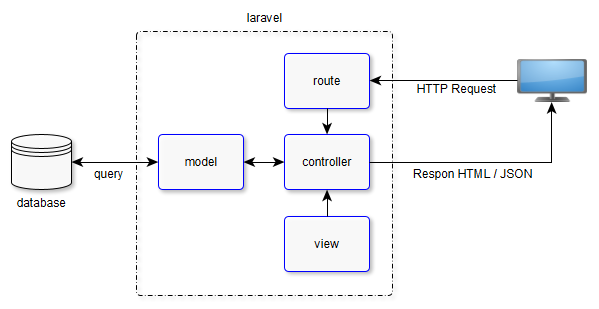
Gambar 2.8 menjelaskan cara kerja dari API, pengguna (*client*) akan mengirimkan permintaan HTTP melalui sebuah *endpoint*, *endpoint* merupakan sebuah URL yang digunakan untuk berinteraksi dan mengakases sumber daya yang disediakan oleh API tersebut. API akan mengirimkan data dengan format JSON sebagai respon dari permintaan *client*. JSON atau Java Script Object Notation merupakan sebuah format data yang yang ditemukan oleh Douglas Crockford pada tahun 2006, JSON menggunakan sintak objek javascript sebagai format data berbasis teks, JSON memiliki ukuran data yang lebih kecil serta waktu proses yang lebih cepat jika dibandingkan dengan format data lainnya[16].

Dengan menggunakan sebuah API maka pengguna tidak akan terhubung secara langsung dengan *database*, sehinggga akan meningkatkan keamanan dari sistem, jika sewaktu waktu terjadi hal-hal yang tidak di ingin kan seperti rusaknya data atau sistem di retas oleh pihak yang tidak bertanggung jawab data asli tetap aman tersimpan pada database tanpa mengalami gangguan sedikitpun[16].

1. *Framework* Laravel

*Framework* adalah struktur kerja yang terdiri dari sekumpulan program, terutama berisi kelas dan fungsi, yang dapat membantu para pengembang (*developer*) dalam menangani berbagai permasalahan yang sering dihadapi dalam pemrograman. Dengan menggunakan framework, para pengembang dapat lebih terfokus dan lebih cepat dalam membangun aplikasi karena berbagai tugas rutin seperti koneksi ke database, pemanggilan variabel, pengelolaan file, dan sebagainya telah diatur dan disediakan dalam framework tersebut[17].

Laravel merupakan kerangka kerja pemrograman web yang menggunakan bahasa pemorgraman PHP yang terbuka dan gratis, laravel diperuntukkan untuk pengembangan aplikasi berbasis *website* maupun API dengan menggunakan pendekatan pola MVC atau Model View Controller[18]. Arsitektut MVC memiliki business logic yang terpisah dari model dan *presentation*, sehingga saat melakukan modifikasi pada program tidak mempengaruhi komponen lain yang tidak diubah, dan proses pengembangan yang lebih cepat, serta dapat menggunakan *reuse of code* dimana fungsi ini berguna dalam pengembangan website tanpa harus melakukan coding dari awal[17].



**Gambar 2.9** Konsep MVC pada Laravel

Gambar 2.9 menjelaskan kosep MVC yang ada pada Laravel, saat pengguna melakukan *request* ke Laravel maka *request* tersebut akan ditangani oleh *route*, *route* disini bertugas sebagai pintu masuk yang menyediakan *url*/*enpoint* mana saja yang dapat diakses oleh *client*. Jika tidak ditemukan masalah maka *route* akan melanjutkan *request* tersebut ke modul MVC dengan tugas sebagai berikut :

1. Model

Model merupakan sebuah modul yang digunakan untuk melakukan proses *query* ke *database*, pada laravel model menyediakan mekanisme *query* *builder* yang menjamin proses *query* berjalan dengan mudah dan aman dengan berbagai filter untuk menghindari kesalahan.

1. *View*

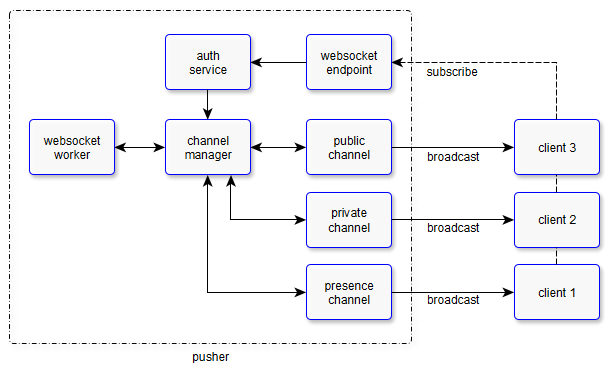
*View* merupakan bagian dari Laravel yang berfungsi untuk menyediakan tampilan antarmuka kepada pengguna, antarmuka akan tampil pada *browser* pengguna. Pada implementasi Laravel sebagai API maka *view* tidak mendapat tugas khusus karena pada API *controller* akan mengembalikan respon dalam format JSON dan tidak mengembalikan tampilan dalam HTML.

1. *Controller*

*Controller* digunakan untuk mengolah data yang diminta oleh pengguna, sebuah *controller* tidak langsung menerima permintaan dari pengguna melainkan *controller* akan dipanggil oleh bagian routing pada saat ada permintaan. *Controller* juga akan mengirimkan respon ke *client* sesuai dengan permintaan yang dikirimkan.

1. *Pusher Web*s*ocket*

*Websocket* merupakan sebuah protokol komunikasi web berbasis *client*-*server*, keberadaan websocket dinilai dapat menggantikan teknilogi AJAX sebagai pendahulu komunikasi *client*-*server*. *Websocket* merupakan teknologi yang mampu memberikan performa terbaik ketika diimplementasikan dalam sistem dengan *rate*-*request* tinggi, dibandingkan dengan teknologi komunikasi lain termasuk AJAX[19]. *Websocket* memungkinkan komunikasi dua arah antara *client* dan *server* dengan menggunakan koneksi yang sudah terjalin, hal ini dikarenakan pada *websocket*, koneksi akan terus terjalin selama tidak terjasi error atau ada permintaan pemutusan koneksi. Salah satu layanan yang menyediakan koneksi *websocket* yaitu Pusher, Pusher menyediakan komunikasi *realtime* antara *server* dan *client* melalui *channel*-*channel* yang telah tersedia.



**Gambar 2.10** Diagram Pusher Channel

Gambar 2.10 diatas menjelaskan bagian-bagian dari pusher channel, pusher hanya mendukung komunikasi satu arah yaitu dari server ke client dengan menggunakan metode broadcasting. Untuk bisa mendapatkan pesan dari broadcasting maka client harus terhubung ke salah satu channel dengan cara melakukan subscribe ke channel tersebut. Didalam pusher terdapat 3 channel yaitu :

* 1. *Public* *Channel*

*Public* *Channel* merupakan *channel* yang tersedia dan dapat di-*subscribe* oleh semua penggguna tanpa adanya proses autentikasi.

* 1. *Private* *Channel*

*Private* *Channel* merupakan *channel* yang hanya tersedia untuk *client* tertentu, sebelum terhubung ke *private* *channel* *client* harus melakukan proses autentikasi untuk mendapatkan kode *signature* channel yang digunakan untuk melakukan *subscribe* ke *channel* tersebut.

* 1. *Presence* *Channel*

*Presence* *Channel* merupakan pengembangan dari *channel* *private*, pada *presence* *channel* perilaku *client* dapat dipantau seperti ada client baru yang melakukan subcribe atau ada client yang keluar.

1. Penelitian Terdahulu

Beberapa penelitian terdahulu yang dijadikan referensi dalam perancangan sistem database dan server pada sistem keamanan kunci pintu gedung menggunakan access control adalah :

**Tabel 2.1** Ringkasan Penelitian Terdahulu

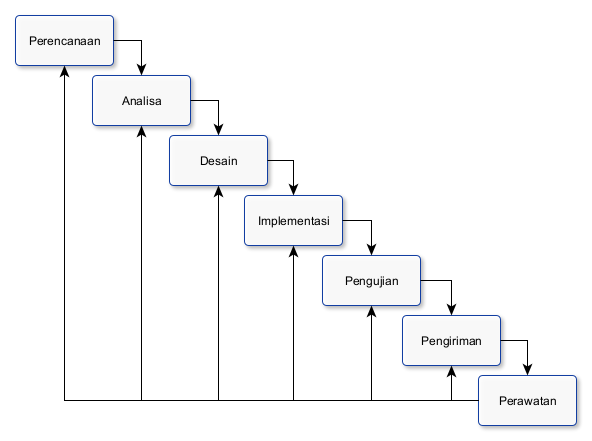
|  |  |  |  |
| --- | --- | --- | --- |
| **Judul** | **Tahun** | **Penulis** | **Pembahasan** |
| *Perancangan Database IoT Berbasi Cloud dengan Restful API* | 2021 | M. Kasyful Anwar | Dalam penelitian ini dibahas mengenai rancangan database berbasis cloud dengan Restful API untuk IoT agar data IoT aman dan memiliki *throughput* yang bagus dengan struktur data yang diatur pada database. |
| *Development of Intelligent Door Lock System for Room Management Using Multi Factor Authentication* | 2023 | I. Hermawan, D. Arnaldy, P. Oktivasari, dan D. A. Fachrudin | Penelitian ini menjelaskan mengenai permasalahan yang muncul pada manajemen kunci secara tradisional seperti jumlah kunci yang banyak, mudah hilang dan mudah diduplikasi. |
| *RFID and GSM Based Attendance Monitoring System using door locking / unlocking system and Its Hardware Implementation* | 2015 | A. Jain, V. L. Kalyani, dan B. Nogiya | Penelitian ini menjelaskan penggunaan RFID untuk melakukan absesnsi siswa sekaligus sebagai kunci untuk membuka pintu ruang kelas. |

BAB III

**PERANCANGAN SISTEM**

1. Metode Perancangan

Dalam perancangan tugas akhir ini menggunakan pendekatan waterfall dengan pertimbangan yang matang. Metode waterfall memberikan struktur dan ketertiban yang sangat diperlukan dalam pengembangan proyek ini. Setiap tahapan akan dilaksanakan secara berurutan sesuai dengan gambar x.x yang telah dirancang sebelumnya.



Keputusan ini diambil karena proyek tugas akhir ini memiliki batasan waktu yang jelas dan persyaratan yang sudah ditetapkan dengan baik. Dengan menggunakan metode waterfall, dapat memastikan bahwa setiap tahapan diselesaikan dengan cermat sebelum melanjutkan ke tahap berikutnya, sehingga mengurangi risiko kesalahan dan memastikan kelancaran jalannya proyek. Selain itu, pendekatan ini juga memungkinkan untuk melakukan perencanaan yang lebih terperinci, memahami kebutuhan proyek secara menyeluruh, dan mengidentifikasi risiko potensial dengan lebih baik. Dengan demikian, metode waterfall menjadi pilihan yang tepat untuk memastikan keberhasilan dan kualitas dari tugas akhir ini.

1. Perencanaan

Pada tahap perencanaan dilakukan pengumpuna informasi mengenai tujuan serta batasan-batasan proyek yang harus terpenuhi. Pada tahap perencanaan juga mendefinisikan beberapa fungsional primer dan sekunder pada rancangan sistem untuk dikembangkan lebih lanjut.

1. Analisa

Pada tahap analisa dilakukan studi lieratur mengenai tujuan dan batasan-batasan proyek serta melakukan pengumpulan informasi mengenai sistem yang akan dibuat dengan melihat dan mempelajari penelitian/perancangan terdahulu. Analisa dilakukan bersama kelompok dengan melakukan penelusuran pustaka serta referensi yang relevan dan disertai dengan diskusi sehingga menghasilkan keluaran berupa gambaran umum mengenai bagian-bagian dari sistem yang akan digunakan pada tahap selanjutnya.

1. Desain

Setelah mendapatkan gambaran mengenai sistem yang akan dibuat, selanjutnya dilakukan pemodelan sistem dengan menggunakan diagram pemodelan. Diagram pemodelan digunakan untuk menggambarkan hubungan dan interaksi antar komponen didalam sistem, dengan menggunakan diagram maka akan menghasilkan gambar sistem yang lebih rinci yang digunakan untuk membangun sistem.

1. Implementasi

Pada tahap implementasi, gambar pemodelan dari sistem kemudian diterjemahkan atau diimplementasikan menggunakan kode program untuk membuat sistem dapat bekerja sesuai dengan dasain awal yang telah ditentukan.

Implementasi menggunakan bahasa pemrograman PHP dengan menggunakan framework Laravel. Laravel mempunyai karakteristik RAD atau Rapid Application Development sehingga implementasi dapat dilaksanakan dengan cepat dan efisien, Laravel juga menyediakan komponen-komponen yang akan digunakan didalam sistem seperti autentikasi, antrian, notifikasi pesan, websocket, dan lain sebagainaya sehingga dapat mempermudah proses implementasi.

1. Pengujian

Pada tahap pengujian, sistem yang telah dibangun diuji kinerjanya dengan mengirimkan beberapa permintaan data dan menganalisa hasil keluaran yang diberikan apakah sesuai atau tidak.

Proses pengujian dilakukan dengan menggunakan software Postman. Postman merupakan aplikasi yang digunakan untuk melakukan pengujian API baik berbasis HTTP maupun Socket. Pengujian lebih lanjut dilakukan untuk menguji kemanan pada autentikasi serta koneksi websocket yang terjalin.

1. Pengiriman

Pada tahap pengiriman, aplikasi yang sudah selesai dilakukan pengujian dan dinyatakan lolos uji selanjutkan akan dipasang atau diinstall pada server. Pada proses deployment juga akan dilakukan pemasangan aplikasi tambahan seperti basis data, penjadwalan, firewall, dan lain sebagainya. Pada tahap ini sistem yang sudah dibuat dapat digunakan oleh pengguna umum.

1. Pemeliharaan

Pada tahap pemeliharaan atau perawatan dilakukan pengecekan secara berkala pada aplikasi yang sudah berjalan pada server. dengan adanya perawatan secara teratur jika ditemukan permasalahan atau komponen yang tidak bekerja maka akan langsung diperbaiki sehingga tidak mengganggu kinerja dari server.

1. Perancangan Database

Database digunakan untuk menyimpan data yang digunakan dalam sistem penguncian pintu gedung seperti data pengguna, data pintu, data riwayat akses dan lain sebagainya. Database yang digunakan harus dapat menunjang kinerja sistem dengan karakteristik respon yang cepat dan pengelolaan data terstruktur.

Berdasarkan penelitian [20] 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.

1. Tabel Pengguna

Tabel pengguna (users) menyimpan data pengguna yang akan digunakan sebagai identitas pengguna untuk keperluan autentikasi dan digunakan juga untuk menyimpan riwayat akses. Struktur dari tabel pengguna dapat dilihat pada tabel x.x berikut :

Tabel Pengguna

| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- | --- |
| 1 | id | char(36) | kunci primer tabel |
| 2 | added\_by | char(36) | referensi aktor |
| 3 | email | varchar(255) | alamat email pengguna |
| 4 | password | varchar(255) | password pengguna |
| 5 | remember\_token | varchar(255) | token untuk fitur "ingat saya" |
| 6 | name | varchar(255) | nama penggua |
| 7 | phone | varchar(255) | nomor hp pengguna |
| 8 | gender | enum | jenis kelamin pengguna |
| 9 | role | enum | jabatan pengguna |
| 10 | avatar | varchar(255) | foto profil pengguna |
| 11 | email\_verified\_at | timestamp | waktu verifikasi email |
| 12 | created\_at | timestamp | waktu data dibuat |
| 13 | updated\_at | timestamp | waktu data diedit |

1. Tabel Gedung

Tabel gedung (offices) digunakan untuk meyimpan data gedung. Pada sistem penguncian pintu gedung ini terdapat beberapa gedung dengan masing-masing operator yang mengatur pengelolaan kunci gedung. Struktur dari tabel gedung dapat dilihat pada tabel x.x berikut :

Tabel Gedung

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | id | char(36) | kunci primer tabel |
| 2 | user\_id | char(36) | referensi operator gedung |
| 3 | name | varchar(255) | nama gedung |
| 4 | created\_at | timestamp | waktu data dibuat |
| 5 | updated\_at | timestamp | waktu data diedit |

1. Tabel Pintu

Tabel pintu (doors) digunakan untuk menyimpan data pintu pada setiap gedung. Tabel pintu akan memberikan informasi terkait pintu seperti status koneksi dan status penguncian. Struktur dari tabel pintu dapat dilihat pada tabel x.x berikut :

Tabel Pintu

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | id | char(36) | kunci primer tabel |
| 2 | office\_id | char(36) | referensi gedung |
| 3 | socket\_id | char(36) | identitas websocket |
| 4 | name | varchar(36) | nama pintu |
| 5 | device\_name | varchar(36) | username pintu |
| 6 | device\_pass | varchar(36) | password pintu |
| 7 | key | varchar(36) | kunci pintu |
| 8 | is\_lock | tinyint(1) | status penguncia |
| 9 | created\_at | timestamp | waktu data dibuat |
| 10 | updated\_at | timestamp | waktu data diedit |

1. Tabel Jadwal

Tabel jadwal (schedules) digunakan untuk menyimpan jadwal operasional pintu secara otomatis. Pada sistem penguncian pintu gedung ini terdapat fitur yang memungkinkan operator untuk mengatur pintu terbuka dan terkunci secara otomatis pada rentang waktu tertentu. Struktur dari tabel jadwal dapat dilihat pada tabel x.x berikut :

Tabel Jadwal

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | id | char(36) | kunci primer tabel |
| 2 | office\_id | char(36) | referensi gedung |
| 3 | name | varchar(255) | nama jadwal |
| 4 | date\_begin | date | tanggal mulai |
| 5 | date\_end | date | tanggal berakhir |
| 6 | time\_begin | time | waktu mulai |
| 7 | time\_end | time | waktu berakhir |
| 8 | is\_repeating | tinyint(1) | status perulangan |
| 9 | day\_repeating | varchar(255) | perulangan hari |
| 10 | status | enum | status pelaksanaan jadwal |
| 11 | created\_at | timestamp | waktu data dibuat |
| 12 | updated\_at | timestamp | waktu data diedit |

1. Tabel Jadwal Pintu

Tabel jadwal pintu (door\_schedules) digunakan untuk menyimpan pintu-pintu yang akan dibuka/dikunci menggunakan jadwal yang telah dibuat. Setiap jadwal akan berisi beberapa pintu yang akan dibuka/dikunci sehingga pintu-pintu tersebut dapat dibuka/dikunci secara bersamaan menggunakan jadwal yang sudah dibuat. Struktur dari tabel jadwal pintu dapat dilihat pada tabel x.x berikut :

Tabel Jadwal Pintu

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | id | char(36) | kunci primer tabel |
| 2 | schedule\_id | char(36) | referensi ke jadwal |
| 3 | door\_id | char(36) | referensi ke pintu |
| 4 | created\_at | timestamp | waktu data dibuat |
| 5 | updated\_at | timestamp | waktu data diedit |

1. Tabel Akses

Tabel akses (access) digunakan untuk menyimpan data kartu akses pengguna ke masing-masing. Dengan menggunakan kartu akses yang tersimpan didalam database maka pengaturan akses pengguna dapat dilakukan dengan efisien dan realtime. Struktur dari tabel akses dapat dilihat pada tabel x.x berikut :

Tabel Akses

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | id | char(36) | kunci primer tabel |
| 2 | user\_id | char(36) | referensi ke pengguna |
| 3 | door\_id | char(36) | referensi ke pintu |
| 4 | time\_begin | time | waktu mulai |
| 5 | time\_end | time | waktu berakhir |
| 6 | date\_begin | date | tanggal mulai |
| 7 | date\_end | date | tanggal berakhir |
| 8 | is\_temporary | tinyint(1) | status akses sementara |
| 9 | is\_running | tinyint(1) | status akses sedang berjalan |
| 10 | created\_at | timestamp | waktu data dibuat |
| 11 | updated\_at | timestamp | waktu data diedit |

1. Tabel Riwayat Akses

Tabel riwayat akses (access\_logs) digunakan untuk menyimpan riwayat akses pengguna ke pintu. Dengan menggunakan riwayat akses pada aktifitas pengguna akan tercatat didalam database dan dapat dilihat oleh masing-masing operator gedung. Struktur dari tabel riwayat akses dapat dilihat pada tabel x.x berikut :

Tabel Riwayat Akses

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | id | char(36) | kunci primer tabel |
| 2 | user\_id | char(36) | referensi ke pengguna |
| 3 | office\_id | char(36) | referensi ke gedung |
| 4 | door\_id | char(36) | referensi ke pintu |
| 5 | log | text | pesan log |
| 6 | created\_at | timestamp | waktu data dibuat |
| 7 | updated\_at | timestamp | waktu data diedit |

1. Tabel OTP

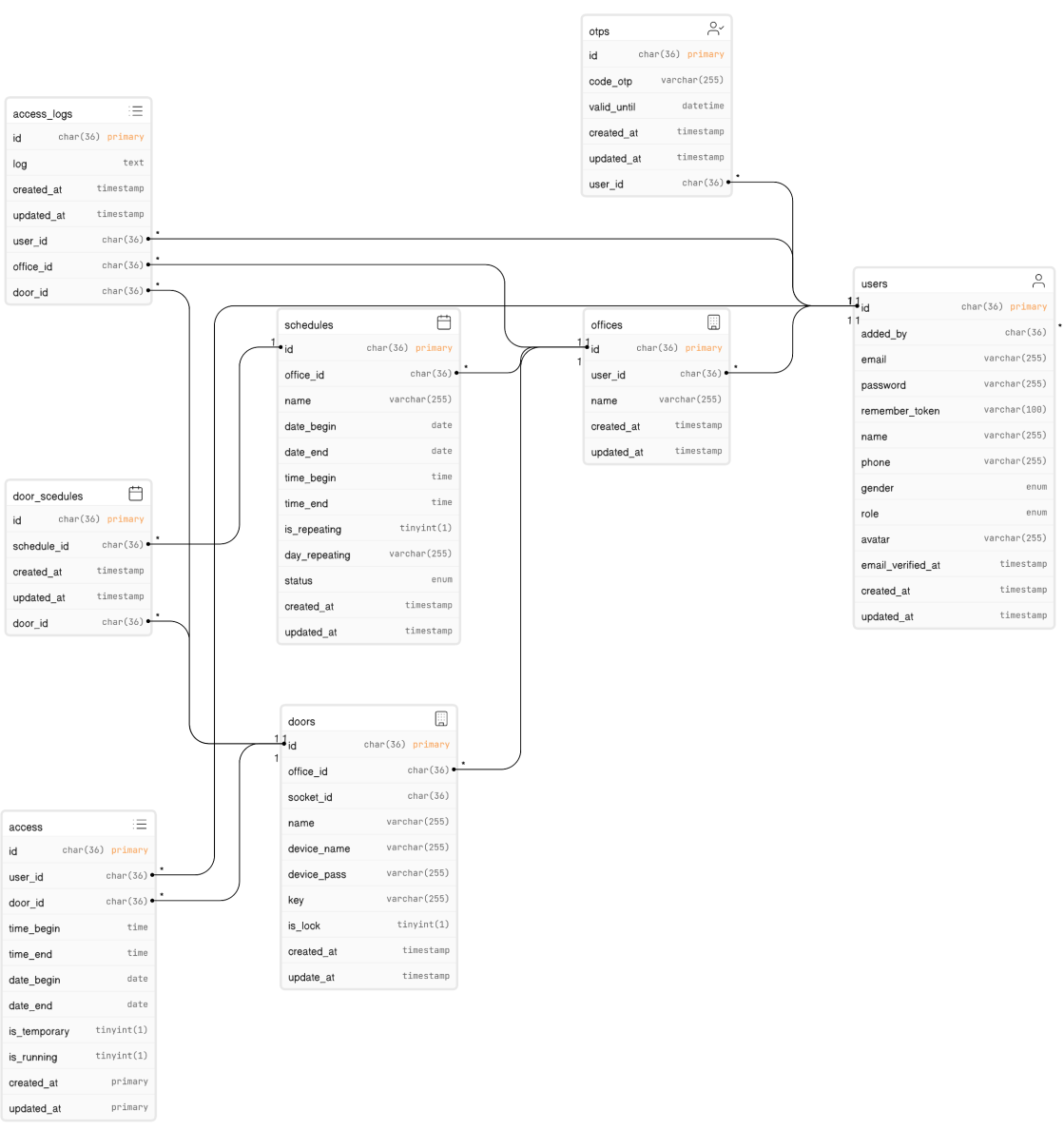
Tabel OTP (otps) atau One Time Password digunakan untuk menyimpan data verifikasi kode OTP. Dengan menggunakan verifikasi OTP maka hanya pengguna dengan email aktif yang akan diterima sehingga meningkatkan keamanan dan keabsahan data pengguna. Struktur dari tabel OTP dapat dilihat pada tabel x.x berikut :

Tabel OTP

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1 | id | char(36) | kunci primer tabel |
| 2 | user\_id | char(36) | referensi ke pengguna |
| 3 | code\_opt | varchar(255) | kode OTP |
| 4 | valid until | datetime | hari dan tanggal kadaluarsa |
| 5 | created\_at | timestamp | waktu data dibuat |
| 6 | updated\_at | timestamp | waktu data diedit |

1. Relasi Antar Tabel

Dengan menggunakan relational database tentunya akan terdapat hubungan atau relasi antara dua tabel atau lebih, relasi memberikan informasi mengenai hubungan antara dua tabel atau lebih yang saling berkaitan beserta dengan perilaku terhadap perubahan data pada tabel induknya.



