

LAPORAN AKHIR STANDARISASI CODING APLIKASI KOMPETENSI DAN E-BBM DAN E- SIGN

Dokumen ini berisi mengenai proses implementasi standarisasi coding dan e-Sign. Aplikasi yang digunakan merupakan pecontohan dan sebagai awalan pengembangan aplikasi di PDAM Surya Sembada

Tim
Pengembang
Aplikasi PDAM
Surya Sembada

Kata Pengantar

Dengan rasa syukur dan semangat pengabdian, kami dengan ini mempersembahkan laporan pekerjaan berjudul "Standarisasi Coding dan Clean Code pada Aplikasi Kompetensi dan E-BBM serta Pengaplikasian E-Sign." Laporan ini mencerminkan perjalanan kami dalam mengimplementasikan praktik terbaik dalam dunia pengembangan perangkat lunak pada dua aplikasi kritis, yakni Aplikasi Kompetensi dan Aplikasi E-BBM. Dalam era dinamis teknologi informasi, standarisasi coding dan clean code menjadi pondasi utama untuk mencapai perangkat lunak yang tidak hanya efisien tetapi juga mudah dipahami, dikelola, dan dikembangkan.

Kami ingin menyampaikan apresiasi setinggi-tingginya kepada seluruh tim pengembang yang telah berdedikasi dalam menghadirkan solusi yang inovatif dan efektif. Dalam laporan ini, kami akan memaparkan perincian langkah-langkah implementasi standarisasi coding dan clean code pada kedua aplikasi tersebut, serta bagaimana penerapan e-sign pada proses pengelolaan dokumen.

Semoga laporan ini dapat memberikan wawasan yang berharga dalam upaya pengembangan perangkat lunak yang berkualitas tinggi dan sesuai dengan tuntutan zaman. Kami menyadari bahwa pembelajaran dan perbaikan terus-menerus adalah kunci keberhasilan, dan laporan ini adalah hasil kolaborasi dan semangat bersama untuk meraih standar keunggulan dalam dunia pengembangan perangkat lunak.

Akhirnya, kami berharap bahwa laporan ini dapat memberikan inspirasi dan panduan bagi para pembaca yang tengah menjalankan atau merencanakan proyek pengembangan perangkat lunak. Terima kasih atas dukungan, kontribusi, dan semangat pantang menyerah yang telah mewarnai perjalanan pengembangan ini.

Tim Penyusun

Daftar Isi

Kata Pengantar.....	1
Daftar Isi	2
Daftar Gambar	4
Daftar Tabel	7
BAB I PENDAHULUAN	8
1.1 Latar Belakang	8
1.2 Tujuan	9
1.3 Ruang Lingkup	9
BAB II STANDARISASI CODING, CLEAN CODE DAN ELEKTRONIK SIGN	10
2.1 Standarisasi Coding	10
2.2 Manfaat Standarisasi Coding	10
2.3 Standar Coding pada Industri	11
2.4 Definisi Clean Code	13
2.5 Manfaat Penggunaan Clean Code	14
2.6 Implementasi Clean Code pada Industri	16
2.7 Elektronik Sign	17
BAB III OBJEK IMPLEMENTASI	19
3.1 Aplikasi Kompetensi	19
3.1.1 Transaksi Penyimpanan Data	19
3.1.2 Proses Validasi Keahlian	20
3.1.3 Use Case Diagram	21
3.1.4 Entity Relation Diagram	21
3.2 Aplikasi eBBM	22
3.2.1 Pembuatan dan Pengelolaan Kupon Digital	22
3.2.2 Penerimaan dan Validasi Kupon Digital saat Pembelian BBM	23
3.2.3 Pembayaran Menggunakan Kupon Digital	24
3.2.4 Use Case Diagram	24
3.2.5 Entity Relation Diagram	26
3.3 Aplikasi Workflow	27
3.3.1 Pengelolaan Surat	28
BAB IV TEKNIS APLIKASI	29

4.1	Software Architecture.....	29
4.2	Software Stack.....	30
BAB V PELAKSANAAN PEKERJAAN.....		31
5.1	Kickoff Meeting.....	31
5.2	Pembuatan Aplikasi	32
5.3	Proses Standarisasi Coding.....	35
5.3.1	Proses Standarisasi Coding Aplikasi Kompetensi.....	36
5.3.2	Proses Standarisasi Coding Aplikasi eBBM.....	39
5.3.3	Proses Implementasi e-Sign pada Workflow	44
5.4	Hasil Aplikasi	50
5.4.1	Aplikasi E-BBM.....	51
5.4.2	Aplikasi Kompetensi	62
5.4.3	Aplikasi Workflow	74

Daftar Gambar

Gambar 1 Use Case Diagram Aplikasi Kompetensi	21
Gambar 2 Entity Relation Digram Dari Aplikasi Kompetensi.....	22
Gambar 3 Use Case Diagram	25
Gambar 4 Entity Relation Diagram Aplikasi E-Bbm	27
Gambar 5 Arsitektur Aplikasi E-Bbm	29
Gambar 6 Proses Kickoff Meeting	31
Gambar 7 Proses Presentasi Pada Kickoff Meeting.....	32
Gambar 8 Proses Diskusi Pengembangan	33
Gambar 9 Prose Diskusi Instalasi.....	33
Gambar 10 Proses Ujicoba Internal	34
Gambar 11 Proses Dikusi Dengan User	34
Gambar 12 Controller Coding Yang Sudah Standarisasi	36
Gambar 13 Unit Test Coding Yang Sudah Standarisasi	37
Gambar 14 Docker Compose Coding Yang Sudah Standarisasi	38
Gambar 15 Git Atau Cid Command Coding Yang Sudah Standarisasi.....	39
Gambar 16 Controller Coding Ebbm Yang Sudah Standarisasi	40
Gambar 17 Unit Test Kupon Ebbm Coding Yang Sudah Standarisasi	41
Gambar 18 Git Cid Ebbm Coding Yang Sudah Standarisasi	42
Gambar 19 Coding Docker Compose Yang Sudah Standarisasi	43
Gambar 20 Function Cekpeg	44
Gambar 21 Function Suratkeluarcetak	44
Gambar 22 Function Datasuratcetak	49
Gambar 23 Approval Link	50
Gambar 24 View Approval	50
Gambar 25 Function Suratkeluar	50

Gambar 26 Dasboard Aplikasi Ebbm	51
Gambar 27 Tampilan Menu Pengajuan Rutin	51
Gambar 28 Tampilan Menu Buat Pengajuan Rutin	52
Gambar 29 Tampilan Menu Pengajuan Insidentil.....	52
Gambar 30 Tampilan Menu Buat Baru Pengajuan Insidentil.....	53
Gambar 31 Tampilan Menu Pegawai.....	53
Gambar 32 Tampilan Menu Master Jabatan	54
Gambar 33 Tampilan Popup Buat Master Jabatan	54
Gambar 34 Tampilan Menu Master Unit	55
Gambar 35 Tampilan Popup Buat Master Unit	55
Gambar 36 Tampilan Menu Master Jenis Bbm	56
Gambar 37 Tampilan Popup Buat Master Jenis Bbm	56
Gambar 38 Tampilan Menu Master Spbu	57
Gambar 39 Tampilan Popup Buat Master Spbu	57
Gambar 40 Tampilan Menu Master Petugas Spbu.....	58
Gambar 41 Tampilan Popup Buat Master Petugas Spbu.....	58
Gambar 42 Tampilan Menu Master Pekerjaan.....	59
Gambar 43 Tampilan Popup Buat Master Pekerjaan.....	59
Gambar 44 Tampilan Menu Master User.....	60
Gambar 45 Tampilan Popup Buat Master User.....	60
Gambar 46 Tampilan Menu Master Role	61
Gambar 47 Tampilan Popup Buat Master Role	61
Gambar 48 Tampilan Utama	62
Gambar 49 Popup Pembuatan Pegawai Baru	62
Gambar 50 Form Pengajuan Kompetensi.....	63
Gambar 51 Form Buat Pengajuan	63
Gambar 52 Page Menu Pegawai.....	64
Gambar 53 Page Master Asal Kompetensi	64

Gambar 54 Popup Master Asal Kompetensi.....	65
Gambar 55 Page Master Kompetensi.....	65
Gambar 56 Popup Master Buat Kompetensi.....	66
Gambar 57 Page Master Jenis Kompetensi.....	66
Gambar 58 Popup Master Buat Jenis Kompetensi	67
Gambar 59 Page Master Jabatan.....	67
Gambar 60 Popup Master Buat Jabatan	68
Gambar 61 Page Master Unit.....	68
Gambar 62 Popup Master Buat Unit	69
Gambar 63 Page Master Penyelenggara.....	69
Gambar 64 Popup Master Buat Penyelenggara	70
Gambar 65 Page Master Jenis Sertifikat	70
Gambar 66 Popup Master Buat Jenis Sertifikat.....	71
Gambar 67 Page Master Pelatihan	71
Gambar 68 Popup Master Buat Pelatihan	72
Gambar 69 Page Master User.....	72
Gambar 70 Popup Master Buat User	73
Gambar 71 Page Master Role	73
Gambar 72 Popup Master Buat Role	74
Gambar 73 Membuat Surat Baru.....	74
Gambar 74 Mengisi Data Surat Keluar	75
Gambar 75 Data Surat Keluar Tampil Setelah Disimpan	75
Gambar 76 Bentuk Cetak Pertama Setelah Pembuatan Dan Belum Di Approve.....	76
Gambar 77 Approve Surat Melalui Pihak Pertama.....	76
Gambar 78 Masukkan Key Id Untuk Melakukan Approve.....	77
Gambar 79 Approve Sukses	77
Gambar 80 Sukses Approve Dan Key Id Sesuai, Maka Surat Telah Memiliki Ttd Barcode	78
Gambar 81 Approve Surat Melalui Pihak Kedua	78

Gambar 82 Memasukkan Key Id.....	79
Gambar 83 File Tersimpan	79
Gambar 84 Surat Setelah Approve Kedua.....	80

Daftar Tabel

TABEL 1 USER ROLE DAN KAPABILITAS	26
---	----

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah menciptakan perubahan yang signifikan dalam berbagai sektor pelayanan publik, termasuk dalam operasional PDAM Surya Sembada yang memiliki peran vital dalam menyediakan layanan air bersih bagi masyarakat. Di tengah dinamika ini, kebutuhan akan efisiensi operasional dan kualitas perangkat lunak yang handal di PDAM menjadi semakin penting. Sebagai bagian dari sektor pelayanan publik, menghadapi berbagai tantangan unik yang memerlukan pendekatan yang tepat dalam pengembangan perangkat lunak. Adopsi standarisasi coding dan konsep Clean Code menjadi langkah strategis untuk mengatasi dinamika dan kompleksitas dalam pengembangan aplikasi PDAM Surya Sembada.

Faktor-faktor yang memberikan konteks khusus terkait penerapan standarisasi coding dan Clean Code dalam PDAM Surya Sembada melibatkan beberapa aspek penting. Pertama, integrasi dengan sistem terdahulu menjadi tantangan utama, mengingat PDAM Surya Sembada umumnya memiliki infrastruktur teknologi yang sudah ada sebelumnya. Adopsi standarisasi coding memberikan manfaat yang membantu memperlancar proses integrasi, memastikan aplikasi baru dapat berjalan harmonis dengan sistem yang sudah ada tanpa menimbulkan konflik atau ketidaksesuaian.

Faktor skalabilitas untuk pertumbuhan pelayanan PDAM Surya Sembada yang berkembang seiring dengan pertumbuhan kota atau wilayah layanan, diperlukan sistem yang dapat berkembang dengan mudah tanpa menimbulkan masalah teknis yang serius. Standarisasi coding dan Clean Code mendukung skalabilitas ini dengan memastikan bahwa aplikasi dapat diadaptasi untuk memenuhi tuntutan pertumbuhan dengan efisien. Aspek pemeliharaan dan dukungan teknis menjadi kunci dalam menjaga kinerja system di PDAM Surya Sembada. Clean Code membantu meminimalkan potensi kesalahan dan kerumitan dalam pemeliharaan, memastikan bahwa tim teknis dapat memberikan dukungan secara efisien dan responsif. Dengan mempertimbangkan faktor-faktor ini, penerapan standarisasi coding dan Clean Code di lingkungan PDAM Surya Sembada dapat memberikan dampak positif yang signifikan dalam menjaga kelancaran operasional dan kualitas layanan yang disediakan.

Aspek keamanan dan ketersediaan layanan merupakan prioritas utama dalam operasional PDAM Surya Sembada. Standarisasi coding menempatkan fokus pada pengembangan aplikasi yang tidak hanya handal namun juga aman dari potensi ancaman keamanan. Keterlibatan dengan pihak eksternal menjadi dinamika lain yang perlu dipertimbangkan. PDAM Surya Sembada sering berkolaborasi dengan pihak ketiga, seperti kontraktor atau penyedia layanan teknologi. Dalam konteks ini, standarisasi coding menyediakan panduan yang konsisten dan jelas, mempermudah

kerjasama dengan pihak eksternal serta meminimalisir potensi hambatan yang mungkin muncul.

Faktor keamanan tidak hanya berkaitan dengan pengembangan perangkat lunak, tetapi juga mencakup aspek keamanan dalam kegiatan administratif, seperti penerbitan surat tugas. Penguatan aspek keamanan dapat dilakukan dengan implementasi e-sign (tanda tangan elektronik) pada kegiatan penerbitan surat tugas menjadi langkah yang baik. Integrasi e-sign meningkatkan tingkat keamanan dalam proses administratifnya, termasuk pengelolaan surat tugas. E-sign memberikan tanda tangan yang terenkripsi dan sah secara hukum, mengurangi risiko pemalsuan dan meningkatkan otentikasi dokumen.

Penerapan standarisasi coding dan Clean Code di PDAM Surya Sembada diharapkan adanya peningkatan terhadap efisiensi operasional dan kualitas layanan. Melalui standarisasi coding, diharapkan dapat mencapai konsistensi, memudahkan integrasi, dan meningkatkan interoperabilitas dengan layanan. Clean Code menghasilkan kode yang mudah dimengerti, sederhana, dan terstruktur, sehingga diharapkan mengurangi potensi kesalahan, dan mempermudah pemeliharaan. Aspek keamanan di mana penambahan e-sign diharapkan dapat mengoptimalkan sistem keamanan dan melindungi data penting. Skalabilitas sistem layanan yang meningkat, memungkinkan PDAM Surya Sembada menyesuaikan diri seiring dengan pertumbuhan wilayah layanan tanpa mengorbankan stabilitas. Clean Code diharapkan meningkatkan dukungan teknis, meminimalisir waktu pemecahan masalah, dan meningkatkan responsivitas terhadap perubahan kebutuhan. Gabungan standarisasi coding dan Clean Code diharapkan membawa PDAM Surya Sembada efisiensi, keberlanjutan, dan kualitas layanan yang lebih baik di era transformasi digital.

1.2 Tujuan

Implementasi standarisasi coding dan clean code memiliki tujuan sebagai berikut:

1. Meningkatkan kualitas *source code* aplikasi yang efisien, berkelanjutan dan mudah dipahami
2. Kemudahan dalam melakukan pemeliharaan dan pengembangan *software* dalam lingkungan PDAM Surya Sembada

1.3 Ruang Lingkup

Ruang lingkup implementasi penggunaan standarisasi coding dan clean code dilakukan pada pengembangan aplikasi kompetensi dan E-BBM. Serta penggunaan e-Sign pada aplikasi surat elektronik Workflow yang telah ada di PDAM Surya Sembada.

BAB II STANDARISASI CODING, CLEAN CODE DAN ELEKTRONIK SIGN

2.1 Standarisasi Coding

Standarisasi coding merujuk pada praktik menyusun aturan dan pedoman yang konsisten dalam penulisan kode sumber dalam suatu proyek perangkat lunak. Tujuannya adalah untuk menciptakan konsistensi, kejelasan, dan kemudahan pemeliharaan dalam lingkungan pengembangan perangkat lunak. Definisi standarisasi coding mencakup berbagai aspek, termasuk gaya penulisan, konvensi penamaan variabel, tata letak kode, komentar, dan pedoman lainnya yang memengaruhi struktur dan penulisan kode.

Adopsi standarisasi coding sangat penting dalam konteks pengembangan perangkat lunak tim, di mana beberapa pengembang bekerja bersama-sama pada proyek yang sama. Dengan standar yang jelas, setiap anggota tim dapat memahami kode yang ditulis oleh rekan-rekan mereka dengan lebih mudah, meminimalkan potensi kesalahan, dan meningkatkan keterbacaan serta pemeliharaan kode.

Definisi standarisasi coding sering kali melibatkan penetapan aturan terkait indentasi, panjang baris kode, penggunaan spasi atau tab, dan gaya penulisan komentar. Selain itu, standar tersebut dapat mencakup pedoman terkait penggunaan fungsi atau metode, manajemen kesalahan, dan praktik terkait keamanan. Adanya standar yang diikuti oleh seluruh tim membantu menciptakan lingkungan pengembangan yang lebih terstruktur, efisien, dan mudah dipelihara.

2.2 Manfaat Standarisasi Coding

Implementasi standarisasi coding di lingkungan PDAM Surya Sembada memiliki sejumlah manfaat untuk mendukung pengelolaan aplikasi yang efisien dalam siklus pengembangan. Selain pengelolaan aplikasi yang efisien terdapat manfaat-manfaat lain yaitu:

- 1. Konsistensi Kode**

Standarisasi coding menciptakan konsistensi dalam penulisan kode di seluruh proyek. Konsistensi ini melibatkan pemilihan format, konvensi penamaan variabel, dan gaya penulisan yang seragam. Konsistensi tersebut mempermudah pemahaman dan kolaborasi tim, mengurangi kebingungan, dan meningkatkan efisiensi kerja.

- 2. Keterbacaan dan Pemeliharaan**

Kode yang ditulis sesuai standar lebih mudah dibaca dan dipahami oleh anggota tim yang lain, bahkan oleh pengembang yang tidak terlibat dalam

penulisan aslinya. Hal ini mempermudah proses pemeliharaan dan perbaikan bug, karena setiap anggota tim dapat dengan cepat menemukan dan memahami bagian-bagian kode yang perlu dimodifikasi.

3. Kualitas Kode yang Tinggi

Standarisasi coding mempromosikan praktik pengkodean yang baik dan mengarah pada penulisan kode yang bersih dan efisien. Hal ini membantu dalam mencegah atau mengidentifikasi potensi kesalahan, meningkatkan keamanan, dan mengurangi risiko bug atau masalah teknis lainnya.

4. Integrasi yang Lebih Mudah

Proyek perangkat lunak sering kali melibatkan integrasi dengan komponen atau sistem lain. Standarisasi coding mempermudah proses integrasi dengan memastikan bahwa antarmuka dan format data yang digunakan konsisten di seluruh proyek.

5. Penyelesaian Konflik

Proses pengembangan perangkat lunak tim, mungkin ada banyak pengembang yang bekerja pada bagian yang berbeda. Standarisasi coding membantu menghindari konflik dalam penggabungan (merge) perubahan kode dari berbagai kontributor, karena semua pengembang mengikuti aturan yang sama.

6. Peningkatan Produktivitas

Konsistensi dan keterbacaan yang tinggi, pengembang dapat bekerja lebih efisien dan produktif. Pengembang dapat lebih fokus pada logika aplikasi daripada terjebak dalam memahami gaya penulisan kode yang bervariasi.

7. Kepatuhan Standar Industri

Mengadopsi standarisasi coding seringkali berarti mematuhi standar industri yang telah diakui. Ini dapat meningkatkan reputasi proyek, memudahkan integrasi dengan alat atau kerangka kerja tertentu, dan meningkatkan portabilitas kode.

2.3 Standar Coding pada Industri

Standarisasi coding industry merujuk pada praktik menyusun aturan dan pedoman yang konsisten dalam penulisan source code yang banyak digunakan atau diadaptasi dalam industri pembuatan perangkat lunak. standarisasi coding industri mencerminkan akumulasi praktik terbaik dan pengalaman dari komunitas pengembang yang luas. Hal ini dapat memberikan pandangan yang lebih luas dan memperkaya pemahaman tim pengembang terhadap cara terbaik untuk merancang, menulis, dan

memelihara kode. Dengan menerapkan standar ini, perusahaan dapat mengoptimalkan proses pengembangan perangkat lunaknya, meningkatkan kualitas produk, dan meningkatkan daya saing dalam industri teknologi yang terus berkembang. Berikut beberapa contoh standarisasi coding industri yang umum diadopsi:

1. PEP 8 (Python Enhancement Proposal 8) – Python

PEP 8 adalah standar penulisan kode untuk bahasa pemrograman Python. Standar ini mencakup pedoman terkait gaya penulisan, penamaan variabel, tata letak kode, dan sejumlah praktik pengkodean lainnya. PEP 8 memastikan bahwa kode Python konsisten dan mudah dibaca oleh pengembang lain.

2. Java Code Conventions - Java

Standar ini mengatur tata letak kode, konvensi penamaan variabel, dan praktik pengkodean untuk bahasa pemrograman Java. Java Code Conventions membantu menciptakan kode yang konsisten di seluruh proyek Java, memudahkan pemeliharaan dan kolaborasi antar pengembang.

3. PSR (PHP Standards Recommendation) - PHP

PSR adalah serangkaian rekomendasi standar untuk bahasa pemrograman PHP yang dikembangkan oleh PHP-FIG (Framework Interoperability Group). PSR mencakup pedoman terkait otentikasi kelas, struktur direktori, dan penamaan interface, sehingga meningkatkan interoperabilitas antar framework dan aplikasi PHP.

4. Style Guide for Google- Java, Python, JavaScript, dan lainnya

Google menyediakan panduan gaya penulisan kode untuk berbagai bahasa pemrograman yang digunakan di perusahaannya, termasuk Java, Python, dan JavaScript. Panduan ini mencakup aspek-aspek seperti format kode, penamaan variabel, komentar, dan praktik pengkodean lainnya.

5. Airbnb JavaScript Style Guide - JavaScript

Airbnb JavaScript Style Guide adalah standar penulisan kode untuk proyek-proyek yang menggunakan JavaScript. Standar ini mencakup pedoman terkait penamaan variabel, struktur fungsi, dan praktik pengkodean lainnya. Airbnb membuat standar ini berdasarkan pengalaman mereka dalam mengembangkan aplikasi JavaScript berskala besar.

6. Microsoft's .NET Framework Design Guidelines - C#

Standar ini memberikan pedoman terinci untuk pengembangan dengan menggunakan bahasa pemrograman C# dan framework .NET. Pedoman ini mencakup aspek-aspek seperti penamaan variabel, struktur kelas, dan

manajemen kesalahan, dengan tujuan untuk meningkatkan konsistensi dan kualitas kode dalam ekosistem .NET.

Menerapkan standarisasi coding industri seperti contoh-contoh di atas membantu memastikan bahwa kode yang dihasilkan konsisten, mudah dibaca, dan dapat dipahami oleh pengembang lain. Hal ini menjadi kunci untuk memelihara proyek dengan efisien, meningkatkan kolaborasi tim, dan meminimalkan potensi kesalahan.

2.4 Definisi Clean Code

Clean code merupakan konsep dalam pengembangan perangkat lunak yang menekankan pada penulisan kode yang mudah dibaca, dimengerti, dan dapat dipelihara dengan baik. Ide dasar di balik clean code adalah menciptakan kode yang efisien, efektif, dan mengikuti praktik terbaik dalam pengkodean. Clean code tidak hanya berfokus pada fungsionalitas suatu program, tetapi juga pada kejelasan dan kebersihan struktur serta gaya penulisannya. Prinsip-prinsip dan praktik umum dalam clean code meliputi:

1. Deskriptif dan Informatif

Clean code harus memiliki nama variabel, fungsi, dan kelas yang deskriptif dan informatif. Nama yang baik membantu pengembang dan pembaca kode untuk dengan cepat memahami tujuan dan fungsi dari bagian kode tersebut.

2. Mudah Dibaca

Struktur kode seharusnya mudah dibaca dan mengikuti pola tertentu. Indentasi yang konsisten, pemformatan yang baik, dan penggunaan spasi atau tab yang konsisten membantu dalam meningkatkan kejelasan.

3. Kecil dan Modular

Clean code mengadvokasi pembuatan fungsi dan kelas yang kecil, fokus pada satu tugas atau tanggung jawab tertentu. Hal ini mempermudah pemahaman dan pemeliharaan kode serta memungkinkan penggunaan ulang yang lebih baik.

4. Menghindari Redundansi

Clean code mengurangi redundansi dengan menciptakan fungsi atau kelas yang dapat digunakan kembali, dan menghindari duplikasi kode yang tidak perlu. Sehingga dalam pembuatan aplikasi lebih efisien dan mengurangi potensi timbulnya kesalahan.

5. Pengelolaan Kesalahan yang Baik

Clean code memperhatikan manajemen kesalahan dengan memastikan bahwa kode memberikan pesan kesalahan yang informatif dan dapat membantu dalam menemukan masalah dengan cepat.

6. Uji Unit yang Baik

Clean code mendukung praktik pengujian unit yang baik. Kode yang mudah diuji membantu memastikan bahwa perubahan pada kode tidak menyebabkan masalah di bagian lain.

7. Mengikuti Prinsip SOLID

Prinsip SOLID adalah seperangkat pedoman yang mencakup Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, dan Dependency Inversion. Menerapkan prinsip-prinsip ini membantu menciptakan desain perangkat lunak yang baik dan mudah dipelihara.

Clean code bukan hanya tentang penampilan visual kode, tetapi juga tentang pemahaman dan pemeliharaan kode secara keseluruhan. Tujuan akhirnya adalah menciptakan basis kode yang kokoh, fleksibel, dan dapat bertahan lama, memungkinkan untuk penambahan fitur dengan minim risiko dan upaya. Praktik clean code tidak hanya berguna dalam pengembangan perangkat lunak tim, tetapi juga berkontribusi pada budaya kolaboratif dan produktif di dalam tim pengembangan.

2.5 Manfaat Penggunaan Clean Code

Penggunaan clean code dalam pembuatan aplikasi memberikan sejumlah manfaat yang signifikan, baik dari segi pengembangan, pemeliharaan, maupun pengelolaan proyek. Berikut adalah beberapa manfaat utama penggunaan clean code:

1. Mudah Dibaca dan Dimengerti

Clean code membantu menciptakan kode yang mudah dibaca dan dimengerti oleh pengembang lain, termasuk oleh diri sendiri di masa depan. Dengan penamaan variabel yang jelas, struktur kode yang teratur, dan komentar yang informatif, pengembang dapat dengan cepat memahami tujuan dan fungsionalitas dari setiap bagian kode.

2. Mempermudah Pemeliharaan

Kode bersih mempermudah proses pemeliharaan dan pengembangan lebih lanjut. Pengembang dapat dengan cepat menemukan dan memodifikasi bagian kode tanpa takut mengganggu fungsi lainnya. Ini berkontribusi pada keberlanjutan proyek dan meminimalkan risiko perubahan pada kode.

3. Kualitas Lebih Tinggi

Clean code mempromosikan penggunaan praktik terbaik dalam pengembangan perangkat lunak. Hal ini berdampak positif pada kualitas keseluruhan aplikasi, dengan mengurangi potensi kesalahan, meningkatkan keamanan, dan memastikan efisiensi performa.

4. Pengelolaan Kesalahan yang Lebih Baik

Kode yang bersih memperhatikan manajemen kesalahan dengan memberikan pesan kesalahan yang informatif. Ini memudahkan proses debugging dan mempercepat identifikasi serta penyelesaian masalah.

5. Pengujian yang Mudah

Clean code mendukung praktik pengujian unit yang baik. Kode yang mudah diuji memastikan bahwa setiap perubahan pada kode dapat diuji dengan efisien, membantu dalam mencegah regresi dan memastikan keberlanjutan fungsionalitas.

6. Fleksibilitas dan Perubahan yang Mudah

Desain kode yang bersih mendukung fleksibilitas dan perubahan yang mudah. Kode modular dan terpisah memungkinkan pengembang untuk menyesuaikan atau menambahkan fitur tanpa mengubah seluruh struktur aplikasi.

