**LEMBAR PENGESAHAN**

**Aplikasi e-Monitoring Program Pengelolaan dan Pengembangan Sistem Penyediaan Air Minum dan Sistem Air Limbah**

Yang dipersiapkan dan disusun oleh :

**ILHAM TAUFIK**

**NIM : 177200026**

Cianjur, \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020

Dosen Kerja Praktik, Pembimbing Perusahaan,

**Asep Deden Rahmat S,ST.,M.Kom., Hari Ramdan. M.**

NIDN: 0405027903Staff Bidang Sanitasi

Mengetahui ;

Kaprodi Teknik Informatika, Dekan Fakultas Teknik,

**Eva Susilawati, S.Kom, M.M. Rysa Sahrial, ST., M.Kom.**

**LEMBAR PERTANGGUNG JAWABAN**

**Segala Isi dari Penulisan Laporan Kerja Praktek ini**

**Sepenuhnya Tanggung Jawab Penulis**

**Cianjur, Desember 2020**

**ILHAM TAUFIK**

**NIM. 177200026**

# KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan berkat-Nya, sehingga penulis dapat menyelesaikan Kerja Praktek di Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cinjuar. Kerja Praktek ini merupakan tugas yang diajukan untuk memenuhi syarat dalam memperoleh gelar Sarjana Teknik Informatik pada Universitas Putra Indonesia Cianjur. Semoga segala bantuan yang telah diberikan kepada penulis mendapat balasan yang setimpal dari Allah SWT.

Penulis menyadari masih banyak kekurangan dalam penyusunan laporan Magang ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan Kerja Praktek ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Cianjur, 22 Desember 2020

Ilham Taufik

177200026

# DAFTAR ISI

[KATA PENGANTAR iii](#_Toc62473825)

[DAFTAR ISI iv](#_Toc62473826)

[DAFTAR TABEL vi](#_Toc62473827)

[DAFTAR GAMBAR vii](#_Toc62473828)

[BAB I PENDAHULUAN 1](#_Toc62473829)

[1.1. Latar Belakang 1](#_Toc62473830)

[1.2. Identifikasi Masalah 3](#_Toc62473831)

[1.3. Rumusan Masalah 3](#_Toc62473832)

[1.4. Batasan Masalah 4](#_Toc62473833)

[1.5. Metode Pengumpulan Data 4](#_Toc62473834)

[1.6. Tujuan Penelitian 5](#_Toc62473835)

[BAB II KAJIAN PUSTAKA 8](#_Toc62473836)

[2.1. Sejarah 8](#_Toc62473837)

[2.1.1. Struktur Organisasi Bidang Air Bersih dan Sanitasi 9](#_Toc62473838)

[2.1.2. Biodata 10](#_Toc62473839)

[2.1.3. Visi & Misi 10](#_Toc62473840)

[2.1.4. Peta Lokasi 11](#_Toc62473841)

[2.2. Bidang Air Bersih Dan Sanitasi 11](#_Toc62473842)

[2.3. Aplikasi 11](#_Toc62473843)

[2.3.1. Jogiyanto (1999:12) 12](#_Toc62473844)

[2.3.2. Rachmad Hakim S (2010:15) 12](#_Toc62473845)

[2.4. Data 12](#_Toc62473846)

[2.5. Informasi 13](#_Toc62473847)

[2.6. Sistem Informasi 14](#_Toc62473848)

[2.7. Database 14](#_Toc62473849)

[2.8. UML 15](#_Toc62473850)

[2.9. Node.js 17](#_Toc62473851)

[2.10. Heroku 18](#_Toc62473852)

[2.11. Strapi 19](#_Toc62473853)

[2.12. MariaDB 20](#_Toc62473854)

[2.13. Vue.js 22](#_Toc62473855)

[2.14. Vuesax 25](#_Toc62473856)

[2.15. API 26](#_Toc62473857)

[2.16. Cypress 27](#_Toc62473858)

[2.17. GrphQL 28](#_Toc62473859)

[2.18. Visual Studio Code 28](#_Toc62473860)

[BAB III SARANA DAN PRASARANA 31](#_Toc62473861)

[3.1 Sarana 31](#_Toc62473862)

[3.1.1 Perangkat Keras 31](#_Toc62473863)

[3.1.2 Perangkat Lunak Pendukung 32](#_Toc62473864)

[3.2 **Prasarana** 32](#_Toc62473865)

[BAB IV ANALISIS DAN PEMBAHASAN 33](#_Toc62473866)

[4.1 Analisis Sistem yang Sedang Berjalan 33](#_Toc62473867)

[4.2 Analisis Sistem yang Akan Diusulkan 33](#_Toc62473868)

[4.3 Perancangan Sistem 35](#_Toc62473869)

[4.3.1. Use Case Diagram 35](#_Toc62473870)

[4.3.2. Class Diagram 36](#_Toc62473871)

[4.3.3. Sequence Diagram 37](#_Toc62473872)

[4.3.4. Activity Diagram User 41](#_Toc62473873)

[4.3.5. Perancangan Antarmuka 42](#_Toc62473874)

[4.4 Perancangan Basis Data 53](#_Toc62473875)

[4.4.1. Relasi Database 54](#_Toc62473876)

[BAB V IMPLEMENTASI DAN PENGUJIAN 55](#_Toc62473877)

[5.1 Implementasi 55](#_Toc62473878)

[5.2 Pengujian 63](#_Toc62473879)

[BAB VI PENUTUP 68](#_Toc62473880)

[6.1 Kesimpulan 68](#_Toc62473881)

[6.2 Saran 68](#_Toc62473882)

[DAFTAR PUSAKA 69](#_Toc62473883)

[BIODATA PENULIS 70](#_Toc62473884)

[LAMPIRAN 71](#_Toc62473885)

# DAFTAR TABEL

[Tabel 1: Rekap Kontrak 53](#_Toc62473886)

[Tabel 2: Koordinat 53](#_Toc62473887)

[Tabel 3: MCK/Sanitasi 53](#_Toc62473888)

[Tabel 4: Air Minum 53](#_Toc62473889)

[Tabel 5: Usulan 54](#_Toc62473890)

# DAFTAR GAMBAR

[Gambar 1: Use Case Diagram 35](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473891)

[Gambar 2: Class Diagram 36](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473892)

[Gambar 3: Sequence Diagram 37](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473893)

[Gambar 4: Sequence Diagram Login 38](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473894)

[Gambar 5: Sequence Diagram Daftar SPAM 39](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473895)

[Gambar 6: Sequence Diagram Daftar Sanitasi 40](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473896)

[Gambar 7: Rancangan Halaman Login 42](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473897)

[Gambar 8: Rancangan Halaman Dashboard 43](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473898)

[Gambar 9: Rancangan Halaman Rekap Kontrak 44](#_Toc62473899)

[Gambar 10: Rancangan Halaman Tambah Rekap Kontrak 45](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473900)

[Gambar 11: Rancangan halaman Ubah Rekap Kontrak 46](#_Toc62473901)

[Gambar 12: Rancangan Halaman Air Minum 47](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473902)

[Gambar 13: Rancangan Halaman Tambah Air Minum 48](#_Toc62473903)

[Gambar 14: Rancangan Halaman Ubah Air Minum 49](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473904)

[Gambar 15: Rancangan Halaman Sanitasi 50](#_Toc62473905)

[Gambar 16: Rancangan Halaman Tambah Sanitasi 51](#_Toc62473906)

[Gambar 17: Rancangan Halaman Ubah Sanitasi 52](#_Toc62473907)

[Gambar 18: Halaman Login 56](#_Toc62473908)

[Gambar 19: Halaman Dashboard 57](#_Toc62473909)

[Gambar 20: Halaman Rekap Kontrak 58](#_Toc62473910)

[Gambar 21: Halaman Input Data Kontrak 59](#_Toc62473911)

[Gambar 22: Halaman Pembangunan SPAM 60](#_Toc62473912)

[Gambar 23: Halaman Input Daftar SPAM 60](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473913)

[Gambar 24: Halam Daftar Pembangunan Sanitasi 61](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473914)

[Gambar 25: Halaman Input Daftar Pembangunan Sanitasi 61](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473915)

[Gambar 26: Hasil Pengujian Halaman Login 63](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473916)

[Gambar 27: Hasil Pengujian Halaman Rekap Kontrak 64](file:///C:\Users\Administrator\Documents\kerja-praktek\LAPORAN_KP.docx#_Toc62473917)

[Gambar 28: Hasil Pengujian Halaman Input Data Kontrak 66](#_Toc62473918)

[Gambar 29: Hasil Pengujian Navigasi Aplikasi 67](#_Toc62473919)

# BAB I PENDAHULUAN

## Latar Belakang

Perkembangan *Information Technology* (IT) telah mengubah cara kerja manusia dalam menjalankan aktifitas sehari-hari. IT telah membawa perubahan yang sangat fundamental bagi organisasi baik swasta maupun publik. Hal ini terjadi karena IT mampu bekerjasama dengan banyak bidang ilmu lainnya, termasuk di bidang pemerintahan. Pada instansi pemerintahan tidak bisa dilepaskan pentingnya IT yang berperan mengumpulkan, mengelola, melaporkan suatu informasi sebagai bahan evaluasi aktifitas kerja.

*Monitoring* perkembangan kegiatan merupakan faktor penting keberhasilan suatu program kegiatan. Apabila IT diterapkan pada sistem pelaporan (elektronik) dapat dikenal dengan istilah *e-monitoring*.

*E-monitoring* merupakan pemantauan dan pelaporan dengan penyampaian data secara elektronik (daring) serta dapat dipantau secara terus menerus untuk penilaian terhadap kualitas dan efektivitas sistem pengendalian untuk meyakinkan bahwa pengendalian telah berjalan sebagaimana yang diharapkan dan diperbaiki sesuai dengan kebutuhan. Dalam upaya menyelenggarakan pembangunan dan penerapan tatakelola pemerintahan yang baik (*good* *govermance*) Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan mempunyai Program Pengelolaan dan Pengembangan Sistem Penyediaan Air Minum dan Sistem Air Limbah. Program tersebut bertujuan untuk Meningkatkan ketersediaan air baku dan air minum untuk masyarakat dan Persentase Rumah Tangga Bersanitasi.

Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur selaku dinas pelaksana bertanggung jawab untuk melakukan *monitoring* terhadap; jumlah Sistem Penyediaan Air Minum (SPAM) terbangun dan terpelihara, jumlah Instalasi Pembuangan Air Limbah (IPAL) terbangun dan terpelihara, dan jumlah MCK terbangun dan terpelihara. Permasalahan yang sering terjadi ketika melakukan inventarisasi kegiatan-kegiatan yang diusulkan atau yang sudah dilaksanakan selalu tidak sinkron dengan data-data yang ada. Berawal dari permasalahan tersebut kemudian menjadi kerangka berfikir dalam membangun Aplikasi e-Monitoring Program Pengelolaan dan Pengembangan Sistem Penyediaan Air Minum dan Sistem Air Limbah, yang merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting.

Aplikasi yang dibangun ini diharapkan dapat mengolah data informasi pelaporan secara teratur, tepat waktu, lengkap dan faktual mengatasi permasalahan-permasalahan yang sering ditemui. Aplikasi ini memuat pengelolaan kegiatan-kegiatan yang diusulkan pada tahun anggaran berikutnya maupun yang sedang atau talah dilaksanakan. Data tersebut akan dikelola kedalam database. Bahasa pemrograman yang digunakan pada penelitian ini menggunakan pemrograman JavaScript dengan *framework* Vue JS.

## Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan di atas maka munculah sebuah identifikasi masalah yang didapatkan yaitu sebagai berikut:

1. Data-data usulan dari desa-desa di Kabupaten Cianjur yang tersedia masih dikelola secara manual dengan catatan di dalam buku dan juga aplikasi Microsoft Office. Semua data tersebut masih belum terorganisir karena belum adanya integrasi dalam pengolahan data.
2. Dalam setiap pembuatan laporan, data-data kegiatan yang sudah/belum dilaksanakan harus dicari dan disusun manual terlebih dahulu untuk diolah menjadi sebuah laporan.
3. Untuk memonitoring kegiatan-kegiatan yang telah dilaksanakan sebagai bahan laporan untuk peningkatan.

## Rumusan Masalah

Dari uraian diatas penulis mencoba membuat rumusan masalah antara lain sebagai berikut:

1. Bagaimana membangun aplikasi yang berfungsi untuk menyimpan, dan mengolah data yang ada.
2. Bagaimana aplikasi tersebut dapat memberikan data yang telah diproses untuk laporan.
3. Bagaimana kepala bidang, kasi, dan staf lainnya dapat memonitoring semua usulan kegiatan yang ada di Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur.

## Batasan Masalah

Sesuai dengan permasalahan yang terjadi pada Rumusan Masalah di atas, penulis membuat batasan-batasan masalah, yaitu:

1. Aplikasi ini meliputi pengolahan data semua kegiatan di Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan.
2. Aplikasi ini dapat mencatat semua kegiatan yang sebelumnya dilakukan manual menggunakan aplikasi Microsoft Office.
3. Aplikasi ini dapat mengolah data dari data-data kegiatan yang dimasukkan kemudian dapat menjadi bahan laporan periodik.

## Metode Pengumpulan Data

Pengumpulan data dilakukan dengan cara:

1. Observasi Langsung  
   Obeservasi langsung adalah teknik pengumpulan data dengan cara mengamati langsung operasi maupun prosedur yang berlaku pada objek penelitian
2. Wawancara  
   Wawancara merupakan teknik pengempulan data dengan mengadakan wawancara dengan pihak-pihak terkait guna mendapatkan data-data serta keterangan yang dibutukan oleh penulis. Dalam hal ini penulis melakukan tanya jawab dengan pejabat di bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan secara langsung dan staff administrasi terkait dengan prosedur sistem yang digunakan sehingga diharapkan dapat memberikan solusi terbaik dalam pembangunan aplikasi.
3. Studi Pustaka

Selain pengambilan data secara langsung, penyusun juga melakukan studi literatur demi tercapainya tujuan pembangunan sistem informasi sehingga dapat memenuhi kriteria pembangunan sistem informasi yang baik dan sistematis. Studi literatur ini yaitu suatu metode pengumpulan data dengan membaca dan mempelajari buku, literatur, artikel, karya ilmiah, catatan perkuliahan, dan sumber-sumber lainnya yang berkaitan dengan topik penelitian.Studi literatur ini dimaksudkan penyusun untuk membandingkan serta mencocokkan antara fakta yang terjadi di dunia nyata dengan aspek-aspek yang ada dalam pembangunan sistem informasi.

## Tujuan Penelitian

Adapun tujuan dari penelitian:

1. Membangun aplikasi yang memudahkan dalam melakukan pelaporan usulan-usulan kegiatan pada Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur.
2. Membangun aplikasi yang dapat menghemat waktu dalam pembuatan laporan pada Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur.
3. Membangun aplikasi yang memudahkan untuk melakukan monitoring semua usuan kegiatan oleh Kepala Bidang di Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur.
4. Manfaat Penelitian

Adapun manfaat dari perancangan sistem informasi adalah:

1. Manfaat bagi penulis adalah sebagai sarana memperluas dan mengimplementasi pengetahuan penulis dalam proses pengoalahan data yang terjadi di Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanaahan Kabupaten Cianjur.
2. Manfaat bagi OPD adalah sebagai sarana pendukung operasional yang diharapkan dapat mempermudah dalam melakukan pelaporan maupun monitoring.
3. Lokasi Penelitian

Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur, Jl. Adi Sucipta No.7 Kelurahan Sayang, Kecamatan Cianjur, Kabupaten Cianjur, Jawa Barat.

1. Sistematika Penulisan

Sistematika penulisan laporan kerja praktek ini adalah sebagai berikut:

BAB I: Pendahuluan

Pada bab ini penulis menguraikan latar belakang, perumusan masalah, tujuan dan manfaat kerja praktek, ruang lingkup pembahasan, metode penulisan dan sistematika penulisan.

BAB II : Kajian Pustaka

Pada bab ini penulis menguraikan mengenai sejarah singkat dan profil Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur, struktur organisasi, visi dan misi yang terdapat di tempat kerja praktek.

BAB III: Sarana dan Prasarana

Pada bab ini penulis menguraikan mengenai sarana dan prasarana pada Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur yang menunjang kebutuhan pembuatan Aplikasi e-Monitoring Bidang Air Bersih dan Sanitasi.

BAB IV: Analisa dan Pembahasan

Pada bab ini penulis menguraikan tentang sistem yang sedang berjalan, perancangan basis data, fungsi dan kegunaan sistem, diagram aliran data, kelemahan dan keuntungan sistem yang dirancang.

BAB V: Implementasi dan Pengujian

Pada bab ini penulis menguraikan tentang implementasi dan pengujian Aplikasi e-Monitoring Bidang ABS yang telah di terapkan.

BAB VI: Penutup

Pada bab ini, penulis mencoba menarik kesimpulan dari keseluruhan sistem yang dibuat serta saran yang membangun untuk pengembangan sistem yang lebih baik, agar dapat bermanfaat bagi penulis secara pribadi, bagi Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur, maupun bagi pihak–pihak yang merasa membutuhkan atas penyusunan laporan kerja praktik.