Relasi Antar Tabel

Dalam skema database yang dirancang pada sistem penguncian pintu gedung ini terdapat 12 relasi antar tabel yaitu sesuai dengan tabel x.x berikut :

Tabel relasi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tabel Induk** | **Anak** | | **Jenis Hubungan** | **On Update** | **On Delete** |
| **Tabel** | **Kolom** |
| 1 | users | users | added\_by | one to many | cascade | restrict |
| 2 | users | access | user\_id | one to many | cascade | cascade |
| 3 | doors | access | door\_id | one to many | cascade | cascade |
| 4 | users | access\_logs | user\_id | one to many | cascade | cascade |
| 5 | doors | access\_logs | door\_id | one to many | cascade | cascade |
| 6 | offices | access\_logs | office\_id | one to many | cascade | cascade |
| 7 | offices | doors | office\_id | one to many | cascade | cascade |
| 8 | schedules | door\_schedule | schedule\_id | one to many | cascade | cascade |
| 9 | doors | door\_schedule | door\_id | one to many | cascade | cascade |
| 10 | users | offices | user\_id | one to one | cascade | restrict |
| 11 | users | otps | user\_id | one to one | cascade | cascade |
| 12 | offices | schedules | office\_id | one to many | cascade | cascade |

Dengan menggunakan relasi seperti yang tertulis pada tabel x.x diatas dapat memberikan struktur data yang sesuai dengan menghilangkan data ganda atau perulangan, sebagai contoh pada beberapa tabel yang membutuhkan data pengguna seperti tabel access, opts, offices dan access\_logs data pengguna tidak dicantumkan didalam tabel tersebut melainkan berada pada tabel users yang menjadi induk dari tabel-tabel tersebut.

Dengan menggunakan relasi seperti yang tertulis pada tabel x.x diatas juga dapat memberikan konsistensi data, seperti contoh pada tabel offices yang berelasi dengan tabel users dimana pada perilaku on delete restrict maka penghapusan pada data pengguna yang berkaitan dengan tabel offices akan dicegah sehingga menjamin bahwa setiap gedung memiliki seorang operator.

1. Perancangan Backend API

Backend API digunakan sebagai salah satu jalur komunikasi yang menghubungkan antara perangkat mobile dan perangkat kunci pintu berbasis IoT dengan server. Selain terdapat website yang digunakna 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.

Backend API yang dikembangkan menggunakan bahasa pemrograman PHP dengan menggunakan kerangka kerja dari Laravel. Laravel menyediakan beberapa fitur yang siap digunakan seperti autentikasi login, reset password, notifikasi email dan lain sebagaimnya sehingga mempermudah dan mempercepat proses pengkodean.

1. Diagram Use Case API

Tidak semua fitur yang ada pada website diimplementasikan dalam bentuk API, hal ini berkaitan dengan kompleksitas dari sistem yang dikembangkan, fitur yang berkaitan dengan manajemen penguncian seperti menambah pintu baru, membuat jadwal, menambahakan pengguna baru, memberikan akses ke pengguna hanya bisa dilakukan melalui website hal ini dimaksudkan untuk mempertahankan kontrol yang lebih ketat terhadap akses dan perubahan yang dilakukan pada sistem. Dengan membatasi akses langsung melalui API, administrator sistem dapat memastikan bahwa hanya proses yang telah melalui lapisan keamanan dan validasi yang benar yang dapat mempengaruhi manajemen penguncian dan pengaturan lainnya. Diagram Use Case dari Backend API dapat dilihat pada gambar x.x dibawah.

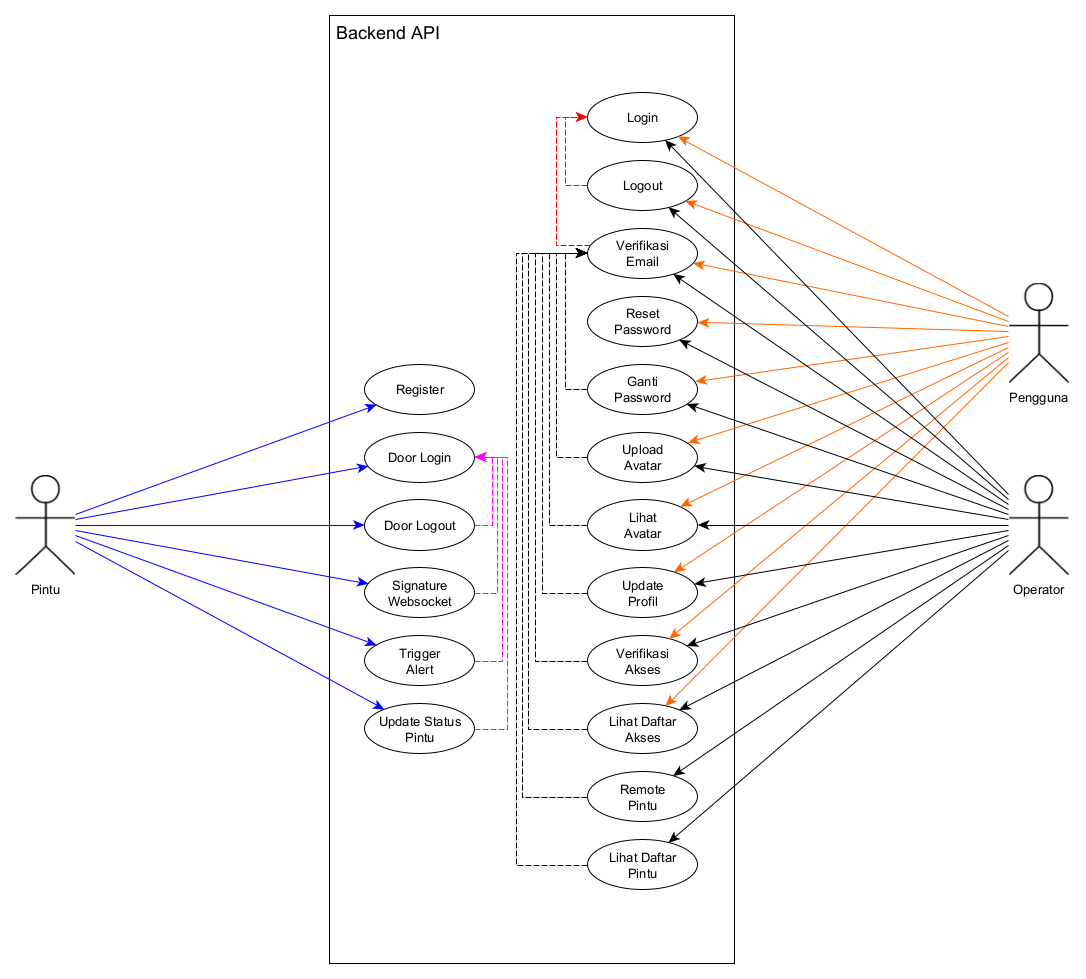


Diagram Use Case API

Dapat dilihat pada gambar x.x diatas, terdapat 3 aktor yang berinteraksi dengan sistem melalui API yaitu pintu, pengguna dan operator. Pintu merupakan perangkat IoT yang digunakan untuk melakukan penguncian pada gedung, sedangkan pengguna dan operator merupakan aplikasi mobile yang digunakan sebagai antarmuka sistem.

Pada gambar x.x diatas juga terlihat beberapa fungsi seperti update profil, lihat avatar, verifikasi akses dan lain sebagainya merupakan extends dari fungsi verifikasi email yang menandakan bahawa fungsi-fungsi tersebut hanya bisa diakses jike client sudah melakukan proses verifikasi email. Fungsi verifikasi email juga merupakan extends dari fungsi login sehingga fungsi verifikasi email hanya bisa diakses jika client sudah mellakukan autentikasi login. Hal ini dimaksudkan supaya hanya client yang terautentikasi dan terverifikasi yang dapat mengakses sumber daya yang disediakan.

1. Endpoint

Dalam perancangan Backend API ini terdapat dua grup API yang digunakan yaitu API untuk aplikasi mobile dan API untuk perangkat kunci pintu IoT dengan detail endpoint dan fungsinya dapat dilihat pada tabel x.x diabawah.

Tabel x.x Endpoint API

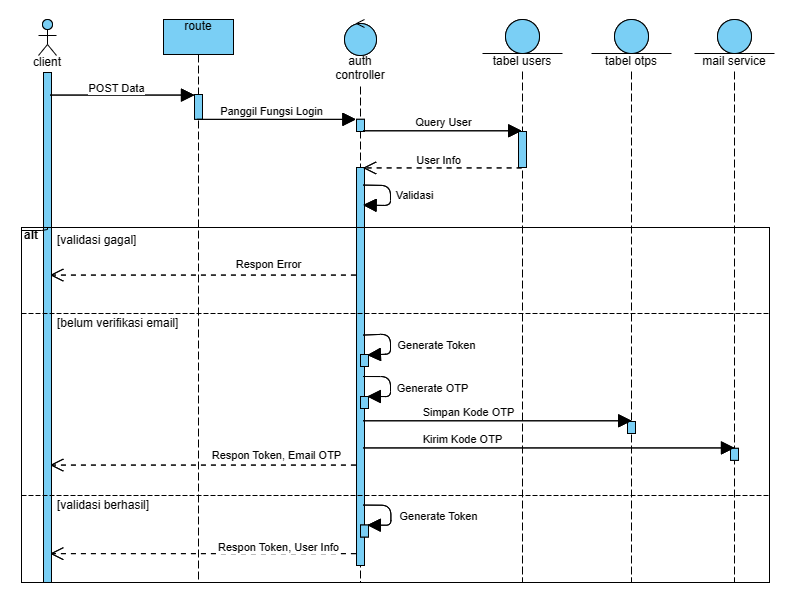
|  |  |  |  |
| --- | --- | --- | --- |
| **Tipe** | **Endpoint** | **Auth** | **Keterangan** |
| POST | /api/login | - | login pengguna dan operator |
| POST | /api/reset-password | - | reset password menggunakan email |
| POST | /api/verify-email | sanctum | verifikasi email |
| GET | /api/logout | 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 x.x diatas, API terbagi menjadi 2 bagian yang ditandai dengan prefix “/api” dan “/door”. Prefix “/api” merupakan endpoint API yang khusus ditujukan untuk penggunaan aplikasi mobile pengguna dan operator sedangkan prefix “/door” merupakan API yang ditujukan untuk perangkat kunci IoT sehingga dengan menggunakan API yang terpisah maka API akan semakin terorganisir dengan baik.

Pada tabel x.x juga terlihat pahwa beberapa endpoint memerlukan autentikasi sanctum dan verified. Autentikasi sanctum merupakan sebuah metode autentikasi berbasis token yang digunakan untuk mengamankan sumberdaya 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. API Login

API login digunakan untuk melakukan autentikasi client melalui username dan password yang dikirimkan, API login juga mengecek apakah pengguna dan operator sudah melakukan verifiaksi email, jika belum maka login akan tertahan sampai pengguna melalukan verifikasi email. Diagram fungsi login dapat dilihat pada gambar x.x dibawah.

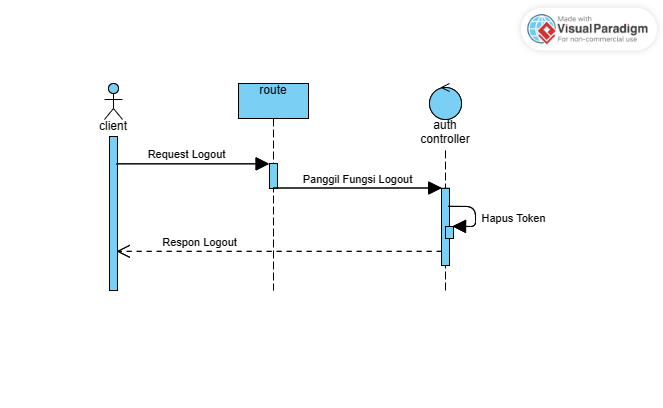


Sequence Diagram Login

API login dimulai oleh client, client akan mengirimkan username dan password mereka melalui endpoint “/api/login” kemudian route akan memanggil fungsi login didalam kontroler, kontroler akan memeriksa username dan password dengan melakukan query ke tabel users, jika cocok maka kontroler akan membuat token menggunakan modul sanctum. Sanctum adalah sebuah paket autentikasi dan autorisasi yang disediakan oleh Laravel yang dirancang untuk memudahkan implementasi otentikasi API yang sederhana namun aman pada aplikasi Laravel. Setelah mendapatkan token kemudian kontroler akan mengembalikan token tersebut disertai dengan detail client seperti nama, email, nomor hp dan lain sebagainya. Jika terdeteksi client belum malakukan verifikasi email maka kontroler akan membuat kode OTP atau One Time Password yang merupakan 6 digit angka acak dan mengirimakan kode terkesebut ke email client. Jika autentikasi yang dilakukan gagal, maka kontroler akan mengembalikan respon error ke client.

1. API Logout

Fungsi logout digunakan untuk mengakhiri sesi login pengguna dan operator dengan cara menghapus semua token yang dimilikinya. Diagram dari API logout dapat dilihat pada gambar x.x dibawah.

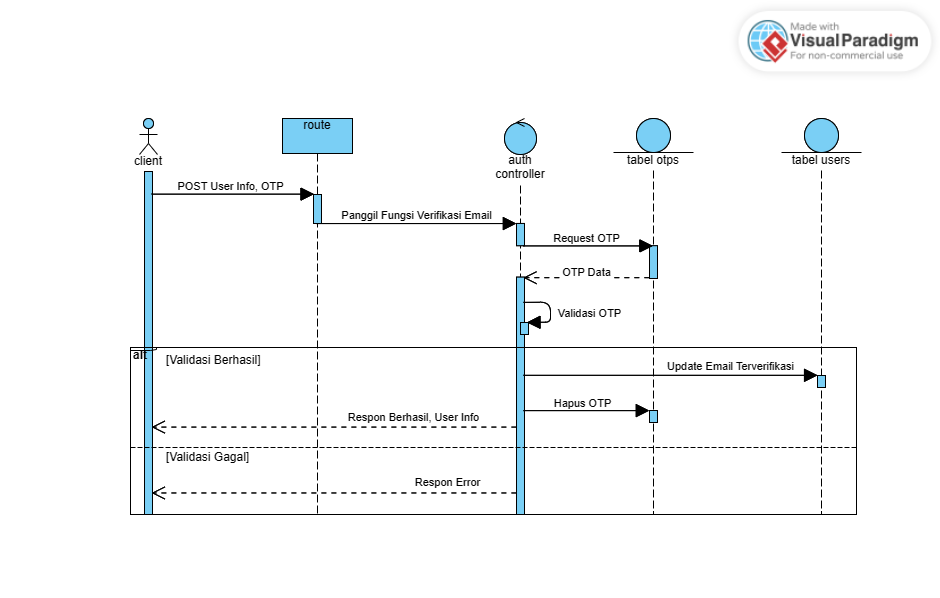


Sequence Diagram Logout

Pada gambar x.x diatas, sebuah metode logout didalam kontroler akan dipanggil oleh route jika ada client yang melakukan request ke endpoint “/api/logout”. Kemudian kontroler akan menghapus token dari client sesuai dengan token yang dilampirkan didalam header pada saat request diterima.

1. API Verifikasi Email

API verifikasi email digunakan untuk memastikan bahwa pengguna dan operator memiliki email yang valid dan aktif hal ini akan meningkatkan keamanan dengan hanya mengijinkan pengguna dan operator yang terpercaya untuk mengakses sumber daya yang ada. Diagram dari API verifikasi email dapat dilihat pada gambar x.x dibawah.

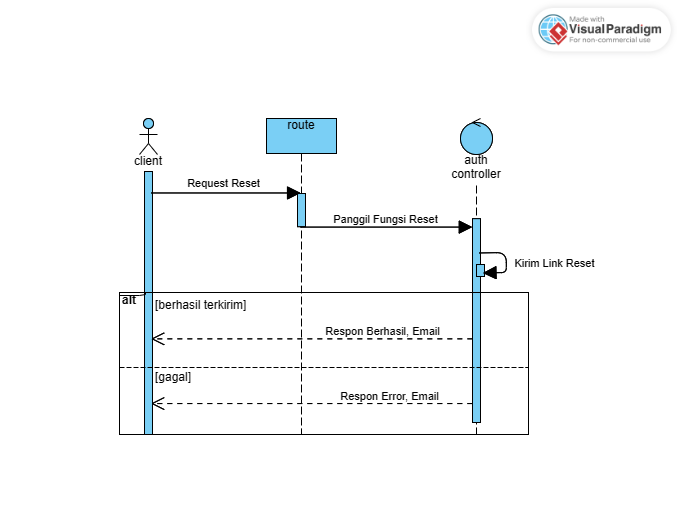


Sequence Diagram Verifikasi Email

Pada gambar x.x diatas dapat dilihat bahwa pengguna melakukan verifikasi email dengan mengirimkan kode OTP yang sudah diterima melalui email disertai dengan detail client seperti id, nama dan email ke endpoint “/api/verify-email”, kemudian pada kontroler akan memeriksa kode yang diterima dengan kode yang tersimpan pada tabel otps, jika cocok dan masih aktif maka kontroler akan memperbarui status client menjadi terverifikasi dan menghapus kode otp yang lama kemudian mengembalikan respon berhasil. Jika kode salah atau sudah kadaluarsa maka kontroller akan mengembalikan respon error.

1. API Reset Password

API reset password digunakan oleh pengguna dan operator untuk memperbarui password jika meraka lupa dengan cara memasukkan email yang sudah terdaftar dan menggunakan link yang telah dikirimkan ke email untuk memperbarui password. Diagram dari API reset password dapat dilihat pada gambar x.x dibawah.



Sequence Diagram Reset Password

Pada gamabar x.x diatas dapat dilihat client memulai reset password dengan melakukan request ke endpoint “/api/reset-password” dengan mengirimkan email yang sudah terdaftar, kemudian kontroler akan memanggil sebuah fungsi kirim link reset yang sudah disediakan oleh Laravel dengan melampirkan email yang telah diterima, jika email berhasil terkirim maka kontroler akan mengembalikan respon berhasil dan jika email gagal terkirim atau email tidak ditemukan maka kontroler akan mengembarikan respon error.

1. API Ganti Password

API ganti password digunakan oleh pengguna dan operator untuk mengganti password mereka melalui aplikasi mobile. Berbeda dengan reset password, API ganti password digunakan untuk mengganti password pengguna tanpa melalui link yang dikirimkan ke email. Diagram dari API ganti password dalap dilihat pada gambar x.x dibawah.

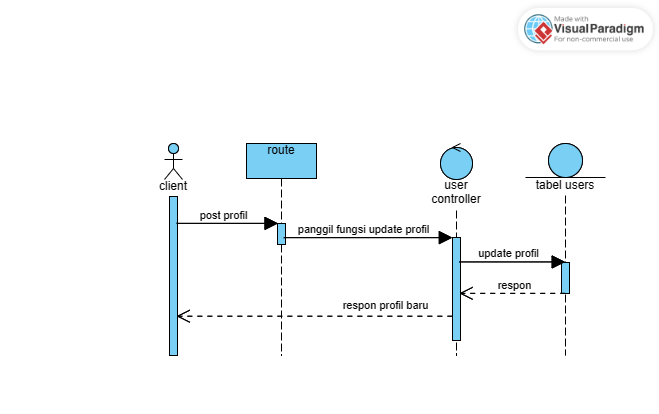


Sequence Diagram Ganti Password

Dapat dilihat pada gambar x.x diatas, untuk mengganti password pertama client mengirimkan permintaan ganti password ke enpoint “/api/change-password” dengan mengirimkan password lama, password baru dan konfirmasi password baru, kemudian didalam kontroler password lama yang dikirimkan akan dicocokkan dengan password client sekarang dengan menggunakan fungsi Hash. Fungsi Hash merupakan sebuah fungsi yang disediakan oleh laravel yang digunakan untuk pengolahan data yang berkaitan dengan enkripsi. Jika kedua password cocok maka kontroler akan memperbarui password pada tabel users dan mengmbalikan respon berhasil, jika password tidak sesuai maka kontroler akan mengembalikan respon error.

1. API Ganti Profil

API ganti profil digunakan untuk mengganti detail pengguna dan operator seperti nama, email, nomor hp dan lain sebagainya melalui aplikasi mobile. Diagram dari API ganti profil dapat dilihat pada gamabr x.x dibawah.

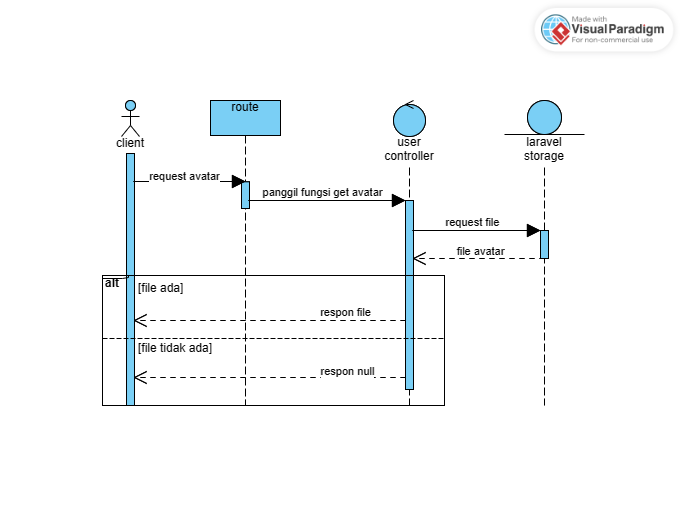


Sequence Diagram Ganti Profil

Dapat dilihat pada gambar x.x diatas, unttuk mengganti profil pertama penguna atau operator mengirimkan profil mereka ke enpoint “/api/update-profile” kemudian didalam kontroler data yang telah terima kan dimasukkan kedalam tabel users untuk memperbarui profil dan terakhir kontroler mengembalikan respon bahwa profil berhasil dirubah.

1. API Lihat Avatar

API lihat avatar digunakan untuk mendapatkan gambar avatar (foto profil) dari pengguna dan operator untuk ditampilkan didalam aplikasi mobile. Diagram dari API lihat avatar dapat dilihat pda gambar x.x dibawah.

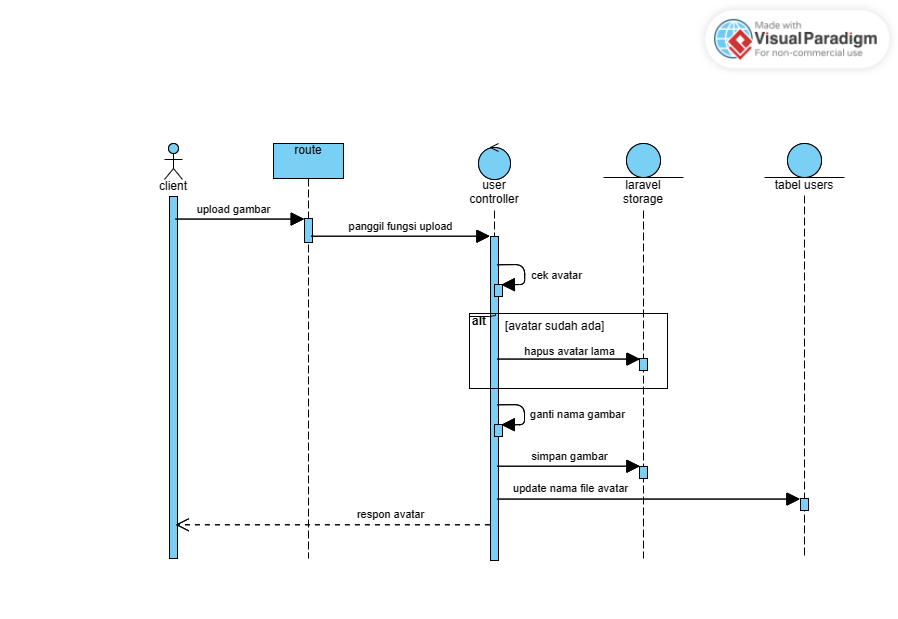


Sequence Diagram Lihat Avatar

Dapat dilihat pada gambar x.x diatas, untuk mendapatkan gambar avatar pertama pengguna atau operator melakukan request avatar melalui endpoint “/api/avatar” kemudian didalam kontroler akan mengambil gambar avatar didalam storage. Storage merupakan salah satu fitur yang disediakan laravel untuk melakukan peyimpana file seperti gambar, log dan berkas lainnya, dengan menggunakan storage kita bisa mengatur sipa saja yang boleh mengakses penyimpanan tersebut sehingga lebih aman. Jika file telah ditemukan maka kontroler akan mengembalikan respon file dan jika tidak ditemukan maka kontroler akan mengembalikan nilai null.

1. API Ganti Avatar

API ganti avatar digunakan untuk mengganti gambar avatar pengguna atau operator, pengguna dapat mengganti gambar avatar melalui aplikasi mobile. Diagarm dari API ganti avatar dapat dilihat pada gambar x.x dibawah.