7. Peningkatan Kolaborasi Tim

Clean code menciptakan dasar yang kuat untuk kolaborasi tim. Kode yang konsisten dan mudah dipahami memungkinkan anggota tim untuk lebih efektif bekerja bersama, berbagi tanggung jawab, dan saling mengerti dalam setiap kontribusi.

8. Penyusunan Kode yang Efisien

Clean code mendorong penggunaan algoritma dan struktur data yang efisien, mengoptimalkan kinerja aplikasi secara keseluruhan.

9. Peningkatan Kepuasan Pengguna

Memastikan kualitas dan keandalan kode, clean code berkontribusi pada pengalaman pengguna yang lebih baik. Aplikasi yang memiliki performa baik dan minim bug memberikan kepuasan kepada pengguna akhir.

10. Pemeliharaan dan Skalabilitas yang Lebih Mudah

Clean code membantu dalam pemeliharaan aplikasi dalam jangka panjang dan meningkatkan skalabilitas, memungkinkan aplikasi untuk tumbuh dan berkembang sejalan dengan kebutuhan yang berkembang.

Penggunaan clean code tidak hanya membawa manfaat secara langsung disisi teknis, tetapi juga berdampak positif pada produktivitas tim, keberlanjutan proyek, dan akhirnya, keberhasilan aplikasi dalam memenuhi kebutuhan dan harapan pengguna.

2.6 Implementasi Clean Code pada Industri

Implementasi clean code yang memberikan banyak manfaat, sehingga saat ini menjadi salah satu keharusan dalam penilaian pembuatan perangkat lunak. Berikut merupakan beberapa contoh implementasi clean code di industry:

1. Airbnb - JavaScript dan React

Airbnb memiliki panduan gaya penulisan kode untuk JavaScript dan React yang ketat. Mereka mendorong penggunaan ES6, menekankan pada penamaan variabel yang jelas, struktur fungsi yang kecil, dan penggunaan spasi yang konsisten. Panduan ini membantu menciptakan kode yang konsisten dan mudah dipelihara.

2. Google – Python

Google mempromosikan clean code melalui panduan penulisan kode untuk Python. Pedoman ini mencakup aturan-aturan tentang format kode, konvensi penamaan variabel, dan tata letak kode yang membantu menciptakan kode yang bersih dan konsisten di seluruh proyek.

3. Netflix – Java

Netflix menerapkan clean code dalam pengembangan perangkat lunak mereka yang menggunakan Java. Mereka menekankan pada prinsip SOLID, pemisahan tugas yang jelas antar kelas, dan penggunaan praktik terbaik Java untuk menciptakan aplikasi yang efisien dan mudah dipelihara.

4. Microsoft - .NET

Microsoft menyediakan panduan dan pedoman desain untuk pengembangan perangkat lunak menggunakan .NET. Prinsip-prinsip SOLID, penggunaan modul kecil, dan manajemen kesalahan yang baik adalah beberapa aspek yang ditekankan untuk menciptakan clean code dalam ekosistem .NET.

5. Uber - Go (Golang)

Uber menggunakan bahasa pemrograman Go (Golang) dan menerapkan praktik clean code dalam pengembangan aplikasi mereka. Mereka menekankan pada pembuatan fungsi dan metode yang kecil, pemilihan nama variabel yang jelas, dan pemisahan tugas yang baik antar paket kode.

6. Facebook – React

Facebook, pengembang React, mendorong praktik clean code dalam pengembangan antarmuka pengguna. Komponen yang kecil, jelas, dan fokus pada satu tugas, serta penamaan yang konsisten, adalah beberapa prinsip clean code yang diterapkan.

7. Twitter - Ruby on Rails

Twitter menggunakan Ruby on Rails untuk pengembangan aplikasinya. Mereka menekankan pada pembuatan model, view, dan controller yang bersih, serta penggunaan metode yang kecil dan fokus pada satu tanggung jawab.

Setiap implementasi clean code ini mengikuti pedoman dan praktik terkini untuk menciptakan kode yang mudah dibaca, dimengerti, dan dipelihara. Adopsi clean code di berbagai industri membantu perusahaan untuk meningkatkan efisiensi pengembangan, meningkatkan kolaborasi tim, dan memastikan kualitas perangkat lunak yang dihasilkan.

2.7 Elektronik Sign

Tanda tangan elektronik, atau e-signature, merujuk pada bentuk tanda tangan yang dibuat dan disimpan secara elektronik sebagai bentuk otorisasi atau persetujuan dalam lingkungan digital. Dengan adanya teknologi digital, tanda tangan elektronik telah menjadi cara yang umum dan sah untuk menyetujui dokumen atau transaksi secara online. Definisi tanda tangan elektronik mencakup berbagai metode dan teknologi, termasuk tetapi tidak terbatas pada:

1. Tanda Tangan Digital

Tanda tangan digital adalah bentuk tanda tangan elektronik yang menggunakan metode enkripsi kriptografis untuk memberikan keabsahan dan integritas kepada dokumen atau pesan yang ditandatangani. Biasanya, tanda tangan digital disertai dengan sertifikat digital yang terkait dengan identitas pihak yang menandatanganinya.

2. Username dan Password

Penggunaan kombinasi username dan password sebagai bentuk identifikasi dan persetujuan dalam pengaturan tertentu juga dapat dianggap sebagai tanda tangan elektronik. Meskipun tidak setinggi tingkat keamanan tanda tangan digital, hal ini umum digunakan dalam skenario seperti masuk ke akun atau menyetujui transaksi online.

3. PIN (Personal Identification Number)

Penggunaan PIN sebagai bentuk identifikasi pribadi yang harus dimasukkan oleh seseorang untuk memberikan otorisasi atau menyetujui suatu tindakan juga dapat dianggap sebagai tanda tangan elektronik. Biasanya digunakan dalam konteks perbankan atau keamanan kartu kredit.

4. Biometrik

Penggunaan data biometrik, seperti sidik jari, pemindaian wajah, atau pengenalan suara, untuk mengidentifikasi dan memberikan otorisasi dapat dianggap sebagai bentuk tanda tangan elektronik. Metode ini semakin populer karena tingkat keamanannya yang tinggi.

5. Klik Tombol (Clickwrap Agreement)

Menekan tombol "Setuju" atau serupa saat berada di halaman web dengan syarat dan ketentuan tertentu dapat dianggap sebagai tanda tangan elektronik. Jenis tanda tangan ini sering digunakan dalam perjanjian online atau pembelian produk dan layanan melalui internet.

6. Email dan Fax Elektronik

Pesan email atau fax yang berisi persetujuan atau tanda tangan dalam bentuk elektronik juga dianggap sebagai tanda tangan elektronik. Penerima pesan dianggap menerima dan menyetujui isi dokumen tersebut.

Tanda tangan elektronik memainkan peran penting dalam mendukung proses bisnis dan transaksi digital, menggantikan tanda tangan kertas tradisional dalam banyak konteks. Penting untuk dicatat bahwa keabsahan tanda tangan elektronik dapat diatur oleh hukum dan regulasi setempat, dan beberapa transaksi mungkin memerlukan bentuk tanda tangan yang lebih canggih untuk kebutuhan keamanan tertentu.

BAB III OBJEK IMPLEMENTASI

3.1 Aplikasi Kompetensi

Aplikasi kompetensi merupakan aplikasi yang digunakan untuk melakukan pencatatan keahlian dari pegawai dan proses verifikasi keahlian. Aplikasi ini nantinya digunakan sebagai pertimbangan tim pengembangan SDM dari PDAM Surya Sembada untuk menentukan kebutuhan peningkatan kemampuan dari masing-masing pegawai.

Aplikasi ini berawal dari permasalahan yang terjadi pada aplikasi Human Resource Information system (HRIS) yang tidak terintegrasi sehingga mengalami kesulitan untuk mengatur posisi dari karyawan dengan keahlian yang sesuai. Aplikasi ini merupakan aplikasi berbasis web yang diakses secara internal dalam perusahaan. Aplikasi ini juga dapat memberikan 3 opsi dalam pengembangan diri. Sesuai dengan keinginan yang telah direncanakan. Fungsi utama dari aplikasi kompetensi yaitu

3.1.1 Transaksi Penyimpanan Data

Aplikasi ini akan melakukan input terhadap data-data pegawai yang telah tersedia sebelumnya di aplikasi HRIS. Pegawai dapat menambahkan keahlian yang dimiliki beserta data-data pendukung lainnya. Transaksi ini hanya bisa dilakukan oleh pegawai sendiri dengan user aplikasi yang telah dibuat.

Aplikasi kompetensi yang mencakup fitur input keahlian pegawai memberikan kemudahan bagi pengguna, baik manajer maupun pegawai itu sendiri, untuk mengelola informasi tentang keahlian, keterampilan, dan kompetensi yang dimiliki oleh setiap anggota tim. Fitur tersebut mencakup pembuatan dan pengelolaan profil keahlian, termasuk informasi tentang pendidikan, sertifikasi, keterampilan teknis, dan bahasa yang dikuasai.

Melalui penilaian keterampilan, baik oleh pegawai maupun atasan, aplikasi ini memberikan pandangan mendalam tentang kemahiran dan pemahaman pegawai terhadap keterampilan tertentu. Adanya katalog keterampilan memudahkan pemilihan keterampilan yang sesuai, sementara pemetaan keterampilan memastikan penugasan pegawai sesuai dengan kebutuhan spesifik pekerjaan atau proyek.

Riwayat keterampilan melacak perubahan dan pembaruan pada profil keahlian dari waktu ke waktu, sementara notifikasi pembaruan memastikan bahwa data keahlian selalu terkini dan akurat. Integrasi dengan program pelatihan dan pengembangan memberikan rekomendasi pelatihan berdasarkan kebutuhan keterampilan pegawai.

Selain itu, penentuan tingkat keterampilan untuk setiap keterampilan memberikan informasi lebih rinci tentang tingkat keahlian dan kemahiran pegawai

dalam berbagai aspek. Semua fitur ini bekerja bersama-sama untuk menciptakan sistem yang terintegrasi dan efisien dalam mengelola data keahlian pegawai, memberikan organisasi wawasan yang lebih baik untuk perencanaan pengembangan dan manajemen sumber daya manusia.

3.1.2 Proses Validasi Keahlian

Fitur validasi keahlian dalam aplikasi kompetensi berperan penting dalam memastikan keakuratan dan keterandalan data yang dimasukkan oleh pengguna. Pertama, fitur ini mencakup pengecekan format data untuk memverifikasi apakah informasi keahlian dimasukkan sesuai dengan format yang benar dan diharapkan. Misalnya, memastikan bahwa tanggal sertifikasi atau kode sertifikasi mematuhi standar tertentu. Selanjutnya, validasi jenis data memastikan bahwa setiap input sesuai dengan tipe data yang diharapkan, seperti memverifikasi bahwa input angka adalah angka. Pemeriksaan keterbatasan nilai menghindari masuknya data di luar rentang nilai yang diizinkan, yang dapat mengakibatkan kesalahan analisis atau distorsi informasi.

Fitur konsistensi data memeriksa apakah informasi keahlian konsisten dengan data di dalam aplikasi atau sistem lainnya, sehingga menciptakan integritas data. Validasi sertifikasi dan lisensi memverifikasi keabsahan dan keberlakuan setiap sertifikasi atau lisensi yang dideklarasikan oleh pegawai. Pemberitahuan kesalahan memberikan pesan atau petunjuk yang jelas jika terjadi kesalahan dalam proses validasi, membantu pengguna untuk segera mengatasi dan memperbaiki masalah. Fitur auditing dan logging mencatat setiap aktivitas validasi dalam log atau catatan audit untuk pemantauan dan pelacakan historis data keahlian pegawai. Dengan menerapkan fitur-fitur ini, organisasi dapat memastikan bahwa data keahlian pegawai tetap akurat, valid, dan bermanfaat dalam konteks manajemen sumber daya manusia.

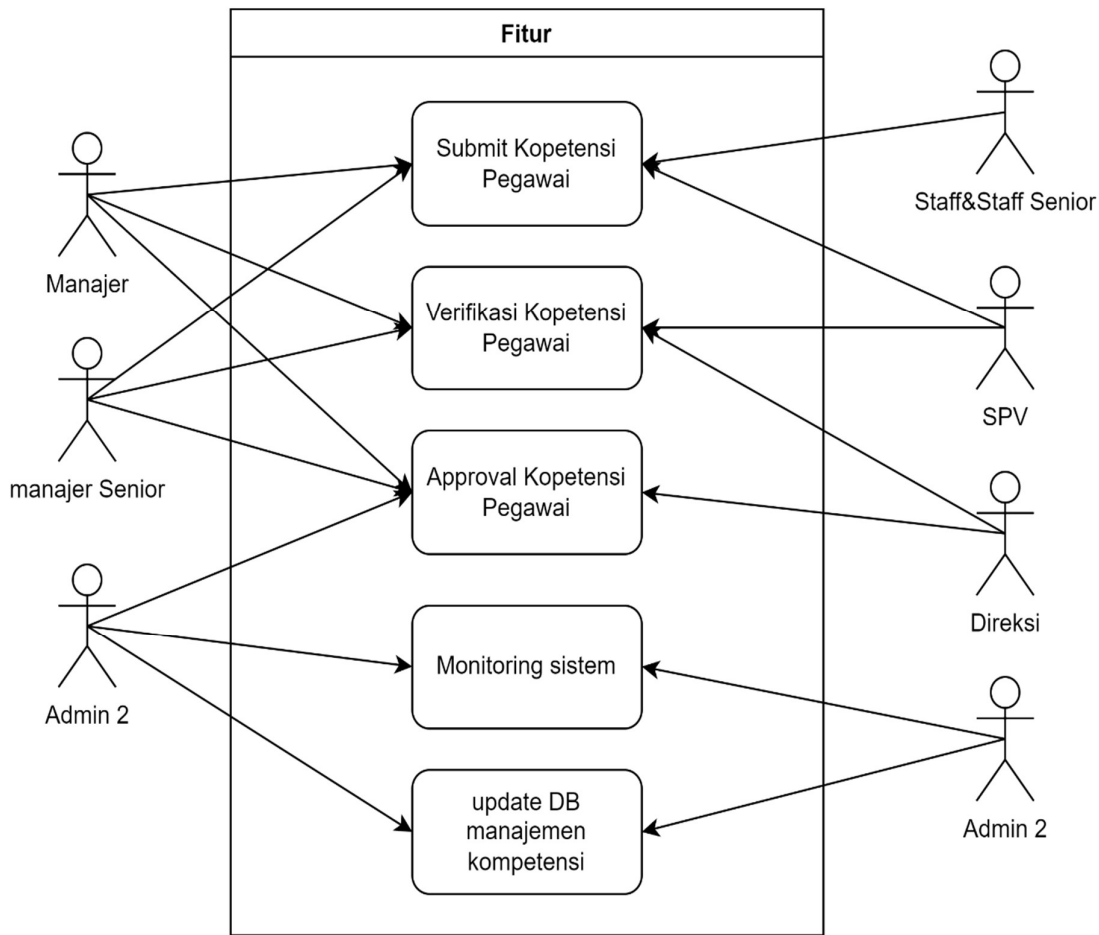
Proses validasi yang dilakukan dapat dilakukan berdasarkan tingkatan pegawai. Verifikator dan approver yang dibuat dalam aplikasi ini terdapat pada matriks dibawah ini:

Table 1 Matrik Approval Aplikasi Kompetensi

User	Verifikator	Approver 1	Approver 2
Staff & senior staff	Supervisor	Manager	Pengembangan SDM
Supervisor	Manajer	Manajer Senior	Pengembangan SDM
Manajer	Manajer Senior	Manajer Senior	Pengembangan SDM
Manajer Senior	Direksi/Pengembangan SDM	Direksi/Pengembangan SDM	Pengembangan SDM

3.1.3 Use Case Diagram

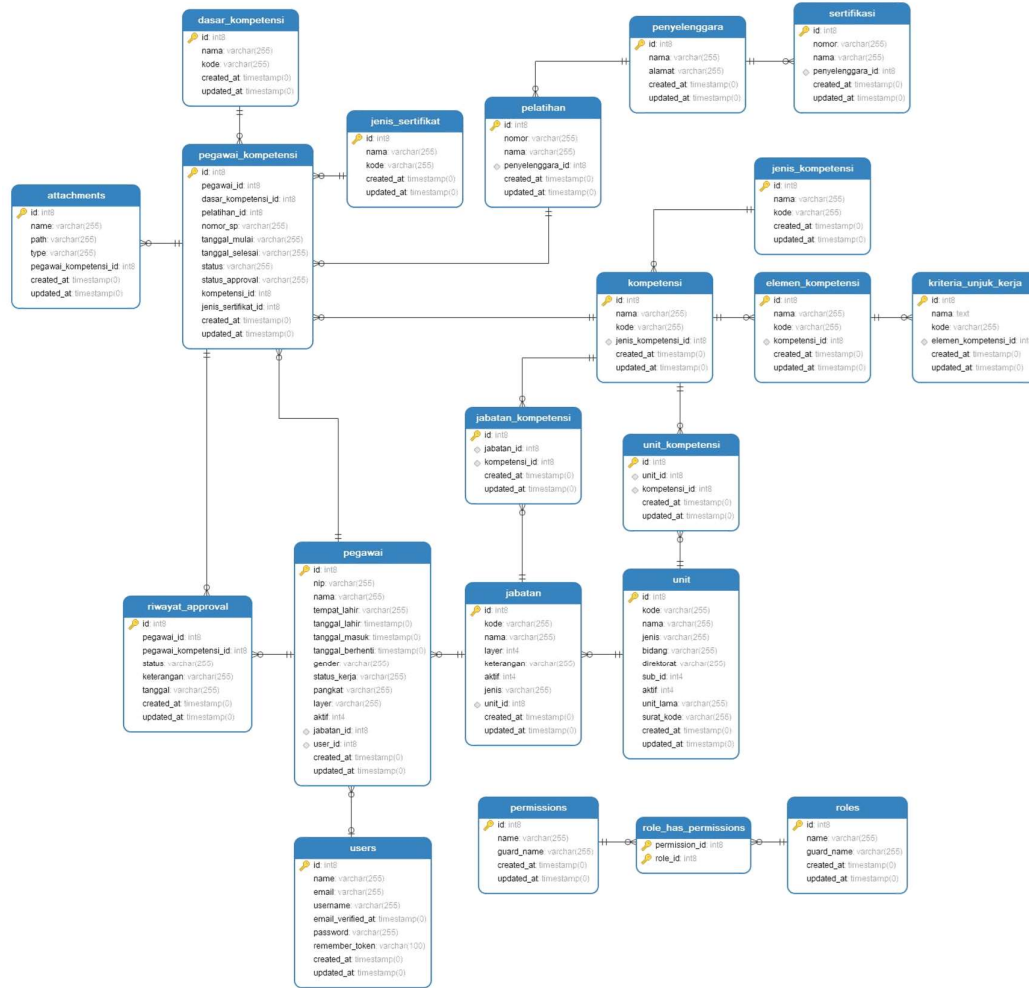
Pengguna yang akan mengoperasikan Aplikasi kompetensi nantinya adalah seluruh pegawai dari PDAM Surya Sembada yang telah dibagi berdasarkan golongan/jabatan. Pihak pengembangan SDM memiliki otoritas untuk melakukan approval dan pengelolaan aplikasi.



Gambar 1 Use Case Diagram Aplikasi Kompetensi

3.1.4 Entity Relation Diagram

Aplikasi kompetensi memiliki beberapa tabel yang digunakan dalam proses operasionalnya. Berikut ini merupakan gambaran dari ER diagram aplikasi kompetensi.



Gambar 2 Entity Relation Digram dari Aplikasi Kompetensi

3.2 Aplikasi eBBM

Aplikasi e-BBM minimum memiliki Tiga fungsi utama yaitu pembuatan dan pengelolaan kupon, penerimaan dan validasi kupon saat transaksi BBM, dan proses pembayaran menggunakan kupon digital. Fungsi-fungsi utama tersebut akan secara detail dijelaskan pada sub-bab dibawah ini.

3.2.1 Pembuatan dan Pengelolaan Kupon Digital

Aplikasi ini memiliki fungsi yang mencakup penciptaan dan pengelolaan kupon digital, memberikan fleksibilitas kepada pengguna untuk mengoptimalkan penggunaan bahan bakar minyak. Pertama-tama, sistem memungkinkan pengguna untuk membuat kupon digital baru dengan mudah. Proses ini melibatkan pengisian informasi seperti nilai kupon, tanggal kadaluarsa, dan jenis

kupon yang ingin dibuat. Dengan fitur ini, perusahaan dapat secara fleksibel menyesuaikan nilai dan parameter kupon sesuai dengan kebijakan dan strategi manajemen biaya yang berlaku.

Aplikasi harus mampu mengelola kupon digital dengan efisien. Ini mencakup beberapa fungsi kunci, seperti kemampuan untuk mengedit informasi kupon, menghapus kupon yang tidak relevan atau telah kadaluarsa, serta melihat detail kupon secara menyeluruh. Fungsi ini memberikan kontrol penuh kepada pengguna dalam memantau dan mengelola inventaris kupon digital perusahaan, memastikan keberlanjutan dan keakuratan data pada setiap transaksi.

Fitur penciptaan dan pengelolaan kupon digital dalam aplikasi ini membantu perusahaan PDAM Surya Sembada dalam mengoptimalkan strategi penggunaan bahan bakar, memberikan fleksibilitas, dan memastikan keteraturan serta kelengkapan dalam pengelolaan kupon-kupon yang ada.

3.2.2 Penerimaan dan Validasi Kupon Digital saat Pembelian BBM

Aplikasi ini memiliki fungsi yang krusial dalam memastikan validitas dan keberlanjutan penggunaan kupon digital dalam pembelian bahan bakar minyak (BBM). Pertama-tama, sistem diwajibkan untuk melakukan validasi kupon digital secara ketat ketika pengguna menukarkannya pada saat pembelian BBM. Proses validasi ini mencakup pemeriksaan terhadap keaslian kupon, memastikan bahwa kupon tersebut belum melewati batas waktu kadaluarsa yang ditetapkan, dan memverifikasi ketersediaan kupon dalam sistem sebelum mengizinkan transaksi pembelian BBM.

Sistem juga harus melakukan pemeriksaan mendalam terkait keaslian kupon, batas waktu kadaluarsa, dan ketersediaan kupon sebelum memberikan izin untuk pembelian BBM. Hal ini bertujuan untuk memastikan bahwa pengguna hanya dapat menggunakan kupon yang masih berlaku dan sesuai dengan ketentuan yang berlaku. Proses pemeriksaan ini mencakup validasi waktu serta ketersediaan kupon di basis data sistem, memastikan bahwa setiap transaksi pembelian BBM sesuai dengan parameter yang telah ditetapkan.

Situasi di mana kupon tidak valid, sistem akan memberikan pesan kesalahan. Langkah ini menjadi penting untuk memberikan informasi yang jelas dan transparan kepada pengguna terkait alasan tidak berhasilnya transaksi. Pesan kesalahan ini dapat mencakup penjelasan mengenai ketidakvalidan kupon, batas waktu kadaluarsa yang telah terlampaui, atau ketersediaan kupon yang telah habis. Dengan demikian, fungsi aplikasi ini tidak hanya memastikan keamanan transaksi, tetapi juga memberikan pengalaman pengguna yang jelas dan transparan. Proses Pembayaran menggunakan Kupon Digital.

3.2.3 Pembayaran Menggunakan Kupon Digital

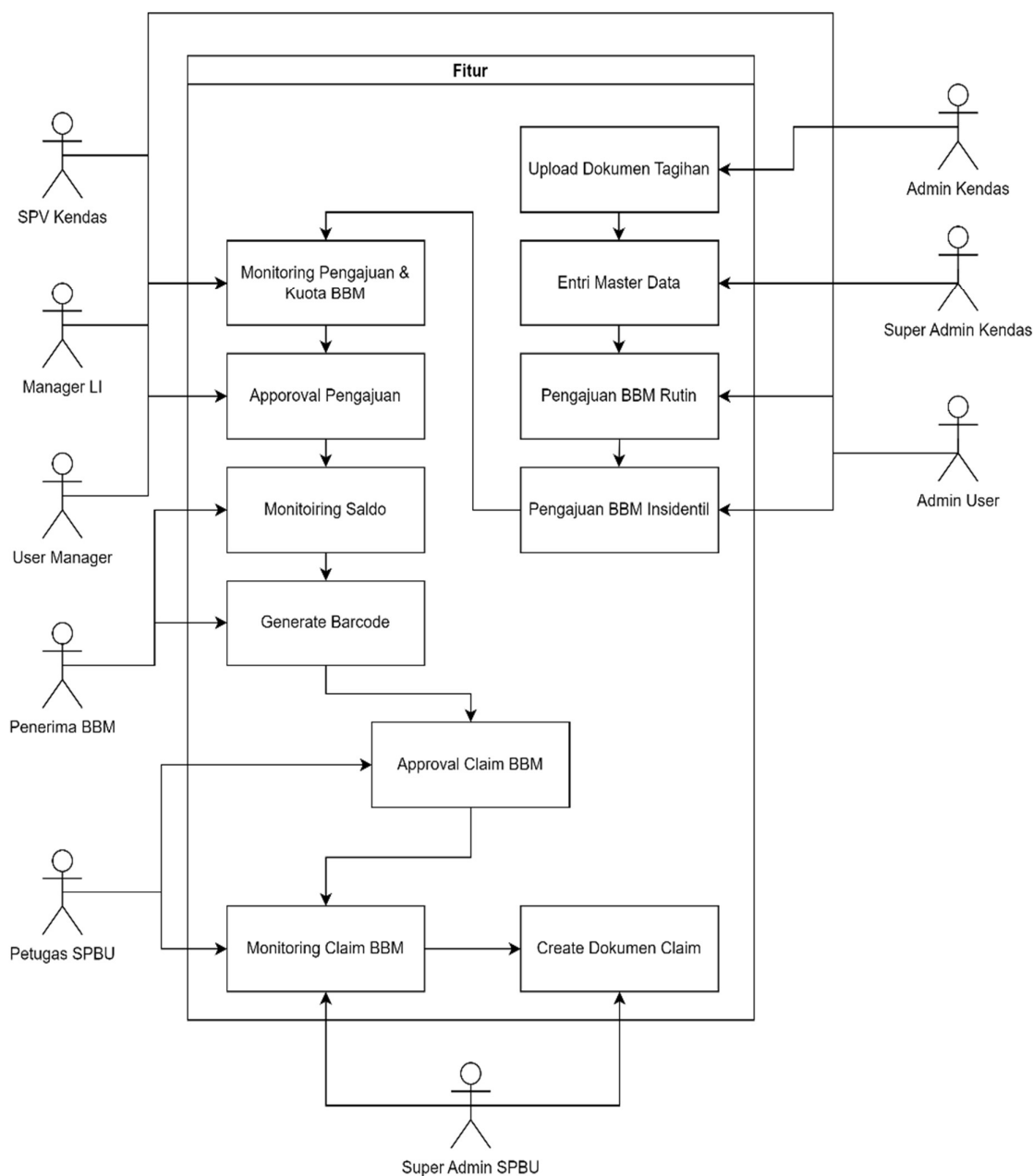
Fitur-fitur pada aplikasi e-bbm ini memiliki peran krusial dalam mendukung efisiensi transaksi pembayaran menggunakan kupon digital serta memastikan keteraturan stok bahan bakar minyak (BBM). Pertama-tama, setelah kupon digital berhasil divalidasi, sistem kemudian akan melanjutkan ke tahap selanjutnya dengan memproses transaksi pembayaran. Sistem akan secara otomatis mengurangi jumlah yang sesuai dari nilai kupon yang digunakan. Proses ini memastikan bahwa transaksi pembayaran berlangsung dengan akurat dan efisien, sesuai dengan nilai yang tertera pada kupon digital yang valid.

Transaksi pembayaran menggunakan kupon digital berhasil dilakukan, aplikasi harus dapat memperbarui stok BBM yang tersedia. Hal ini sangat penting untuk memastikan keteraturan dan akurasi dalam mengelola inventaris BBM perusahaan. Dengan melakukan pembaruan stok secara otomatis setelah setiap transaksi, sistem akan memberikan gambaran yang real-time terkait dengan jumlah BBM yang masih tersedia, membantu perusahaan dalam perencanaan dan pengelolaan persediaan bahan bakar dengan lebih efektif.

