**LAPORAN PRAKTIK KERJA LAPANGAN (PKL)**

**PERUSAHAAN/INDUSTRI**

**IMPLEMENTASI SISTEM TERDISTRIBUSI BERBASIS *MICROSERVICE* PADA LAYANAN E-MAKAM PEMERINTAH KOTA MALANG**

**Diajukan untuk memenuhi sebagian persyaratan Kurikulum Sarjana**



Disusun oleh :

1. **Firman Budi Safrizal 155150201111147**
2. **Fadhlan Muhammad 155150201111108**
3. **Dian Zaki Yudhono 155150200111085**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2018**

# PENGESAHAN

LAPORAN PRAKTIK KERJA LAPANGAN (PKL)

PERUSAHAAN/INDUSTRI

DINAS KOMUNIKASI DAN INFORMATIKA KOTA MALANG

IMPLEMENTASI SISTEM TERDISTRIBUSI BERBASIS *MICROSERVICE* PADA LAYANAN E-MAKAM PEMERINTAH KOTA MALANG

Diajukan untuk memenuhi sebagian persyaratan Kurikulum Sarjana

Program Studi Teknik Informatika

Bidang Komputasi Berbasis Jaringan

Disusun Oleh:

**Firman Budi Safrizal 155150201111147**

**Fadhlan Muhammad 155150201111108**

**Dian Zaki Yudhono 155150200111085**

Praktik Kerja Lapangan ini dilaksanakan pada

25 Juni sampai dengan 19 September 2018

Telah diperiksa dan disetujui oleh:

Mengetahui, Menyetujui,

Ketua Jurusan Teknik Informatika Dosen Pembimbing PKL

Tri Astoto Kurniawan, S.T, M.T, Ph.D Adhitya Bhawiyuga, S.Kom, M.Sc

NIP: 197105182003121001 NIP: 198907202018031002

# KATA PENGANTAR

Assalamu’alaikum Wr. Wb.

Dengan selalu memanjatkan puji syukur kehadirat Tuhan Yang Maha Esa, berkat limpahan-Nya sehingga kami dapat menyelesaikan laporan kegiatan Praktik Kerja Lapangan (PKL) yang dilaksanakan di Kantor Dinas Komunikasi dan Informatika Kota Malang selama 3 bulan dengan baik.

Dalam pelaksanaan PKL dan penyusun laporan ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, kami mengucapkan terima kasih kepada :

1. Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya kami mampu menyelesaikan PKL dan laporan kegiatan ini dengan baik.
2. Orang tua kami dan keluarga besar serta saudara-saudara yang selalu memberi dukungan moral dan materi.
3. Agus Wahyu Widodo, S.T, M.Cs selaku Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Adhitya Bhawiyuga, S.Kom, M.Sc selaku dosen pembimbing PKL.
6. Dinas Komunikasi dan Informatika selaku tempat kami melaksanakan PKL.

Penulis menerima kritik dan saran yang membangun agar dalam pelaksanaan penyusunan penyusunan laporan Praktik Kerja Lapangan pada masa yang akan datang kami dapat melaksanakan dan menghasilkan penelitian yang lebih sempurna, penulis berharap agar hasil laporan Praktik Kerja Lapangan ini berguna bagi semua pihak.

Malang, 8 September 2018

Penulis

# DAFTAR ISI

[PENGESAHAN i](#_Toc531864907)

[KATA PENGANTAR ii](#_Toc531864908)

[DAFTAR ISI iii](#_Toc531864909)

[DAFTAR GAMBAR vii](#_Toc531864910)

[DAFTAR TABEL ix](#_Toc531864911)

[BAB 1 PENDAHULUAN 1](#_Toc531864912)

[1.1 Latar Belakang 1](#_Toc531864913)

[1.2 Waktu dan Lokasi PKL 2](#_Toc531864914)

[1.3 Rumusan Masalah 2](#_Toc531864916)

[1.4 Tujuan 2](#_Toc531864917)

[1.5 Manfaat 3](#_Toc531864918)

[1.6 Batasan Masalah 3](#_Toc531864919)

[1.7 Sistematika Pembahasan 4](#_Toc531864920)

[BAB 2 GAMBARAN UMUM PERUSAHAAN 5](#_Toc531864921)

[2.1 Sejarah Dinas Komunikasi dan Informatika 5](#_Toc531864922)

[2.2 Visi, Misi dan Tujuan Dinas Komunikasi dan Informatika 6](#_Toc531864923)

[2.2.1 Visi 6](#_Toc531864924)

[2.2.2 Misi 6](#_Toc531864925)

[2.2.3 Tujuan 7](#_Toc531864926)

[2.3 Logo Dinas Komunikasi dan Informatika Kota Malang 7](#_Toc531864927)

[2.4 Struktur Organisasi Dinas Komunikasi dan Informatika Kota Malang 8](#_Toc531864928)

[BAB 3 DASAR TEORI 9](#_Toc531864929)

[3.1 Sistem Terdistribusi 9](#_Toc531864930)

[3.2 Arsitektur *Microservice* 9](#_Toc531864931)

[3.2.1 Kinerja *Microservice* 10](#_Toc531864932)

[3.2.2 Pengembangan *Microservice* 10](#_Toc531864933)

[3.3 *Webservice* 10](#_Toc531864934)

[3.3.1 REST (*Representational State Transfer*) 11](#_Toc531864935)

[3.3.2 SOAP 11](#_Toc531864936)

[3.3.3 WSDL 11](#_Toc531864937)

[3.4 *Gateway* API 11](#_Toc531864938)

[3.5 Laravel 12](#_Toc531864939)

[3.6 *ReactJS* 12](#_Toc531864940)

[3.7 PHP Hypertext Preprocessor (PHP) 12](#_Toc531864941)

[BAB 4 METODOLOGI 13](#_Toc531864942)

[4.1 Studi Literatur 14](#_Toc531864943)

[4.2 Analisis Kebutuhan 14](#_Toc531864944)

[4.3 Perancangan 14](#_Toc531864945)

[4.4 Implementasi 14](#_Toc531864946)

[4.5 Pengujian 15](#_Toc531864947)

[4.6 Kesimpulan 15](#_Toc531864948)

[BAB 5 PEMBAHASAN DAN HASIL 16](#_Toc531864949)

[5.1 Analisis Kebutuhan 16](#_Toc531864950)

[5.1.1 Deskripsi Umum 16](#_Toc531864951)

[5.1.2 Diagram Alur Sistem 16](#_Toc531864952)

[5.1.3 Aktor Sistem 18](#_Toc531864953)

[5.1.4 Kebutuhan Fungsional 18](#_Toc531864954)

[5.1.5 Kebutuhan Non-Fungsional 22](#_Toc531864961)

[5.2 Perancangan Sistem 23](#_Toc531864964)

[5.2.1 Distribusi Sistem 23](#_Toc531864965)

[5.2.2 Perancangan *Service* 23](#_Toc531864966)

[5.3 Perancangan Database 24](#_Toc531864967)

[5.4 Implementasi 25](#_Toc531864968)

[5.4.1 Implementasi Database 25](#_Toc531864969)

[5.4.2 Implementasi API *Gateway* 38](#_Toc531864970)

[5.4.3 Implementasi Fitur Manajemen TPU 41](#_Toc531864971)

[5.4.4 Implementasi Fitur Manajemen *User* 43](#_Toc531864972)

[5.4.5 Implementasi Fitur Manajemen *User* Access 45](#_Toc531864973)

[5.4.6 Implementasi Fitur Manajemen Kecamatan Access 46](#_Toc531864974)

[5.4.7 Implementasi FItur Registrasi Perijinan 47](#_Toc531864975)

[5.4.8 Implementasi Fitur Dokumen 48](#_Toc531864976)

[5.4.9 Implementasi Fitur Manajemen Makam 56](#_Toc531864977)

[5.4.10 Implementasi Fitur Manajemen Blok Makam 57](#_Toc531864978)

[5.4.11 Implementasi Fitur Manajemen Penghuni Makam 58](#_Toc531864979)

[5.4.12 Implementasi Fitur *Authentication* 60](#_Toc531864980)

[5.4.13 Implementasi Mail *Service* 61](#_Toc531864981)

[5.4.14 Persiapan Deployment Layanan pada Server 63](#_Toc531864982)

[5.4.15 Deployment Layanan pada Server 66](#_Toc531864983)

[5.5 Lingkungan Pengujian 70](#_Toc531864984)

[5.5.1 Parameter Pengujian 70](#_Toc531864985)

[5.5.2 Skenario Pengujian 70](#_Toc531864986)

[5.5.3 Pengujian Fungsionalitas 71](#_Toc531864987)

[5.5.4 Pengujian Kinerja 81](#_Toc531864998)

[5.5.5 Pengujian Skalabilitas 84](#_Toc531865010)

[BAB 6 PENUTUP 87](#_Toc531865011)

[6.1 Kesimpulan 87](#_Toc531865012)

[6.2 Saran 88](#_Toc531865013)

[DAFTAR PUSTAKA 88](#_Toc531865014)

# DAFTAR GAMBAR

[Gambar 2.1 Logo Dinas Komunikasi dan Informatika Kota Malang 7](#_Toc531865072)

[Gambar 2.2 Struktur Organisasi Dinas Komunikasi dan Informatika Kota Malang 8](#_Toc531865073)

[Gambar 4.1 Diagram Alir Penelitian 13](file:///D:\github-pkl\projectpkl_emakam\IMPLEMENTASI%20SISTEM%20TERDISTRIBUSI%20BERBASIS%20MICROSERVICE%20PADA%20LAYANAN%20E-MAKAM%20PEMERINTAH%20KOTA%20MALANG%20REV3.0.docx#_Toc531865074)

[Gambar 5.1 Diagram Perancangan Sistem Terdistribusi 23](#_Toc531865075)

[Gambar 5.2 Gambar Update Sistem Operasi Ubuntu 18.04 Server 63](#_Toc531865076)

[Gambar 5.3 Gambar Upgrade Sistem Operasi Ubuntu 18.04 Server 63](#_Toc531865077)

[Gambar 5.4 Instalasi apache2 64](#_Toc531865078)

[Gambar 5.5 Cek Status Apache2 64](#_Toc531865079)

[Gambar 5.6 Instalasi MySQL-Server 65](#_Toc531865080)

[Gambar 5.7 Cek Status MySQL Server 65](#_Toc531865081)

[Gambar 5.8 Instalasi PHP7.2 66](#_Toc531865082)

[Gambar 5.9 Membuat Database E-makam 67](#_Toc531865083)

[Gambar 5.10 Import Database E-makam 67](#_Toc531865084)

[Gambar 5.11 Copy Direktori API Emakam 68](#_Toc531865085)

[Gambar 5.12 Ubah Hak akses dan grup Direktori 68](#_Toc531865086)

[Gambar 5.13 a2dissite konfigurasi 69](#_Toc531865087)

[Gambar 5.14 Menghapus DIrektori Storage 70](#_Toc531865088)

[Gambar 5.15 Membuat Link Storage 70](#_Toc531865089)

[Gambar 5.16 aplikasi pembuatan TPU 71](#_Toc531865090)

[Gambar 5.17 aplikasi setelah pembuatan TPU 72](#_Toc531865091)

[Gambar 5.18 aplikasi pembuatan User 72](#_Toc531865092)

[Gambar 5.19 aplikasi setelah pembuatan User 73](#_Toc531865093)

[Gambar 5.20 aplikasi create user access 73](#_Toc531865094)

[Gambar 5.21 aplikasi setelah create user access 74](#_Toc531865095)

[Gambar 5.22 aplikasi create kecamatan access 74](#_Toc531865096)

[Gambar 5.23 aplikasi setelah create kecamatan access 75](#_Toc531865097)

[Gambar 5.24 aplikasi form registrasi 75](#_Toc531865098)

[Gambar 5.25 aplikasi trace dokumen 76](#_Toc531865099)

[Gambar 5.26 aplikasi cetak dokumen 76](#_Toc531865100)

[Gambar 5.27 hasil dokumen yang di generate cetak dokumen 77](#_Toc531865101)

[Gambar 5.28 aplikasi create blokmakam 77](#_Toc531865102)

[Gambar 5.29 aplikasi setelah create blokmakam 78](#_Toc531865103)

[Gambar 5.30 aplikasi create makam 78](#_Toc531865104)

[Gambar 5.31 aplikasi makam setelah create makam 79](#_Toc531865105)

[Gambar 5.32 aplikasi penghuni makam 79](#_Toc531865106)

[Gambar 5.33 hasil data penghuni makam setelah create 80](#_Toc531865107)

[Gambar 5.34 aplikasi sebelum signin 80](#_Toc531865108)

[Gambar 5.35 aplikasi setelah signin sebagai admin 81](#_Toc531865109)

[Gambar 5.36 Hasil Pengujian Fitur Manajemen TPU 81](#_Toc531865110)

[Gambar 5.37 Hasil Pengujian Fitur Manajemen User 81](#_Toc531865111)

[Gambar 5.38 Hasil Pengujian Fitur Manajemen User Access 82](#_Toc531865112)

[Gambar 5.39 Hasil Pengujian Service Kecamatan 82](#_Toc531865113)

[Gambar 5.40 Hasil Pengujian FItur Registrasi Perijinan 82](#_Toc531865114)

[Gambar 5.41 Hasil Pengujian Fitur Dokumen 82](#_Toc531865115)

[Gambar 5.42 Hasil Pengujian Fitur Manajemen Makam 83](#_Toc531865116)

[Gambar 5.43 Hasil Pengujian Fitur Manajemen Blok Makam 83](#_Toc531865117)

[Gambar 5.44 Hasil Pengujian Fitur Manajemen Penghuni Makam 83](#_Toc531865118)

[Gambar 5.45 Hasil Pengujian Fitur Authentication 83](#_Toc531865119)

[Gambar 5.46 Hasil Pengujian Service Mail 84](#_Toc531865120)

[Gambar 5.47 Hasil Pengujian Skalabilitas Skenario 1 84](#_Toc531865121)

[Gambar 5.48 Hasil Pengujian Skalabilitas Skenario 2 84](#_Toc531865122)

[Gambar 5.49 Hasil Pengujian Skalabilitas Skenario 3 85](#_Toc531865123)

[Gambar 5.50 Hasil Pengujian Skalabilitas Skenario 4 85](#_Toc531865124)

[Gambar 5.51 Hasil Pengujian Skalabilitas Skenario 5 85](#_Toc531865125)

[Gambar 5.52 Hasil Pengujian Skalabilitas Skenario 6 85](#_Toc531865126)

[Gambar 5.53 Grafik Error Rate 86](#_Toc531865127)

[Gambar 5.54 Grafik Troughput Rate 86](#_Toc531865128)

# DAFTAR TABEL

[Tabel 5.1 Diagram Alur Kebutuhan 16](#_Toc531865129)

[Tabel 5.2 Tabel Kebutuhan Fungsional Admin Pusat 18](#_Toc531865130)

[Tabel 5.3 Tabel Kebutuhan Fungsional Admin TPU 19](#_Toc531865131)

[Tabel 5.4 Tabel Kebutuhan Fungsional Kepala UPT Pemakaman 21](#_Toc531865132)

[Tabel 5.5 Tabel Kebutuhan Fungsional Kepala Dinas 21](#_Toc531865133)

[Tabel 5.6 Tabel Kebutuhan Fungsional Kepala Kecamatan 21](#_Toc531865134)

[Tabel 5.7 Tabel Kebutuhan Fungsional Admin Kecamatan 22](#_Toc531865135)

[Tabel 5.8 Tabel Kebutuhan Non-Fungsional 22](#_Toc531865136)

[Tabel 5.9 Implementasi Database 25](#_Toc531865137)

[Tabel 5.10 Implementasi API Gateway 38](#_Toc531865138)

[Tabel 5.11 Implementasi Fitur Manajemen TPU 41](#_Toc531865139)

[Tabel 5.12 Implementasi Fitur Manajemen User 43](#_Toc531865140)

[Tabel 5.12 Implementasi Fitur Manajemen User Access 45](#_Toc531865141)

[Tabel 5.13 Implementasi Fitur Manajemen Kecamatan Access 46](#_Toc531865142)

[Tabel 5.14 Implementasi FItur Registrasi Perijinan 47](#_Toc531865143)

[Tabel 5.15 Implementasi Fitur Dokumen 49](#_Toc531865144)

[Tabel 5.16 Implementasi Fitur Manajemen Makam 56](#_Toc531865145)

[Tabel 5.17 Implementasi Fitur Manajemen Blok Makam 57](#_Toc531865146)

[Tabel 5.18 Implementasi Fitur Manajemen Penghuni Makam 59](#_Toc531865147)

[Tabel 5.19 Implementasi Fitur Authentication 60](#_Toc531865148)

[Tabel 5.20 Implementasi Mail Service 61](#_Toc531865149)

[Tabel 5.21 Implementasi Mail Notification 62](#_Toc531865150)

# PENDAHULUAN

## Latar Belakang

Dinas Komunikasi dan Informatika Kota Malang adalah instansi negara yang memiliki tugas untuk menyusun dan melaksanakan kebijakan pemerintahan dalam bidang informatika. Dinas komunikasi dan informatika kota Malang dibentuk ini dibentuk untuk memenuhi ketentuan peraturan pemerintah nomor 41 tahun 2007 tentang organisasi perangkat daerah serta memperhatikan kebutuhan pelayanan masyarakat sebagai hasil analisa tim analisis jabatan dan pengukuran beban kerja dalam rangka pengembangan dan pengelolaan teknologi informasi dan komunikasi.

Dinas Komunikasi dan informatika memiliki fungsi yaitu meningkatkan pelayanan publik. Salah satu pelayanan publik yang perlu diperhatikan adalah pelayanan perizinan makam. Berdasarkan hasil survei oleh BPS Malang nilai indeks kepuasan pelayanan publik memiliki indeks sangat baik, akan tetapi memiliki jarak yang signifikan antara nilai indeks kepuasan dan nilai indeks harapan (Pemkot 2015). Sebagai salah satu contoh pelayanan masyarakat adalah perizinan makam. Pada saat ini perizinan makam dilakukan secara manual oleh ahli waris dengan membuat form perizinan ke upt pemakaman. Dan kemudian melanjutkan permohonan ke dinas pemerintahan sekitar dan kecamatan. Seluruh dokumen masih berbentuk hardcopy dan manajemen datanya masih sangat manual. Penyelesaian pelayanan tersebut membutuhkan waktu 4 hari kerja. Oleh karena itu sistem pelayanan pendataan dan perizinan dinilai kurang efektif dan efisien waktu.

Solusi untuk permasalahan pelayanan perizinan makam adalah sebuah sistem E-makam. E-makam adalah sebuah sistem berbasis web ditujukan untuk membuat proses perizinan makam lebih sederhana. Penulisan form perizinan hanya cukup sekali dan sisanya merupakan data scan. Proses permohonan ke dinas pemerintahan dan kecamatan juga lebih sederhana dengan click sebuah tombol maka surat permohonan dan perizinan akan di generate lalu diteruskan ke tahap berikutnya secara otomatis. *Maintain* data tpu, blok makam, makam serta penghuni makam juga akan menjadi lebih *traceable* karena datanya berbentuk digital.

Sistem E-makam adalah sebuah sistem yang cukup kompleks jika dilihat dari berapa banyak fungsi yang di butuhkan. Implementasinya kemungkinan besar pasti akan dikerjakan dalam tim. Pengerjaan tim akan memungkinkan dimana setiap *developer* fungsi menggunakan Bahasa pemrograman yang berbeda-beda. Dan juga dalam proses pengembangannya juga akan bersifat kontinyu jika suatu saat diperlukan sebuah fungsi tambahan (Vimalizar et al., 2015).

Penulis membuat sebuah sistem E-makam dengan arsitektur micro *service*. Arsitektur micro *service* menstrukturkan aplikasi kedalam beberapa kumpulan *service*-*service* kecil. Hal ini ditujukan untuk mempermudah *developer* untuk membuat sistem yang bekerja sebagai tim dalam sebuah projek berskala besar dan bersifat kontinyu. Dengan dipecahnya layanan menjadi blok *service* yang lebih kecil, deployment dapat berjalan terpisah serta waktu deployment juga semakin berkurang. Sistem E-makam kami akan menerima *request* dari *frontend* dalam bentuk URL dan akan di *routing* berdasarkan fungsi spesifiknya masing-masing.

## Waktu dan Lokasi PKL

Pelaksanaan PKL ini dilaksanakan pada :

|  |  |  |
| --- | --- | --- |
| Tanggal | : | 25 Juni 2018 – 19 September 2018 |
| Tempat | : | Dinas Komunikasi dan Informatika Kota Malang |
| Alamat | : | Perkantoran Terpadu, Jl. Mayjend Sungkono Kel. Arjowinangun, Kec Kedungkandang, Kota Malang |
| Waktu | : | Senin – Jum’at 08.00 – 16.00 |
| Website | : | kominfo.malangkota.go.id |

## Rumusan Masalah

Berdasarkan latar belakang yang ada dapat disimpulkan rumusan masalah :

1. Bagaimana rancangan arsitektur berbasis micro *service* pada sistem E-makam?
2. Bagaimana kinerja *Microservice* pada layanan E-makam?

## Tujuan

Maksud dari diadakannya praktik kerja lapangan pada Fakultas Ilmu Komputer program studi Teknik Informatika Universitas Brawijaya Malang memiliki beberapa tujuan yaitu sebagai berikut.

Tujuan Umum:

1. Membekali mahasiswa agar memiliki pengalaman bekerja pada yang memiliki keterkaitan dengan kajian di bidang Teknologi dan Informasi secara menyeluruh atau sebagian.
2. Memberikan wawasan baru bagi mahasiswa di bidang Teknologi dan Informasi.
3. Mahasiswa dapat mengetahui dan memahami rangkaian kegiatan pada perusahaan/institusi tersebut.
4. Mahasiswa dapat membandingkan kajian teoritis dengan kenyataan di lapangan serta belajar mengambil sikap (menempatkan diri) di dalam bekerja sehubungan dengan keterkaitan berbagai aspek/bidang dalam suatu perusahaan/institusi.
5. Sebagai hasil praktik lapangan diharapkan memberi kemampuan pada mahasiswa untuk mengidentifikasi masalah dan belajar menganalisanya.
6. Untuk menguji kemampuan mahasiswa dan menjadikannya sebagai tolak ukur terhadap ilmu yang didapat sesuai dengan bidangnya selama kuliah.

Tujuan Khusus:

1. Untuk mengetahui cara melakukan perancangan sistem terdistribusi berbasis *microservice* pada layanan E-Makam Kota Malang.
2. Untuk menyediakan sistem pelayanan yang lebih efisien kepada pengguna layanan E-Makam Pemerintah kota Malang.

## Manfaat

Adapun manfaat diadakannya praktik kerja lapangan pada Dinas Komunikasi dan Informatika Kota Malang adalah sebagai berikut.

**Bagi Mahasiswa**

1. Dengan mengikuti kegiatan praktik kerja lapangan mahasiswa mampu memperoleh pengalaman secara nyata bagaimana kultur dan cara bekerja secara nyata pada sebuah perusahaan.
2. Mahasiswa akan lebih mampu dan mengetahui apa saja yang harus dilakukan ketika memasuki dunia kerja.
3. Dapat melakukan tolak ukur tentang kompetensi dan ilmu yang dimilikinya saat ini dengan yang dibutuhkan pada dunia kerja.
4. Dapat menambah ilmu pengetahuan yang dimilikinya.
5. Sebagai salah satu syarat kelengkapan akademik yang harus diselesaikan oleh mahasiswa Universitas Brawijaya.

**Bagi Dinas Komunikasi dan Informatika Kota Malang**

1. Sebagai sarana untuk mengetahui kualitas pendidikan di Universitas Brawijaya dan di Fakultas Ilmu Komputer khususnya.
2. Sebagai sarana untuk memanfaatkan sumber daya yang potensial

## Batasan Masalah

Adapun batasan masalah penelitian adalah :

1. Melakukan implementasi Sistem Terdistribusi berbasis *microservice* pada *backend* API.
2. *Framework* *backend* menggunakan laravel 5.6

## Sistematika Pembahasan

**BAB 1 PENDAHULUAN**

Bab ini membahas tentang uraian umum yang terdiri dari latar belakang, rumusan masalah, tujuan dari pelaksanaan PKL, manfaatnya serta tempat dan waktu dari pelaksanaan PKL tersebut.

**BAB 2 GAMBARAN UMUM PERUSAHAAN**

Bab ini membahas tentang sejarah, bidang usaha, Struktur Organisasi dan gambaran umum yang berkaitan dengan Dinas Komunikasi dan Informatika Kota Malang.

**BAB 3 DASAR TEORI**

Bab ini menjelaskan tentang uraian dasar teori dan referensi yang mendasari tentang impelementasi sistem terdistribusi yang akan dilakukan pada layanan E-Makam di Kota Malang.

**BAB 4 METODOLOGI**

Merupakan Deskripsi dari uraian kegiatan dan tugas yang telah diberikan oleh Dinas Komunikasi dan Informatika Kota Malang.

**BAB 5 HASIL DAN PEMBAHASAN**

Bab ini berisi tentang perancangan, implementasi, serta pengujian sistem terdistribusi berbasis *microservice* pada layanan E-Makam Kota Malang.

**BAB 6 PENUTUP**

Memuat kesimpulan dari keseluruhan bab-bab sebelumnya, serta saran-saran dari hasil yang diperoleh, yang diharapkan dapat bermanfaat dalam pembelajaran selanjutnya.

# GAMBARAN UMUM PERUSAHAAN

## Sejarah Dinas Komunikasi dan Informatika

Dinas komunikasi dan informatika kota Malang merupakan satuan kerja perangkat daerah baru dimana dinas ini dibentuk untuk memenuhi ketentuan peraturan pemerintah nomor: 41 tahun 2007 tentang organisasi perangkat daerah serta memperhatikan kebutuhan pelayanan masyarakat sebagai hasil analisa tim analisis jabatan dan pengukuran beban kerja dalam rangka pengembangan dan pengelolaan teknologi informasi dan komunikasi, maka dibentuklah dinas KOMINFO Kota Malang. Bidang – bidang yang ada pada Dinas Kominfo Kota Malang :

1. Bidang Pengelolaan Komunikasi Publik dan Persandian

Bidang Pos dan Telekomunikasi terdiri dari :

* Seksi Sarana dan Prasarana Komunikasi Publik
* Seksi Pengelolaan dan Kemitraan Komunikasi Publik
* Seksi Persandian

1. Bidang Statistik

Bidang Statistik terdiri dari :

* Seksi Seksi Pengelolaan Data Statistik
* Seksi Seksi Layanan Data dan Informasi Statistik
* Seksi Seksi Evaluasi dan Pelaporan

1. Bidang Aplikasi Informatika

Bidang Aplikasi dan Informatika terdiri dari :

* Seksi Seksi Pengelolaan e-Gov
* Seksi Seksi Sarana Prasarana Informatika
* Seksi Seksi pemberdayaan Teknologi Informasi dan Komunikasi Aplikasi

1. Bidang Informasi Publik

Bidang Informasi Publik terdiri dari :

* Seksi Pengelolaan Informasi Publik
* Seksi Layanan Informasi
* Seksi Pemberdayaan dan Kemitraan Informasi Publik

1. UPT LPSE

Untuk menyusun sistem pengadaan secara elektronik dalam rangka efisiensi, efektifitas, transparansi, persaingan sehat dan akuntabilitas pengadaan barang/jasa pemerintah di lingkungan Pemerintah Kota Malang, maka berdasarkakan [Peraturan Walikota Malang Nomor : 13 Tahun 2011](http://uptlpse.malangkota.go.id/?wpdmact=process&did=MTQuaG90bGluaw==), tanggal 24 Januari 2011 dibentuklah **Unit Pelaksana Teknis Layanan Pengadaan Secara Elektronik (UPT LPSE) Pada Dinas Komunikasi dan Informatika.** UPT LPSE merupakan unsur pelaksana sebagian kegiatan teknis operasional Dinas Komunikasi dan Informatika, yang melaksanakan tugas pokok :

1. Memfasilitasi Pengguna Anggaran/Kuasa Pengguna Anggaran mengumumkan rencana umum pengadaan
2. Memfasilitasi Panitia Pengadaan menayangkan pengumuman pelaksanaan pengadaan
3. Memfasilitasi Panitia/Pejabat Pengadaan melaksanakan pemilihan penyedia barang/jasa secara elektronik
4. Memfasilitasi penyedia barang/jasa dan pihak-pihak lain yang berkepentingan menjadi Pengguna SPSE

## Visi, Misi dan Tujuan Dinas Komunikasi dan Informatika

Dinas Komunikasi dan Informatika memiliki visi dan misi yang menuntun lembaga tersebut untuk melakukan tindakan sesuai dengan yang direncanakan untuk mencapai sebuah tujuan. Adapun Visi dan Misi Dinas Komunikasi dan Informatik Kota Malang adalah sebagai berikut :

### Visi

Visi Dinas Komunikasi dan Informatika Kota Malang adalah terwujudnya Kota Malang INFORMATIF (INovatif, Fleksibel, ORiginal, MAju, Terukur, Integritas, Faktual).