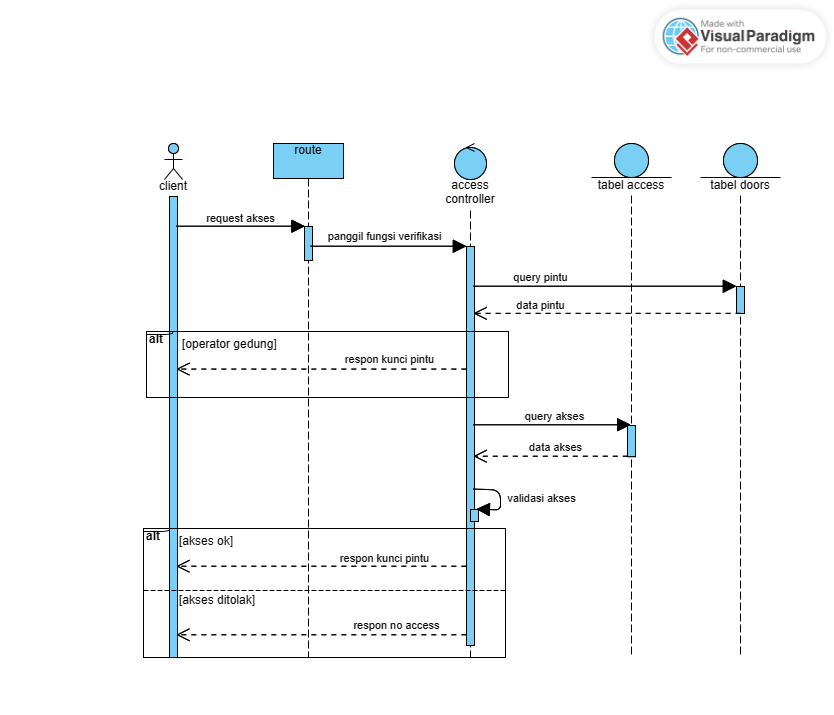


Sequence Diagram Ganti Avatar

Dapat dilihat pada gamabr x.x diatas, untuk mengganti avatar pertama pengguna atau operatoa akan mengirimakan file avatar yang baru melalui endpoint “/api/update-avatar”, kemudian didalam kontroler akan diperiksa apakah sebelumnya pengguna atau operator sudah memiliki avatar jika sudah maka file avatar sebelumnya akan dihapus dari storage, kemudian file avatar yang baru akan disesuaikan namanya sesuai dengan format server, setelah ganti nama kemudian file akan disimpan didalam storage dan kontroler juga akan memperbarui nama avatar didalam tabel users, terakhir kontorler akan mengembalikan respon yang menginformasikan avatar telah diganti.

1. API Verifikasi Akses

Verifikasi akses digunakan oleh pengguna dan operator untuk memverifikasi diri mereka dan untuk mendapatkan akses terhadap suatu pintu dengan cara memindai kode QR dengan perangkat mobile. Diagram dari API verifikasi akses dapat dilihat pada gambar x.x dibawah.

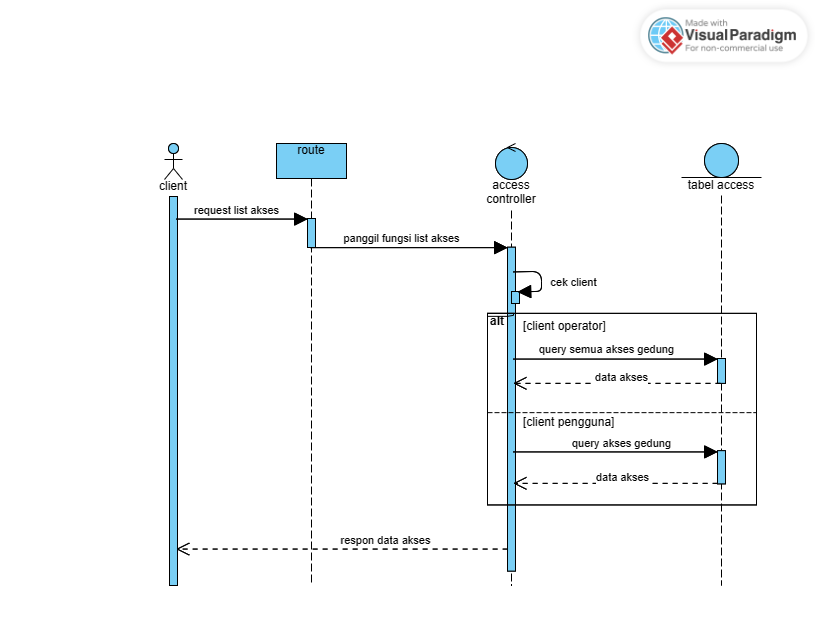


Sequence Diagram Verifikasi Akses

Untuk mendapatkan akases ke pintu, setelah memindai kode QR pada pintu maka pengguna atau operator akan mendapatkan data informasi terkait pintu tersebut, kemudian data tersebut dikirimkan ke serve melalui endpoint “/api/verify-access/{door-id}”, kemudian kontroler akan mengambil data pintu pada tabel doors, jika permintaan berapsal dari operator gedung dimana pintu tersebut berada maka kontroler akan mengijinkan dengan mengambalikan respon berupa kode kunci pintu. Jika pengguna biasa maka kontroler akan memeriksa daftar akses didalam tabel access, jika pengguna memiliki akses dan masil berlaku maka kontroler akan mengembalikan respon diijinkan dan mengirimkan kode kunci untuk membuak pintu.

1. API Daftar Akses

API daftar akses digunakan untuk mendapatkan daftar dari pintu mana saja yang dapat diakses oleh pengguna, sehingga pengguna dapat mnegetahui pintu mana saja yang bisa diakses oleh dirinya. Diagram dari API dafar akses dapat dilihat pada gambar x.x dibawah.

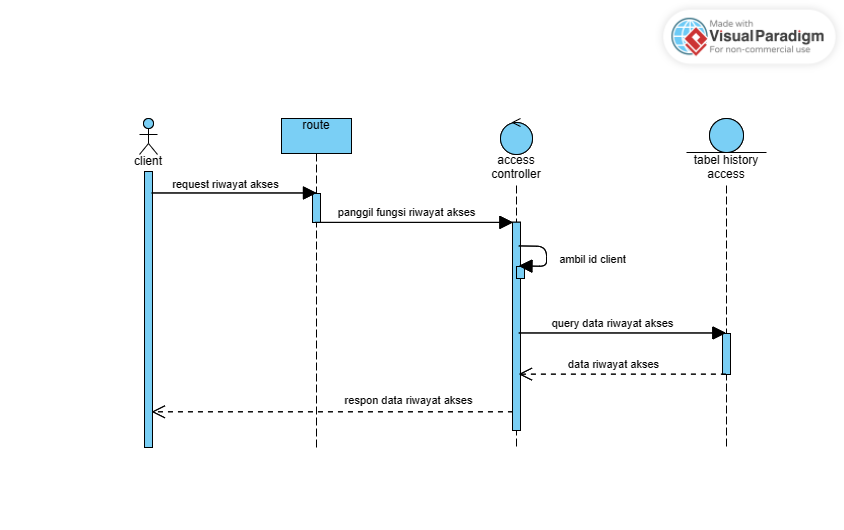


Sequence Diagram Daftar Akses

Dapat dilihat pada gambar x.x diatas, untuk mendapatkan daftar akses yang dimiliki pengguna mengirimkan request endpoint “/api/my-access”, kemudian didalam kontroler akan diperiksa apakah permintaan datang dari seorang operator atau pengguna biasa. Jika permintaan berasal dari seorang operator maka kontroler akan mengembalikan semua akses pintu atau semua daftar pintu yang ada namun jika permintaan datang dari pengguna biasa maka kontroler akan mengembalikan data akses sesuai dengan pengguna yang meminta.

1. API Riwayat Akses

API riwayat akses digunakan untuk memberikan informasi pencatatan akses pengguna, riwayat akses pengguna dapat dilihat pada masing-masing aplikasi mobile pengguna. Diagram dari API riwayat akses dapat dilihat pada gambar x.x dibawah.

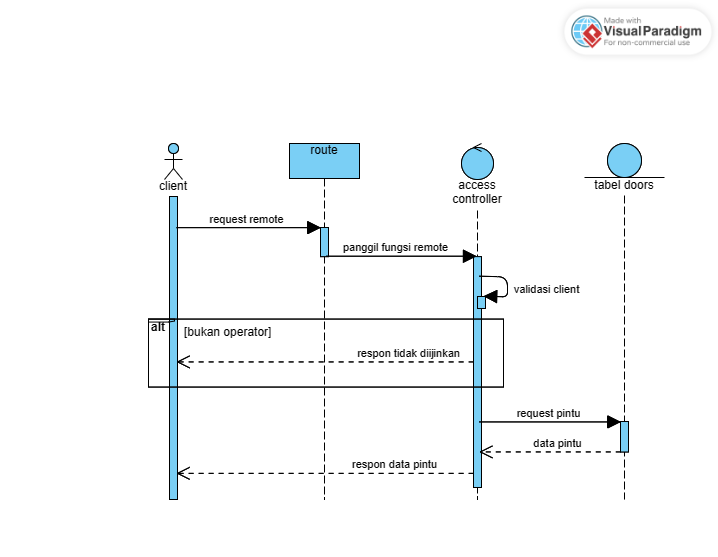


Sequence Diagram Riwayat Akses

Dapat dilihat pada gambar x.x untuk mendapatkan data riwayat akses maka pengguna akan melakukan request ke enpoint “/api/my-history”. Kemudian kontroller akan mengambil data riwayat akses pengguna dengan melakukan query ke databse dengan menyertakan identitas pengguna sebagai parameter. Selanjutnya data yang sudah diperoleh akan dikirimkan kembali ke pengguna sebagai respon untuk diolah pada tampilan aplikasi.

1. API Daftar Pintu

API daftar pintu digunakan oleh operator untuk menampilkah daftar pintu yang ada pada satu gedung, sehingga operator dapat mengetahui jumlah dan status pada setiap pintu. Diagram dari API daftar pintu dapat dilihat pada gambar x.x dibawah.

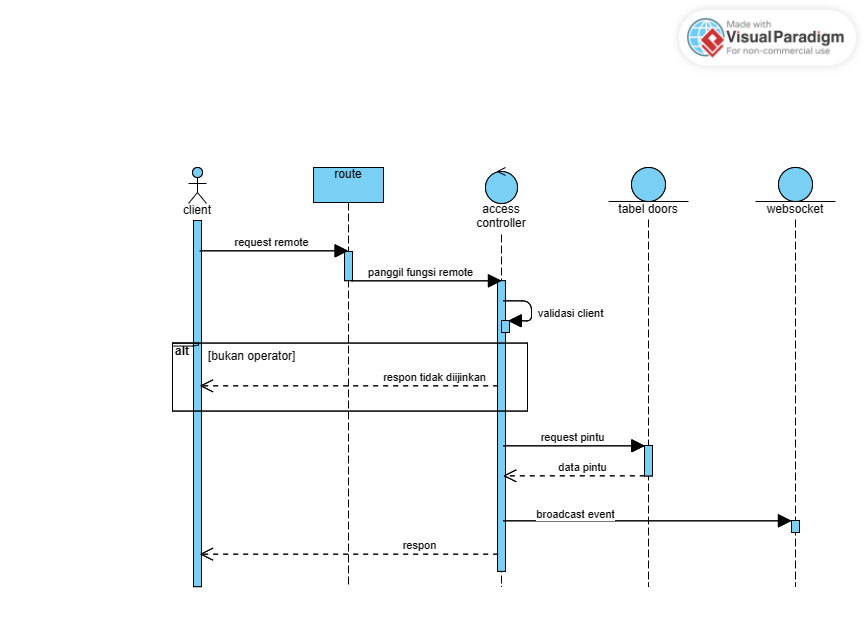


Sequence Diagram Daftar Pintu

Dapat dilihat pada gambar x.x diatas, untuk mendapatkan daftar pintu maka pertama operator akan melalakukan request ke endpoint “/api/get-door” kemudian didalam kontroler dilakukan pengecekan untuk memastikan permintaan hanya berasal dari operator, jika permintaan berasal dari pengguna biasa maka kontroler akan mengembalikan respon tidak diijinkan. Selanjutnya kontroler akan mengambil data pintu pada tabel doors dan mengirimkan rebagai respon ke operator.

1. API Remote Pintu

API remote pintu digunakan oleh operator untuk membuka atau mengunci pintu secara jarak jauh melalui aplikasi mobile. Dengan adanya fitur ini operator dapat mengendalikan pintu melaui aplikasi mobile dimana saja dan kapan saja tanpa harus berapa di ruangan operasional dengan menggunaka komputer. Diagram dari API remote pintu dapat dilihat pada gambar x.x dibawah.

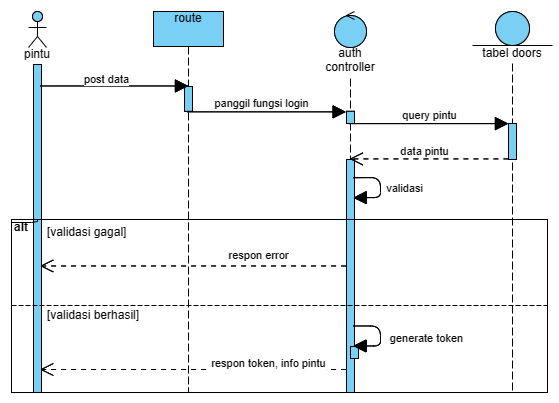


Sequence Diagram Remote Pintu

Dapat dilihat pada gamabar x.x diatas, untuk melakukan remote pintu pertama operator akan melakukan request ke endpoint “/api/remote-access” dengan mengirimkan id dari pintu yang akan dikendalikan, kemudian kontroler akan memeriksa client untuk memastika permintaan berasal dari operator, jika barasal dari pengguna biasa maka kontroler akan mengeembalikan respon tidak diijinkan. Selanjutnya kontroler akan mengambil data pintu untuk melengkapi informasi yang dibutuhkan seperti kode kunci, kode gedung, token dan lain sebagainya, kemudian data akan disiarkan ke perangkat kunci pintu melalui koneksi websocket yang sudah terhubung, terakhir kontroler akan mengembalikan respon remote pintu telah dilaksanakan.

1. API Door Login

API door login digunakan untuk proses autentikasi perangkat kunci pintu, berbeda dengan login pada pengguna dan operator pada login pintu hanya terdapat pengecekan username dan password saja tanpa ada pengeceka verifikasi email dan lainnya hal ini dikarenakan pada perangkat kunci pintu tidak menggunakan email sebagai identitasnya. Diagram dari API door login dapat dilihat pada gambar x.x dibawah.

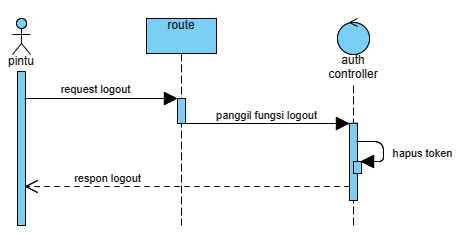


Sequence Diagram Door Login

Dapat dilihat pada gambar x.x diatas, perangkat kunci pintu akan mengirimkan data username dan password melalui endpoint “/door/login”, kemudian didalam kontroler akan memeriksa dan membandingkan dengan data pada tabel doors, jika sesuai maka kontroler akan membuat token baru menggunakan sanctum dan mengembalikan respon token yang menandakan login berhasil, nama jika data tidak sesuai maka kontroler akan mengembalikan respon error.

1. API Door Logout

untuk mengakhiri sesi login perangkat kunci pintu dengan cara menghapus semua token yang dimilikinya. Diagram dari API door logout dapat dilihat pada gambar x.x dibawah.

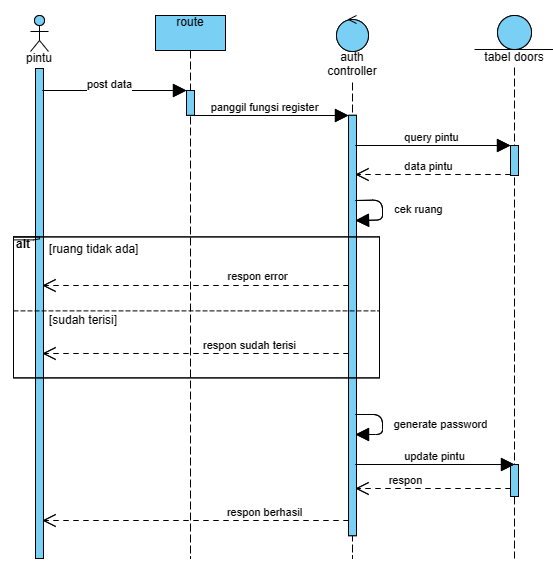


Sequence Diagram Logout

Pada gambar x.x diatas, sebuah metode logout didalam kontroler akan dipanggil oleh route jika ada perangkat kunci pintu yang melakukan request ke endpoint “/door/logout”. Kemudian kontroler akan menghapus token dari perangkat kunci pintu sesuai dengan token yang dilampirkan didalam header pada saat request diterima. Terakhir kontroler akan mengembalikan respon logout.

1. API Door Register

API door register digunakan untuk menambahkan perangkat penguncian yang baru kedalam pintu. Pada saat operator menambahakan pintu baru melalui dashboard websote operator maka pintu tersebut belum terpasang perangkat kunci pintu sehingga harus ditambahkan secara manual melalui prosedur register. Diagram dari API door register dapat dilihat pada gambar x.x dibawah.

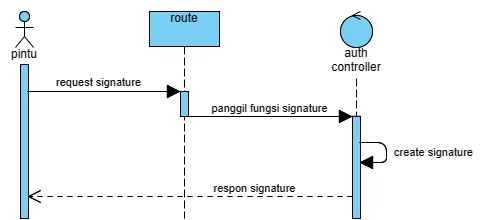


Sequence Diagram Door Register

Dapat dilihat pada gambar x.x diatas, untuk menambahakan perangkat penguncian baru maka perangkat kunci pintu akan mengirimkan data nama perangkat yang akan didaftarkan dan kode pintu yang akan ditempati melalui endpoint “/door/register” kemudian kontroler akan mengambil data pada tabel doors untuk memastika bahwa pintu masih kosong, jika pintu sudah ada perangkat pengunciannya maka kontroler akan mengembalikan respon sudah terisi dan kontroler akan mengembalikan respon error jika pintu yang dituju tidak ditemukan. Selanjutnya kontroler akan membuat password random yang akan digunaka untuk autentikasi, dengan menggunakan metode ini maka akan meningkatkan keamanana karena password bersiftar acak dan hanya diketahui oleh perangkat kunci pintu dan server. Terakhir kontroler akan mengembalikan respon berhasil direstai dengan password tadi untuk disimpan pada perangkat kunci pintu.

1. API Door Signature

API door signature digunakan untuk mendapatkan kode unik yang digunakn untuk melakukan subcribe ke channer pusher. Pusher merupakan sebuah protokol komunikasi notifikasi yang dibangun menggunakan websocket, didalam pusher terdapat channel-channel yang dapat disubcribe untuk nantinya client dapat menunggun event yang disiarkan melalui broadcast pada setiap channel. Diagram dari API door signature dapat dilihat pada gambar x.x dibawah.



Sequence Diagram Door Signature

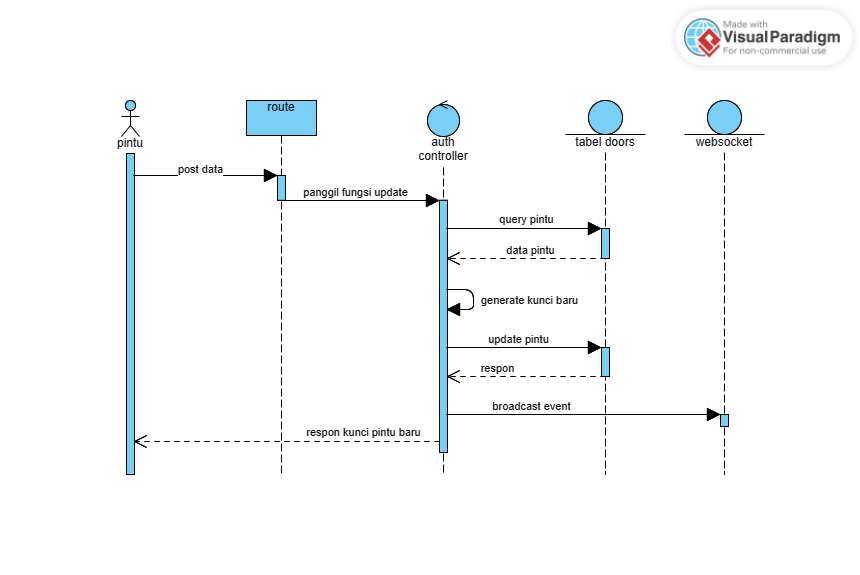
Dapat dilihat pada gamabr x.x diatas, untuk mendapatkan kode signature pusher pertama perangkat kunci pintu melakukan request ke endpoint “/door/get-signature” dengan mengirimkan data-data seperti socket-id, office-id dan channel-data, dari data tersebut kemudian kontroler akan membuat kode signature menggunakan metode yang ada pada protokol pusher yaitu :

sign = hash\_hmac('sha256','socket:channel:data','secret\_key')

setelah mendapatkan nilai signature kemudian kontroler akan megembalikan respon kode signature ke perangkat kunci pintu.

1. API Door Update Status

API door update status digunakn oleh perangkat kunci pintu untuk memperbarui status pintu seperti pintu terbuka, pintu terkunci atau pintu terkoneksi. Pada setiap proses updata ini kode kunci pintu akan diperbarui sehingga meningkatkan keamanan karena kode kunci selalu berubah. Diagram dari API door update status dapat dilihat pada gambar x.x dibawah.

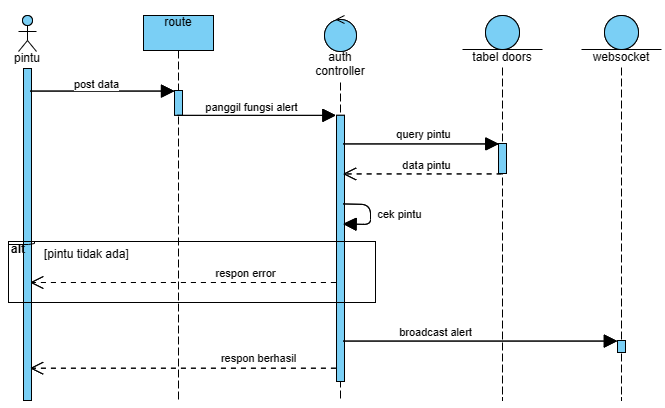


Sequence Diagram Door Update Status

Terlihat pada gambar x.x diatas, untuk melakukan update status pertama perangkat kunci pintu akan mengirimkan data-data seperti status penguncian dan id socket melalui endpoint “/door/update-status”, kemudian kontroler akan mengambil data pintu yang terkait untuk diperbarui menggunaka data status dan kode kunci yang baru, kemudian kontroler juga akan melakukan broadcarst untuk menyiarkan bahwa status pintu berubah sehingga setiap informasi perubahan dapat tersampaikan secara langsung, terakhir kontroler akan mengembalikan respon update status telah dilaksanakan.

1. API Door Alert

API door alert digunakan oleh perangkat kunci pintu untuk memberikan perintatan kepada operator bahwa pintu dalam kondisi yang tidak aman seperti terbuka tanpa autentikasi yang sah. Diagram dari API door alert dapat dilihat pada gambar x.x dibawah.



Sequence Diagram Door Alert

Dapat dilihat pada gambar xx diatas, untuk memberikan peringatan pertama perangkat kunci pintu akan mengirimakan data-data seperti id-pintu, id-office dan status peringatan melalui endpoint “/door/alert” kemudian kontroler akan memeriksa pada tabel pintu untuk memastika bahwa pintu valid, jika pintu tudak ditemuakan maka kontroler akan mengembalikan respon error. Selanjutnya kontroler akan menyiarkan peringatan melalui websocket, terakhir kontroler akan mengembalikan respon peringatan sudah dilaksanakan.

1. Perancangan Penjadwalan

Penjadwalan digunakan untuk memeriksa data jadwal yang ada pada tabel schedules. Penjadwalan bekerja dengan menggunakan kernel yang disediakan oleh laravel, kernel ini nantinya akan dijalankan oleh laravel dengan menggunakan crontab yang diatur pada sistem operasi server. Crontab akan diatur untuk menjalankan perintah “schedule:run” setiap 1 menit sekali yang akan menjalankan fungsi yang ada pada kernel.

1. Pengecekan Jadwal

Pengecekan jadwal dilakukan dengan cara memeriksa data jadwal pada tabel schedules setiap satu menit sekali. Kernel akan menjalankan job setiap satu menit sekali untuk memeriksa jadwal, jika ada jadwal yang harus dilaksanakan seperti membuka pintu atau mengunci pintu maka perintah akan dikirimkan ke perangkat kunci pintu melalui websocket. Diagram dari pengecekan jadwal dapat dilihat pada gambar x.x dibawah.