Fitur-fitur pada aplikasi e-bbm bertujuan untuk menyederhanakan dan mengotomatisasi proses transaksi pembayaran menggunakan kupon digital, sambil memastikan bahwa stok BBM yang tersedia selalu tercatat dengan akurat. Hal ini tidak hanya meningkatkan efisiensi operasional, tetapi juga memberikan informasi yang penting dalam pengambilan keputusan terkait dengan pengelolaan stok bahan bakar perusahaan.

3.2.4 Use Case Diagram

Pengguna yang akan mengoperasikan Aplikasi BBM nantinya adalah, bagian Layanan Internal dan jajaran Manajer serta rAdmin Bagian yang ada di PDAM, sedangkan karyawan selain yang disebutkan diatas akan menjadi pihak penerima BBM atau user.



Gambar 3 Use Case Diagram

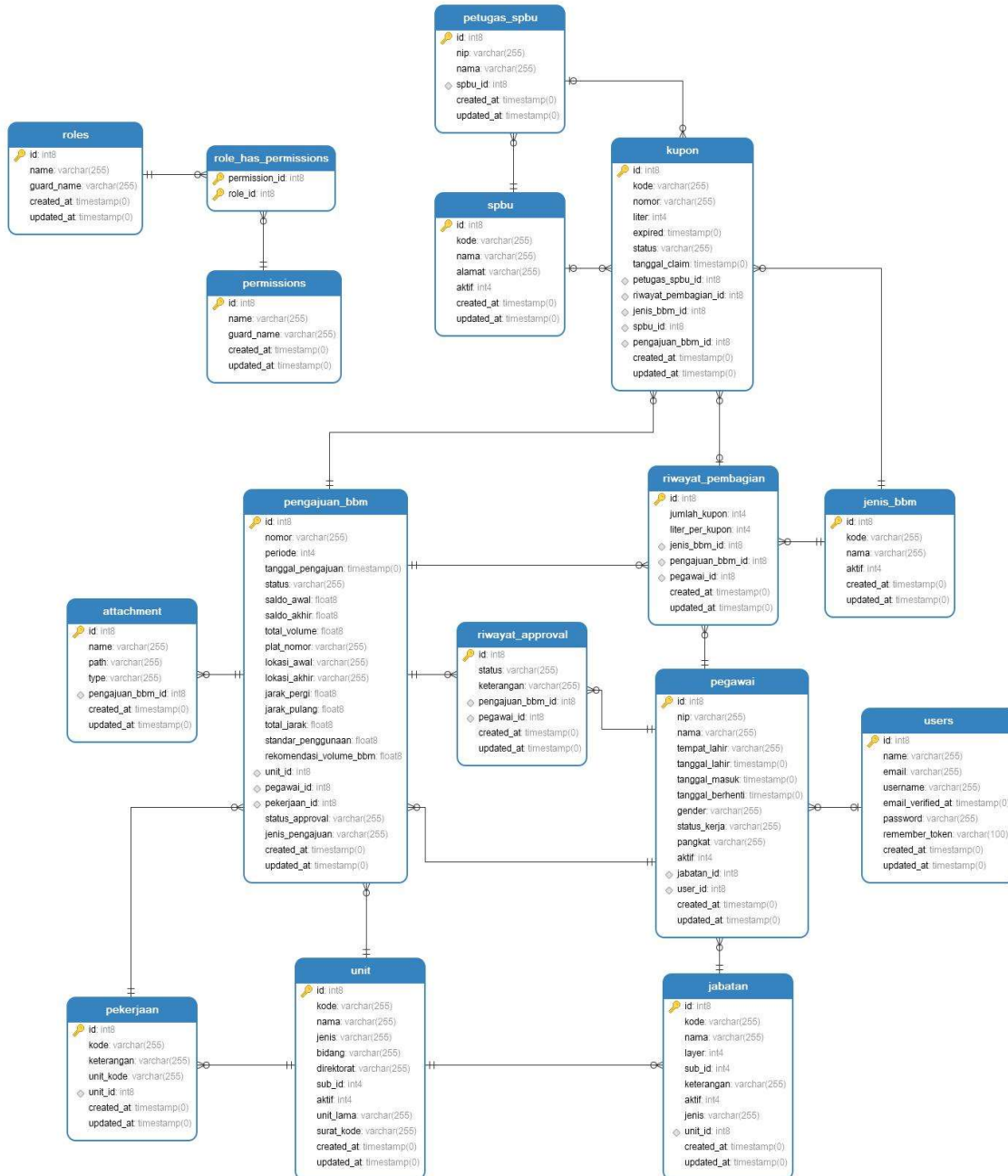
Berikut adalah deskripsi peran masing-masing actor dalam diagram use case Pada Gambar 1 yang disajikan dalam bentuk tabel tertera pada Tabel 1.

Tabel 1 User Role dan Kapabilitas

User Role	Kapabilitas
Super Admin Kendas	- Mengentry master data
Admin Bagian Kendas	- Mengupload dokumen tagihan - Memonitoring Pengajuan & Kuota BBM
SPV Bagian Kendas	- Memonitoring Pengajuan & Kuota BBM - Memberi persetujuan berjenjang pada order BBM yang masuk
Manajer LI	- Memonitoring Pengajuan & Kuota BBM - Memberi persetujuan berjenjang pada order BBM yang masuk
Petugas SPBU	- Approve Claim kupon digital BBM yang diberikan oleh User Penerima BBM - Memonitoring claim kupon digital
Super Admin SPBU	- Memonitoring claim kupon digital - Mencetak dokumen claim untuk melakukan proses Tagihan
Admin Bagian	- Membuat Pengajuan Order BBM Rutin - Membuat Pengajuan Order BBM Insidentil - Monitoring Pengajuan BBM
User Manajer	- Memonitoring order BBM rutin/insidentil serta kuota yang masuk di Aplikasi - Memberi persetujuan berjenjang pada order BBM yang masuk
User Petugas Penerima BBM	- Monitoring Saldo Pribadi - Generate kuota pribadi untuk menjadi Barcode sehingga dapat diklaim

3.2.5 Entity Relation Diagram

Entity Relationship Diagram (ERD) atau diagram hubungan entitas adalah alat visualisasi yang digunakan dalam desain database untuk menggambarkan hubungan antara entitas atau objek dalam sistem. ERD digunakan untuk merepresentasikan data yang akan disimpan dalam database dan cara entitas tersebut berinteraksi satu sama lain. ERD biasanya digunakan oleh pengembang sistem informasi untuk menggambarkan struktur database dan hubungan antara tabel dalam database. Aplikasi eBBM, ERD dapat digunakan untuk menggambarkan hubungan antara entitas-entitas yang terlibat dalam aplikasi E-BBM



Gambar 4 Entity Relation Diagram Aplikasi E-BBM

3.3 Aplikasi Workflow

Aplikasi workflow merupakan aplikasi yang telah ada dan digunakan untuk mengakomodir proses pembuatan surat tugas, pengumuman secara internal maupun eksternal. Aplikasi ini memiliki fitur yang dijelaskan pada sub bab dibawah ini.

3.3.1 Pengelolaan Surat

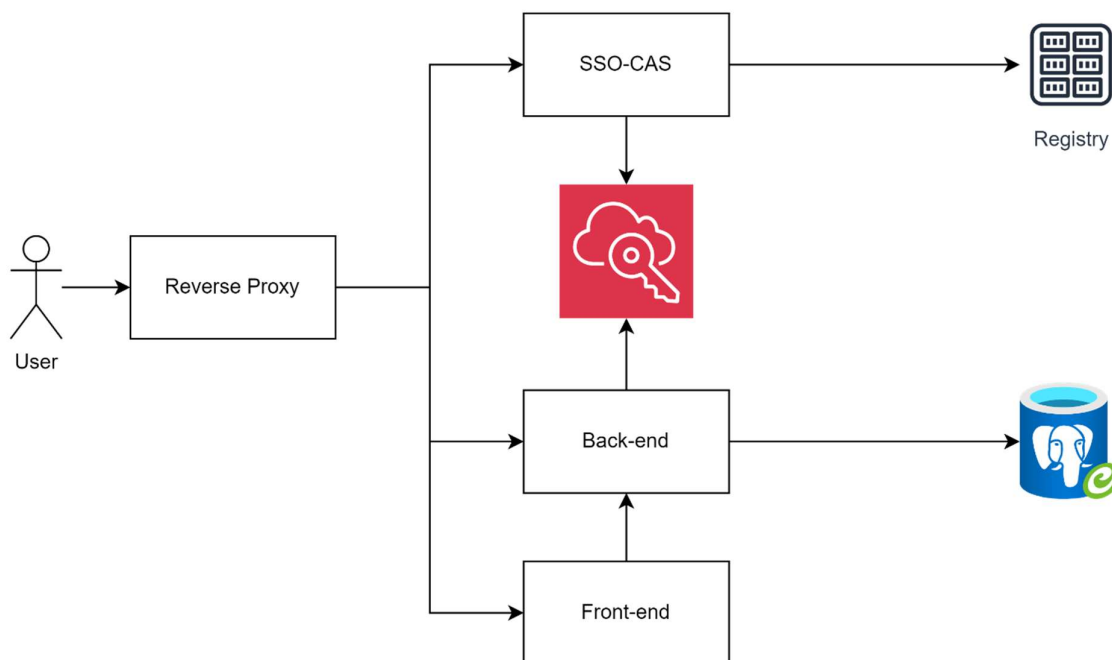
Pengelolaan surat

BAB IV TEKNIS APLIKASI

4.1 Software Architecture

Arsitektur software yang ideal memiliki beberapa komponen penting, termasuk backend, frontend. Implementasi operasional aplikasi SSO pada PDAM Surya Sembada, mengharuskan pengembangan aplikasi saat ini dapat dilakukan integrasi dengan SSO yang ada. Penyesuaian arsitektur aplikasi dengan adanya integrasi SSO dapat dilihat pada gambar dan penjelasan dibawah ini.

1. Backend adalah bagian dari software yang bertanggung jawab untuk memproses data dan logic di sisi server. Backend menyediakan API (Application Programming Interface) untuk digunakan oleh aplikasi klien atau aplikasi frontend. Dalam konteks platform Docker, backend dapat dikemas dalam container Docker untuk memudahkan deploy dan scaling.



Gambar 5 Arsitektur Aplikasi e-BBM

2. Frontend adalah bagian dari software yang bertanggung jawab untuk memproses data dan logic di sisi klien atau browser. Frontend terdiri dari HTML, CSS, dan JavaScript yang digunakan untuk membangun antarmuka pengguna yang interaktif. Dalam konteks platform Docker, frontend dapat dikemas dalam container Docker untuk memudahkan deploy dan scaling. CI/CD juga dapat diterapkan pada pengembangan frontend untuk membantu mengurangi waktu

yang diperlukan untuk membangun, menguji, dan menerapkan perangkat lunak baru.

3. SSO (Single Sign-On) adalah mekanisme otentikasi yang memungkinkan pengguna untuk mengakses berbagai aplikasi menggunakan satu set kredensial. SSO umumnya melibatkan penyedia layanan identitas (LDAP) yang memungkinkan pengguna untuk masuk ke satu aplikasi, dan kemudian mengakses aplikasi lain tanpa perlu memasukkan kredensial tambahan.

4.2 Software Stack

Software stack yang akan dilaksanakan pada aplikasi e-Audit terdapat komponen backend, frontend secara detail sebagai berikut:

1. Backend :

- PHP 8.0
- Laravel 10.0
- PHP Unit Test
- Nginx
- JWT
- SSO CAS
- Mysql/Postgres/Oracle

2. Frontend

- Node JS
- Typescript
- React JS
- Materio Theme
- Cypress Automated Testing
- Mobile App

BAB V PELAKSANAAN PEKERJAAN

5.1 Kickoff Meeting

Kickoff meeting dalam pengembangan aplikasi e-bbm merupakan pertemuan awal proyek ini bertujuan untuk memberikan pemahaman menyeluruh kepada tim pengembangan dan pemangku kepentingan terkait aplikasi e-bbm yang akan diimplementasikan. Pertama-tama, tujuan aplikasi secara rinci dijelaskan, termasuk manfaat yang diharapkan dan dampak positifnya terhadap pengelolaan bahan bakar minyak perusahaan. Lingkup aplikasi, termasuk fitur dan fungsionalitas yang akan dimasukkan, juga didefinisikan dengan jelas untuk memastikan pemahaman yang seragam di antara seluruh tim.



Gambar 6 Proses Kickoff Meeting

Kickoff meeting membahas anggota tim pengembangan aplikasi e-bbm, memperkenalkan peran dan tanggung jawab masing-masing, dan menyusun struktur tim. Ini membantu membangun kerjasama tim dan memastikan setiap anggota tim memiliki pemahaman yang jelas tentang kontribusi mereka dalam mengembangkan aplikasi. Selain itu, waktu dan jadwal pengembangan aplikasi diperkenalkan, mencakup penjelasan tentang milestone utama dan tenggat waktu untuk setiap tahap pengembangan. Ini memberikan pemahaman waktu yang jelas kepada semua pihak terkait, termasuk pemangku kepentingan, tentang kapan aplikasi e-bbm dapat diimplementasikan.



Gambar 7 Proses Presentasi pada Kickoff Meeting

Kickoff meeting membahas risiko dan mitigasi pengembangan aplikasi e-bbm, mengidentifikasi potensi risiko dan merumuskan strategi untuk mengatasinya. Proses komunikasi dan frekuensi pelaporan juga dibahas agar semua pihak tahu cara terbaik untuk berkomunikasi dan melaporkan perkembangan pengembangan aplikasi. Harapan dan ekspektasi pemangku kepentingan diidentifikasi agar dapat memastikan bahwa pemangku kepentingan merasa terlibat dan dapat memberikan masukan yang berharga selama pengembangan aplikasi. Dengan demikian, kickoff meeting membentuk dasar yang kokoh untuk memulai pengembangan aplikasi e-bbm dengan pemahaman yang seragam dan tujuan yang jelas.

5.2 Pembuatan Aplikasi

Proses pengembangan aplikasi e-bbm di PDAM Surya Sembada melibatkan langkah-langkah penting, mulai dari perancangan hingga implementasi. Setelah melalui tahap kickoff meeting dan survei, tim pengembang memulai dengan merancang struktur sistem, menulis kode, dan melakukan uji coba untuk memastikan aplikasi sesuai dengan standar. Berikut adalah langkah-langkah yang dilibatkan:

1. Perencanaan dan Desain

Tim memilih alat pengembangan tertentu untuk backend dan frontend, memastikan bahwa kebutuhan fungsional dan struktur basis data didefinisikan dengan baik. Desain antarmuka pengguna juga diperhatikan untuk memastikan tampilan yang baik.

2. Pengembangan Backend

Setelah perencanaan, langkah selanjutnya adalah mengimplementasikan fungsi-fungsi inti di sisi backend menggunakan alat pengembangan tertentu. Hal

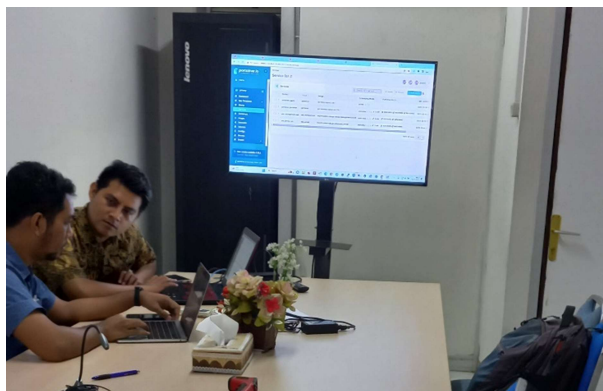
ini melibatkan manajemen basis data dan persiapan API untuk berinteraksi dengan frontend.



Gambar 8 Proses Diskusi Pengembangan

3. Pengembangan Frontend

Tim berfokus pada pengembangan antarmuka pengguna yang interaktif di sisi frontend. Mereka memastikan bahwa desain responsif dan fungsionalitas aplikasi terpenuhi, sambil berkomunikasi dengan backend melalui API yang sudah disiapkan.



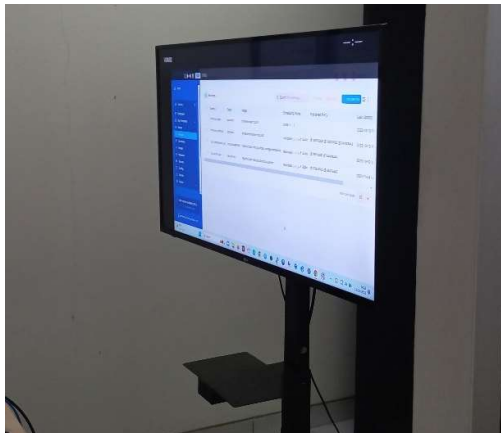
Gambar 9 Proses Diskusi Instalasi

4. Pengembangan Aplikasi Mobile

Setelah menyelesaikan implementasi frontend, tim melanjutkan dengan pengembangan aplikasi mobile untuk memperluas akses pengguna. Mereka memilih teknologi yang sesuai dan mengimplementasikan fitur-fitur khusus untuk perangkat seluler.

5. Pengujian Semua Komponen

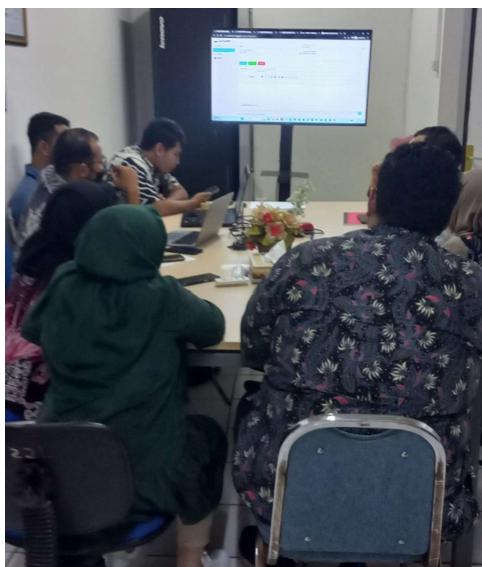
Proses pengujian melibatkan verifikasi fungsi-fungsi utama melalui uji coba otomatis. Ini mencakup pengujian backend, frontend, dan aplikasi mobile untuk memastikan semua komponen berjalan sesuai yang diharapkan.



Gambar 10 Proses Ujicoba Internal

6. Integrasi dan Peluncuran

Setelah tahap pengujian, semua komponen diintegrasikan untuk memastikan keterhubungan yang baik. Manajemen versi menggunakan Git, dan strategi peluncuran disiapkan untuk memastikan peluncuran aplikasi berjalan dengan sukses.



Gambar 11 Proses Dikusi dengan User

Pada akhirnya, pasca-implementasi melibatkan pemeliharaan dan pengembangan lebih lanjut berdasarkan umpan balik pengguna dan perbaikan bug. Kolaborasi yang baik antara tim pengembang, tim pengujian, dan manajemen proyek

menjadi kunci kesuksesan dalam menciptakan aplikasi e-bbm yang berkualitas dan memenuhi harapan pengguna.

5.3 Proses Standarisasi Coding

Proses standarisasi coding merupakan langkah utama dalam pengembangan perangkat lunak yang bertujuan untuk menciptakan konsistensi dalam gaya penulisan dan struktur kode. Langkah awal dalam proses ini adalah penetapan standar coding yang melibatkan pemilihan konvensi dan pedoman tertentu yang akan diikuti oleh seluruh tim pengembang. Ini dapat mencakup aturan terkait penamaan variabel, tata letak kode, indentasi, komentar, dan praktik pengkodean lainnya. Standar-standar ini bertujuan untuk memastikan bahwa setiap bagian kode dapat mudah dibaca, dimengerti, dan dikelola oleh semua anggota tim, bahkan jika proyek tersebut melibatkan pengembang yang berbeda.



Setelah penetapan standar, langkah berikutnya adalah menyosialisasikan dan memberikan pelatihan kepada seluruh tim pengembang untuk memastikan pemahaman dan adopsi standar tersebut. Komunikasi yang efektif mengenai keuntungan dan alasan di balik standar coding juga sangat penting untuk mendapatkan dukungan dan keterlibatan penuh dari anggota tim.

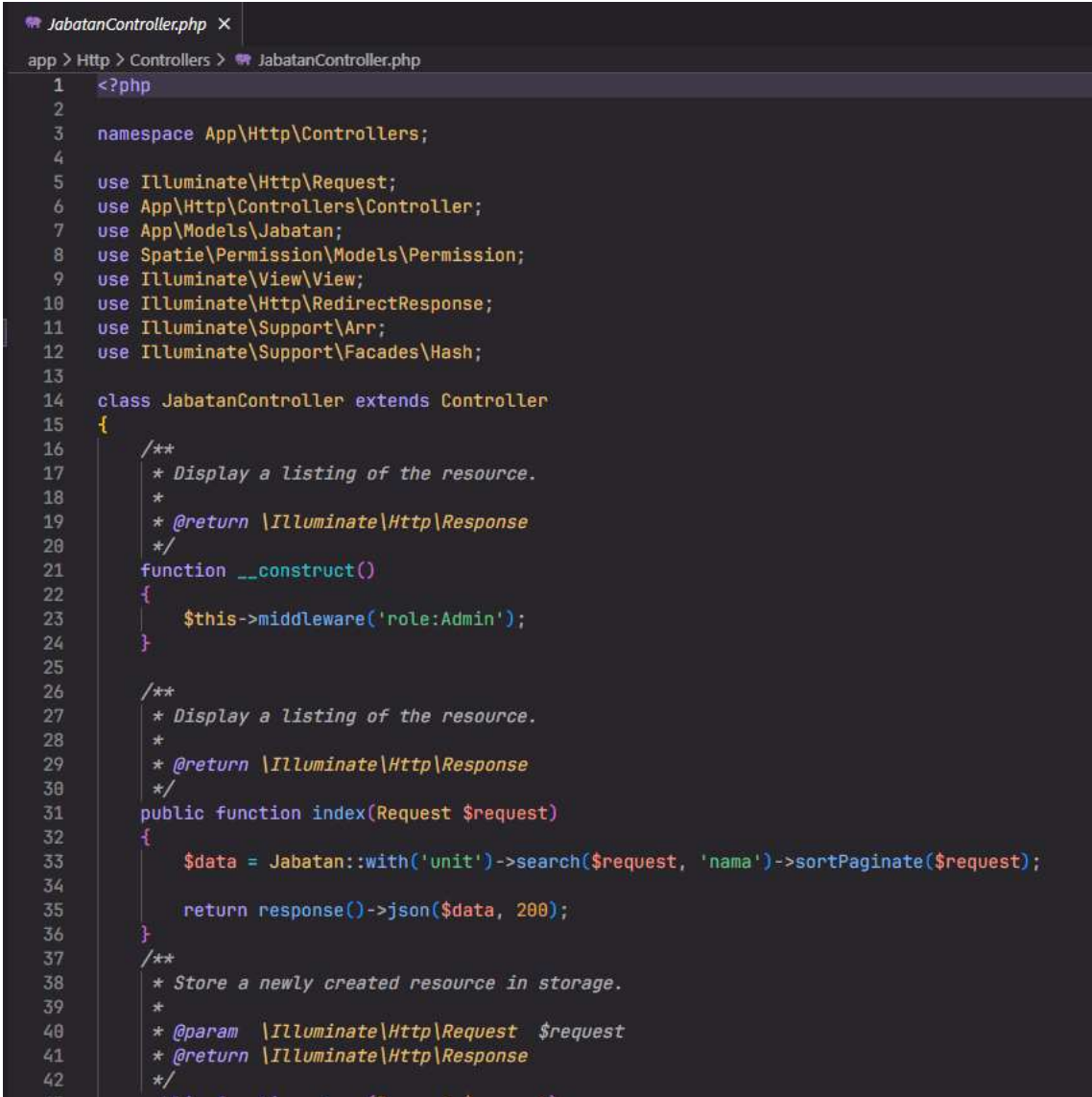
Selama tahap pengembangan, proses standarisasi coding melibatkan penerapan langsung dari standar tersebut pada setiap tahapan proyek. Pemantauan dan peninjauan kode secara berkala dilakukan untuk memastikan kepatuhan terhadap standar dan mengidentifikasi peluang perbaikan atau peningkatan. Beberapa organisasi juga menggunakan alat otomatis untuk memeriksa kode secara konsisten sesuai dengan standar yang telah ditetapkan.

Manfaat dari standarisasi coding mencakup peningkatan kerja sama tim, memudahkan pemeliharaan dan pengembangan berkelanjutan, serta mengurangi potensi kesalahan atau bug karena konsistensi dalam pendekatan pengkodean. Dengan menerapkan proses standarisasi coding secara efektif, sebuah tim pengembang dapat

mencapai tingkat kualitas dan keterbacaan kode yang lebih tinggi, menghasilkan perangkat lunak yang lebih dapat diandalkan dan mudah dikelola.

5.3.1 Proses Standarisasi Coding Aplikasi Kompetensi

Sub bab ini menunjukan tangkapan layar coding yang dibuat untuk aplikasi kompetensi dengan hasil coding sudah terstandar.