### Misi

Misi Dinas Komunikasi dan Informatika Kota Malang :

1. Mewujudkan birokrasi layanan komunikasi dan informatika yang profesional dan memiliki integritas moral yang tinggi
2. Mengoptimalkan Sistem Layanan Kepemerintahan (e-Government) yang terintegrasi
3. Meningkatkan kecukupan informasi masyarakat dengan karakteristik komunikasi lancar dan informasi benar dalam rangka meningkatan kapasitas layanan publik yang dapat dipertanggungjawabkan
4. Meningkatkan Pemberdayaan Masyarakat serta Mengembangkan Kemitraan, dan Lembaga Komunikasi dalam Penyebaran Informasi Publik berbasis Kearifan Lokal
5. Mewujudkan profesionalisme Layanan Pos dan Telekomunikasi

### Tujuan

Untuk mewujudkan visi dan misi yang diinginkan, perlu ditetapkan tujuan yang dimana tujuan dari Dinas Komunikasi dan Informatika adalah:

1. Meningkatkan Sumber Daya Manusia Aparatur yang mempunyai kompetensi di bidang Teknologi Informasi Komunikasi (TIK);
2. Meningkatkan komitmen dan integritas Aparatur yang tinggi dalam melaksanakan tugas dan fungsinya di bidang Teknologi Informasi Komunikasi (TIK)
3. Peningkatan Kualitas Pelayanan Publik yang didukung oleh sarana/prasarana yang memadai dan data yang menunjang
4. Pengembangan Master Plan e-Government Kota Malang
5. Pembangunan sistem layanan kepemerintahan yang terintegrasi dan memiliki interoperabilitas dalam layanan publik secara online (e-Citizen, e-Licensing dan e-Bussines)
6. Peningkatan Sistem Keamanan Komunikasi dan Informatika
7. Pengembangan dan Pelayanan Informasi Publik dalam mendukung Keterbukaan Informasi Publik
8. Peningkatan kualitas pengelolaan, penyebaran dan pemerataan informasi publik
9. Pengembangan dan Pemberdayaan Kelompok Informasi Masyarakat (KIM)
10. Peningkatan Diseminasi dan Pendistribusian Informasi melalui Media Massa maupun Media komunikasi lainnya serta Kemitraan Media.
11. Percepatan Pembangunan National Broadband Network serta penataan dan pengendalian proses perizinan menara telekomunikasi
12. Peningkatan Pendapatan Asli Daerah di bidang Telekomunikasi

## Logo Dinas Komunikasi dan Informatika Kota Malang

Dinas Komunikasi dan Informatika pemerintah Kota Malang yang dibentuk untuk memenuhi ketentuan peraturan pemerintah nomor: 41 tahun 2007 mengenai organisasi perangkat daerah yang mempunyai logo seperti pada Gambar 1.



Gambar 2.1 Logo Dinas Komunikasi dan Informatika Kota Malang

## Struktur Organisasi Dinas Komunikasi dan Informatika Kota Malang

Berikut adalah gambar struktur organisasi Dinas Komunikasi dan Informatika Kota Malang:



Gambar 2.2 Struktur Organisasi Dinas Komunikasi dan Informatika Kota Malang

Gambar 2 merupakan gambaran struktur organisasi dari Dinas Komunikasi dan Informatika Kota Malang yang dipimpin oleh kepala dinas Bapak Zulkifli Amrizal, S.Sos, M.Si.

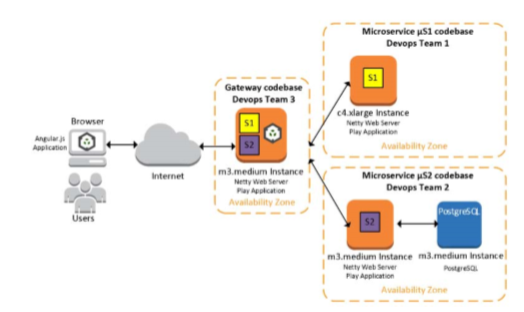
# DASAR TEORI

## Sistem Terdistribusi

Sistem Terdistribusi adalah Sekumpulan komputer otonom yang terhubung ke suatu jaringan, dimana bagi pengguna sistem terlihat sebagai satu komputer, saling terhubung, berkomunikasi dan berkoordinasi lewat pesan untuk mencapai tujuan yang sama. (Coulouris, Dollimore dan Kindberg, 2012). Infrastruktur utama sistem terdistribusi adalah jaringan, *hardware*, *software* dan pengguna yang terkait di dalamnya. Dalam sistem terdistribusi terdapat pembagian pekerjaan antara elemen yang satu dengan elemen yang lain. Sarana komunikasi antar elemen dijembatani dengan jaringan. Tata cara komunikasi antar elemen diatur dengan sebuah perjanjian sehingga terjadi komunikasi yang dapat dipahami antara masing-masing elemen yang terlibat.

## Arsitektur *Microservice*

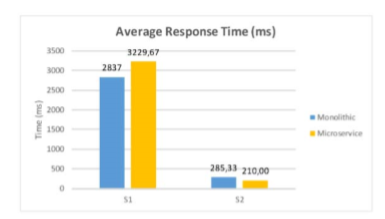
Arsitektur *microservice* adalah kumpulan proses independen dan kecil yang berkomunikasi antara satu dengan lainnya untuk membentuk aplikasi kompleks yang dinamis terhadap semua bahasa API. Layanan-layanan ini terdiri dari blok-blok kecil, terpisah, dan fokus pada tugas-tugas ringan untuk memfasilitasi metode modular dalam pembangunan sistem (Chen, 2018). Arsitektur bergaya *microservice* mulai menjadi standar dalam pembangunan sistem yang dinamis dan konstan berkembang. Arsitektur *microservice* memiliki beberapa kelebihan, diantaranya *maintenance* lebih mudah, aplikasi aman dan handal, dan setiap *service* berdiri sendiri tanpa mengganggu *service* yang lain. Berikut adalah contoh arsitektur micro *service*



Gambar 3.1 Gambar arsitektur microservice pada amazon webservice (villamizar., 2015)

### Kinerja *Microservice*

Dalam hal kinerja micro *service* memiliki kekurangan dalam response time dari setiap *request*. Rata rata response time dari *request* arsitektur micro *service* cenderung lebih besar dibandingkan dengan hasil dari arsitektur monolitik pada instansi *service* yang berukuran terlalu besar. Sedangkan untuk *service* berukran sedang cenderung sama (Vamilizar, 2015). Berikut adalah grafik perbandingan kinerja monolitik dan *microservice*



Gambar 3.2 Gambar grafik perbandingan response time (Villamizar, 2015)

### Pengembangan *Microservice*

Dalam pengembangan *service* dengan mengunakan arsitektur *microservice* bisa memungkinkan pada pengembang melakukan pekerjaannya secara terpisah tanpa memperdulikan bagaimana pekerjaan micro *service* lain. Setiap tim dapat menggunakan teknologi yang berbeda beda walaupun penggunaan teknologi yang terlalu beragam tetap akan sulit untuk di atur dan dibuat dokumentasinya. Hal ini juga akan memudahkan pengembangan secara kontinyu (Villamizar, 2015)

## *Webservice*

*Webservice* merupakan sebuah sistem yang mengandung informasi dan perintah yang mendukung interoperabilitas komunikasi antara mesin ke mesin melalui suatu protokol jaringan HTTP atau HTTPS (Booth, 2004). *Webservice* menggunakan jenis komunikasi seperti REST, SOAP dan WSDL untuk pertukaran data. Salah satu bentuk layanan *webservice* adalah API (*Application Programming Interface*) merupakan metode suatu program yang menyediakan kumpulan fungsi atau perintah untuk mengakses data ke database. *Webservice* memiliki kelebihan diantaranya mudah diterapkan dan dikembangkan serta merupakan sistem yang paling banyak digunakan oleh pengembang web saat ini.

### REST (*Representational State Transfer*)

REST mendefinisikan sekelompok prinsip arsitektur yang dapat kita design. REST adalah *webservice* yang memiliki fokus pada *resource* sistem, termasuk bagaimana state dari *resource* dialamatkan di transfer melalui HTTP. Karakteristik dari REST adalah penggunakan metode HTTP yang eksplisit (Rodriguez, A., 2008).

### SOAP

SOAP adalah sebuah protokol untuk komunikasi *remote procedure calls (RPCs)* berbasis XML. Soap juga menggunakan HTTP, SMTP dan MQSerises. Pada intinya *message* SOAP sangat sederhana yaitu elemen XML dengan dua *child element*, salah satunya berisi *header* dan satunya lagi adalah *body*. Sebagai tambahan pada struktur *message* pada SOAP didefnisikan model bagaimana penerima memproses *message* SOAP. Model ini termasuk *aktor* dan siapa yang seharusnya meproses *message* tersebut. Pada awalnya SOAP dikembangkan oleh Microsoft yang akhirnya di kembangkan dengan berkolaborasi dengan developmentor, IBM, Lotus, dan *User*land.(Curbera. F et al., 2002).

### WSDL

WDSL adalah format XML yang dikembangkan oleh IBM dan Microsoft untuk menDeskripsikan *webservice* sebagai sekumpulan end point komunikasi yang dapat bertukar pesan tertentu. Dengan kata lain WDSL menDeskripsikan *Webservice* sebagai *interface* dan memberikan *user* sebuah *point of contact.* DeskripsiWSDL *service* memiliki 2 bagian informasi yaitu deksipsi application level atau *abstract interface* dan *specific protocol-dependent* yang harus diikuti oleh pengguna untuk mendapatkan akses *service* (Curbera. F et al., 2002).

## *Gateway* API

*Gateway* API adalah *interface* yang menyampaikan *call* atau *request* antara dua atau lebih *microservice*s. *Gateway* API dapat menggabungkan API untuk beberapa *microservice*s menjadi satu *interface* *client*, dan juga dapat mendistribusikan atau merutekan *call* atau *request* dari satu titik masuk ke beberapa *microservice*s target (Lu, 2017). Sehingga semua *request* dari client pertama akan melewati API *Gateway* kemudian diteruskan ke *microservice* API yang tepat. Fungsi lain yang dapat dijalankan oleh API seperti *authentication*, *monitoring*, *load balancing*, dan *security*.

## Laravel

Laravel merupakan salah satu *framework* PHP untuk sistem aplikasi web dengan sintaks yang ekspresif dan elegan dan memberi solusi untuk pengembangan pada sebagian besar proyek web (Alfat, 2015). Aplikasi web yang dibangun dengan laravel memiliki konsep arsitektur *model, view, controller (MVC)*. MVC merupakan konsep pemisahan antara manipulasi data, *controller* dan *user interface*. Laravel memudahkan dalam proses pengembangan web dengan menyederhanakan proses back-end seperti *routing*, *authentication* dan sessions. Beberapa fungsi pada laravel yang memudahkan dalam pengembangan sistem web yaitu migrasi database, keamanan, dan *reverse* *routing*.

## *ReactJS*

*ReactJS* adalah sebuah *library* dari *JavaScript* untuk membuat *User* Inteface (*ReactJS*.org). Dan *JavaScript* itu sendiri adalah sebuah Bahasa *script* yang juga dapat digunakan untuk membuat sebuah aplikasi web lebih dinamis. *JavaScript* biasanya digunakan untuk mengembangkan *frontend* dari sebuah aplikasi. *JavaScript* ditujukan agar client mampu melakukan sebuah pemrosesan sehingga tidak semua pemrosesan diserahkan pada server.

## PHP Hypertext Preprocessor (PHP)

*PHP Hypertext Preprocessor (PHP*). Adalah sebuah Bahasa *script* yang ditujukan agar *developer* dapat membuat sebuah web yang dapat mengenerate alamat atau komponennya secara dinamis. PHP adalah *script* yang umumnya digunakan sebagai *backend* *system* sebuah aplikasi baik berupa *backend* web maupun API untuk sebuah aplikasi *Mobile*.

# METODOLOGI

Pada bab ini akan menjelaskan langkah-langkah penelitian yang digunakan dalam melakukan implementasi sistem terdistribusi berbasis *microservice* pada layanan sistem E-makam di Pemerintah Kota Malang. Dimulai dengan studi literatur yang berhubungan dengan topik yang diangkat, kemudian analisis kebutuhan, perancangan jaringan, dan kesimpulan sebagai rangkuman hasil akhir dari penelitian yang dilakukan.

Studi Literatur

Analisis Kebutuhan

Perancangan Sistem

Kesimpulan

Implementasi

Pengujian

Gambar 4.1 Diagram Alir Penelitian

## Studi Literatur

Dalam tahap ini studi literatur dilakukan studi pustaka atau literature untuk mencari referensi teori yang relevan dengan kasus atau permasalahan dimana referensi tersebut ditemukan dalam buku, paper, jurnal yang terlah ada. Studi literature dilakukan untuk membantu dan mempermudah dalam melaksanakan penelitian. Dalam implementasi kami menggunakan referensi dari jurnal atau buku untuk membantu pengerjaan kami. Berikut adalah referensi yang kami gunakan:

* 1. J. Stubbs, W. Moreira and R. Dooley, "Distributed *System*s of *Microservice*s Using Docker and Serfnode," 2015 7th International Workshop on Science *Gateway*s, Budapest, 2015, pp. 34-39.

## Analisis Kebutuhan

Analisis dilakukan untuk mendapatkan kebutuhan-kebutuhan yang diperlukan dan mendukung seluruh tahapan dalam penelitian. Analisis kebutuhan bisa diperoleh dari proses pencarian studi literature. Kebutuhan pada penelitian ini meliputi kebutuhan fungsional dan non-fungsional yang digunakan untuk membangun sistem. Kebutuhan fungsional sistem, merupakan kebutuhan yang dibutuhkan agar sistem dapat berjalan. Kebutuhan tersebut yaitu proses manajemen makam dan proses registrasi perizinan izin. Sedangkan kebutuhan non-fungsional berfokus pada perilaku dan batasan fungsi dari sistem. Kebutuhan non-fungsional pada penelitian ini adalah skalabilitas dan interopabilitas.

## Perancangan

Tahap perancangan sistem merupakan tahapan yang dilaksanakan setelah kebutuhan pada tahap analisis kebutuhan terpenuhi. Tahapan ini dimaksudkan agar mempermudah implementasi selanjutnya. Manfaat dari tahapan perancangan sistem, implementasi bisa berjalan dengan sistematis dan terstruktur. Penelitian ini menggunakan sistem terdistribusi berbasis *microservice.* Agar hal tersebut terwujud diperlukan perancangan terhadap penelitian yang akan dibangun. Perancangan tersebut yaitu:

* Perancangan sistem terdistribusi
* Perancangan database

## Implementasi

Di tahap ini akan dilakukan implementasi berdasarkan perancangan yang telah dilakukan. Langkah awal dalam proses implementasi pada penelitian ini adalah implementasi *database*. Kemudian dilanjutkan dengan implementasi *API* *Gateway* yang bertugas sebagai penjembatan antara aplikasi dengan backend *service*. Setelah itu akan dilakukan implementasi *backend* *service*.

## Pengujian

Tahap pengujian bertujuan untuk mengetahui kebutuhan fungsional yang telah tercapai oleh sistem yang dibangun. Jika terdapat kebutuhan fungsional yang belum tercapai maka dilakukan evaluasi terhadap perancangan dan implementasi sistem. Selain menguji kebutuhan fungsional yang telah dicapai sistem, pengujian terhadap aspek non-fungsional juga dilakukan, dalam hal ini adalah skalabilitas. Pengujian skalabilitas sistem bertujuan untuk mengetahui kapasitas sistem dalam menangani berbagai proses yang diterima ketika terdapat perubahan jumlah beban yang lebih besar dari sebelumnya. Data pengujian kemudian dianalisis untuk dapat diambil kesimpulan dan saran.

## Kesimpulan

Pada tahap ini, akan ditarik sebuah kesimpulan dari proses yang telah dilakukan pada penilitian ini. Sehingga dapat diketahui kinerja dari perancangan sistem terdistribusi berbasis *microservice* setelah proses penelitian. Kesimpulan dibuat berdasarkan hasil analisis sehingga diperoleh inti dari keseluruhan proses penelitian dan berutujuan untuk menjawab permasalahan dalam rumusan masalah.

# PEMBAHASAN DAN HASIL

## Analisis Kebutuhan

### Deskripsi Umum

Sistem E-makam adalah sebuah sistem digitalisasi proses perizinan makam dari awal proses permintaan oleh ahli waris hingga surat izin keluar. Sistem akan menyimpan seluruh dokumen dan mempermudah proses perizinan. Sistem dapat melakukan pelacakan proses perizinan. Sistem juga memiliki fungsi manajemen tpu serta blok makam dan makam berserta penghuninya. Sistem ini akan mengenali *user* berdasarkan role aktor dan hak aksesnya terhadap lokasi tpu dan kecamatannya masing-masing.

### Diagram Alur Sistem

Kebutuhan didapatkan dari pemangku tangan yaitu Dinas Kominfo Kota Malang, kebutuhan diDeskripsikan kedalam sebuah alur diagram. Berikut adalah alur diagram kebutuhan sistem

Tabel 5.1 Diagram Alur Kebutuhan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kegiatan | Pelaksana | | | | |
| Pemohon /Ahli waris | Koordinator  TPU | Kepala UPT pemakaman | Kepala dinas perkim | Kantor Kecamatan di wilayah Kota Malang |
| Membawa berkas surat ke koordinator Tempat pemakaman umum dan mengisi formulir permohonan |  |  |  |  |  |
| Memeriksa permohonan, menandatangani dan memberikan tanda terima permohonan serta menyiapkan lahan makam,  Memasukan seluruh data, dan mengecek kelengkapan, |  |  |  |  |  |
| Melihat Surat permohonan, KTP, KK dan seluruh dokumen dari koordinator tempat pemakaman umum,  Memberikan rekomendasi dengan menekan tombol rekomendasi untuk dibaca oleh kepala dinas |  |  |  |  |  |
| validasi dokumen , dan memberikan nomor surat permohonan, dengan menekan tombol kepala dinas akan mendapatkan Salinan dokumen hasil dan sistem akan meneruskan data untuk diproses kepala kecamatan, |  |  |  |  |  |
| Validasi surat oleh kepala camat dengan menekan tombol, serta memberikan nomor surat perizinan oleh administrator . sistem akan mengirimkan email kepada pemohon. Administrator mencetak surat surat untuk diberikan kepada pemohon/ahliwaris |  |  |  |  |  |
| Menerima surat izin |  |  |  |  |  |

### Aktor Sistem

Berdasarkan alur sistem pada sub bab sebelumnya dapat ditarik kesimpulan bahwa aktor-aktor yang akan ada pada sistem adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deksripsi |
| 1 | Admin Pusat | Administrator sistem aplikasi |
| 2 | Admin TPU | Administrator tpu spesifik |
| 3 | Kepala TPU | Kepala tpu spesifik |
| 4 | Kepala Dinas | Kepala dinas |
| 5 | Admin Kecamatan | Administrator kecamatan spesifik |
| 6 | Kepala Kecamatan | Kepala kecamatan spesifik |

### Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus ada dan dipenuhi dalam sistem. Kebutuhan fungsional juga mengacu pada proses-proses yang dapat dilakukan oleh sistem. Terdapat enam aktor pada sistem, yang memiliki kebutuhan fungsional yang berbeda.

### Kebutuhan Admin Pusat

Kebutuhan fungsional untuk aktor Admin Pusat dapat diliat pada tabel berikut ini:

Tabel 5.2 Tabel Kebutuhan Fungsional Admin Pusat

|  |  |  |
| --- | --- | --- |
| No | Nama Fungsi | Deskripsi |
| 1 | Create TPU | Sistem dapat menambahkan TPU Baru melalui halaman Manajemen TPU |
| 2 | Read TPU | Sistem dapat membaca TPU yang tersimpan di database dan menampilkannya di halaman Manajemen TPU |
| 3 | Update TPU | Sistem dapat menyunting TPU yang tersimpan di database melalui halaman Manajemen TPU |
| 4 | Delete TPU | Sistem dapat Menghapus TPU yang tersimpan di database melalui halaman Manajemen TPU |
| 5 | Create *User* | Sistem dapat menambahkan Pengguna Baru melalui halaman Manajemen Pengguna |
| 6 | Read *User* | Sistem dapat membaca Pengguna yang tersimpan di database dan menampilkannya di halaman Manajemen Pengguna |
| 7 | Update *User* | Sistem dapat menyunting Pengguna yang tersimpan di database melalui halaman Manajemen Pengguna |
| 8 | Delete *User* | Sistem dapat Menghapus Pengguna yang tersimpan di database melalui halaman Manajemen Pengguna |
| 9 | Create *User* Access | Sistem dapat menambahkan hak akses kepada *user* melalui halaman Manajemen Hak Akses |
| 10 | Read *User* Access | Sistem dapat membaca Hak Akses *user* yang tersimpan di database dan menampilkannya di halaman Manajemen Hak Akses |
| 11 | Update *User* Access | Sistem dapat menyunting Hak Akses *User* yang tersimpan di database melalui halaman Manajemen Hak Akses |
| 12 | Delete *User* Access | Sistem dapat Menghapus Hak Akses *User* yang tersimpan di database melalui halaman Manajemen Hak Akses |
| 13 | Create Kecamatan Access | Sistem dapat menambahkan Hak Akses Kecamatan kepada *user* melalui halaman Manajemen Akses Kecamatan |
| 14 | Read Kecamatan Access | Sistem dapat membaca Hak Akses Kecamatan yang tersimpan di database dan menampilkannya di halaman Manajemen Akses Kecamatan |
| 15 | Update Kecamatan Access | Sistem dapat menyunting Hak Akses Kecamatan yang tersimpan di database melalui halaman Manajemen Akses Kecamatan |
| 16 | Delete Kecamatan Access | Sistem dapat Menghapus Hak Akses Kecamatan yang tersimpan di database melalui halaman Manajemen Akses Kecamatan |
| 17 | Cari Data | Sistem dapat mencari data penghuni makam yang tersimpan di database dan menampilkannya di halaman Cari Data |
| 18 | Cari Dokumen | Sistem dapat mencari data dokumen yang tersimpan di database dan menampilkannya di halaman Dokumen |

### Kebutuhan Admin TPU

Kebutuhan fungsional untuk aktor Admin TPU dapat diliat pada tabel berikut ini:

Tabel 5.3 Tabel Kebutuhan Fungsional Admin TPU

|  |  |  |
| --- | --- | --- |
| No | Nama Fungsi | Deskripsi |
| 1 | Registrasi Perijinan | Sistem dapat menambahkan registrasi perijinan baru melalui halaman Registrasi Perijinan |
| 2 | Create Blok Makam | Sistem dapat menambahkan Blok Makam Baru melalui halaman Manajemen Blok Makam |
| 3 | Read Blok Makam | Sistem dapat membaca Blok Makam yang tersimpan di database dan menampilkannya di halaman Manajemen Blok Makam |
| 4 | Update Blok Makam | Sistem dapat menyunting Blok Makam yang tersimpan di database melalui halaman Manajemen Blok Makam |
| 5 | Delete Blok Makam | Sistem dapat Menghapus Blok Makam yang tersimpan di database melalui halaman Manajemen Blok Makam |
| 6 | Create Makam | Sistem dapat menambahkan Makam Baru melalui halaman Manajemen Makam |
| 7 | Read Makam | Sistem dapat membaca Makam yang tersimpan di database dan menampilkannya di halaman Manajemen Makam |
| 8 | Update Makam | Sistem dapat menyunting Makam yang tersimpan di database melalui halaman Manajemen Makam |
| 9 | Delete Makam | Sistem dapat Menghapus Makam yang tersimpan di database melalui halaman Manajemen Makam |
| 10 | Create Penghuni Makam | Sistem dapat menambahkan Penghuni Makam Baru melalui halaman Manajemen Penghuni Makam |
| 11 | Read Penghuni Makam | Sistem dapat membaca Penghuni Makam yang tersimpan di database dan menampilkannya di halaman Manajemen Penghuni Makam |
| 12 | Update Penghuni Makam | Sistem dapat menyunting Penghuni Makam yang tersimpan di database melalui halaman Manajemen Penghuni Makam |
| 13 | Delete Penghuni Makam | Sistem dapat Menghapus Penghuni Makam yang tersimpan di database melalui halaman Manajemen Penghuni Makam |
| 14 | Cari Data | Sistem dapat mencari data penghuni makam yang tersimpan di database dan menampilkannya di halaman Cari Data |
| 15 | Cari Dokumen | Sistem dapat mencari data dokumen yang tersimpan di database dan menampilkannya di halaman Dokumen |
| 16 | Acc Dokumen | Sistem dapat menyetujui dokumen |
| 17 | Cari Data | Sistem dapat mencari data penghuni makam yang tersimpan di database dan menampilkannya di halaman Cari Data |

### 

### Kebutuhan Kepala UPT

Kebutuhan fungsional untuk aktor Kepala UPT Pemakaman dapat diliat pada tabel berikut ini:

Tabel 5.4 Tabel Kebutuhan Fungsional Kepala UPT Pemakaman

|  |  |  |
| --- | --- | --- |
| No | Nama Fungsi | Deskripsi |
| 1 | Cari Dokumen | Sistem dapat mencari data dokumen yang tersimpan di database dan menampilkannya di halaman Dokumen |
| 2 | Acc Dokumen | Sistem dapat menyetujui dokumen |

### Kebutuhan Kepala Dinas

Kebutuhan fungsional untuk aktor Kepala Dinas dapat diliat pada tabel berikut ini:

Tabel 5.5 Tabel Kebutuhan Fungsional Kepala Dinas

|  |  |  |
| --- | --- | --- |
| No | Nama Fungsi | Deskripsi |
| 1 | Cari Dokumen | Sistem dapat mencari data dokumen yang tersimpan di database dan menampilkannya di halaman Dokumen |
| 2 | Acc Dokumen | Sistem dapat menyetujui dokumen |
| 3 | Cetak Surat Permohonan | Sistem dapat menampilkan surat permohonan dengan tipe data pdf kepada browser |

### Kebutuhan Kepala Kecamatan

Kebutuhan fungsional untuk aktor Kepala Kecamatan dapat diliat pada tabel berikut ini:

Tabel 5.6 Tabel Kebutuhan Fungsional Kepala Kecamatan

|  |  |  |
| --- | --- | --- |
| No | Nama Fungsi | Deskripsi |
| 1 | Cari Dokumen | Sistem dapat mencari data dokumen yang tersimpan di database dan menampilkannya di halaman Dokumen |
| 2 | Acc Dokumen | Sistem dapat menyetujui dokumen |

### Kebutuhan Admin kecamatan

Kebutuhan fungsional untuk aktor Admin kecamatan dapat diliat pada tabel berikut ini:

Tabel 5.7 Tabel Kebutuhan Fungsional Admin Kecamatan

|  |  |  |
| --- | --- | --- |
| No | Nama Fungsi | Deskripsi |
| 1 | Cari Dokumen | Sistem dapat mencari data dokumen yang tersimpan di database dan menampilkannya di halaman Dokumen |
| 2 | Acc Dokumen | Sistem dapat menyetujui dokumen |
| 3 | Cetak Surat Perizinan | Sistem dapat menampilkan surat perizinan dengan tipe data pdf kepada browser |

### Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan pendukung yang berfungsi dalam membatasi fungsi pada suatu sistem. Adapun kebutuhan non-fungsional dapat dilihat pada tabel sebagai berikut:

Tabel 5.8 Tabel Kebutuhan Non-Fungsional

|  |  |  |
| --- | --- | --- |
| No | Nama Fungsi | Deskripsi |
| 1 | Skalabilitas | Sistem dapat menangani minimal 500 *request* secara bersamaan |
| 2 | Availabilitas | Sistem dapat tersedia sesuai dengan jam kerja pegawai |

### Kebutuhan Perangkat keras

Kebutuhan perangkat keras dalam sistem tedistribusi kami antara lain:

1. 1 perangkat server
2. Minimal 1 buah perangkat Host

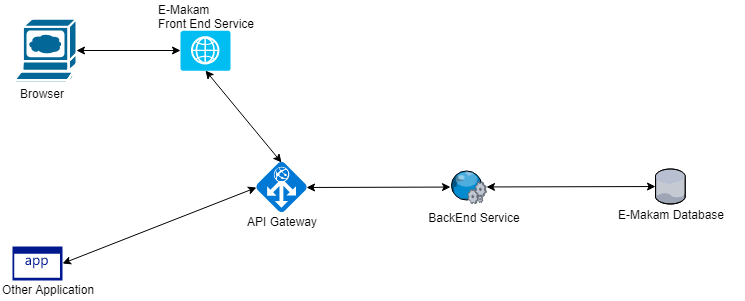
### Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam sistem tedistribusi kami antara lain:

* 1. Ubuntu Server 18.04
  2. MySQL Server
  3. Php7.2
  4. Apache2
  5. Web Client

## Perancangan Sistem

Berikut adalah perancangan sistem terdistribusi yang ingin kami lakukan:



Gambar 5.1 Diagram Perancangan Sistem Terdistribusi

*User* akan me*request* aplikasi melalui web browser. Kemudian aplikasi akan melakukan *request* kepada api *Gateway* sesuai dengan fungsinya masing-masing. *Gateway* akan mengarahkan *request* sesuai dengan permintaan ke*service* yang sesuai. *Gateway* juga akan mampu melayani aplikasi dari luar jika diinginkan dan menyediakan *service* yang ada untuk masa yang akan mendatang. Berdasarkan kebutuhan yang ada beberapa *service* terpisah yang dihubungkan ke sebuah *Gateway*.

### Distribusi Sistem

Distribusi sistem akan dilakukan pada pemisahan *Frontend service* dan *Backend* *service* sistem yang dibuat secara terpisah. Komunikasi antar *service* akan dibantu oleh *gateway* yang akan dibuat. Api *gateway* akan menangani *request* dari *frontend service* yang akan diproses oleh *backend service*. Komunikasi terjadi dua arah berupa *request response*, *gateway* akan mengatur arahnya *request* dari suatu *service* untuk di teruskan ke *service* tujuan. Hal serupa juga berlaku untuk *response* dari *service*. Dalam sistem kami akan dibuat dua buah *service* terpisah yaitu *frontend* *service* dan *backend* *service* yang dihubungkan dengan sebuah *gateway*. Seperti yang ditampilkan pada gambar 5.1.

### Perancangan *Service*

*Service* yang akan kami bangun dalam implementasi ini akan terbagi dua yaitu *Frontend service* dan *backend service* . Frontend *service* kami adalah *service* yang melayani user secara langsung melalui tampilan berbasis grafis pada *Web browser* . *Frontend service* ini sendiri akan dikembangan menggunakan teknologi *ReactJS. Backend service* yang akan kami bangun akan melayani *frontend service* untuk terhubung dengan database. *Backend service* yang akan kami kembangkan. Kami kelompokkan kedalam beberapa fitur terpisah berdasarkan fungsinya.