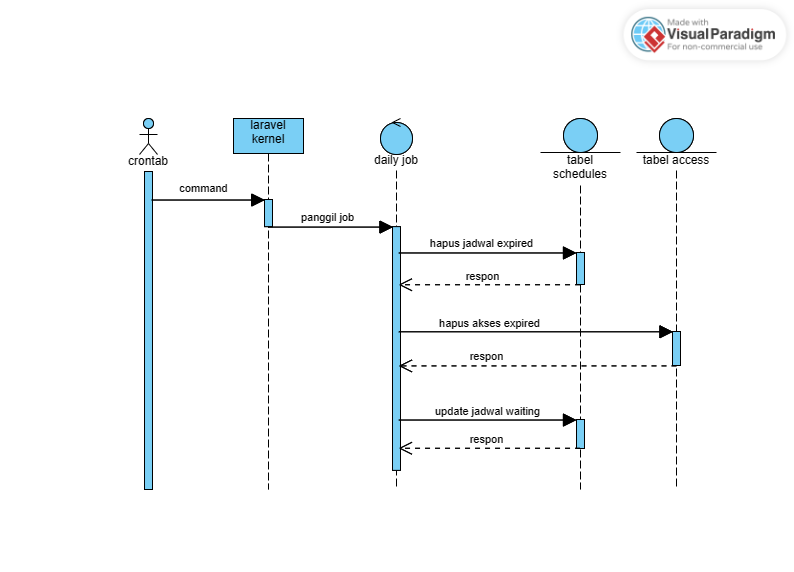


Diagram Pengecekan Jadwal

Terlihat pada gambar x.x diatas, crontab akan mengirimkan perintah ke kernel setiap 1 menit sekali dan kernel akan menjalankan sebuah job yang digunakan untuk memeriksa jadwal. Pertama job tersebut akan mengambil data jadwal hari ini yang akan dilaksanakan tentunya dengan menggunakan query untuk mengambil data yang sesuai dengan memeprthatikan tanggal, waktu dan status jadwal, setelah data didapatkan kemudian job akan mengirimkan event ke setiap perangkat kunci pintu yang ada pada data jadwal tersebut, setelah semua event dikirimkan kemudian job akan memperbarui status jadwal menjadi “running” atau sedang berjalan. Terakhir job akan memperbarui semua jadwal yang sudah dijalankan dan sudah melewati waktu pelaksanaan menjadi “done” atau sudah selesai dilaksanakan.

1. Atur Ulang Jadwal

Setiap jadwal yang sudah dilaksanakan haruslah dihapus dari database dan jadwal yang berulang haruslah diatur statusnya seperti awal mula sehingga memerlukan proses atur ulang jadwal untuk membersihkan dan mengatur ulang jadwal. Jadwal akan dibersihkan dan diatur ulang setiap 24 jam sekali dengan menggunakan sebuah job pada kernel laravel. Diagram dari job atur ulang jadwal dapat dilihat pada gambar x.x dibawah.



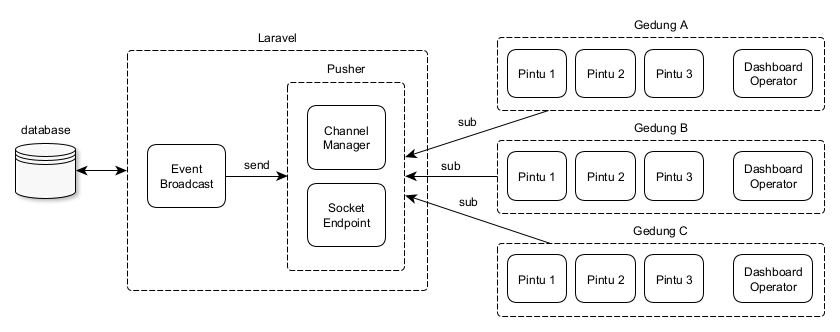
Atur Ulang Jadwal

Dapat dilihat pada gambar x.x diatas, crontab akan mengirimkan perintah ke kernel untuk menjalankan job atur ulang jadwal. Didalam job tersebut pertama job akan menghapus data jadwal yang sudah kadaluarsa pada tabel schedules yaitu jadwal yang sudah dilaksanakan dan tidak berulang, kemudian pada job ini juga akan menghapus kartu akses pengguna yang sudah kadaluarsa, kemudian job akan memperbarui setiap jadwal yang sudah pernah dilaksanakan menjadi “waiting” atau menunggu untuk dilaksanakan kembali.

1. Perancangan Komunikasi Websocket

Websocket digunakan sebagai jalur komunikasi yang menghubungkan antara server dengan perangkan kunci pintu, dengan adanya komunikasi websocket maka perangkat kunci pintu dan server akan selalu terhubung sehingga dapat berkomunikasi secara langsung.

Pada perancangan backend server untuk mendukung kinerja perangkat kunci pintu ini menggunakan pusher sebagai protokol komunikasi websocket yang menghubungkan antara perangkat kunci pintu dengan server dengan detail konfigurasi seperti yang terlihat pada gambar x.x dibawah.



Konfigurasi Websocket

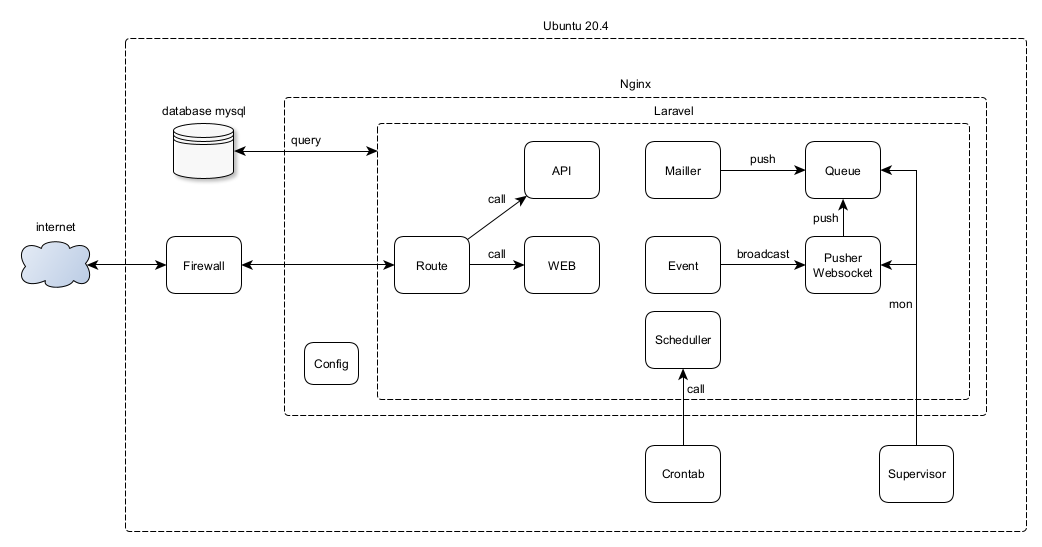
Dapat dilihat pada gambar x.x diatas, setiap pintu akan dikelompokkan berdasarkan dengan gedung dengan satu orang operator, setiap gedung akan memiliki satu channel pusher yang dapat disubscribe oleh perangkat penguncian didalam gedung tersebut, untuk mengawasi semua aktifitas yang terhubung ke channel dengan mudah maka konfigurasi channel menggunakan presence channel, dengan menggunakan presence channel maka aktifitas semua perangkat kunci pintu seperti perangkat kunci melakukan subscribe dan koneksi terputus dapat diketahui secara langsung.

Laravel mengirimkan data atau perintah ke perangkat kunci pintu melalui pusher menggunakan sebuah event, setiap event seperti kunci atau buka pintu dipanggil baik melalui website maupun API maka event tersebut akan disiarkan ke channel sesuai dengan kode perintah dan channel yang dituju.

Didalam pusher sendiri terdapat 2 komponen utama yaitu channel manager dan socket endpoint, channel manager digunakan untuk mengelola semua channel yang ada serta mengelola semua subsciber dan melakukan broadcast event, sedangkan socket endpoint digunakan sebagai pintu masuk berupa URL yang digunakan untuk memulai koneksi dengan websocket.

1. Perancangan Server

Dengan menggunakan laravel sebagai backend yang mengatur kinerja dari perangkat kunci pintu tentunya diperlukan sebuah server. Server ini akan bertindak sebagai pusat pengolahan data dan berfungsi untuk menerima permintaan dari perangkat kunci pintu, mengatur akses, memproses logika bisnis, dan berkomunikasi dengan database. Diagram dari backend server dapat dilihat pada gambar x.x dibawah.



Konfigurasi Server

Dapat dilihat pada gambar x.x diatas, server dibangun menggunakan sistem operasi ubuntu 20.04, ubuntu merupakan bagian dari sistem operasi linux yang biasa digunakan baik untuk perangkat dekstop maupun server karena open source dan ringan. Dengan menggunakan ubuntu 20.04 sebagai sistem operasi server, kita dapat memanfaatkan kestabilan, keamanan, dan dukungan jangka panjang untuk menjalankan aplikasi laravel dengan aman dan efisien.

Didalam sistem operasi ubuntu 20.04 dipasang nginx yang digunakan sebagai web server untuk menjalankan aplikasi laravel. Dengan menggunakan nginx maka aplikasi laravel dapat dijalankan dengan efisien dikarenakan nginx memiliki karakteristik ringan dan cepat. Dengan menggunakan nginx kita juga bisa membagi server menjadi beberapa blok yang dapat digunakan untuk menjalankan API dan websocket secara bersamaan. Pada nginx juga dipasang sertifikat SSL yang digunakan untuk mengenkripsi semua komunikasi dengan menggunakan HTTPS sehingga kemanan data akan terjamin dengan adanya jalur komunikasi yang terenkripsi.

Didalam ubuntu juga dipasang MySQL sebagai pusat penyimpanan data yang terhubung ke laravel, dengan menggunakan database yang berjalan pada server yang sama maka kecepatan transfer data akan sangat cepat dengan menghilangkan latensi jaringan hal ini sesuai dengan karakteristik sistem yang dibangun yaitu sistem cepat dan efisien.

Dengan adanya fitur penjadwalan otomatis didaiam sistem kunci pintu maka kita juga harus memasang crontab untuk menjalankan penjadwalan yang ada pada laravel, penjadwalan akan dipanggil setiap 1 menit sekali untuk memeriksa apakah ada jadwal yang harus dilaksanakan atau tidak.

Pada laravel juga mengimplementasikan fitur antrian yang bertujuan untuk meningkatkan kinerja dasi sistem sehingga fungsi antrian harus dijaga agar selalu berada dalam kondisi berjalan, oleh karena itu maka pada server suja dipasang supervisor yang digunakan untuk memantau kinerja dari antrian. Supervisor merupakan sebuah sistem pengawas atau process control system yang digunakan untuk mengontrol dan mengelola proses-proses yang berjalan didalam sistem operasi. Supervisor akan memastikan bahwa proses antrian pada laravel tetap berjalan secara terus-menerus dan akan memulai ulang proses jika terjadi kegagalan serta mengelola jumlah pekerja antrian atau queue workers yang berjalan secara paralel untuk meningkatkan efisiensi dan kinerja.

Supervisor juga digunakn untuk menjaga websocket berjalan secara terus menerus, websocket memegang peranan penting didalam siste ini karena websocket menjadi jalur komunikasi yang menghubungkan perangkat kunci pintu dengan server pusat, sehingga jika terjadi gangguan pada kinerja websocket maka seluruh kinerja dari perangkat kunci pintu akan terganggu, oleh karena itu diperlukan pengawasan menggunakan supervisor untuk menjaga kinerja dari websocket.

BAB IV

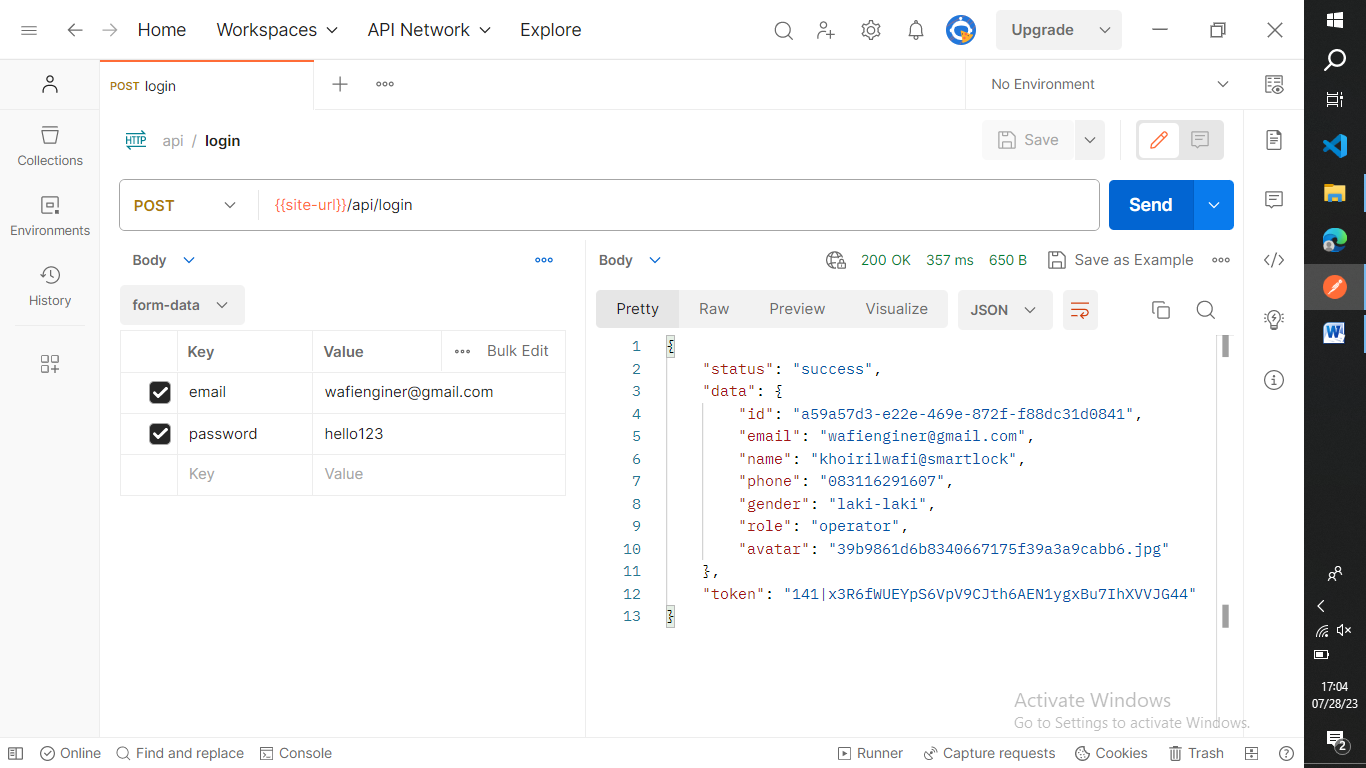
**HASIL DAN PEMBAHASAN**

1. Kajian Hasil Perancangan

Untuk melihat hasil dari perancangan API aplikasi mobile dan perangkat kunci pintu maka dilakukan uji coba menggunakan postman. Postman merupakan sebuah software yang digunakan untuk melakukan request ke API, dengan menggunakan postman maka akan terlihat hasil respon yang dikirimkan oleh server sesuai dengan permintaan pelanggan melalui endpoint tertentu.

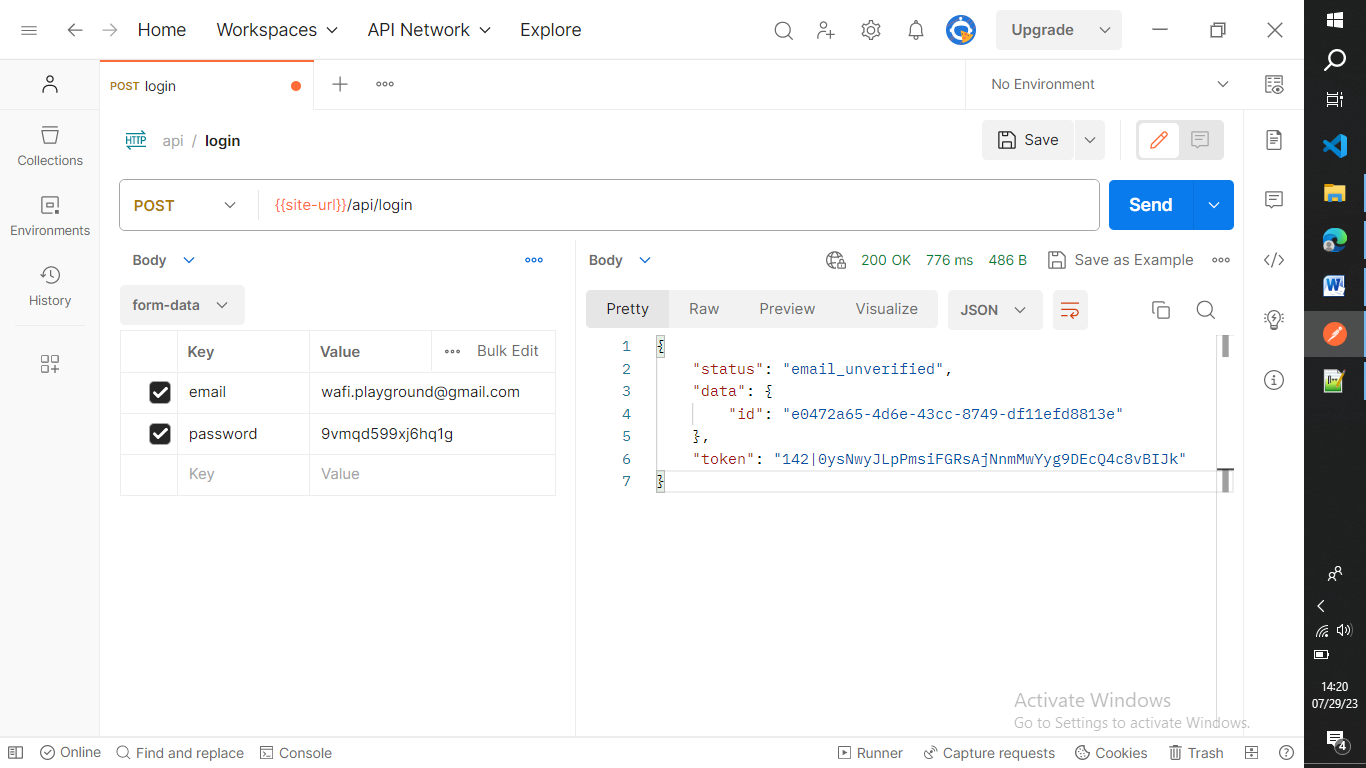
1. API Login

API login akan menerima alamat email dan password pengguna untuk melakukan autentikasi. Hasil dari perancangan dari API login dapat dilihat pada gambar x.x dibawah.



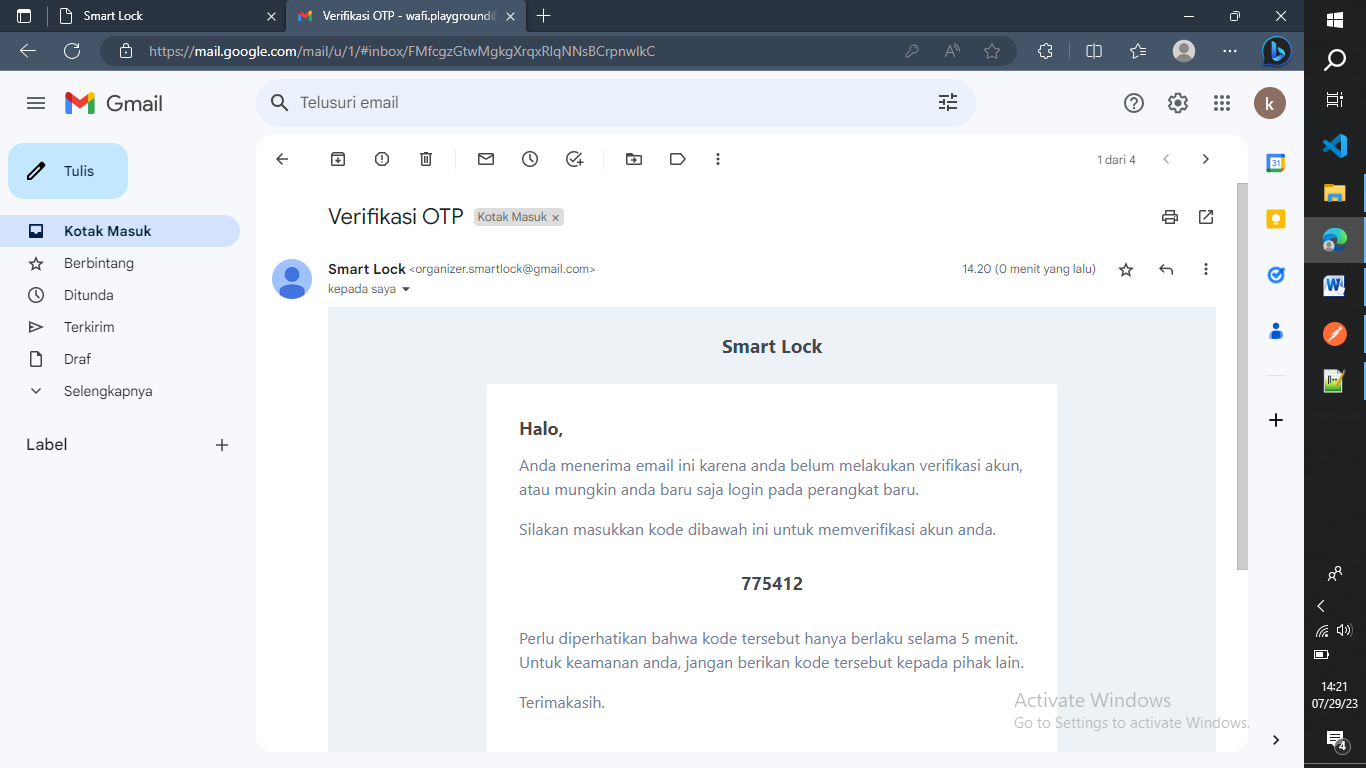
Hasil API Login

Gambar x.x menampilkan hasil request login yang dikirimkan melalui enpoint “/api/login”. API login akan memberikan respon status login, data pengguna yang melakukan login serta token yang akan digunakan untuk mengakses API lainnya yang membutuhkan autentikasi token.



Hasil API Login Verifikasi Email

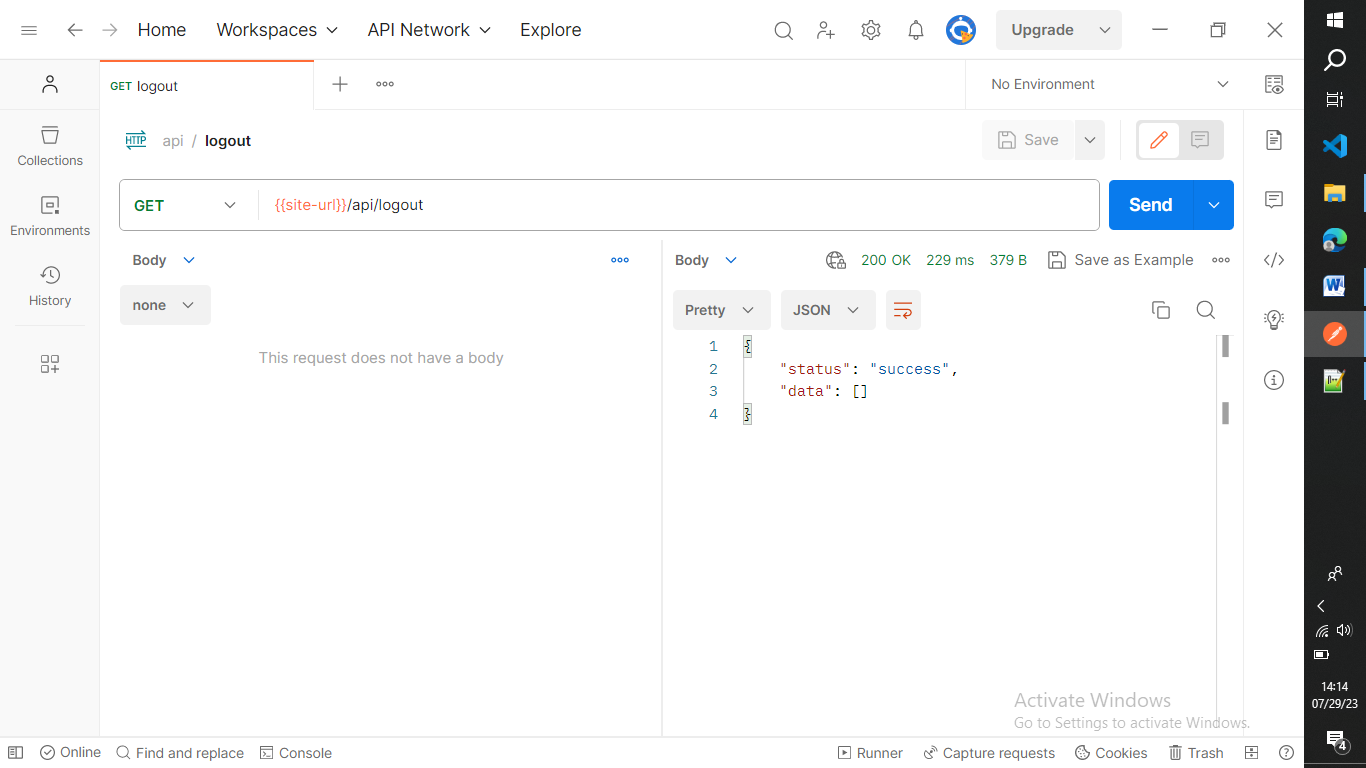
Pada API login juga dilakukan pengecekan email pengguna, seperti yang terlihat pada gambar x.x pengguna yang belum melakukan verifikasi email akan menerima respon status email belum terverifikasi dan pengguna akan menerima kode OTP yang dikirimkan ke email pengguna seperti yang terlihat pada gambar x.x. Pada kondisi ini API juga memberikan token yang akan digunakan oleh pengguna untuk melakukan verifikasi email karena API untuk melakukan verifikasi email membutuhkan akses login.