# BAB II KAJIAN PUSTAKA

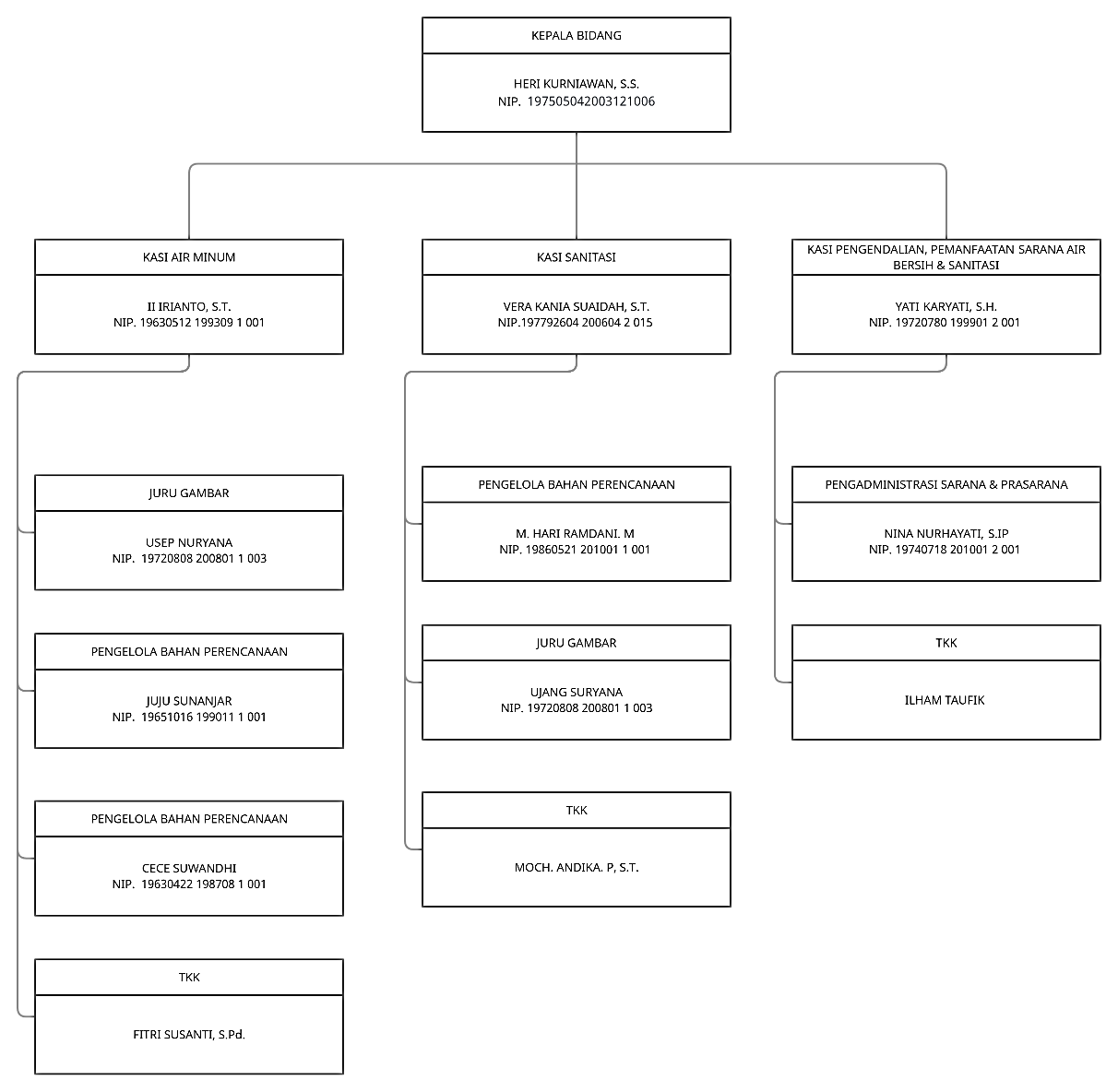
## Sejarah

Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur merupakan salah satu unsur Organisasi Pemerintah Daerah (OPD) Kabupaten Cianjur yang mempunyai Tugas pokok melaksanakan urusan pemerintahan daerah bidang permukiman dan perumahan berdasarkan asas otonomi, dekonsentrasi dan tugas pembantuan, serta kebijakan teknis urusan bidang permukiman dan perumahan yang meliputi tata permukiman dan perumahan.

Sejarah Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur berhubungan erat dengan Sejarah Pekerjaan Umum (PU). Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur merupakan salah satu Dinas yang bergerak dalam bidang Ke-Cipta Karyaan, sebelumnya Dinas Tata Ruang dan Permukiman Kabupaten Cianjur (DISTARKIM) sampai pada tahun 2017 berubah menjadi Dinas Perumahan Kawasan Permukiman dan Pertanahan (DPKPP) berdasarkan Peraturan Bupati Nomor 76 Tahun 2016 tentang Tugas Dan Fungsi Serta Tata Kerja Unit Organisasi Di Lingkungan Dinas Perumahan, Kawasan Permukiman Dan Pertanahan Kabupaten Cianjur.

Bidang Air Bersih dan Sanitasi merupakan salah satu bidang yang ada pada Dinas Perumahan Kawasan Permukiman dan Pertanahan, bertugas untuk melaksanakan Program Pengembangan Kinerja Air Minum, Sanitasi, dan Air Limbah.

1. Struktur Organisasi Bidang Air Bersih dan Sanitasi



1. Biodata

Nama OPD : Bidang Air Bersih dan Sanitasi DPKPP

Alamat : Jl. Adi Sucipta No. 7 – Cianjur

Kecamatan : Cianjur

Kabupaten : Cianjur

Provinsi : Jawa Barat

Kode Pos : 43211

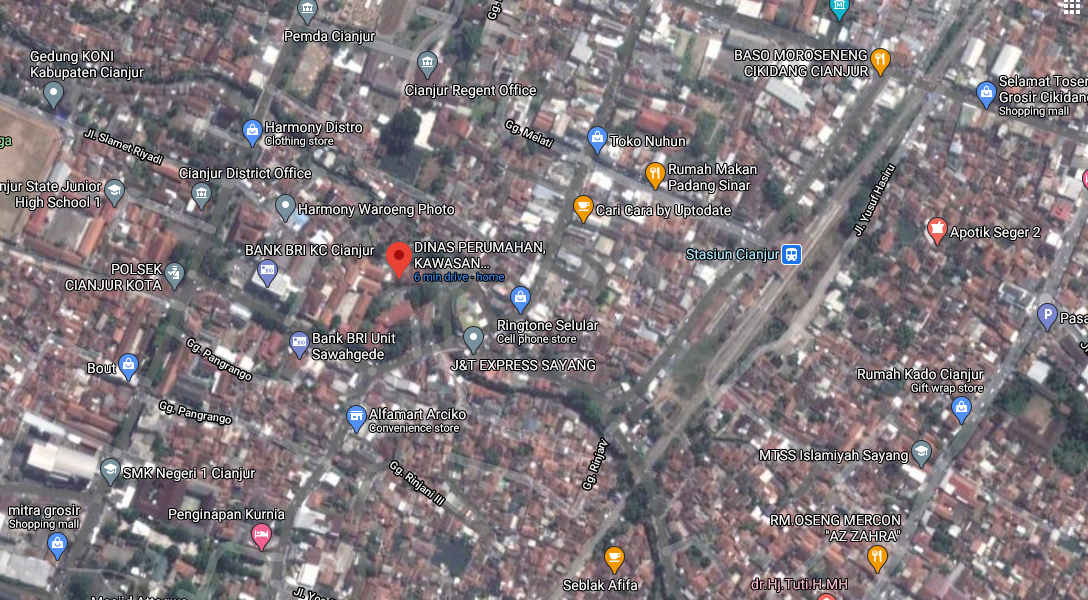
1. Visi & Misi
2. **Visi**

“Cianjur Lebih Maju dan Agamis”

1. **Misi**

Misi Pembangunan jangka menengah Kabupaten Cianjur 2016-2021 selama lima tahun kedepan sebagai komitmen untuk mencapai visi, sebagai berikut:

1. Meningkatkan pembangunan infrastruktur yang terintegrasi dan berwawasan lingkungan.
2. Meningkatkan pembangunan keagamaan.
3. Meningkatkan pembangunan manusia melalui akselerasi di bidang pendidikan, kesehatan dan ekonomi.
4. Peta Lokasi



## Bidang Air Bersih Dan Sanitasi

Bidang Air Bersih dan Sanitasi merupakan salah satu bidang yang ada pada Dinas Perumahan Kawasan Permukiman dan Pertanahan, bertugas untuk melaksanakan Program Pengembangan Kinerja Air Minum, Sanitasi, dan Air Limbah.

## Aplikasi

Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Aplikasi bisa dikatakan suatu perangkat lunak yang siap pakai dengan menjalankan instruksi-instruksi dari pengguna (*user*), aplikasi banyak diciptakan untuk membantu berbagai keperluan, seperti contoh: membuat laporan, percetakan dan lain-lain. Adapun defenisi aplikasi menurut para ahli:

1. Jogiyanto (1999:12)

Pengertian aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian sehingga komputer dapat memproses input menjadi output

1. Rachmad Hakim S (2010:15)

Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows, permainan (gim), dan sebagainya.

## Data

Data merupakan suatu kumpulan yang terdiri dari fakta-fakta untuk memberikan gambaran yang luas terkait dengan suatu keadaan. Seseorang yang akan mengambil sebuah kebijakan atau keputusan umumnya akan menggunakan data sebagai bahan pertimbangan. Melalui data seseorang dapat menganalisis, menggambarkan, atau menjelaskan suatu keadaan.

Di samping itu, data dikumpulkan melalui cara-cara tertentu kemudian diolah menjadi suatu informasi yang jelas dan mudah dipahami setiap orang. Data juga dibutuhkan di dalam berbagai macam keperluan, seperti penjualan, penelitian hingga kependudukan. Data memiliki jenis dan fungsinya masing-masing, hal ini sesuai dengan kebutuhan pengelola data.

Secara umum, data banyak digunakan untuk suatu penelitian tertentu. Namun, seiring berjalannya waktu, data dibutuhkan untuk memenuhi berbagai keperluan di berbagai bidang. Tentunya hal ini bertujuan untuk memberikan informasi yang jelas dan benar setelah data tersebut diolah lebih lanjut.

1. Nuzulla Agustina (1992:203)

Data adalah keterangan mengenai sesuatu hal yang sudah sering terjadi dan berupa himpunan fakta, angka, grafik, tabel, gambar, lambang, kata, huruf-huruf yang menyatakan sesuatu pemikiran, objek, serta kondisi dan situasi.

1. Kristanto (2008:7)

Data merupakan suatu fakta mengenai objek yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan dan kejadian.

## Informasi

Informasi adalah pesan (ucapan atau ekspresi) atau kumpulan pesan yang terdiri dari order sekuens dari simbol, atau makna yang dapat ditafsirkan dari pesan atau kumpulan pesan. Informasi dapat direkam atau ditransmisikan. Hal ini dapat dicatat sebagai tanda-tanda, atau sebagai sinyal berdasarkan gelombang. Informasi adalah jenis acara yang mempengaruhi suatu negara dari sistem dinamis. Para konsep memiliki banyak arti lain dalam konteks yang berbeda. Informasi bisa dikatakan sebagai pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi. Informasi telah digunakan untuk seluruh segi kehidupan manusia secara individual, kelompok maupun organisasi. Pada tingkat individu, informasi digunakan untuk pengetahuan tentang pendidikan, kesehatan, lapangan pekerjaan maupun jenis produk atau jasa. Kegunaan informasi ditentukan oleh tujuan pengguna, ketelitian pengolahan data, ruang dan waktu serta bentuk dan keadaan semantik. (Rusdiana, H. A., dan Irfan, M, 2014:75)

Anton M. Moeliono (1990:33) juga mendefinisikan informasi sebagai data yang telah diproses, namun pemrosesan tersebut dilakukan untuk suatu tujuan tertentu. Selanjutnya Anton M. Moeliono juga menyatakan bahwa informasi merupakan keterangan, kabar berita, pemberitahuan, penerangan, atau bahan nyata lainnya yang dapat digunakan sebagai bahan kajian analisis untuk mengambil kesimpulan atau keputusan tertentu.

## Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan

Terdapat empat langkah dalam membangun suatu sistem informasi, yaitu menentukan serta mengetahui masalah, menyediakan solusi alternatif, memilih solusi, dan mengimplementasikan solusi yang dipilih. Ketiga langkah pertama disebut sebagai analisis sistem. (Laudon, Kenneth, 2007:42).

## Database

Database adalah kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer. Pada saat pangkalan data menjadi semakin kompleks, maka pangkalan data dikembangkan menggunakan teknik perancangan dan pemodelan secara formal.

Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis data disebut sistem manajemen basis data (database management system, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan di mana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili denga menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. (Kroenke, David M. dan David J. Auer, 2007:32)

## UML

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem. (Jones Rama)

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSe, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG). (David M. Kroenke).

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. Use Case Diagram untuk memodelkan proses bisnis.
2. Conceptual Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. Sequence Diagram untuk memodelkan pengiriman pesan (*message*) antar *objects*.
4. Collaboration Diagram untuk memodelkan interaksi antar *objects*.
5. State Diagram untuk memodelkan perilaku *objects* di dalam sistem.
6. Activity Diagram untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
7. Class Diagram untuk memodelkan struktur kelas.
8. Object Diagram untuk memodelkan struktur *object*.
9. Component Diagram untuk memodelkan komponen *object*.
10. Deployment Diagram untuk memodelkan distribusi aplikasi.

## Node.js

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi client / browser saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server web seperti Apache atau Nginx.

Untuk mengeksekusi Javascript sebagai bahasa server diperlukan engine yang cepat dan mempunyai performansi yang bagus. Engine Javascript dari Google bernama V8 engine yang dipakai oleh Node.js yang juga merupakan engine yang dipakai oleh browser Google Chrome.

Berbeda dengan bahasa pemrograman sisi server pada umumnya yang bersifat blocking, Node.js bersifat non-blocking, sebagaimana halnya JavaScript bekerja. Node.js berjalan dengan basis event (event-driven). Maksud dari Blocking secara sederhana adalah, bahwa suatu kode program akan dijalankan hingga selesai, baru kemudian beralih ke kode program selanjutnya. (internet)

## Heroku

Heroku adalah sebuah cloud platform yang menjalankan bahasa pemrograman tertentu, Heroku mendukung bahasa pemrograman seperti Ruby, Node.js, Python, Java, PHP, dan lain-lain.

Heroku termasuk ke dalam kriteria Platform As A Service (PaaS), sehingga bagi anda yang ingin melakukan deploy aplikasi ke heroku cukup hanya dengan melakukan konfigurasi aplikasi yang ingin di deploy dan menyediakan platform yang memungkinkan pelanggan untuk mengembangkan, menjalankan, dan mengelola aplikasi tanpa kompleksitas membangun dan memelihara infrastruktur yang biasanya terkait dengan pengembangan dan peluncuran aplikasi.

Manfaat menggunakan Heroku adalah layanannya yaitu menjalankan script app langsung tanpa memerlukan setting yang sangat rumit, memungkinkan pengembang aplikasi lebih fokus pada kode aplikasi mereka, tanpa terlalu dipusingkan dengan arsitektur dan server. (internet)

## Strapi

Strapi adalah Headless CMS open-soure terpopuler, Strapi memberikan kebebasan kepada pengembang untuk menggunakan alat dan kerangka kerja favorit mereka.

Strapi digunakan sebagai pengelola konten atau data, Strapi ini berjalan di sisi server sebagai backend yang langsung terhubung ke database,

selain itu kita dapat mengakses data tersebut menggunakan aplikasi client seperti frontend atau mobile yang dihubungkan menggunakan rest API pada Strapi

Fitur-fitur Strapi:

1. Fleksibilitas struktur konten

Struktur jenis konten yang dapat disesuaikan dengan kebutuhan layaknya struktur dalam database, pengguna dapat mengatur jenis field dan table

1. Kustomisasi Project

seperti framework pada umumnya, pengguna dapat menggunakan aturan kerangka kerja Strapi untuk menambahkan fungsi dan alur program

1. Struktur Rest API & GraphQL

struktur konten yang sudah dibuat, selanjutnya Strapi secara ototmatis membuat struktur Rest API dan GraphQL yang dapat kita hubungkan dengan aplikasi frontend, mobile app, dll

1. Dukungan Database

dukungan basis data dengan MySQL, Postgres, SQLite, & MongoDB

1. Authentication & Permissions

mengamankan akses konten pada rest API sesuai dengan peran pengguna yang diperlukan. (internet)

## MariaDB

MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB.

MariaDB adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MariaDB, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. Semua kemampuan MySQL dimiliki pula oleh MariaDB yakni:

1. Portabilitas. MariaDB dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MariaDB didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. MariaDB dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MariaDB memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MariaDB memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan fungsi. MariaDB memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MariaDB memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan**.** MariaDB mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MariaDB dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Pelokalan bahasa. MariaDB dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antarmuka. MariaDB memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan peralatan. MariaDB dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MariaDB memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

(internet)

## Vue.js

Vue.js adalah sebuah open-source model-view-viewmodel *frontend* *framework* JavaScript untuk membangun antarmuka pengguna dan aplikasi satu halaman. Ini dibuat oleh Evan You, dan dikelola oleh dia dan anggota tim inti lainnya yang aktif.

Vue.js menampilkan arsitektur yang dapat beradaptasi secara bertahap yang berfokus pada rendering deklaratif dan komposisi komponen. Pustaka inti difokuskan pada lapisan tampilan saja. Fitur-fitur canggih yang diperlukan untuk aplikasi yang kompleks seperti perutean, manajemen status, dan perkakas build ditawarkan melalui pustaka dan paket pendukung yang dikelola secara resmi, dengan Nuxt.js sebagai salah satu solusi paling popular.

Vue dibuat oleh Evan You setelah bekerja untuk Google menggunakan AngularJS di sejumlah proyek. Dia kemudian menyimpulkan proses berpikirnya: "Saya pikir, bagaimana jika saya bisa mengekstrak bagian yang sangat saya sukai dari Angular dan membuat sesuatu yang sangat ringan." Komitmen kode sumber pertama untuk proyek ini bertanggal Juli 2013, dan Vue pertama kali dirilis pada Februari berikutnya, pada 2014.

Komponen Vue memperluas elemen HTML dasar untuk merangkum kode yang dapat digunakan kembali. Pada tingkat tinggi, komponen adalah elemen khusus yang perilaku kompilernya dilampirkan Vue. Di Vue, komponen pada dasarnya adalah instance Vue dengan opsi yang telah ditentukan sebelumnya

Vue menggunakan sintaks template berbasis HTML yang memungkinkan pengikatan DOM yang dirender ke data instance Vue yang mendasarinya. Semua template Vue adalah HTML valid yang dapat diurai oleh browser yang memenuhi spesifikasi dan parser HTML. Vue mengompilasi template ke dalam fungsi render DOM virtual. Model Objek Dokumen virtual (DOM) memungkinkan Vue merender komponen dalam memorinya sebelum memperbarui browser. Dikombinasikan dengan sistem reaktivitas, Vue mampu menghitung jumlah minimal komponen untuk dirender ulang dan menerapkan jumlah minimal manipulasi DOM ketika status aplikasi berubah.

Pengguna Vue dapat menggunakan sintaks template atau memilih untuk langsung menulis fungsi render menggunakan JSX. Fungsi render memungkinkan aplikasi dibangun dari komponen perangkat lunak. (internet)

Vue menampilkan sistem reaktivitas yang menggunakan objek JavaScript biasa dan rendering ulang yang dioptimalkan. Setiap komponen melacak ketergantungan reaktifnya selama perenderan, sehingga sistem tahu persis kapan harus merender ulang, dan komponen mana yang akan dirender ulang.

Vue menyediakan berbagai cara untuk menerapkan efek transisi ketika item dimasukkan, diperbarui, atau dihapus dari DOM. Ini termasuk alat untuk:

* Terapkan kelas secara otomatis untuk transisi dan animasi CSS
* Integrasikan pustaka animasi CSS pihak ketiga, seperti Animate.css
* Gunakan JavaScript untuk secara langsung memanipulasi DOM selama pengait transisi
* Integrasikan pustaka animasi JavaScript pihak ketiga, seperti Velocity.js

Ketika elemen yang dibungkus dalam komponen transisi disisipkan atau dihapus, inilah yang terjadi:

1. Vue secara otomatis akan mengendus apakah elemen target memiliki transisi CSS atau animasi yang diterapkan. Jika ya, kelas transisi CSS akan ditambahkan / dihapus pada waktu yang tepat.
2. Jika komponen transisi menyediakan hook JavaScript, hook ini akan dipanggil pada waktu yang tepat.
3. Jika tidak ada transisi / animasi CSS yang terdeteksi dan tidak ada kait JavaScript yang disediakan, operasi DOM untuk penyisipan dan / atau penghapusan akan segera dijalankan pada frame berikutnya.