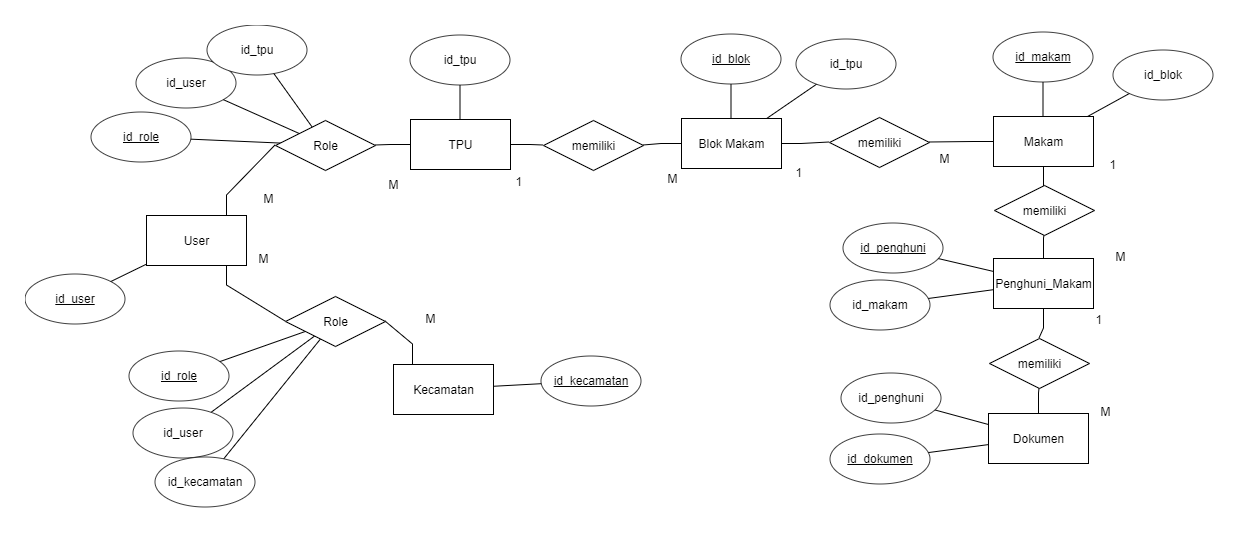
Berikut adalah rincian fitur yang akan kami implementasikan pada backend *service* kami.

* 1. Manajemen TPU
  2. Manajemen *User*
  3. Manajemen *User* Access
  4. Manajemen Kecamatan
  5. Manajemen Registrasi
  6. Manajemen Dokumen
  7. Manajemen Makam
  8. Manajemen Blok Makam
  9. Manajemen Penghuni Makam
  10. *Authentication*
  11. Mailing

Fitur manajemen TPU adalah fitur yang menangani request Create, Read, Update, dan Delete dari database tabel TPU. Fitur manajemen *user* adalah fitur yang menangani *request* Create, Read, Update, dan Delete dari database tabel *User*. Fitur manajemen akses kecamatan adalah fitur yang menangani Create, Read, Update, Delete dari database tabel kecamatan. Fitur Manajemen *User* access akan menangani role dari *user* ke tpu atau ke kecamatannya sesuai hak aksesnya masing-masing. Fitur registrasi perijinan adalah fitur yang menangani registrasi Create, Read, Update, Delete dokumen. Fitur dokumen adalah fitur yang menangani request dokumen dan cetak dokumen. Fitur manajemen makam adalah fitur yang menangani Create, Read, Update, Delete database tabel makam. Fitur manajemen blok makam akan menangani Create, Read, Update, Delete database tabel blok makam. Fitur manajemen penghuni makam akan menangani Create, Read, Update, Delete database tabel penghuni makam. Fitur *authentication* akan menangani *authentication* login dan *generating* *token*. Fitur *mailing* akan menangani seluruh mailing sistem.

## Perancangan Database

Database akan dibuat dengan sebuah database MySQL. Berdasarkan rincian kebutuhan tedapat enam aktor yang memiliki peran dan keterhubungan masing-masing. Berikut adalah perancangan database dalam bentuk *Entity Relationship Diagram* (ERD) :



Gambar 5.2 Diagram Perancangan Sistem Terdistribusi

Berdasarkan gambar 5.2 seluruh aktor disimplifikasi menjadi sebuah tabel sederhana yaitu *user* dan akan memiliki hubungan *Many to Many* ke TPU dan kecamatan. Yang mana hubungan tersebut akan membuat tabel relasi baru yaitu role. Setiap Tpu akan memiliki Blok makam dan setiap blok makam memiliki makamnya. Setiap makam memiliki beberapa penghuni dan setiap penghuni bisa memiliki beberapa dokumen. Masing-masing entitas memiliki relasi one to many.

## Implementasi

### Implementasi Database

Berdasarkan perancangan yang telah dilakukan sebelumnya, database akan diimplementasikan ke *MySQL* sehingga database ini dapat digunakan pada backend *service*. Berikut adalah *source* *code* untuk *SQL database.*

Tabel 5.9 Implementasi Database

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549  550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697  698  699  700  701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717  718  719  720  721  722  723  724  725  726  727  728  729  730  731  732  733  734  735  736  737  738  739  740  741  742 | -- phpMyAdmin SQL Dump  -- version 4.8.1  -- https://www.phpmyadmin.net/  --  -- Host: 127.0.0.1  -- Generation Time: Nov 12, 2018 at 02:17 PM  -- Server version: 10.1.33-MariaDB  -- PHP Version: 7.2.6  SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";  SET AUTOCOMMIT = 0;  START TRANSACTION;  SET time\_zone = "+00:00";  /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  /\*!40101 SET NAMES utf8mb4 \*/;  --  -- Database: `emakam`  --  CREATE DATABASE IF NOT EXISTS `emakam` DEFAULT CHARACTER SET latin1 COLLATE latin1\_swedish\_ci;  USE `emakam`;  -- --------------------------------------------------------  --  -- Table structure for table `blok\_makam`  --  CREATE TABLE `blok\_makam` (  `id\_blok` int(11) NOT NULL,  `id\_tpu` int(11) NOT NULL,  `kode\_blok` varchar(45) DEFAULT NULL,  `area\_peta\_blok` int(11) DEFAULT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `blok\_makam`  --  INSERT INTO `blok\_makam` (`id\_blok`, `id\_tpu`, `kode\_blok`, `area\_peta\_blok`) VALUES  (1, 1, 'KASIN-A', 123),  (2, 1, 'KASIN-B', 3425435),  (3, 2, 'XXX-A', 3452354),  (4, 2, 'XXX-B', NULL),  (5, 1, 'KASIN-C', NULL),  (6, 3, 'SAMAAN-A ', NULL),  (7, 4, 'GG-A', NULL),  (10, 4, 'GG-B', NULL),  (11, 4, 'GG-C', NULL),  (12, 5, 'A-1', NULL),  (13, 5, 'A-2', NULL),  (14, 5, 'A-3', NULL),  (19, 3, 'SAMAAN-B', NULL),  (23, 4, 'GG-qwe', NULL),  (29, 2, 'XXX-A', NULL),  (31, 4, 'GG-BB', NULL),  (33, 4, 'GG-DD', NULL);  -- --------------------------------------------------------  --  -- Table structure for table `dokumen`  --  CREATE TABLE `dokumen` (  `id` int(11) NOT NULL,  `kode\_registrasi` varchar(25) DEFAULT NULL,  `nama\_almarhum` varchar(20) DEFAULT NULL,  `nama\_pewaris` varchar(20) DEFAULT NULL,  `tgllhr\_ahli\_waris` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,  `email` varchar(50) NOT NULL DEFAULT 'mail.defailt@mail.com',  `pekerjaan\_ahli\_waris` varchar(50) NOT NULL,  `id\_penghuni\_makam` int(11) NOT NULL,  `id\_tpu` int(11) NOT NULL,  `id\_kecamatan` int(11) NOT NULL,  `file\_ktp` varchar(255) DEFAULT NULL,  `file\_kk` varchar(255) DEFAULT NULL,  `file\_surat\_izin` varchar(255) NOT NULL,  `file\_sk` varchar(255) NOT NULL,  `file\_sk\_lama` varchar(255) DEFAULT 'Kosong',  `kelengkapan\_dokumen` varchar(100) NOT NULL DEFAULT 'Tidak Lengkap',  `status` varchar(50) NOT NULL,  `tanggal\_surat\_permohonan` varchar(50) DEFAULT NULL,  `no\_surat\_permohonan` varchar(50) DEFAULT NULL,  `tanggal\_surat\_perizinin` varchar(50) DEFAULT NULL,  `no\_surat\_perizinan` varchar(50) DEFAULT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `dokumen`  --  INSERT INTO `dokumen` (`id`, `kode\_registrasi`, `nama\_almarhum`, `nama\_pewaris`, `tgllhr\_ahli\_waris`, `email`, `pekerjaan\_ahli\_waris`, `id\_penghuni\_makam`, `id\_tpu`, `id\_kecamatan`, `file\_ktp`, `file\_kk`, `file\_surat\_izin`, `file\_sk`, `file\_sk\_lama`, `kelengkapan\_dokumen`, `status`, `tanggal\_surat\_permohonan`, `no\_surat\_permohonan`, `tanggal\_surat\_perizinin`, `no\_surat\_perizinan`) VALUES  (36, 'GG-36', 'niini', 'nana', '1995-05-05 00:00:00', 'firenov53@gmail.com', 'asdasfasd', 64, 4, 4, 'public/files/i1xFgcixuA1W7IL21OcFuDU7p4eJOTZkrOrrBxJA.png', 'public/files/a9Pc3YYtAWmI2V88I7cYP96k0eu4kwQKKZ6NGd4A.png', 'public/files/lSvXUMUiunT994EhTmQte7s0HgkXOhrvVHppQFXx.png', 'public/files/lSvXUMUiunT994EhTmQte7s0HgkXOhrvVHppQFXx.png', 'public/files/lSvXUMUiunT994EhTmQte7s0HgkXOhrvVHppQFXx.png', 'Lengkap', 'Proses Selesai', '12 November 2018', 'SI/11/12', '12 November 2018', 'SI/11/12'),  (37, 'GG-37', 'joka', 'haha', '1995-05-05 00:00:00', 'firenov53@gmail.com', 'asfasdasd', 65, 4, 4, 'public/files/UBR3CVUzplItKYbOiOmNk9yUBhoEvem1Xlx3yB44.png', 'public/files/RJt7rAtrZDZrj0hnYjFmMf1UY7nf7tcqdysohtR9.jpeg', 'public/files/sgfvxBJ1KHccxIGecyAfeunNMwzeEiDX5aXapZhr.jpeg', 'public/files/sgfvxBJ1KHccxIGecyAfeunNMwzeEiDX5aXapZhr.jpeg', 'public/files/sgfvxBJ1KHccxIGecyAfeunNMwzeEiDX5aXapZhr.jpeg', 'Lengkap', 'Menunggu Persetujuan Kepala UPT', NULL, NULL, NULL, NULL);  -- --------------------------------------------------------  --  -- Table structure for table `kecamatan`  --  CREATE TABLE `kecamatan` (  `id\_kecamatan` int(11) NOT NULL,  `nama` varchar(20) NOT NULL,  `alamat\_kecamatan` varchar(100) DEFAULT NULL,  `no\_telp\_kecamatan` varchar(15) DEFAULT NULL,  `kode\_pos\_kecamatan` varchar(10) DEFAULT NULL,  `website\_kecamatan` varchar(100) DEFAULT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `kecamatan`  --  INSERT INTO `kecamatan` (`id\_kecamatan`, `nama`, `alamat\_kecamatan`, `no\_telp\_kecamatan`, `kode\_pos\_kecamatan`, `website\_kecamatan`) VALUES  (1, 'Klojen', 'Jl. Surabaya 6', '0341-362468', '65111', 'https://kecklojen.malangkota.go.id/'),  (2, 'Blimbing', 'Jl. Raden Intan Kav. 14', '0341-491330', '65125', 'http://kecblimbing.malangkota.go.id/'),  (3, 'Kedungkandang', 'Jl. Mayjen Sungkono 59', '0341-752273', '65137', 'http://keckedungkandang.malangkota.go.id/'),  (4, 'Lowokwaru', 'Jl. Cengger Ayam I/12', '0341-493162', '65141', 'http://keclowokwaru.malangkota.go.id/'),  (5, 'Sukun', 'Jl. Keben I', '0341-362468', '65147', 'http://kecsukun.malangkota.go.id/');  -- --------------------------------------------------------  --  -- Table structure for table `makam`  --  CREATE TABLE `makam` (  `id\_makam` int(11) NOT NULL,  `nomor\_makam` int(11) DEFAULT NULL,  `kode\_makam` varchar(45) DEFAULT NULL,  `id\_blok` int(11) NOT NULL,  `lat` varchar(100) NOT NULL,  `lng` varchar(100) NOT NULL,  `created\_at` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `makam`  --  INSERT INTO `makam` (`id\_makam`, `nomor\_makam`, `kode\_makam`, `id\_blok`, `lat`, `lng`, `created\_at`) VALUES  (4, 12, 'XXX-B-12', 4, '-7.953922605129994', '112.6122323857984', '2018-10-15 08:16:17'),  (5, 341, 'XXX-A-341', 3, '-7.951656541897828', '112.6141133646347', '2018-10-15 08:16:17'),  (6, 12312, 'XXX-B-12312', 4, '-7.946656362112344', '112.61682445323652', '2018-10-15 08:16:17'),  (10, 2, 'GG-B-2', 10, '-7.952820052190582', '112.61318532031555', '2018-10-15 08:16:17'),  (11, 1, 'GG-C-1', 11, '-7.953796884276093', '112.61485488711116', '2018-10-15 08:16:17'),  (12, 12, 'A-1-12', 12, '-7.954972048346357', '112.61320043874832', '2018-10-15 08:16:17');  -- --------------------------------------------------------  --  -- Table structure for table `notifikasi`  --  CREATE TABLE `notifikasi` (  `id\_notifikasi` int(11) NOT NULL,  `id\_user` int(11) NOT NULL,  `content` varchar(100) NOT NULL,  `status` varchar(10) NOT NULL DEFAULT 'Unread'  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `notifikasi`  --  INSERT INTO `notifikasi` (`id\_notifikasi`, `id\_user`, `content`, `status`) VALUES  (1, 12, 'Entri perizinan baru menunggu persetujuan anda 1', 'read'),  (2, 12, 'Entri perizinan baru menunggu persetujuan anda 2', 'read'),  (3, 12, 'Entri perizinan baru menunggu persetujuan anda 3', 'read'),  (4, 12, 'Entri perizinan baru menunggu persetujuan anda 4', 'read'),  (11, 12, 'Entri perizinan baru menunggu persetujuan anda 5', 'read'),  (12, 12, 'Entri perizinan baru menunggu persetujuan anda 6', 'read'),  (13, 12, 'Entri perizinan baru menunggu persetujuan anda 7', 'read'),  (14, 12, 'Entri perizinan baru menunggu persetujuan anda 8', 'read'),  (15, 4, 'Entri perizinan baru menunggu persetujuan anda 9', 'read'),  (16, 4, 'Entri perizinan baru menunggu persetujuan anda 10', 'read');  -- --------------------------------------------------------  --  -- Table structure for table `penghuni\_makam`  --  CREATE TABLE `penghuni\_makam` (  `id\_penghuni\_makam` int(11) NOT NULL,  `nama` varchar(45) DEFAULT NULL,  `jenis\_kelamin` varchar(10) DEFAULT NULL,  `alamat\_terakhir` varchar(45) DEFAULT NULL,  `tanggal\_lahir\_alm` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,  `tanggal\_wafat` datetime DEFAULT NULL,  `tanggal\_pemakaman` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,  `status` varchar(20) DEFAULT NULL,  `id\_makam` int(11) NOT NULL,  `nama\_ahli\_waris` varchar(45) DEFAULT NULL,  `alamat\_ahli\_waris` varchar(45) DEFAULT NULL,  `nik\_ahli\_waris` varchar(45) DEFAULT NULL,  `kontak\_ahli\_waris` varchar(45) DEFAULT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `penghuni\_makam`  --  INSERT INTO `penghuni\_makam` (`id\_penghuni\_makam`, `nama`, `jenis\_kelamin`, `alamat\_terakhir`, `tanggal\_lahir\_alm`, `tanggal\_wafat`, `tanggal\_pemakaman`, `status`, `id\_makam`, `nama\_ahli\_waris`, `alamat\_ahli\_waris`, `nik\_ahli\_waris`, `kontak\_ahli\_waris`) VALUES  (24, 'dummy2', 'Perempuan', 'alamat dummy2', '2018-09-17 11:28:09', '2018-08-16 00:00:00', '2018-09-17 11:18:43', 'Diperpanjang', 11, 'ahli waris dummy 2', 'alamat dummy 2', '444444444444', '33333333333333'),  (25, 'Rizal', 'Laki-Laki', 'Malang', '2018-09-17 11:28:09', '2018-02-28 00:00:00', '2018-09-17 11:18:43', NULL, 6, 'Firman', 'Malang', '12312', '0822'),  (26, 'test', 'Laki-Laki', 'test', '2018-09-17 11:28:09', '2018-08-06 00:00:00', '2018-09-17 11:18:43', 'Expired', 10, 'test', 'test', '12', '123'),  (27, 'test', 'Laki-Laki', 'test', '2018-09-17 11:28:09', '2018-08-06 00:00:00', '2018-09-17 11:18:43', 'Expired', 10, 'test', 'test', '12', '123'),  (28, 'test', 'Laki-Laki', 'test', '2018-09-17 11:28:09', '2018-08-06 00:00:00', '2018-09-17 11:18:43', 'Expired', 10, 'test', 'test', '12', '123'),  (57, 'baru', 'Laki-Laki', 'baru', '2018-09-17 11:28:09', '2018-09-02 00:00:00', '2018-09-17 11:18:43', 'Ditimpa', 4, 'ahli waris dummy 1', 'adhliwarisdokumen1', '51345123', '1231523123'),  (58, 'dokumen baru', 'Laki-Laki', 'alamat dummy2', '2018-09-26 00:00:00', '2018-09-30 00:00:00', '2018-09-22 00:00:00', 'Expired', 12, 'ahli waris dummy 1', 'alamat dummy 2', '22222222222222', '1231523123'),  (59, 'dokumen1', 'Laki-Laki', NULL, '2018-09-27 00:00:00', '2018-09-27 00:00:00', '2018-09-07 00:00:00', 'Diperpanjang', 11, 'test', 'test', '22222222222222', '123123'),  (63, 'ihi', 'Laki-Laki', 'ihi', '2018-09-05 00:00:00', '2018-09-06 00:00:00', '2018-09-13 00:00:00', NULL, 10, 'aha', 'aha', '111', '111'),  (64, 'niini', 'Perempuan', 'asfsadasd', '1999-05-05 00:00:00', '2010-05-05 00:00:00', '2015-05-05 00:00:00', 'Diperpanjang', 10, 'nana', 'asdfasdsad', '1111111111', '1111111111111'),  (65, 'joka', 'Laki-Laki', 'asdgasd', '1999-11-21 00:00:00', '2018-05-12 00:00:00', '2018-07-12 00:00:00', 'Baru', 10, 'haha', '123124', '1111111111', '12123123'),  (66, 'yahi', 'Laki-Laki', 'asdasdad', '2000-12-15 00:00:00', '2015-03-31 00:00:00', '2015-03-04 00:00:00', 'Expired', 12, 'asdasd', 'asdasd', 'asdasd', 'asdasd');  -- --------------------------------------------------------  --  -- Table structure for table `polygon`  --  CREATE TABLE `polygon` (  `id\_polygon` int(11) NOT NULL,  `id\_blok` int(11) NOT NULL,  `lat` varchar(100) NOT NULL,  `lng` varchar(100) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `polygon`  --  INSERT INTO `polygon` (`id\_polygon`, `id\_blok`, `lat`, `lng`) VALUES  (14, 1, '-7.952832006046328', '112.61283826761473'),  (15, 1, '-7.953161401045218', '112.61292409830321'),  (16, 1, '-7.953023267690795', '112.6135356419586'),  (17, 1, '-7.952576990381623', '112.61362147264708'),  (23, 2, '-7.952343225882998', '112.61311185293425'),  (24, 2, '-7.952911698409032', '112.61266660623778'),  (25, 2, '-7.952869195817513', '112.61339616708983'),  (42, 4, '-7.953076395909545', '112.61282217436064'),  (43, 4, '-7.952428231170638', '112.61286508970488'),  (44, 4, '-7.952762939319455', '112.61353027754058'),  (50, 3, '-7.952948888172991', '112.61285436086882'),  (51, 3, '-7.952980765110841', '112.61325132780303'),  (52, 3, '-7.952449482489793', '112.61353027754058'),  (53, 3, '-7.952327287389608', '112.61291873388518'),  (54, 3, '-7.952566364725473', '112.61271488600005'),  (62, 5, '-7.953118898479589', '112.61268269949187'),  (63, 5, '-7.953357975354007', '112.61292409830321'),  (64, 5, '-7.953320785627145', '112.61327278547515'),  (65, 5, '-7.95306045744464', '112.61355709963073'),  (66, 5, '-7.952614180175923', '112.61362147264708'),  (67, 5, '-7.952576990381623', '112.61311721735228'),  (68, 5, '-7.952709811060108', '112.61268269949187'),  (84, 13, '-7.953097647195114', '112.61289191179503'),  (85, 13, '-7.952847944520112', '112.61273097925414'),  (86, 13, '-7.952454795319413', '112.61275243692626'),  (87, 13, '-7.952300723232575', '112.61305284433593'),  (88, 13, '-7.952316661727009', '112.61346590452422'),  (89, 13, '-7.952587616037477', '112.61364829473723'),  (90, 13, '-7.9528798214658', '112.61357855730284'),  (91, 13, '-7.953097647195114', '112.61342835359801'),  (107, 14, '-7.95237775928322', '112.61302334003676'),  (108, 14, '-7.952622149417128', '112.61289459400405'),  (109, 14, '-7.952919667644436', '112.61278194122542'),  (110, 14, '-7.952893103525773', '112.61328619652022'),  (111, 14, '-7.952595585279179', '112.61345249347914'),  (112, 14, '-7.952399010604997', '112.61318963699568'),  (113, 14, '-7.952298066816779', '112.613066255381'),  (114, 14, '-7.952260876993801', '112.61293214493026'),  (115, 14, '-7.95245213890461', '112.61285704307784'),  (116, 14, '-7.952553082654913', '112.61273902588118'),  (117, 14, '-7.952404323435265', '112.61264246635665'),  (118, 14, '-7.952584959623526', '112.61258882217635'),  (119, 14, '-7.952739031603599', '112.61258882217635'),  (120, 14, '-7.952935606114824', '112.6125620000862'),  (121, 14, '-7.953036549746125', '112.612706839373'),  (155, 19, '-7.952992718961905', '112.61280339889754'),  (156, 19, '-7.952716452092901', '112.61274975471724'),  (157, 19, '-7.953194606171558', '112.6133720272087'),  (158, 19, '-7.9528227086029935', '112.61358123951186'),  (159, 19, '-7.952594257072243', '112.61327010326613'),  (186, 23, '-7.952235641040576', '112.61300724678267'),  (187, 23, '-7.952936934320654', '112.61342030697097'),  (188, 23, '-7.952198451211938', '112.61333984070052'),  (215, 29, '-7.9529419815676485', '112.61319497628438'),  (216, 29, '-7.95278790966382', '112.61273900075184'),  (217, 29, '-7.952437263046268', '112.61295357747304'),  (218, 29, '-7.952702904450733', '112.61343101067769'),  (223, 31, '-7.95300573544195', '112.61331299348103'),  (224, 31, '-7.952368196252921', '112.61297503514515'),  (225, 31, '-7.952761345536622', '112.61285165353047'),  (226, 31, '-7.952649776183601', '112.61357048554646'),  (239, 33, '-7.95254378520527', '112.61292678051223'),  (240, 33, '-7.9529263086738435', '112.61269611053694'),  (241, 33, '-7.953128195916192', '112.61305552654494'),  (242, 33, '-7.9528891189078355', '112.61345785789717'),  (243, 33, '-7.952522533890999', '112.6133183830284');  -- --------------------------------------------------------  --  -- Table structure for table `role\_kecamatan`  --  CREATE TABLE `role\_kecamatan` (  `id\_role` int(11) NOT NULL,  `id\_user` int(11) NOT NULL,  `id\_kecamatan` int(11) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `role\_kecamatan`  --  INSERT INTO `role\_kecamatan` (`id\_role`, `id\_user`, `id\_kecamatan`) VALUES  (3, 13, 1),  (2, 13, 2),  (4, 13, 3),  (1, 13, 4),  (5, 13, 5),  (11, 15, 1),  (7, 15, 2),  (8, 15, 3),  (9, 15, 4),  (10, 15, 5);  -- --------------------------------------------------------  --  -- Table structure for table `role\_tpu`  --  CREATE TABLE `role\_tpu` (  `id\_role\_tpu` int(11) NOT NULL,  `id\_tpu` int(11) NOT NULL,  `id\_user` int(11) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `role\_tpu`  --  INSERT INTO `role\_tpu` (`id\_role\_tpu`, `id\_tpu`, `id\_user`) VALUES  (7, 1, 4),  (8, 1, 2),  (9, 4, 2),  (11, 2, 4),  (12, 4, 4),  (14, 3, 4),  (15, 5, 4),  (22, 4, 12),  (23, 1, 12),  (24, 10, 4),  (25, 13, 4),  (26, 3, 12),  (27, 2, 12),  (28, 5, 12),  (29, 10, 12),  (30, 13, 12);  -- --------------------------------------------------------  --  -- Table structure for table `tpu`  --  CREATE TABLE `tpu` (  `id\_tpu` int(11) NOT NULL,  `nama\_tpu` varchar(45) DEFAULT NULL,  `kode\_tpu` varchar(10) DEFAULT NULL,  `id\_kecamatan` int(11) DEFAULT NULL,  `lat` varchar(250) NOT NULL,  `lng` varchar(250) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `tpu`  --  INSERT INTO `tpu` (`id\_tpu`, `nama\_tpu`, `kode\_tpu`, `id\_kecamatan`, `lat`, `lng`) VALUES  (1, 'TPU Kasin Malang', 'KASIN', 2, '-7.951947154203484', '112.61320675285833'),  (2, 'TPU xxx', 'XXX', 4, '', ''),  (3, 'TPU Samaan', 'SAMAAN', 1, '', ''),  (4, 'TPU GreenGarden', 'GG', 4, '', ''),  (5, 'TPU A', 'A', 5, '', ''),  (10, 'TPU Baru', 'B', 4, '', ''),  (13, 'TPU Curug', 'C', 3, '', '');  -- --------------------------------------------------------  --  -- Table structure for table `user`  --  CREATE TABLE `user` (  `id\_user` int(11) NOT NULL,  `username` varchar(45) DEFAULT NULL,  `password` varchar(255) DEFAULT NULL,  `role` int(11) DEFAULT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Dumping data for table `user`  --  INSERT INTO `user` (`id\_user`, `username`, `password`, `role`) VALUES  (2, 'admin', '$2y$10$UCsJtdGuLt0ADZgAUSjHCeBpcXnL27D7DxpmIoTV1eWkrJmDjDcKi', 0),  (4, 'admintpu', '$2y$10$2fE3IJjHbAU5JGCrLgD6yex.ZZ.5WlHHTgLv6hZogT80wn4DbSXZK', 1),  (12, 'kepalaupt', '$2y$10$VFmRGgPBQOBRd7fCr3PsCuNA5MzinLxn/mHEsuTsgBYtYkvZCx7tG', 2),  (13, 'kepalakecamatan', '$2y$10$pJUqeai/.OH/mHi7yzvCcerC70Y5vmI56MLhLzQdrXXuyjQdH4jni', 4),  (14, 'kepaladinas', '$2y$10$BogmAKSq9N4FCKfL8Qt9K.7YHVsm8yJzx2QuEj47QkefEkXwV.XAS', 3),  (15, 'adminkecamatan', '$2y$10$.KtROaPHS7KpCa5pm86waujuH361uEUc8fWfDHEJQO5NfZWntELKm', 5);  --  -- Indexes for dumped tables  --  --  -- Indexes for table `blok\_makam`  --  ALTER TABLE `blok\_makam`  ADD PRIMARY KEY (`id\_blok`),  ADD KEY `fk\_blok\_makam\_tpu\_idx` (`id\_tpu`),  ADD KEY `id\_blok` (`id\_blok`);  --  -- Indexes for table `dokumen`  --  ALTER TABLE `dokumen`  ADD PRIMARY KEY (`id`),  ADD KEY `id\_kecamatan` (`id\_kecamatan`),  ADD KEY `id\_tpu` (`id\_tpu`);  --  -- Indexes for table `kecamatan`  --  ALTER TABLE `kecamatan`  ADD PRIMARY KEY (`id\_kecamatan`);  --  -- Indexes for table `makam`  --  ALTER TABLE `makam`  ADD PRIMARY KEY (`id\_makam`),  ADD KEY `fk\_makam\_blok\_makam1\_idx` (`id\_blok`);  --  -- Indexes for table `notifikasi`  --  ALTER TABLE `notifikasi`  ADD PRIMARY KEY (`id\_notifikasi`),  ADD KEY `id\_user` (`id\_user`);  --  -- Indexes for table `penghuni\_makam`  --  ALTER TABLE `penghuni\_makam`  ADD PRIMARY KEY (`id\_penghuni\_makam`),  ADD KEY `fk\_penghuni\_makam\_makam1\_idx` (`id\_makam`);  --  -- Indexes for table `polygon`  --  ALTER TABLE `polygon`  ADD PRIMARY KEY (`id\_polygon`),  ADD KEY `polygon\_ibfk\_1` (`id\_blok`);  --  -- Indexes for table `role\_kecamatan`  --  ALTER TABLE `role\_kecamatan`  ADD PRIMARY KEY (`id\_role`),  ADD KEY `id\_user` (`id\_user`,`id\_kecamatan`),  ADD KEY `id\_kecamatan` (`id\_kecamatan`);  --  -- Indexes for table `role\_tpu`  --  ALTER TABLE `role\_tpu`  ADD PRIMARY KEY (`id\_role\_tpu`),  ADD KEY `fk\_role\_tpu\_tpu1\_idx` (`id\_tpu`),  ADD KEY `fk\_role\_tpu\_user1\_idx` (`id\_user`);  --  -- Indexes for table `tpu`  --  ALTER TABLE `tpu`  ADD PRIMARY KEY (`id\_tpu`),  ADD KEY `id\_kecamatan` (`id\_kecamatan`);  --  -- Indexes for table `user`  --  ALTER TABLE `user`  ADD PRIMARY KEY (`id\_user`);  --  -- AUTO\_INCREMENT for dumped tables  --  --  -- AUTO\_INCREMENT for table `blok\_makam`  --  ALTER TABLE `blok\_makam`  MODIFY `id\_blok` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=34;  --  -- AUTO\_INCREMENT for table `dokumen`  --  ALTER TABLE `dokumen`  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=38;  --  -- AUTO\_INCREMENT for table `kecamatan`  --  ALTER TABLE `kecamatan`  MODIFY `id\_kecamatan` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;  --  -- AUTO\_INCREMENT for table `makam`  --  ALTER TABLE `makam`  MODIFY `id\_makam` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=13;  --  -- AUTO\_INCREMENT for table `notifikasi`  --  ALTER TABLE `notifikasi`  MODIFY `id\_notifikasi` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=17;  --  -- AUTO\_INCREMENT for table `penghuni\_makam`  --  ALTER TABLE `penghuni\_makam`  MODIFY `id\_penghuni\_makam` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=67;  --  -- AUTO\_INCREMENT for table `polygon`  --  ALTER TABLE `polygon`  MODIFY `id\_polygon` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=244;  --  -- AUTO\_INCREMENT for table `role\_kecamatan`  --  ALTER TABLE `role\_kecamatan`  MODIFY `id\_role` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=12;  --  -- AUTO\_INCREMENT for table `role\_tpu`  --  ALTER TABLE `role\_tpu`  MODIFY `id\_role\_tpu` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=31;  --  -- AUTO\_INCREMENT for table `tpu`  --  ALTER TABLE `tpu`  MODIFY `id\_tpu` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=15;  --  -- AUTO\_INCREMENT for table `user`  --  ALTER TABLE `user`  MODIFY `id\_user` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=17;  --  -- Constraints for dumped tables  --  --  -- Constraints for table `blok\_makam`  --  ALTER TABLE `blok\_makam`  ADD CONSTRAINT `blok\_makam\_ibfk\_1` FOREIGN KEY (`id\_tpu`) REFERENCES `tpu` (`id\_tpu`);  --  -- Constraints for table `dokumen`  --  ALTER TABLE `dokumen`  ADD CONSTRAINT `dokumen\_ibfk\_1` FOREIGN KEY (`id\_kecamatan`) REFERENCES `kecamatan` (`id\_kecamatan`),  ADD CONSTRAINT `dokumen\_ibfk\_2` FOREIGN KEY (`id\_tpu`) REFERENCES `tpu` (`id\_tpu`);  --  -- Constraints for table `makam`  --  ALTER TABLE `makam`  ADD CONSTRAINT `makam\_ibfk\_1` FOREIGN KEY (`id\_blok`) REFERENCES `blok\_makam` (`id\_blok`);  --  -- Constraints for table `penghuni\_makam`  --  ALTER TABLE `penghuni\_makam`  ADD CONSTRAINT `penghuni\_makam\_ibfk\_1` FOREIGN KEY (`id\_makam`) REFERENCES `makam` (`id\_makam`);  --  -- Constraints for table `polygon`  --  ALTER TABLE `polygon`  ADD CONSTRAINT `polygon\_ibfk\_1` FOREIGN KEY (`id\_blok`) REFERENCES `blok\_makam` (`id\_blok`);  --  -- Constraints for table `role\_kecamatan`  --  ALTER TABLE `role\_kecamatan`  ADD CONSTRAINT `role\_kecamatan\_ibfk\_1` FOREIGN KEY (`id\_kecamatan`) REFERENCES `kecamatan` (`id\_kecamatan`),  ADD CONSTRAINT `role\_kecamatan\_ibfk\_2` FOREIGN KEY (`id\_user`) REFERENCES `user` (`id\_user`);  --  -- Constraints for table `role\_tpu`  --  ALTER TABLE `role\_tpu`  ADD CONSTRAINT `role\_tpu\_ibfk\_1` FOREIGN KEY (`id\_tpu`) REFERENCES `tpu` (`id\_tpu`),  ADD CONSTRAINT `role\_tpu\_ibfk\_2` FOREIGN KEY (`id\_user`) REFERENCES `user` (`id\_user`);  --  -- Constraints for table `tpu`  --  ALTER TABLE `tpu`  ADD CONSTRAINT `tpu\_ibfk\_1` FOREIGN KEY (`id\_kecamatan`) REFERENCES `kecamatan` (`id\_kecamatan`);  COMMIT;  /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/; |