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Http\Controllers\Controller;
7 use App\Models\Jabatan;
8 use Spatie\Permission\Models\Permission;
9 use Illuminate\View\View;
10 use Illuminate\Http\RedirectResponse;
11 use Illuminate\Support\Arr;
12 use Illuminate\Support\Facades\Hash;
13
14 class JabatanController extends Controller
15 {
16     /**
17      * Display a listing of the resource.
18      *
19      * @return \Illuminate\Http\Response
20      */
21     function __construct()
22     {
23         $this->middleware('role:Admin');
24     }
25
26     /**
27      * Display a listing of the resource.
28      *
29      * @return \Illuminate\Http\Response
30      */
31     public function index(Request $request)
32     {
33         $data = Jabatan::with('unit')->search($request, 'nama')->sortPaginate($request);
34
35         return response()->json($data, 200);
36     }
37
38     /**
39      * Store a newly created resource in storage.
40      *
41      * @param \Illuminate\Http\Request $request
42      * @return \Illuminate\Http\Response
43      */
```

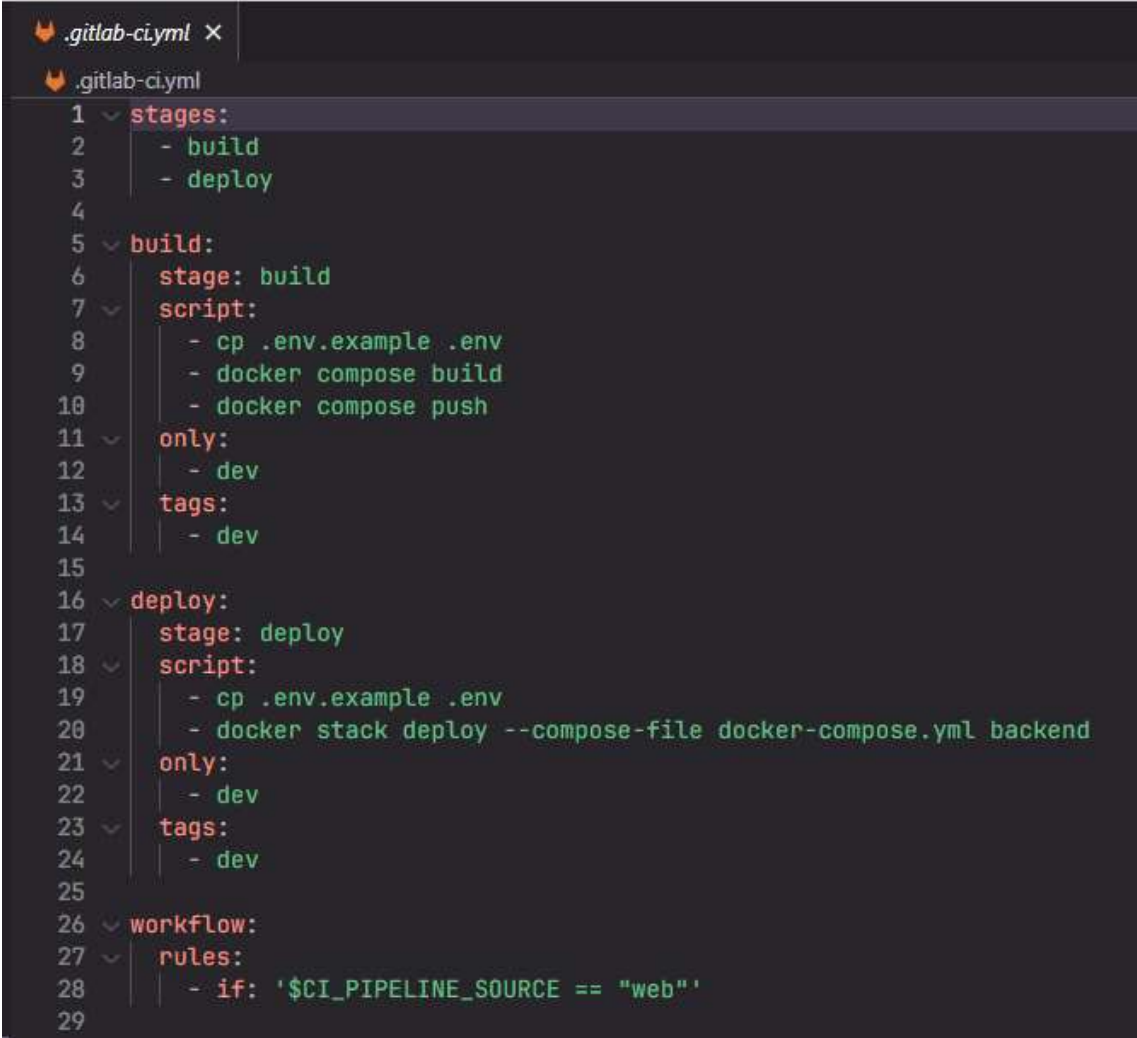
Gambar 12 Controller Coding yang Sudah Standarisasi

```
JabatanTest.php X
tests > Unit > JabatanTest.php
1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6  use App\Models\Jabatan;
7  use App\Models\User;
8  use Illuminate\Foundation\Testing\WithFaker;
9  use Illuminate\Foundation\Testing\RefreshDatabase;
10
11 class JabatanTest extends TestCase
12 {
13     use RefreshDatabase;
14
15     /** @test */
16     public function it_can_get_all_jabatan()
17     {
18         $user = User::factory()->create()->assignRole('Admin');
19         $jabatan = Jabatan::factory()->count(3)->create();
20
21         $response = $this->actingAs($user)
22             ->getJson('/api/jabatan');
23
24         $response->assertStatus(200)
25             ->assertJsonCount(3, 'data');
26     }
27
28     /** @test */
29     public function it_can_create_a_jabatan()
30     {
31         $user = User::factory()->create()->assignRole('Admin');
32         $jabatan = Jabatan::factory()->make();
33
34         $response = $this->actingAs($user)->postJson('/api/jabatan', $jabatan->toArray());
35
36         $response->assertStatus(201)
37             ->assertJson(['data' => ['nama' => $jabatan->nama]]);
38
39         $this->assertDatabaseHas('jabatan', [
40             'nama' => $jabatan->nama,
41         ]);
42     }
43 }
```

Gambar 13 Unit Test Coding yang Sudah Standarisasi

```
docker-compose.yml x
docker-compose.yml
1  version: "3.8"
2  services:
3    nginx:
4      build:
5        context: .
6        dockerfile: ./deployment/docker/nginx.Dockerfile
7      image: registry.pdam-sby.go.id/ekompetensi:nginx
8      depends_on:
9        - php
10     ports:
11       - "5002:80"
12     networks:
13       - default
14     deploy:
15       replicas: 3
16       resources:
17         limits:
18           cpus: '0.50'
19           memory: 50M
20         reservations:
21           cpus: '0.25'
22           memory: 30M
23
24     php:
25       build:
26         context: .
27         dockerfile: ./deployment/docker/php.Dockerfile
28       image: registry.pdam-sby.go.id/ekompetensi:php
29       working_dir: /app
30       env_file: .env
31       networks:
32         - default
33       extra_hosts:
34         - "host.docker.internal:host-gateway"
35       deploy:
36         replicas: 3
37         resources:
38           limits:
39             cpus: '1'
40             memory: 4GB
41           reservations:
42             cpus: '0.25'
43             memory: 512M
```

Gambar 14 Docker Compose Coding yang Sudah Standarisasi



```
1 stages:
2   - build
3   - deploy
4
5 build:
6   stage: build
7   script:
8     - cp .env.example .env
9     - docker compose build
10    - docker compose push
11  only:
12    - dev
13  tags:
14    - dev
15
16 deploy:
17   stage: deploy
18   script:
19     - cp .env.example .env
20     - docker stack deploy --compose-file docker-compose.yml backend
21  only:
22    - dev
23  tags:
24    - dev
25
26 workflow:
27   rules:
28     - if: '$CI_PIPELINE_SOURCE == "web"'
29
```

Gambar 15 Git atau CICD Command Coding yang Sudah Standarisasi

5.3.2 Proses Standarisasi Coding Aplikasi eBBM

Proses yang sama dilakukan juga pada aplikasi e-BBM. Proses standarisasi coding menghasilkan coding yang dapat dilihat pada gambar-gambar dibawah ini

Controller Kupon

```
KuponController.php X
app > Http > Controllers > KuponController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Http\Controllers\Controller;
7  use App\Models\Kupon;
8  use App\Models\PetugasSpu;
9  use Spatie\Permission\Models\Permission;
10 use Illuminate\View\View;
11 use Illuminate\Http\RedirectResponse;
12 use Illuminate\Support\Arr;
13 use Illuminate\Support\Facades\Hash;
14 use Carbon\Carbon;
15
16 class KuponController extends Controller
17 {
18     /**
19      * Display a listing of the resource.
20      *
21      * @return \Illuminate\Http\Response
22      */
23     function __construct()
24     {
25         $this->middleware('role:Admin');
26     }
27
28     /**
29      * Display a listing of the resource.
30      *
31      * @return \Illuminate\Http\Response
32      */
33     public function index(Request $request)
34     {
35         $data = Kupon::with('spbu', 'jenis_bbm', 'riwayat_pembagian.pegawai')
36             ->when($request->get('pengajuan_bbm_id'), function($query) use ($request) {
37                 $query->where('pengajuan_bbm_id', $request->get('pengajuan_bbm_id'));
38             })
39             ->when($request->get('pegawai_id'), function($query) use ($request) {
40                 $query->whereHas('riwayat_pembagian', function($sql) use ($request) {
41                     $sql->where('pegawai_id', $request->get('pegawai_id'));
42                 });
43             })
44             ->orderBy('status', 'desc')
45             ->orderBy('expired', 'asc')
46             ->search($request, 'nama')->sortPaginate($request);
```

Gambar 16 Controller Coding eBBM yang Sudah Standarisasi

Unit test kupon

```
KuponTest.php x
tests > Unit > KuponTest.php
1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6  use App\Models\Kupon;
7  use App\Models\User;
8  use Illuminate\Foundation\Testing\WithFaker;
9  use Illuminate\Foundation\Testing\RefreshDatabase;
10
11 class KuponTest extends TestCase
12 {
13     use RefreshDatabase;
14
15     /** @test */
16     public function it_can_get_all_kupon()
17     {
18         $user = User::factory()->create()->assignRole('Admin');
19         $kupon = Kupon::factory()->count(3)->create();
20
21         $response = $this->actingAs($user)
22             ->getJson('/api/kupon');
23
24         $response->assertStatus(200)
25             ->assertJsonCount(3, 'data');
26     }
27
28     /** @test */
29     public function it_can_create_a_kupon()
30     {
31         $user = User::factory()->create()->assignRole('Admin');
32         $kupon = Kupon::factory()->make();
33
34         $response = $this->actingAs($user)->postJson('/api/kupon', $kupon->toArray());
35
36         $response->assertStatus(201)
37             ->assertJson(['data' => ['nomor' => $kupon->nomor]]);
38
39         $this->assertDatabaseHas('kupon', [
40             'nomor' => $kupon->nomor,
41         ]);
42     }
43 }
```

Gambar 17 Unit Test Kupon eBBM Coding yang Sudah Standarisasi

```
.gitlab-ci.yml x
.gitlab-ci.yml
1  stages:
2    - build
3    - deploy
4
5  build:
6    stage: build
7    script:
8      - cp .env.example .env
9      - docker compose build
10     - docker compose push
11    only:
12      - dev
13    tags:
14      - dev
15
16  deploy:
17    stage: deploy
18    script:
19      - cp .env.example .env
20      - docker stack deploy --compose-file docker-compose.yml backend_bbm
21    only:
22      - dev
23    tags:
24      - dev
25
26  workflow:
27    rules:
28      - if: '$CI_PIPELINE_SOURCE == "web"'
29
```

Gambar 18 GIT CICD eBBM Coding yang Sudah Standarisasi

```
docker-compose.yml X
docker-compose.yml
1  version: "3.8"
2  services:
3    nginx:
4      build:
5        context: .
6        dockerfile: ./deployment/docker/nginx.Dockerfile
7      image: registry.pdam-sby.go.id/ebbm:nginx
8      depends_on:
9        - php
10     ports:
11       - "5003:80"
12     networks:
13       - default
14     deploy:
15       replicas: 3
16       resources:
17         limits:
18           cpus: '0.50'
19           memory: 50M
20         reservations:
21           cpus: '0.25'
22           memory: 30M
23
24     php:
25       build:
26         context: .
27         dockerfile: ./deployment/docker/php.Dockerfile
28       image: registry.pdam-sby.go.id/ebbm:php
29       working_dir: /app
30       env_file: .env
31       networks:
32         - default
33       extra_hosts:
34         - "host.docker.internal:host-gateway"
35       deploy:
36         replicas: 3
37         resources:
38           limits:
39             cpus: '1'
40             memory: 4GB
41           reservations:
42             cpus: '0.25'
43             memory: 512M
```

Gambar 19 Coding Docker Compose yang Sudah Standarisasi

5.3.3 Proses Implementasi e-Sign pada Workflow

```
public function cekpeg($id)
{
    $cek = HRIS_pegawai::where('NIP',$id)->first();
    $nik = HRIS_personal::find($cek->personal_id);
    return $nik->nik;
}
```

Gambar 20 Function Cekpeg

```
public function suratkeluarCetakDev($recid)
{
    if(strpos($recid,'.pdf') !== false){
        $recid = str_replace('.pdf','', $recid);
    }

    $data = Suratkeluar::with('jabatan')
        ->with(['approval' => function($query) {
            $query->with('role', 'jabatan')->orderBy('order_apprv', 'asc');
        }])
        ->with(['tembusan' => function($query) {
            $query->with('jabatan')->orderBy(DB::raw("lpad(jab_tembusan,3,'0')"), 'asc');
        }])
        ->find($recid);

    $direktori_local = '/pdf';
    $nama_file = (empty($nama_file)) ? $direktori_local.'/surat_'.slugify($data->recid_suratkeluar).'pdf' : $nama_file;
    $path = public_path().$nama_file;
    if(!File::exists($path)) {
        $cetak = self::dataSuratCetak($data);
        $pdf = $cetak['pdf'];
        $cek = $pdf::Output(public_path($nama_file),'I');
        return true;
    }else{
        return redirect(public_path($nama_file));
    }
}
```

Gambar 21 Function SuratkeluarCetak

```
//Format surat cetak
public function dataSuratCetak($data,$status=null)
{
    $ttd = '';
    $n = 0;
    $ttd = [' ', ' ', ' ', ' '];
    $info_ttd = [];
    $qr_info = '';

    $data->status = (!empty($status) ? $status : $data->status);

    foreach ($data->approval as $row) {
        $nama = $row->role->nama;
        $nip = $row->nip_apprv;
        $jabatan = $row->jabatan->namajabatan;

        $splh = find_pelimpahan_dev($nip,$row->jabatan->recidjabatan);
        $tipe_pelimpahan = '';
        if ($splh) {
            foreach ($splh as $idx => $val){
                $tipe_pelimpahan = $val->tipepelimpahan.' ' . $tipe_pelimpahan;
                $jabatan = $tipe_pelimpahan.$row->jabatan->namajabatan;
                $nama = $val->penerima->nama;
                $nip = $val->penerima->nip;
            }
        }

        if(!in_array(trim($nip),['DIRUT','DIRYAN','DIROP','DIRKEU'])){
            $sket_nip = '<br><b>NIP. ' . substr($nip, 0, 1) . ' . ' . substr($nip, 1, 2) . ' . ' . substr($nip, 3, 5).'</b>';
            $nipku = substr($nip, 0, 1) . ' . ' . substr($nip, 1, 2) . ' . ' . substr($nip, 3, 5);
        } else {
            $sket_nip = '';
            $nipku = '';
        }
    }
}
```

```
$sket_apprv = '';
if($data->jenis == 'Internal'){
    $sket_apprv = $row->ket_apprv;
} else {
    if ($row->jabatan->leveljab == 0) {
        $sket_apprv = '<br><table border="0" nobr="true"><tr><td>DIREKSI PERUSAHAAN DAERAH AIR MINUM<br>SURYA SEMBADA KOTA SURABAYA,</td></tr></table>';
    } elseif ($row->jabatan->leveljab == 1) {
        $sket_apprv = '<br><table border="0" nobr="true"><tr><td><b>a.n.</b> DIREKSI PERUSAHAAN DAERAH AIR MINUM<br>SURYA SEMBADA KOTA SURABAYA,</td></tr></table>';
    } elseif ($row->jabatan->leveljab == 2) {
        $sket_apprv = '<br><table border="0"><tr><td><b>a.n.</b> DIREKSI PERUSAHAAN DAERAH AIR MINUM<br>SURYA SEMBADA KOTA SURABAYA<br>'. $row->jabatan->atasan->namajabatan.',<br><b>u.b</b></td></tr></table>';
    }
}
}
```

```

$ceknips = Auth::user()->userid;
$ceknip = substr($ceknips, 0, 1) . '.' . substr($ceknips, 1, 2) . '.' . substr($ceknips, 3, 5);
$path = public_path('pdf/'.$nipku.'_'. $data->recid_suratkeluar.'.png');
if(!File::exists($path)) {
    $imgku = 'img/ttde-pdam.png'.'" style="width:100px;">';
} else {
    $imgku = 'img src="'.public_path('pdf/'.$nipku.'_'. $data->recid_suratkeluar.'.png').'" style="width:500px;">';
}
if($data->status == 10){
    if($data->language == 'en'){
        if(trim($jabatan) == 'Direktur Utama'){
            $jabatan = '<i>President Director of</i> PDAM Surya Sembada Kota Surabaya';
            // $jabatan .= '<br>Direktur Utama PDAM Surya Sembada Kota Surabaya';
        }
        $tt[$n] = '
        <table cellpadding="5px" cellspacing="0" style="width:100%;padding-top:10px;border-radius:20px;">
        <tr nobr="true">
        <td width="20%" v-align="middle" align="center" style="padding-top:10px;padding-bottom:10px">
        
        </td>
        <td width="3%" align="left" style="font-size:8pt;"></td>
        <td width="77%" align="left" style="font-size:8pt;">This document has been electronically signed by:
        <br><b>'. $nama.'</b>
        <br>'. $jabatan.'
        </td>
        </tr>
        </table>
        ';
    } else {
        $tt[$n] = $ket_apprv . '<br><br><table cellpadding="5px" cellspacing="0" style="width:100%;padding-top:10px;border-radius:20px;">
        <tr nobr="true">
        <td width="20%" v-align="middle" align="center" style="padding-top:10px;padding-bottom:10px">
        
        </td>
        <td width="3%" align="left" style="font-size:8pt;"></td>
        <td width="77%" align="left" style="font-size:8pt;">Telah ditandatangani secara elektronik oleh:
        <br><b>'. $jabatan.'</b>
        <br><b>'. $nama.'</b>
        <br><b>NIP. ' . $nipku.'</b>
        </td>
        </tr>
        </table>
        ';
    }
} else {
    if($data->jenis == 'Eksternal'){
        if($data->language == 'en'){
            if(trim($jabatan) == 'Direktur Utama'){
                $jabatan = '<i>President Director of</i> PDAM Surya Sembada Kota Surabaya';
            }
            $tt[$n] = '
            <br><b>'. $nama.'</b>
            <br><table border="0"><tr><td height="80px">'.trim($tipe_pelimpahan.$jabatan).'</td></tr></table>
            ';
        } else {
            $tt[$n] = $ket_apprv . '
            <br><table border="0"><tr><td height="80px">'.trim($tipe_pelimpahan.$jabatan).'</td></tr></table>
            <br><b>'. $nama.'</b>
            ';
        }
    }
    if(in_array(trim($nip),['DIRUT','DIRYAN','DIROP','DIRKEU'])){
        $tt[$n] .= '<br><b>NIP. ' . substr($nip, 0, 1) . '.' . substr($nip, 1, 2) . '.' . substr($nip, 3, 5).'</b>';
    }
} else {

```

```

    $tt[$n] = $ket_aprv . '<br><br><table cellpadding="5px" cellspacing="0" style="width:100%;padding-top:10px;border-radi
    ">
        <tr nobr="true">
            <td width="20%" v-align="middle" align="center" style="padding-top:10px;padding-bottom:10px">
                '. $imgku.'
            </td>
            <td width="3%" align="left" style="font-size:8pt;"></td>
            <td width="77%" align="left" style="font-size:8pt;">Telah ditandatangani secara elektronik oleh:
            <br><b>'. $jabatan.'</b>
            <br><b>'. $nama.'</b>
            <br><b>NIP. '. $nipku.'</b>
            </td>
        </tr>
    </table>
    ';
}

$infottd[$n] = [
    'nama' => $nama,
    'nip' => substr($nip, 0, 1) . '.' . substr($nip, 1, 2) . '.' . substr($nip, 3, 5),
    'jabatan' => $jabatan
];

$qr_info .= 'Atasan '. ($n+1). '%0A';
$qr_info .= 'Nama : '. $row->role->nama. '%0A';
$qr_info .= 'NIP : '. substr($nip, 0, 1). '.' . substr($nip, 1, 2). '.' . substr($nip, 3, 5). '%0A';
$qr_info .= 'Jabatan : '. $jabatan. '%0A';
$n++;
}

```

```

if(count($data->approval) == 1){
    if($data->language == 'en'){
        $ttd .= '<br><table border="0"><tr><td style="height:80px"></td></tr></table><table style="width:100%;">
            <tr>
            <td>'. $tt[0] . '</td>
            </tr>';
    } else {
        $ttd .= '<table style="width:100%;">
            <tr>
            <td>'. $tt[0] . '</td>
            </tr>';
    }
}elseif (count($data->approval) < 3) {
    $ttd .= '<table style="width:100%;">
        <tr>
        <td style="width: 46%;text-align: center;">
            ' . $tt[1] . '
        </td>
        <td style="width: 8%;"></td>
        <td style="width: 46%;text-align: center;">
            ' . $tt[0] . '
        </td>
        </tr>';
}

```



```
if (count($data->approval) == 3) {
    $ttd .= '<table style="width:100%;">
        <tr>
            <td style="width: 45%;text-align: center;">
                ' . $tt[1] . '
            </td>
            <td style="width: 10%;"></td>
            <td style="width: 45%;text-align: center;">
                ' . $tt[0] . '
            </td>
        </tr>
        <tr>
            <td colspan="3" style="text-align: center;">
                ' . $tt[2] . '
            </td>
        </tr>';
} elseif (count($data->approval) > 3) {
    $ttd .= '<table>
        <tr>
            <td style="width: 45%;text-align: center;">
                ' . $tt[2] . '
            </td>
            <td style="width: 10%;"></td>
            <td colspan="2" style="width: 45%;text-align: center">
                ' . $tt[3] . '
            </td>
        </tr>';
}
$ttd .= '</table>';
```

```
$tembusan = '';
if(count($data->tembusan) > 0){
    $tembusan .= '<br><br><table style="width:100%" >';
    $tembusan .= '<tr><td colspan="2">Tembusan :</td></tr>';
    if(count($data->tembusan) == 1){
        $tembusan .= '<tr>';
        if($data->jenis == 'Internal'){
            //cek plh
            $jabatan = $data->tembusan[0]->jabatan->namajabatan;
            $plh = find_pelimpahan($data->tembusan[0]->nip_tembusan,$data->tembusan[0]->jabatan->recidjabatan,['Plt','Pjs']);
            if ($plh) {
                $jabatan = $plh->tipepelimpahan.' ' . $plh->pemberi->namajabatan;
            }
            $tembusan .= '<td colspan="2">' . $jabatan . '</td>';
        } else {
            $tembusan .= '<td colspan="2">' . $data->tembusan[0]->tembusan_nama . '</td>';
        }
        $tembusan .= '</tr>';
    } else {
        foreach ($data->tembusan as $key => $row){
            $tembusan .= '<tr><td style="width:3.5%">' . ($key+1) . '&nbsp;</td>';
            if($data->jenis == 'Internal'){
                //cek plh
                $jabatan = $row->jabatan->namajabatan;
                $plh = find_pelimpahan($row->nip_tembusan,$row->jabatan->recidjabatan,['Plt','Pjs']);
                if ($plh) {
                    $jabatan = $plh->tipepelimpahan.' ' . $plh->pemberi->namajabatan;
                }
                $tembusan .= '<td style="width:96.5%">' . $jabatan . '</td>';
            } else {
                $tembusan .= '<td style="width:96.5%">' . $row->tembusan_nama . '</td>';
            }
            $tembusan .= '</tr>';
        }
    }
}
```

```
$pdf::SetAutoPageBreak(TRUE, 8);
$pdf::setImageScale(PDF_IMAGE_SCALE_RATIO);
// CONTE-----
$tagvs = [
    'ol' => [
        ['h'=>1.15, ],
        ['h'=>1.15, ],
    ],
    'li' => [
        ['h'=>1.15, ],
        ['h'=>1.15, ],
    ],
    'div' => [
        ['h'=>1.15, ],
        ['h'=>1.15, ],
    ],
    'p' => [
        ['h'=>1.15, ],
        ['h'=>1.15, ],
    ],
];
$pdf::setHtmlVSpace($tagvs);
$pdf::setListIndentWidth(5);
$pdf::AddPage('P', 'A4');
$pdf::SetFont('helvetica', '', $data->fontsize);
$view = 'PDF.suratkeluar-dev'.(!empty($data->language) ? '-'.$data->language : '');
$pdf::writeHTML(view($view, ['data' => $data, 'ttd' => $ttd, 'disp' => '', 'tembusan' => $tembusan]), true, false, true, false, '');

if($data->jenis == 'Eksternal'){
    $pdf::Image(public_path('img/footer-eksternal.png'), 10, 279, 190, 10, '', '', 'T', false, 0, '', false, false, 1, false, false);
}

return ['pdf'=>$pdf, 'info'=>$info_ttd];
}
```

Gambar 22 Function dataSuratCetak

```
public function approvalLink() {
    return view('v2.approval');
}
```

Gambar 23 Approval Link