Email Kode OTP

1. API Logout

API logout digunakan untuk mengakhiri sesi login pengguna dengan menghapus semua token yang dimiliki oleh pengguna. API logout akan menerikan permintaan logout pengguna dengan melampirkan token yang dimiliki. Hasil dari API logout dapat dilihat pada gambar x.x dibawah.

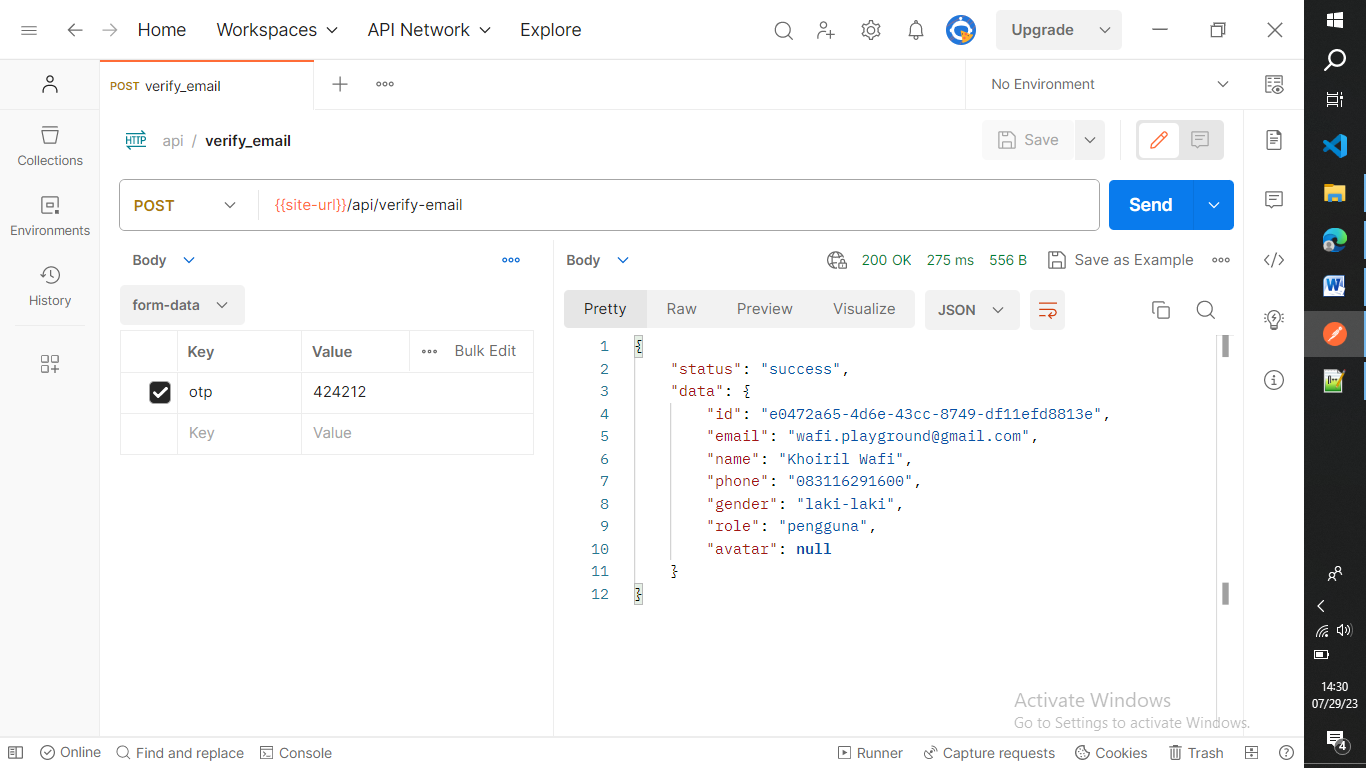


Hasil API Logout

Gambar x.x menampilkan hasil request dari API logout yang dikirimkan melalui endpoint “/api/logout”. API logout akan memberikan respon status logout yang dilakukan oleh pengguna dengan melampirkan token yang dimiliki oeleh pengguna yang bersangkutan pada bagian header request.

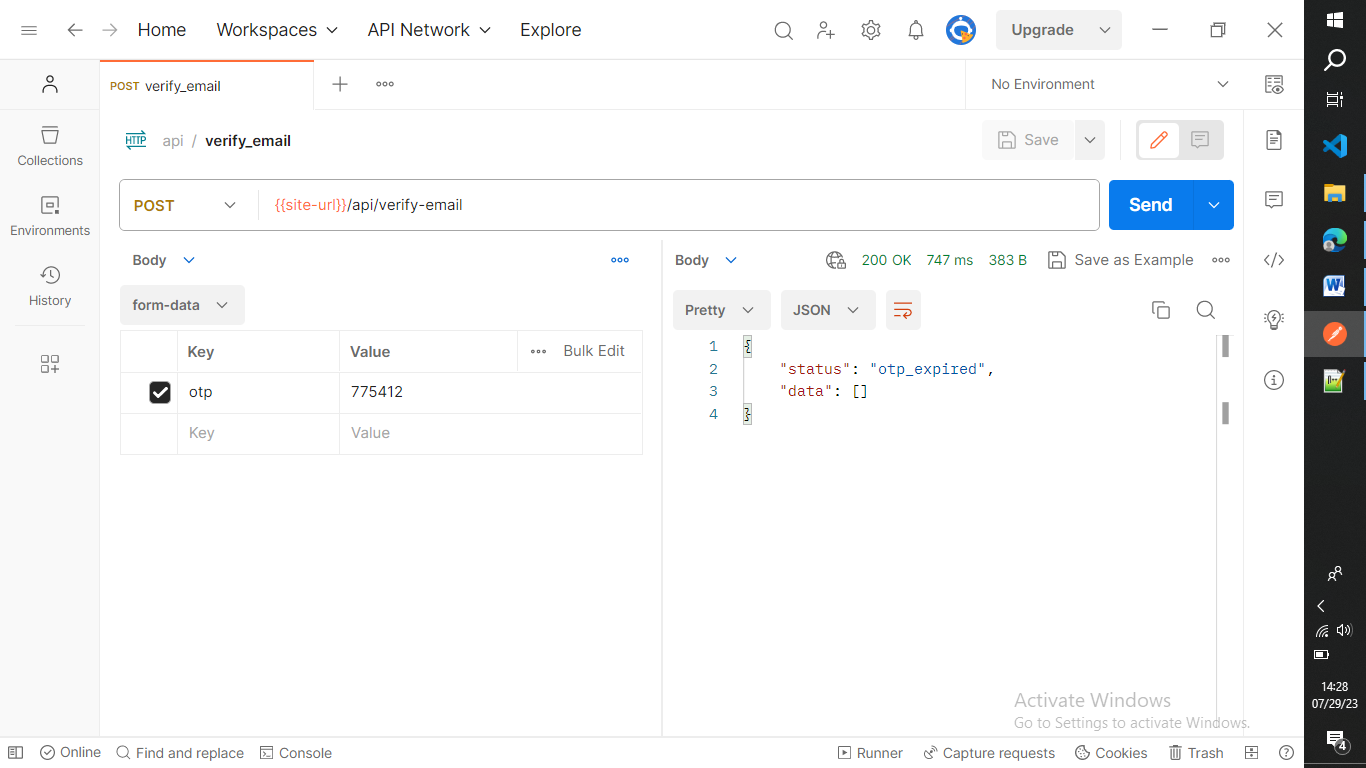
1. API Verifikasi Email

Pada saat pengguna melakukan login dan mendapatkan respon email belum terverifikasi maka penggunakan akan mendapatkan sebuah kode OTP yang dikirimkan ke email pengguna. API verifikasi email akan menerima token OTP yang telah diterima untuk melakukan verifikasi email. Hasil dari API verifikasi email dapat dilihat pada gambar x.x dibawah.



Hasil API Verifikasi Email

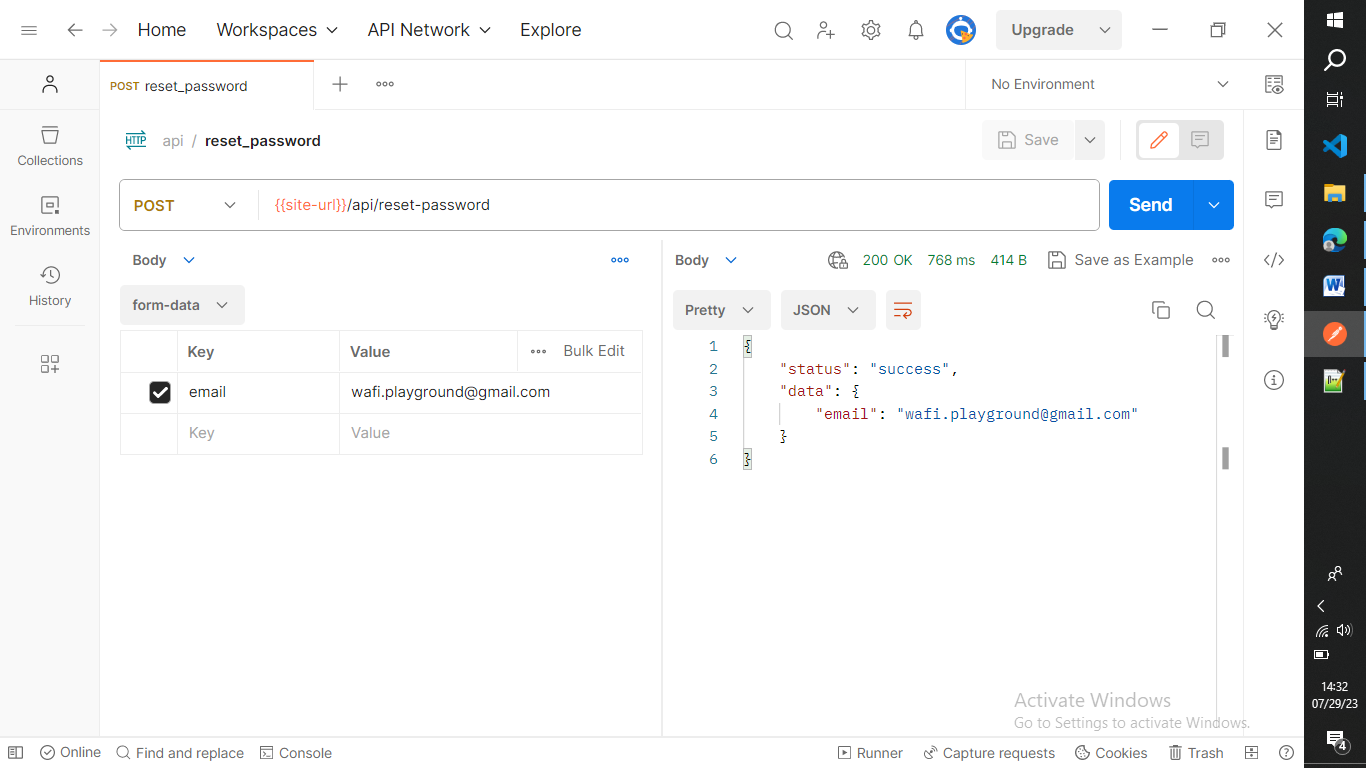
Gambar x.x menampilkan hasil dari API verifikasi email yang dikirimkan melalui enapoint “api/verify-email”. Pengguna akan mengirimkan kode OTP melaui body serta melampirkan token login pada header request, jika verifikasi berhasil maka API akan memberikan respon status dan mengirimkan data pengguna seperti pada saat login. API verifikasi email juga akan memeriksa waktu kadaluarsa kode OTP, jika kode OTP sudah kadaluarsa maka verifikasi akan gagal seperti yang terlihat pada gambar x.x dibawah.



Kode OTP Kadaluarsa

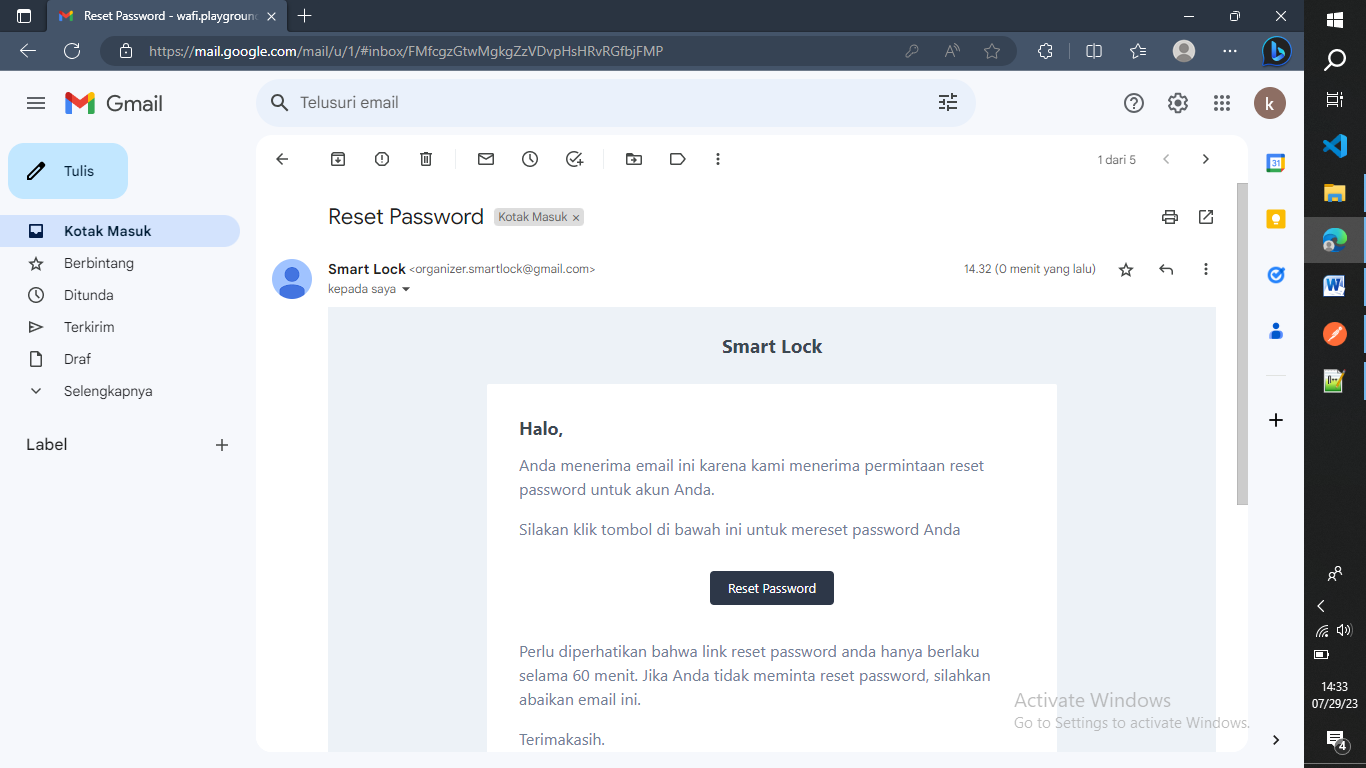
1. API Reset Password

API reset password akan menerima email dari pengguna dan mengirimkan url yang dapat digunakan untuk membuat password baru ke email pengguna. Hasil dari API reset password dapat dilihat pada gambar x.x dibawah.



Hasil API Reset Password

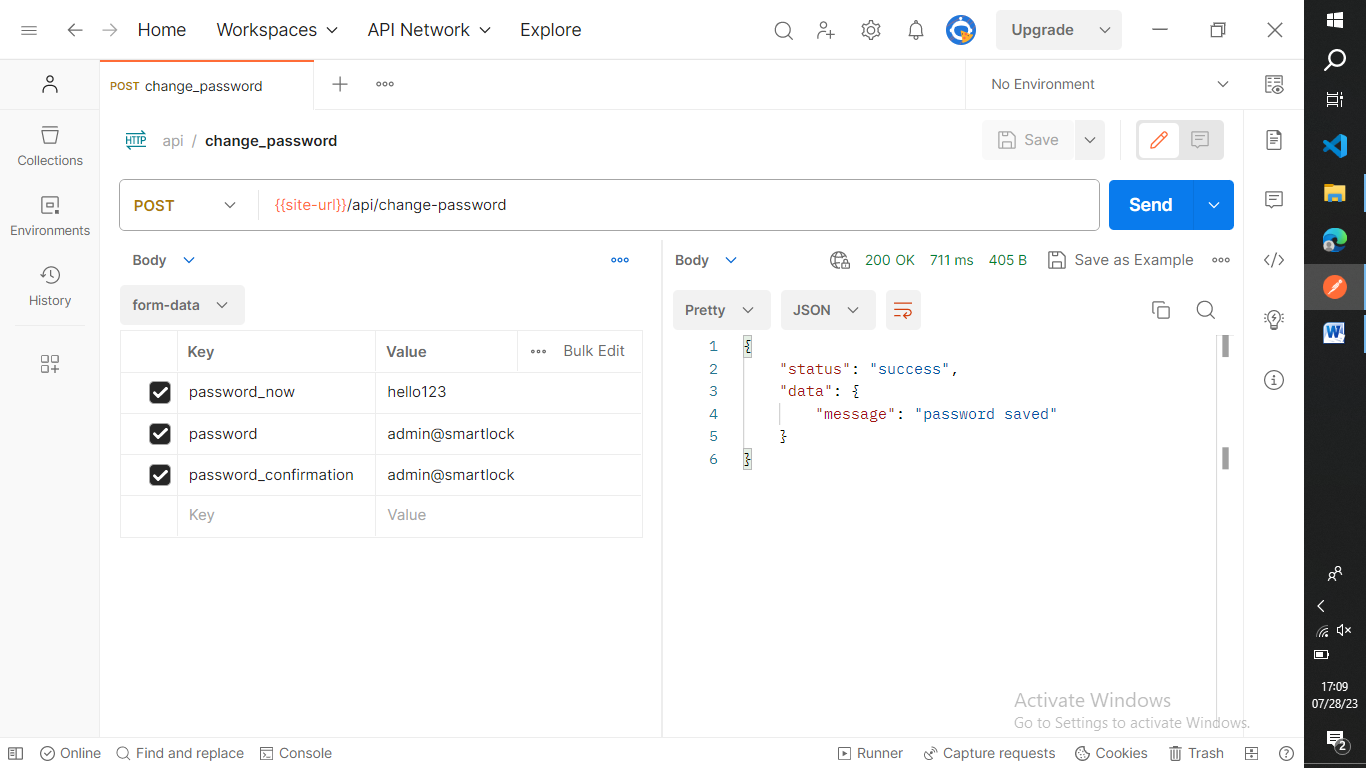
Gambar x.x merupakan hasil API reset password yang dikirimkan melalui endpoint “/api/reset-password”. Pengguna akan mengirimkan alamat email pada body request dan API akan memberikan respon status beserta data alamat email. Pada proses ini API juga akan mengirimkan url reset password ke email pengguna seperti yang terlihat pada gambar x.x, pengguna dapat menggunakan url tersebut untuk membuat password baru melalui halaman website.



Email Reset Password

1. API Ganti Password

API ganti password akan menerima password lama dan password baru pengguna untuk memperbarui password pengguna. API ganti password juga akan memeriksa token login yang dikirimkan untuk memastikan bahwa pengguna sudah melakukan proses autentikasi login. Hasil dari API ganti password dapat dilihat pada gambar x.x dibawah.

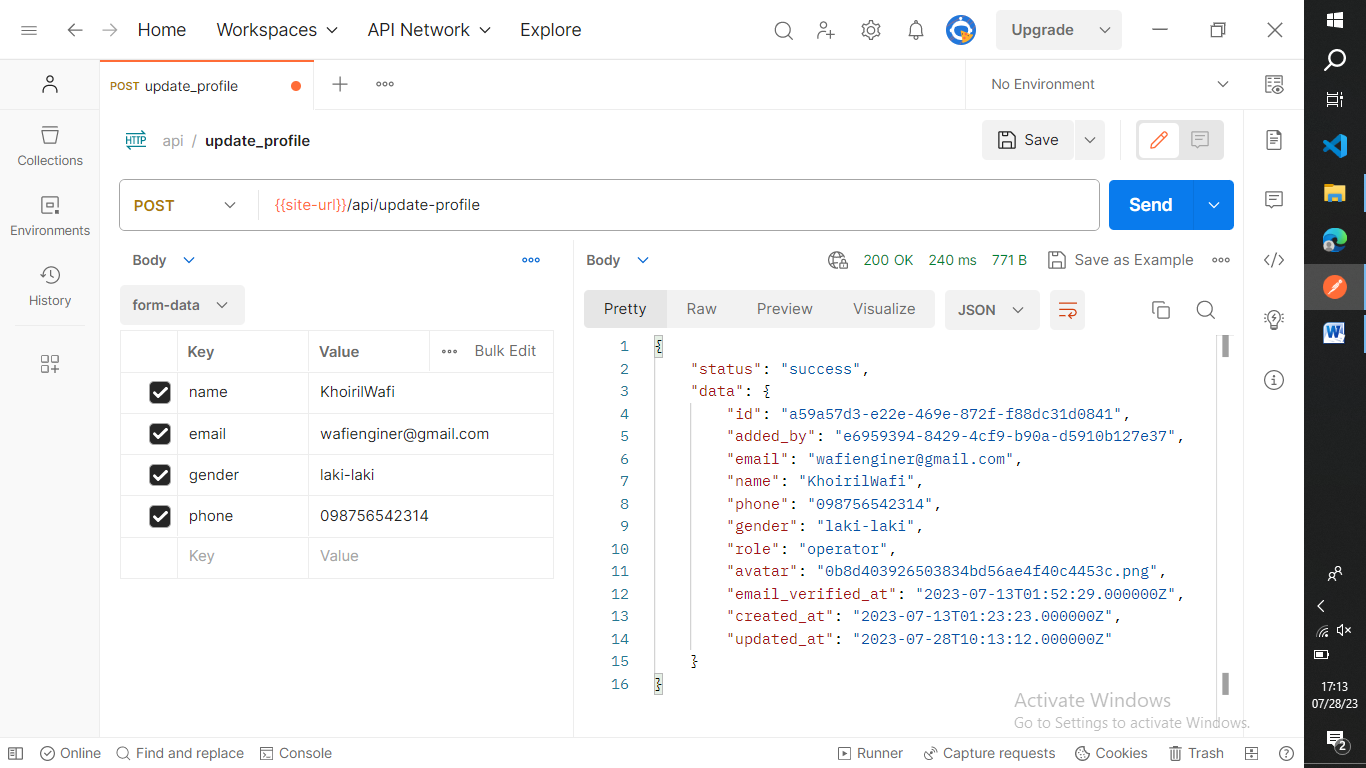


Hasil API Ganti Password

Gambar x.x merupakan hasil dari API ganti password yang dikirimkan melalui enpoint “/api/change-password”. Pengguna akan mengirimkan password lama dan password baru serta konfirmasi password baru pada body request, penggunakan juga memenrlukan token login untuk mengakses enpoint ini. API ganti password akan memberikan respon status dan pesan apakah password baru berhasil tersimpan atau tidak.

1. API Ganti Profil

API ganti profil akan menerima data pengguna yang akan diganti seperti nama, email, jenis kelamin dan nomor hp, API ganti profile juga memerlukan token login untuk memastikan pengguna sedah melakukan autentikasi. Hasil dari API ganti profil dapat dilihat pada gambar x.x dibawah.

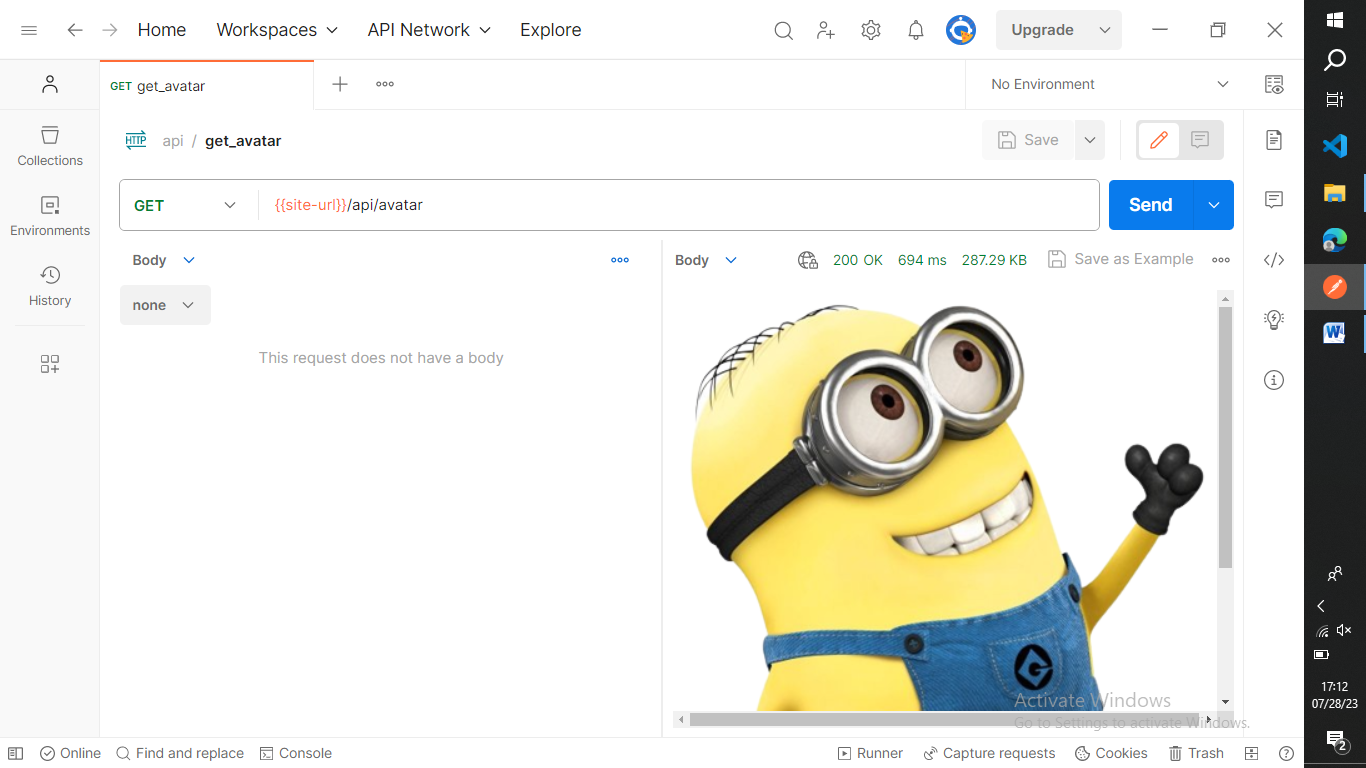


API Ganti Profil

Gambar x.x merupakan hasil dari API ganti profil yang dikirimkan melalui endpoint “/api/update-profile”. Penggunakan akan mengirimkan data yang akan diganti seperti nama, email, jenis kelamin dan nomor hp pada body request, penggunak juga menyertakan token login pada header request. API akan memberikan respon berupa status dan data penggunaka yang sudah dirubah seperti yang terlihat pada gambar x.x diatas.