### Implementasi API *Gateway*

API *Gateway* adalah merupakan gerbang dari beberapa API, bertugas sebagai penjembatan antara aplikasi dengan *backend* *service*. API *Gateway* akan memberikan *request* yang sesuai dengan permintaan aplikasi. API berbentuk REST API menggunakan protokol http. Setiap *request* dengan method GET, POST, PUT, atau DELETE akan dilayani API *Gateway*. Berikut adalah source code API *Gateway*.

Tabel 5.10 Implementasi API Gateway

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147 | <?php  use Illuminate\Http\*Request*;  //ADMIN PUSAT  //login  Route::middleware('cors')->post('login', 'AuthController@login');  Route::middleware('cors')->post('signin', 'AuthController@signin');  Route::middleware('cors')->post('hash', 'AuthController@hash');  //change password  Route::middleware('cors')->post('change\_password', 'AuthController@change\_password');  //*user*  Route::middleware('cors')->get('*user*/view', 'AdminPusatController@view');  Route::middleware('cors')->post('create\_*user*', 'AdminPusatController@create\_*user*');  Route::middleware('cors')->put('update\_*user*/{id\_*user*}', 'AdminPusatController@update\_*user*');  Route::middleware('cors')->delete('delete\_*user*/{id\_*user*}', 'AdminPusatController@delete\_*user*');  Route::middleware('cors')->get('constraint\_*user*', 'AdminPusatController@constraint\_*user*');  //kecamatan  Route::middleware('cors')->get('kecamatan/constraint\_*user*', 'AdminPusatController@constraint\_*user*\_kecamatan');  Route::middleware('cors')->get('kecamatan/view', 'AdminKecamatan@view\_kecamatan');  Route::middleware('cors')->get('kecamatan/viewall', 'AdminKecamatan@view\_all\_kecamatan');  Route::middleware('cors')->post('kecamatan/create\_role', 'AdminPusatController@create\_role\_kecamatan');  Route::middleware('cors')->put('kecamatan/update\_role/{id\_role}', 'AdminPusatController@update\_role\_kecamatan');  Route::middleware('cors')->delete('kecamatan/delete\_role/{id\_role}', 'AdminPusatController@delete\_role\_kecamatan');  //tpu  Route::middleware('cors')->get('tpu/view\_by*User*', 'AdminTPUController@view\_tpu\_by*User*');  Route::middleware('cors')->get('tpu/view', 'AdminPusatController@view\_tpu');  Route::middleware('cors')->post('tpu/create', 'AdminPusatController@create\_tpu');  Route::middleware('cors')->put('tpu/edit/{id}', 'AdminPusatController@edit\_tpu');  Route::middleware('cors')->delete('tpu/delete/{id}', 'AdminPusatController@delete\_tpu');  //role\_tpu  Route::middleware('cors')->get('role\_tpu/view\_by*user*', 'AdminPusatController@view\_role\_by*user*');  Route::middleware('cors')->get('role\_tpu/view', 'AdminPusatController@view\_role\_tpu');  Route::middleware('cors')->post('create\_role\_tpu', 'AdminPusatController@create\_role\_tpu');  Route::middleware('cors')->put('update\_role\_tpu/{id\_role\_tpu}', 'AdminPusatController@update\_role\_tpu');  Route::middleware('cors')->delete('delete\_role\_tpu/{id\_role\_tpu}', 'AdminPusatController@delete\_role\_tpu');  //ADMIN TPU  //penghuni makam  Route::middleware('cors')->get('penghuni\_makam/view', 'AdminTPUController@view\_penghunimakam');  Route::middleware('cors')->post('penghuni\_makam/create', 'AdminTPUController@create\_penghunimakam');  Route::middleware('cors')->put('penghuni\_makam/update/{id}', 'AdminTPUController@update\_penghunimakam');  Route::middleware('cors')->delete('penghuni\_makam/delete/{id}', 'AdminTPUController@delete\_penghunimakam');  //makam  Route::middleware('cors')->get('makam/view', 'AdminTPUController@view\_makam');  Route::middleware('cors')->post('makam/create', 'AdminTPUController@create\_makam');  Route::middleware('cors')->put('makam/edit/{id}', 'AdminTPUController@edit\_makam');  Route::middleware('cors')->delete('makam/delete/{id}', 'AdminTPUController@delete\_makam');  Route::middleware('cors')->delete('makam/delete\_ex', 'AdminTPUController@delete\_expire\_makam');  //blok makam  Route::middleware('cors')->get('blok/view', 'AdminTPUController@view\_blok');  Route::middleware('cors')->post('blok/create', 'AdminTPUController@create\_blok');  Route::middleware('cors')->put('blok/edit/{id}', 'AdminTPUController@edit\_blok');  Route::middleware('cors')->delete('blok/delete/{id}', 'AdminTPUController@delete\_blok');  //polygon  Route::middleware('cors')->get('polygon/view', 'AdminTPUController@view\_polygon');  Route::middleware('cors')->post('polygon/create', 'AdminTPUController@create\_polygon');  Route::middleware('cors')->delete('polygon/delete/{id}', 'AdminTPUController@delete\_polygon');  //Registrasi  Route::middleware('cors')->post('dokumen/upload', 'AdminTPUController@upload');  //PENGGUNA  //Search  Route::middleware('cors')->get('blok/view\_search', 'PenggunaController@view\_search\_blok');  Route::middleware('cors')->get('penghuni\_makam/view\_search', 'PenggunaController@view\_search\_penghunimakam');  //mailing  Route::middleware('cors')->get('send', 'MailController@send');  //Dokumen  Route::middleware('cors')->get('dokumen/view', 'KepalaTPUController@view\_dokumen');  Route::middleware('cors')->post('dokumen/update', 'KepalaTPUController@update\_dokumen');  Route::middleware('cors')->post('dokumen/create', 'KepalaTPUController@create\_dokumen');  Route::middleware('cors')->get('dokumen/view\_picture', 'KepalaTPUController@view\_picture');  Route::middleware('cors')->get('dokumen/view\_siap\_cetak', 'AdminKecamatan@view\_dokumen\_siap\_cetak');  Route::middleware('cors')->get('dokumen/cetak', 'AdminKecamatan@cetak\_dokumen');  Route::middleware('cors')->get('dokumen/cetak\_surat\_permohonan', 'AdminKecamatan@cetak\_surat\_permohonan');  Route::middleware('cors')->get('dokumen/cetak\_surat\_perizinan', 'AdminKecamatan@cetak\_surat\_perizinan');  //Notifikasi  Route::middleware('cors')->get('notifikasi/view', 'NotifikasiController@view\_notifikasi');  Route::middleware('cors')->put('notifikasi/update', 'NotifikasiController@update\_notifikasi'); |

Kami menggunakan fungsi dari laravel dan memanggil file cors sebagai middlewarenya. file cors ini berisi handler *request* agar *request* dapat berjalan seusai dengan yang diinginkan, dan pada fungsi laravel kita akan mendefinisikan fungsi mana dan pada controller apa fungsi dibuat seperti pada baris 7. Pada baris tersebut login adalah path url untuk api tersebut dan AuthController adalah nama controller dan nama fungsinya adalah login.

Pada baris 7-9 berisi route fungsional dari fitur *authentication* beberapa fungsional tresebut adalah fungsi login yang di route ke auth controller, lalu fungsi signin pada controller auth contronnler dan juga fungsi hash pada authcontroller. Pada baris 12-13 adalah route fitur untuk perubahan password yang fitur yang bisa digunakan semua aktor. Pada baris 16-25 berisi route fungsional fitur *user* secara umum fungsional dari fitur *user* akan di route ke adminpusatcontroller dan disesuaikan dengan setiap fungsinyamasing-masing. Pada baris 28-40 adalah fungsional dari fitur kecamatan dan hak akses kecamatan yang akan melakukan route ke controller dan fungsinya masing-masing. Pada bari 44-53 berisi fungsional Fitur Manajemen TPU yang pada umumnya melakukan route ke adminpusatcontroller. Pada baris 56-65 merupakan fungsional untuk role TPU. Pada baris 69-77 adalah route untuk fungsi fitur penghuni makam. Pada baris 80-91 merupakan route untuk fungsi fitur makam. Pada baris 93-100 adalah route untuk fungsi fitur blok makam, pada baris 103-108 adalah route untuk fungsi polygon yang akan digunakan pada fitur blokmakam. Pada baris 110-112 adalah route untuk fungsi registrasi perizinan pada fitur registrasi. Pada baris 116-119 adalah route untuk fungsi search yang akan digunakan oleh beragam fitur. Pada baris 123 adalah route untuk fungsi mailing pada fitur mail. Pada baris 126-138 adalah route untuk fungsi fitur dokumen.

### Implementasi Fitur Manajemen TPU

Fitur Manajemen TPU akan berisi fungsi *create, read, update, delete* data tpu. Fitur akan melakukan query ke database MySQL untuk menjalankan fungsi. Berikut adalah source code untuk Fitur Manajemen TPU.

Tabel 5.11 Implementasi Fitur Manajemen TPU

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56 | function view\_tpu(){  $view = DB::tabel('tpu')  ->join('kecamatan', 'kecamatan.id\_kecamatan', '=', 'tpu.id\_kecamatan')  ->select('\*')  ->get();  return $view;    }  function create\_tpu(*Request* $*request*){  return Tpu::create($*request*->all());  }  function edit\_tpu(*Request* $*request*,$id){  $tpu = Tpu::findOrFail($id);  $tpu->update($*request*->all());  return response()->json($tpu);  }  public function delete\_tpu(*Request* $*request*, $id)  {  $penghuni\_makam = DB::tabel('penghuni\_makam')  ->join('makam', 'penghuni\_makam.id\_makam', '=', 'makam.id\_makam')  ->join('blok\_makam', 'makam.id\_blok', '=', 'blok\_makam.id\_blok')  ->join('tpu', 'blok\_makam.id\_tpu', '=', 'tpu.id\_tpu')  // ->select('penghunimakam.\*','blok\_makam.\*', 'tpu.\*')  ->where('tpu.id\_tpu','=',$id)  ->delete();  $makam = DB::tabel('makam')  ->join('blok\_makam', 'makam.id\_blok', '=', 'blok\_makam.id\_blok')  ->join('tpu', 'blok\_makam.id\_tpu', '=', 'tpu.id\_tpu')  // ->select('penghunimakam.\*','blok\_makam.\*', 'tpu.\*')  ->where('tpu.id\_tpu','=',$id)  ->delete();  $blokmakam = DB::tabel('blok\_makam')  ->join('tpu', 'blok\_makam.id\_tpu', '=', 'tpu.id\_tpu')  // ->select('penghunimakam.\*','blok\_makam.\*', 'tpu.\*')  ->where('tpu.id\_tpu','=',$id)  ->delete();    $tpu = Tpu::findOrFail($id);  $tpu->delete();  return response()->json($tpu);  } |

Fitur Manajemen TPU memiliki beberapa fungsi. Fungsi pertama adalah fungsi view TPU pada baris 1-10, dimana fungsi ini akan mengembalikan data TPU sekaligus lokasi kecamatannya dengan melakukan query select pada tabel tpu dengan join dengan kecamatan dengan tpu\_id sebagai referensinya. Kemudian fungsi create TPU yaitu fungsi untuk membuat TPU baru pada database pada baris 12-15. Fungsi ini akan mengambil seluruh *request* dan memasukkannya pada database. Fungsi edit TPU berada pada baris 17-24. Fungsi ini akan menyunting data TPU berdasarkan IDnya dengan mengambil seluruh *request* dan melakukan update pada database. Dan fungsi delete TPU berada pada baris 26-56. Fungsi akan menghapus TPU berdasarkan IDnya sekaligus seluruh data yang mereferensikan data tpu tersebut dengan melaukan query pada tabel tpu dan dilakukan join dengan tabel blok makam dan makam dengan mereferensikan id dari masing-masing. Dan di hapus sesuai dengan idnya.

### Implementasi Fitur Manajemen *User*

Fitur Manajemen *User* akan berisi fungsi *create, read, update, delete* data *user*. Fitur akan melakukan query ke database MySQL untuk menjalankan fungsi. Berikut adalah source code untuk Fitur Manajemen *User*

Tabel 5.12 Implementasi Fitur Manajemen User

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 | function view(){  $*user* = *User*::all();  return response()->json($*user*);  }  function create\_*user*(*Request* $*request*)  {  $*user*name = $*request*->input('*user*name');  $password = Hash::make($*request*->input('password'));  $role = $*request*->input('role');  //$id\_tpu = $*request*->input('id\_tpu');  //$id\_*user* = "";  $*user* = new *User*([  '*user*name' => $*user*name,  'password' => $password,  'role' => $role  ]);  $*user*->save();  }  function update\_*user*(*Request* $*request*, $id\_*user*)  {  $*user* = *User*::findOrFail($id\_*user*);  $*user*name = $*request*->input('*user*name');  $password = $*request*->input('password');  $role = $*request*->input('role');  $*user*->update([  '*user*name' => $*request*->input('*user*name'),  'password' => $*request*->input('password'),  'role' => $*request*->input('role')  ]);  $role\_tpu->save();  return response()->json($*user*name);  }  function delete\_*user*(*Request* $*request*, $id\_*user*)  {  $role = $*request*->input('role');  $*user* = DB::tabel('*user*')->select('id\_*user*')->where('id\_*user*', '=', $id\_*user*)->value('id\_*user*');  if(empty($*user*)){  $msg = ([  'msg' => "Data *User* tidak ditemukan !"  ]);  } else {  if($role==1||$role==2){  $role\_tpu = DB::tabel('role\_tpu')->select('id\_*user*')->where('id\_*user*', '=', $id\_*user*)->value('id\_*user*');  if(empty($role\_tpu)){  $*user* = *User*::findOrFail($id\_*user*);  $*user*->delete();  $msg = ([  'msg' => "*User* berhasil dihapus"  ]);  } else {  $msg = ([  'msg' => "Anda harus menghapus hak akses *user* terlebih dahulu !"  ]);  }  }else if($role==5||$role==4){  $role\_kecamatan = DB::tabel('role\_kecamatan')->select('id\_*user*')->where('id\_*user*', '=', $id\_*user*)->value('id\_*user*');  if(empty($role\_kecamatan)){  $*user* = *User*::findOrFail($id\_*user*);  $*user*->delete();  $msg = ([  'msg' => "*User* berhasil dihapus"  ]);  } else {  $msg = ([  'msg' => "Anda harus menghapus hak akses *user* terlebih dahulu !"  ]);  }  }else{  $*user* = *User*::findOrFail($id\_*user*);  $*user*->delete();  $msg = ([  'msg' => "*User* berhasil dihapus"  ]);  }    }  return response()->json($msg);  } |

Dalam fitur manajemen *user* akan terdapat beberapa fungsi. Fungsi pertama dalah fungsi view pada baris 1-5 yang akan mengembalikan hasil query tabel *user* *user* dalam bentuk json. Fungsi berikutnya adalah fungsi create *user* pada baris 7-22, fungsi create *user* membuat entri *user* baru sekaligus dengan *user*name, password. Entri data baru dilakukan dengan melakukan query insert dan memasukan seluruh data *request*, akan tetapi khusus untuk password akan melakukan proses *hashing* terlebih dahulu. Fungsi edit adalah fungsi untuk menyunting *user* berdasarkan idnya. Fungsi tersebut dapat ditemukan pada baris 24-41. Pada baris 43-96 adalah fungsi delete *user* dengan melakukan query delete pada tabel database. Sebelum query delete dijalankan akan memasuki proses pemeriksaan terlebih dahulu apakah memiliki hak akses atau tidak jika memiliki hak akses delete maka query tidak akan dijalankan.

### Implementasi Fitur Manajemen *User* Access

Fitur Manajemen *User* access akan berisi fungsi *create, read, update, delete* data role TPU. Fitur akan melakukan query ke database MySQL dan melayani hak akses *user* yang memiliki role sebagai adminTPU atau kepada Kepala UPT Pemakaman berdasarkan hak akses TPUnya masing-masing. Berikut adalah source code untuk Fitur Manajemen *User* access.

Tabel 5.12 Implementasi Fitur Manajemen User Access

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | function view\_role\_tpu(){  $role\_tpu = Role\_tpu::all();  return response()->json($role\_tpu);  }  function create\_role\_tpu(*Request* $*request*)  {  $id\_tpu = $*request*->input('id\_tpu');  $id\_*user* = $*request*->input('id\_*user*');  $role\_tpu = new Role\_tpu([  'id\_tpu' => $id\_tpu,  'id\_*user*' => $id\_*user*  ]);  $role\_tpu->save();  return response()->json(['msg' => "Hak Akses Berhasil dibuat!"]);  }  function update\_role\_tpu(*Request* $*request*, $id\_role\_tpu)  {  $role\_tpu = Role\_tpu::findOrFail($id\_role\_tpu);    $role\_tpu->update($*request*->all());  return response()->json(['msg' => "Hak Akses Berhasil di update!"]);  }  function delete\_role\_tpu(*Request* $*request*, $id\_role\_tpu)  {  $role\_tpu = Role\_tpu::findOrFail($id\_role\_tpu);  $role\_tpu->delete();  return response()->json(['msg' => "Hak Akses Berhasil dihapus!"]);  } |

Fitur Manajemen *User* Access memiliki beberapa fungsi. Fungsi view\_role pada baris 1-4 akan mengembalikan seluruh data pada role untuk menampilkan seluruh datapada role dengan melakukan query pada tabel role\_tpu. Fungsi create role pada baris 6-20 digunakan untuk membuat role baru berdasarkan id\_*user* dan id\_tpu dengan melakukan query insert pada tabel role\_tpu. Pada baris 22-30 terdapat fungsi update role yang akan menyunting data role berdasarkan id\_role. Penyuntingan dilakukan dengan melakukan query update pada tabel role\_tpu. Pada baris 32-39 Delete role akan menghapus data berdasarkan id rolenya.

### Implementasi Fitur Manajemen Kecamatan Access

Fitur manajemen kecamatan access akan berisi fungsi *create, read, update, delete* data role kecamatan. Fitur akan melakukan query ke database MySQL dan melayani hak akses *user* yang memiliki role sebagai admin Kecamatan atau kepala kecamatan berdasarkan hak akses kecamatannya nya masing-masing. Berikut adalah source code untuk Fitur manajemen kecamatan access.

Tabel 5.13 Implementasi Fitur Manajemen Kecamatan Access

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | function create\_role\_kecamatan(*Request* $*request*)  {  $id\_tpu = $*request*->input('id\_tpu');  $id\_*user* = $*request*->input('id\_*user*');  $role = Role\_kecamatan::create($*request*->all());  return response()->json(['msg' => "Hak Akses Berhasil dibuat!"]);  }  function update\_role\_kecamatan(*Request* $*request*, $id\_role)  {  $role = Role\_kecamatan::findOrFail($id\_role);    $role->update($*request*->all());  return response()->json(['msg' => "Hak Akses Berhasil di update!"]);  }  function delete\_role\_kecamatan(*Request* $*request*, $id\_role)  {  $role = Role\_kecamatan::findOrFail($id\_role);  $role ->delete();  return response()->json(['msg' => "Hak Akses Berhasil dihapus!"]);  } |

Fitur Manajemen Kecamatan Access memiliki beberapa fungsi. Pada baris 1-10 terdapat fungsi create role untuk membuat role baru berdasarkan id\_*user* dan id\_kecamatan dengan query insert. Pada baris 12-20 terdapat fungsi update role akan menyunting data role berdasarkan id\_role dengan query update pada tabel. Pada baris 22-29 Delete role akan menghapus data berdasarkan id rolenya dengan query delete pada tabel.

### Implementasi FItur Registrasi Perijinan

FItur Registrasi Perijinan akan berisi fungsi *create, read, update, delete*  data Dokumen. Fitur akan melakukan query ke database MySQL dan melayani bagaimana dan dimana *resource* file yang di upload disimpan. Registrasi perizinan memiliki fungsi upload yang digunakan untuk upload file registrasi perizinan dan penambahan data baru untuk entri database dokumen. Berikut adalah source code untuk Fitur Registrasi Perijinan.

Tabel 5.14 Implementasi FItur Registrasi Perijinan

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71 | function upload(*Request* $*request*){  $ktp = $*request*->file('file\_ktp');  $kk = $*request*->file('file\_kk');  $sk = $*request*->file('file\_sk');  $sk\_lama = $*request*->file('file\_sk\_lama');  $surat\_izin = $*request*->file('file\_surat\_izin');  $progress= 'Menunggu Persetujuan Kepala UPT';  $dokumen= 'Dokumen Kurang';  $status1 = false; $status2 = false; $status3 = false; $status4 = false;    if(!empty($ktp)) {  $status1= true;  }  if(!empty($kk)) {  $status2 = true;  }  if(!empty($surat\_izin)) {  $status3 = true;  }  if(!empty($sk)) {  $status4 = true;  }  if($status1 && $status2 && $status3 && $status3){  $path\_ktp = $ktp->store('public/files');  $path\_kk = $kk->store('public/files');  $path\_surat\_izin = $surat\_izin->store('public/files');  $path\_sk = $surat\_izin->store('public/files');  $path\_sk\_lama;    if(!empty($sk\_lama)){  $path\_sk\_lama = $surat\_izin->store('public/files');  }else{  $path\_sk\_lama="Kosong";  }  Dokumen::create(array(    'nama\_almarhum' => $*request*->input('nama\_almarhum'),  'nama\_pewaris' => $*request*->input('nama\_pewaris'),  'email' => $*request*->input('email'),  'pekerjaan\_ahli\_waris' => $*request*->input('pekerjaan\_ahli\_waris'),  'id\_penghuni\_makam' => $*request*->input('id\_penghuni\_makam'),  'id\_tpu' => $*request*->input('id\_tpu'),  'id\_kecamatan' => $*request*->input('id\_kecamatan'),  'tgllhr\_ahli\_waris' => $*request*->input('tgllhr\_ahli\_waris'),  'file\_ktp' => $path\_ktp,  'file\_kk' => $path\_kk,  'file\_surat\_izin' => $path\_surat\_izin,  'file\_sk' => $path\_sk,  'file\_sk\_lama' => $path\_sk\_lama,  'status' => $progress ,  'kelengkapan\_dokumen' => $dokumen,  ));  return response()->json('upload sukses!');  } else{  return response()->json('upload gagal!');  }  } |

Pada baris 1-11 adalah deklarasi variable baru yang berisi semua data *request* dan variabel untuk mendefinisikan status salah satu kolom dokumen dan juga status ada atau tidaknya lampiran suatu file. Baris 15-26 akan memeriksa lampiran. Sedangkan baris 27-71 akan berisi action ketika seluruh status dokumen lengkap. Ketika dokumen lengkap maka akan dilakukan query create untuk membuat entri baru dan untuk dokumen file akan di simpan kedalam direktori storage dan pathnya akan disimpan pada database

### Implementasi Fitur Dokumen

Fitur Dokumen berisi fungsi untuk mencetak sebuah dokumen dengan tipe data pdf. Fitur akan melakukan query ke database MySQL untuk menjalankan fungsi. Berikut adalah source code untuk Fitur Dokumen.