Vue menyediakan antarmuka untuk mengubah apa yang ditampilkan pada halaman berdasarkan jalur URL saat ini - terlepas dari bagaimana itu diubah (baik dengan tautan email, refresh, atau tautan dalam halaman). Selain itu, menggunakan router front-end memungkinkan untuk transisi yang disengaja dari jalur browser ketika peristiwa browser tertentu (yaitu klik) terjadi pada tombol atau link. Vue sendiri tidak hadir dengan hash routing front-end. Tetapi paket open source "vue-router" menyediakan API untuk memperbarui URL aplikasi, mendukung tombol kembali (menavigasi riwayat), dan menyetel ulang kata sandi email atau tautan verifikasi email dengan parameter URL otentikasi. Ini mendukung pemetaan rute bersarang ke komponen bersarang dan menawarkan kontrol transisi yang sangat halus. Dengan Vue, pengembang sudah menyusun aplikasi dengan blok bangunan kecil yang membangun komponen yang lebih besar. (internet)

## Vuesax

Vuesax adalah kerangka kerja komponen UI yang dibuat dengan Vuejs untuk membuat proyek dengan mudah dan dengan gaya yang unik dan menyenangkan, vuesax dibuat dari awal dan dirancang untuk semua jenis pengembang dari pecinta frontend hingga backend yang ingin dengan mudah membuat pendekatan visualnya kepada pengguna akhir

Vuesax fokus untuk merampingkan pekerjaan programmer dengan memberikan komponen-komponen yang dibuat secara utuh dan dengan kustomisasi yang mandiri serta sangat mudah diimplementasikan, sehingga kreativitas ada di tangan kami namun kami tidak mengabaikan bahwa setiap project berbeda baik secara visual maupun ekosistemnya.

Vuesax tidak memiliki garis desain seperti kerangka kerja komponen lain berdasarkan Desain Material, mereka percaya bahwa sudah ada kerangka kerja kurus yang terlihat secara visual dan dalam UI / UX dan kami tidak ingin menjadi salah satu dari sekian banyak, selain itu kami senang membuat dan merancang pengalaman baru dan mengejutkan pengguna dengan elemen atau detail baru yang hanya dapat kami lakukan dengan bebas secara visual.

Vuesax adalah kerangka kerja yang relatif baru dengan desain yang menyegarkan dan tren terbaru, vuesax berdasarkan vuejs yang berarti bahwa kami berjalan seiring dengan salah satu kerangka kerja javascript paling populer di dunia dan dengan komunitas besar yang dengannya pengguna akan memiliki semuanya.

1. Vuesax, tidak seperti banyak kerangka kerja, dirancang dari awal dan kami tidak terikat pada garis desain mana pun, ini adalah sesuatu yang hebat karena proyek Anda akan menjadi unik dan sangat berbeda dari yang lain
2. Fokus pada pembuatan proyek yang cepat dan mudah yang memberikan garis visual yang indah tetapi tanpa melupakan personalisasi dan kemandirian pengembang
3. Vuesax menggunakan variabel css asli untuk penyesuaian yang lebih baik dan perubahan produksi seperti mengubah ke tema gelap atau mengubah warna utama seluruh aplikasi dengan beberapa baris javascript
4. Vuesax adalah *frame* yang dirancang untuk memiliki dampak visual yang hebat dan selalu menjadi tren dalam hal desain.
5. Komunitas *open-source* untuk membuat, meningkatkan, dan memperbaiki komponen atau fungsi apa pun.
6. Komponen independen untuk menghindari impor kode yang tidak perlu.
7. Dokumen penurunan harga untuk keberlanjutan yang lebih baik.
8. dan banyak lagi.

Vuesax adalah kerangka kerja komponen UI yang dapat digunakan dalam aplikasi atau proyek dan menghemat banyak waktu serta pekerjaan dalam membuatnya sendiri, komponen dibuat untuk disesuaikan dan dapat digunakan kembali sehingga pengguna dapat menggunakannya di mana saja dan logis. (internet)

## API

Sebuah antarmuka pemrograman aplikasi (API) adalah antarmuka komputasi yang mendefinisikan interaksi antara beberapa perantara perangkat lunak. Ini mendefinisikan jenis panggilan atau permintaan yang dapat dibuat, bagaimana membuatnya, format data yang harus digunakan, konvensi yang harus diikuti, dll. Ia juga dapat menyediakan mekanisme ekstensi sehingga pengguna dapat memperluas fungsionalitas yang ada dengan berbagai cara dan untuk berbagai tingkat. API dapat sepenuhnya dibuat khusus, khusus untuk suatu komponen, atau dirancang berdasarkan standar industri untuk memastikan interoperabilitas. Melalui penyembunyian informasi, API mengaktifkan pemrograman modular, memungkinkan pengguna untuk menggunakan antarmuka secara independen dari implementasi.

Dalam membangun aplikasi, API (antarmuka pemrograman aplikasi) menyederhanakan pemrograman dengan mengabstraksi implementasi yang mendasarinya dan hanya mengekspos objek atau tindakan yang dibutuhkan pengembang. Meskipun antarmuka grafis untuk klien email mungkin memberikan tombol kepada pengguna yang melakukan semua langkah untuk mengambil dan menyorot email baru, API untuk masukan / keluaran file mungkin memberi pengembang fungsi yang menyalin file dari satu lokasi ke lokasi lain tanpa mengharuskan pengembang memahami operasi sistem file yang terjadi di belakang layar. (internet)

## Cypress

Cypress merupakan alat untuk membuat dan menjalankan pengujian end-to-end berbasis framework javascript. Pengujian bekerja dengan menggunakan browser, mengunjungi halaman aplikasi, dan menjalankan serangkaian langkah yang telah ditentukan seperti yang dilakukan pengguna biasa. Bertujuan untuk memverifikasi bahwa hasilnya adalah apa yang diharapkan.

Jenis pengujian end-to-end memang lambat dibandingkan dengan pengujian unit dan integrasi, tetapi pengujian ini dapat melakukan pekerjaan luar biasa untuk memastikan aplikasi dapat berfungsi seperti yang diharapkan bagi pengguna akhir.

(internet)

## GrphQL

Graphql adalah sebuah konsep baru dalam membangun sebuah API. Graphql(*Query Language*) dikembangkan oleh Facebook dan diimplementasikan pada sisi server. Meskipun sebuah *query language* tetapi Graphql ini tidak berhubungan secara langsung dengan database, dengan kata lain GraphQL tidak terbatas untuk database tertentu baik sql ataupun nosql. Posisi Graphql ini berada pada sisi client dan server yang berhubungan / mengakses suatu API. Salah satu tujuan pengembangan bahasa query ini adalah untuk mempermudah komunikasi data antara backend dan frontend/mobile aplikasi.

GraphQL dapat diimplementasikan di berbagai bahasa sisi client seperti react, vue, meteor, dll. Apapun jenis framework nya selama dapat mengakses data dengan API. (internet)

## Visual Studio Code

Visual Studio Code adalah editor kode sumber gratis yang dibuat oleh Microsoft untuk Windows, Linux, dan macOS. Fitur termasuk dukungan untuk debugging, syntax highlighting, intelligent code complete, snippet, code refactoring, dan embedded Git. Pengguna dapat mengubah tema, pintasan keyboard, preferensi, dan memasang ekstensi yang menambahkan fungsionalitas tambahan.

Microsoft telah merilis Visual Studio Kode ini kode sumber pada VSCode repositori GitHub.com, di bawah permisif Lisensi MIT, sedangkan rilis disusun adalah freeware.

Dalam Survei Pengembang Stack Overflow 2019, Visual Studio Code menduduki peringkat alat lingkungan pengembang paling populer, dengan 50,7% dari 87.317 responden melaporkan bahwa mereka menggunakannya.

Visual Studio Code diumumkan pada 29 April 2015 oleh Microsoft pada konferensi Build 2015. Sebuah build Pratinjau dirilis segera setelah itu.

Pada tanggal 18 November 2015, Visual Studio Code dirilis di bawah Lisensi Expat dan kode sumbernya dikirim ke GitHub. Dukungan perpanjangan juga diumumkan. Pada 14 April 2016, Visual Studio Code lulus tahap pratinjau publik dan dirilis ke web

Visual Studio Code adalah editor kode sumber yang dapat digunakan dengan berbagai bahasa pemrograman, termasuk Java, JavaScript, Go, Node.js, dan C++. Ini didasarkan pada kerangka kerja Electron, yang digunakan untuk mengembangkan aplikasi Web Node.js yang berjalan pada mesin tata letak Blink. Visual Studio Code menggunakan komponen editor yang sama (nama kode "Monaco") yang digunakan di Azure DevOps (sebelumnya disebut Visual Studio Online dan Visual Studio Team Services).

Alih-alih sistem proyek, ini memungkinkan pengguna untuk membuka satu atau lebih direktori, yang kemudian dapat disimpan di ruang kerja untuk digunakan kembali di masa mendatang. Ini memungkinkannya untuk beroperasi sebagai editor kode bahasa-agnostik untuk bahasa apa pun. Ini mendukung sejumlah bahasa pemrograman dan serangkaian fitur yang berbeda per bahasa. File dan folder yang tidak diinginkan dapat dikecualikan dari pohon proyek melalui pengaturan. Banyak fitur Visual Studio Code tidak diekspos melalui menu atau antarmuka pengguna tetapi dapat diakses melalui palet perintah.

Visual Studio Code dapat diperpanjang melalui ekstensi, tersedia melalui repositori pusat. Ini termasuk penambahan editor dan dukungan bahasa. Fitur penting adalah kemampuan untuk membuat ekstensi yang menambahkan dukungan untuk bahasa baru, tema, dan debugger, melakukan analisis kode statis, dan menambahkan linter kode menggunakan Protokol Server Bahasa.

Visual Studio Code menyertakan beberapa ekstensi untuk FTP, memungkinkan perangkat lunak digunakan sebagai alternatif gratis untuk pengembangan web. Kode dapat disinkronkan antara editor dan server, tanpa mengunduh perangkat lunak tambahan.

Visual Studio Code memungkinkan pengguna untuk mengatur halaman kode di mana dokumen aktif disimpan, karakter baris baru, dan bahasa pemrograman dokumen aktif. Ini memungkinkannya untuk digunakan di platform apa pun, di lokasi mana pun, dan untuk bahasa pemrograman apa pun.

Visual Studio Code menyertakan dukungan dasar untuk sebagian besar bahasa pemrograman umum. Dukungan dasar ini mencakup penyorotan sintaks, pencocokan braket, pelipatan kode , dan cuplikan yang dapat dikonfigurasi. Visual Studio Code juga dikirimkan dengan IntelliSense untuk JavaScript, TypeScript, JSON, CSS, dan HTML, serta dukungan debugging untuk Node.js. Dukungan untuk bahasa tambahan dapat disediakan oleh ekstensi yang tersedia secara gratis di VS Code Marketplace. (internet)

# BAB III SARANA DAN PRASARANA

* 1. Sarana

Untuk melaksanakan sistem yang terkomputerisasi diperlukan suatu sarana pendukung agar dapat berjalan sebagaimana mestinya. Sarana dan Prasarana pendukung tersebut adalah perangkat lunak dan perangkat keras yang harus dapat menunjukan kerja yang baik dan sesuai dengan yang diharapkan. Berikut ini adalah perangkat – perangkat yang digunakan untuk membuat e-Monitoring Program Pengelolaan dan Pengembangan Sistem Penyediaan Air Minum dan Sistem Air Limbah Bidang Air Bersih dan Sanitasi ini adalah :

* + 1. Perangkat Keras

Adapun perangkat keras yang akan digunakan untuk pembuatan sistem ini adalah perangkat komputer atau laptop dengan spesifikasi yang mampu dan sesuai dengan kebutuhan,sehingga perogram atau aplikasi yang akan digunakan lancar tanpa *hang* atau gangguan.

1. Komputer

Komputer adalah perangkat elektronik yang memanipulasi informasi atau data. Komputer mampu menyimpan, mengambil, dan mengolah data. Komputer yang digunakan oleh Bidang Air Bersih dan Sanitasi memiliki spesifikasi sebagai berikut :

* ***Processor*:** Intel i3
* ***RAM*:** 4 GB
* ***Harddisk*:** Seagete 1 TB
* ***VGA*:** NVIDIA GEFORCE
* [*Monitor*](http://www.dimensidata.com/category/monitor/?orderby=15)**:** HP 16 Inch

b. Printer

adalah alat yang dapat kita gunakan untuk mencetak documen dari komputer ke media kertas atau yang sejenis. Printer yang digunakan oleh staff adalah : *Printer epson L310.*

* + 1. Perangkat Lunak Pendukung

Perangkat lunak pendukung merupakan perangkat lunak pembangun untuk pendokumentasian maupun pembuatan system yang mendukung pembuatan laporan kerja praktik ini, diantara perangkat lunak pendukung yang digunakan adalah :

1. Microsoft Office 2019.
2. Node.js, Vue.js, MariaDB, Strapi, Heroku, Git.
3. Visual Studio Code.
   1. **Prasarana**

Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan mempunyai 5 ruang kerja utama, yaitu; 1 ruang kepala bidang, 3 ruang kepala seksi dan 1 ruang kerja staf. dalam mendukung tupoksi Bidang Air Bersih dan Sanitasi.

# BAB IV ANALISIS DAN PEMBAHASAN

## Analisis Sistem yang Sedang Berjalan

Analisis sistem sedang berjalan didefinisikan sebagai penguraian dari suatu sistem yang utuh menjadi komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan yang terjadi. Sistem pengumpulan data yang sedang berjalan saat ini yaitu pengumpulan data-data kegiatan masih dilakukan dengan menggunakan aplikasi dari Microsoft Office maupun ditulis pada buku jurnal. Pada proses pengumpulan data ini, semua data yang telah terkumpul tidak terintegrasi pada suatu sistem, hal ini menjadikan masalah apabila ingin melakukan pencarian data pada parameter tertentu. Pencarian harus dilakukan secara manual dengan membuka file-file Microsoft Office terlebih dahulu.

## Analisis Sistem yang Akan Diusulkan

1. Analisis Masalah

Pada tahap ini dilakukan analisis permasalahan untuk mendapatkan permasalahan apa saja yang menjadi acuan dalam pembuatan sistem ini. Dari hasil pengamatan yang dilakukan, penulis menampilkan data permasalahan sebagai berikut:

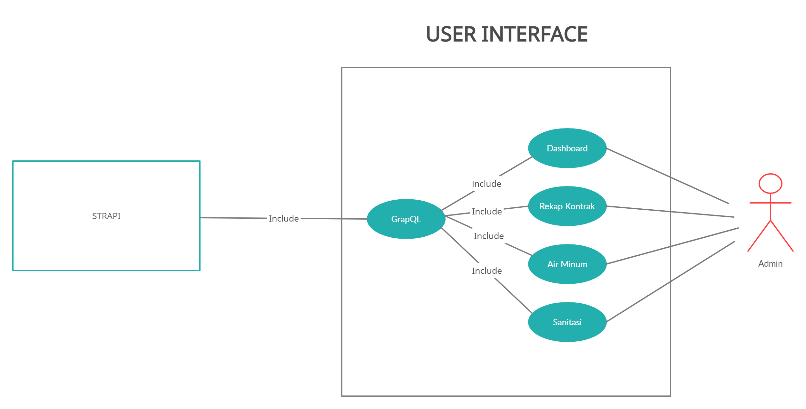
1. Belum adanya integerasi data dari data-data yang sudah tersedia di Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur.
2. Pengarsipan data yang masih menggunakan *paper-based* sehingga menjadi kesulitan apabila akan mencari data-data dari tahun anggaran sebelumnya.
3. Belum ada database untuk inventarisasi pengelola untuk hasil dari kegiatan-kegiatan yang telah dilaksanakan oleh Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur.
4. Analisis Kebutuhan Sistem

Pelbagai kebutuhan dalam membangun system ini adalah sebagai berikut:

1. Analisis Kebutuhan Antarmuka (*interface*)
2. Sistem dapat menampilkan form isi untuk data-data kegiatan;
3. Sistem akan menampilkan tabel data-data kegiatan untuk melakukan monitoring terhadap kegiatan yang akan, sedang, maupun sudah dilaksanakan;
4. Sistem akan dibangun dengan UI/UX yang familiar dan mudah digunakan oleh pengguna.
5. Kebutuhan Data
6. Data Kontrak;
7. Data SPAM terbangun;
8. Data MCK/IPAL terbangun;
9. Kebutuhan Fungsional
10. Monitoring kegiatan/pekerjaan;
11. Analisis kegiatan/pekerjaan;

## Perancangan Sistem

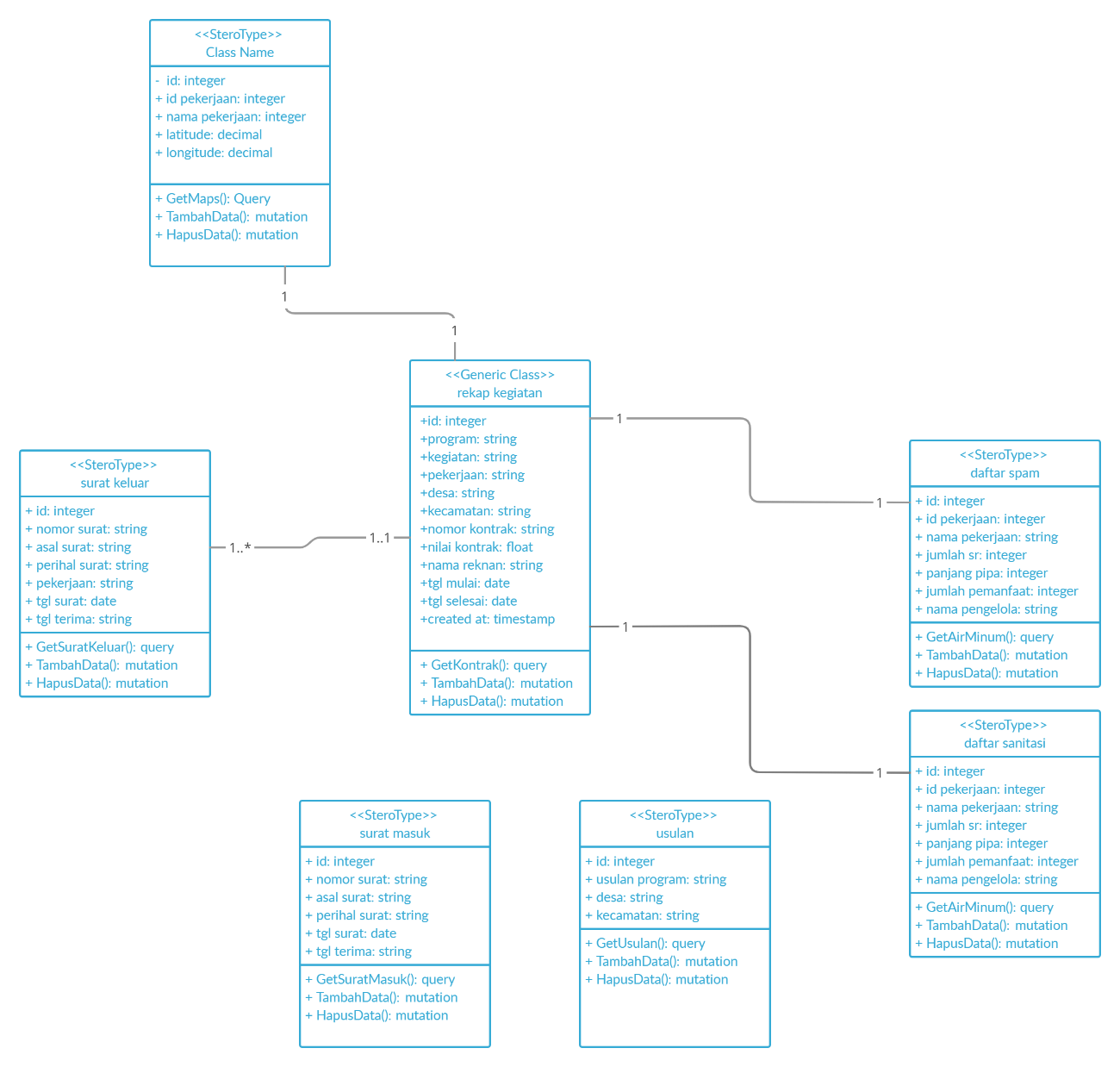
1. Use Case Diagram

Use case diagram merupakan gambaran skenario dari interaksi antara pengguna dengan sistem. Use case diagram menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.