1. API Lihat Avatar

Setiap gambar avatar pengguna tidak tersedia untuk publik, pengguna akan melakukan request untuk mendapatkan gambar avatar dengan melampirkan token login ke API lihat avatar. Hasil dari API lihat avatar dapat dilihat pada gambar x.x dibawah.

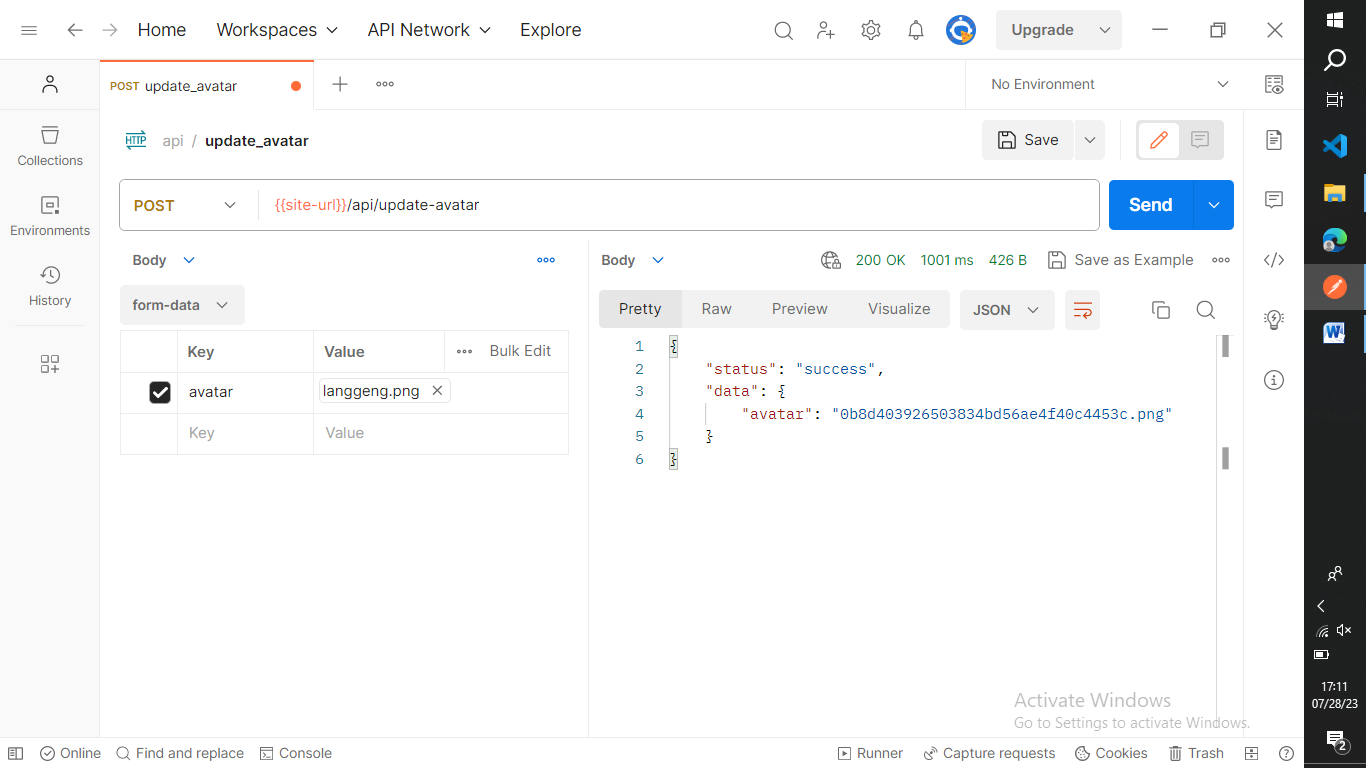


Hasil API Lihat Avatar

Gambar x.x merupakan hasil dari API livat avatar yang dikirimkan melalui endpoint “/api/avatar”, pengguna akan mengirimkan permintaan disertai dengan token login dan API akan memberikan respon berupa gambar avatar masing-masing pengguna dalam format file gambar. Jika pengguna belum memiliki avatar maka API akan mengembalikan respon null.

1. API Ganti Avatar

Untuk mengganti avatar melalui aplikasi mobile pengguna akan mengirimkan gambar avatar melalui API ganti avatar, pengguna juga memerlukan akses login untuk mengakses API ini. Hasil dari API ganti avatar dapat dilihat pada gambar x.x dibawah ini.

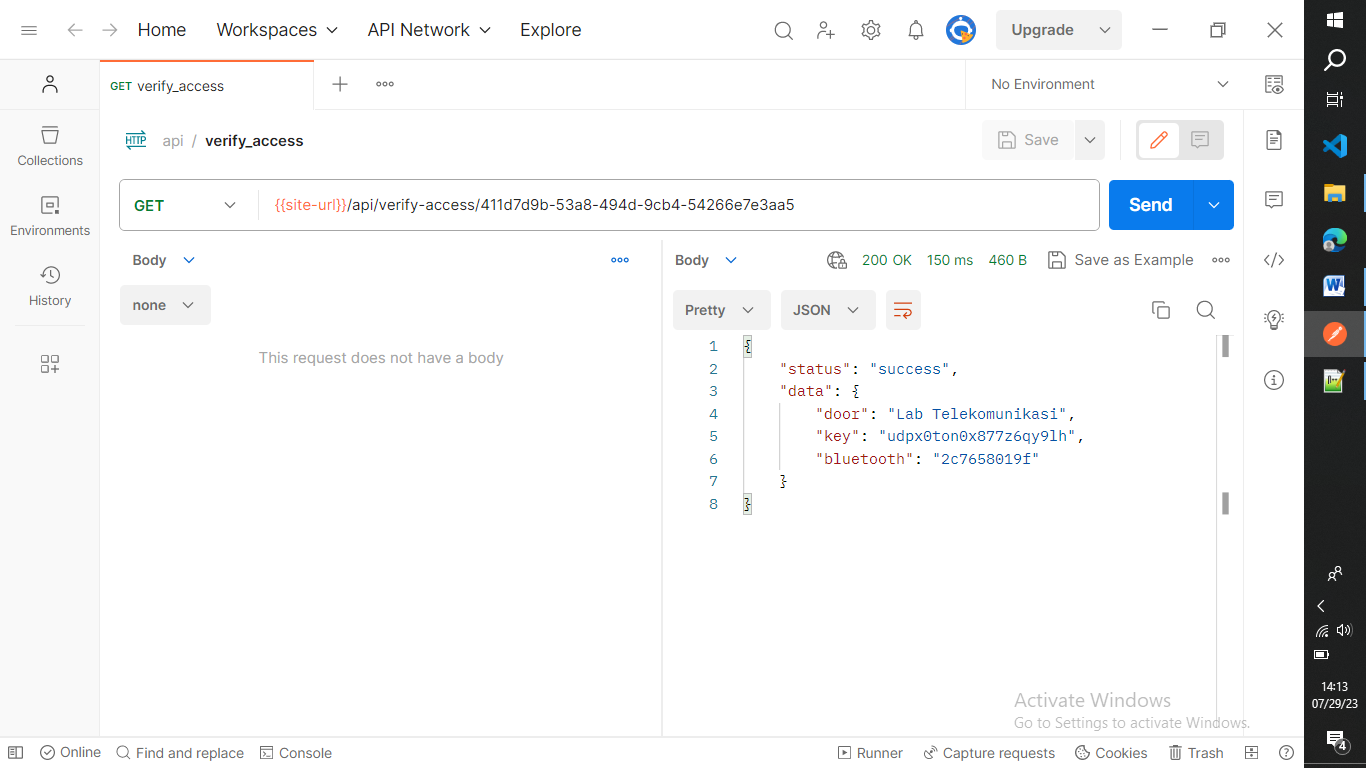


Hasil API Ganti Avatar

Gambar x.x merupakan hasil dari API ganti avatar yang dikirimkan melalui endpoint “/api/update-avatar”. Pengguna akan mengirimkan file gambar avatar ke API disertai dengan token login yang dimiliki, API akan memberikan respon status dan nama file avatar yang berhasil tersimpan didalam server.

1. API Verifikasi Akses

Ketika pengguna melakukan pemindaian kode QR untuk membuka pintu maka perangkat mobile akan mengirimkan request ke API verifikasi akses untuk mendapatkan kunci dari pintu tersebut. Kode QR yang terpasang nantinya akan menyimpan informasi identitas dari pintu yang digunakan untuk membedakan pintu satu dengan yang lainnya. Hasil dari API verifikasi akses dapat dilihat pada gambar x.x dibawah.

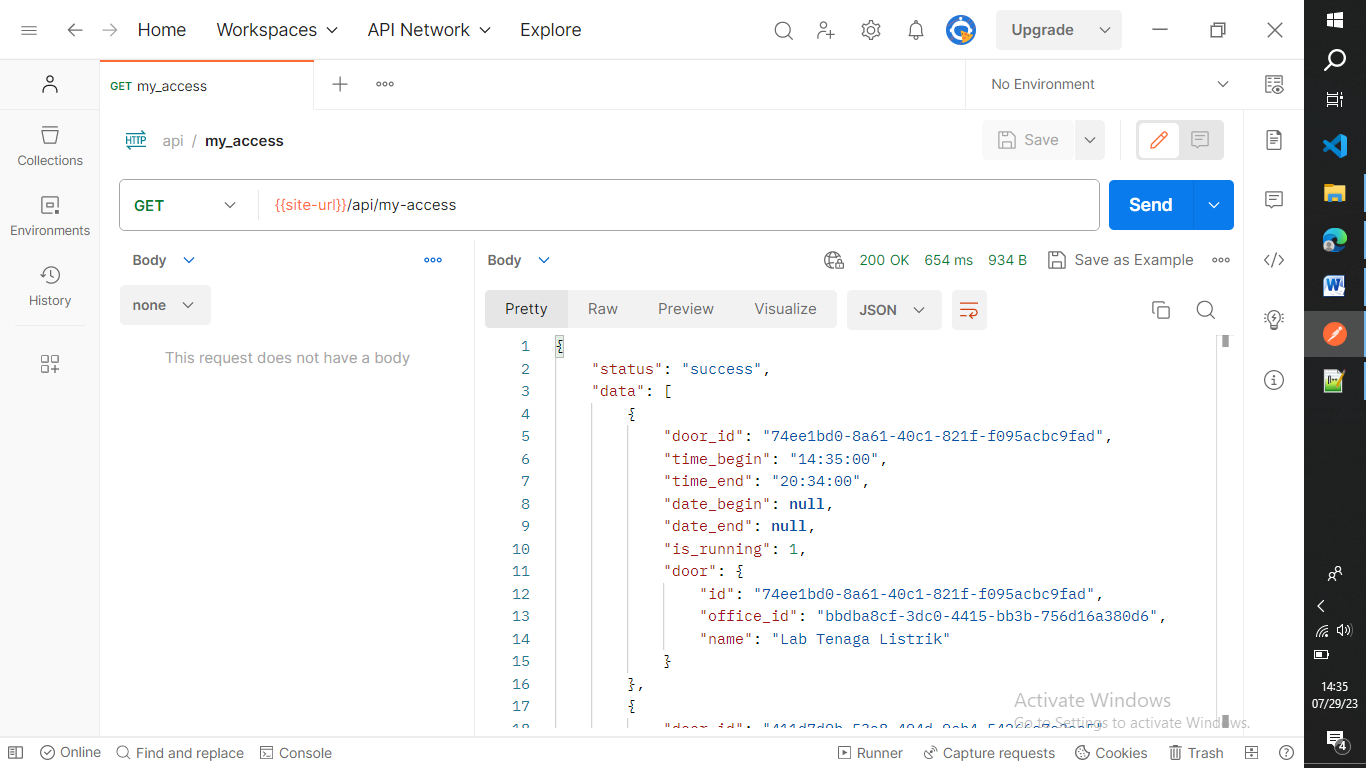


Hasil API Verifikasi Akses

Gambar x.x merupakan hasil dari API verifikasi akses yang dikirimkan melalui endpoint “/api/verify-access/{id}” dimana id merupakan identitas dari pintu. Pengguna akan mengirimkan token login yang dimiliki dan API akan memberikan respon terkait akses yang dimiliki pengguna tersebut, jika diizinkan maka API akan memberikan respon status dan data untuk membuka pintu seperti nama pintu, kunci pintu dan nama bluetooth untuk berkomunikasi dengan perangkat kunci pintu.

1. API Daftar Akses

API daftar akses akan memberikan informasi pintu mana saja yang dapat diakses oleh pengguna. Setiap penggguna dengan pengguna lain mungkin memiliki akses yang berbeda, pemberikan akses pintu kepada pengguna tertentu akan diatur oleh seoreang operator. Seorang operator dapat memberikan akses atau menghapus akses pengguna pada pintu tertentu, operator juga dapat menghentikan sementara akses pengguna untuk keperluan tertentu. Hasil dari API daftar akses dapat dilihat pada gambar x.x dibawah.

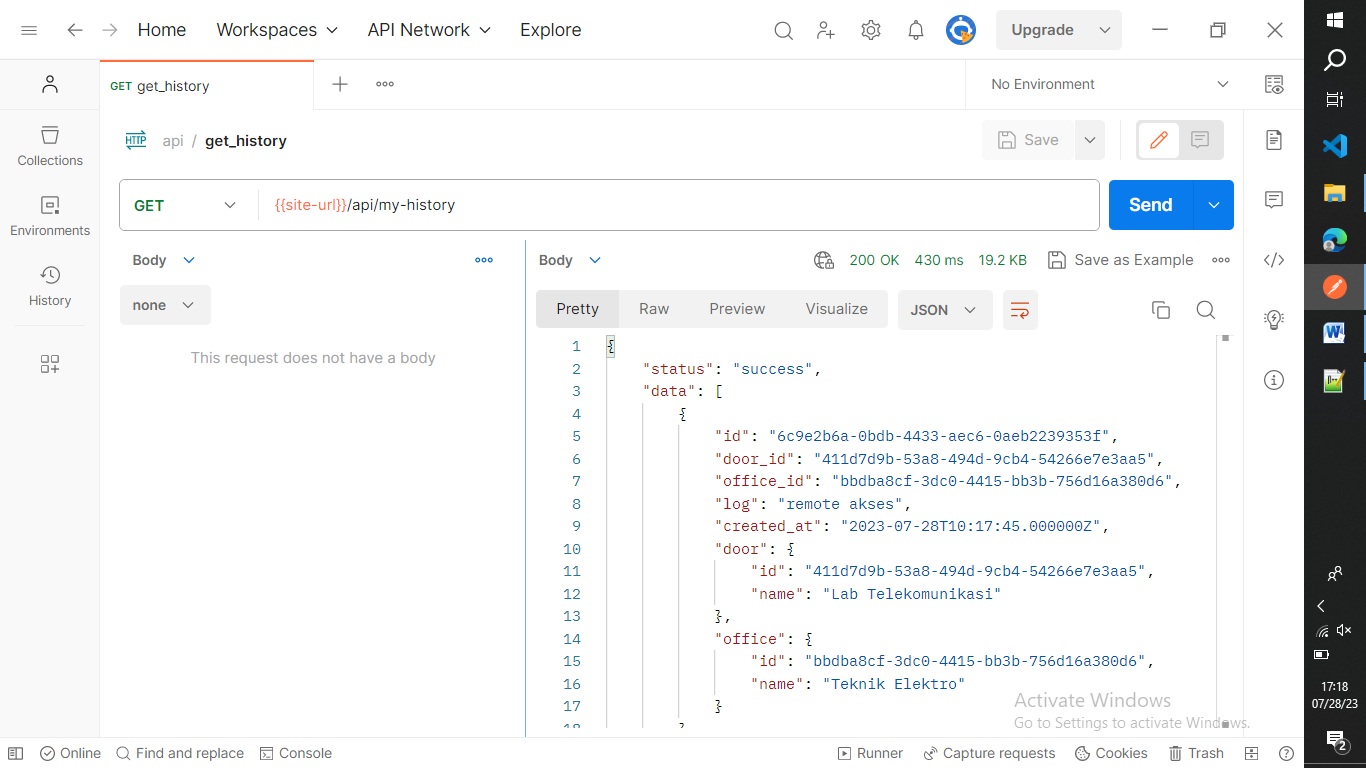


Hasil API Daftar Akses

Gambar x.x merupakan hasil dari API daftar akses yang dikirimkan oleh pengguna melalui enpoint “/api/my-access”. API daftar akses akan memberikan respon berupa daftar informasi pintu yang dapat diakses oleh penggua yaitu berupa nama pintu, nama gedung, rentang waktu akses, rentang tanggal akses dan status akses. Dengan menggunakan data akses yang dimiliki pengguna dapat mengetahui pintu mana saja yang dapat diakses oleh pengguna tersebut dan juga mengetahui jika akses pengguna masih aktif atau sedang ditangguhkan.

1. API Riwayat Akses

API riwayat akses akan memberikan informasi kepada pengguna terkait dengan catatan aktifitas pengguna didalam sistem penguncian gedung ini dengan menampilkan data riwayat aktifitas pada aplikasi pengguna. API ini dapat diakses oleh pengguna maupun operator untuk melihat aktifitas mereka. Hasil dari API dapat dilihat pada gambar x.x dibawah.

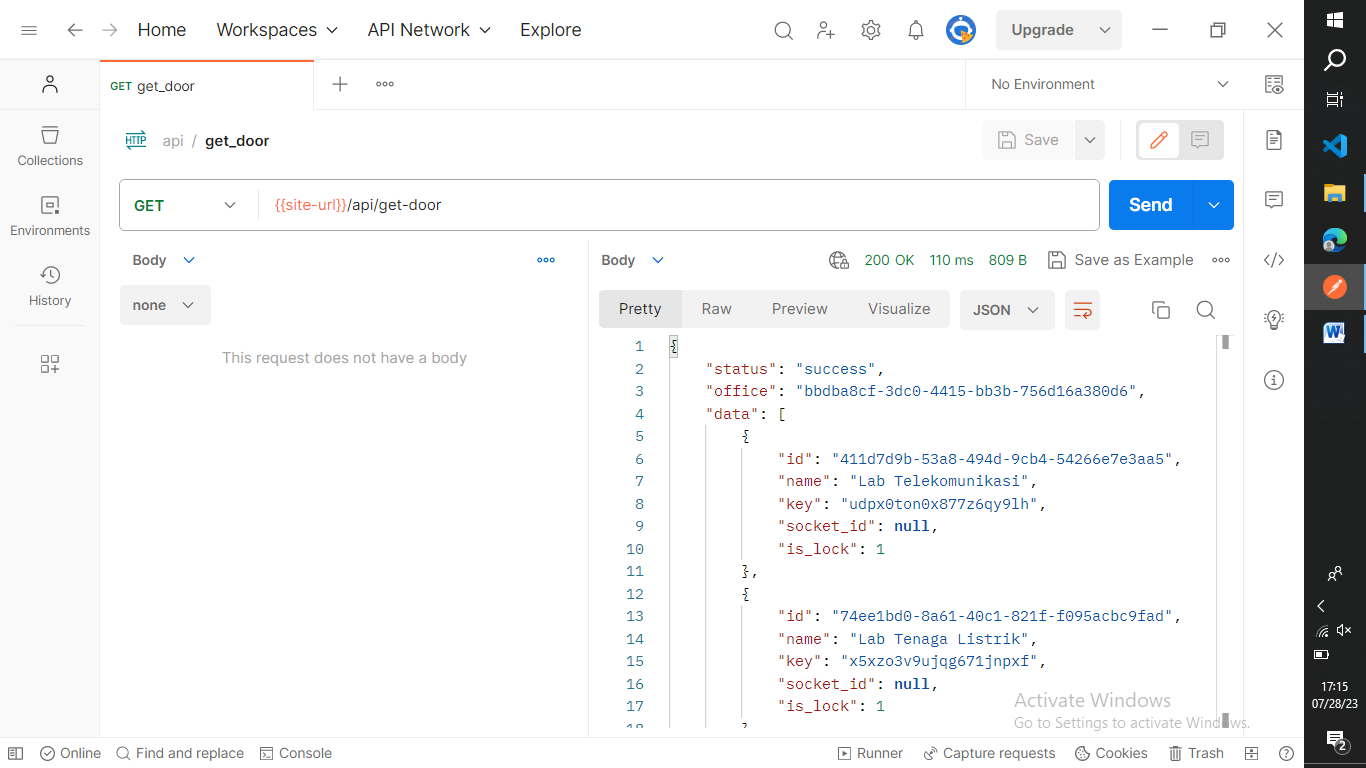


API Riwayat Akses

Gambar x.x merupakan hasil dari API riwayat akses. API riwayat akses akan memberikan data pencatatan aktifitas pengguna seperti melakukan pindai kode QR dan membuka pintu disertai dengan waktu dan tanggal aktifitas tersebut. Setiap aktifitas penguncian akan tersimpan didalam database sebagai riwayat aktifitas, serang operator dapat melihat semua aktifitas yang terjadi didalam sistem penguncian tetapi pengguna hanya dapat melihat aktifitas dirinya sendiri. Pada gambar x.x terlihat data riwayat aktifitas berupa nama pintu, waktu, dan jenis aktifitas yang dilakukan. Data riwayat akses hanya tersedia dalam kurun waktu 1 bulan dan setiap awal bulan maka data riwayat akses akan di pindahkan kedalam penyimpanan server (backup) dan tabel riwayat akses dalam database akan dibersihkan.

1. API Daftar Pintu

API daftar pintu digunakan untuk mendapatkan daftar pintu yang ada pada sebuah gedung, data tersebut kemudian ditampilkan pada aplikasi mobile yang akan memberikan informasi terkait kondisi pintu apakah terkunci, offline atau kondisi lainnya. API ini hanya dapat diakses oleh operator gedung karena hak pengelolaan pintu gedung berada pada tangan operator. Hasil dari API daftar pintu dapat dilihat pada gambar xx dibawah.