Tabel 5.15 Implementasi Fitur Dokumen

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407 | function cetak\_surat\_permohonan(*Request* $*request*){  $no\_surat\_permohonan = $*request*->input("no\_surat\_permohonan");  $tanggal\_sekarang = $*request*->input("tanggal\_sekarang");  $nama\_ahli\_waris = $*request*->input("nama\_ahli\_waris");  $alamat\_ahli\_waris = $*request*->input("alamat\_ahli\_waris");  $ttl\_ahli\_waris = $this->tgl\_indo($*request*->input("ttl\_ahli\_waris"));  $tanggal\_wafat = $this->tgl\_indo($*request*->input("tanggal\_wafat"));  $nama\_almarhum = $*request*->input("nama\_almarhum");  $ttl\_almarhum = $this->tgl\_indo($*request*->input("ttl\_almarhum"));  $umur\_almarhum = $this->get\_age($*request*->input("ttl\_almarhum"));  $jenis\_kelamin\_almarhum = $*request*->input("jenis\_kelamin\_almarhum");  $tpu\_almarhum = $*request*->input("tpu\_almarhum");  $alamat\_almarhum = $*request*->input("alamat\_almarhum");  $tanggal\_pemakaman = $this->tgl\_indo($*request*->input("tanggal\_pemakaman"));  $blok\_almarhum = $*request*->input("blok\_almarhum");  Fpdf::AddPage();  Fpdf::Image( storage\_path("app/logo\_kota\_Malang.png"),30,8,20,25);  Fpdf::Cell(45);  Fpdf::SetFont("Times","B","14");  Fpdf::Cell(133,5,"PEMERINTAH KOTA MALANG",0,1,"C");  Fpdf::Cell(45);  Fpdf::Cell(133,5,"DINAS PERUMAHAN DAN KAWASAN PERMUKIMAN",0,1,"C");  Fpdf::Cell(45);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(133,5,"Jl. Bingkil No. 1 Telp (0341) 369377 Fax (0341) 344872",0,1,"C");  Fpdf::Cell(45);  Fpdf::Cell(133,5,"www.dpkp.Malangkota.go.id, email:dpkp@Malangkota.go.id/dpkp.2017@gmail.com",0,1,"C");  Fpdf::Cell(45);  Fpdf::Cell(120,5,"MALANG",0,0,"C");  Fpdf::Cell(13,5,"Kode Pos 65148",0,1,"R");  Fpdf::SetLineWidth(1);  Fpdf::Line(20,36,190,37);  Fpdf::SetLineWidth(0);  Fpdf::Line(20,37,190,37);  Fpdf::Cell(27);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(150,12,"Malang, $tanggal\_sekarang",0,1,"R");  Fpdf::Cell(10);  Fpdf::Cell(15,5,"Nomor",0,0,"L");  Fpdf::Cell(2,5,":",0,0,"L");  Fpdf::Cell(40,5,"$no\_surat\_permohonan",0,1,"L");  Fpdf::Cell(10);  Fpdf::Cell(15,5,"Sifat",0,0,"L");  Fpdf::Cell(2,5,":",0,0,"L");  Fpdf::Cell(40,5,"Biasa",0,1,"L");  Fpdf::Cell(10);  Fpdf::Cell(15,5,"Lampiran",0,0,"L");  Fpdf::Cell(2,5,":",0,0,"L");  Fpdf::Cell(40,5,"--",0,1,"L");  Fpdf::Cell(10);  Fpdf::Cell(15,5,"Hal",0,0,"L");  Fpdf::Cell(2,5,":",0,0,"L");  Fpdf::MultiCell(60,5,"Rekomendasi Perpanjangan Ijin Penggunaan Tanah Makam",0,"L",false);  Fpdf::Cell(35);  Fpdf::Cell(15,8,"Berkaitan dengan surat permohonan Ahli Waris :",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"1.",0,0,"L");  Fpdf::Cell(40,5,"Tanggal",0,0,"L");  Fpdf::Cell(20,5,": $ttl\_ahli\_waris",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"2.",0,0,"L");  Fpdf::Cell(40,5,"Nama",0,0,"L");  Fpdf::Cell(20,5,": $nama\_ahli\_waris",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"3.",0,0,"L");  Fpdf::Cell(40,5,"Alamat",0,0,"L");  Fpdf::Cell(0,5,": $alamat\_ahli\_waris",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"4.",0,0,"L");  Fpdf::Cell(40,5,"Perihal",0,0,"L");  Fpdf::Cell(0,5,": Rekomendasi Perpanjangan Ijin Penggunaan Tanah Makam",0,1,"L");  Fpdf::Cell(0,3,"",0,1,"L");  Fpdf::Cell(27);  Fpdf::MultiCell(150,5,"Berdasarkan Peraturan Daerah Kota Malang Nomor 3 Tahun 2006 tentang Penyelenggaraan Pemakaman, Peraturan Walikota Malang Nomor 28 Tahun 2016 tentang Kedudukan, Susunan, Organisasi, Tugas dan Fungsi Serta Tata Kerja Dinas Perumahan dan Kawasan Permukiman serta Peraturan Walikota Malanmg Nomor 12 Tahun 2015 tentang Tata Cara Pelayanan Perijinan di Kecamatan, telah dilakukan pemeriksaan dan pengecekan terhadap permohonan Ahli Waris, permohonan tersebut telah memenuhi persyaratan yang diatur dalam peraturan perundangan yang berlaku, sehingga dapat diberikan rekomendasi Perpanjangan Ijin Penggunaan Tanah Makam kepada Pemohon dengan data sebagai berikut :",0,"J",false);  Fpdf::Cell(0,3,"",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"1.",0,0,"L");  Fpdf::Cell(40,5,"Nama",0,0,"L");  Fpdf::Cell(20,5,": $nama\_almarhum",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"2.",0,0,"L");  Fpdf::Cell(40,5,"Tanggal Lahir",0,0,"L");  Fpdf::Cell(20,5,": $ttl\_almarhum",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"3.",0,0,"L");  Fpdf::Cell(40,5,"Umur",0,0,"L");  Fpdf::Cell(0,5,": $umur\_almarhum Tahun",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"4.",0,0,"L");  Fpdf::Cell(40,5,"Jenis Kelamin",0,0,"L");  Fpdf::Cell(0,5,": $jenis\_kelamin\_almarhum",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"5.",0,0,"L");  Fpdf::Cell(40,5,"Alamat",0,0,"L");  Fpdf::Cell(0,5,": $alamat\_almarhum",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"6.",0,0,"L");  Fpdf::Cell(40,5,"Tanggal Pemakaman",0,0,"L");  Fpdf::Cell(0,5,": $tanggal\_pemakaman",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"7.",0,0,"L");  Fpdf::Cell(40,5,"Lokasi Pemakaman",0,0,"L");  Fpdf::Cell(0,5,": $tpu\_almarhum",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(5,5,"8.",0,0,"L");  Fpdf::Cell(40,5,"Blok",0,0,"L");  Fpdf::Cell(0,5,": $blok\_almarhum",0,1,"L");  Fpdf::Cell(0,4,"",0,1,"L");  Fpdf::Cell(0,6,"",0,1,"L");  Fpdf::Cell(27);  Fpdf::Cell(150,5,"Demikian rekomendasi ini dibuat untuk dapatnya diproses lebih lanjut.",0,1,"R");  Fpdf::Cell(0,6,"",0,1,"L");  Fpdf::Cell(107);  Fpdf::MultiCell(70,5,"a.n.Plt. KEPALA DINAS PERUMAHAN DAN KAWASAN PERMUKIMAN, SEKRETARIS",0,"C",false);  Fpdf::Cell(107);  Fpdf::Cell(70,30,"",0,1,"L");  //Fpdf::Cell(107);  Fpdf::Image( storage\_path("app/ttd.png"),143,223,20,25);  Fpdf::Cell(107);  Fpdf::SetFont("Times","U","10");  Fpdf::Cell(70,5,"Dra. NUNUK SRI RUSGIYANTI",0,1,"C");  Fpdf::Cell(107);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(70,5,"Pembina Tingkat I",0,1,"C");  Fpdf::Cell(107);  Fpdf::Cell(70,5,"NIP.19640919 199003 2 005",0,1,"C");  Fpdf::Output("D","Surat\_Permohonan\_$nama\_almarhum.pdf");  }  function cetak\_surat\_perizinan(*Request* $*request*){  $id\_kecamatan = $*request*->input("id\_kecamatan");  $view = DB::tabel("kecamatan")  ->where("id\_kecamatan","=",$id\_kecamatan)  ->select("\*")  ->get();  $nama\_kecamatan = strtoupper($view[0]->nama);  $nama\_kecamatan\_normal = $view[0]->nama;  $alamat\_kecamatan = $view[0]->alamat\_kecamatan;  $no\_telp\_kecamatan = $view[0]->no\_telp\_kecamatan;  $kode\_pos\_kecamatan = $view[0]->kode\_pos\_kecamatan;  $website\_kecamatan = $view[0]->website\_kecamatan;  $no\_surat\_perizinan = $*request*->input("no\_surat\_perizinan");  $no\_surat\_permohonan = $*request*->input("no\_surat\_permohonan");  $tanggal\_surat\_permohonan = $*request*->input("tanggal\_surat\_permohonan");  $tanggal\_sekarang = $*request*->input("tanggal\_sekarang");  $nama\_ahli\_waris = $*request*->input("nama\_ahli\_waris");  $alamat\_ahli\_waris = $*request*->input("alamat\_ahli\_waris");  $nama\_almarhum = $*request*->input("nama\_almarhum");  $ttl\_almarhum = $this->tgl\_indo($*request*->input("ttl\_almarhum"));  $jenis\_kelamin\_almarhum = $*request*->input("jenis\_kelamin\_almarhum");  $tpu\_almarhum = $*request*->input("tpu\_almarhum");  $tanggal\_pemakaman = $this->tgl\_indo($*request*->input("tanggal\_pemakaman"));  $blok\_almarhum = $*request*->input("blok\_almarhum");  Fpdf::AddPage();  Fpdf::Image(storage\_path("app/logo\_kota\_Malang.png"),30,4.5,20,25);  Fpdf::Cell(45);  Fpdf::SetFont("Times","B","10");  Fpdf::Cell(133,4.5,"PEMERINTAH KOTA MALANG",0,1,"C");  Fpdf::Cell(45);  Fpdf::SetFont("Times","B","14");  Fpdf::Cell(133,4.5,"KECAMATAN $nama\_kecamatan",0,1,"C");  Fpdf::Cell(45);  Fpdf::SetFont("Times","B","10");  Fpdf::Cell(133,4.5,"$alamat\_kecamatan Telepon $no\_telp\_kecamatan",0,1,"C");  Fpdf::Cell(45);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(133,4.5,"$website\_kecamatan",0,1,"C");  Fpdf::Cell(45);  Fpdf::Cell(120,4.5,"",0,0,"C");  Fpdf::Cell(13,4.5,"Kode Pos $kode\_pos\_kecamatan",0,1,"R");  Fpdf::SetLineWidth(1);  Fpdf::Line(21,33,190,33);  Fpdf::SetLineWidth(0);  Fpdf::Line(21,34,190,34);  Fpdf::Cell(0,7,"",0,1,"R");  Fpdf::SetFont("Times","B","10");  Fpdf::Cell(0,4.5,"KEPUTUSAN CAMAT $nama\_kecamatan KOTA MALANG",0,1,"C");  Fpdf::Cell(0,4.5,"NOMOR : $no\_surat\_perizinan",0,1,"C");  Fpdf::Cell(0,4.5,"TENTANG",0,1,"C");  Fpdf::Cell(0,4.5,"IZIN PENGGUNAAN TANAH MAKAM/MAKAM TUMPANGAN",0,1,"C");  Fpdf::Cell(0,4.5,"CAMAT $nama\_kecamatan,",0,1,"C");  Fpdf::Cell(0,3,"",0,1,"C");  Fpdf::Cell(10);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(25,4.5,"Menimbang",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"a.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Bahwa PERMOHONAN lain Penggunaan Tanah Makam/Makam Tumpangan dari Sdr. $nama\_ahli\_waris Tanggal 3 Juli 2017 Walikota Malang Nomor 12 Tahun 2015 tentang Tata Cara Pelayanan Perizinan di Kecamatan;",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(25,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"b.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Bahwa sesuai rekomendasi Dinas Perumahan Dan Kawasan Permukiman No. $no\_surat\_permohonan Tanggal $tanggal\_surat\_permohonan perihal Perpanjangan Ijin Penggunaan Tanah Makam;",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(25,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"c.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Bahwa berdasarkan pertimbangan sebagaimana dimaksud dalam huruf a dan b perlu menetapkan Keputusan Camat $nama\_kecamatan\_normal tentang Izin Penggunaan Tanah Makam/Makam Tumpangan",0,"J",false);  Fpdf::Cell(0,3,"",0,1,"C");  Fpdf::Cell(10);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(25,4.5,"Mengingat",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"1.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Undang-Undang Nomor 23 Tahun 2014 tentang Pemerintahan Daerah sebagaimana telah diubah dengan Undang-Undang Nomor 2 Tahun 2015;",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(25,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"2.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Peraturan Daerah Kota Malang Nomor 3 Tahun 2006 tentang Penyelenggaraan Pemakaman;",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(25,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"3.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Peraturan Daerah Kota Malang Nomor 8 Tahun 2008 tentang Organisasi dan Tata Kerja Kecamatan dan Kelurahan;",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(25,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"4.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Peraturan Walikota Nomor 03 Tahun 2012 tentang Pelimpahan Sebagian Kewenangan Walikota kepada Camat;",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(25,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"",0,0,"L");  Fpdf::Cell(5,4.5,"5.",0,0,"L");  Fpdf::MultiCell(140,4.5,"Peraturan Walikota Malang Nomor 12 Tahun 2015 tentang Tata Cara Pelayanan Perizinan di Kecamatan;",0,"J",false);  Fpdf::Cell(0,3,"",0,1,"R");  Fpdf::SetFont("Times","B","10");  Fpdf::Cell(0,4.5,"Memutuskan : ",0,1,"C");  Fpdf::Cell(10);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(20,4.5,"Menetapkan",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::MultiCell(140,4.5,"KEPUTUSAN CAMAT $nama\_kecamatan TENTANG IZIN PENGGUNAAN TANAH MAKAM/TANAH TUMPANGAN",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(20,4.5,"KESATU",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::MultiCell(140,4.5,"Memberikan Rekomendasi Perpanjang Ijin Penggunaan Tanah Makam kepada :",0,"J",false);  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Nama",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$nama\_ahli\_waris",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Alamat",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$alamat\_ahli\_waris",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Untuk Makam Jenazah : ",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Nama",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$nama\_almarhum",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Jenis Kelamin",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$jenis\_kelamin\_almarhum",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Tanggal Lahir",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$ttl\_almarhum",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Tanggal Pemakaman",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$tanggal\_pemakaman",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Di Tempat Pemakaman",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$tpu\_almarhum",0,1,"L");  Fpdf::Cell(35);  Fpdf::Cell(40,4.5,"Blok",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::Cell(5,4.5,"$blok\_almarhum",0,1,"L");  Fpdf::Cell(10);  Fpdf::Cell(20,4.5,"KEDUA",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::MultiCell(140,4.5,"Masa berlaku ijin 2 (DUA) tahun dan dapat dilaksanakan perpanjangan/daftar ulang.",0,"J",false);  Fpdf::Cell(10);  Fpdf::Cell(20,4.5,"KETIGA",0,0,"L");  Fpdf::Cell(5,4.5,":",0,0,"L");  Fpdf::MultiCell(140,4.5,"Keputusan Camat $nama\_kecamatan\_normal ini mulai berlaku pada tanggal ditetapkan.",0,"J",false);    Fpdf::Cell(0,4.5,"",0,1,"L");  Fpdf::Cell(115);  Fpdf::Cell(70,4.5,"Ditetapkan di Malang,",0,1,"L");  Fpdf::Cell(115);  Fpdf::MultiCell(70,4.5,"Pada Tanggal, $tanggal\_sekarang",0,"L",false);  Fpdf::Cell(115);  Fpdf::MultiCell(70,4.5,"CAMAT $nama\_kecamatan KOTA MALANG,",0,"L",false);  Fpdf::Cell(115);  Fpdf::Cell(70,25,"",0,1,"L");  //Fpdf::Cell(107);  Fpdf::Image( storage\_path("app/ttd.png"),133,235,20,20);  Fpdf::Cell(10);  Fpdf::Cell(105,4.5,"Tembusan : ",0,0,"L");  Fpdf::SetFont("Times","U","10");  Fpdf::Cell(70,4.5,"SINARNI, SIP. MM.",0,1,"L");  Fpdf::Cell(10);  Fpdf::SetFont("Times","","10");  Fpdf::Cell(17,4.5,"Yth. Bpk. :",0,0,"L");  Fpdf::Cell(4,4.5,"1. ",0,0,"L");  Fpdf::Cell(84,4.5,"Walikota Malang (sebagai laporan);",0,0,"L");  Fpdf::Cell(70,4.5,"Pembina Tingkat I",0,1,"L");  Fpdf::Cell(10);  Fpdf::Cell(17,4.5,"",0,0,"L");  Fpdf::Cell(4,4.5,"2. ",0,0,"L");  Fpdf::Cell(84,4.5,"Kepala Dinas Dan Kawasan Pemukiman Kota Malang;",0,0,"L");  Fpdf::Cell(70,4.5,"NIP.19640919 199003 2 005",0,1,"L");  Fpdf::Cell(10);  Fpdf::Cell(17,4.5,"",0,0,"L");  Fpdf::Cell(4,4.5,"3. ",0,0,"L");  Fpdf::Cell(84,4.5,"Kepala Satuan Polisi Pamong Praja Kota Malang.",0,1,"L");  Fpdf::Output("D","Surat\_Izin\_$nama\_almarhum.pdf");  } |

Dalam implementasi generate file digunakan *library* tambahan khusus untuk generate file yaitu *library* Fpdf. Fitur dokumen adalah fungsi cetak surat yang melakukan generate file berdasarkan data-data yang ada pada database. Generate file dibuat berdasarkan template yang sudah ada dimana beberapa parameter akan menjadi variabel yang dapat diubah ubah berdasarkan query database yang sesuai dengan dokumen yang ingin dicetak.

### Implementasi Fitur Manajemen Makam

Fitur Manajemen Makam berisi *create, read, update, delete* data Makam dan juga fungsi untuk menyesuaikan *request* data spesifik berdasarkan role login dan hak akses yang di lakukan *user*. Fitur akan melakukan query ke database MySQL untuk menjalankan fungsi. Berikut adalah source code untuk Fitur Manajemen Makam.

Tabel 5.16 Implementasi Fitur Manajemen Makam

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | function view\_makam(*Request* $*request*){  $id\_*user*=$*request*->input('id\_*user*');  $view = DB::tabel('makam')  ->join('blok\_makam', 'makam.id\_blok', '=', 'blok\_makam.id\_blok')  ->join('tpu', 'tpu.id\_tpu', '=', 'blok\_makam.id\_tpu')  ->join('role\_tpu','role\_tpu.id\_tpu','=', 'blok\_makam.id\_tpu')  ->select('blok\_makam.\*', 'makam.\*','tpu.\*')  ->where('role\_tpu.id\_*user*','=',$id\_*user*)  ->get();  return response()->json($view);  }  function create\_makam(*Request* $*request*){  return Makam::create($*request*->all());  }  function edit\_makam(*Request* $*request*,$id){  $makam = Makam::findOrFail($id);  $makam->update($*request*->all());  return response()->json($makam);  }  function delete\_makam(*Request* $*request*,$id){  $penghuni\_makam = DB::tabel('penghuni\_makam')  ->where('penghuni\_makam.id\_makam','=',$id)  ->delete();  $makam = Makam::findOrFail($id);  $makam->delete();  return response()->json($makam);  }  function delete\_expire\_makam(*Request* $*request*){  $makam = DB::tabel('makam')  ->leftjoin('penghuni\_makam', 'penghuni\_makam.id\_makam', '=', 'makam.id\_makam')  ->select('makam.\*','penghuni\_makam.\*')  ->where('created\_at','<=',NOW()->subYears(3))  ->delete();  return response()->json($makam);  } |

Fitur Manajemen Makam berisi beberapa fungsi. Baris 1-14 adalah fungsi view makan yang akan mengembalikan seluruh data pada data makam dan juga sekaligus dengan informasi blok dan tpu yang bersesuaian dengan tiap datanya. Baris 16-20 adalah fungsi create makam yang akan membuat entri data baru untuk sebuah makam baru sesuai dengan *request* yang diberikan. Baris 22-29 adalah fungsi edit makam yang akan mengupdate entri data spesifik sesuai dengan id yang diminta. Baris 31-41 Fungsi delete makam akan digunakan untuk menghapus makam beserta seluruh data yang mereferensikan data pada id tersebut. Baris 43-53 merupakan fungsi delete expire yang akan menghapus data makam yang memiliki umur lebih dari 3 tahun bersamaan dengan seluruh data yang mereferensikan data pada id tersebut.

### Implementasi Fitur Manajemen Blok Makam

Fitur Manajemen Blok Makam berisi *create, read, update, delete* data Blok Makam dan juga fungsi untuk menyesuaikan *request* data spesifik berdasarkan role login dan hak akses yang di lakukan *user*. Fitur akan melakukan query ke database MySQL untuk menjalankan fungsi. Berikut adalah source code untuk Fitur Manajemen Blok Makam.

Tabel 5.17 Implementasi Fitur Manajemen Blok Makam

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54 | function view\_blok(*Request* $*request*){  $id\_*user* = $*request*->input('id\_*user*');  $view = DB::tabel('blok\_makam')  ->join('tpu', 'blok\_makam.id\_tpu', '=', 'tpu.id\_tpu')  ->join('role\_tpu','role\_tpu.id\_tpu','=', 'tpu.id\_tpu')  ->select('blok\_makam.\*', 'tpu.\*')  ->where('role\_tpu.id\_*user*','=',$id\_*user*)  ->get();  return response()->json($view);  }  function view\_search\_blok(){  $view = DB::tabel('blok\_makam')  ->join('tpu', 'blok\_makam.id\_tpu', '=', 'tpu.id\_tpu')  ->select('blok\_makam.\*', 'tpu.\*')  ->get();  return response()->json($view);  }  function create\_blok(*Request* $*request*){  return Blok\_Makam::create($*request*->all());  }  public function edit\_blok(*Request* $*request*, $id)  {  $blok = Blok\_Makam::findOrFail($id);  $blok->update($*request*->all());  return $blok;  }  public function delete\_blok(*Request* $*request*, $id)  {  $penghuni\_makam = DB::tabel('penghuni\_makam')  ->join('makam', 'penghuni\_makam.id\_makam', '=', 'makam.id\_makam')  ->join('blok\_makam', 'makam.id\_blok', '=', 'blok\_makam.id\_blok')  ->where('blok\_makam.id\_blok','=',$id)  ->delete();  $makam = DB::tabel('makam')  ->join('blok\_makam', 'makam.id\_blok', '=', 'blok\_makam.id\_blok')  ->where('blok\_makam.id\_blok','=',$id)  ->delete();  $blok = Blok\_Makam::findOrFail($id);  $blok->delete();  return response()->json($blok);  } |

Fitur Manajemen Blok Makam berisi beberapa fungsi. Baris 1-11 adalah fungsi view blok yang akan mengembalikan data yang sesuai dengan *request* id *user* dan rolenya pada blok makam dan informasi tpu berserta kecamatan yang bersesuaian dengan setiap data. Pada baris 13-19 adalah fungsi search blok yang akan mengembalikan seluruh data blok sekaligus informasi tpu yang bersesuaian. Pada baris 21-25 adalah fungsi create blok yang akan membuat entri data baru untuk blok makam sesuai dengan *request* yang diberikan. Pada baris 27-32 adalah fungsi edit blok akan mengupdate blok sesuai dengan id yang diminta. Baris 34-54 adalah fungsi delete makam yang akan menghapus entri data yang sesuai dengan *request* id dan seluruh data yang mereferensikan data dengan id tersebut.

### Implementasi Fitur Manajemen Penghuni Makam

Fitur Manajemen Penghuni Makam berisi *create, read, update, delete* (CRUD) data Penghuni makam dan juga fungsi untuk menyesuaikan *request* data spesifik berdasarkan role login dan hak akses yang di lakukan *user*. Fitur akan melakukan query ke database MySQL untuk menjalankan fungsi. Berikut adalah source code untuk Fitur Manajemen Penghuni Makam.

Tabel 5.18 Implementasi Fitur Manajemen Penghuni Makam

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 | function view\_penghunimakam(*Request* $*request*){  $id\_*user*=$*request*->input('id\_*user*');  $view = DB::tabel('penghuni\_makam')  ->join('makam', 'penghuni\_makam.id\_makam', '=', 'makam.id\_makam')  ->join('blok\_makam', 'makam.id\_blok', '=', 'blok\_makam.id\_blok')  ->join('tpu', 'tpu.id\_tpu', '=', 'blok\_makam.id\_tpu')  ->join('role\_tpu','role\_tpu.id\_tpu','=', 'blok\_makam.id\_tpu')  ->select('penghuni\_makam.\*', 'makam.\*','tpu.\*')  ->where('role\_tpu.id\_*user*','=',$id\_*user*)  ->get();  return response()->json($view);  }  function view\_search\_penghunimakam(){  $view = DB::tabel('penghuni\_makam')  ->join('makam', 'penghuni\_makam.id\_makam', '=', 'makam.id\_makam')  ->join('blok\_makam', 'makam.id\_blok', '=', 'blok\_makam.id\_blok')  ->select('penghuni\_makam.\*', 'makam.\*')  ->get();  return response()->json($view);  }  function create\_penghunimakam(*Request* $*request*){  return $penghuni\_makam = Penghuni\_makam::create($*request*->all());  }  public function update\_penghunimakam(*Request* $*request*, $id)  {  $penghuni\_makam = Penghuni\_makam::findOrFail($id);  $penghuni\_makam->update($*request*->all());  return $penghuni\_makam;  }  public function delete\_penghunimakam(*Request* $*request*, $id)  {  $penghuni\_makam = Penghuni\_makam::findOrFail($id);  $penghuni\_makam->delete();  return 204;  } |

Fitur Manajemen Penghuni Makam memiliki beberapa fungsi. Pada baris 1-15 terdapat fungsi view penghuni makam yang akan mengembalikan data penghuni makam sesuai dengan role *user* dan hak aksesnya. Baris 17-26 terdapat fungsi view search penghuni makam yang akan mengembalikan seluruh data penghuni makam tanpa kecuali. Baris 28-33 terdapat fungsi create penghuni makam yang akan membuat entri baru untuk tabel penghuni makam sesuai dengan *request*. Baris 35-40 terdapat fungsi update penghuni makam yang akan mengupdate data sesuai dengan id yang di *request*. Baris 42-48 terdapat fungsi delete penghuni makam yang akan menghapus data penghuni makam yang bersesuaian dengan *request*

### Implementasi Fitur *Authentication*

Fitur *Authentication* berisi fungsi yang melayani login dan credential token. Fitur akan melakukan query ke database MySQL untuk menjalankan fungsi. Berikut adalah source code untuk Fitur *Authentication*.

Tabel 5.19 Implementasi Fitur Authentication

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | public function signin(*Request* $*request*)  {  $credentials = [  '*user*name' => $*request*->input('*user*name'),  'password' => $*request*->input('password')  ];    $rules = [  '*user*name' => 'required',  'password' => 'required',  ];  $*user* = *User*::where('*user*name','=',$*request*->input('*user*name'),'AND','password','=',Hash::make($*request*->input('password')))->select('id\_*user*','role','*user*name')->get();  $validator = Validator::make($credentials, $rules);  if($validator->fails()) {  return response()->json(['success'=> false, 'error'=> $validator->*message*s()]);  }    $token = null;  try {  if (! $token = JWTAuth::attempt($credentials)) {  return response()->json(['success' => false, 'error' => 'Akun tidak ditemukan!',  '*user*name'=> $*request*->input('*user*name'),  'password'=> Hash::make($*request*->input('password'))],401);  }  } catch (JWTAuthException $e) {  return response()->json(['msg'=>'Login Gagal !','success' => false, 'error' => 'Failed to login, please try again.'], 500);  }  if($*user*[0]->role == 1){  $role\_tpu = Role\_tpu::where('id\_*user*','=',$*user*[0]->id\_*user*)->select('id\_*user*')->get();  $resp = [  [  'msg'=>'Login Berhasil !',  'success' => true,  'token'=> $token,  '*user*name' => $*user*[0]->*user*name,  'role' => $*user*[0]->role,  'id\_*user*' => $role\_tpu[0]->id\_*user*,  ]  ];  } else {  $resp = [  [  'msg'=>'Login Berhasil !',  'success' => true,  'token'=> $token,  '*user*name' => $*user*[0]->*user*name,  'role' => $*user*[0]->role,  'id\_*user*' => $*user*[0]->id\_*user*,  ]  ];  }    return response()->json($resp);  } |

Dalam implementasi, digunakan *library* tambahan yaitu JWTAuth untuk melakukan generate token dan akan diminta pada setiap *request* pada api. Dalam *service* terdapat fungsi signin yang akan melayani login dengan pencocokan data tabel *user* berdasarkan *user*name dan password. Fungsi signin akan mengembalikan token, *user*name, id *user*, berserta rolenya. Pada baris 3-11 akan menyimpan data *request*. Dan baris 14-16 adalah query untuk mengambil data *user* dari tabel *user* pada database. Pada baris 17-21 adalah validator fungsi dan pada baris 23-36 adalah validator credential dengan JWT Auth. Pada baris 38-66 adalah pemilihan aksi pengembalian data berdasarkan role

### Implementasi Mail *Service*

*Service* mail berisi fungsi pengiriman email. Berikut adalah source code *service* mail.

Tabel 5.20 Implementasi Mail Service

|  |  |
| --- | --- |
| 1  2  3 | function send(){  Mail::send(new EmailNotification());  } |

Fungsi *send* akan menggunakan *library* bawaan mail dan membuat sebuah objek baru dari kelas email notification yang mengextend kelas mailable, berikut adalah source code dari class email notification.