```
function approveSurat(recid){
    Swal.fire({
        title: "Apakah Anda yakin?",
        text: "Masukkan Key ID untuk approve (menyetujui).",
        icon: "warning",
        input: "text",
        inputLabel: "Key ID",
        inputPlaceholder: "Masukkan Key ID",
        inputValue: "12341234",
        showCancelButton: true,
        confirmButtonColor: "#DD6B55",
        confirmButtonText: "Ya, approve!",
        cancelButtonText: "Tidak",
        closeOnConfirm: false
    },
```

Gambar 24 View Approval

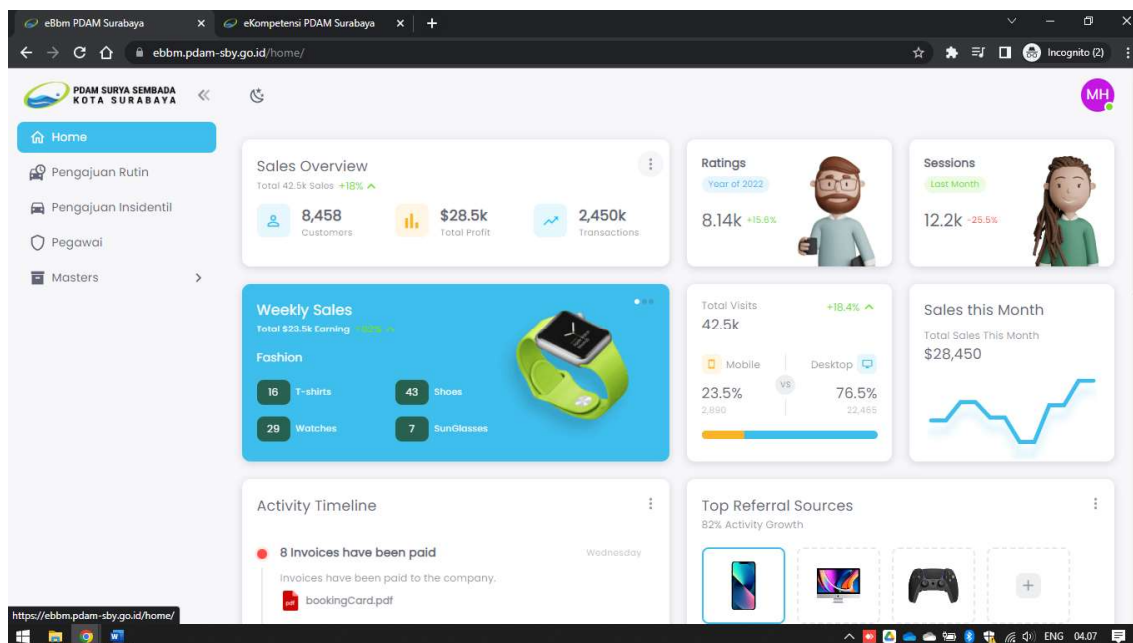
```
).then((result) => {
    if (result.isConfirmed) {
        console.log(result);
        if (result.value == "") {
            approveSurat(recid);
        }else{
            return $.ajax({
                url: "{{route('v2:SK-approve2')}}",
                type: "POST",
                data: {
                    '_token' : '{{csrf_token()}}',
                    'recid' : recid,
                    'keyid' : result.value,
                },
                success: function(result){
                    if(result['sukses'] == 1){
                        Swal.fire("Sukses!", "Surat berhasil diapprove", "success");
                    }
                    else{
                        Swal.fire("Gagal!", "Surat gagal diapprove", "error");
                    }
                    $('#tabel').DataTable().ajax.reload();
                }
            });
        }
    }
});
```

Gambar 25 Function Suratkeluar

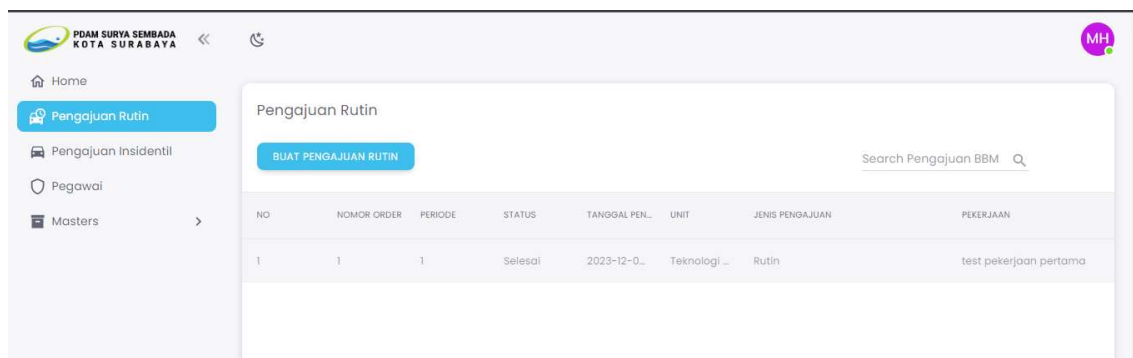
5.4 Hasil Aplikasi

Pada Sub-bab berikut merupakan tangkapan layar dari aplikasi yang telah dibuat dengan keterangan pada masing-masing gambar.