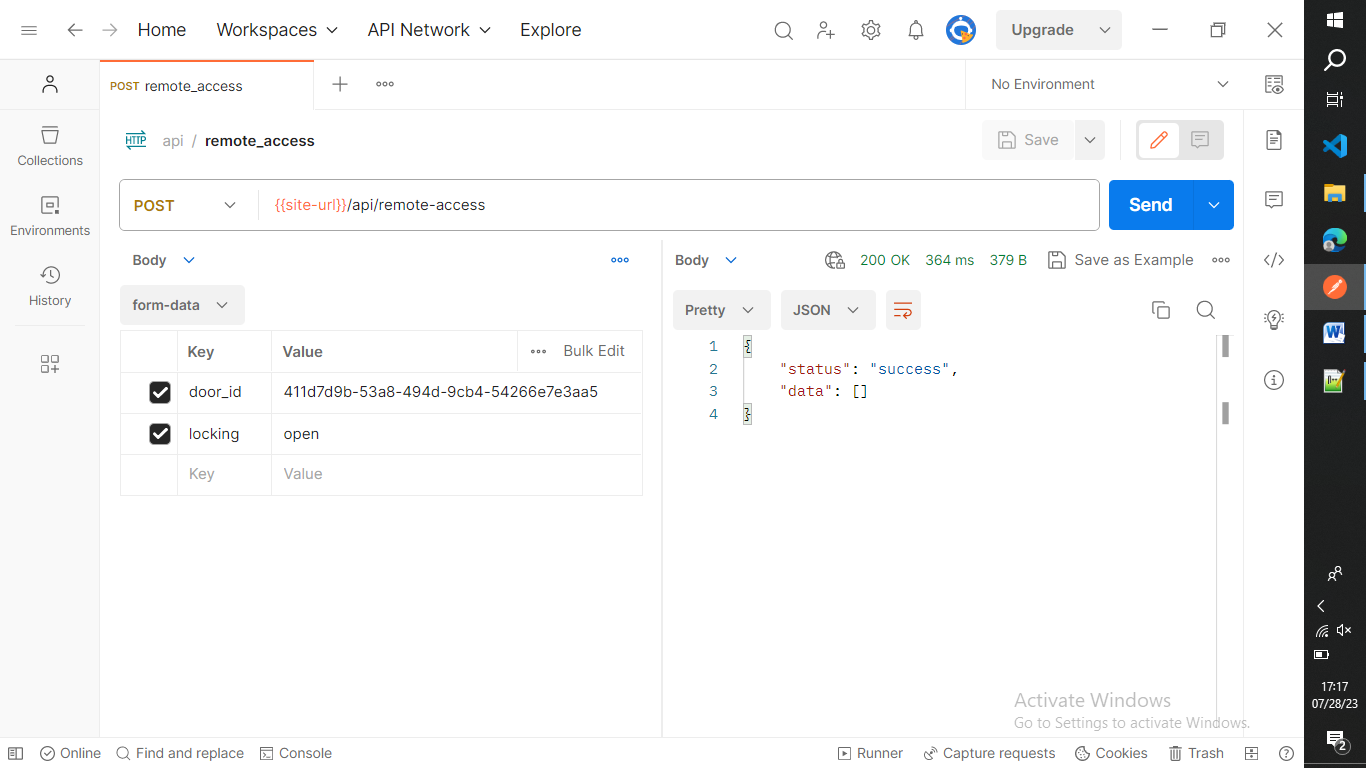


Hasil API Daftar Pintu

Gambar x.x merupakan hasil dari API daftar gedung yang dikirimkan melalui endpoint “/api/get-door”. Pengguna akan mengirimkan token login yang dimiliki, jika pengguna merupakan seorang operator maka PAI akan memberikan respon daftar pintu yang ada pada gedung tersebut beserta dengan informasi tambahan seperti nama pintu, kondisi penguncian, koneksi, dan kunci pintu.

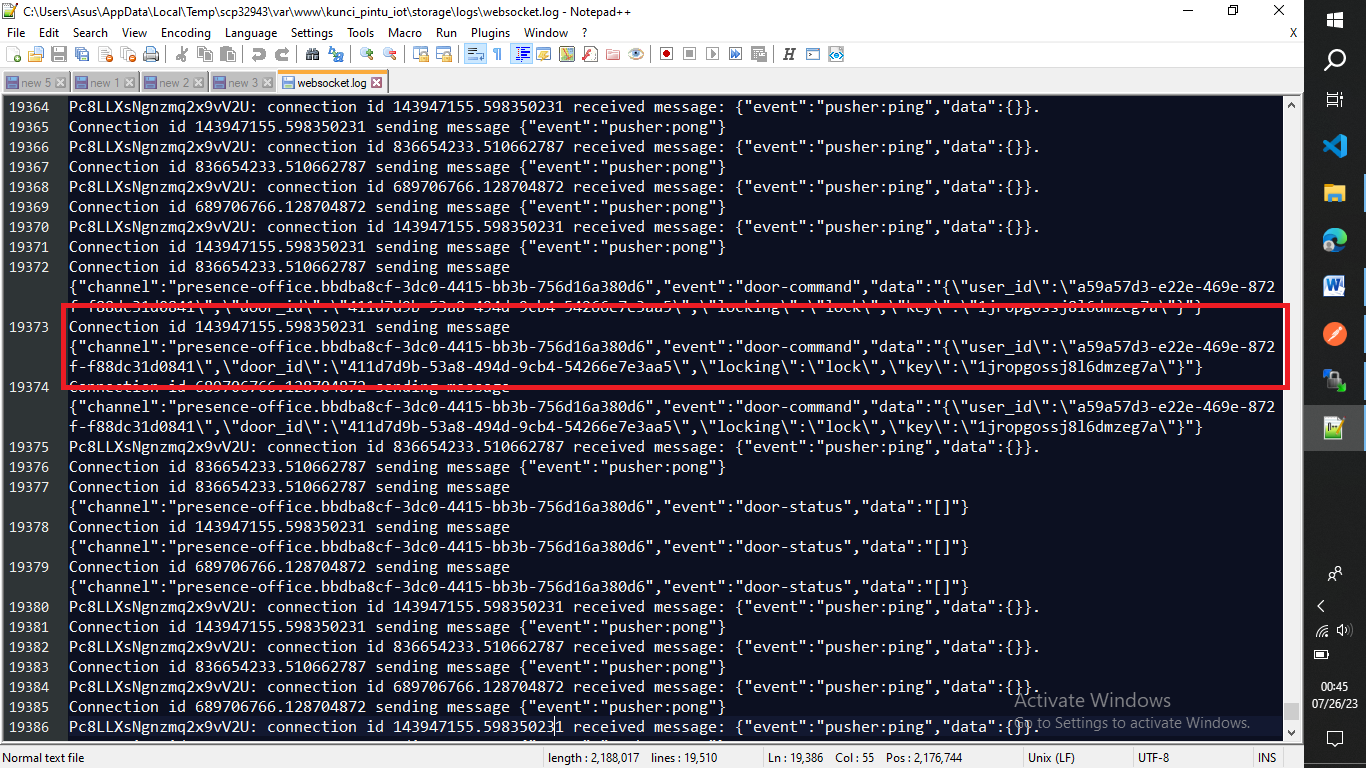
1. API Remote Pintu

Pada aplikasi yang digunakan oeleh operator terdapat tombol yang dapat digunakan untuk membuka dan mengunci pintu sehingga memungkinkan operator dapat melakukan pengelolaan pintu secara jarak jauh melalui koneksi internet. Operator akan melakukan request dengan mengirimkan identitas pintu beserta token login yang dimiliki, jika sesuai maka server akan mengirimkan perintah ke perangkat kunci pintu melalui koneksi websocket yang sudah terhubung. Hasil dari API remote pintu dapat dilihat pada gambar x.x dibawah.



Hasil API Remote Pintu

Gambar x.x merupakan hasil dari API remote pintu yang dikirmkan melalui endpoint “/api/remote-access”. Untuk membuka atau mengunci pintu sacara jarak jauh melalui koneksi internet operator akan mengirimkan identitas pintu yang dituju serta aktifitas yang dilakukan seperti lock atau open yang menjelaskan aktifitas membuka dan mengunci pintu, kemudian server akan mmemeriksa permintaan dan mengirimkan perintah ke perangkat kunci pintu melalui koneksi websocket seperti yang terlihat pada gambar x.x dibawah.

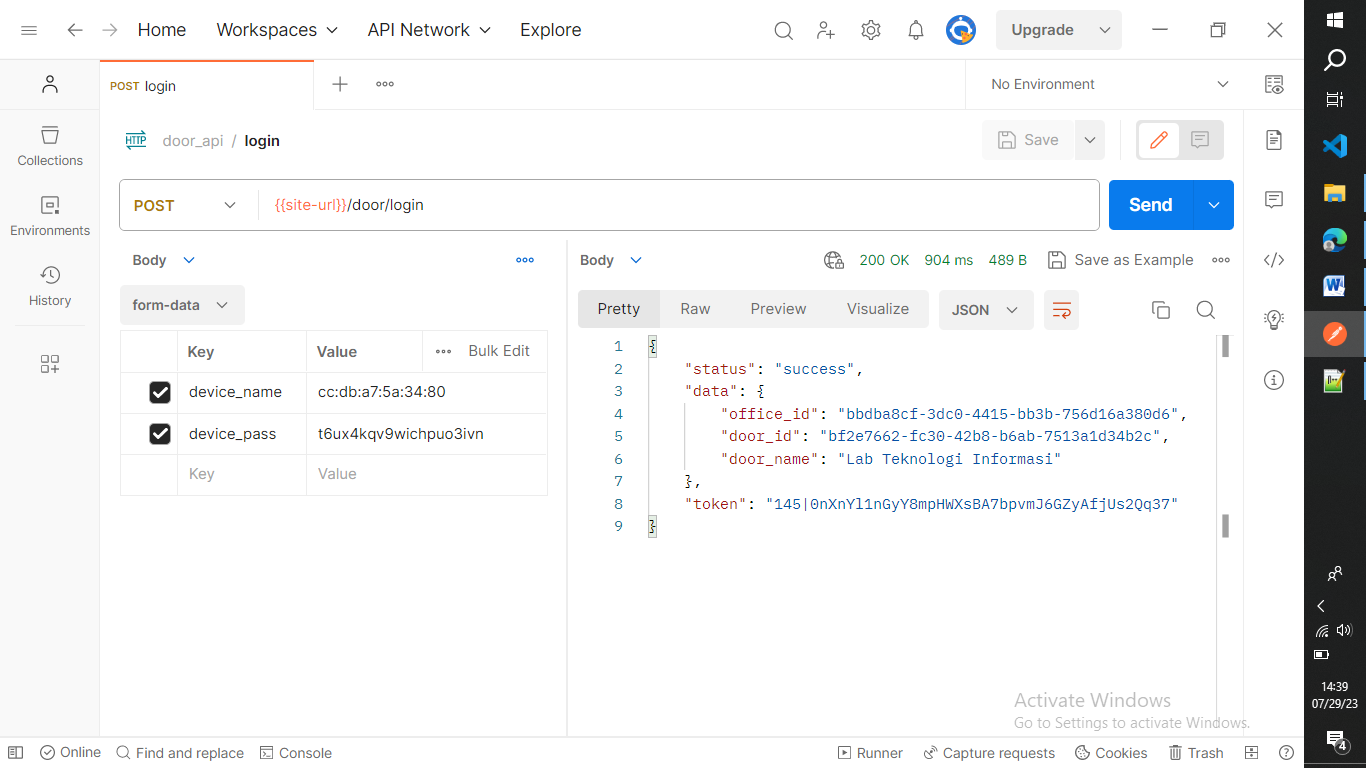


Aktifitas Door Remote

Gambar x.x diatas memperlihatkan aktifitas websoket pada server, pada saat API remote pintu dipanggil maka server akan mengirimkan perintah ke perangkat kunci pintu dengan pada channel yang sudah ditentukan. Websocket akan mengirimkan data identitas dari pintu, aktifitas yang harus dilakukan serta kode kunci dari pintu tersebut. Sebagai tabahan untuk pencatatan riwayat aktifitas maka ditambahkan identitas aktor yang melakukan request ke API tersebut.

1. API Door Login

API door login digunakan oleh setiap perangkat kunci pintu untuk mendapatkan token akses. Setiap pintu akan mengirimkan username dan password yang dimiliki untuk mendapatkan token akses. Hasil dari API door login dapat dilihat pada gambar x.x dibawah.

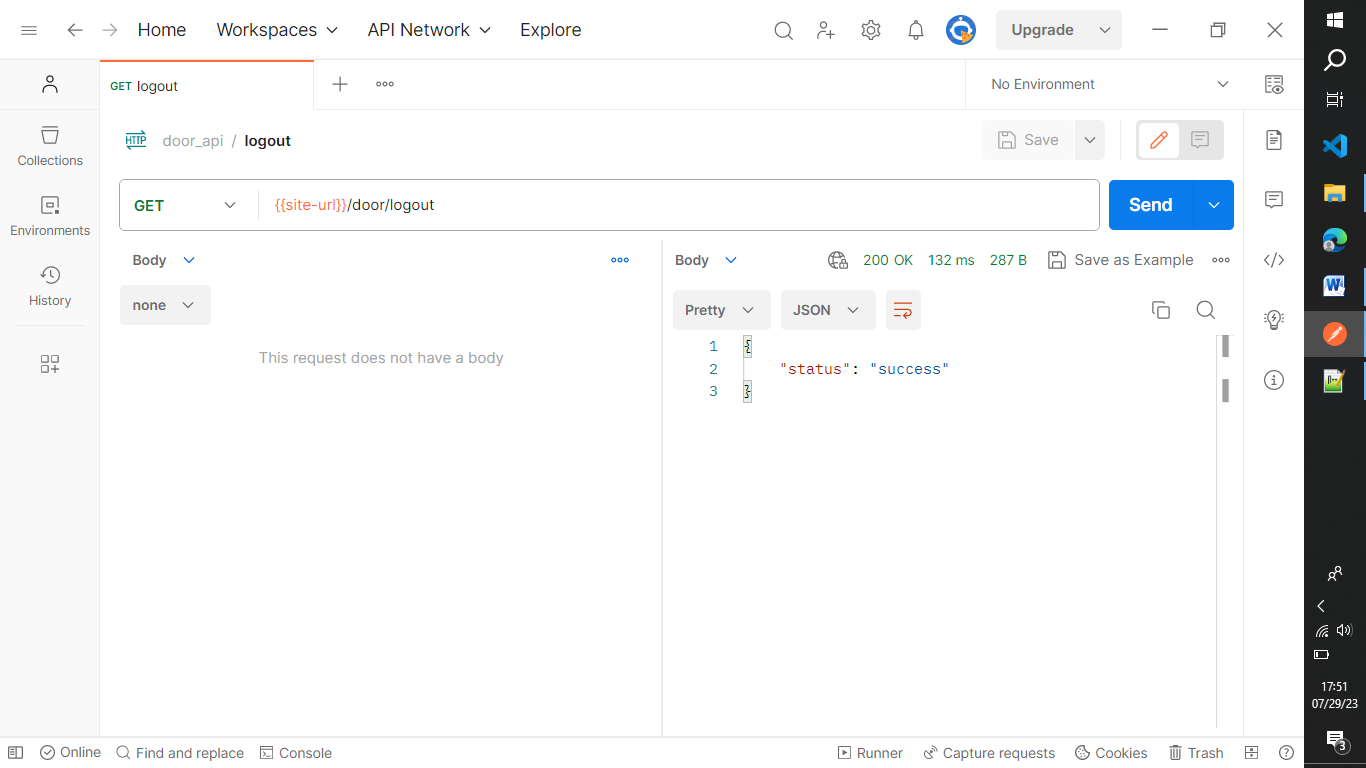


Hasil API Door Login

Gambar x.x diatas merupakan hasil dari API door login yang dikirimkan melalui endpoint “/door/login”. Setiap pintu akan mengirimkan device\_name sebagai username dan device\_pass sebagai password untuk mendapatkan token akses. Jika login berhasil maka PAI akan memberikan respon berupa token akses disertai dengan data pendukung seperti identitas gedung dan identitas pintu.

1. API Door Logout

API door logout digunakan untuk menghapus semua token yang dimiliki oleh pintu tersebut. Sesekali pintu akan melakukan permintaan logout untuk menghapus semua token yang disimpan didalam database. Hasil dari API door logout dapad dilihat pada gambar x.x dibawah.

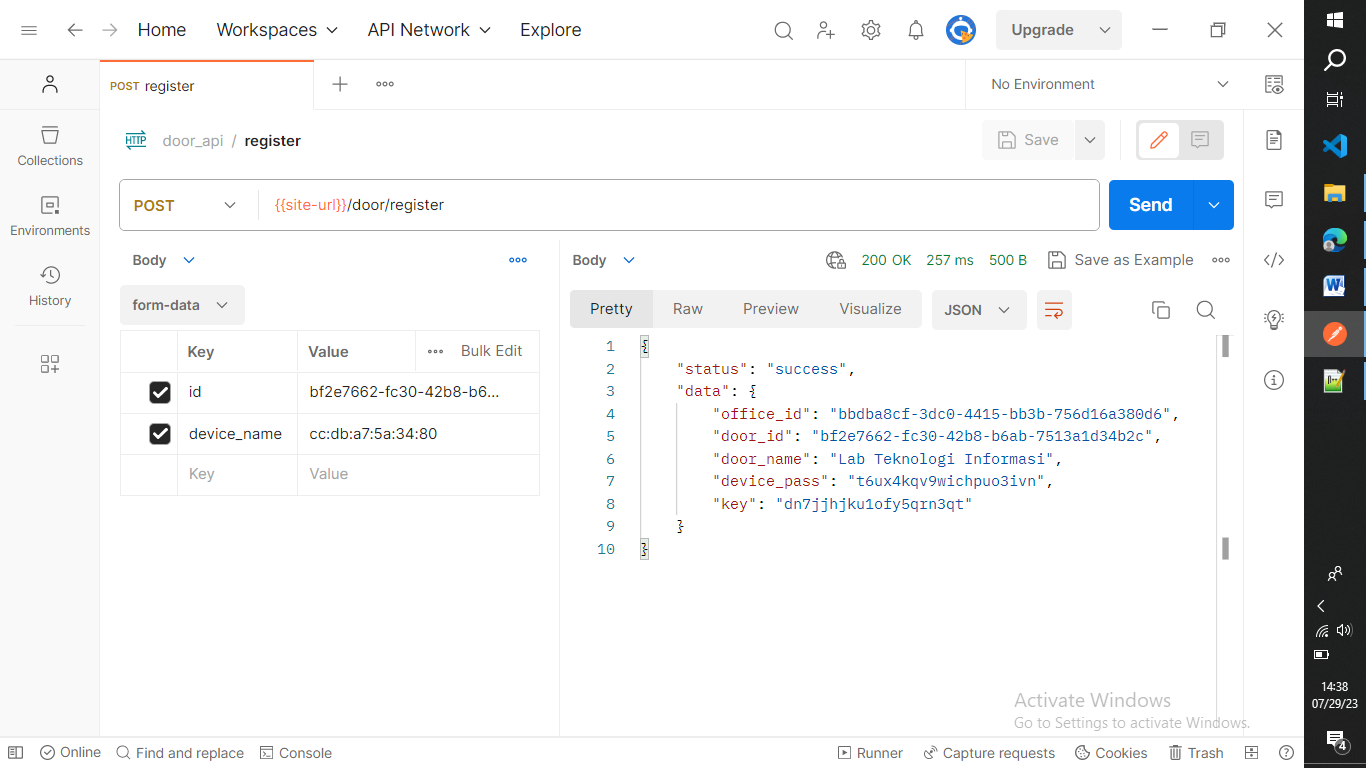


Hasil API Door Logout

Gamabr x.x diatam merupakan hasil dari API door logout yang dikirimkan melalui endpint “/door/logout”. Pintu akan mengirimkan permintaan dengan melampirkan token akses yang digunakan untuk autentikasi, jika autentikasi berhasil maka API akan memberikan respon berhasil atau sukses.

1. API Door Register

API door register digunakan oleh perangkat kunci pintu untuk mendaftarkan perangkat ke server sehingga perangkat tersebut dapat dikenali dan dapat diatur konerjanya oleh server. Hasil dari API door login dapat dilihat pada gambar x.x dibawah.

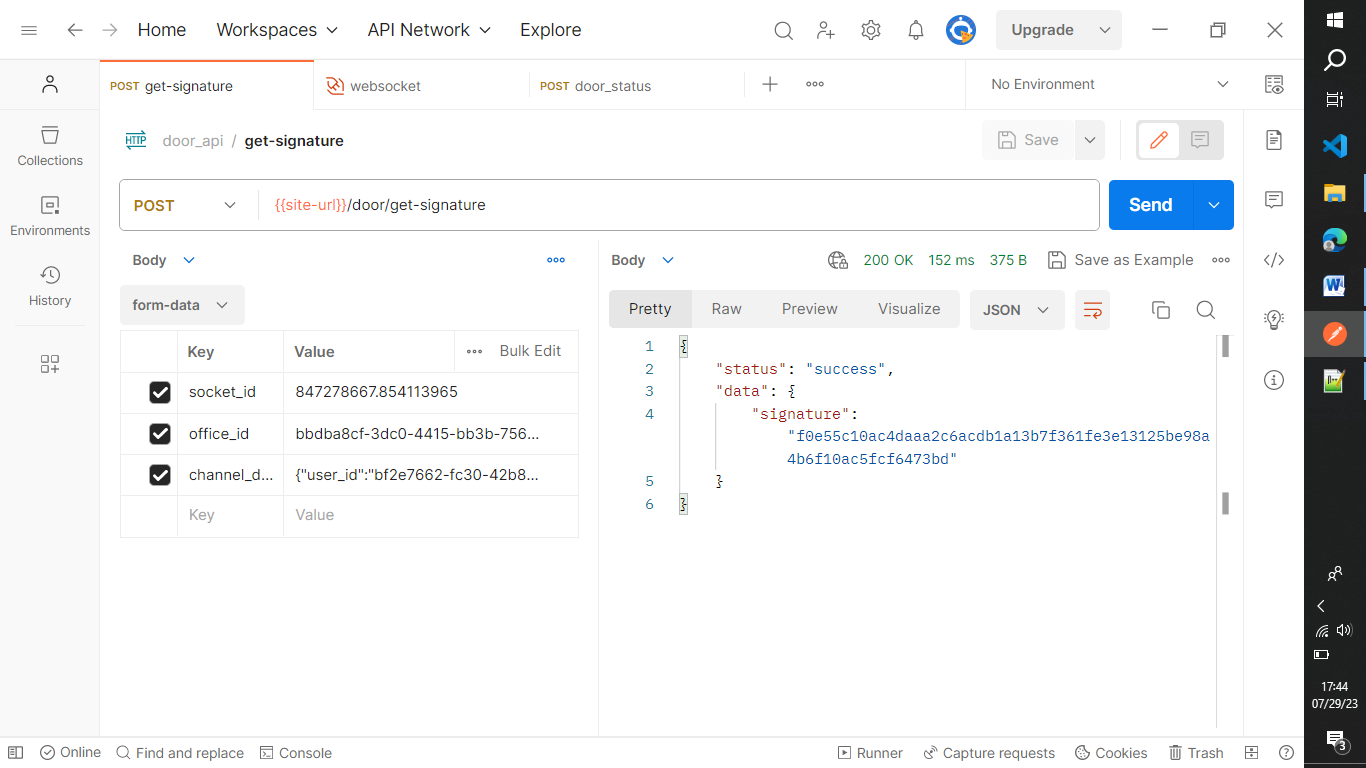


Hasil API Door Register

Gambar x.x diatas merupaka hasil dari API door login yang dikirmkan melalui endpoint “/door/register”. Perangkat kunci pintu akan mengirimkan identitas unik yang nantinya akan digunakan sebagai nama perangkat yang dikenali oleh server. Jika prosese pendaftaran berhasil maka API akan memberikan respon sukse dan mengirimkan data seperti identitas pintu, identitas gedung dan juga password untuk melakukan proses login.

1. API Door Signature

API door signature digunakan oleh perangkat kunci pintu untuk mendapatkan kunci websocket. Komunikasi antara server ke perangkat kunci pintu dilakukan menggunakan websocket, dengan menggunakan websocket maka server dapat langsung memberikan perintah ke perangkat kunci pintu tanpa harus menunggu polling yang dilakukan oleh perangkat kuni pintu. Untuk mengamankan jalur komunikasi websocket ini menggunakan kunci signature yang harus dikirimkan oleh perangkat kunci untuk bisa terhubung dan menerima perintah melalui koneksi websokcet. Haisl dari API door signature dapat dilihat pada gambar x.x dibawah.

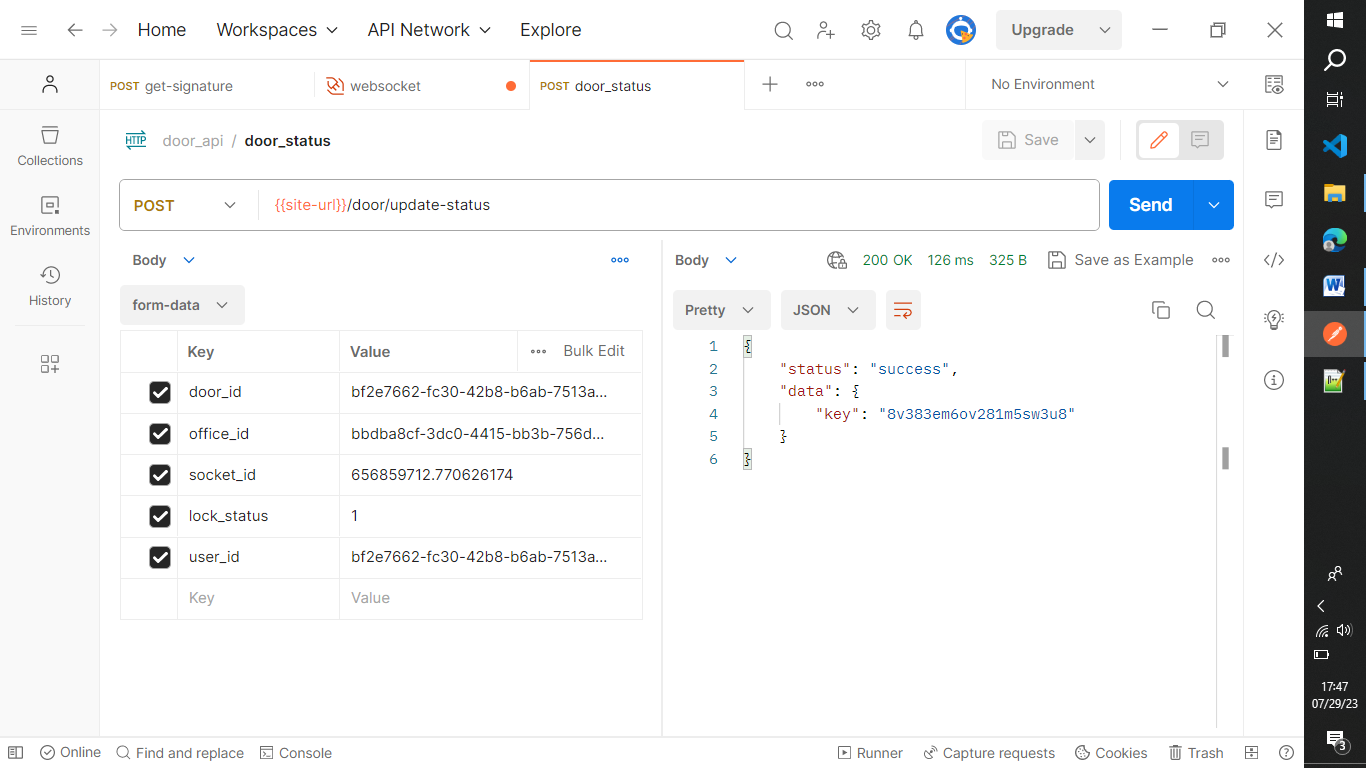


Hasil API Door Signature

Gambar x.x diatas merupakan hasil dari API door signature yang dikirimkan melalui endpoint “/api/signature”. Parangkat kunci pintu akan mengirimkan identitas socket yang sudah terjalin disertai dengan tada pendukung seperti identitas gedung, identitas pintu dan tentunya token akses, kemudian server akan membuat kunci signature yang dapat digunakan untuk terhubung ke channel websocket.