Referensi: https://lonsdalesystems.com/post/rebooting-the-use-case-diagram

Gambar 1: Use Case Diagram

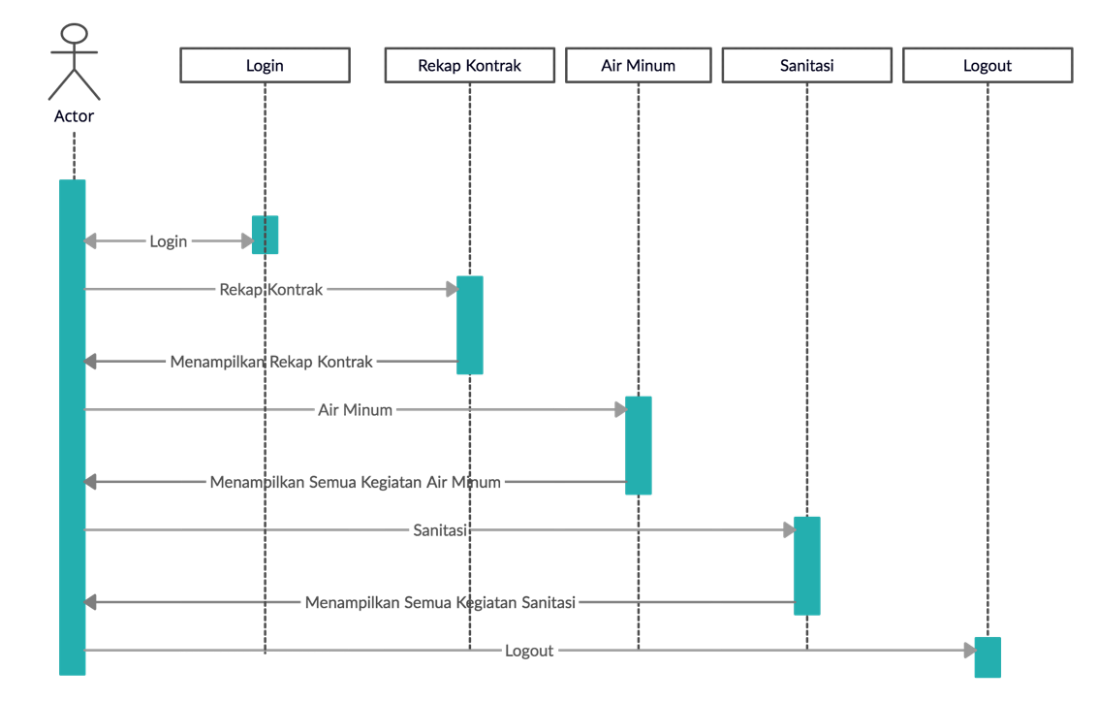
1. Class Diagram

Class Diagram merupakan diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Referensi: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

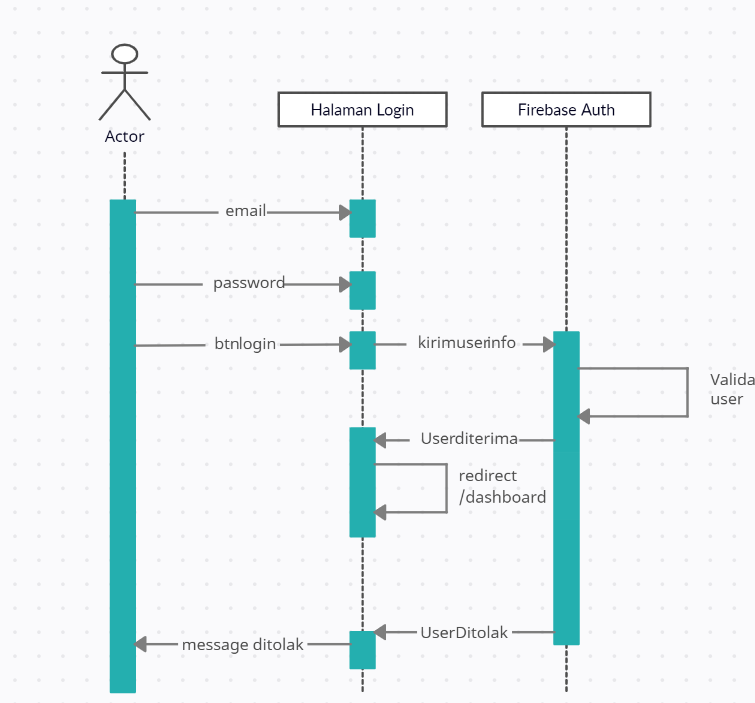
Gambar 2: Class Diagram

1. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa message terhadap waktu. Pembuatan sequence diagram bertujuan agar perancangan aplikasi lebih mudah dan terarah. Interaksi-interaksi yang terjadi dalam aplikasi ini adalah:

Referensi https://www.researchgate.net/figure/A-sequence-diagram-showing-the-realization-of-modifying-a-record-use-case\_fig3\_261242686

Gambar 3: Sequence Diagram

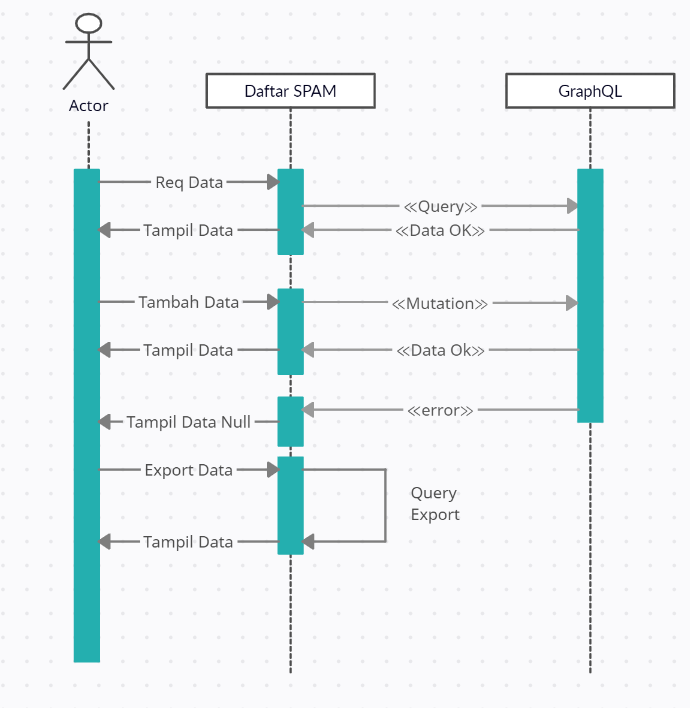


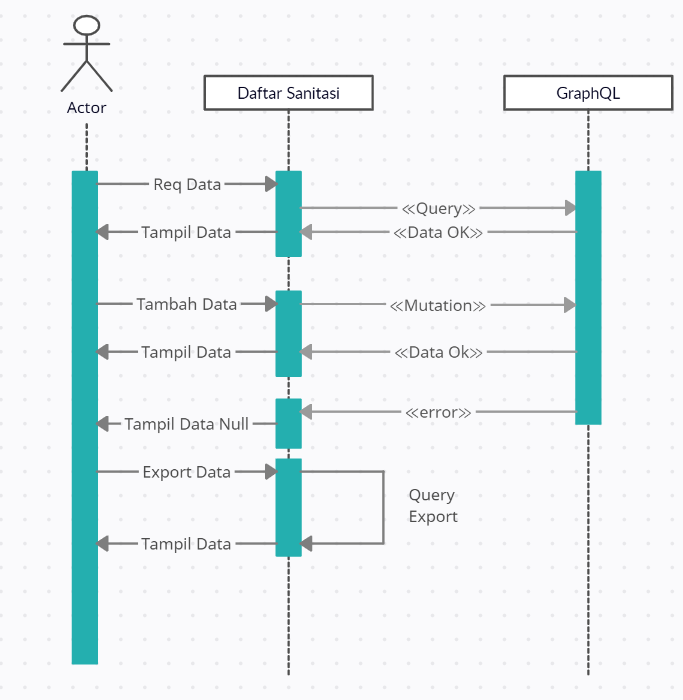
Referensi https://www.researchgate.net/figure/A-sequence-diagram-showing-the-realization-of-modifying-a-record-use-case\_fig3\_261242686

Gambar 4: Sequence Diagram Login

Referensi https://www.researchgate.net/figure/A-sequence-diagram-showing-the-realization-of-modifying-a-record-use-case\_fig3\_261242686

Gambar 5: Sequence Diagram Daftar SPAM

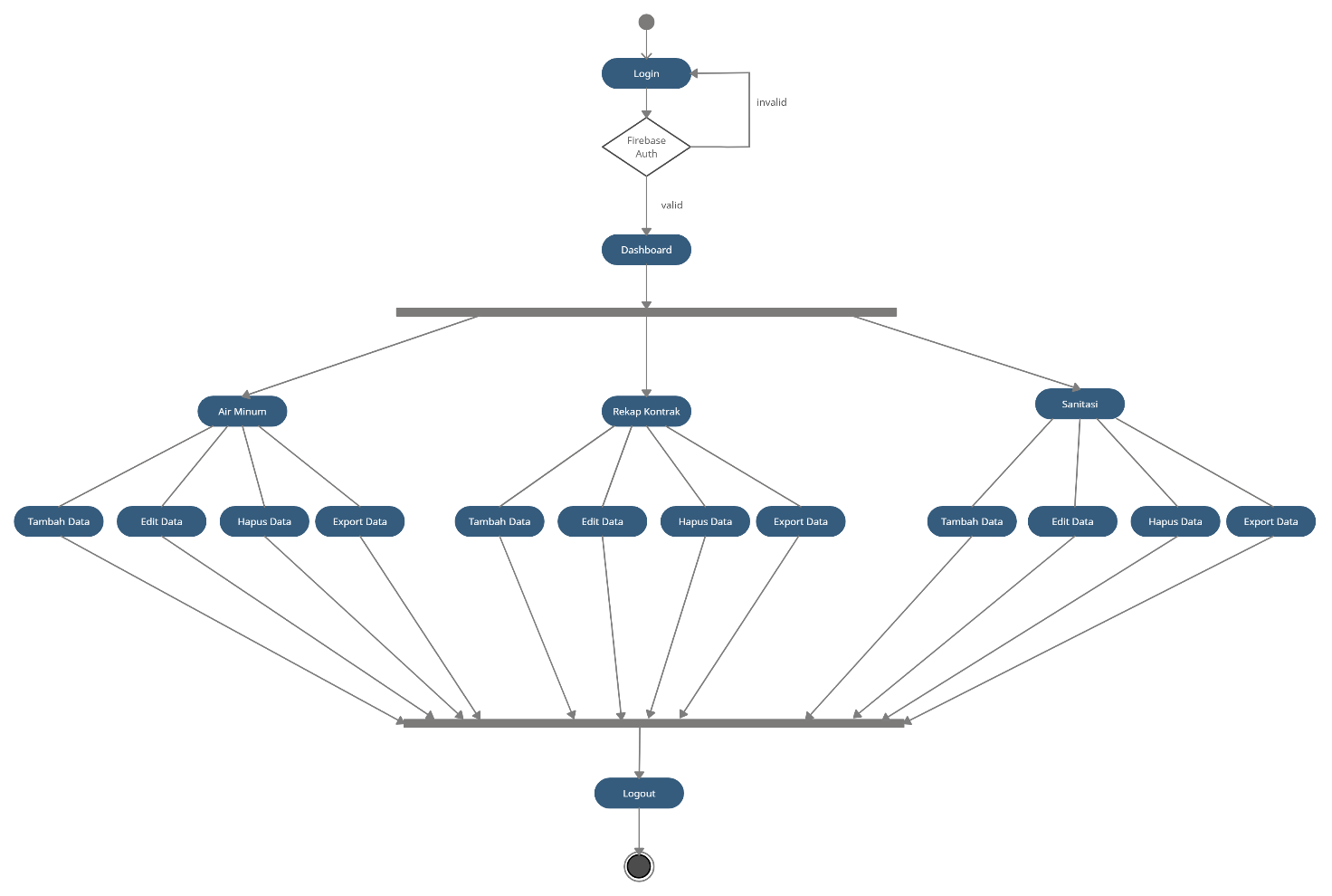




Referensi https://www.researchgate.net/figure/A-sequence-diagram-showing-the-realization-of-modifying-a-record-use-case\_fig3\_261242686

Gambar 6: Sequence Diagram Daftar Sanitasi

1. Activity Diagram User

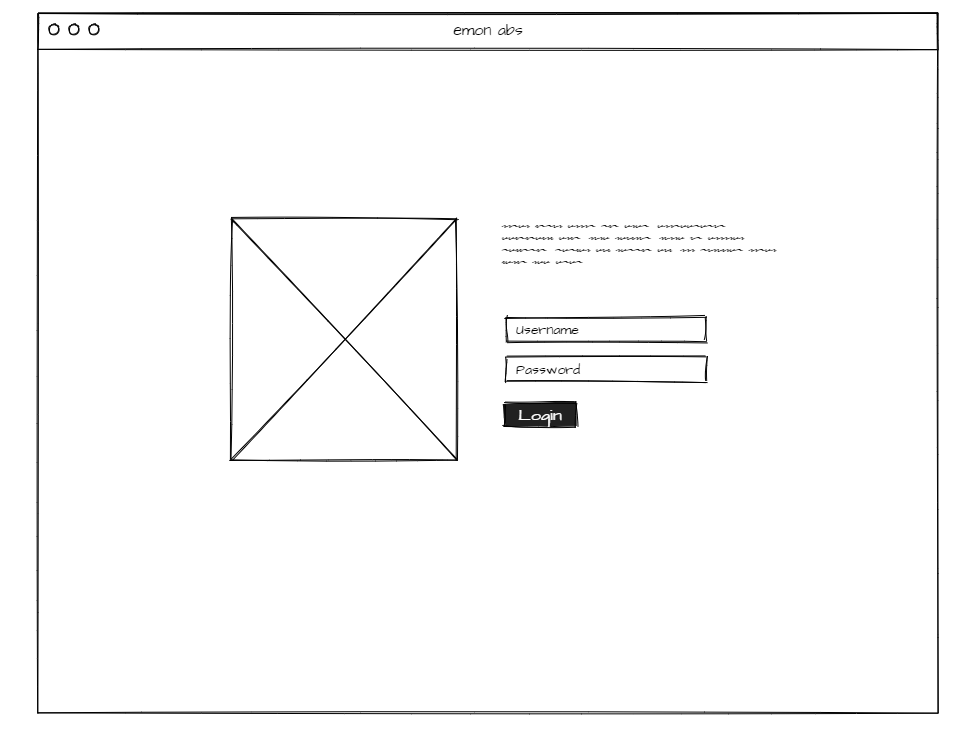
Activity Diagram adalah representasi grafis dari seluruh tahapan alur kerja yang mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem. Adapun activity diagram dari navigasi sistem ini adalah sebagai berikut:

Referensi: https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/

Gambar 7: Activity Diagram

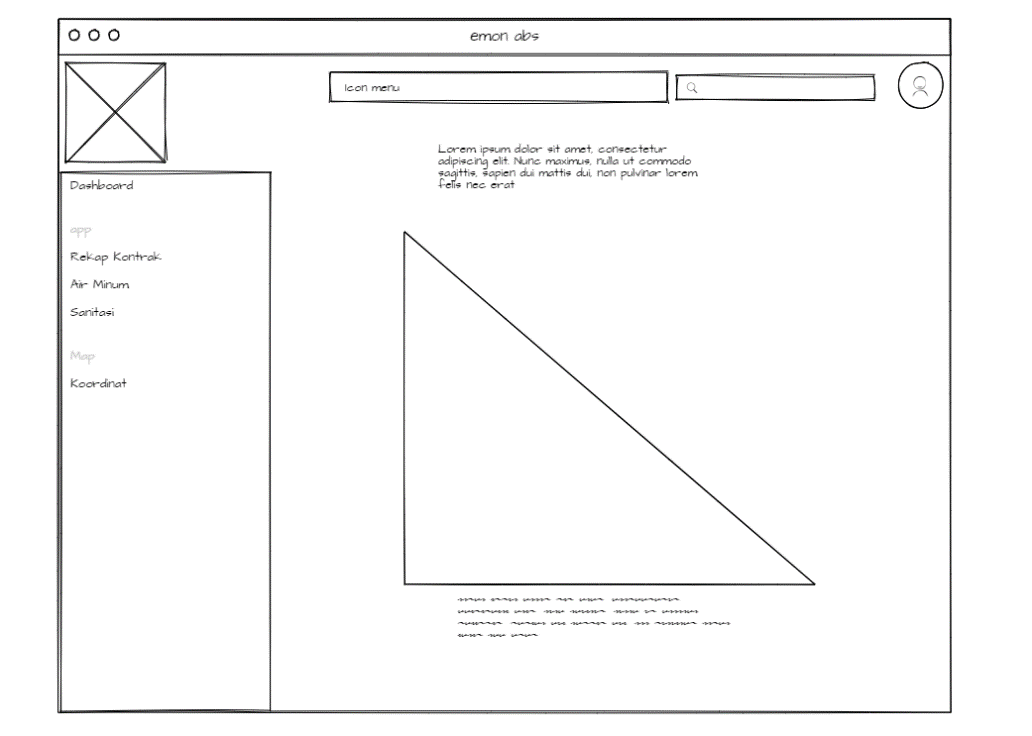
1. Perancangan Antarmuka

Perancangan antarmuka merupakan aspek penting dalam perancangan aplikasi, karena berhubungan dengan tampilan dan interaksi yang memudahkan user dalam menggunakannya. Adapun rancangan antarmuka pada sistem ini sebagai berikut:

1. Halaman Login

Gambar 8: Rancangan Halaman Login

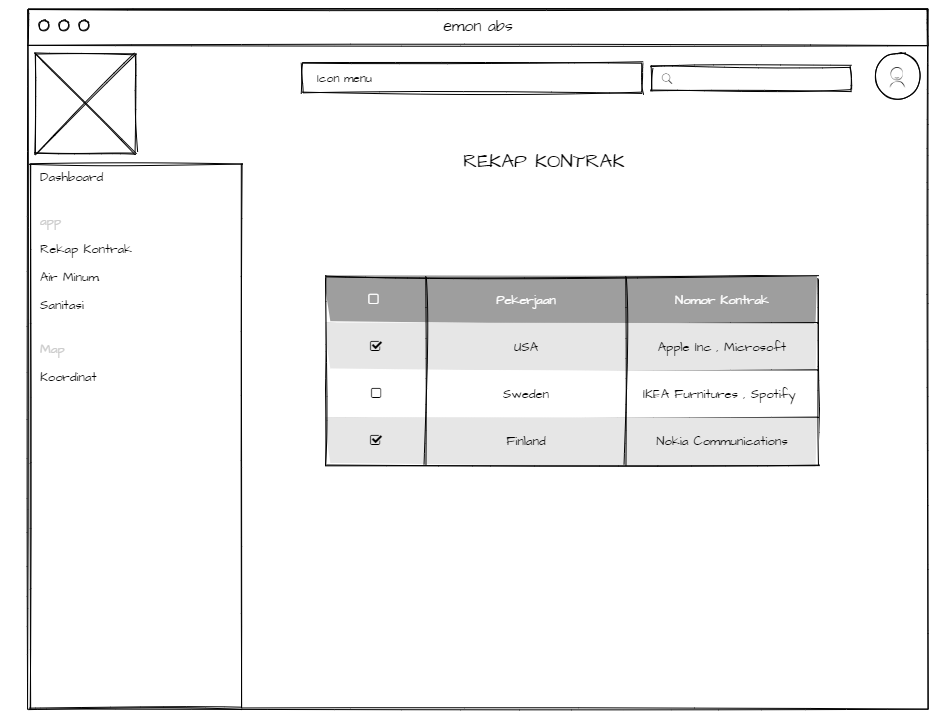
Halaman Login berfungsi untuk validasi dan autentikasi pengguna untuk mengakses halaman-halaman lainnya yang terdapat pada aplikasi dengan menggunakan email dan password. Pada halaman ini, data pengguna disimpan dan autentikasi menggunakan aplikasi pihak ketiga, yaitu Firebase.