5.4.1 Aplikasi E-BBM



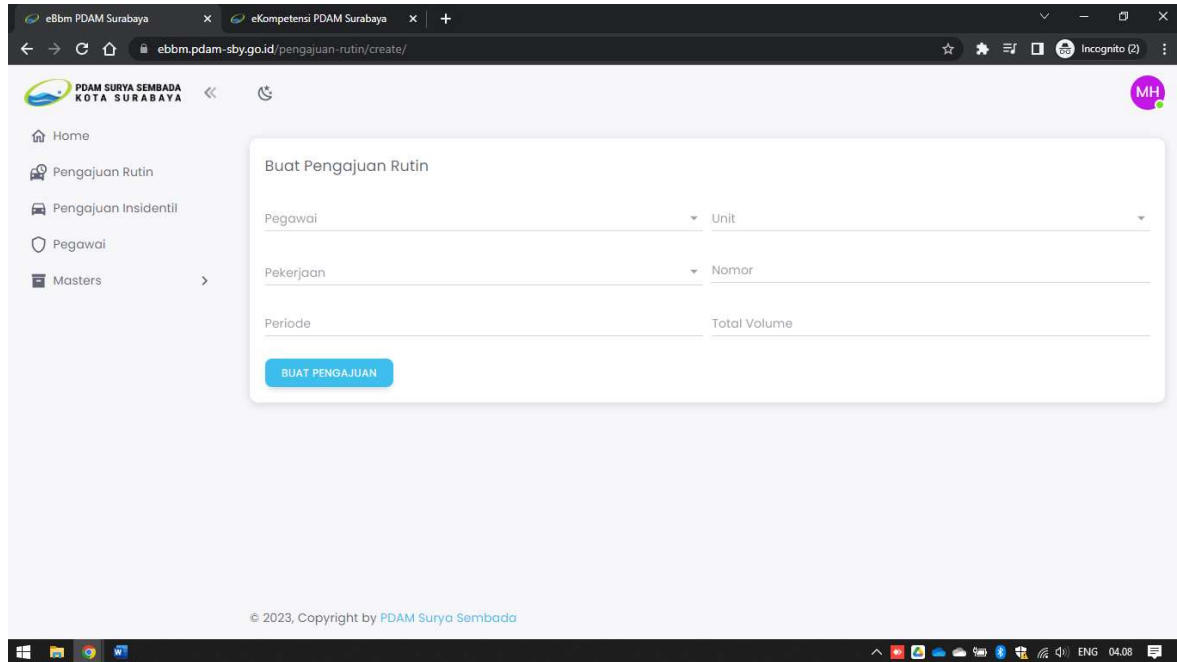
Gambar 26 Dashboard Aplikasi eBBM



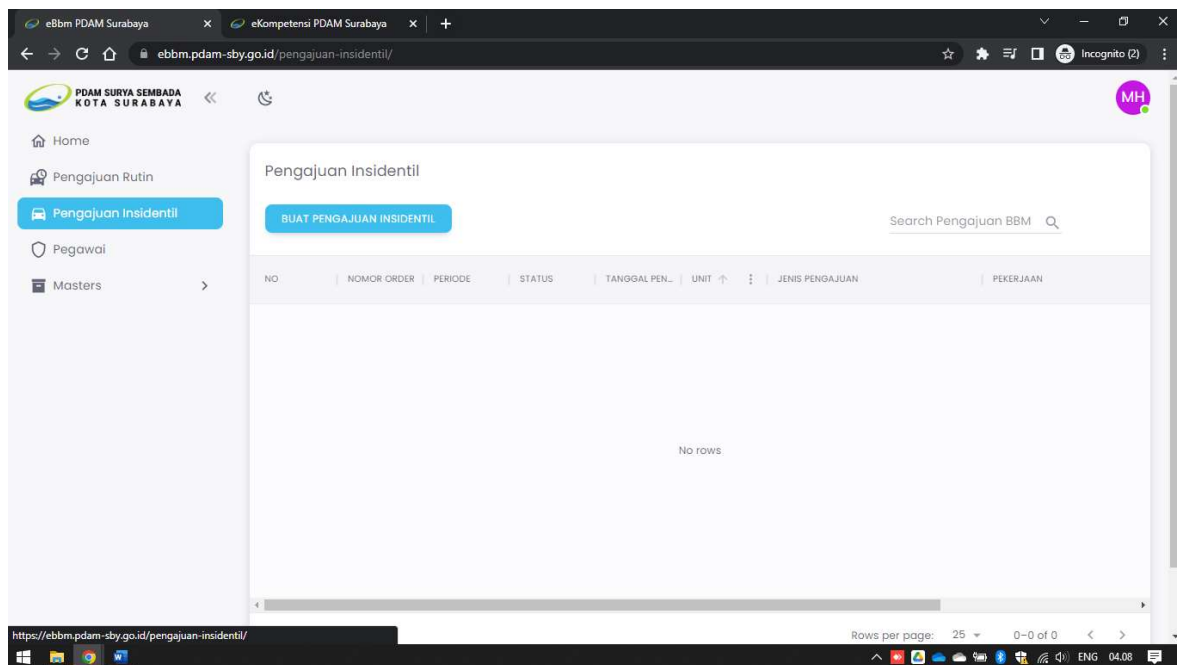
The screenshot shows the 'Pengajuan Rutin' (Regular Submission) menu. It includes a sidebar menu with options: Home, Pengajuan Rutin, Pengajuan Insidentil, Pegawai, and Masters. The main content area features a 'BUAT PENGAJUAN RUTIN' button and a search bar labeled 'Search Pengajuan BBM'. Below the search bar is a table with the following data:

NO	NOMOR ORDER	PERIODE	STATUS	TANGGAL PEN...	UNIT	JENIS PENGAJUAN	PEKERJAAN
1	1	1	Selesai	2023-12-0...	Teknologi ...	Rutin	test pekerjaan pertama

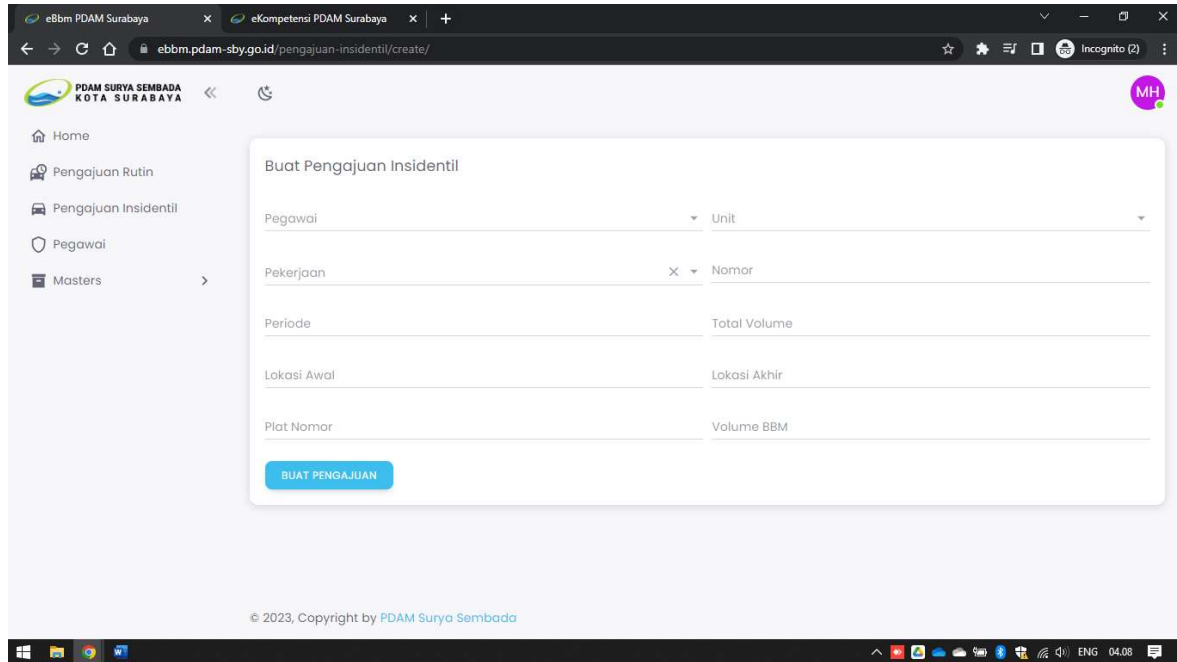
Gambar 27 Tampilan Menu Pengajuan Rutin



Gambar 28 Tampilan Menu Buat Pengajuan Rutin



Gambar 29 Tampilan Menu Pengajuan Insidentil



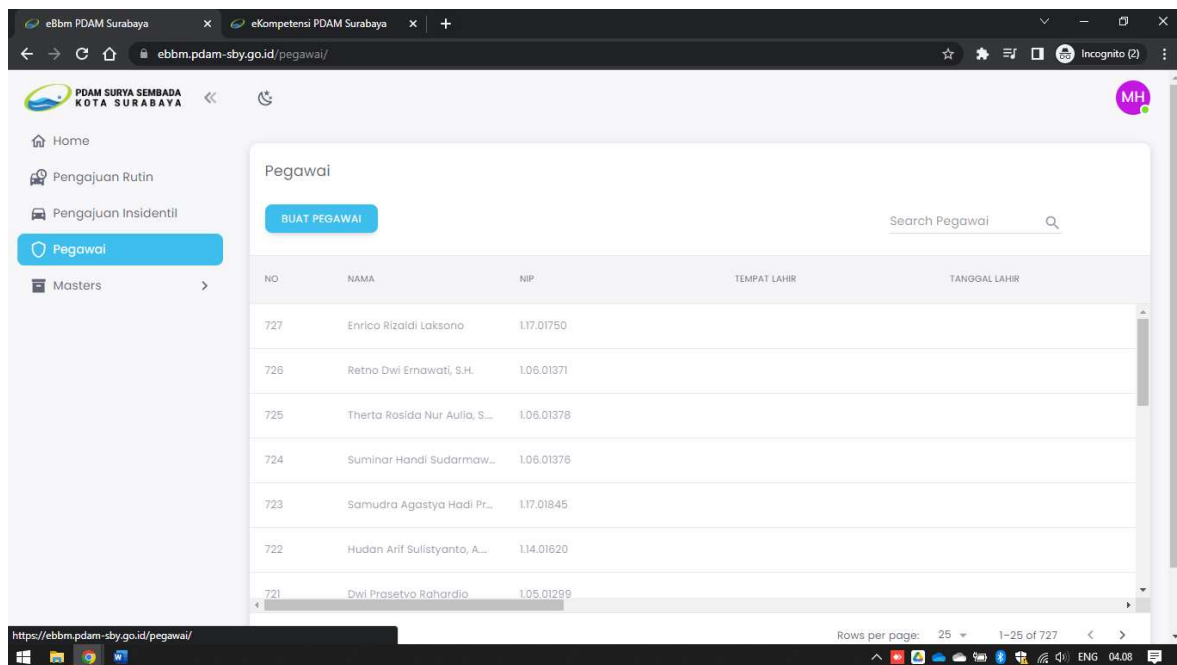
Buat Pengajuan Insidentil

Pegawai	Unit
Pekerjaan	Nomor
Periode	Total Volume
Lokasi Awal	Lokasi Akhir
Plat Nomor	Volume BBM

BUAT PENGAJUAN

© 2023, Copyright by PDAM Surya Sembada

Gambar 30 Tampilan Menu Buat Baru Pengajuan Insidentil



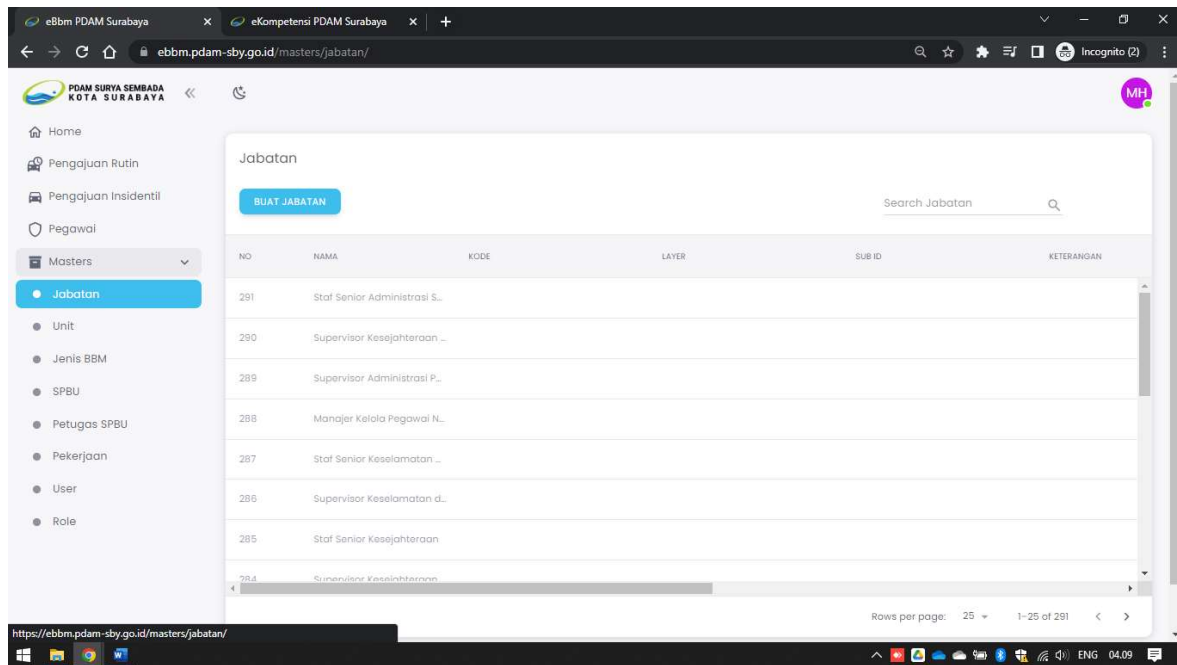
Pegawai

BUAT PEGAWAI Search Pegawai

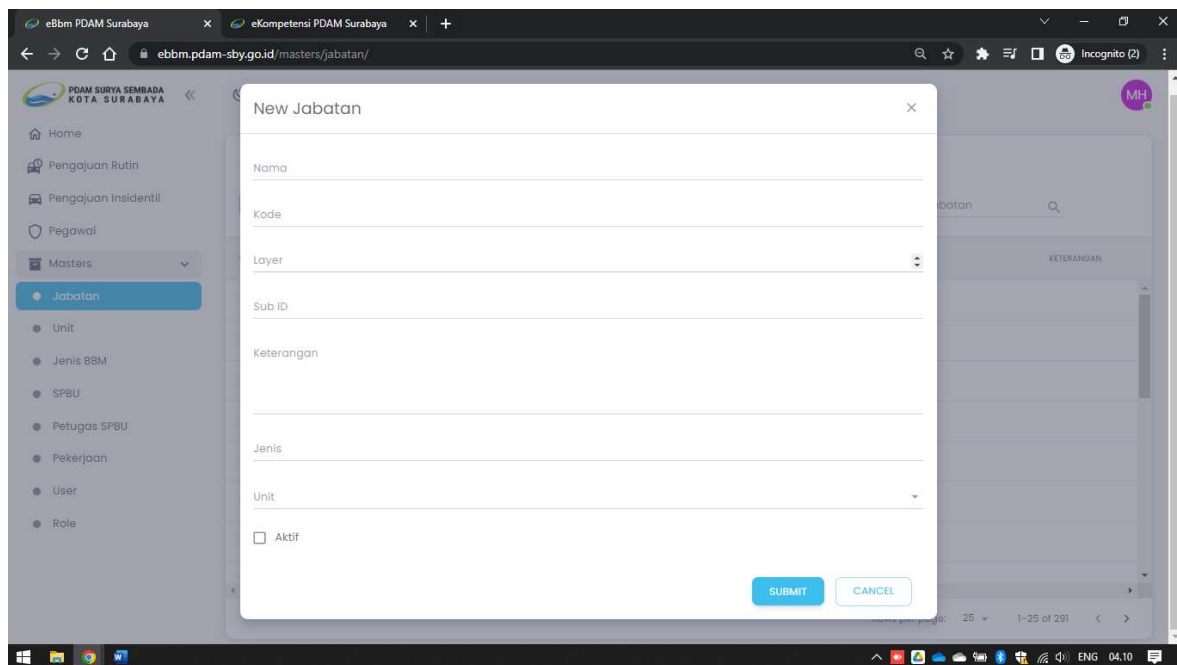
NO	NAMA	NIP	TEMPAT LAHIR	TANGGAL LAHIR
727	Enrico Rizaldi Laksono	1.17.01750		
726	Retno Dwi Ernawati, S.H.	1.06.01371		
725	Therta Rosida Nur Aulia, S...	1.06.01378		
724	Suminar Handi Sudarmaw...	1.06.01378		
723	Samudra Agastya Hadi Pr...	1.17.01845		
722	Hudan Arif Sulistyanto, A...	1.14.01620		
721	Dwi Prasetyo Rahardjo	1.05.01299		

Rows per page: 25 1-25 of 727

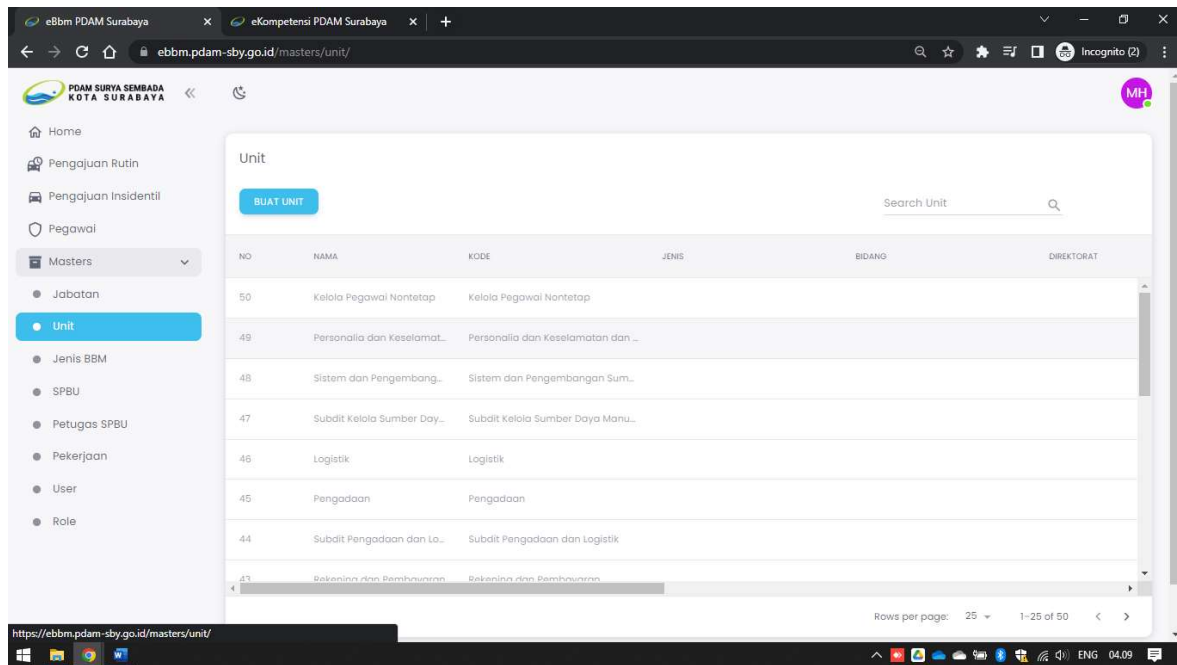