1. API Door Update Status

API door update status digunakan oleh perangkat kunci pintu untuk memperbarui status penguncian pintu, jika ada perubaha status pintu seperti terkunci atau terbuka maka perangkat kunci pintu akan mengirimkan status tersebut ke server. Proses update status ini akan memastikan adanya perubahan kondisi pada pintu sehingga jika ada perintah yang dikirimkan ke perangkat kunci pintu maka perangkat kunci pintu hanya akan memberikan respon jika perubahan benar benar terjadi dan memastikan bahwa perintah sudah benar-benar dikerjakan. Hasil dari API update status dapat dilihat pada gambar x.x dibawah.

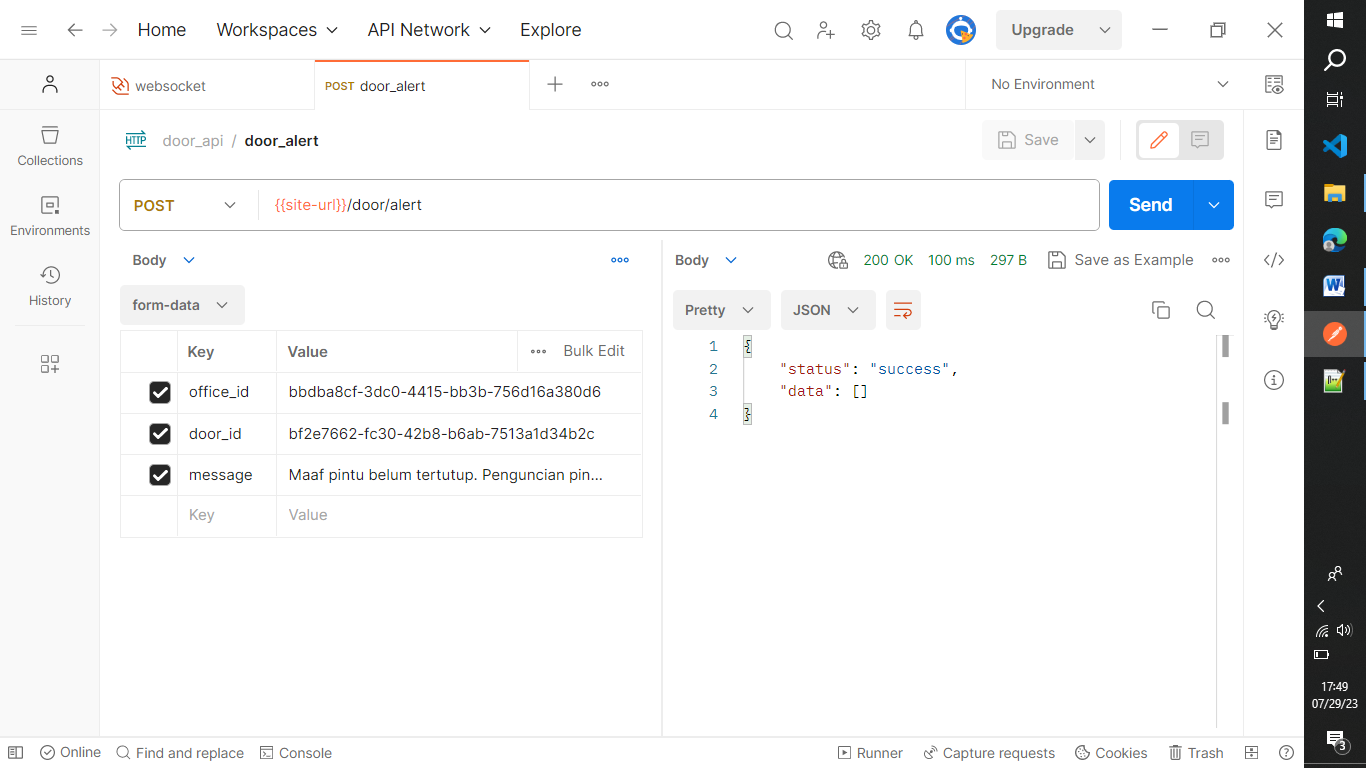


Hasil API Update Status

Gambar x.x diatas merupak hasil dari API update status yang dikirmkan melalui endpoint “/door/update-status”. Perangkat kunci pintu akan mengirimakan data pendukung seperti identitas pintu, identitas gedung, identitas socket, status penguncian dan identitas aktor yang melakukan kegiatan tersebut. Jika data yang dikirmkan sesuai maka server akan mengembalikan respon status sukses yang mengirimkan kunci pintu baru. Setiap melakukan update status maka server akan memberikan kunci pintu baru sehingga kunci pada setiap pintu bersifat dinamis, hal ini dimaksudkan untuk memperkuat keamana sistem penguncian dengan menggunakan kode kunci pintu yang berubah ubah untuk melakukan autentikasi.

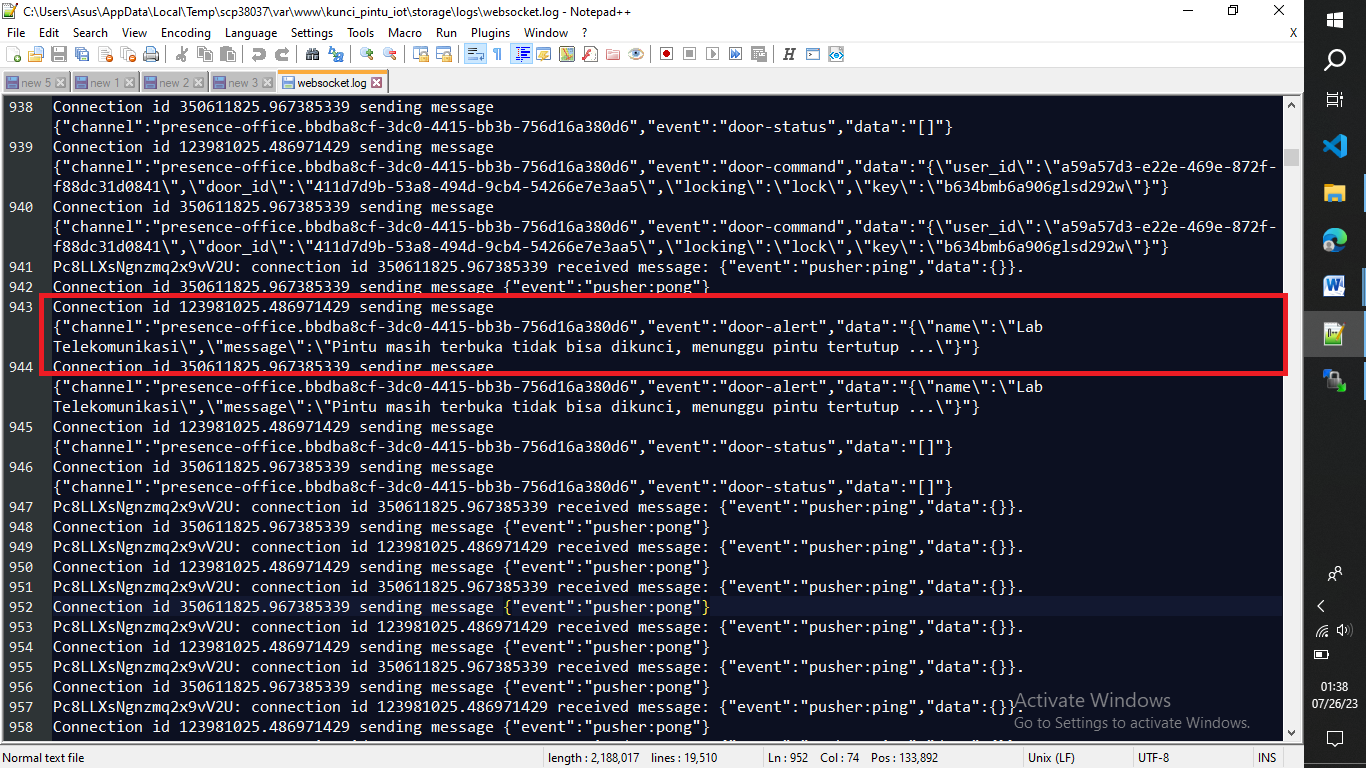
1. API Door Alert

API door alert digunakan oleh perangkat kunci pintu untuk mengirimkan status peringatan ke server, peringatan yang dikirmkan dapat berupa pintu yang terbuka tanpa autentikasi yang sah atau juaga pintu dalam koneisi terbuka sehingga menghalangi perintah penguncian untk dijalankan. Hasil dari API door alert dapat dilihat pada gambar x.x dibawah.



Hasil API Door Alert

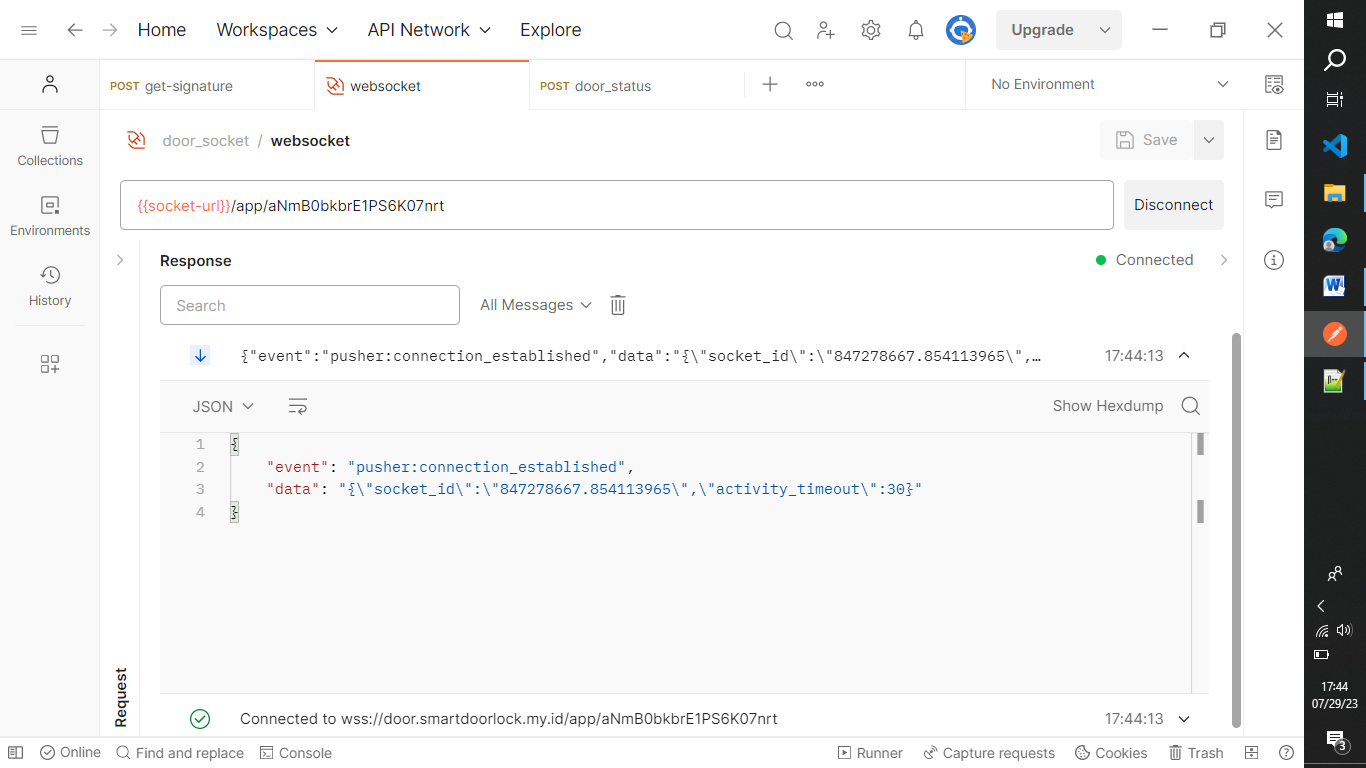
Gambar x.x merupakan hasil dari API door alert yang dikirimkan melalui endpoint “/door/alert”. Perangkat kunci pintu akan mengirimkan identitas pintu, identitas gedung dan pesan peringatan. Jika data yang dikirimkan sesuai maka server akan memberikan respon status sukses dan mengirimkan peringatan ke operator melalui koneksi websocket seperti yang terlihat pada gambar x.x dibawah.

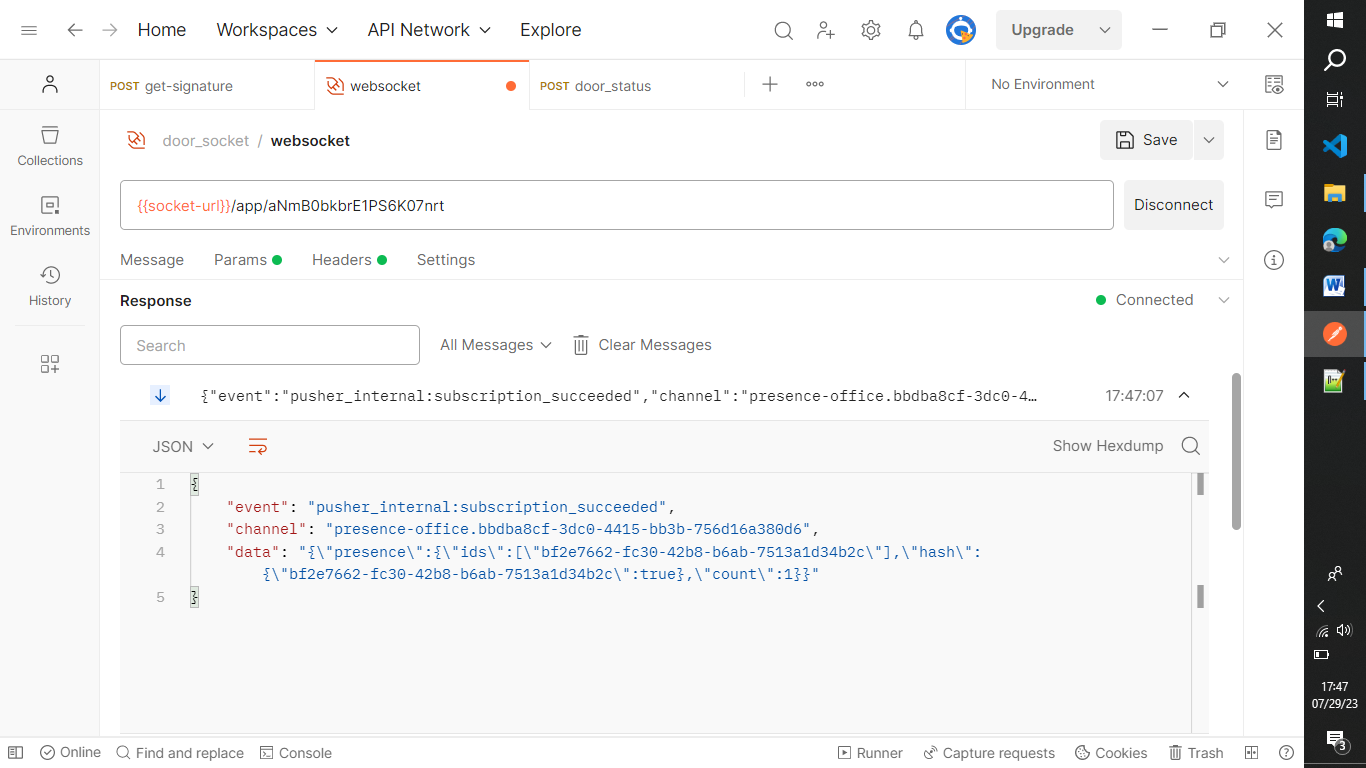


Aktifitas Door Alert

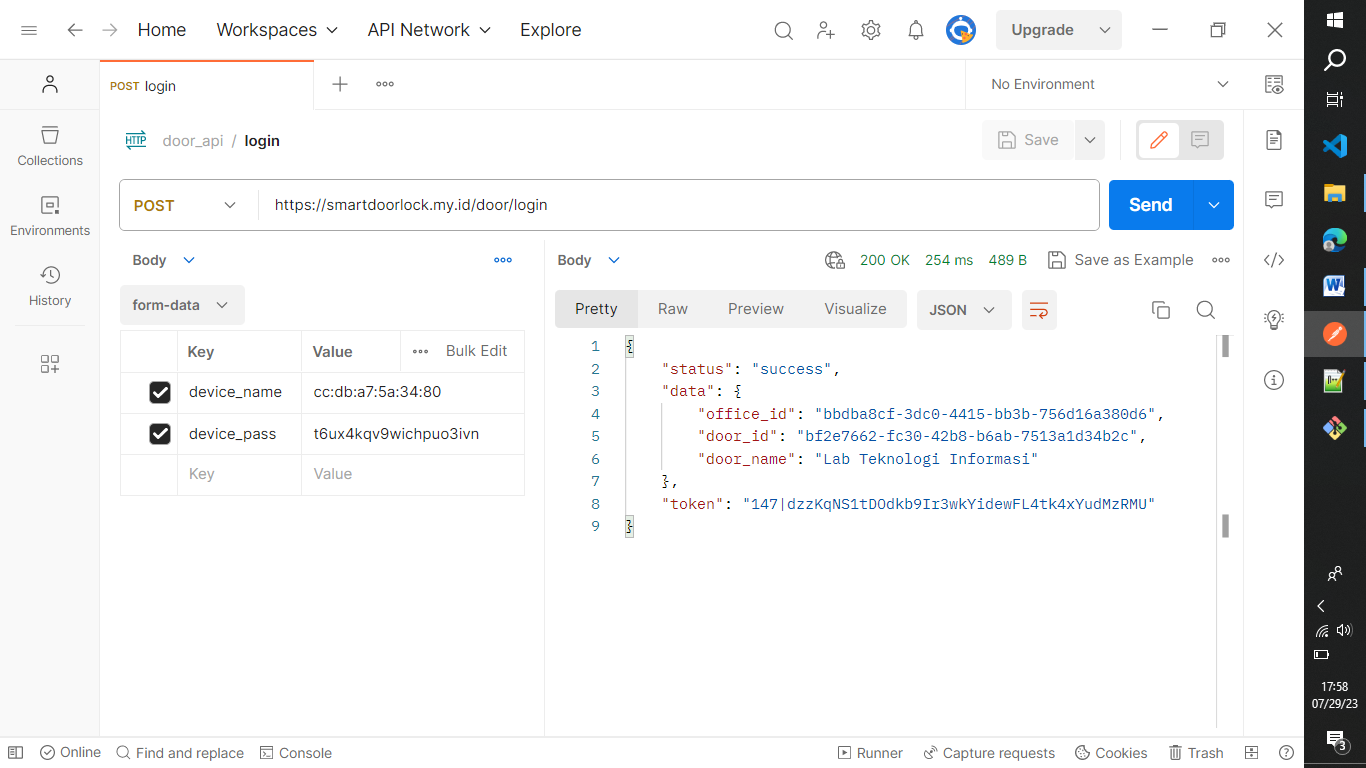
Gambar x.x diatas merupakan aktifitas dari peringatan pintu pada koneksi websocket. Setiap ada peringatan yang dikirmkan oleh perangkat kunci pintu maka server akan mengirimkan ke operator baik website maupun aplikasi mobile melalui koneksi websocket sehingg aperngatan dapat diterima oleh operator secara realtime.

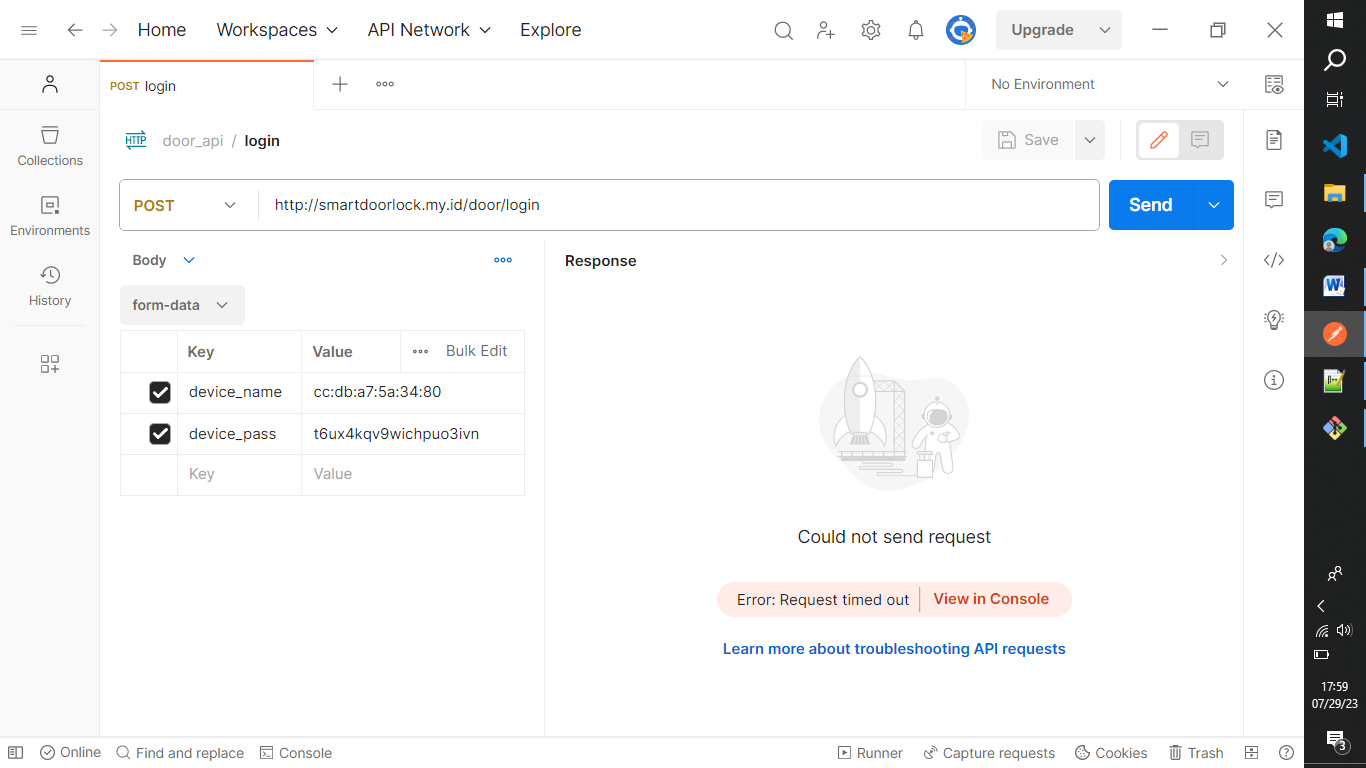
1. Pengecekan Jadwal
2. Atur Ulang Jadwal
3. Komunikasi Websocket





1. Koneksi HTTPS





1. Pengujian

BAB V

**PENUTUP**

1. Kesimpulan

Berdasarkan hasil pengerjaan dan analisis yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut :

1. Saran

Setelah dilakukan analisan, perancangan dan pengujian didapatkan beberapa saran antara lain :

**DAFTAR PUSTAKA**

**BIODATA MAHASISWA**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nama | : | Muhammad Khoiril Wafi |
| NIM | : | 21060119140133 |
| Konsentrasi | : | Teknologi Informasi |
| Tempat Tanggal Lahir | : | Demak, 4 Maret 2001 |
| Alamat Sekarang | : | Jl. Baskoro Raya No. 61, RT.03 / RW.07, Tembalang, Kec. Tembalang, Kota Semarang, Jawa Tengah 50275 |
| Alamat E-mail | : | [khoirilwafi123@gmail.com](mailto:khoirilwafi123@gmail.com) |
| Nama Orang Tua | : | Muarifin |
| Alamat Orang Tua | : | Jl. Putrajaya No. 06 02/06 Desa Karangrejo Kec. Bonang Kab. Demak |
| IPK | : | - |

**Pengalaman dan Prestasi yang Pernah Diraih:**

|  |  |
| --- | --- |
| **Nama pengalaman / prestasi** | **Tahun** |
|  |  |
|  |  |
|  |  |

Semarang, 23 Juni 2023

(Muhammad Khoiril Wafi)

NIM. 21060119140133

**LAMPIRAN A**

**SENARAI PROGRAM**

**LAMPIRAN B**

**DATA TRAINNING**

**LAMPIRAN C**

**MAKALAH TUGAS AKHIR**