Tabel 5.21 Implementasi Mail Notification

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | <?php  namespace App\Mail;  use Illuminate\Bus\Queueable;  use Illuminate\Mail\Mailable;  use Illuminate\Queue\SerializesModels;  use Illuminate\Contracts\Queue\ShouldQueue;  use Illuminate\Http\*Request*;  class EmailNotification extends Mailable  {  use Queueable, SerializesModels;  /\*\*  \* Create a new *message* instance.  \*  \* @return void  \*/  public function \_\_construct()  {  //  }  /\*\*  \* Build the *message*.  \*  \* @return $this  \*/  public function build(*Request* $*request*)  {  return $this->view('mail',['msg' => $*request*->input('*message*')])  ->to($*request*->input('to'))  ->subject('Notification Email')  ->from('hehe@gmail.com','Project PKL Emakam 2018');  }  } |

Fungsi builld akan mengirimkan email dengan subjek tertentu pada baris 36 dan pesan dapat diberikan pada parameter msg pada baris 33 rserta tujuan yang bisa disesuaikan dari *request* oada baris 37. Email akan dikirimkan dengan konten yang telah didefinisikan dalam view mail baris 33 yang menjadi template pengiriman email

### Persiapan Deployment Layanan pada Server

Pastikan sistem operasi server telah terupdate hingga versi terakhir

Update server dengan menjalankan perintah :

$sudo apt update

$sudo apt upgrade

|  |
| --- |
| Screenshot proses update sistem operasi:    Gambar 5.2 Gambar Update Sistem Operasi Ubuntu 18.04 Server    Gambar 5.3 Gambar Upgrade Sistem Operasi Ubuntu 18.04 Server |

Setelah Dipastikan server sudah up to date , lakukan instalasi apache2

Untuk melakukan Instalasi apache2 dengan menjalankan perintah

$sudo apt install apache2

|  |
| --- |
| Screenshot proses instalasi apache2    Gambar 5.4 Instalasi apache2 |

Setelah instalasi berhasil dilakukan pastikan apache2 telah berjalan dengan baik

Untuk memeriksa status apache 2 lakukan perintah berikut

$sudo *system*ctl status apache2

|  |
| --- |
| Screenshot hasil pemeriksaan status :    Gambar 5.5 Cek Status Apache2 |

Setelah melakukan instalasi apache 2 , kita akan melakukan instalasi MySQL server

Lakukan Instalasi MySQL-server dengan menjalankan perintah :

$sudo apt install MySQL-server

|  |
| --- |
| Gambar 5.6 Instalasi MySQL-Server |

Setelah instalasi selesai periksa status MySQL-server apakah sudah berjalan dengan baik

Untuk memeriksa status MySQL-server jalankan perintah

$sudo *system*ctl status MySQL

|  |
| --- |
| Screenshot hasil pemeriksaan status MySQL :    Gambar 5.7 Cek Status MySQL Server |

Pertama kita perlu mendambahkan PPA php7.2

Jalankan perintah berikut:

$sudo apt-get install *software*-properties-common  
$sudo add-apt-repository ppa:ondrej/php  
$sudo apt update

Setelah itu lakukan Instalasi php7.2 FPM dan modul yang terkait dengan perintah

$sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-mbstring php7.2-xmlrpc php7.2-soap php7.2-gd php7.2-xml php7.2-intl php7.2-MySQL php7.2-cli php7.2-curl

|  |
| --- |
| Screenshot proses instalasi php7.2 :    Gambar 5.8 Instalasi PHP7.2 |

Setelah menginstall php7.2, jalankan perintah berikut untuk membuka php-fpm default file

$sudo nano /etc/php/7.2/apache2/php.ini

Ubah value untuk beberapa parameter berikut:

file\_uploads = On  
allow\_url\_fopen = On  
upload\_max\_filesize = 100M

Kemudian lakukan restart apache2 dengan perintah  
$sudo *system*ctl restart apache2

Lakukan seluruh langkah diatas pada kedua server yang akan digunakan kecuali untuk msysql-server cukup dilakukan pada salah satu server yang ingin digunakan

### Deployment Layanan pada Server

Pada salah satu server yang telah disiapkan akan dideploy *Backend* server atau *Webservice* untuk aplikasi yang akan digunakan pada bagian ini akan dijelaskan langkah langkah deployment *backend* server. Hal pertama yang harus dilakukan adalah melakukan cloning aplikasi yang telah dibuat repository aplikasi tersedia pada tautan

<https://github.com/frmnjn/projectpkl_emakam>

Lakukan cloning repository dengan perintah berikut:

$git clone https://github.com/frmnjn/projectpkl\_emakam

Setelah aplikasi telah di clone, import database aplikasi yang telah disiapkan pada repository

Buat database dengan perintah berikut :

$sudo MySQL –u root –p

>create database emakam

|  |
| --- |
| Hasil create database :    Gambar 5.9 Membuat Database E-makam |

Berikutnya lakukan import sql dari repository pada database yang baru dibuat

Lakukan dengan perintah

$sudo MySQL –u root –p emakam < emakamv6.1

|  |
| --- |
| Screenshot Hasil import:    Gambar 5.10 Import Database E-makam |

Salin direktori *webservice* repository kedalam direktori /var/www/html dengan masuk ke direktori projectpkl\_emakam, copy folder dengan perintah :

$sudo cp -R api\_emakam /var/www/html

|  |
| --- |
| Screenshot hasil copy direktori    Gambar 5.11 Copy Direktori API Emakam |

Kemudian Ubah hak akses direktori setra group kepemilikan direktori

Lakukan perintah berikut

$sudo chgrp -R www-data /var/www/html/api\_emakam  
$sudo chmod -R 775 /var/www/html/api\_emakam/storage  
$sudo chmod -R 775 /var/www/html/api\_emakam/bootstrap/cache

Setelah itu periksa hasilnya dengan perintah berikut :

$ls –l /var/www/html/api\_emakam/

|  |
| --- |
| Screenshot hasil pemeriksaan :    Gambar 5.12 Ubah Hak akses dan grup Direktori |

Setelah itu buat konfigurasi file untuk laravel pada apache2, edit file laravel.conf dengan text editor

$sudo nano /etc/apache2/sites-available/laravel.conf

Isikan file dengan kode berikut:

<VirtualHost \*:80>

ServerName localhost

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html/api\_emakam/public

<Directory /var/www/html/api\_emakam>

AllowOverride All

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Berikutnya jalankan perintah

$sudo ad2issite 000-default.conf

|  |
| --- |
| Screenshot Hasil perintah:    Gambar 5.13 a2dissite konfigurasi |

Lakukan link direktori storage, tetapi sebelumnya hapus terlebih dahulu direktori link yang sudah ada dari hasil repository:

Lakukan perintah berikut

$cd /var/www/html/api\_emakam/public

$rm –r storage/

$sudo php artisan storage:Link

|  |
| --- |
| Hasil hapus direktori :    Gambar 5.14 Menghapus DIrektori Storage  Hasil link storage :    Gambar 5.15 Membuat Link Storage |

## Lingkungan Pengujian

Pengujian dilakukan terhadap sistem terdistribusi berbasis *microservice* yang telah diimplementasi sebelumnya. Terdapat tiga jenis pengujian yang akan dilakukan yakni pengujian fungsionalitas dan skalabilitas. Pengujian fungsionalitas berfungsi untuk mengetahui apakah fitur yang telah dibuat sebelumnya berjalan atau tidak. Pengujian skalabilitas berfungsi untuk melihat seberapa handal sistem dalam menangani *request* dalam jumlah banyak dalam satu waktu.

### Parameter Pengujian

Berdasarkan penjelasan pada sub bab 5.5 akan ada tiga jenis pengujian yaitu fungsionalitas, kinerja dan skalabilitas berikut adalah parameter yang akan kami gunakan untuk setiap pengujian

1. Fungsionalitas : keberhasilan fungsional dalam penambahan/perubahan entri data
2. Kinerja : latency
3. Skalabilitas: presentase error sistem

### Skenario Pengujian

**Pengujian Fungsionalitas :**

Sistem akan dijalankan dan di *webservice* akan diakses melalui *frontend* *system*. Beberapa fungsionalitas dari setiap fitur akan diambil sebagai sampel dan dilakukan *request* melalui *frontend*. Hasil dari pengujian akan di screenshot sebagai bukti keberhasilan.

**Pengujian Kinerja:**

*Backend* *service* akan dijalankan dan akan diuji menggunakan *tools* Apache JMeter untuk me*request* fitur. Hasil pengujian kinerja akan diambil latency nya dan di screenshot sebagai bukti dan dianalisis sebagai kesimpulan.

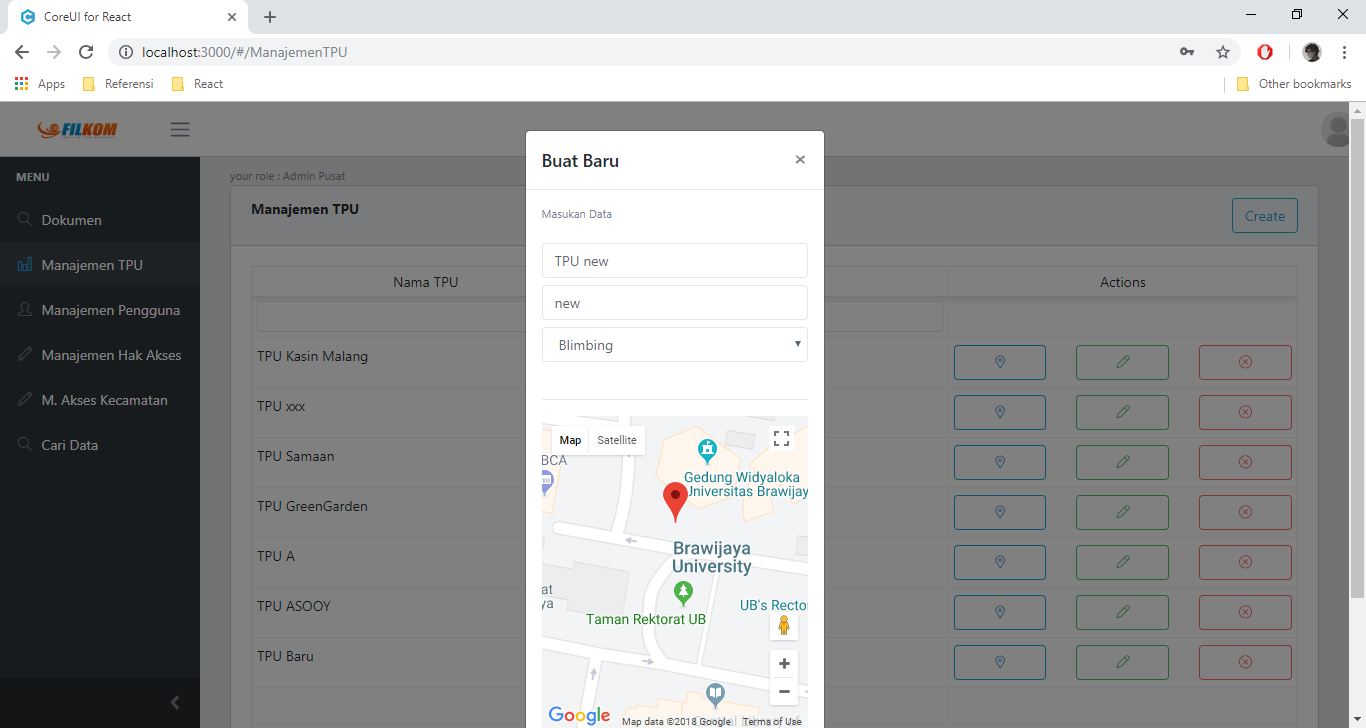
**Pengujian Skalabilitas**

*Backend* *service* akan dijalankan dan diuji menggunakan *tools* Apache JMeter. Pada pengujian skalabilitas akan dilakukan enam skenario dengan jumlah *request* yang berbeda yaitu 100, 250, 500, 750, 1000, dan 1500 *request* dalam satu waktu bersamaan. Tujuan dilakukan pengujian ini adalah untuk melihat kemampuan skalabilitas sistem yang telah dibuat sebelumnya. Dari hasilnya akan di screenshot sebagai bukti dan diambil presentase error *request*nya dan akan dianalisis untuk ditarik kesimpulan.

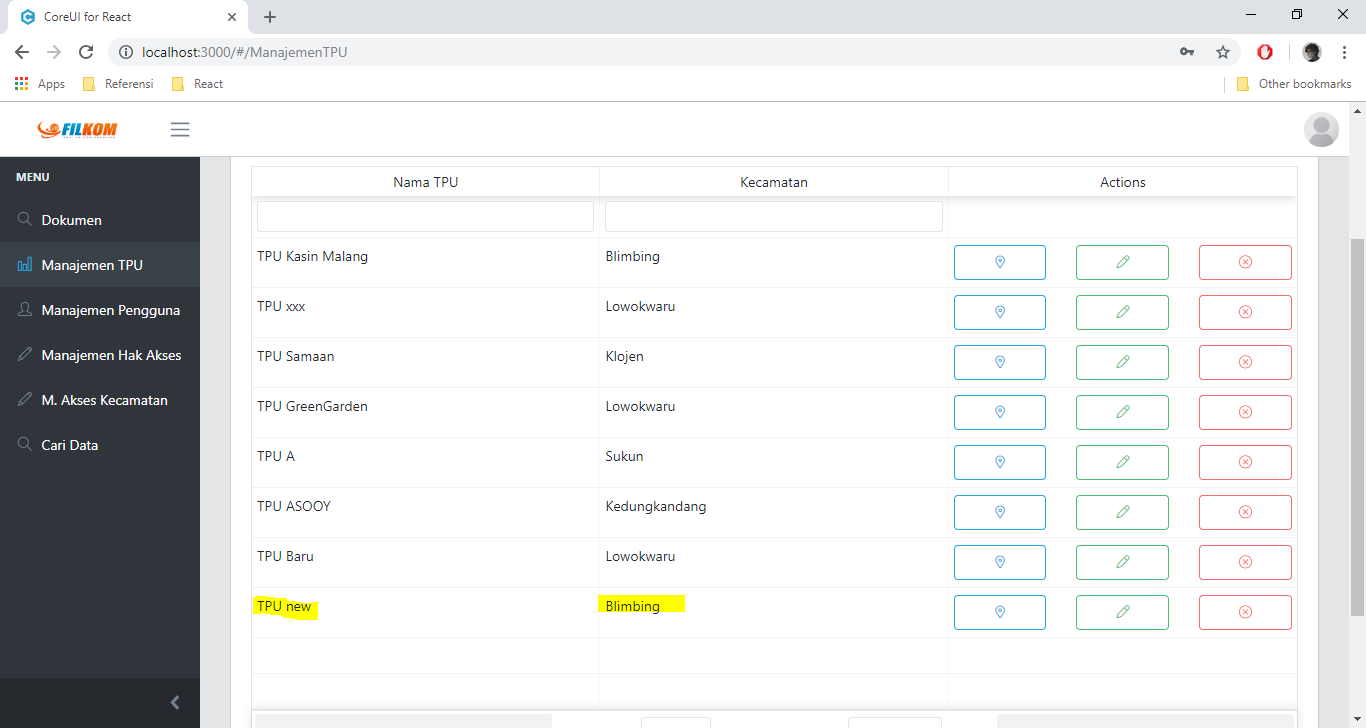
### Pengujian Fungsionalitas

### Fitur Manajemen TPU

Fungsi yang akan diuji pada Fitur Manajemen TPU adalah create TPU dengan melakukan input data baru dari aplikasi. Berikut adalah hasil pengujiannya :



Gambar 5.16 aplikasi pembuatan TPU

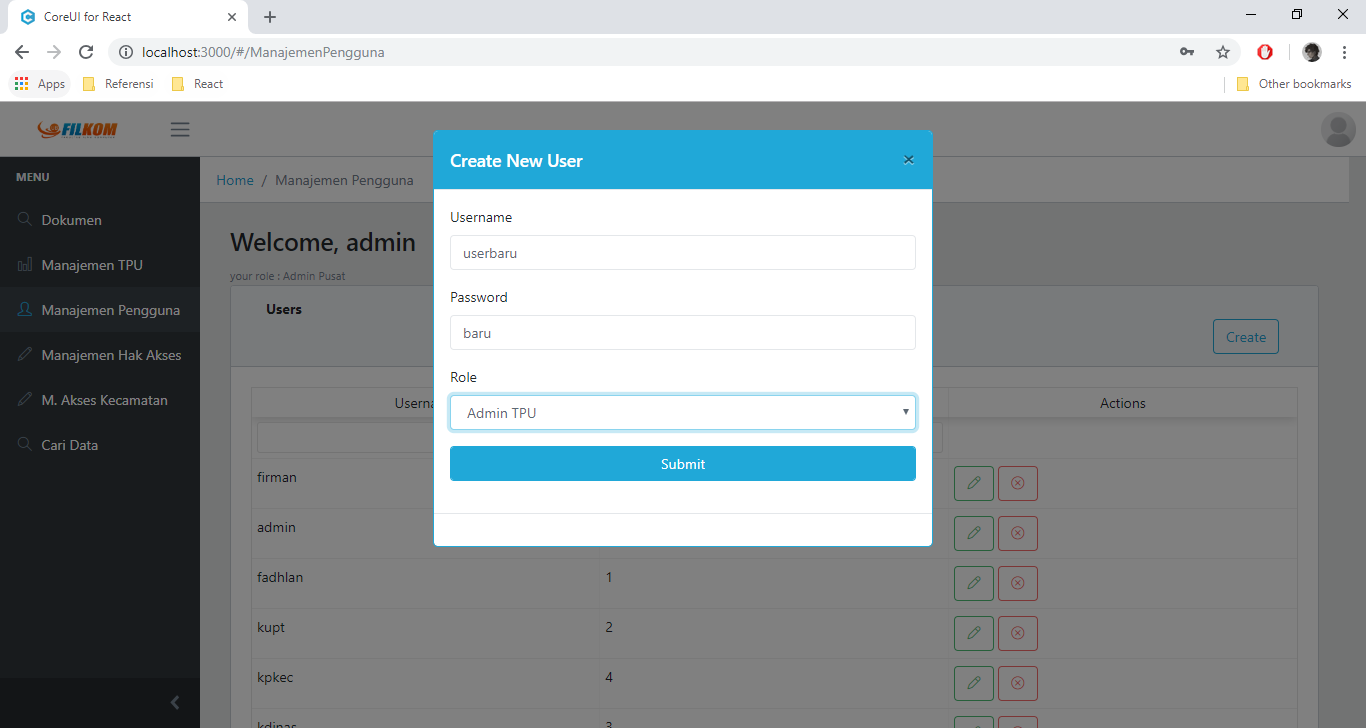


Gambar 5.17 aplikasi setelah pembuatan TPU

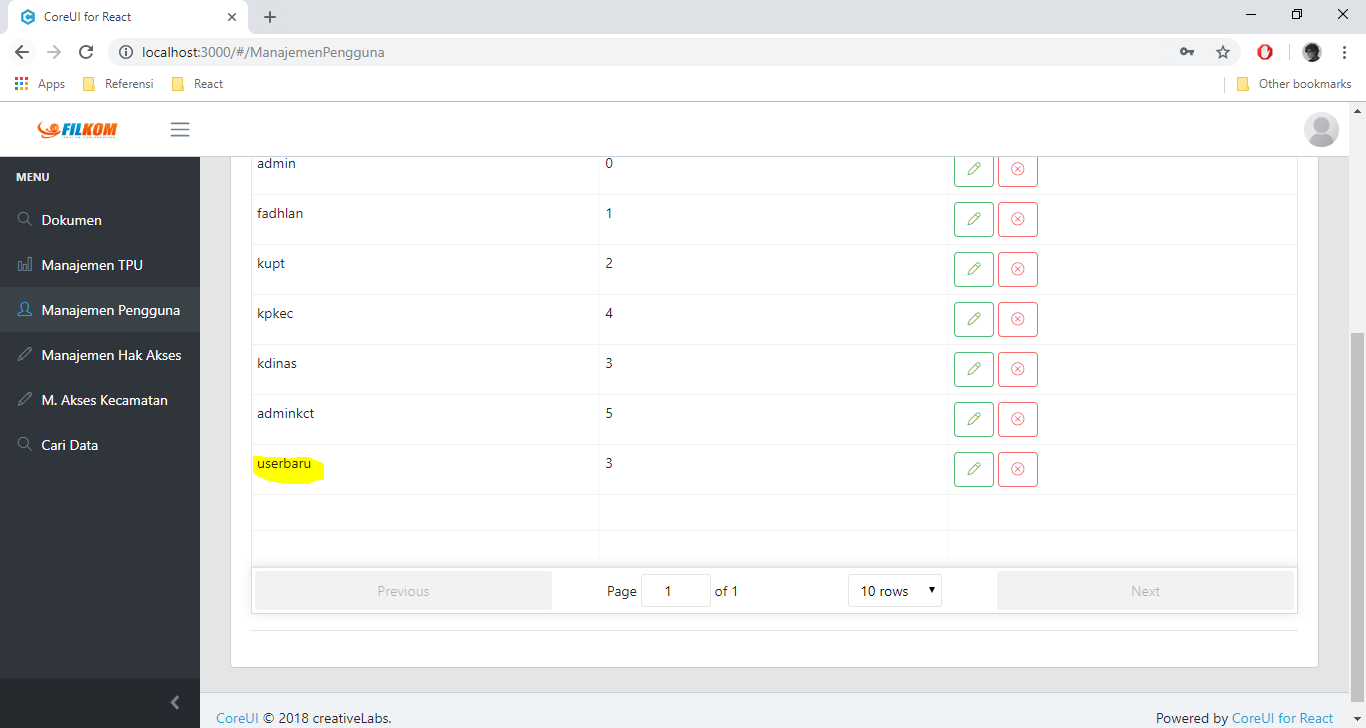
Dapat kita lihat bahwa terdapat data TPU baru sesuai dengan input pada aplikasi pembuatan TPU.

### Fitur *User*

Fungsional Fitur *user* yang akan kami uji adalah fungsional create *user* dengan mengisikan data baru melalui aplikasi. Berikut adalah hasil pengujiannya :



Gambar 5.18 aplikasi pembuatan User

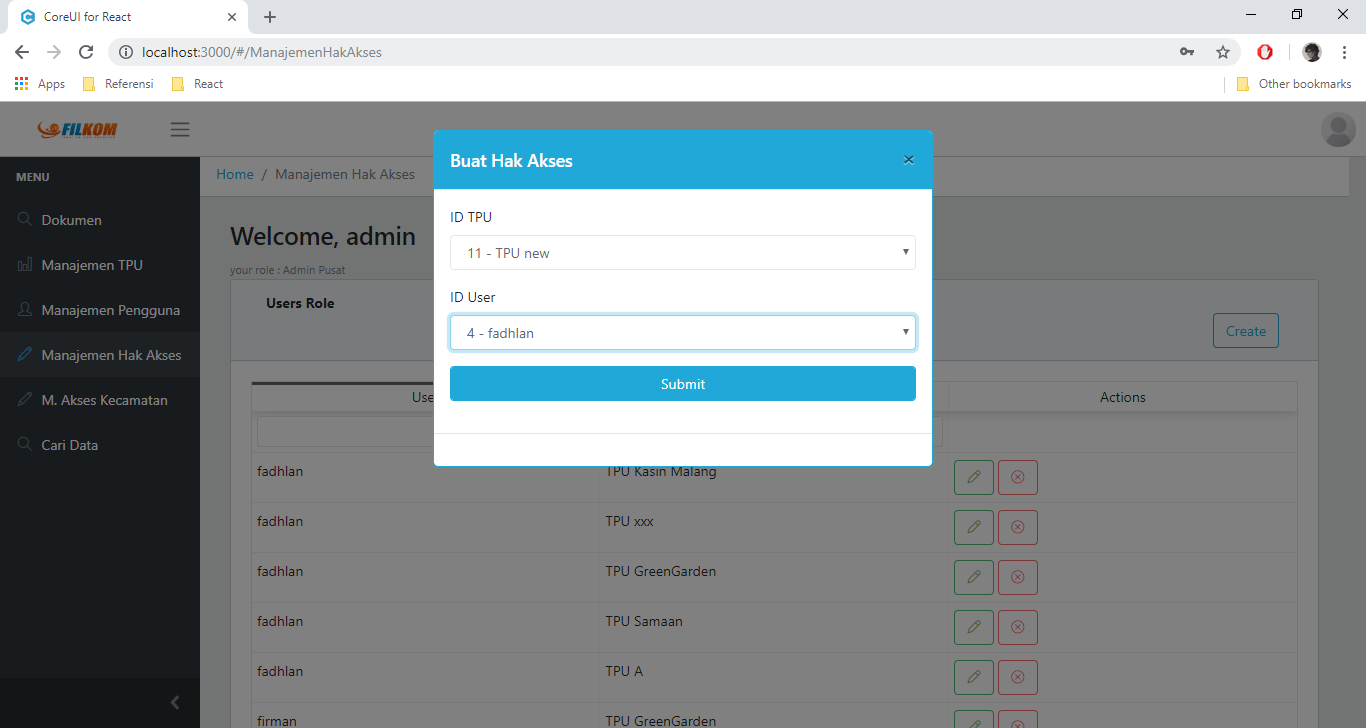


Gambar 5.19 aplikasi setelah pembuatan User

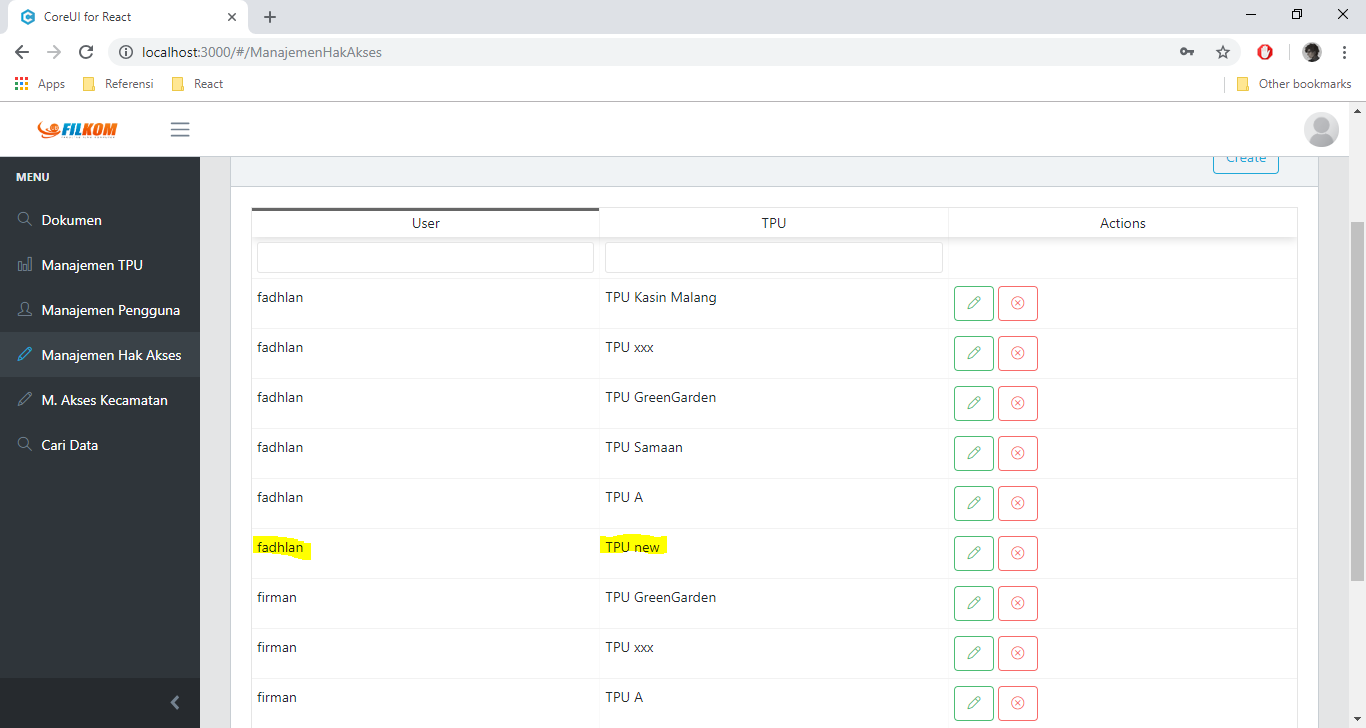
Dapat kita bandingkan bahwa setelah memasukan input data pembuatan *user* didapatkan data *user* yang baru

### Fitur Manajemen *User* Access

Dalam pengujian kami akan melakukan pengujian pada fungsi create hak akses. Berikut adalah hasil pengujiannya :



Gambar 5.20 aplikasi create user access

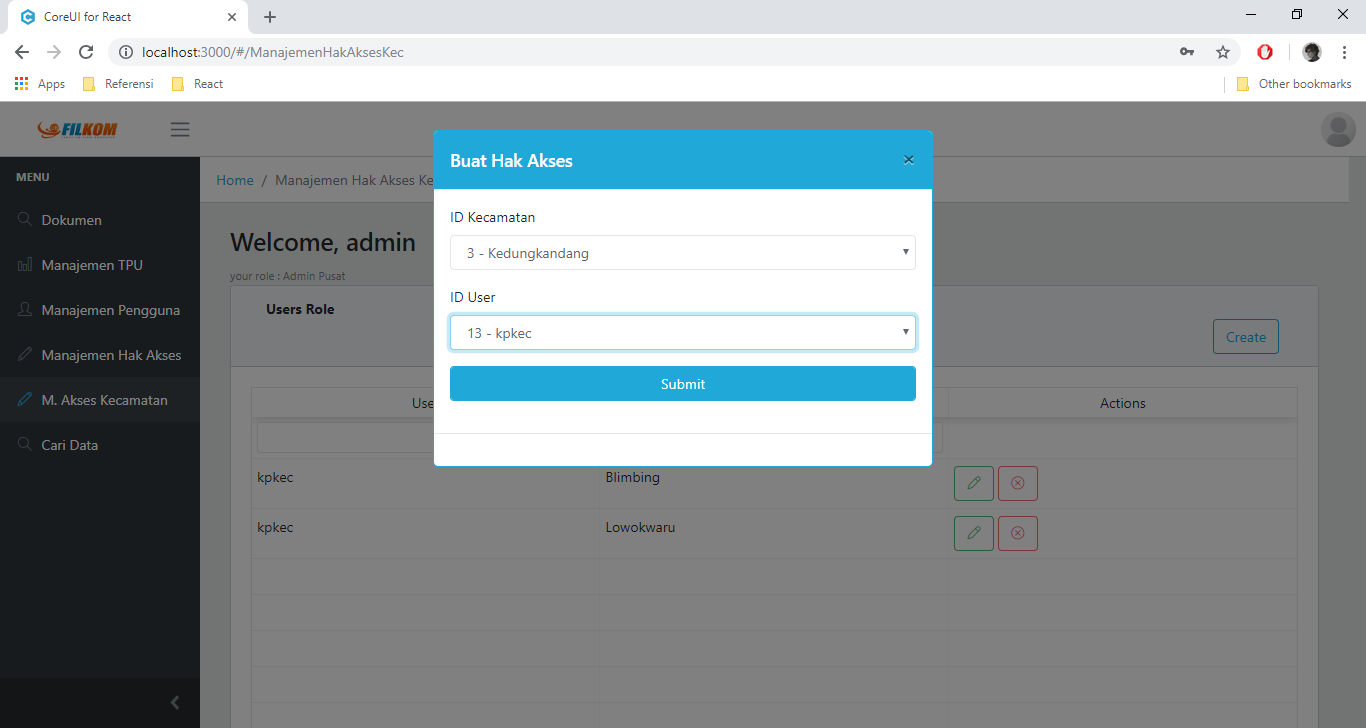


Gambar 5.21 aplikasi setelah create user access

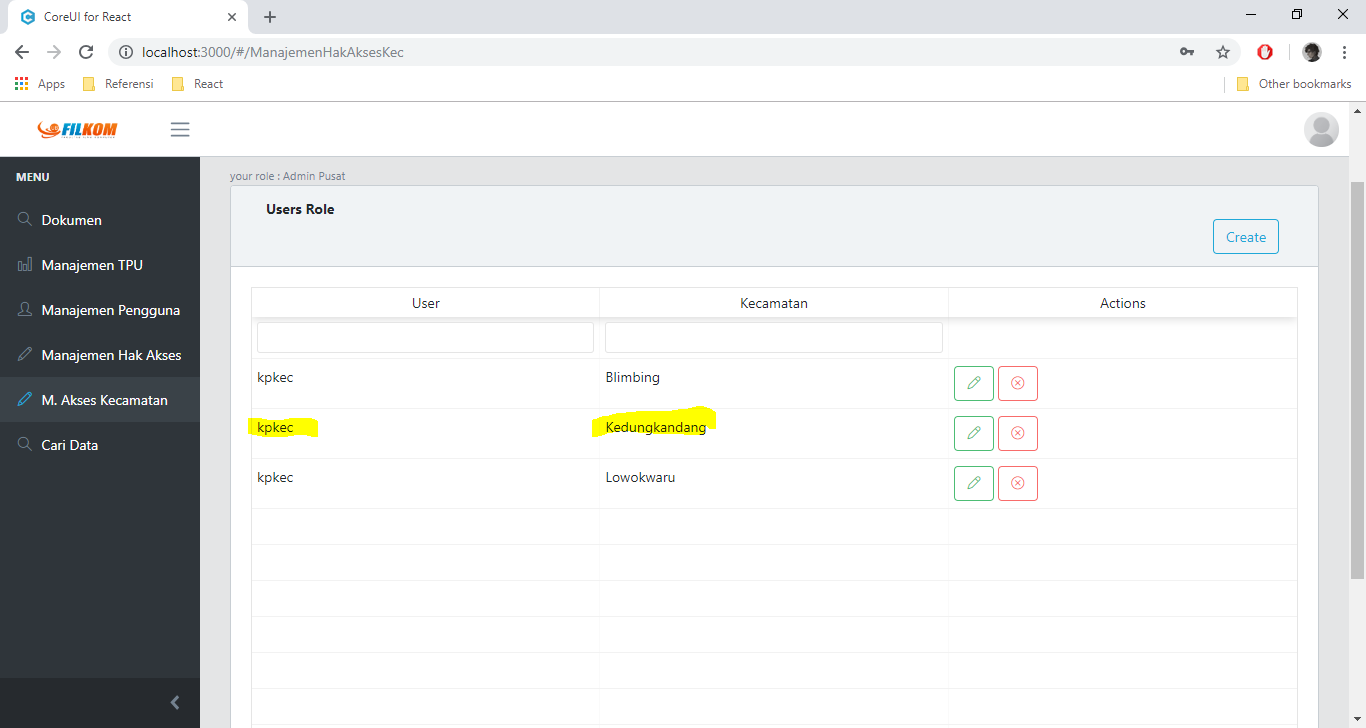
Dapat kita lihat bahwa tedapat entri data baru sesuai dengan input yang dilakukan padi apikasi create hak akses.