1. Halaman Dashboard

Gambar 9: Rancangan Halaman Dashboard

Halaman Dashboard berisi informasi chart jumlah Output dari pekerjaan-pekerjaan dalam kegiatan Sistem Penyediaan Air Minum dan Sistem Air Limbah.

1. Halaman Rekap Kontrak

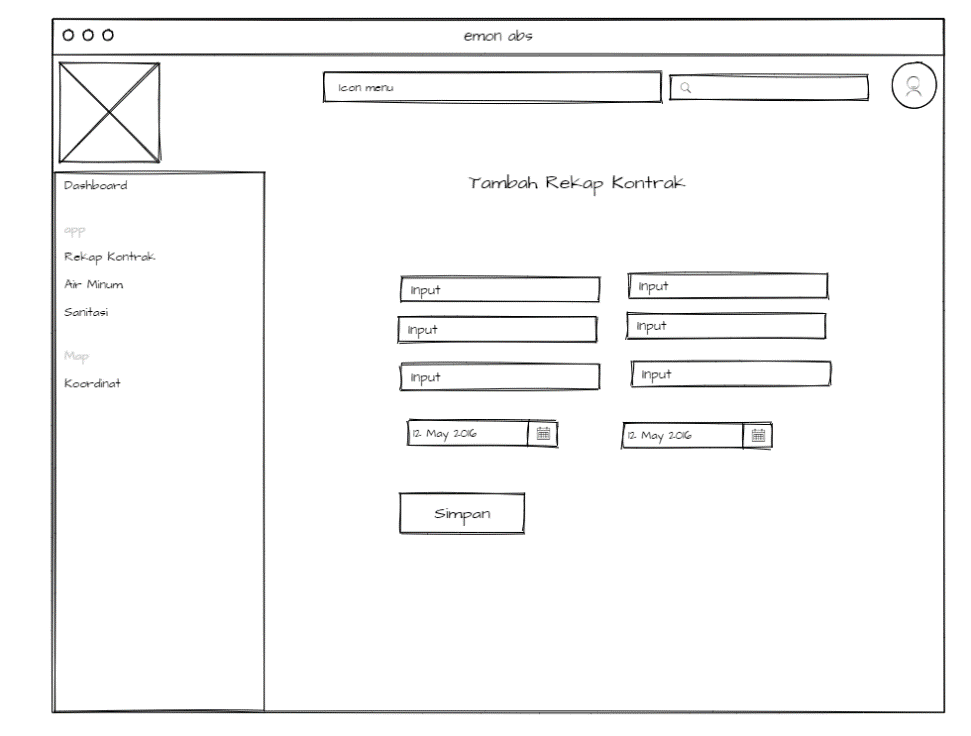


Gambar 10: Rancangan Halaman Rekap Kontrak

Halaman Rekap Kontrak menampilkan data-data kontrak pekerjaan yang sedang atau telah dilaksanakan dalam Tahun Anggaran berjalan.

1. Halaman Tambah Rekap Kontrak

Gambar 11: Rancangan Halaman Tambah Rekap Kontrak



Halaman Tambah Data Kontrak berisi form input untuk menambahkan data kontrak pekerjaan pada kegiatan yang sedang atau sudah dilaksanakan dalam Tahun Anggaran berjalan.

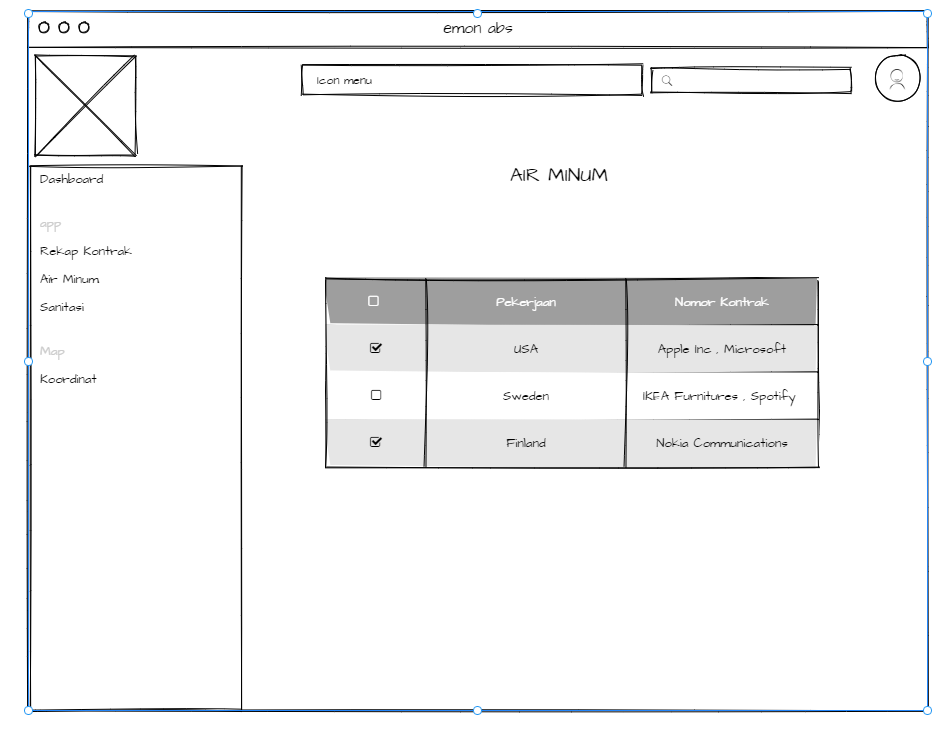
1. Halaman Ubah Rekap Kontrak



Gambar 12: Rancangan halaman Ubah Rekap Kontrak

Halaman Ubah Data Kontrak berisi form input untuk mengubah data kontrak pekerjaan pada kegiatan yang sedang atau sudah dilaksanakan dalam Tahun Anggaran berjalan.

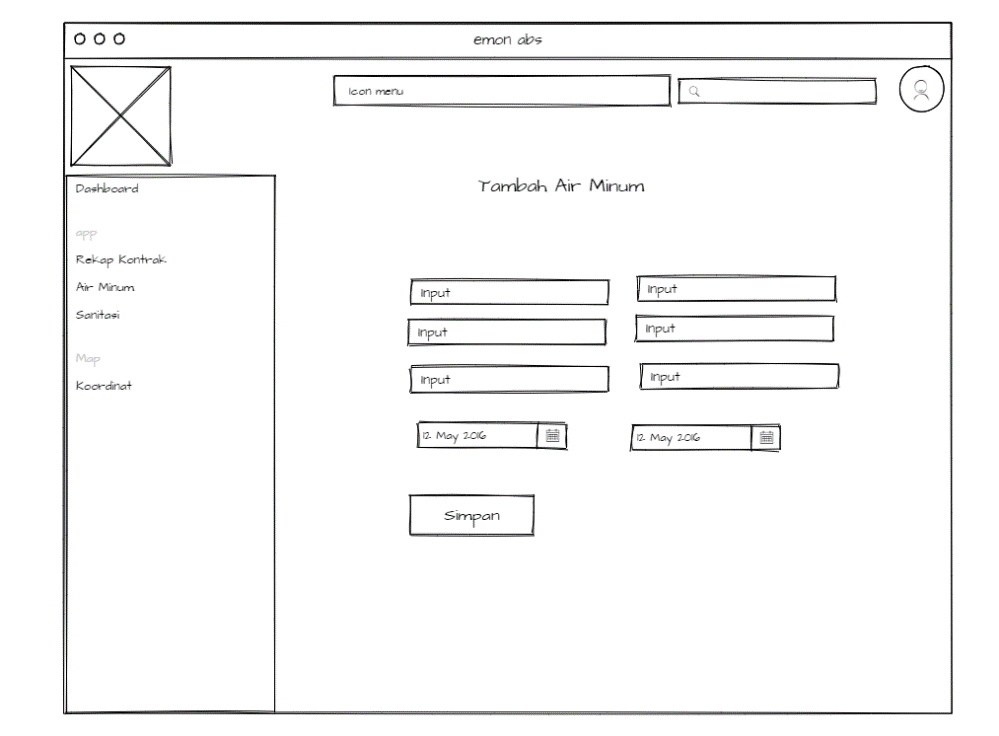
1. Halaman Daftar SPAM



Gambar 13: Rancangan Halaman Air Minum

Halaman Daftar SPAM menampilkan data pekerjaan pada kegiatan Pembangunan SPAM Perdesaan.

1. Halaman Tambah Daftar SPAM

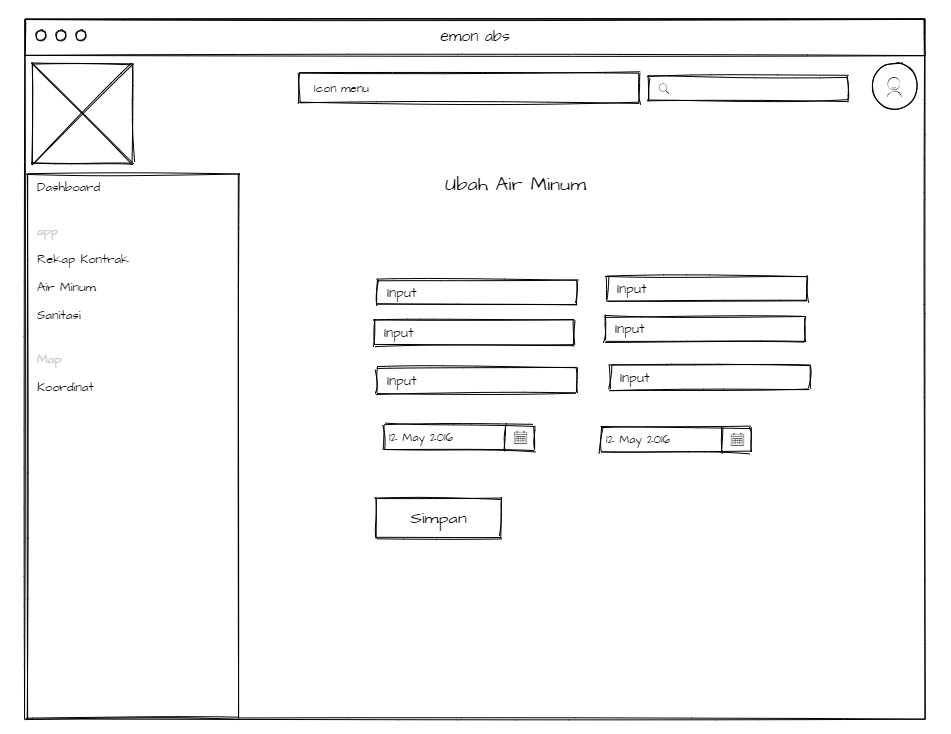


Gambar 14: Rancangan Halaman Tambah Air Minum

Halaman Tambah Daftar SPAM berisi form input untuk menambahkan data kontrak pekerjaan pada kegiatan Pembangunan SPAM Perdesaan yang sedang atau sudah dilaksanakan dalam Tahun Anggaran berjalan.

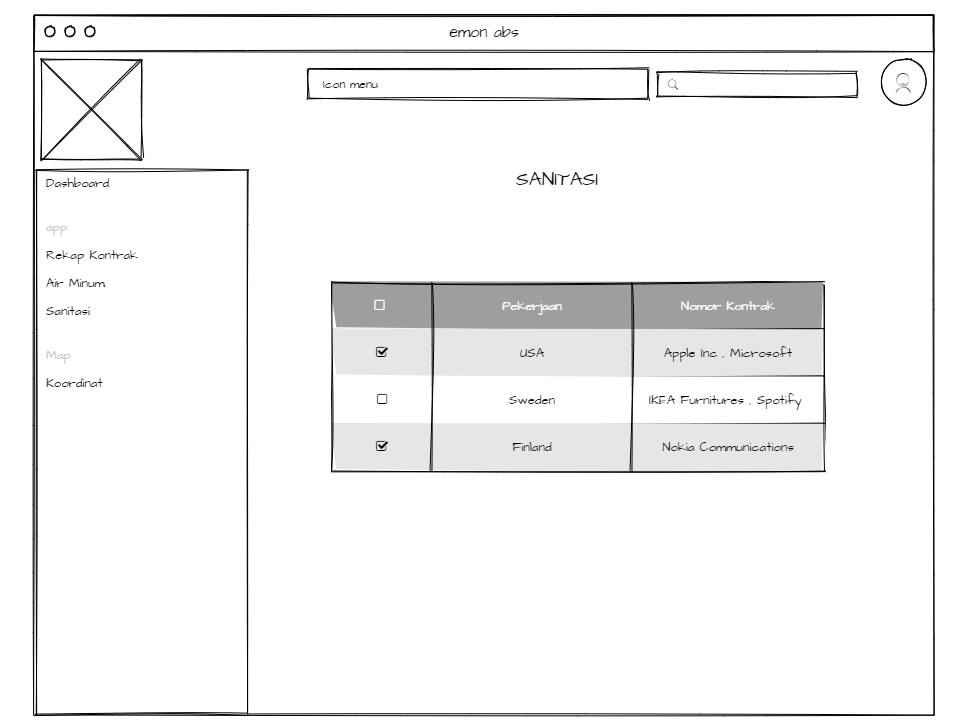
1. Halaman Daftar SPAM

Gambar 15: Rancangan Halaman Ubah Air Minum



Halaman Tambah Daftar SPAM berisi form input untuk mengubah data kontrak pekerjaan pada kegiatan Pembangunan SPAM Perdesaan yang sedang atau sudah dilaksanakan dalam Tahun Anggaran berjalan.

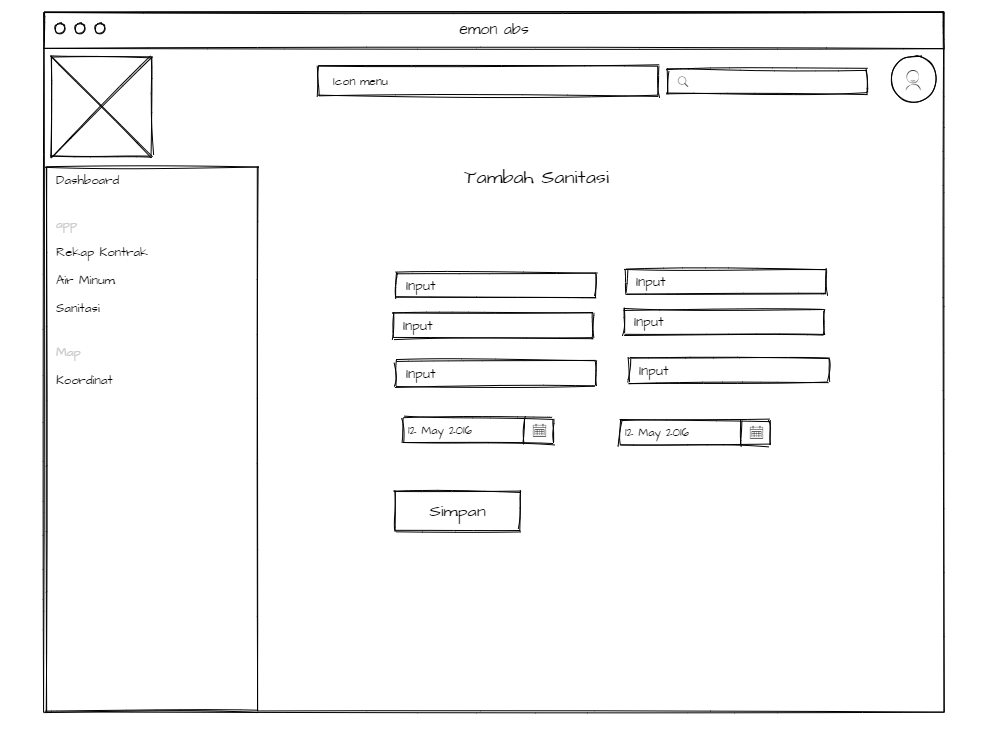
1. Halaman Sanitasi



Gambar 16: Rancangan Halaman Sanitasi

Halaman Tambah Daftar Sanitasi menampilkan data pekerjaan pada kegiatan Pembangunan Sistem Air Limbah yang sedang atau sudah dilaksanakan dalam Tahun Anggaran berjalan.