Gambar 31 Tampilan Menu Pegawai



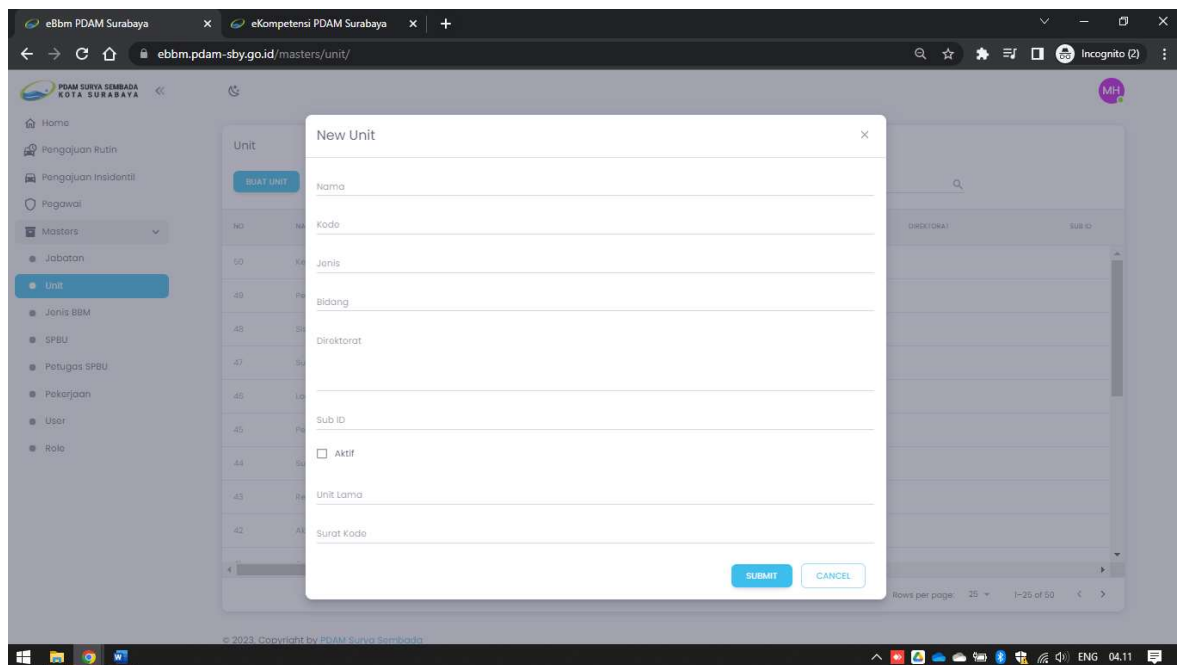
Gambar 32 Tampilan Menu Master Jabatan



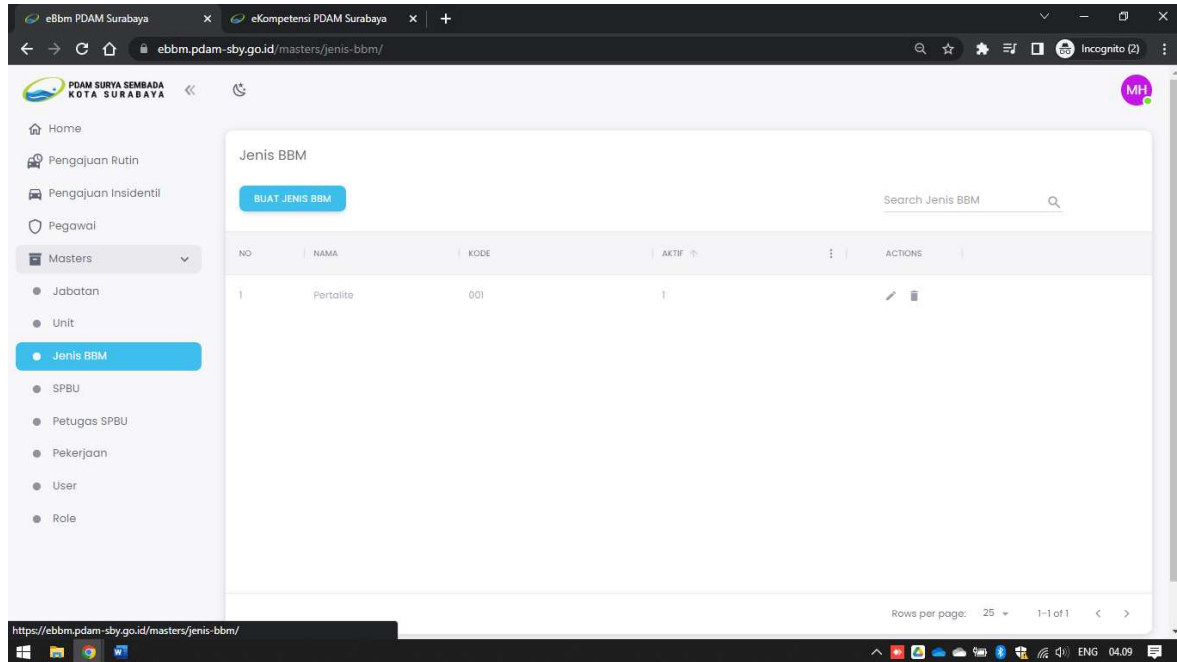
Gambar 33 Tampilan Popup Buat Master Jabatan



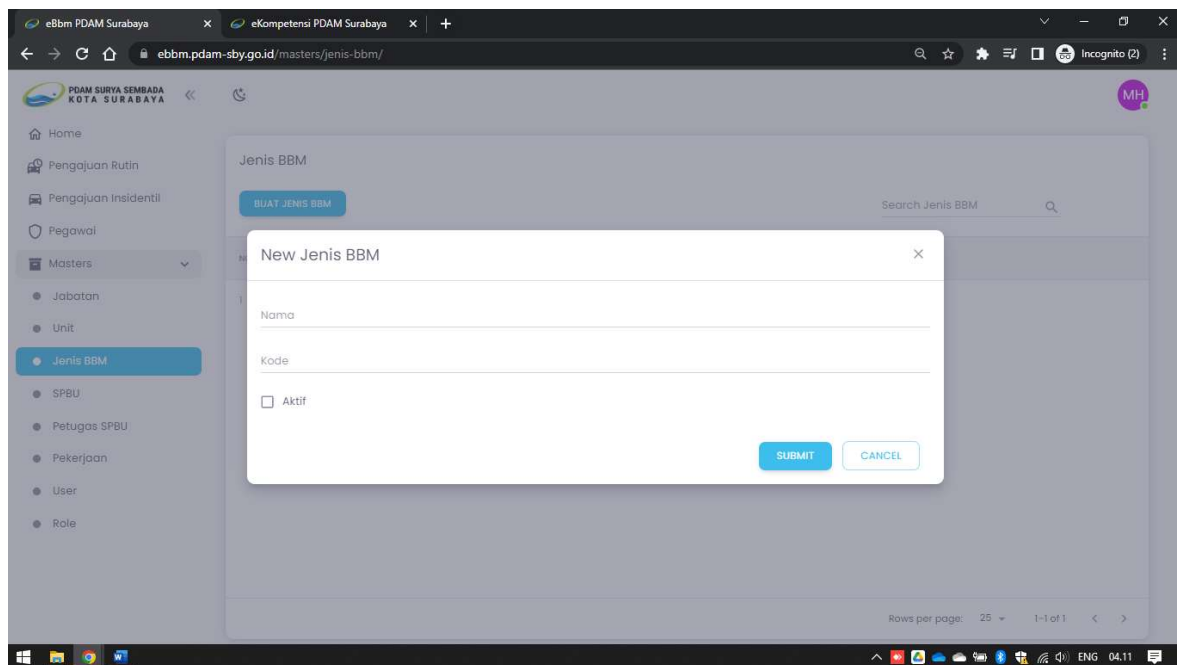
Gambar 34 Tampilan Menu Master Unit



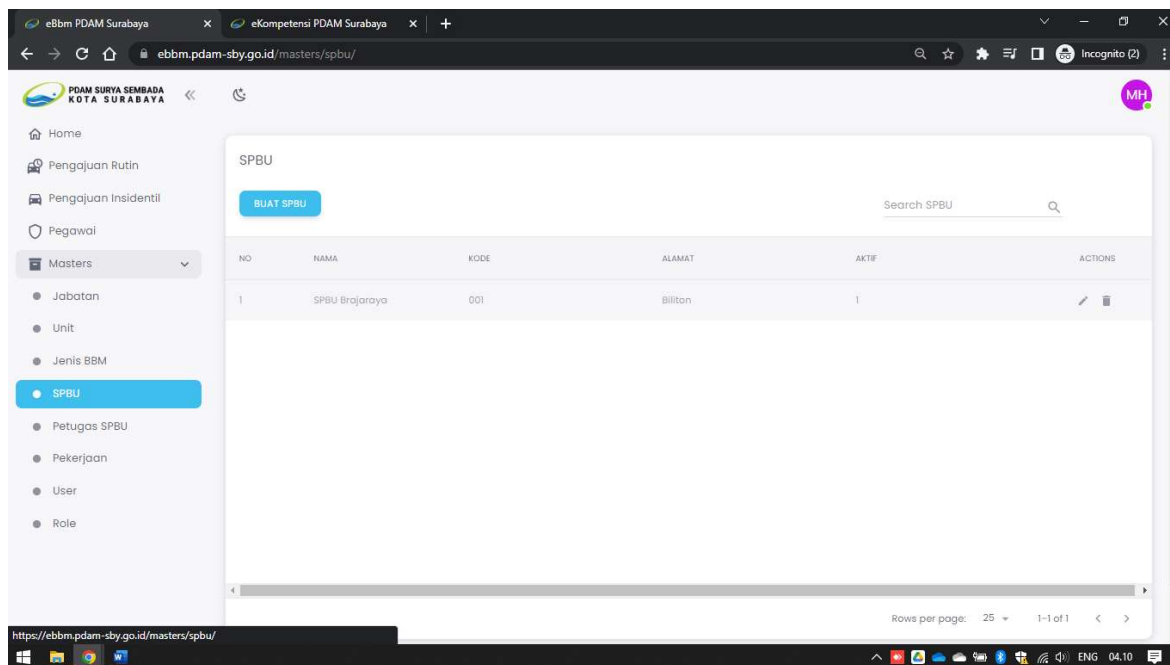
Gambar 35 Tampilan Popup Buat Master Unit



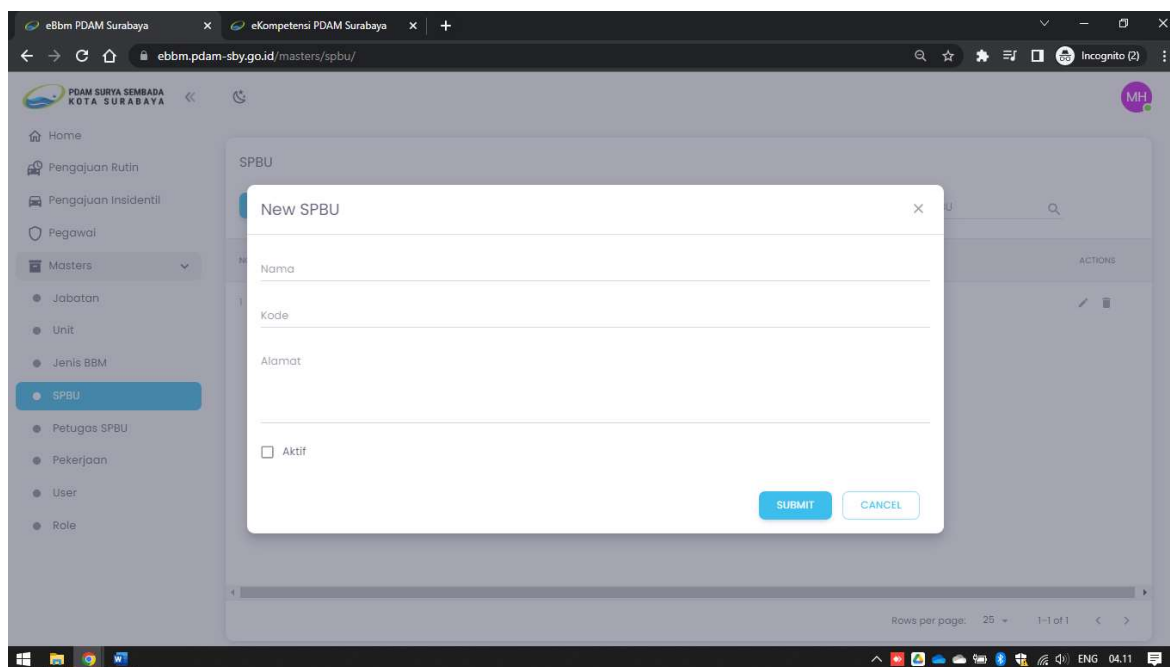
Gambar 36 Tampilan Menu Master Jenis BBM



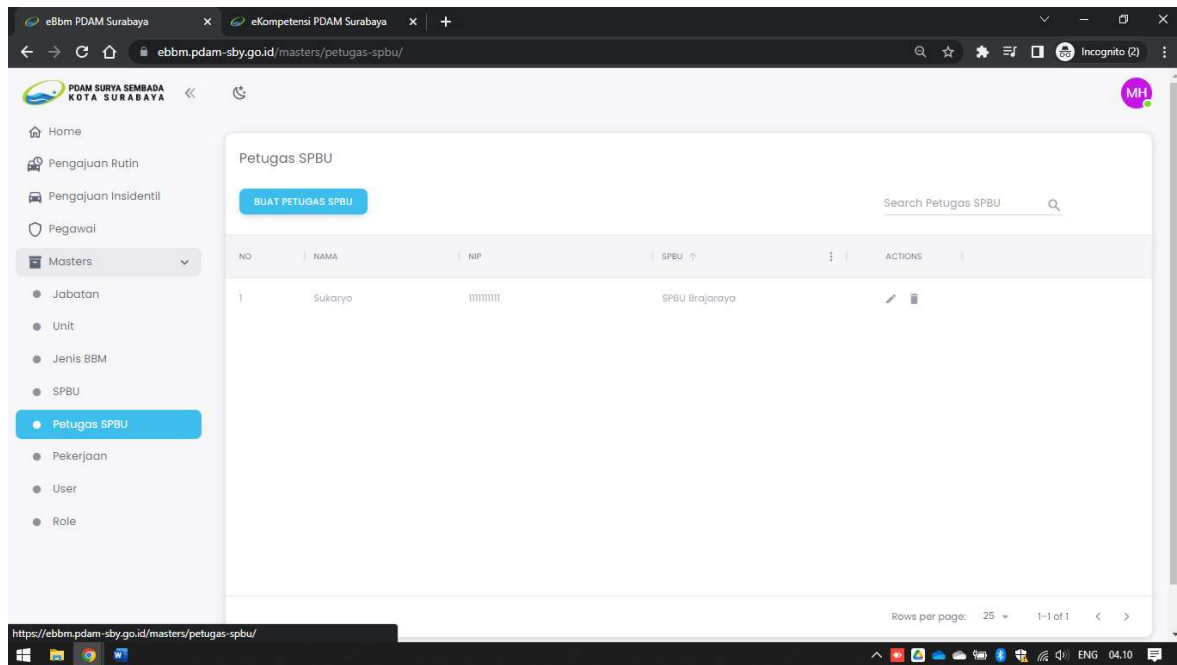
Gambar 37 Tampilan Popup Buat Master Jenis BBM



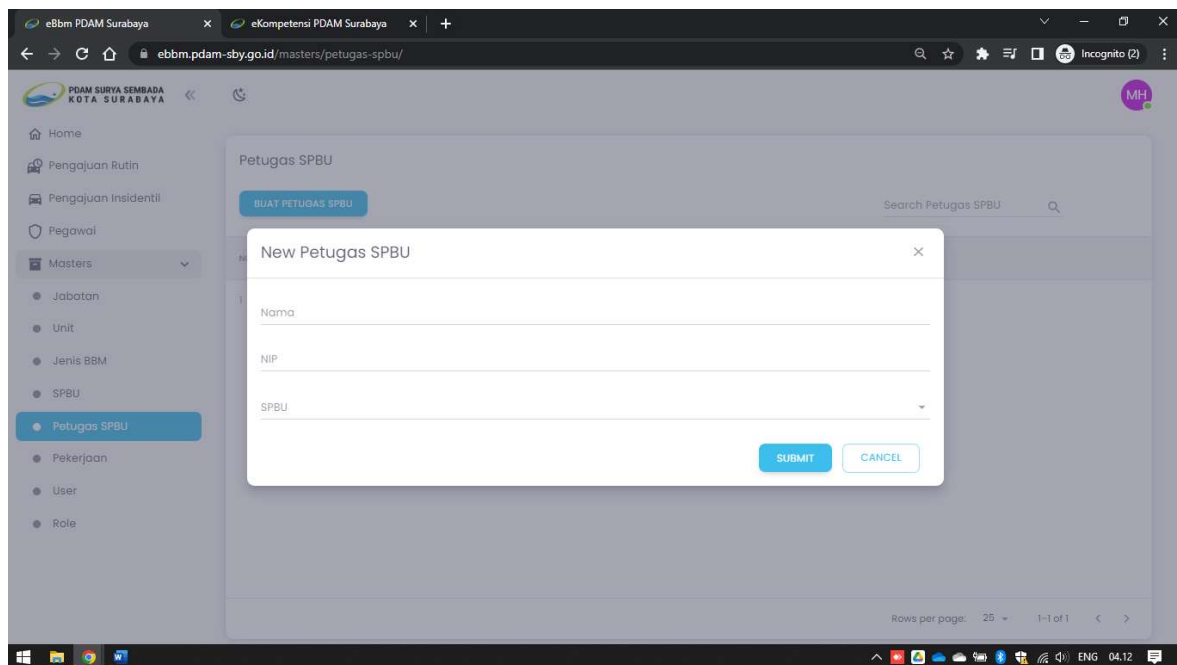
Gambar 38 Tampilan Menu Master SPBU



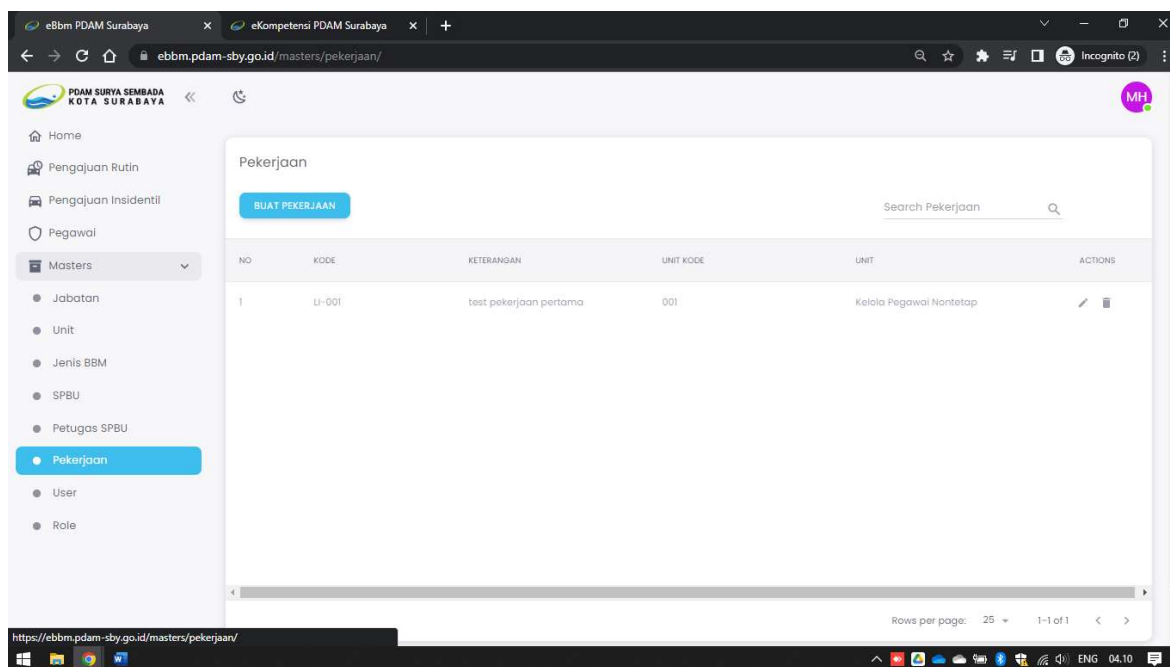
Gambar 39 Tampilan Popup Buat Master SPBU



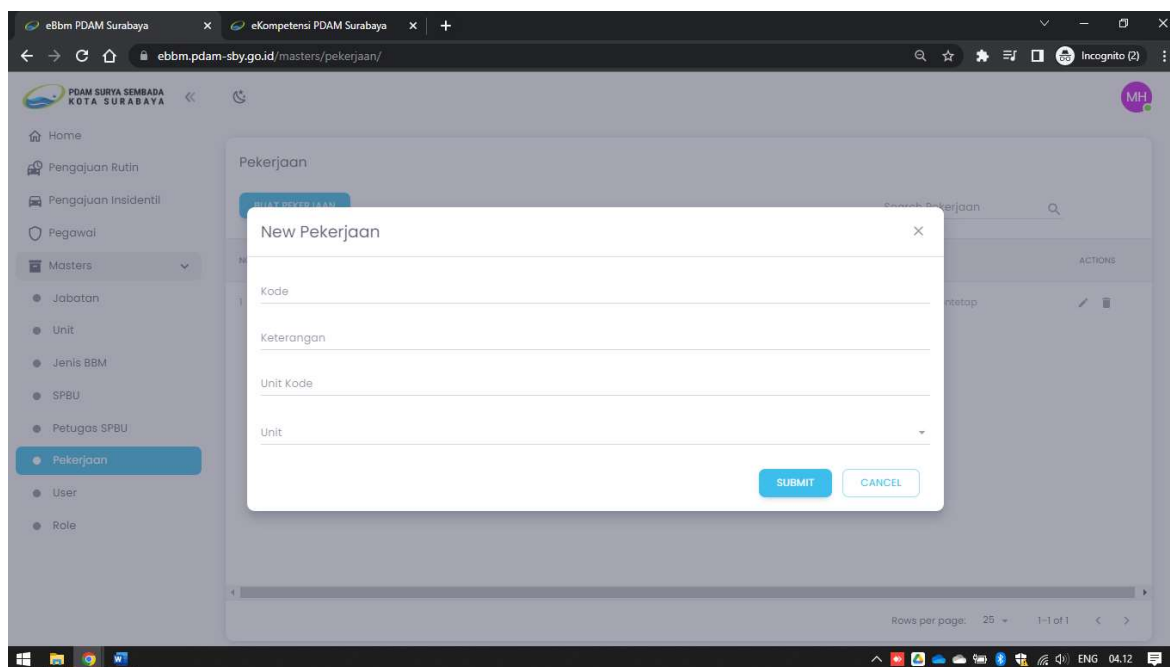
Gambar 40 Tampilan Menu Master Petugas SPBU



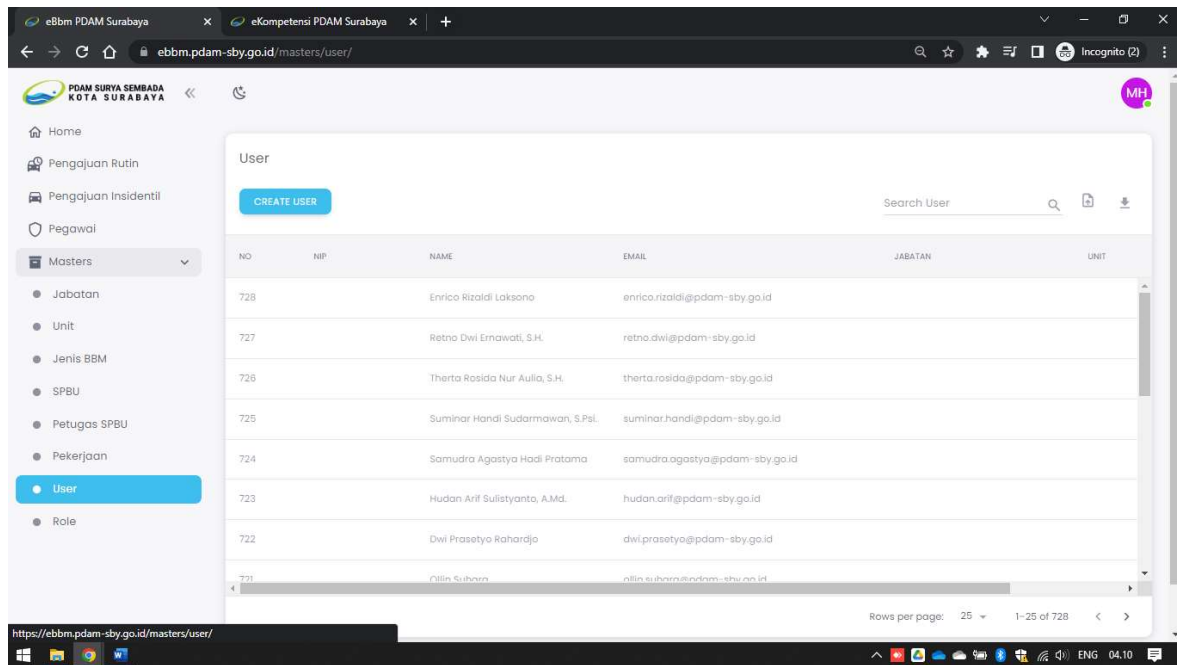
Gambar 41 Tampilan Popup Buat Master Petugas SPBU



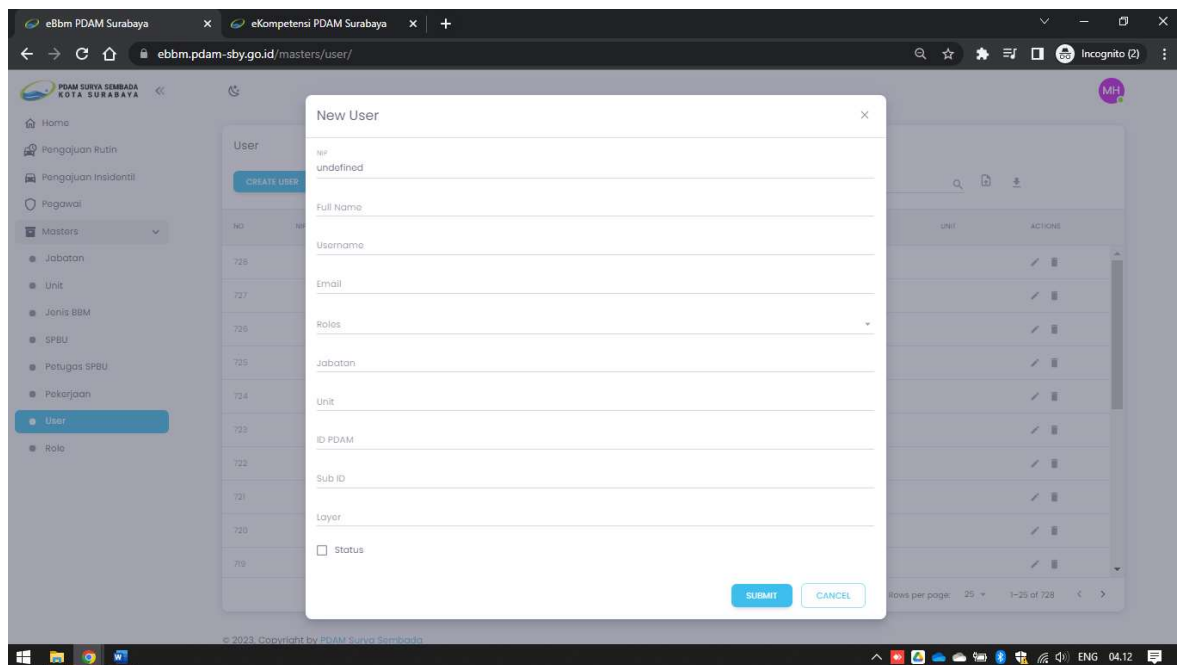
Gambar 42 Tampilan Menu Master Pekerjaan



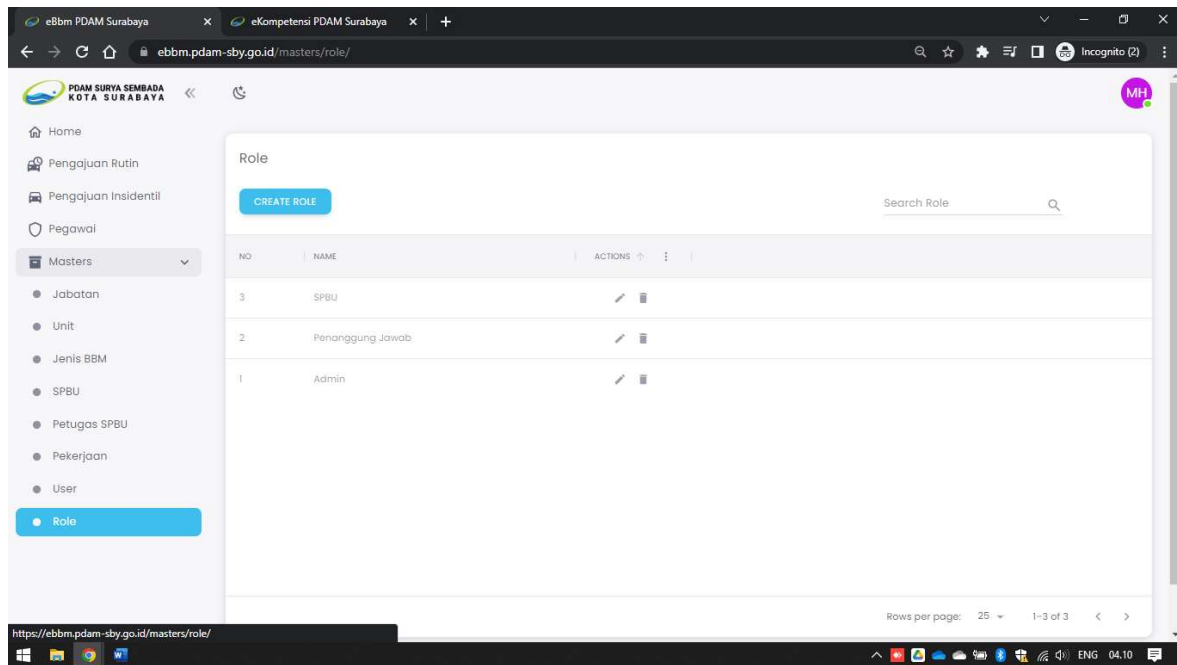
Gambar 43 Tampilan Popup Buat Master Pekerjaan



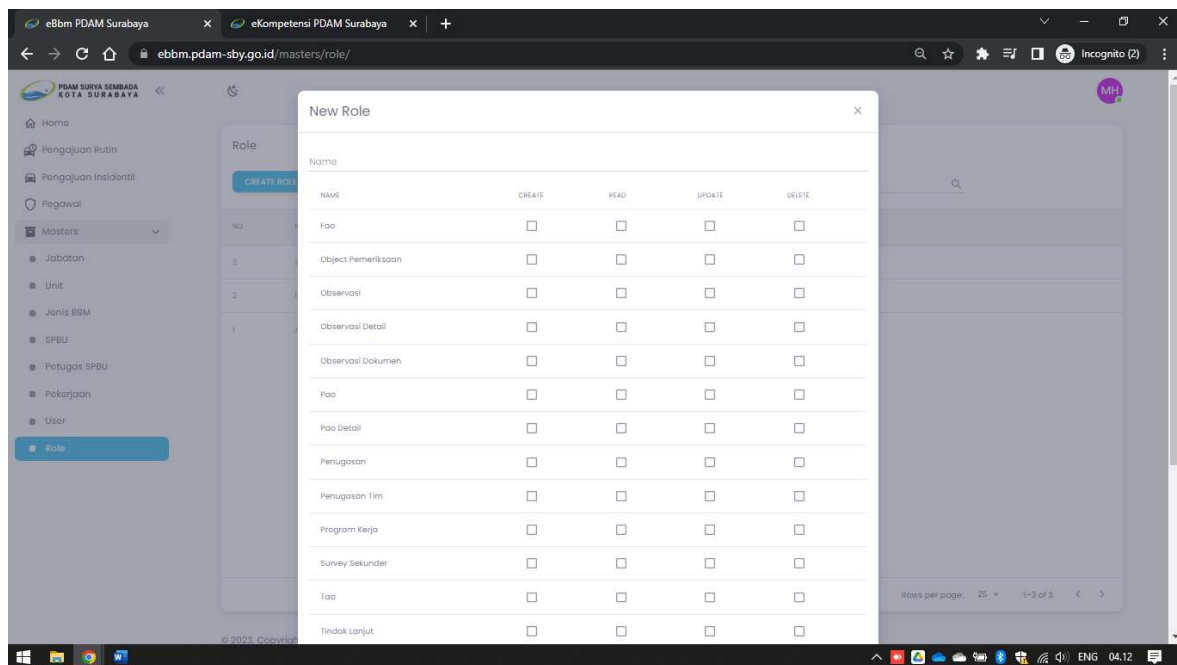
Gambar 44 Tampilan Menu Master User



Gambar 45 Tampilan Popup Buat Master User

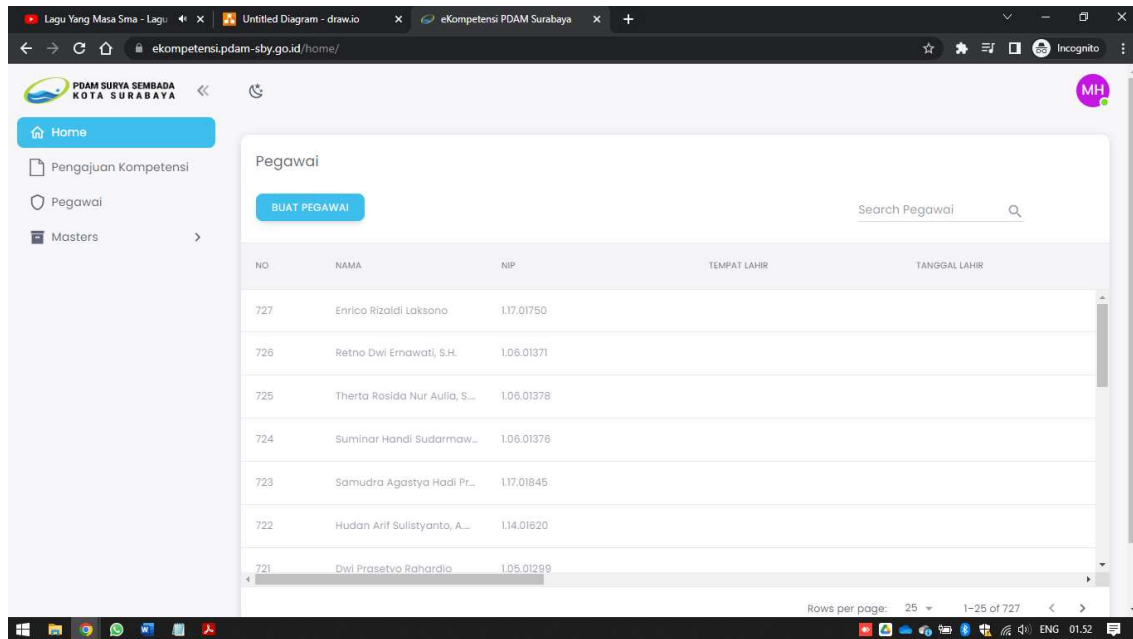


Gambar 46 Tampilan Menu Master Role

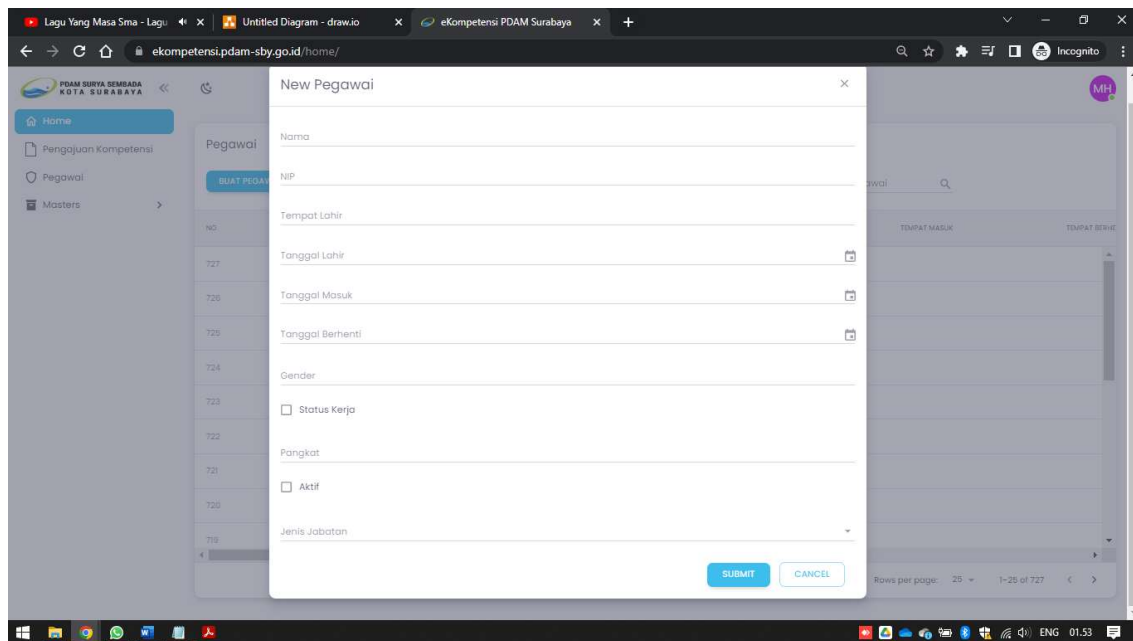


Gambar 47 Tampilan Popup Buat Master Role

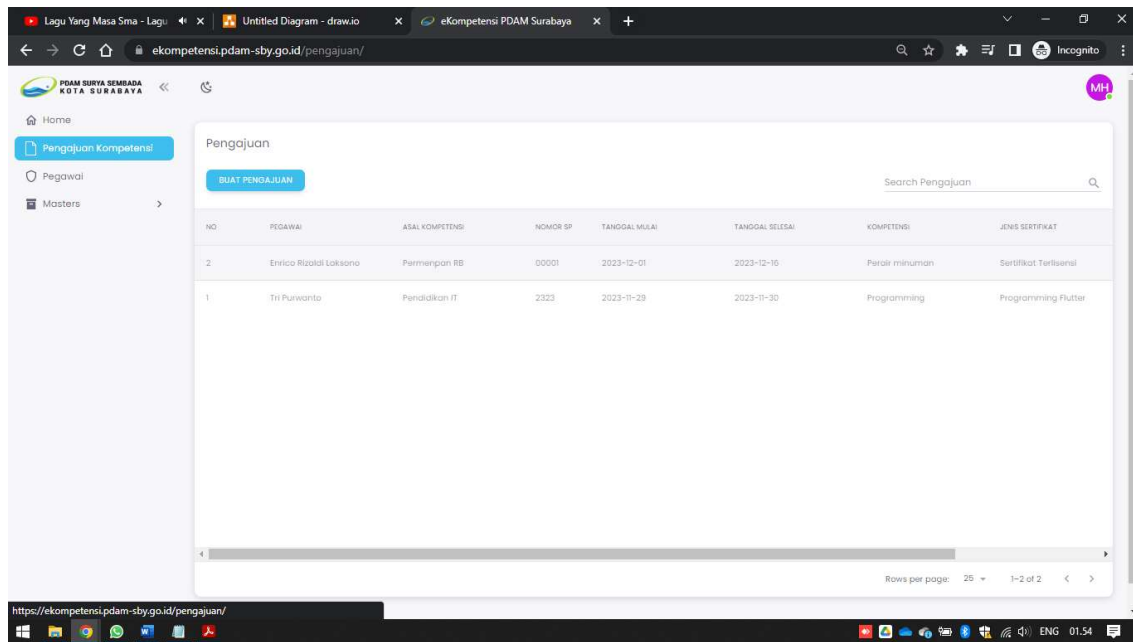
5.4.2 Aplikasi Kompetensi



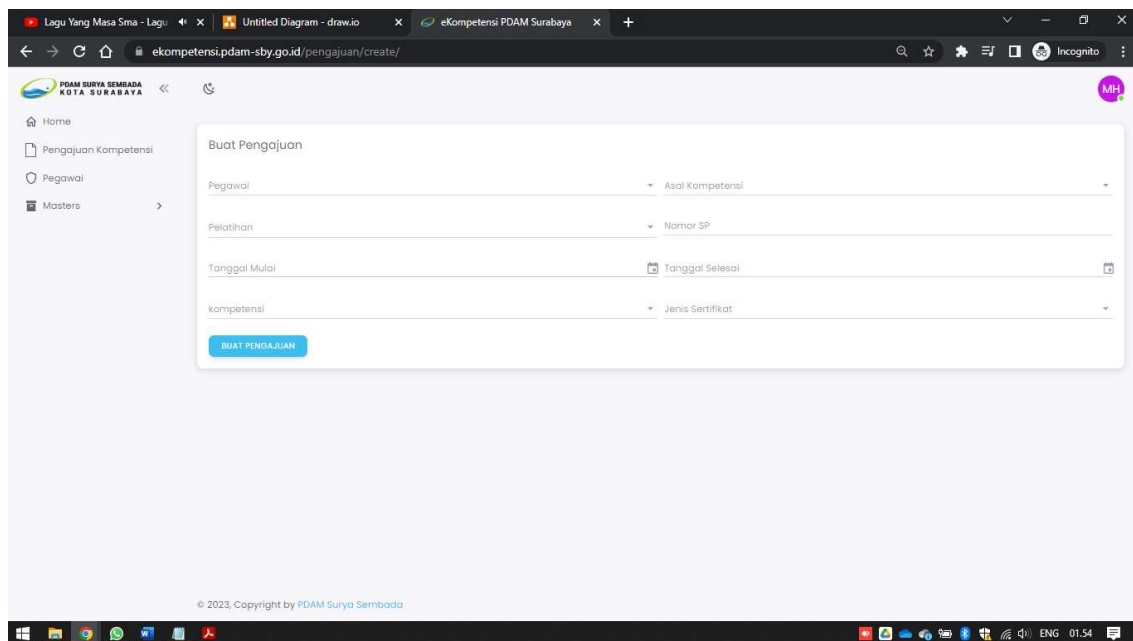
Gambar 48 Tampilan Utama



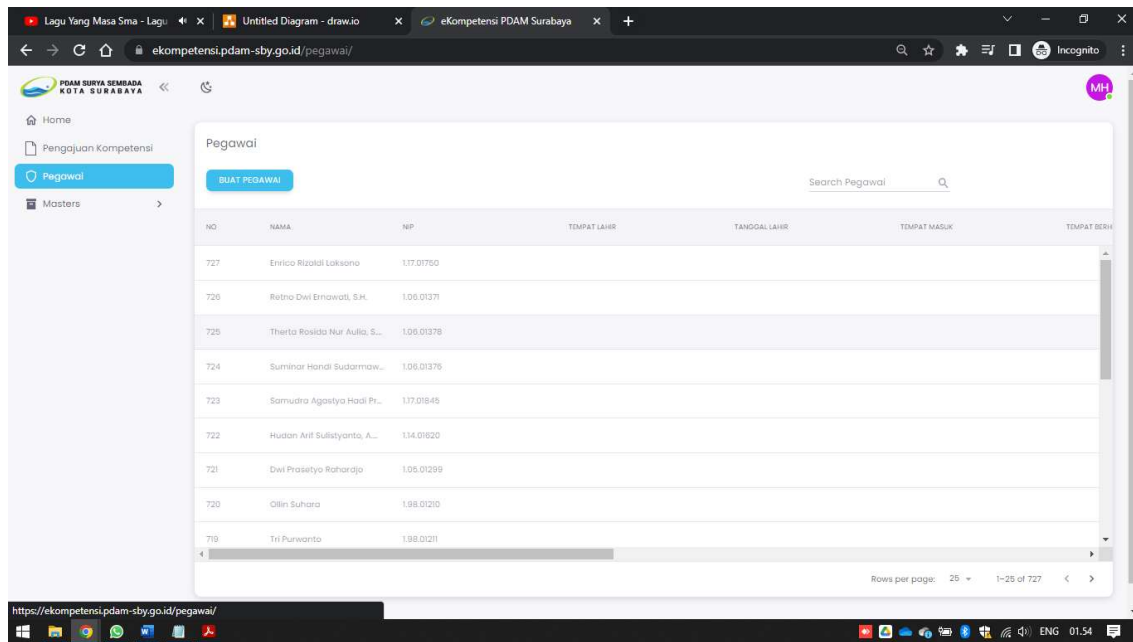
Gambar 49 Popup Pembuatan Pegawai Baru



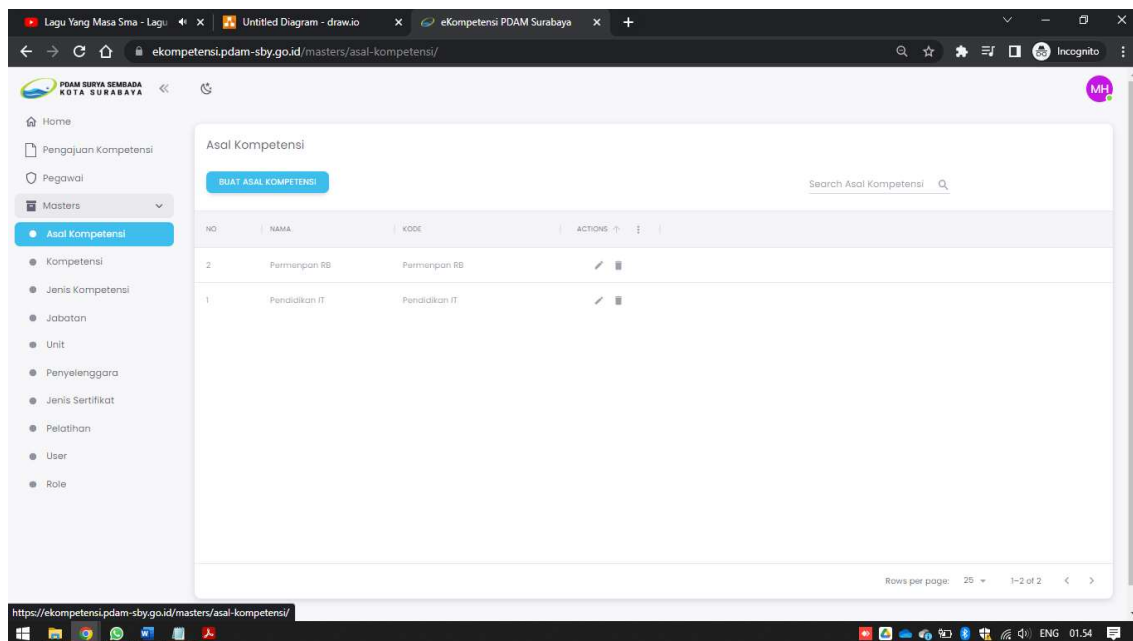