### Fitur Kecamatan Access

Dalam fitur ini fungsional yang akan kami uji adalah fungsional create untuk membuat hak akses baru:



Gambar 5.22 aplikasi create kecamatan access

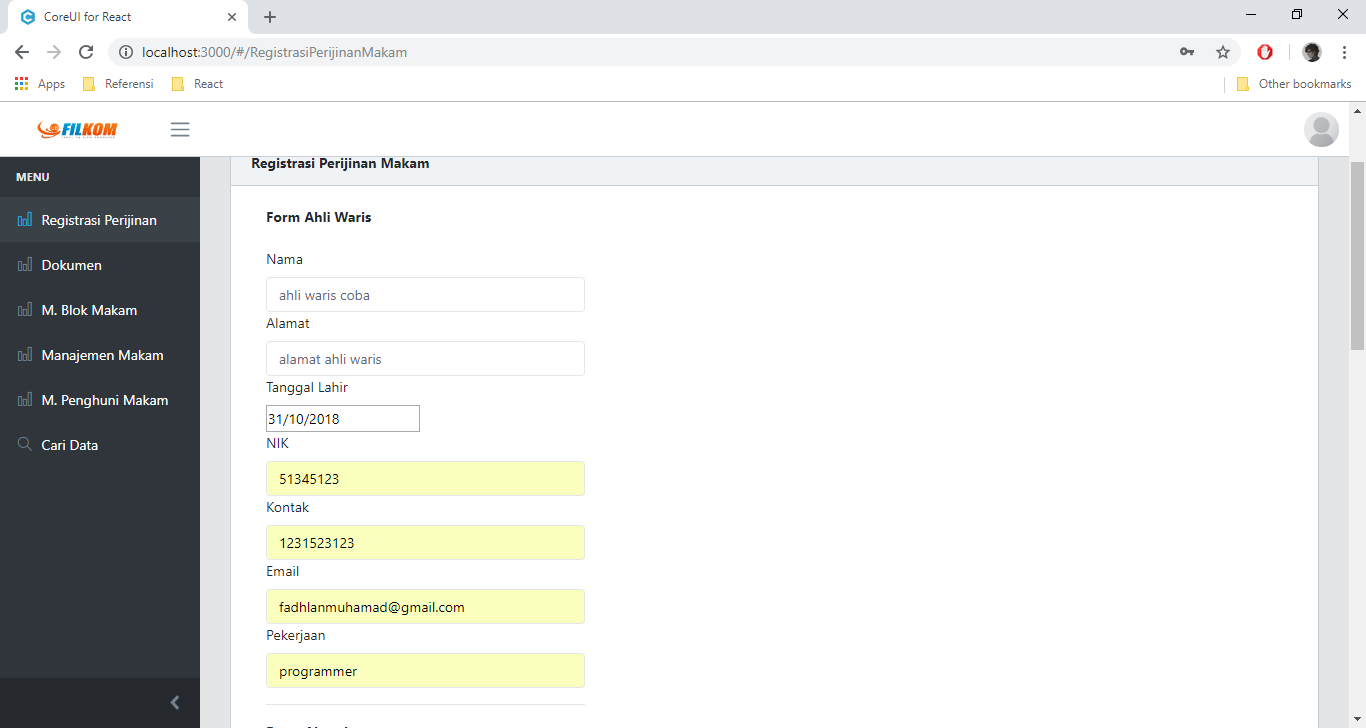


Gambar 5.23 aplikasi setelah create kecamatan access

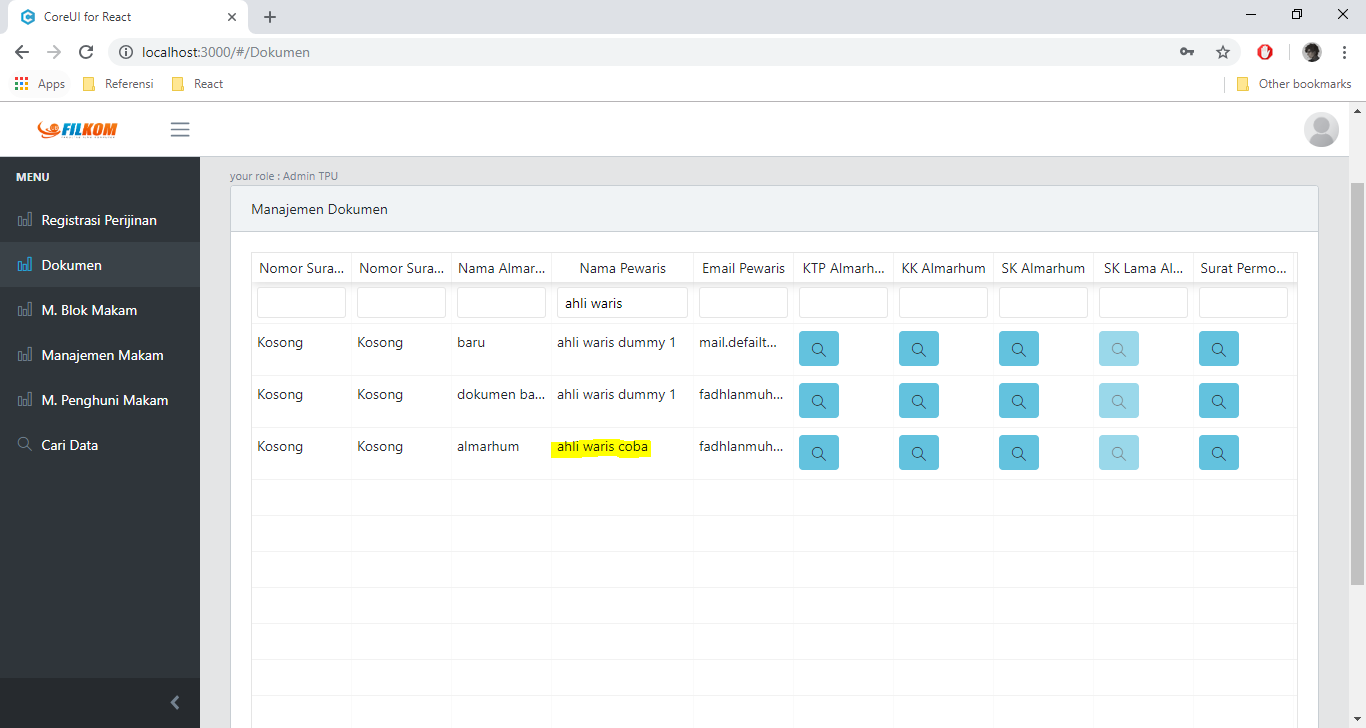
Dapat kita lihat bahwa muncul sebuah entri baru yang sesuai dengan input aplikasi hak akses kecamatan

### Fitur Registrasi Perijinan

Fungsi yang akan diuji pada fitu registrasi perizinan adalah fungsi pembuatan registrasi perizinan. Berikut adalah hasil pengujiannya :



Gambar 5.24 aplikasi form registrasi

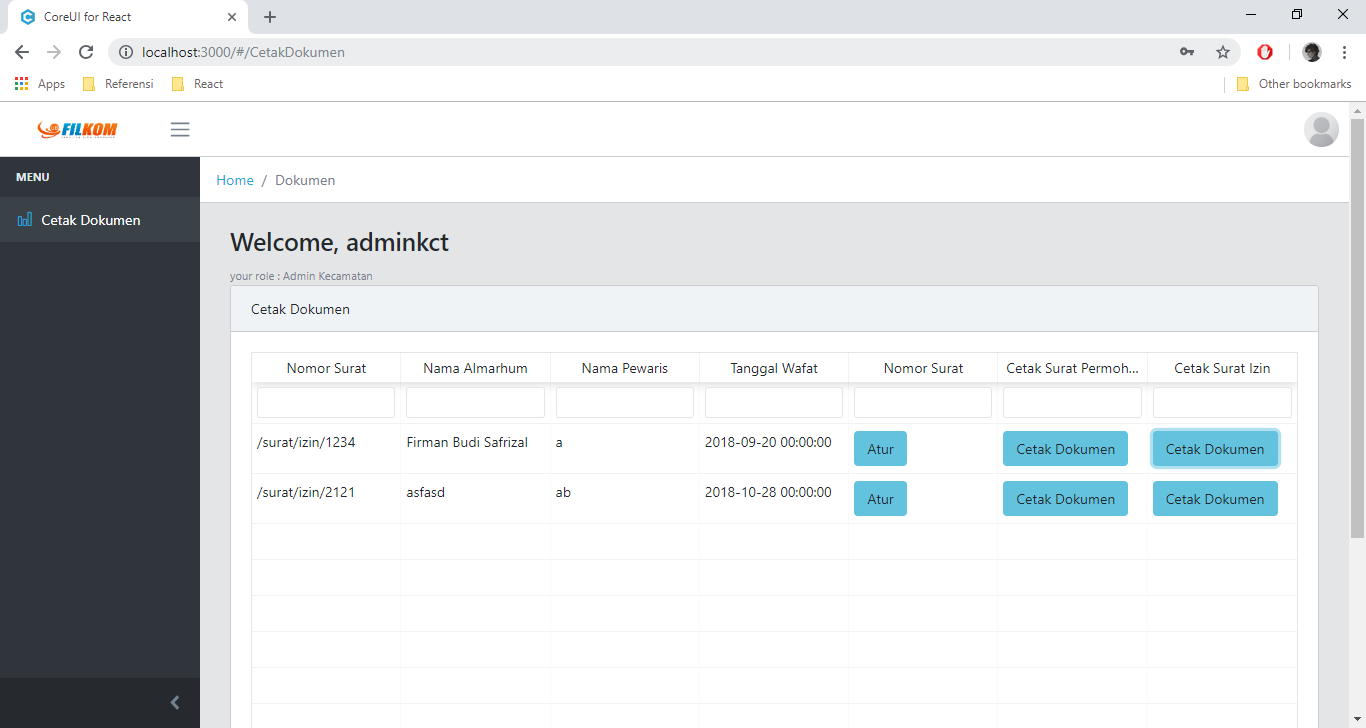


Gambar 5.25 aplikasi trace dokumen

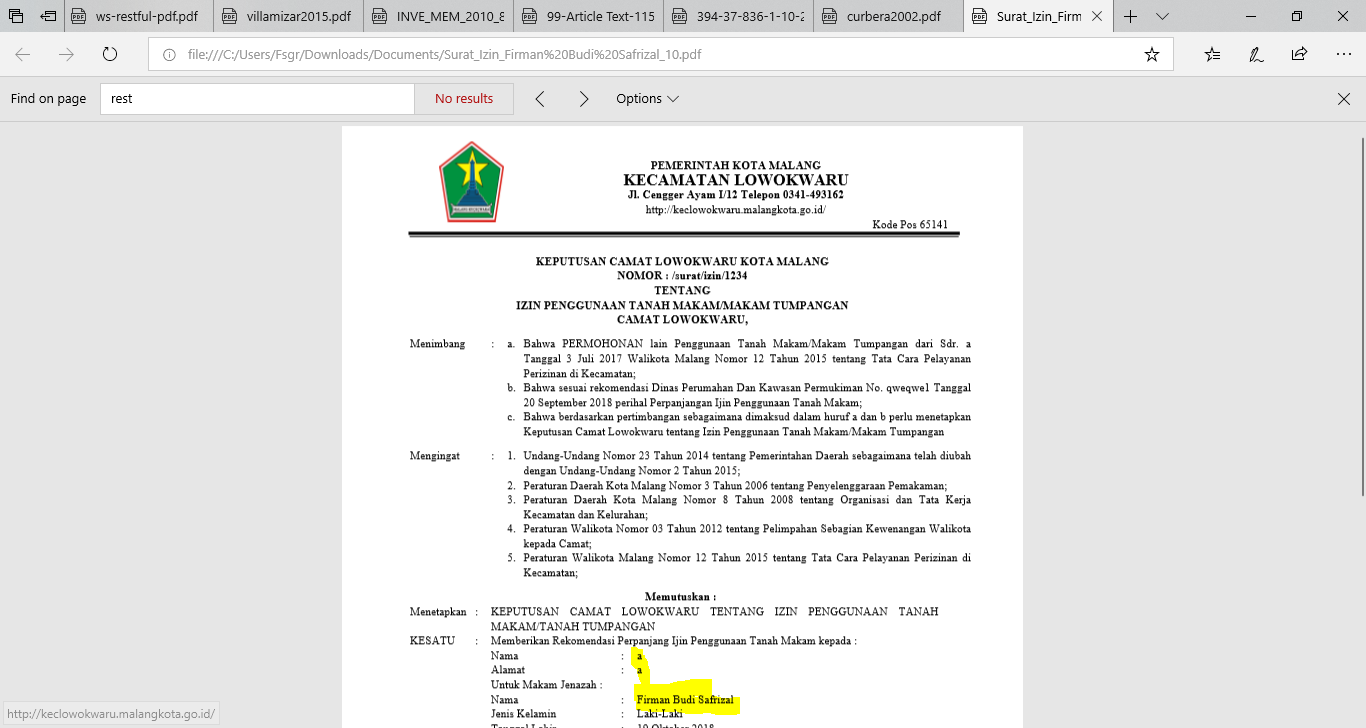
Dapat dilihat bahwa document telah masuk dan tersimpan dengan baik.

### Fitur Dokumen

Pengujian dilakukan dengan menguji percetakan dokumen yang telah terproses. Dengan menekan tombol cetak dokumen pada suratizin atau surat permohonan.



Gambar 5.26 aplikasi cetak dokumen

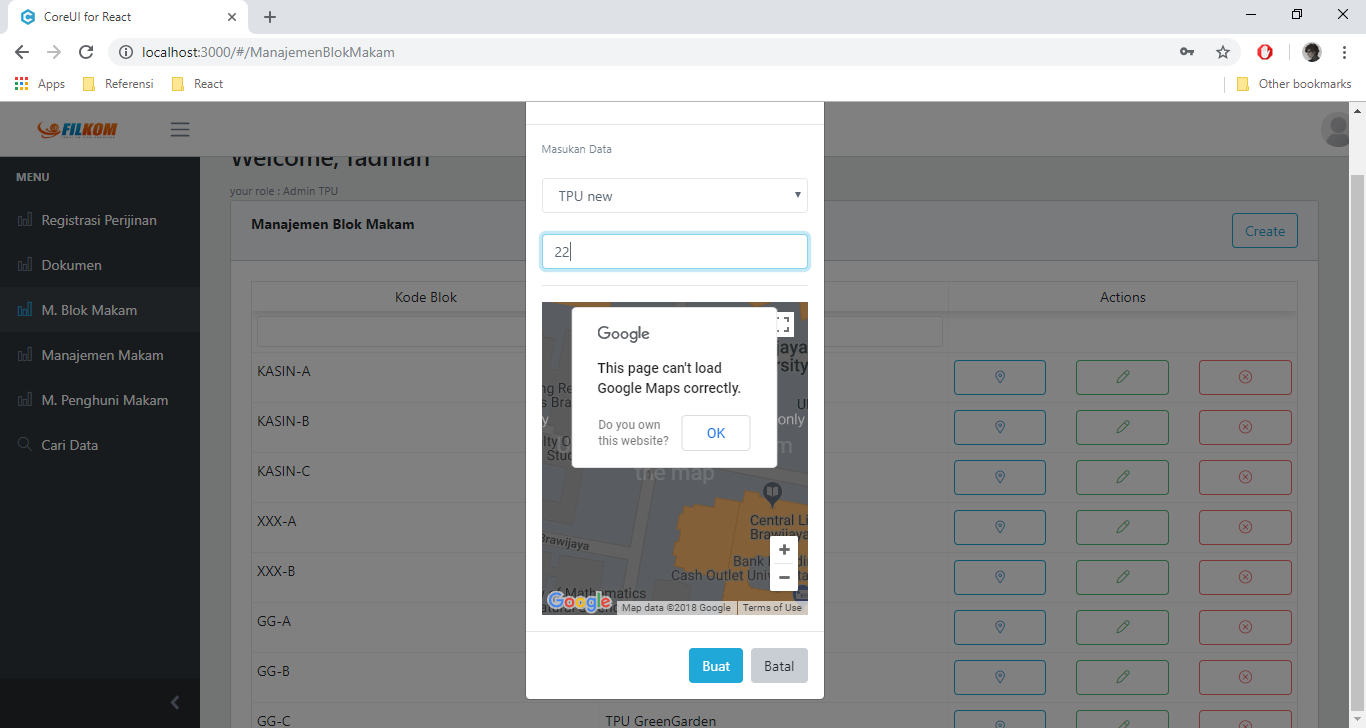


Gambar 5.27 hasil dokumen yang di generate cetak dokumen

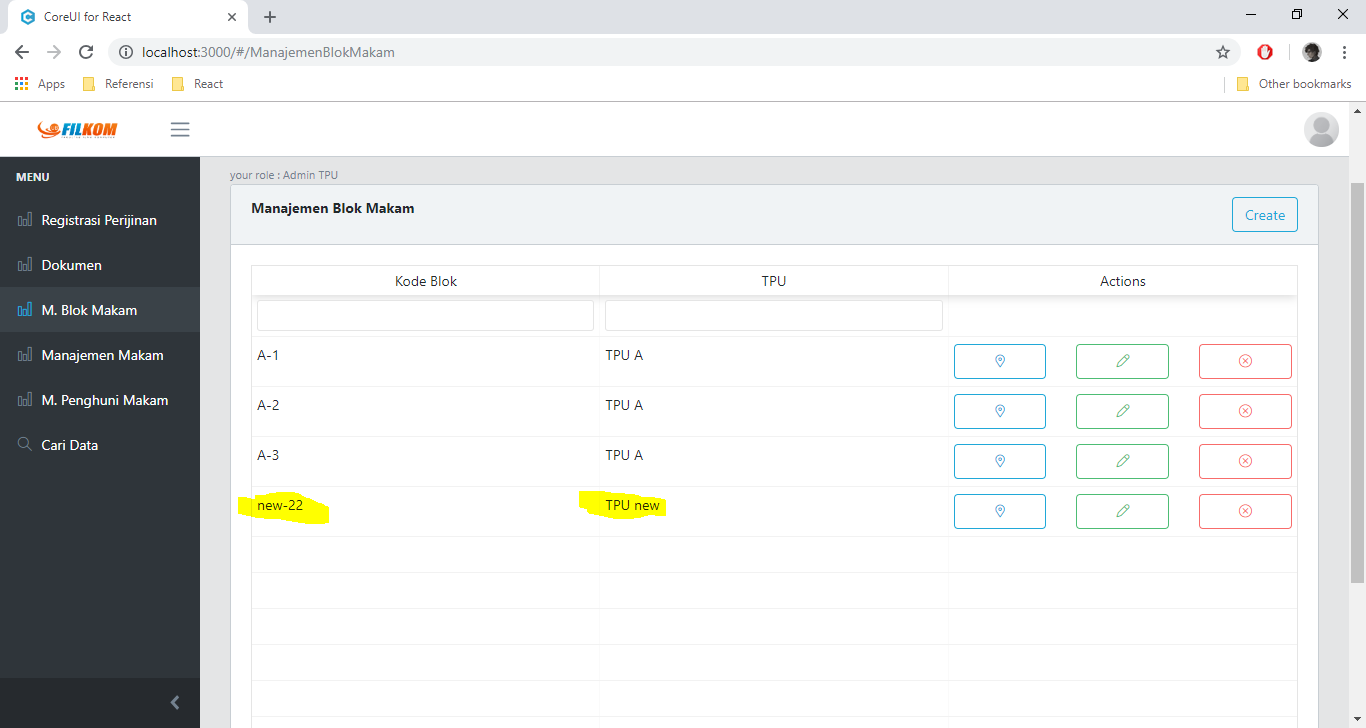
Dapat kita lihat bahwa hasil generate cetak dokumen cocok dengan nama pewaris dan nama almarhum pada aplikasi cetak dokumen

### Fitur Blok Makam

Pada fitur makam kami akan mencoba membuat sebuah blok makam baru untuk menguji fungsionalitas create blok makam. Berikut adalah hasil pengujiannya :



Gambar 5.28 aplikasi create blokmakam

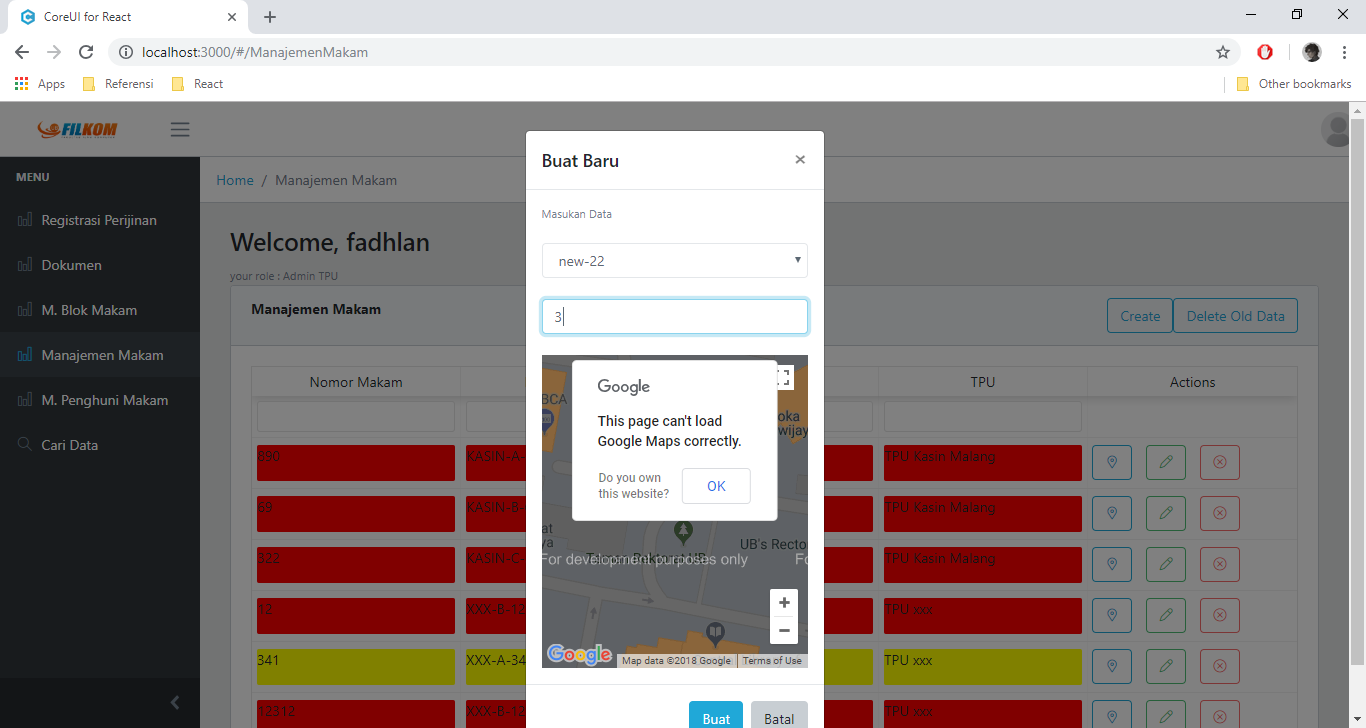


Gambar 5.29 aplikasi setelah create blokmakam

Dapat kita lihat terdapat entri baru yang sesuai dengan input yang dilakukan

### Fitur Makam

Pada fitur makam kami akan mencoba membuat sebuah makam baru untuk menguji fungsionalitas create makam. Berikut adalah hasil pengujiannya :