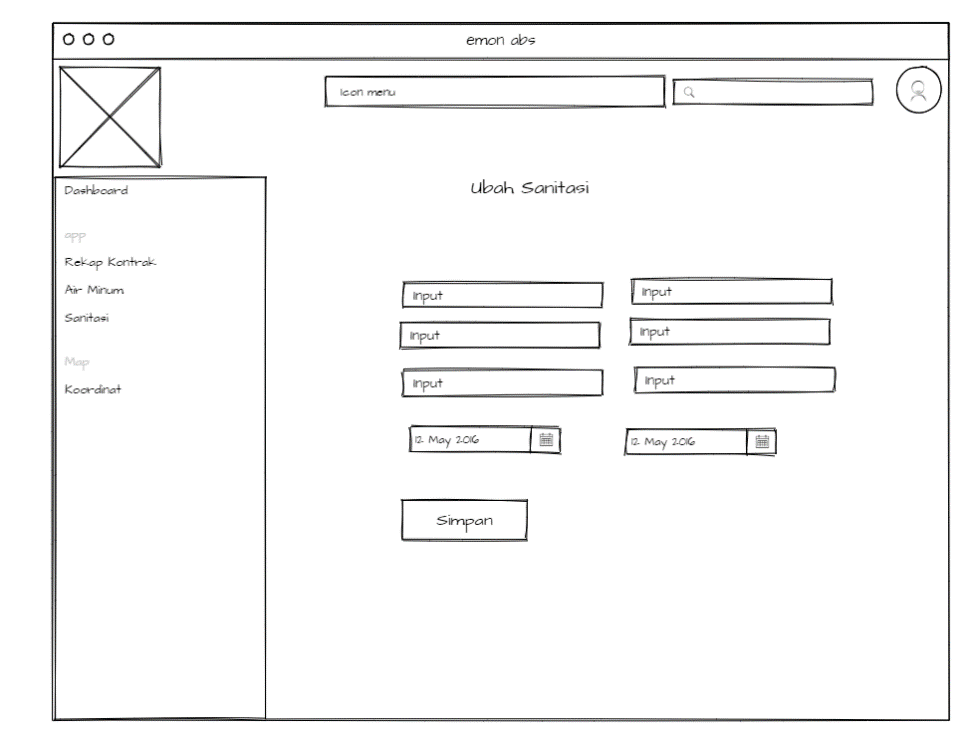
1. Halaman Tambah Sanitasi



Gambar 17: Rancangan Halaman Tambah Sanitasi

Halaman Tambah Daftar Sanitasi berisi form input untuk menambahkan data kontrak pekerjaan pada kegiatan Pembangunan Sistem Air Limbah yang sedang atau sudah dilaksanakan dalam Tahun Anggaran berjalan.

1. Halaman Ubah sanitasi



Gambar 18: Rancangan Halaman Ubah Sanitasi

Halaman Ubah Daftar Sanitasi berisi form input untuk mengubah data kontrak pekerjaan pada kegiatan Pembangunan Sistem Air Limbah yang sedang atau sudah dilaksanakan dalam Tahun Anggaran berjalan.

## Perancangan Basis Data

Tabel 1: Rekap Kontrak

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type Data | Length | Keterangan |
| id | Int | 5 | Primary Key |
| program | Char | 32 |  |
| kegiatan | Char | 32 |  |
| pekerjaan | Char | 32 |  |
| desa | Char | 32 |  |
| kecamatan | Char | 32 |  |
| nomor\_kontrak | Char | 32 |  |
| nilai\_kontrak | Float | 64 |  |
| nama\_rekanan | Char | 32 |  |
| tgl\_mulai | Date |  |  |
| tgl\_selesai | Date |  |  |

Tabel 2: Koordinat

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type Data | Length | Keterangan |
| id | Int | 10 | \*Primary Key |
| id\_pekerjaan | Int | 10 |  |
| lat | Float | 10 |  |
| long\_ | Float | 10 |  |

Tabel 3: MCK/Sanitasi

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type Data | Length | Keterangan |
| id | Int | 10 | \*Primary Key |
| id\_pekerjaan | Int | 10 |  |
| nama\_pengelola | Text | 50 |  |
| jumlah\_sr | Int | 10 |  |
| jumlah\_penduduk | Int | 10 |  |
| penduduk\_terlayani | Int | 10 |  |

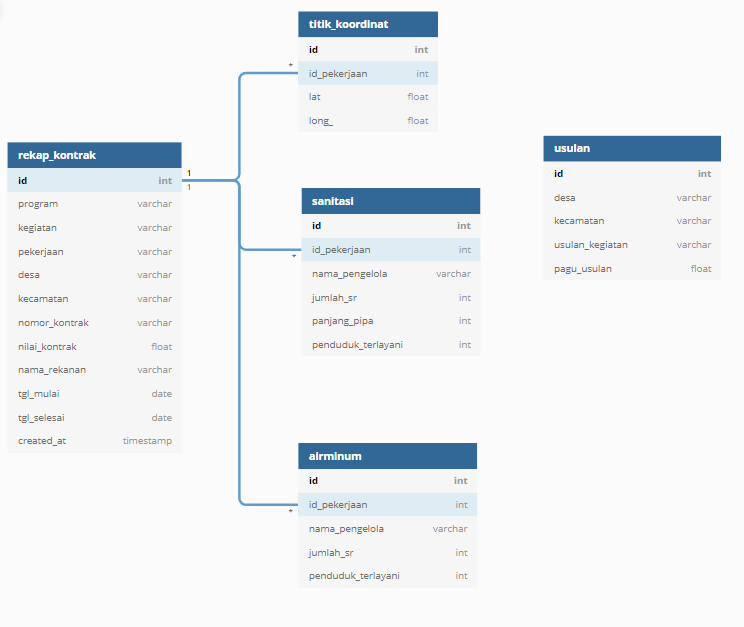
Tabel 4: Air Minum

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type Data | Length | Keterangan |
| id | Int | 10 | \*Primary Key |
| id\_pekerjaan | Int | 10 |  |
| nama\_pengelola | Text | 50 |  |
| jumlah\_sr | Int | 10 |  |
| penduduk\_terlayani | Int | 10 |  |

Tabel 5: Usulan

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type Data | Length | Keterangan |
| id | Int | 10 | \*Primary Key |
| desa | Char | 32 |  |
| kecamatan | Char | 32 |  |
| usulan\_kegiatan | Char | 32 |  |
| pagu\_usulan | Float | 10 |  |

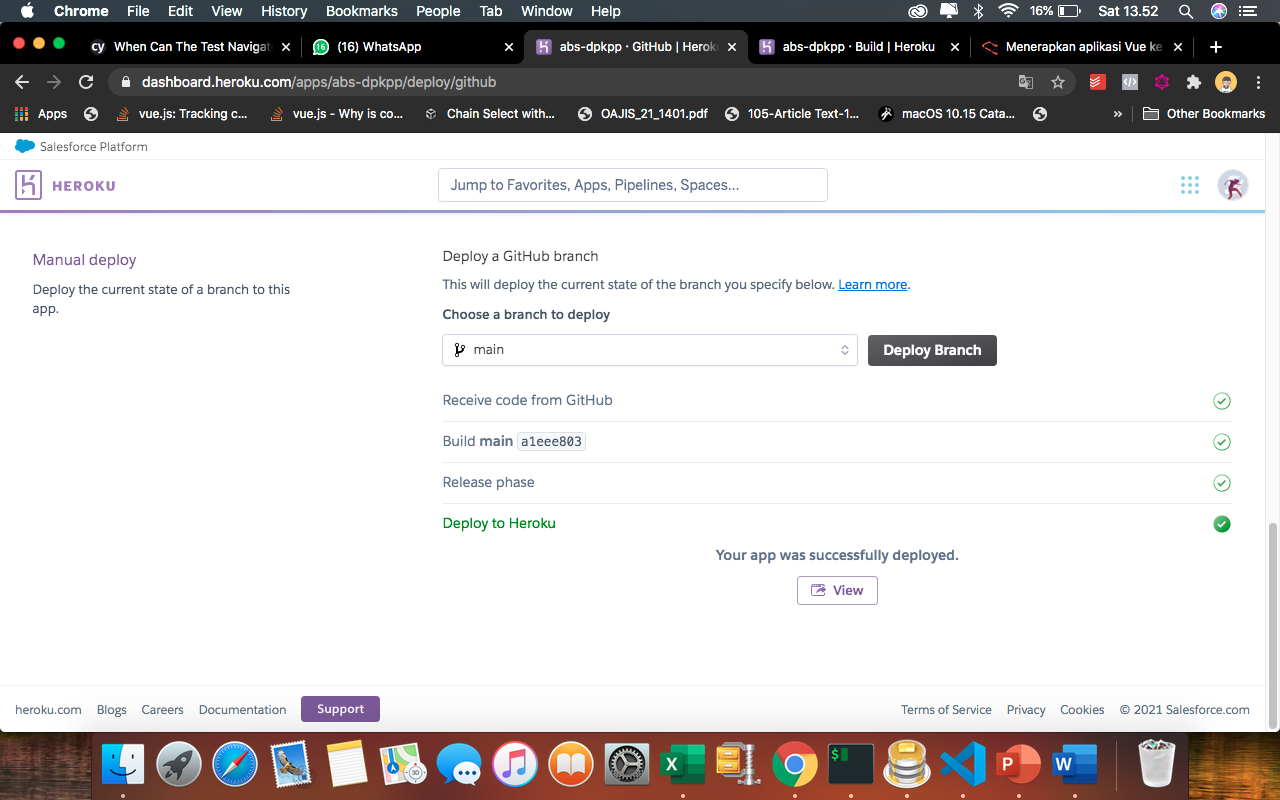
### Relasi Database



# BAB V IMPLEMENTASI DAN PENGUJIAN

## Implementasi

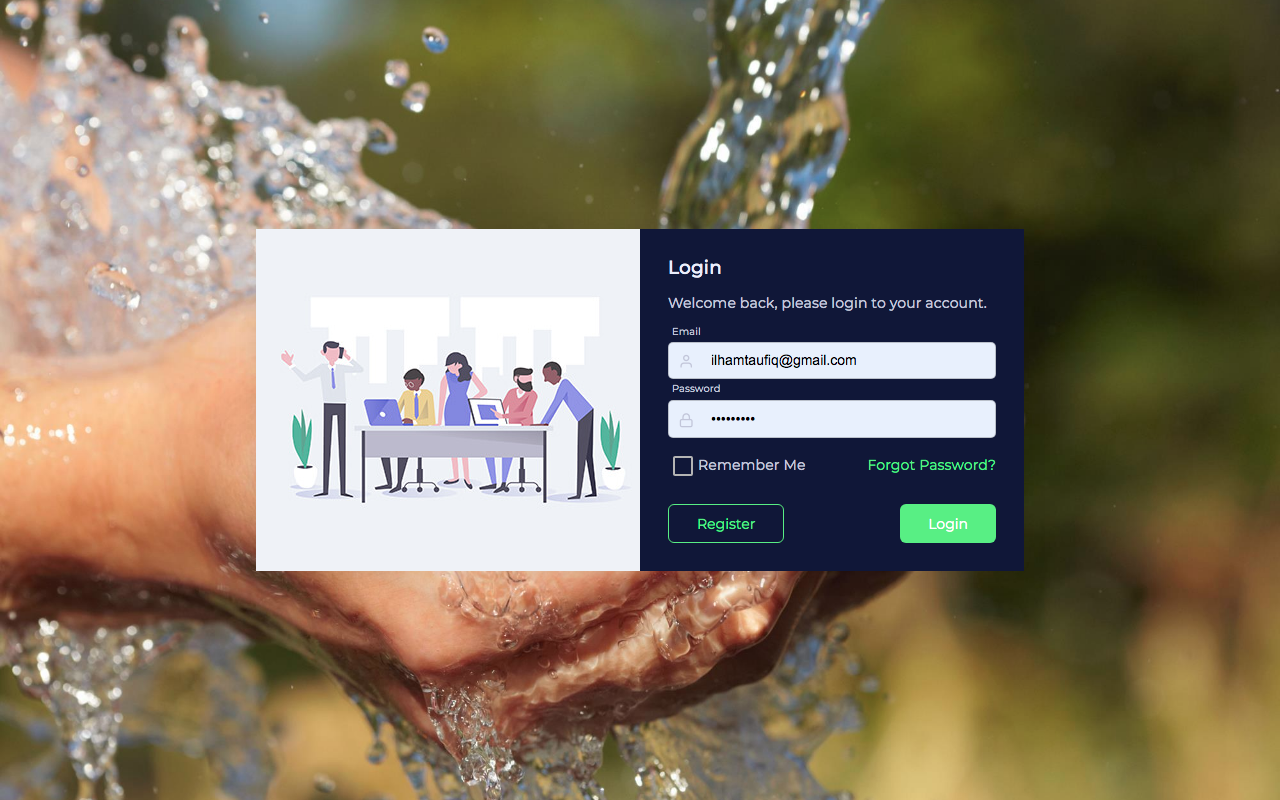
1. Deployment

Deployment dengan mode production dari Branch Main Repository Github ke Heroku. Untuk dapat deploy ke Heroku terdapat beberapa konfigurasi terlebih dahulu, diantaranya;

* 1. Install Heroku cli
  2. Menambahkan variable NODE\_ENV=production pada Configuration Variables di Heroku.
  3. Menambahkan Procfile pada root folder project untuk informasi aplikasi.

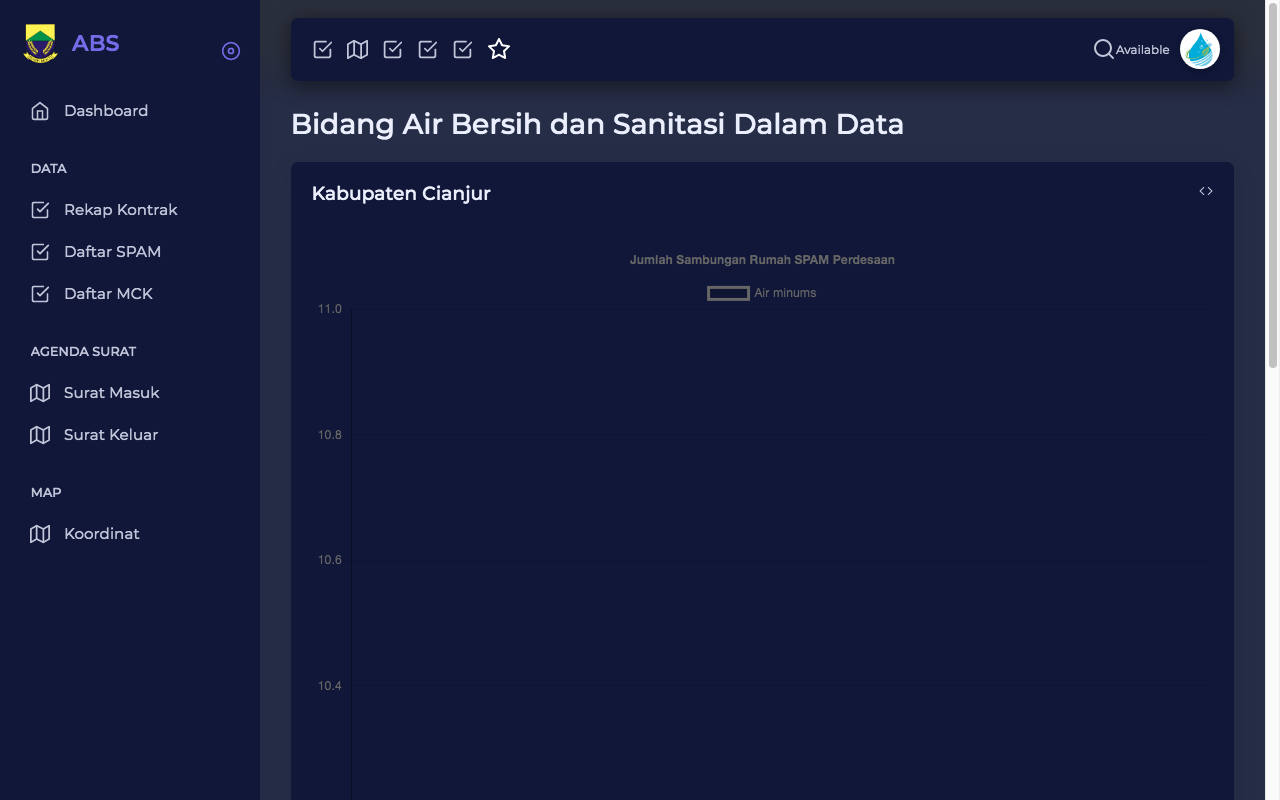
Apabila dalam log terdapat informasi build successful dengan centang hijau, aplikasi dengan status production siap digunakan dengan diuji membuka halaman URL aplikasi pada *browser*.

1. Halaman Login

 Halaman Login berfungsi untuk validasi dan autentikasi pengguna untuk mengakses halaman-halaman lainnya yang terdapat pada aplikasi dengan menggunakan email dan password. Pada halaman ini, data pengguna disimpan dan autentikasi menggunakan aplikasi pihak ketiga, yaitu Firebase.

Gambar 19: Halaman Login

1. Dashboard

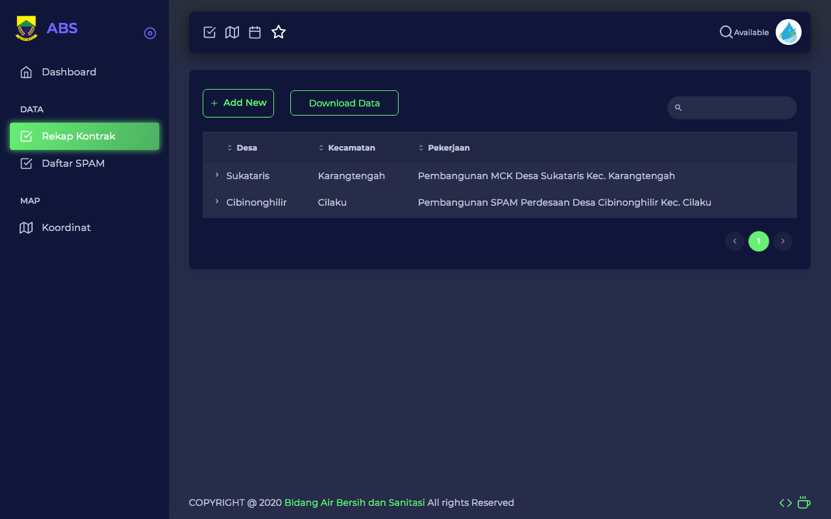
 Halaman Dashboard berisi informasi chart jumlah Output dari pekerjaan-pekerjaan dalam kegiatan Sistem Penyediaan Air Minum dan Sistem Air Limbah.

Gambar 20: Halaman Dashboard

Di dalam halaman dashboard terdapat chart jumlah Sambungan Rumah yang telah terbangun pada pekerjaan Pembangunan SPAM Perdesaan.

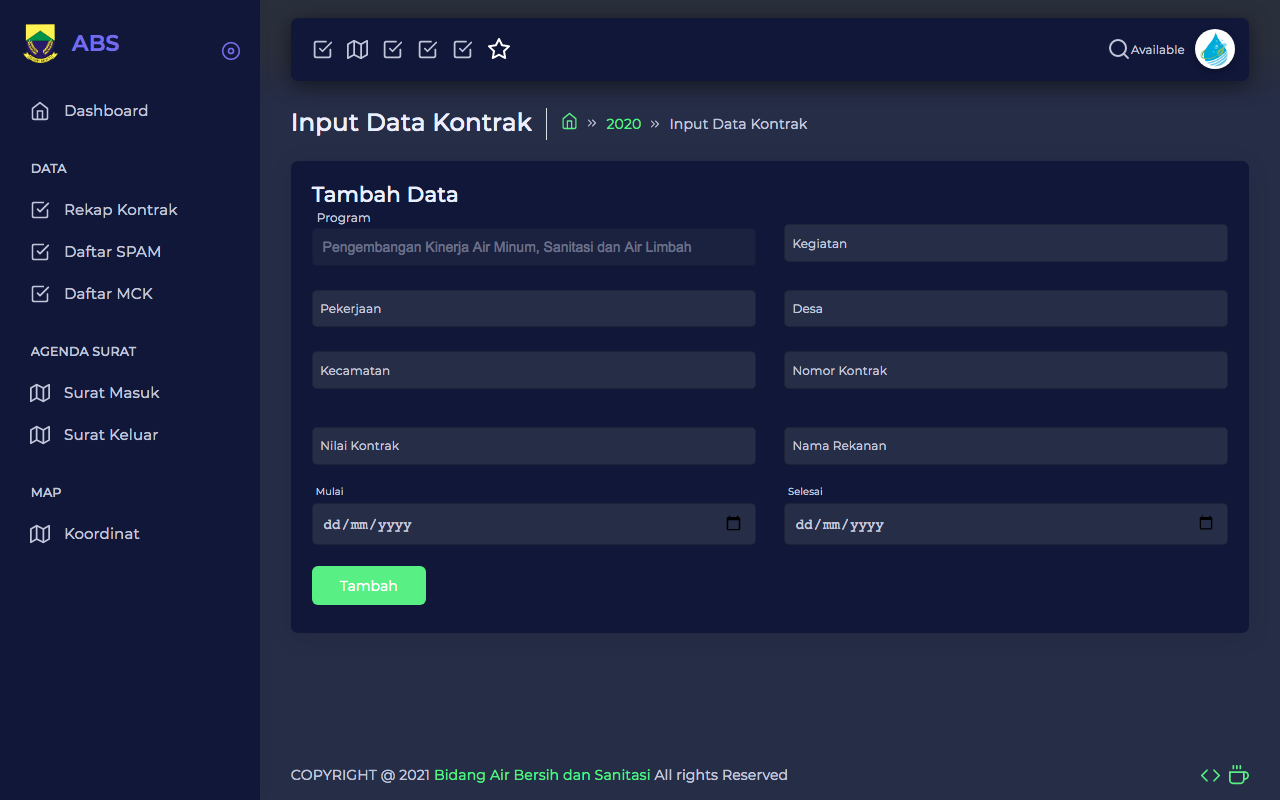
1. Rekap Kontrak

Halaman Rekap Kontrak memuat data kontrak dari semua pekerjaan yang sedang atau telah dilaksanakan dalam Tahun Anggaran berjalan.

Button tambah dalam halaman Rekap Kontrak berfungsi untuk menambahkan data kontrak, Button Download data berfungsi untuk mengunduh file data Rekap Kontrak.

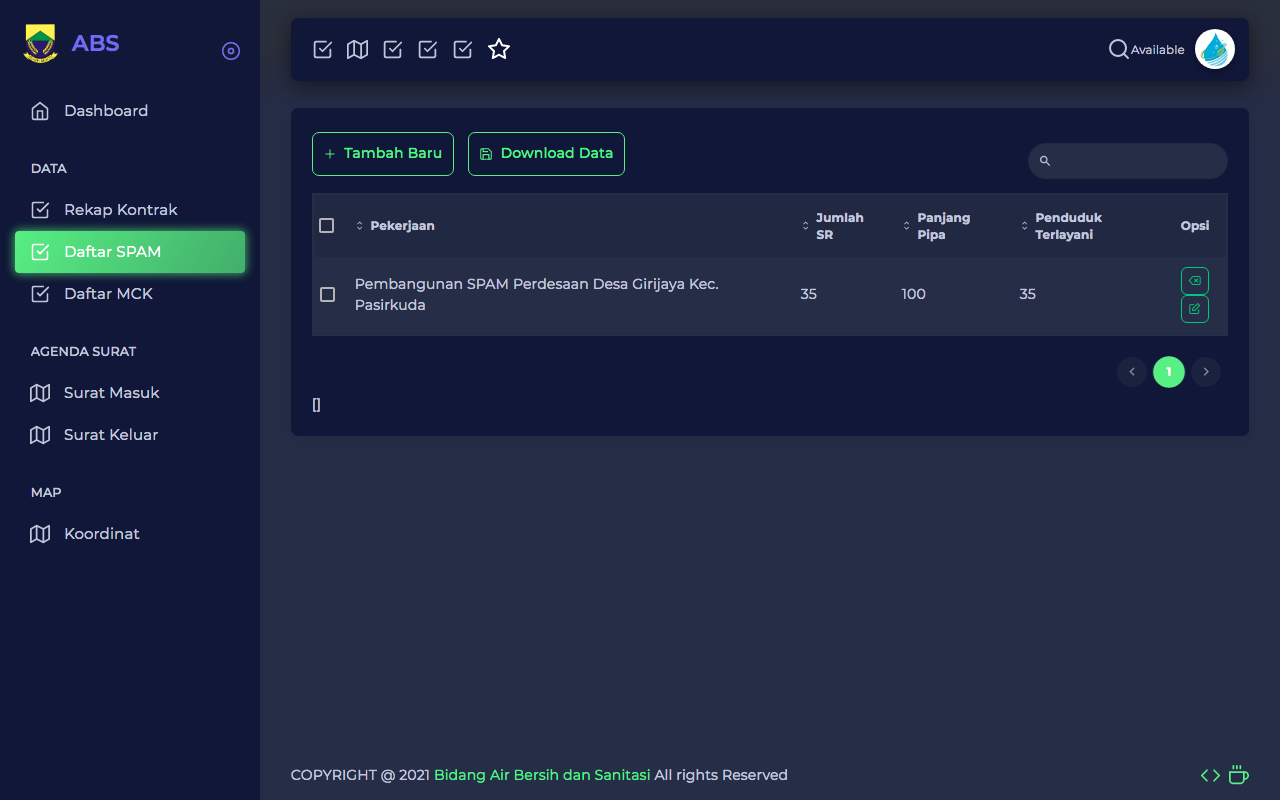
Gambar 21: Halaman Rekap Kontrak

1. Input Data Kontrak

Memasukkan data kontrak pekerjaan yang sedang atau sudah dilaksanakan pada Tahun Anggaran berjalan.

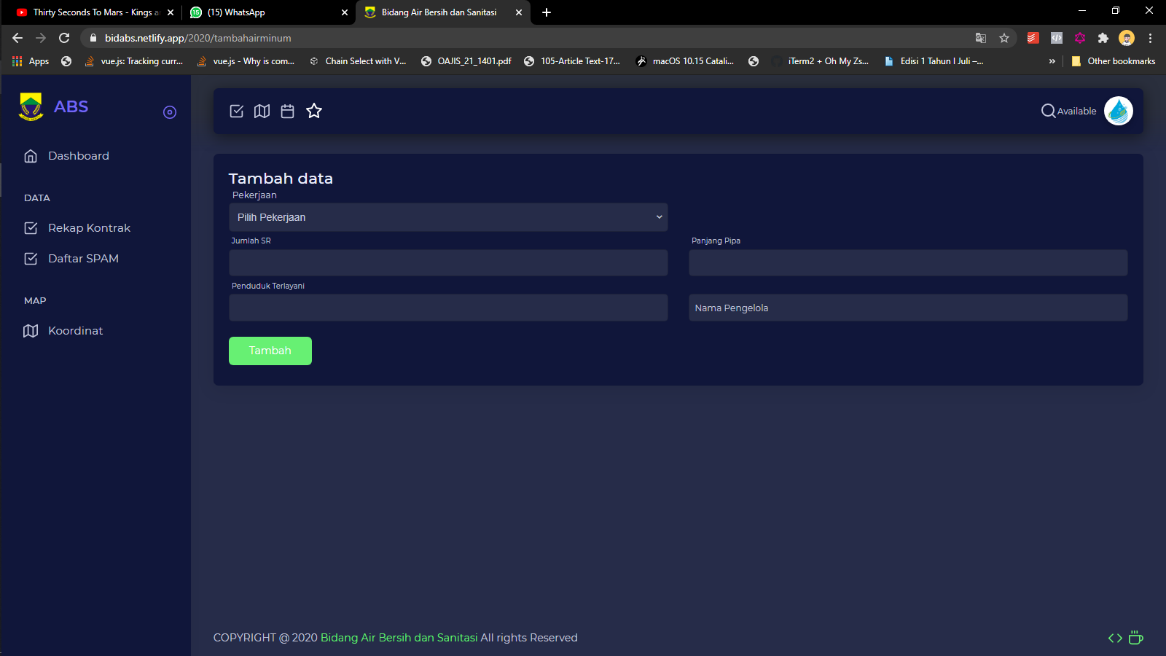
Gambar 22: Halaman Input Data Kontrak

1. Pembangunan SPAM

Menampilkan semua pekerjaan dengan kegiatan Pembangunan SPAM Perdesaan.

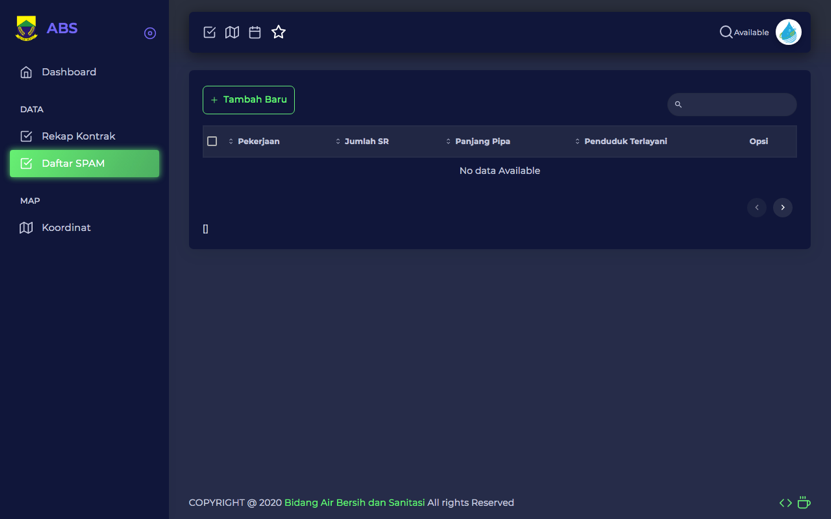
Gambar 23: Halaman Pembangunan SPAM

1. Input Pembangunan SPAM

Menampilkan halaman form input pembangunan SPAM

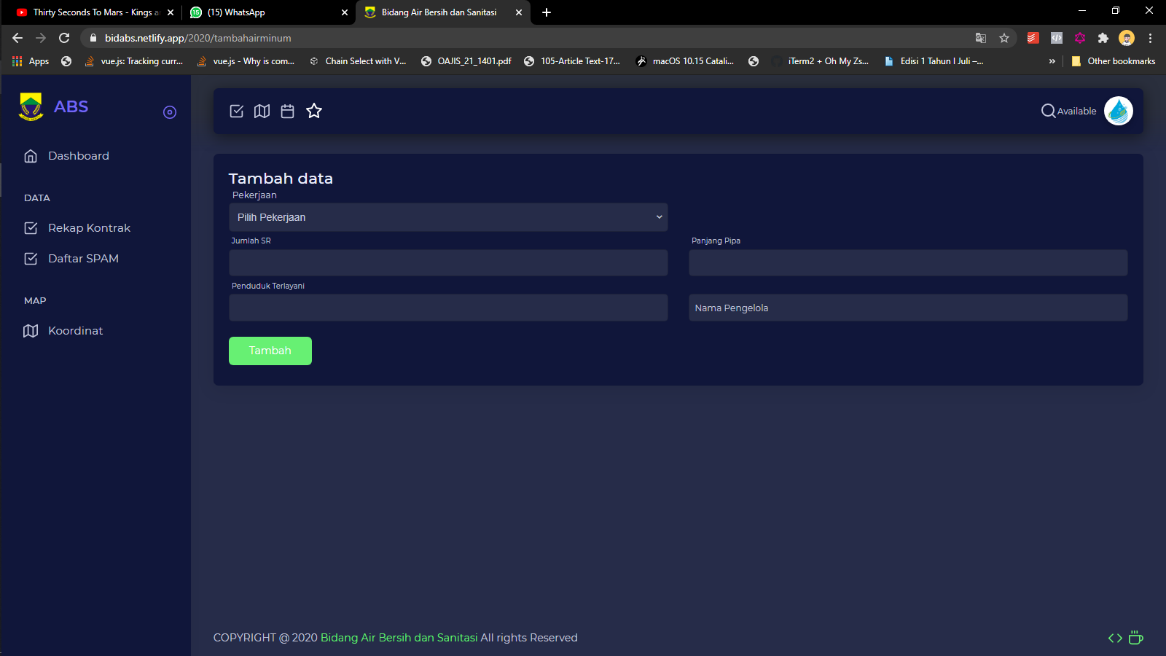
Gambar 24: Halaman Input Daftar SPAM

1. Pembangunan Sanitasi

Menampilkan semua pekerjaan dengan kegiatan Pembangunan Sanitasi

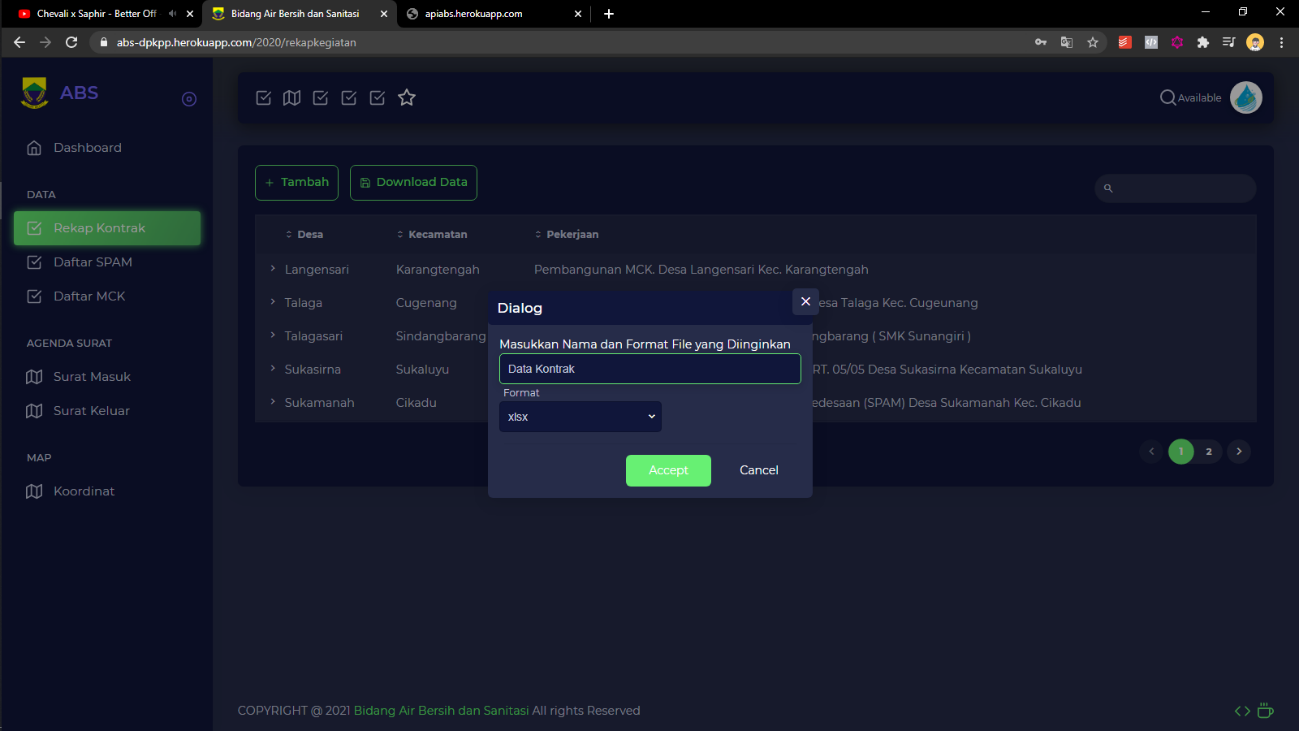
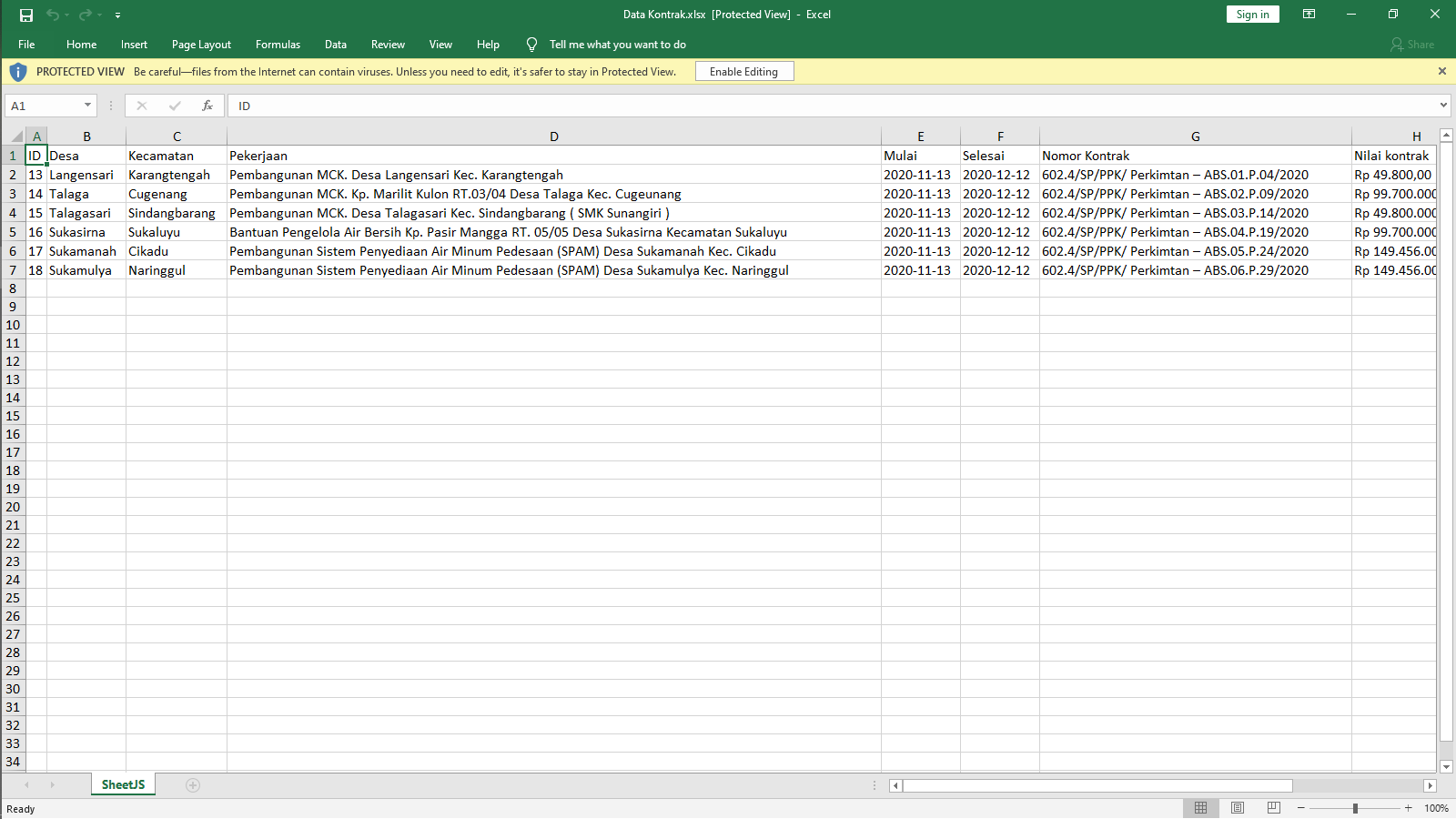
Gambar 25: Halam Daftar Pembangunan Sanitasi

1. Input Pembangunan Sanitasi

Menampilkan form input data pembangunan Sanitasi.