Gambar 5.30 aplikasi create makam

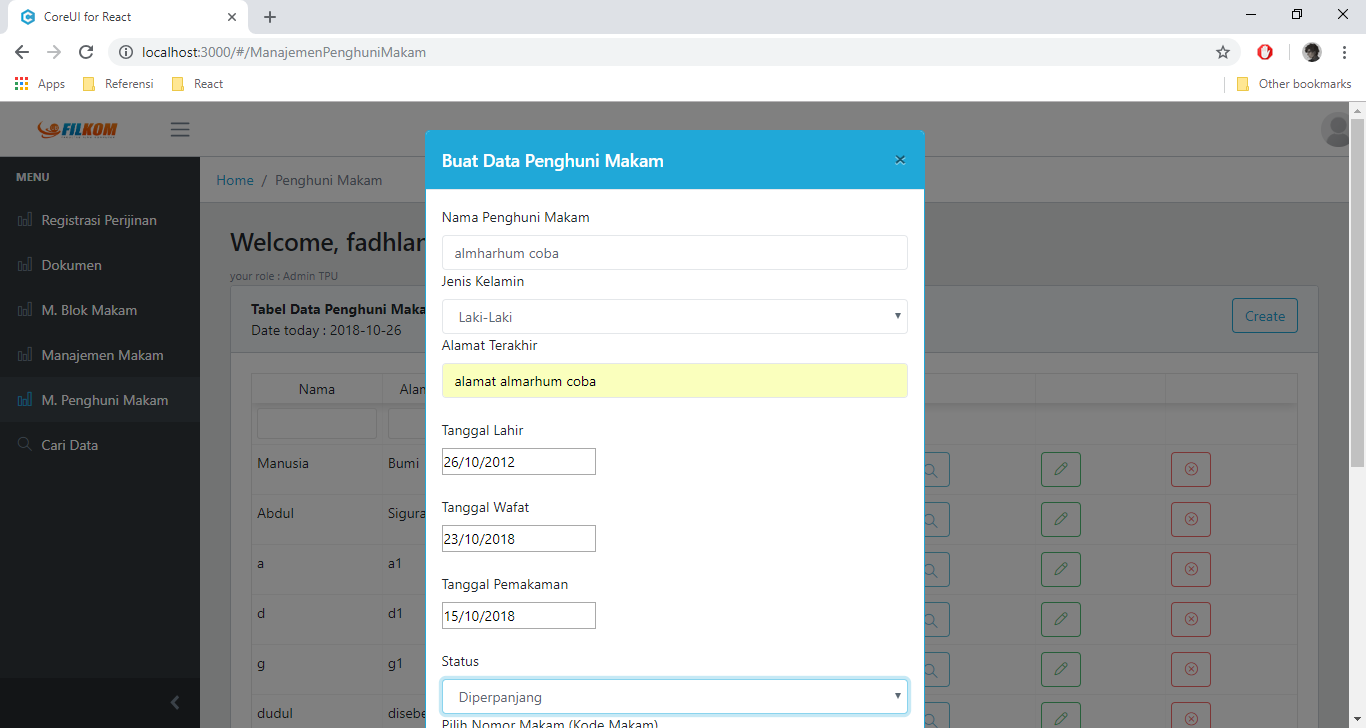


Gambar 5.31 aplikasi makam setelah create makam

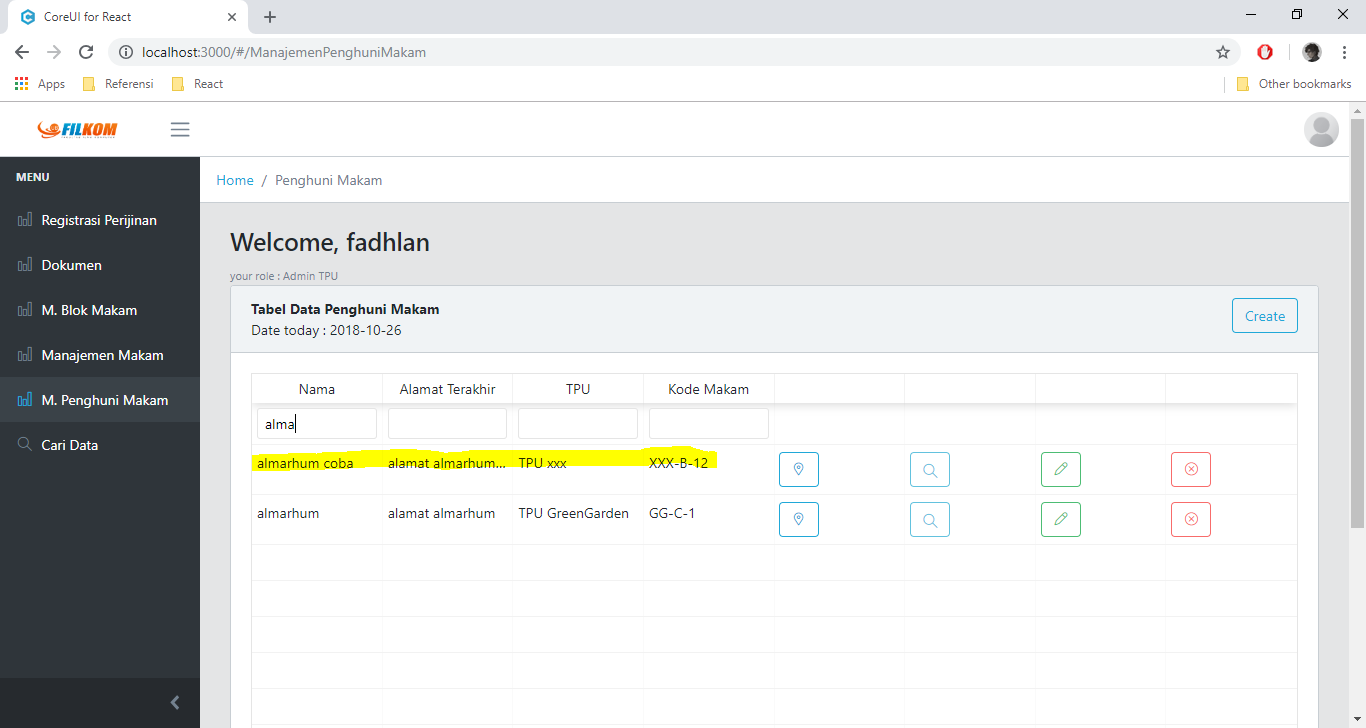
Dapat kita lihat terdapat entri baru berupa sebuah data makam baru pada baris terbawah yang memiliki data yg cocok dengan input data yg dilakukan

### Fitur Penghuni Makam

Fungsi yang akan diuji adalah fungsi create penghuni makam dengan membuat entri data baru penghuni makam melalui aplikasi



Gambar 5.32 aplikasi penghuni makam

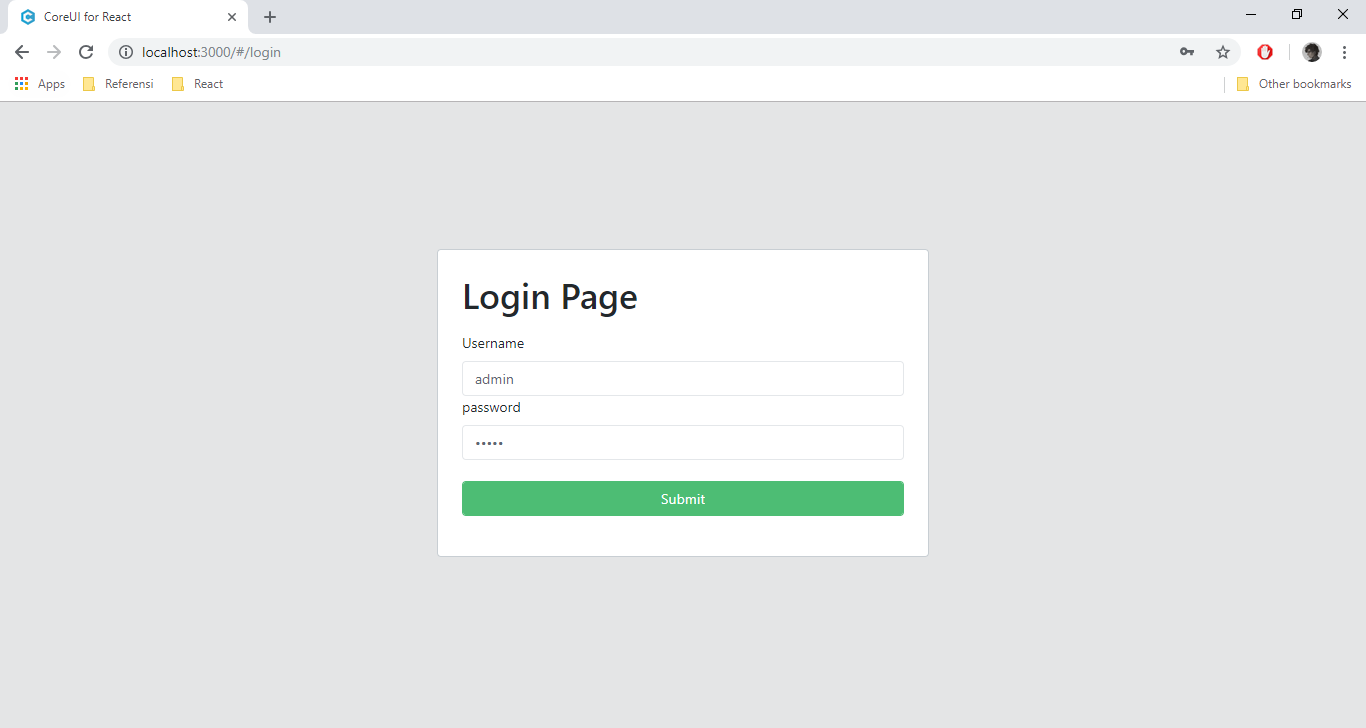


Gambar 5.33 hasil data penghuni makam setelah create

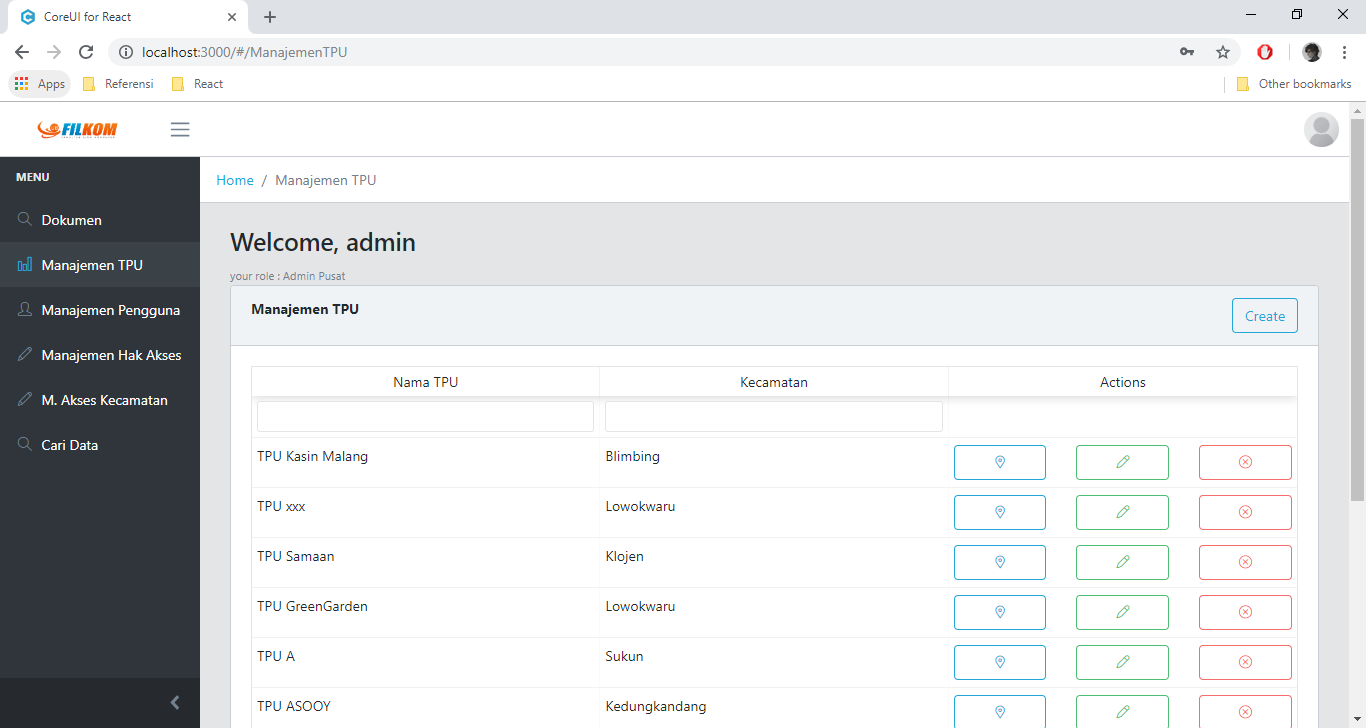
Dapat kita lihat terdapat entri data baru yang sesuai dengan input data yang dilakukan dalam pengujian.

### Fitur *Authentication*

Fungsi yang akan diuji pada fitur *Authentication* adalah fungsi signin.beriktu adalah hasil pengujiannya



Gambar 5.34 aplikasi sebelum signin

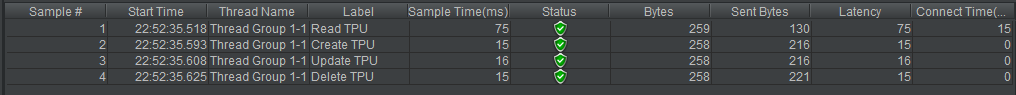


Gambar 5.35 aplikasi setelah signin sebagai admin

Setelah dilakukan pengujian fungsionalitas dari fungsi signin berkerja dan aplikasi telah masuk sebagai admin.

### Pengujian Kinerja

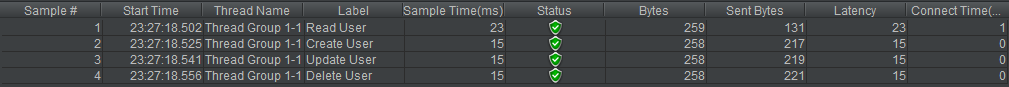
### Fitur Manajemen TPU



Gambar 5.36 Hasil Pengujian Fitur Manajemen TPU

Setelah dilakukan pengujian, semua fungsional pada Fitur Manajemen TPU berjalan dengan lancar. Terdapat empat fungsi pada Fitur Manajemen TPU, yakni read, create, update, dan delete. Fungsi read berjalan baik dengan nilai latency 75. Fungsi create, update dan delete juga berjalan baik dengan nilai latency 15.

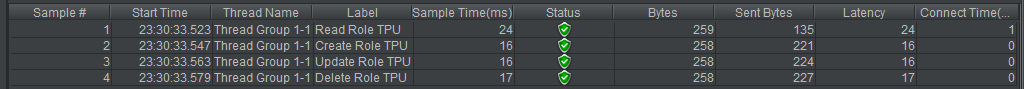
### Fitur *User*



Gambar 5.37 Hasil Pengujian Fitur Manajemen User

Setelah dilakukan pengujian, semua fungsional pada Fitur Manajemen *User* berjalan dengan lancar. Terdapat empat fungsi pada Fitur Manajemen *User*, yakni read, create, update, dan delete. Fungsi read berjalan baik dengan nilai latency 23. Fungsi create, update dan delete juga berjalan baik dengan nilai latency 15.

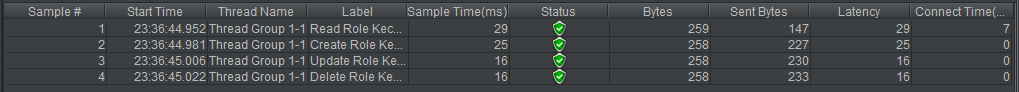
### Fitur Manajemen *User* Access



Gambar 5.38 Hasil Pengujian Fitur Manajemen User Access

Setelah dilakukan pengujian, semua fungsional pada Fitur Manajemen *User* Access berjalan dengan lancar. Terdapat empat fungsi pada Fitur Manajemen *User* Access, yakni read, create, update, dan delete. Fungsi read berjalan baik dengan nilai latency 24. Fungsi create dan update berjalan baik dengan nilai latency 15. Fungsi delete juga berjalan baik dengan nilai latency 16.

### Fitur Kecamatan Access



Gambar 5.39 Hasil Pengujian Service Kecamatan

Setelah dilakukan pengujian, semua fungsional pada Fitur manajemen kecamatan access berjalan dengan lancar. Terdapat empat fungsi pada Fitur manajemen kecamatan access, yakni read, create, update, dan delete. Fungsi read berjalan baik dengan nilai latency 29. Fungsi create berjalan baik dengan nilai latency 25. Fungsi update dan delete juga berjalan baik dengan nilai latency 16.

### Fitur Registrasi Perijinan



Gambar 5.40 Hasil Pengujian FItur Registrasi Perijinan

Setelah dilakukan pengujian, fungsional pada FItur Registrasi Perijinan berjalan dengan lancar. Terdapat satu fungsi pada FItur Registrasi Perijinan, yakni registrasi perijinan. Fungsi registrasi perijinan berjalan baik dengan nilai latency 41.

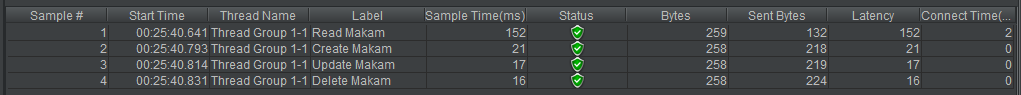
### Fitur Dokumen



Gambar 5.41 Hasil Pengujian Fitur Dokumen

Setelah dilakukan pengujian, semua fungsional pada Fitur Dokumen berjalan dengan lancar. Terdapat dua fungsi pada Fitur Dokumen, yakni Cetak Surat Permohonan dan Cetak Surat Perizinan. Fungsi Cetak Surat Permohonan berjalan baik dengan nilai latency 547. Fungsi Cetak Surat Perizinan juga berjalan dengan baik dengan nilai latency 508.

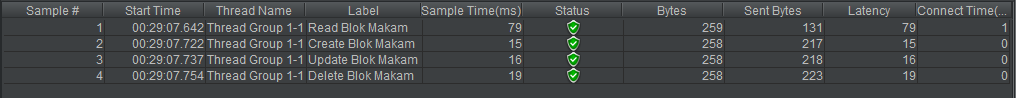
### Fitur Makam



Gambar 5.42 Hasil Pengujian Fitur Manajemen Makam

Setelah dilakukan pengujian, semua fungsional pada Fitur Manajemen Makam berjalan dengan lancar. Terdapat empat fungsi pada Fitur Manajemen Makam, yakni read, create, update, dan delete. Fungsi read berjalan baik dengan nilai latency 152. Fungsi create berjalan baik dengan nilai latency 21. Fungsi update berjalan baik dengan nilai latency 17. Dan fungsi delete berjalan baik dengan nilai latency 16.

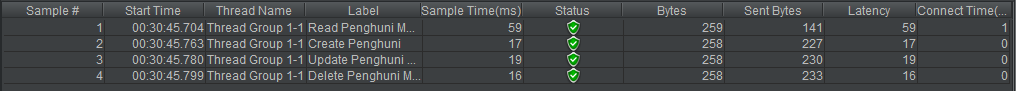
### Fitur Blok Makam



Gambar 5.43 Hasil Pengujian Fitur Manajemen Blok Makam

Setelah dilakukan pengujian, semua fungsional pada Fitur Manajemen Blok Makam berjalan dengan lancar. Terdapat empat fungsi pada Fitur Manajemen Blok Makam, yakni read, create, update, dan delete. Fungsi read berjalan baik dengan nilai latency 79. Fungsi create berjalan baik dengan nilai latency 15. Fungsi update berjalan baik dengan nilai latency 16. Dan fungsi delete berjalan baik dengan nilai latency 19.

### Fitur Penghuni Makam



Gambar 5.44 Hasil Pengujian Fitur Manajemen Penghuni Makam

Setelah dilakukan pengujian, semua fungsional pada Fitur Manajemen Penghuni Makam berjalan dengan lancar. Terdapat empat fungsi pada Fitur Manajemen Penghuni Makam, yakni read, create, update, dan delete. Fungsi read berjalan baik dengan nilai latency 59. Fungsi create berjalan baik dengan nilai latency 17. Fungsi update dan delete juga berjalan baik dengan nilai latency 19.

### Fitur *Authentication*



Gambar 5.45 Hasil Pengujian Fitur Authentication

Setelah dilakukan pengujian, semua fungsional pada Fitur *Authentication* berjalan dengan lancar. Terdapat satu fungsi pada Fitur *Authentication*, yakni Sign in. Fungsi sign in berjalan dengan baik dengan nilai latency 133.

### Fitur Mail



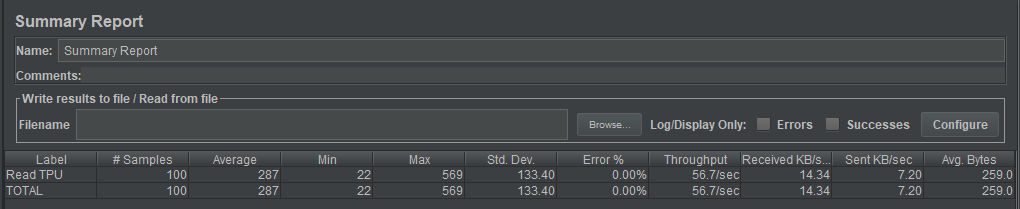
Gambar 5.46 Hasil Pengujian Service Mail

Setelah dilakukan pengujian, semua fungsional pada *service* Mail berjalan dengan lancar. Terdapat satu fungsi pada Fitur *Authentication*, yakni Sign in. Fungsi sign in berjalan dengan baik dengan nilai latency 3453.

### Pengujian Skalabilitas

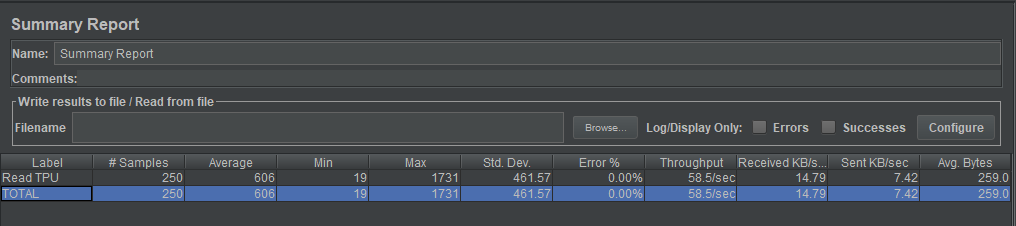
Pengujian skalabilitas dilakukan untuk mengetahui berapa maksimal *user* yang dapat dilayani dalam satu waktu.

Skenario 1 dilakukan 100 HTTP *request* dalam waktu bersamaan. Pengujian ini selesai dalam waktu 2 detik dengan tidak ada error pada semua *request*.



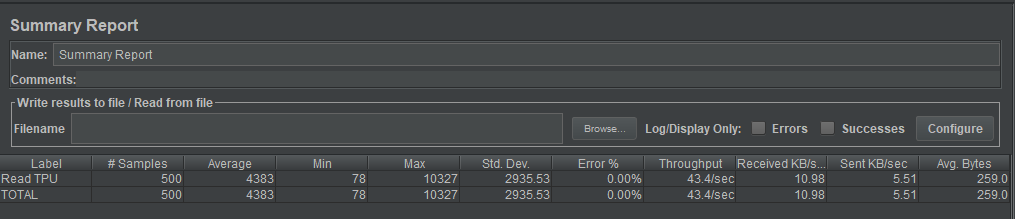
Gambar 5.47 Hasil Pengujian Skalabilitas Skenario 1

Skenario 2 dilakukan 250 HTTP *request* dalam waktu bersamaan. Pengujian ini selesai dalam waktu 4 detik dengan tidak ada error pada semua *request*.



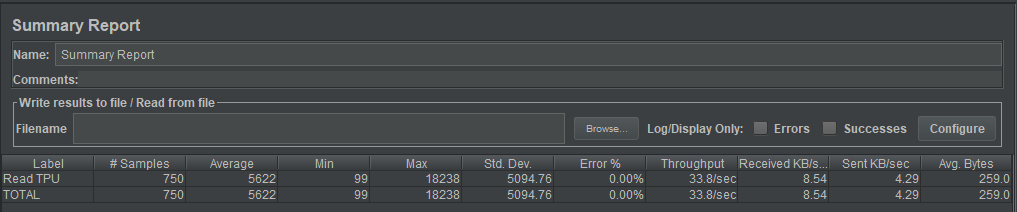
Gambar 5.48 Hasil Pengujian Skalabilitas Skenario 2

Skenario 3 dilakukan 500 HTTP *request* dalam waktu bersamaan. Pengujian ini selesai dalam waktu 12 detik dengan tidak ada error pada semua *request*.



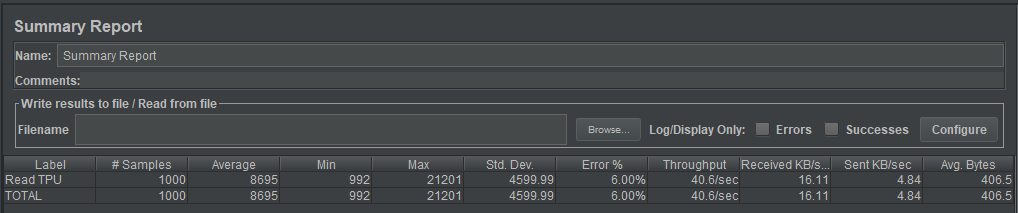
Gambar 5.49 Hasil Pengujian Skalabilitas Skenario 3

Skenario 4 dilakukan 750 HTTP *request* dalam waktu bersamaan. Pengujian ini selesai dalam waktu 22 detik dengan tidak ada error pada semua *request*.



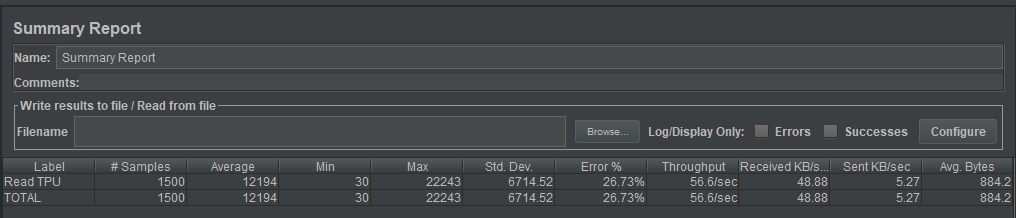
Gambar 5.50 Hasil Pengujian Skalabilitas Skenario 4

Skenario 5 dilakukan 1000 HTTP *request* dalam waktu bersamaan. Pengujian ini selesai dalam waktu 24 detik. Namun pada pengujian ini terdapat 6% error dari semua *request* yang dikirim.



Gambar 5.51 Hasil Pengujian Skalabilitas Skenario 5

Skenario 6 dilakukan 1500 HTTP *request* dalam waktu bersamaan. Pengujian ini selesai dalam waktu 29 detik. Namun pada pengujian ini terdapat 26,73% error dari semua *request* yang dikirim.



Gambar 5.52 Hasil Pengujian Skalabilitas Skenario 6

Gambar 5.53 Grafik Error Rate

Setelah dilakukan pengujian skalabilitas, error tidak ditemukan ketika jumlah *request* yang dikirimkan adalah 100, 250, 500, dan 750. Namun, terdapat error sebesar 6% dari 1000 paket yang dikirimkan. Kemudian terdapat error sebesar 26.73% dari 1500 paket yang dikirimkan. Kesimpulan yang didapat dari grafik ini adalah terlihat bahwa sistem mengalami kesulitan dalam menangani *request* dengan jumlah lebih dari 1000.

Gambar 5.54 Grafik Troughput Rate

Setelah dilakukan pengujian skalabilitas, troughput tidak stabil dan mengalami naik turun dalam setiap skenario yang dilakukan. Ketika jumlah *request* adalah 100, troughput yang didapat adalah 56,7. Ketika jumlah *request* ditambah menjadi 250 troughput mengalami kenaikan menjadi 58,5. Lalu ketika jumlah *request* ditambah menjadi 500 dan 750 troughput menurun menjadi 43,4 dan 33,8. Namun sistem tetap bisa melayani semua *request* tanpa error. Lalu troughput mengalami kenaikan kembali ketika *request* ditambah menjadi 1000 dan 1500, hal ini terjadi dikarenakan dari jumlah *request* yang dikirimkan, tidak semua dapat dilayani oleh sistem. Terdapat error rate yang dapat dilihat pada grafik sebelumnya. Terdapat kurang lebih 60 request yang tidak dapat dilayani ketika dilakukan pengujian 1000 request. Dan terdapat 401 request yang tidak dapat dilayani ketika dilakukan pengujian 1500 request.

# PENUTUP

## Kesimpulan

Sistem Terdistribusi berarti membagi sistem dalam beberapa proses terpisah. Sesuai implementasi sistem terdistribusi yang kami lakukan pada layanan E-makam Kota Malang, semua proses *backend* dibagi menjadi beberapa *service*-*service* kecil terpisah dengan *frontend*. Aplikasi akan mengirimkan *request* *service* tertentu kepada API *Gateway*, kemudian API *Gateway* akan mengakses *service* tersebut dan mengirimkan hasilnya kepada Aplikasi. *Service*-*service* tersebut dapat diletakkan di beberapa server yang berbeda dan tidak akan mempengaruhi kinerja sistem secara keseluruhan.

Performa Arsitektur *Microservice* pada layanan E-Makam Pemerintah Kota Malang memiliki beberapa keunggulan diantaranya *maintenance* lebih mudah, aplikasi aman dan handal, dan setiap *service* berdiri sendiri tanpa mengganggu *service* yang lain. Sesuai dengan pengujian yang telah dilakukan pada bab sebelumnya, sistem terdistribusi berbasis *microservice* terbukti dapat menangani 750 *Request* secara bersamaan.

## Saran

Saran yang perlu dipertimbangkan untuk pengembangan penelitian ini adalah *service*-*service* dapat dibagi ke beberapa server yang berbeda agar dapat melayani jumlah *user* yang lebih banyak, dan membuat *backend* server bekerja lebih ringan.

# DAFTAR PUSTAKA

Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., Casallas, R. and Gil, S., 2015, September. Evaluating the monolithic and the *microservice* architecture pattern to deploy web applications in the cloud. In Computing Colombian Conference (10CCC), 2015 10th (pp. 583-590). IEEE.

Rodriguez, A., 2008. Restful *webservice*s: The basics. IBM *developer*Works, 33.

Chen, L., 2018. *Microservice*s: Architecting for Continuous Delivery and DevOps. *Proceedings - 2018 IEEE 15th International Conference on Software Architecture, ICSA 2018*, hal.39–46.

Coulouris, G., Dollimore, J. dan Kindberg, T., 2012. *Distributed Systems: Concepts and Design*. [daring] *Computer*, Tersedia pada: <http://www.amazon.com/dp/0321263545>.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N. and Weerawarana, S., 2002. Unraveling the *Webservice*s web: an introduction to SOAP, WSDL, and UDDI. IEEE Internet computing, 6(2), pp.86-93.

J. Stubbs, W. Moreira and R. Dooley, "Distributed *System*s of *Microservice*s Using Docker and Serfnode," 2015 7th International Workshop on Science *Gateway*s, Budapest, 2015, pp. 34-39 .doi: 10.1109/IWSG.2015.16

Alfat, L., Triwiyatno, A., Isnanto R., 2015. Sentinel Web: Implementation of Laravel *Framework* in Web Based Temperature and Humidity *Monitoring* *System*. In: IEEE, 2015.*Proc. of 20l5 2nd Int. Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*. Semarang, Indonesia, Oct l6-l8th. Semarang: IEEE

Booth, D., Haas, H., 2004. *Webservice*s Architecture. *W3C Working Group Note*, [online] Tersedia di: < https://www.w3.org/TR/ws-arch/ >

[Diakses 21 Oktober 2018]

Lu, D., Huang, D., Walenstein, A., dan Medhi, D., 2017. A Secure *Microservice* *Framework* for IoT. In: IEEE, 2017. *2017 IEEE Symposium on Service-Oriented System Engineering*. San Francisco, CA, USA, Oct l6-l8th. San Francisco: IEEE