Gambar 26: Halaman Input Daftar Pembangunan Sanitasi

1. Fitur Laporan

Pada setiap menu yang pada aplikasi, terdapat fitur download data untuk mengunduh rekap data kedalam bentuk file .xlsx atau Microsoft Excel, CSV, dan txt.

Gambar 27: Data Excel

Gambar 28: Download Data

## Pengujian

Pengujian dilakukan dengan metode e2e testing menggunakan plugin Cypress pada vue.js.

1. Pengujian Halaman Login

Pengujian e2e testing Halaman *Login* berhasil dengan waktu penyelesaian dalam 13.13 detik.

*Script* dan Hasil Pengujian Halaman *Login* dengan perintah *Logout* apabila halaman *dashboard* berhasil dimuat:

describe('Pengujian Halaman Login', () **=>** {

it('Pengujian e2e Halaman Login', () **=>** {

**const** user = {

email: 'ilhamtaufiq@gmail.com',

password: 'cianjur22'

}

cy.visit('http://localhost:8080/pages/login')

cy.get('input[name="email"]').type(user.email)

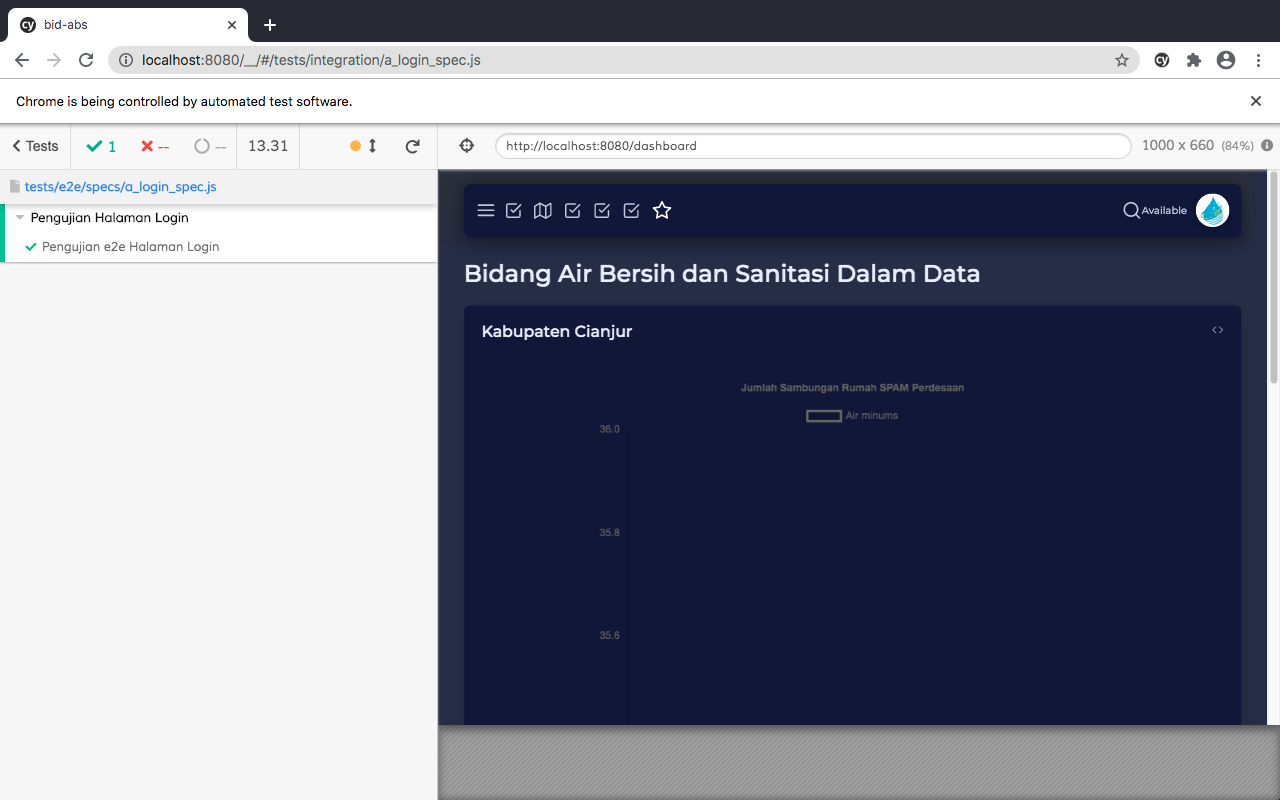
cy.get('input[name="password"]').type(user.password).type('{enter}')

cy.get('.vs-con-dropdown').click()

cy.contains('Logout').click()

cy.url().should('contain', 'http://localhost:8080/')

})

})

Gambar 29: Hasil Pengujian Halaman Login

1. Pengujian Halaman Rekap Kontrak

Pengujian e2e testing Halaman Rekap Kontrak berhasil dengan waktu penyelesaian dalam 14.15 detik.

*Script* dan Hasil Pengujian Halaman Rekap kegiatan dengan perintah *download* *data* setelah halaman berhasil dimuat:

describe('Pengujian Halaman Rekap Kontrak, () **=>** {

it('Pengujian e2e Halaman Rekap Kontrak, () **=>** {

**const** user = {

email: 'ilhamtaufiq@gmail.com',

password: 'cianjur22',

}

cy.visit('http://localhost:8080/2020/rekapkegiatan')

cy.get('input[name="email"]').type(user.email)

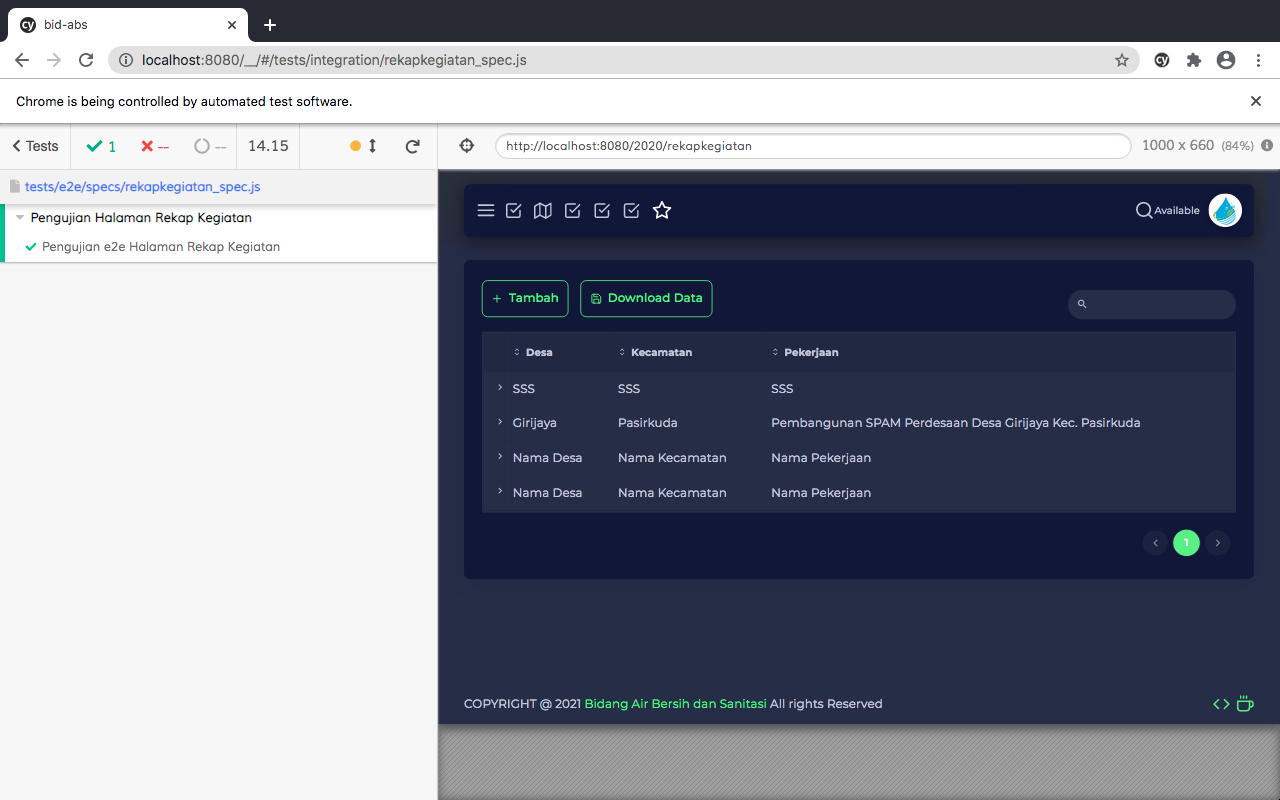
cy.get('input[name="password"]').type(user.password).type('{enter}')

cy.contains('Download Data').click()

cy.url().should('contain', 'http://localhost:8080/2020/rekapkegiatan')

cy.contains('Accept').click()

})

})

Gambar 30: Hasil Pengujian Halaman Rekap Kontrak

1. Pengujian Halaman Input Daftar Kontrak

Pengujian e2e testing Halaman Input Data Kontrak berhasil dengan waktu penyelesaian dalam 21.96 detik.

*Script* dan Hasil Pengujian Halaman Input Data Kontrak dengan *dummy* input kedalam form input:

describe('Pengujian Halaman Input Data Kegiatan', () **=>** {

it('Pengujian e2e Halaman Input Data Kegiatan', () **=>** {

**const** user = {

email: 'ilhamtaufiq@gmail.com',

password: 'cianjur22',

}

cy.visit('http://localhost:8080/2020/rekapkegiatan')

cy.get('input[name="email"]').type(user.email)

cy.get('input[name="password"]').type(user.password) .type('{enter}')

cy.contains('Tambah').click()

cy.url().should('contain', 'http://localhost:8080/2020/datakontrak')

cy.get(':nth-child(3) > :nth-child(1) > .vs-component > .vs-con-input > .vs-inputx').type('Nama Pekerjaan')

cy.get(':nth-child(2) > :nth-child(2) > .vs-component > .vs-con-input > .vs-inputx').type('Nama Kegiatan')

cy.get(':nth-child(3) > :nth-child(2) > .vs-component > .vs-con-input > .vs-inputx').type('Nama Desa')

cy.get(':nth-child(4) > .mb-2 > .vs-component > .vs-con-input > .vs-inputx').type('Nama Kecamatan')

cy.get(':nth-child(4) > .mb-6 > .vs-component > .vs-con-input > .vs-inputx').type('Nomor Kontrak')

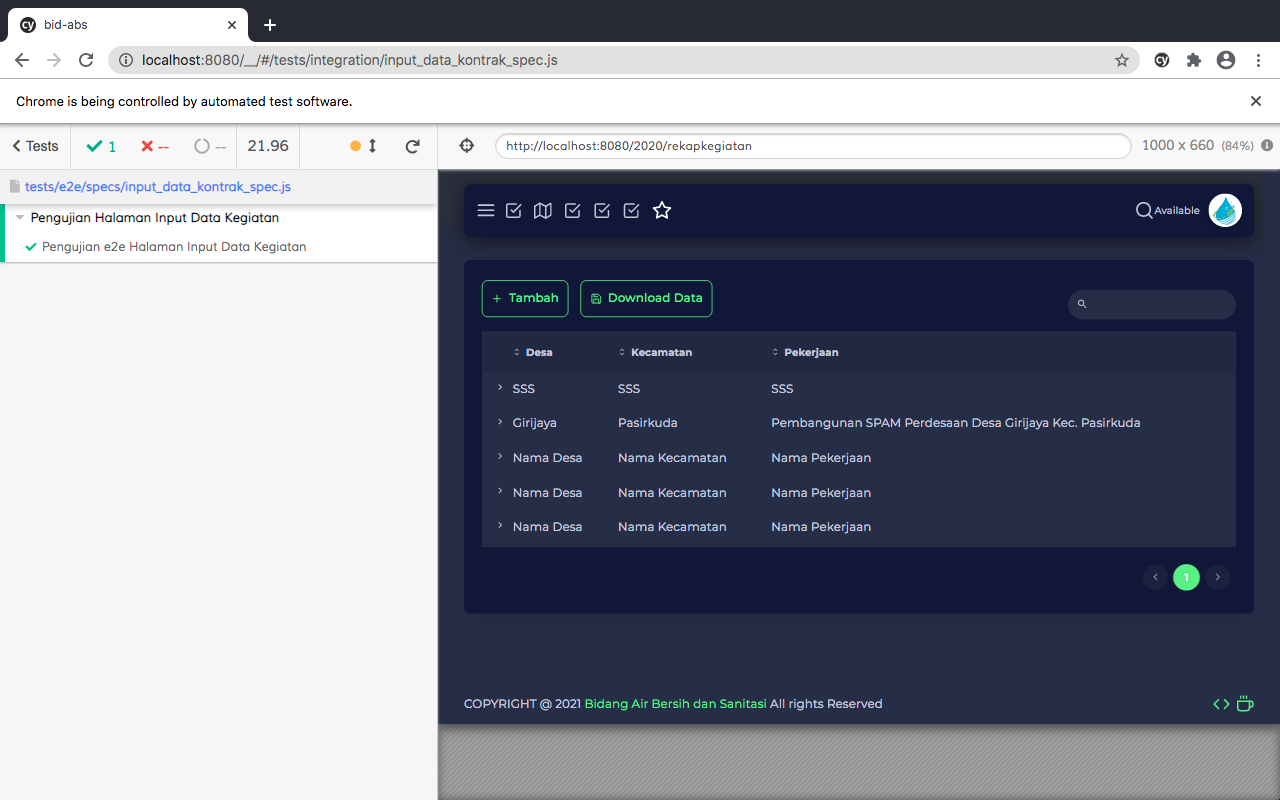
cy.get(':nth-child(5) > .mb-2 > .vs-component > .vs-con-input > .vs-inputx').type('1919191')

cy.get(':nth-child(5) > :nth-child(2) > .vs-component > .vs-con-input > .vs-inputx').type('Nama Rekanan')

cy.get('.vs-button-text').click()

})

})



Gambar 31: Hasil Pengujian Halaman Input Data Kontrak

1. Pengujian Navigasi Menu

Pengujian e2e testing semua Menu Navigasi yang terdapat pada aplikasi berhasil dengan waktu penyelesaian dalam 15.21 detik.

*Script* dan Hasil Pengujian Menu Navigasi:

context('Pengujian Navigasi', () **=>** {

it('Pengujian e2e navigasi', () **=>** {

**const** user = {

email: 'ilhamtaufiq@gmail.com',

password: 'cianjur22',

}

cy.visit('http://localhost:8080');

cy.get('input[name="email"]').type(user.email)

cy.get('input[name="password"]').type(user.password) .type('{enter}')

cy.get(':nth-child(1) > .con-vs-tooltip > .p-2 > .feather').click()

cy.get(':nth-child(2) > .con-vs-tooltip > .p-2 > .feather').click()

cy.get(':nth-child(3) > .con-vs-tooltip > .p-2 > .feather').click()

cy.get(':nth-child(4) > .con-vs-tooltip > .p-2 > .feather').click()

cy.get(':nth-child(5) > .con-vs-tooltip > .p-2 > .feather').click()

});

});



Gambar 32: Hasil Pengujian Navigasi Aplikasi

# BAB VI PENUTUP

## Kesimpulan

Berdasarkan hasil analisis Aplikasi e-Monitoring Program Pengelolaan dan Pengembangan Sistem Penyediaan Air Minum dan Sistem Air Limbah, maka penulis mengambil kesimpulan bahwa:

1. aplikasi dapat berfungsi untuk menyimpan, dan mengolah data yang ada.
2. aplikasi dapat memberikan data yang telah diproses untuk laporan.
3. Kepala bidang, kasi, dan staf lainnya dapat memonitoring semua usulan kegiatan yang ada di Bidang Air Bersih dan Sanitasi Dinas Perumahan Kawasan Permukiman dan Pertanahan Kabupaten Cianjur.

## Saran

Setelah penulis membangun dan merancang Aplikasi e-Monitoring Program Pengelolaan dan Pengembangan Sistem Penyediaan Air Minum dan Sistem Air Limbah, penulis dapat mengetahui apa yang menjadi kelebihan dan kekurangan dari sistem yang penulis susun.

Diharapkan program Aplikasi e-Monitoring Program Pengelolaan dan Pengembangan Sistem Penyediaan Air Minum dan Sistem Air Limbah telah dirancang dapat dikembangkan kembali dengan menambahkan hal-hal lainnya yang diperlukan untuk mendukung terlaksananya *good govermance* di Lingkungan Pemerintah Daerah Kabupaten Cianjur, khususnya di Bidang Air Bersih dan Sanitasi pada Dinas Perumahan Kawasan Permukiman dan Pertanahan.

# DAFTAR PUSAKA

Agustina, Nuzulla. *Sistem Pengolahan Data*, Jakarta: PT Elex Media. Komputindo, 1992

Clarke, Steven. "*Measuring API Usability*".   
Dr. Dobb's, 2004.

David M. Kroenke, *Database Processing* Jilid 1 edisi 9, halaman 60.   
Erlangga

Dharwiyant, Sri. *Pengantar Unified Modeling Language* (UML), IlmuKomputer.Com, 2003.

Hakim S, Rachmad, dan Ir. Sutarto, Msi. *Mastering JavaTM Konsep Pemrograman dan Penerapannya Untuk Membuat Software Aplikasi*, Jakarta: PT. Elex Media Komputindo, 2010.

Jogiyanto, Hartono. *Analisis dan Desain Sistem Informasi*. Yogyakarta: CV. Andi Offset, 1999.

Kristanto, Andri. *Perancangan Sistem Informasi dan Aplikasinya*. Yogyakarta: Gava Media, 2008.

Kroenke, David M. dan David J. Auer. *Database Concepts*. edisi ke-3. New York: Prentice, 2007.

Laudon, Kenneth C.; Laudon, Jane P. *Sistem Informasi Manajemen*. Palgrave, Basingstoke, 2007.

Peraturan Bupati Nomor 76 Tahun 2016 *Tentang Tugas Dan Fungsi Serta Tata Kerja Unit Organisasi Di Lingkungan Dinas Perumahan, Kawasan Permukiman Dan Pertanahan Kabupaten Cianjur*, 2016.

Rama, Jones, *Sistem Informasi Akuntansi* halaman 78.   
Salemba.

(internet) Hariyadi, Efendi. “Membuat Proyek Pertama Menggunakan Heroku”. <https://www.codepolitan.com/membuat-proyek-pertama-heroku-58b872c6217eb> (diakses pada: 22/12/2020)

(Internet) vuejs.org “Introduction” <https://vuejs.org/v2/guide/> (dikases pada: 22/12/2020)

(Internet) vuesax.com “Documentation” <https://vuesax.com/docs/guide/> (diakses pada: 22/12/2020)

(Internet) medium.com/codelabs-unikom “Memahami GraphQL (Query Language)” [https://medium.com/codelabs-unikom/memahami-graphql-query-language-964fc6ea6a12](https://medium.com/codelabs-unikom/memahami-graphql-query-language-964fc6ea6a12%20%20)  (diakses pada: 22/12/2020)

# BIODATA PENULIS

# LAMPIRAN