Gambar 50 Form Pengajuan Kompetensi



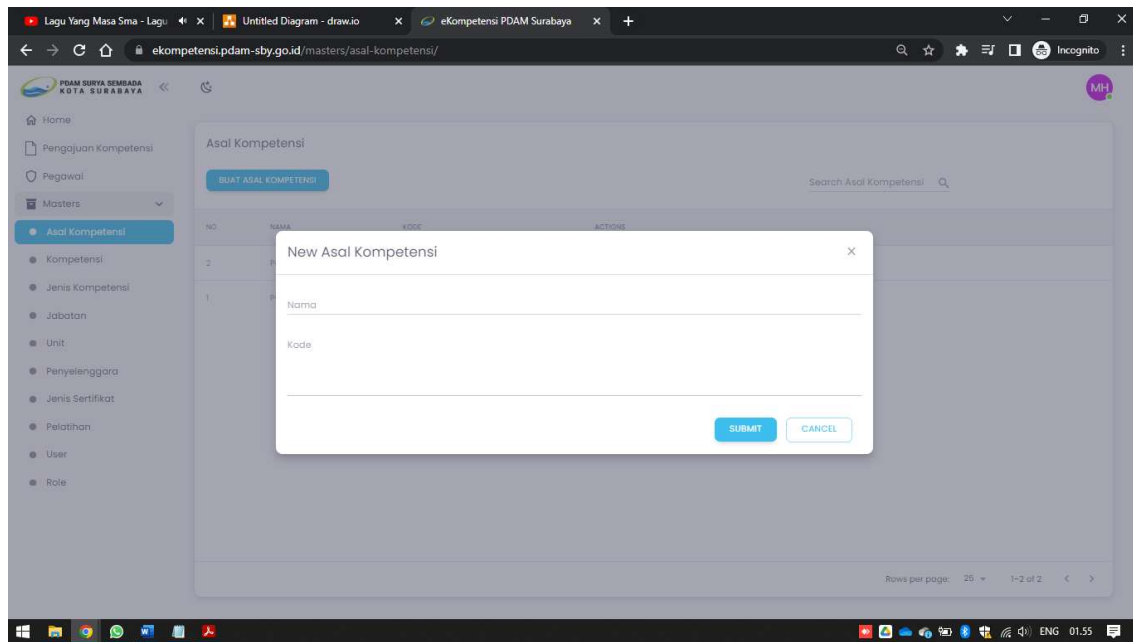
Gambar 51 Form Buat Pengajuan



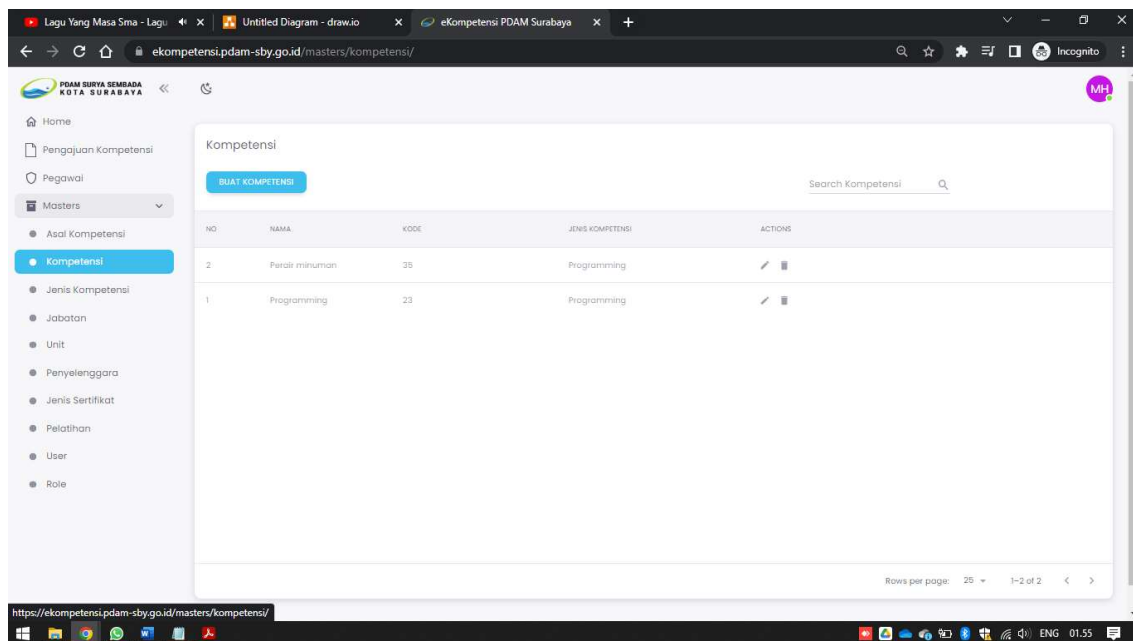
Gambar 52 Page Menu Pegawai



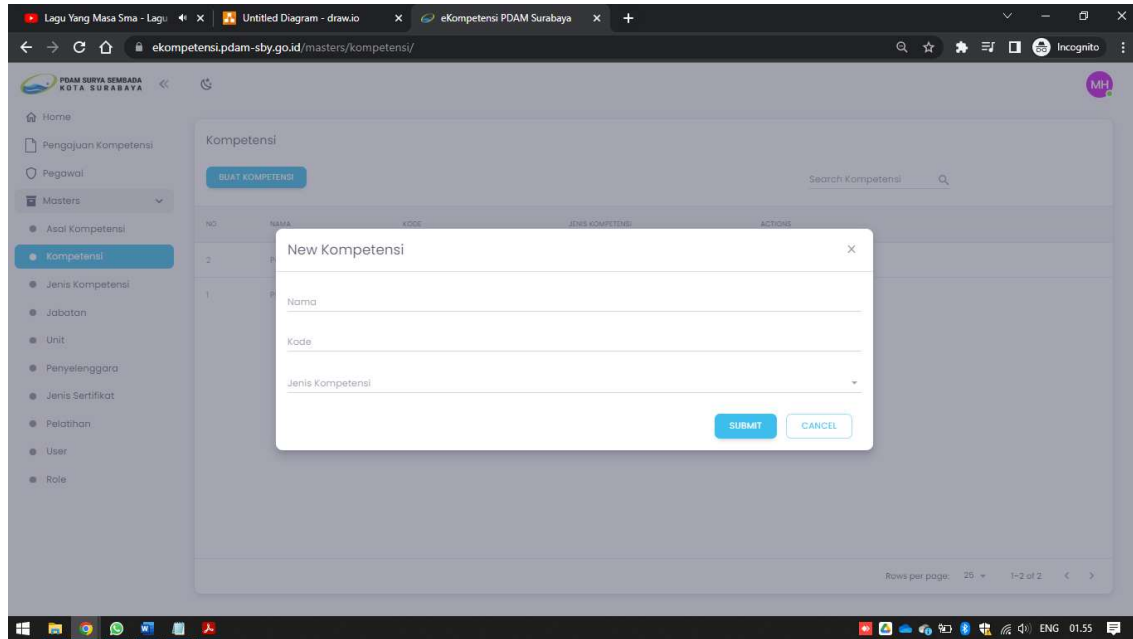
Gambar 53 Page Master Asal Kompetensi



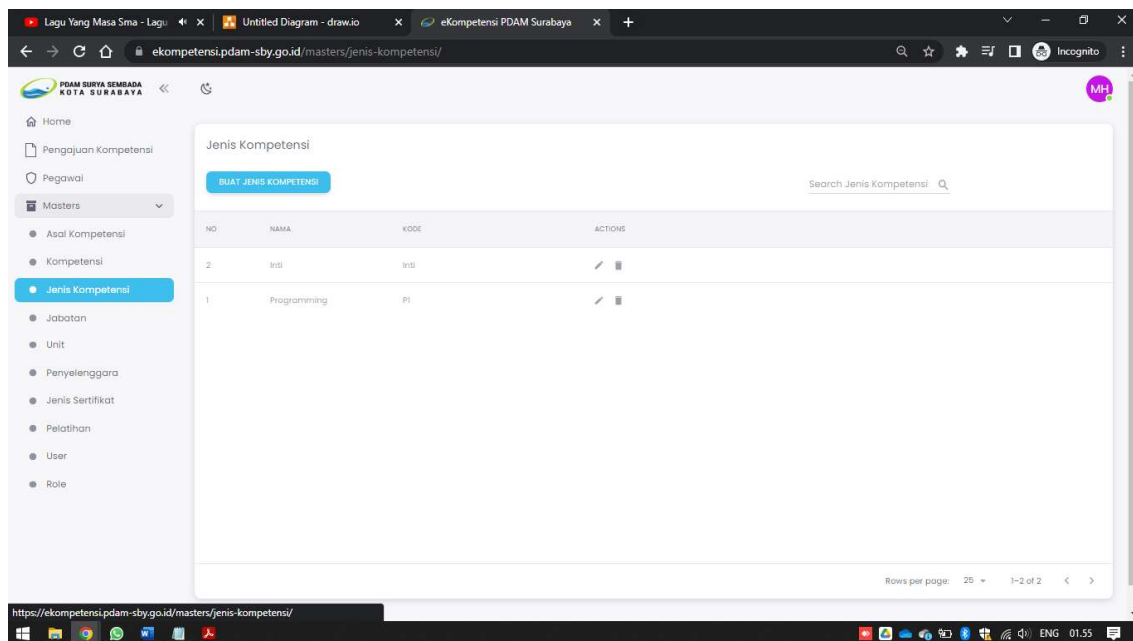
Gambar 54 Popup Master Asal Kompetensi



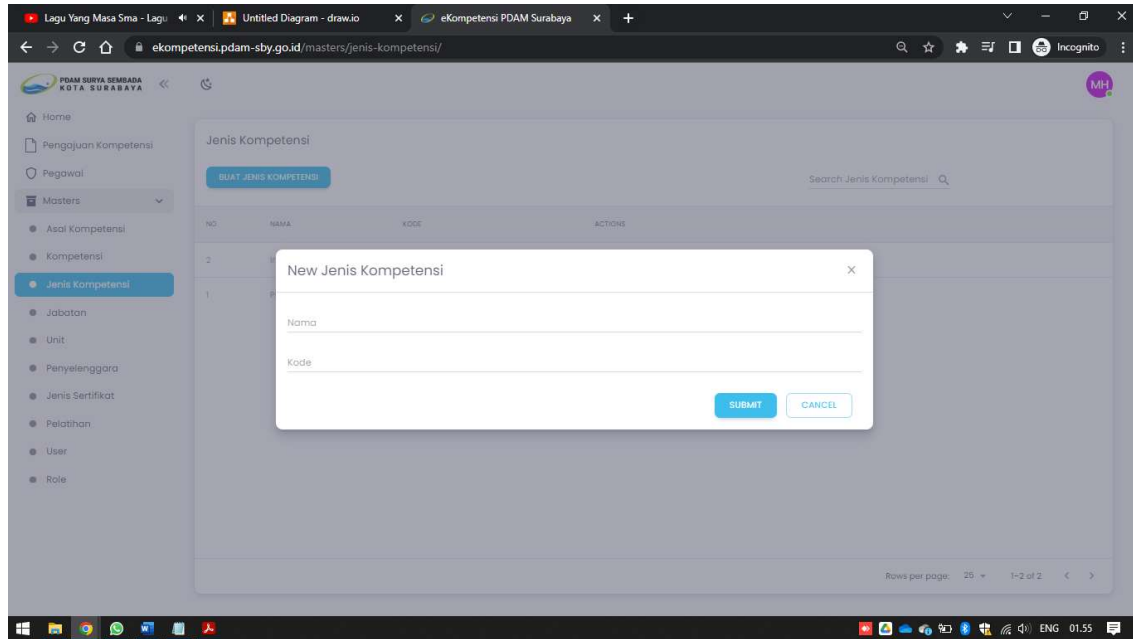
Gambar 55 Page Master Kompetensi



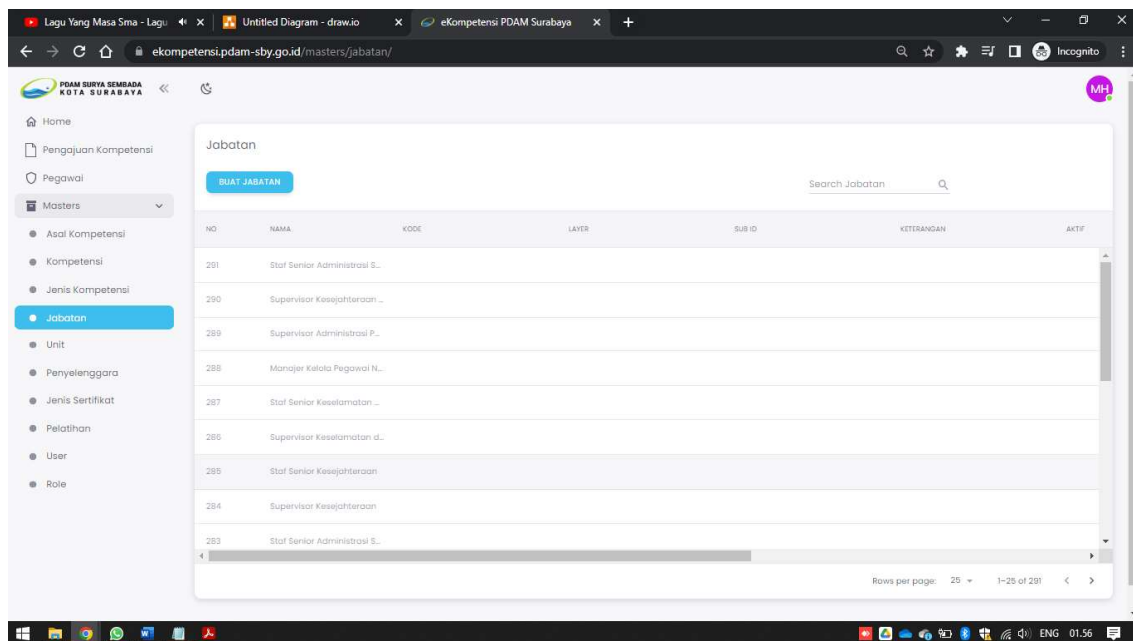
Gambar 56 Popup Master Buat Kompetensi



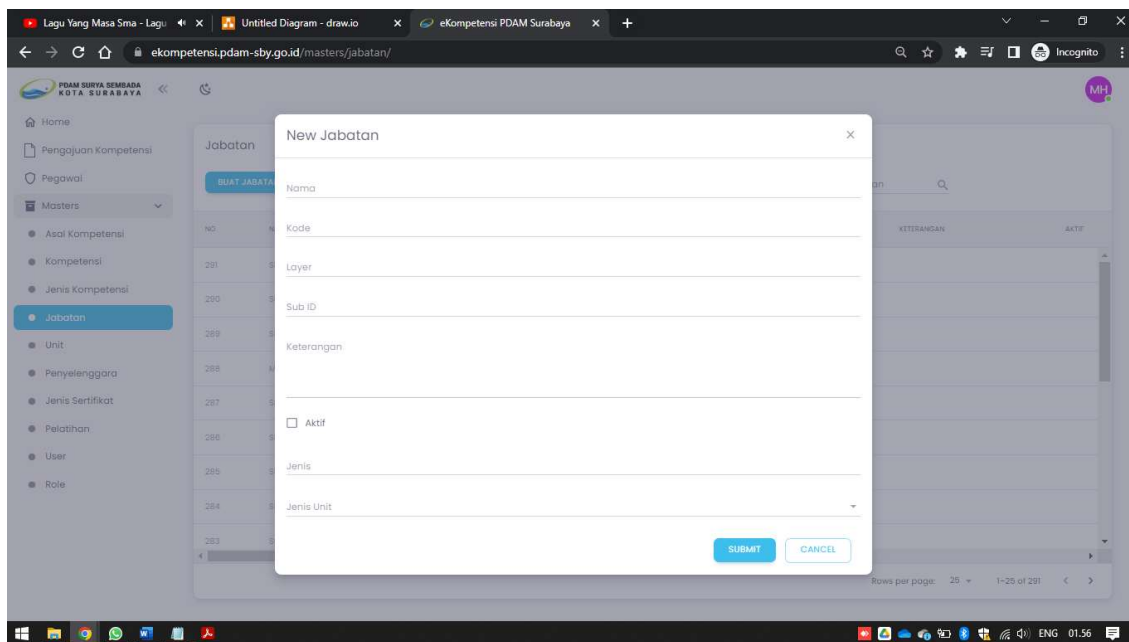
Gambar 57 Page Master Jenis Kompetensi



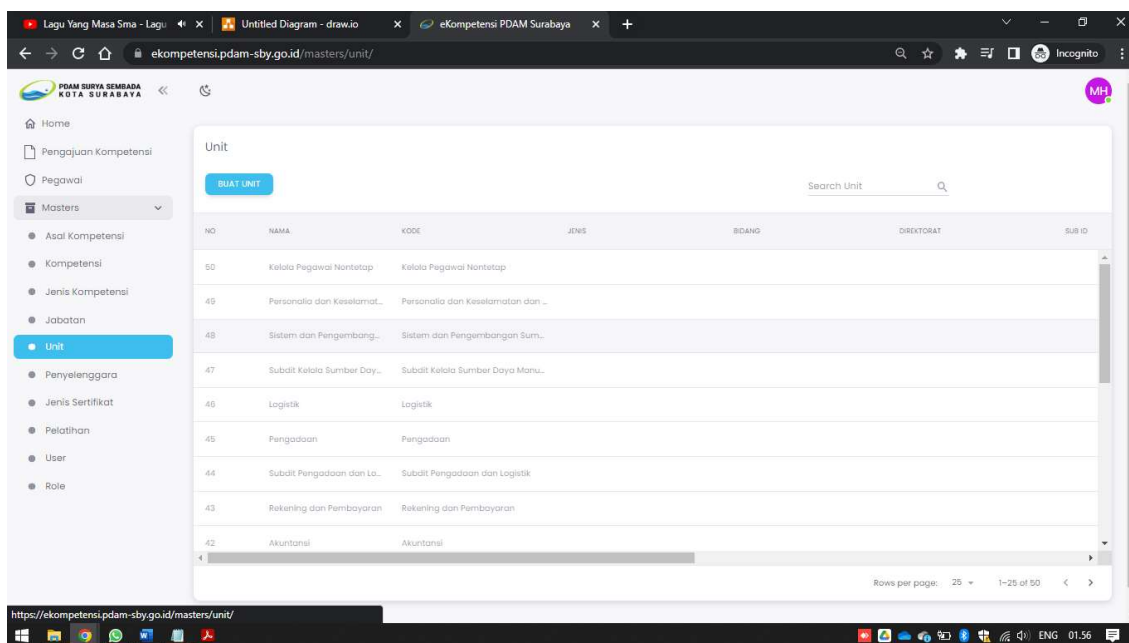
Gambar 58 Popup Master Buat Jenis Kompetensi



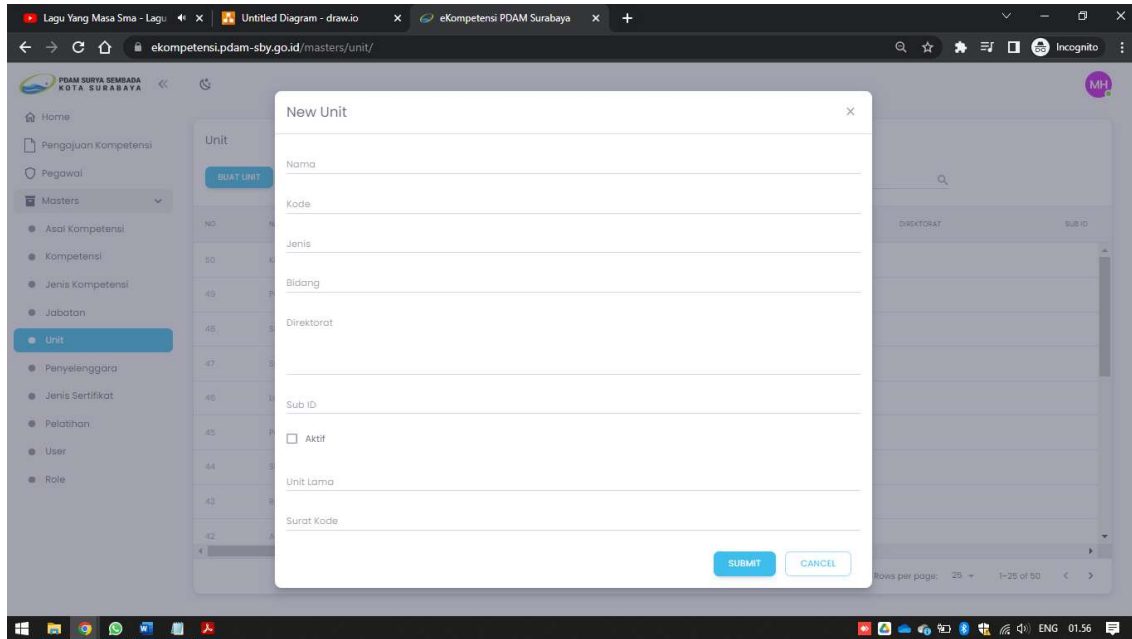
Gambar 59 Page Master Jabatan



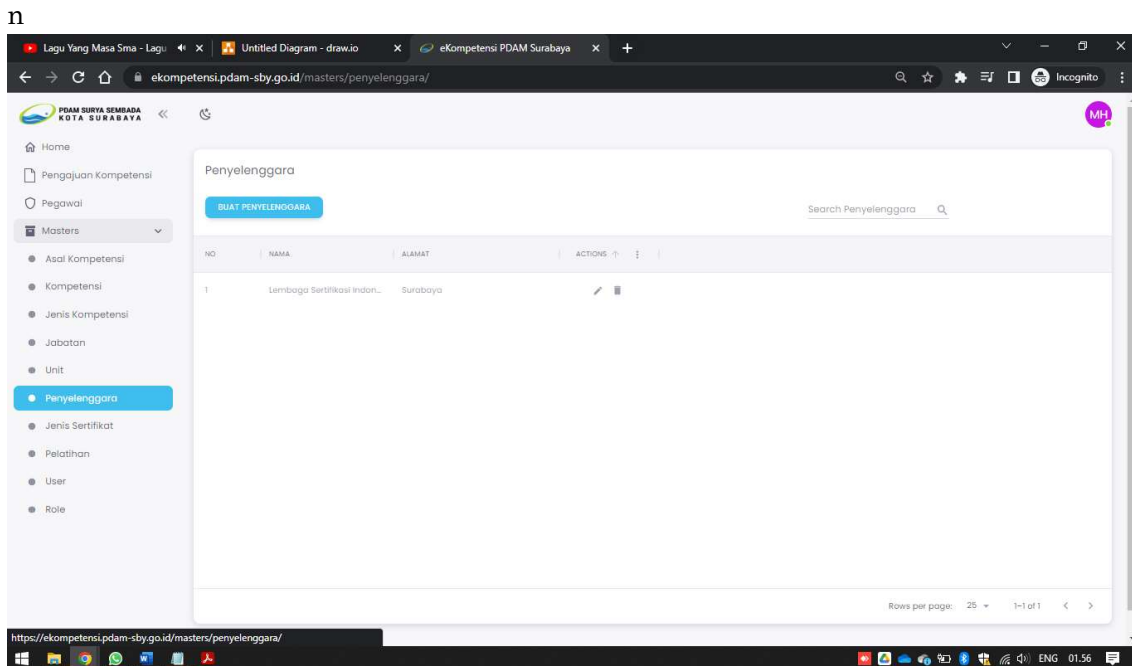
Gambar 60 Popup Master Buat Jabatan



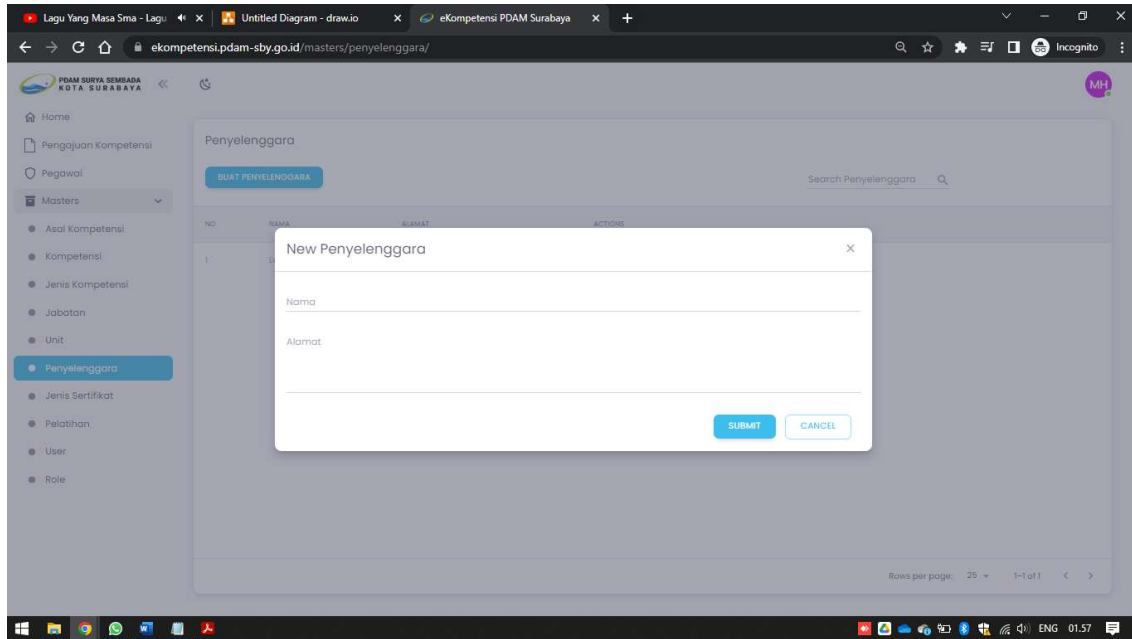
Gambar 61 Page Master Unit



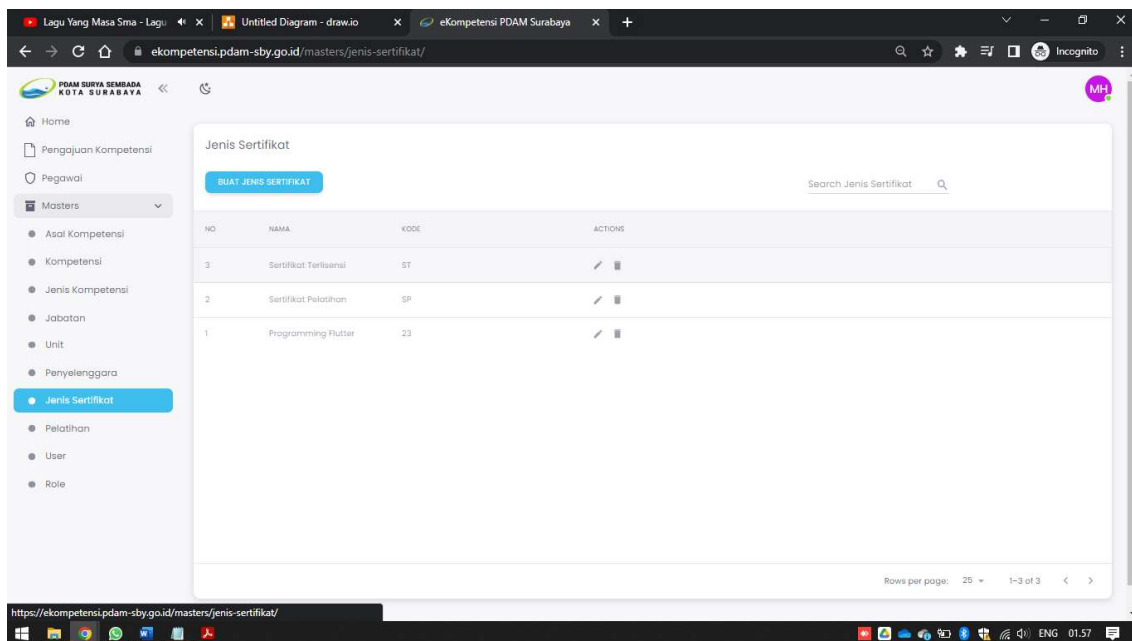
Gambar 62 Popup Master Buat Unit



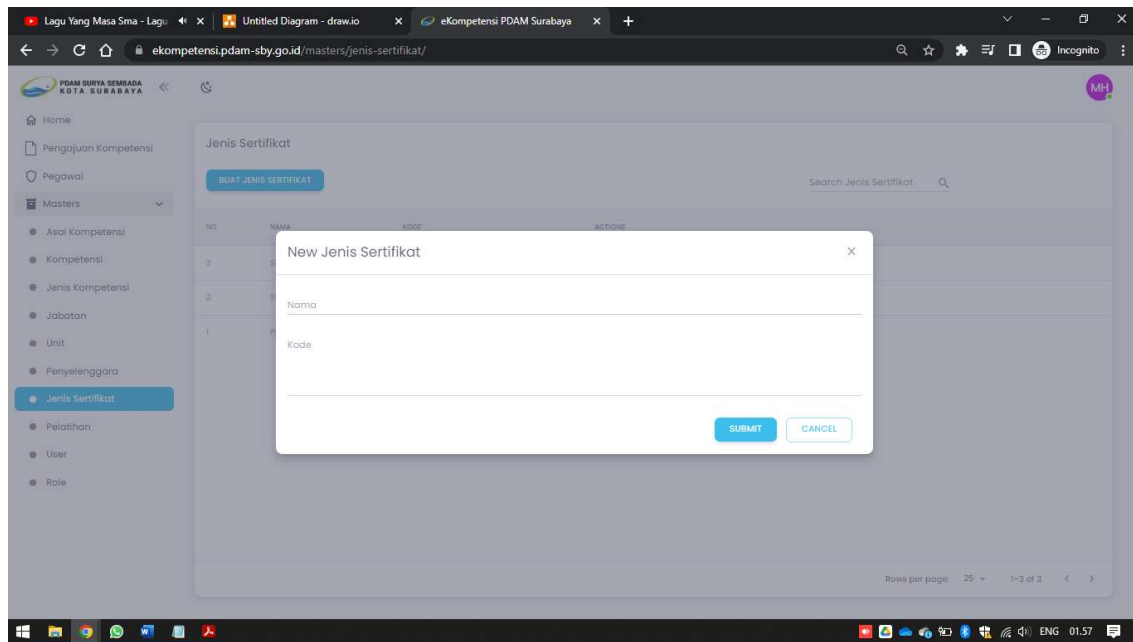
Gambar 63 Page Master Penyelenggara



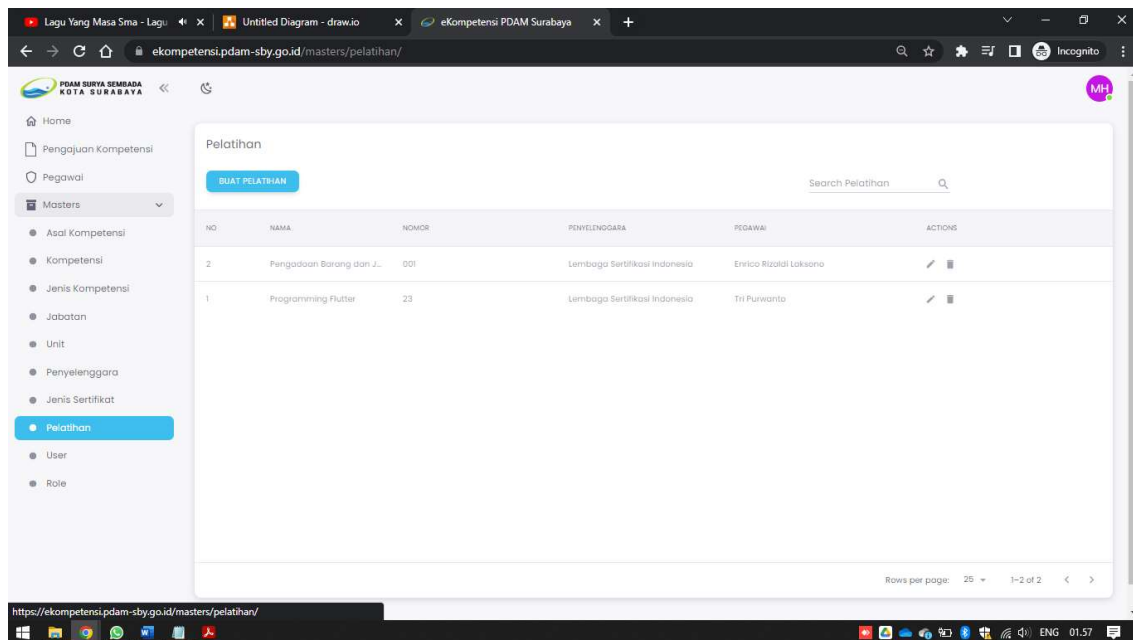
Gambar 64 Popup Master Buat Penyelenggara



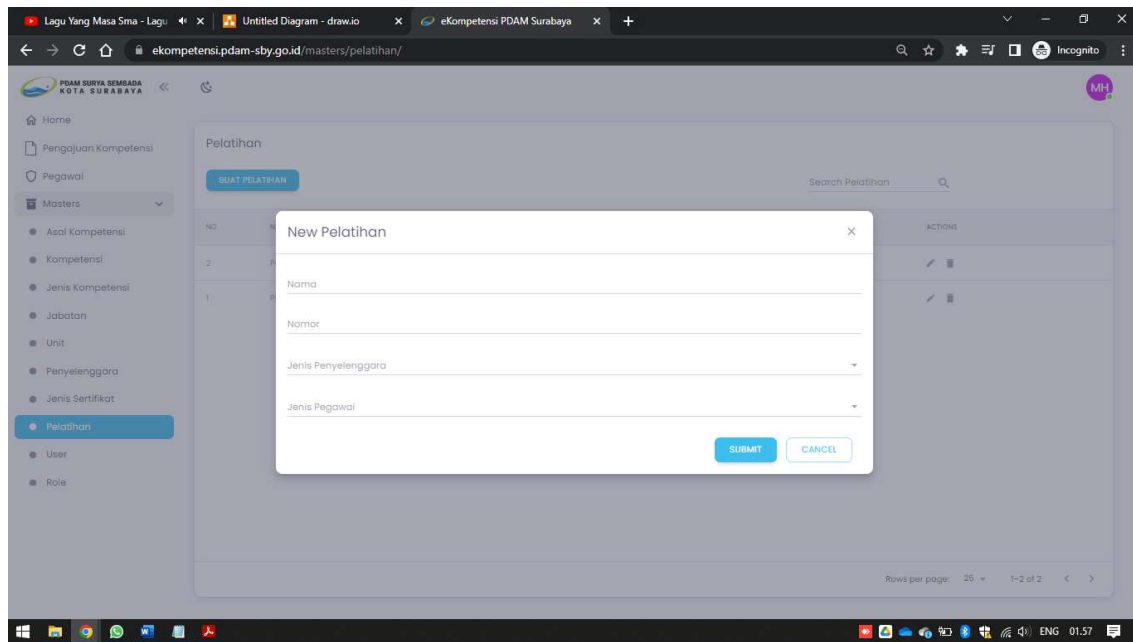
Gambar 65 Page Master Jenis Sertifikat



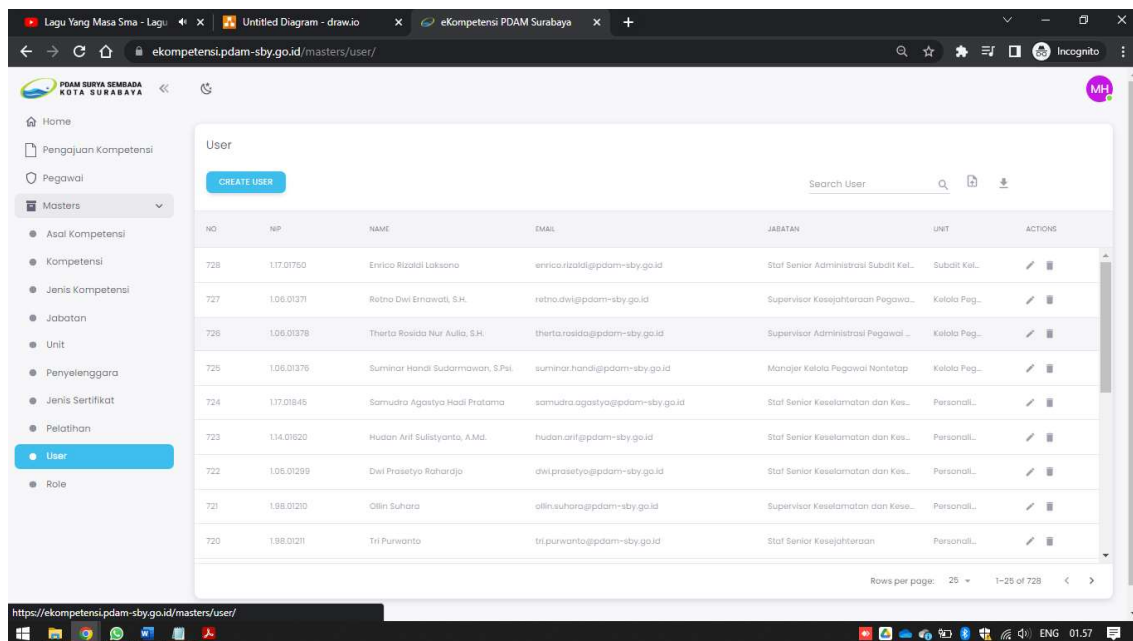
Gambar 66 Popup Master Buat Jenis Sertifikat



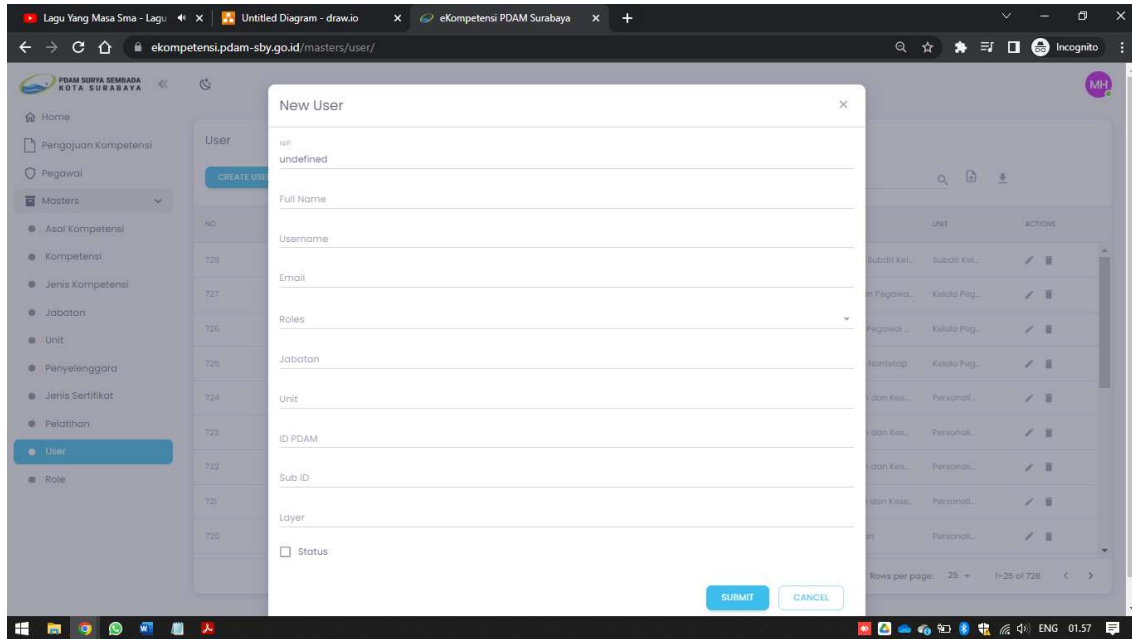
Gambar 67 Page Master Pelatihan



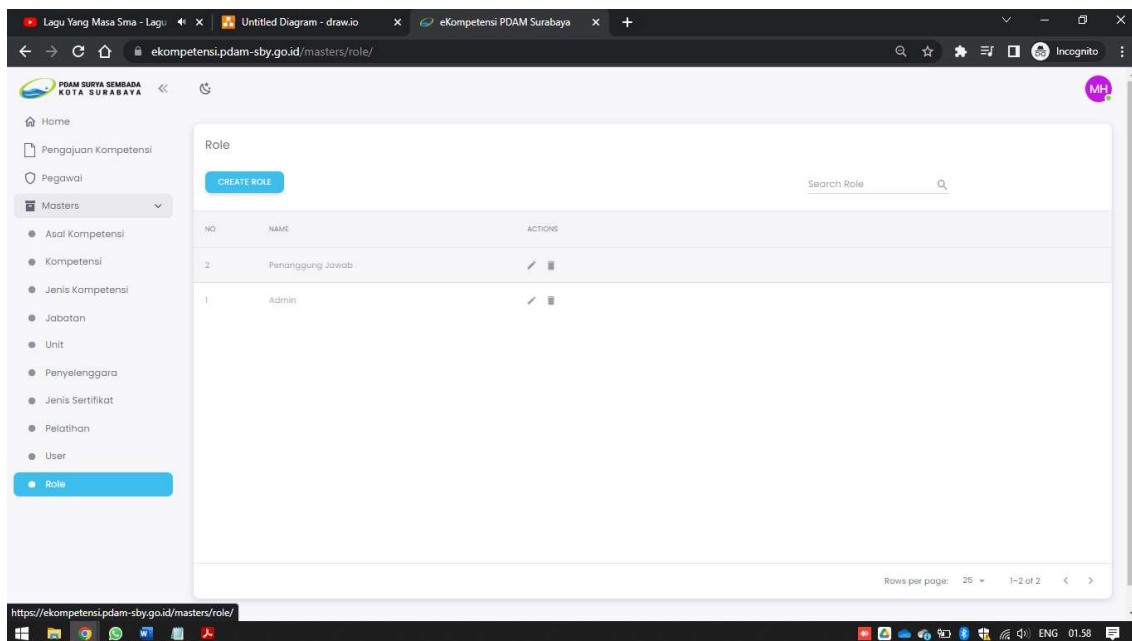
Gambar 68 Popup Master Buat Pelatihan



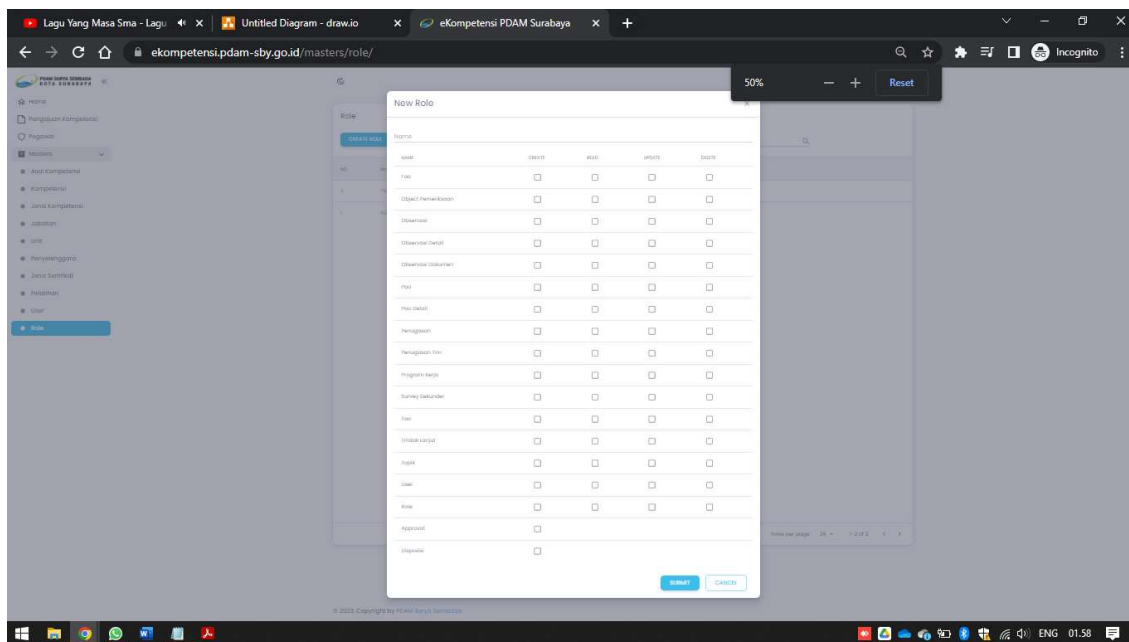
Gambar 69 Page Master User



Gambar 70 Popup Master Buat User

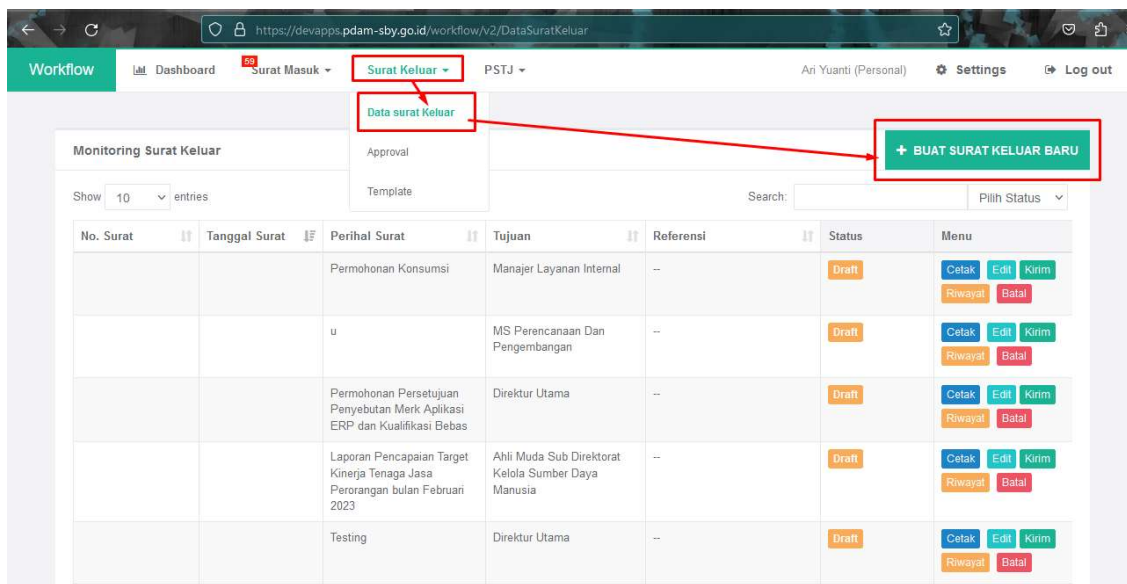


Gambar 71 Page Master Role

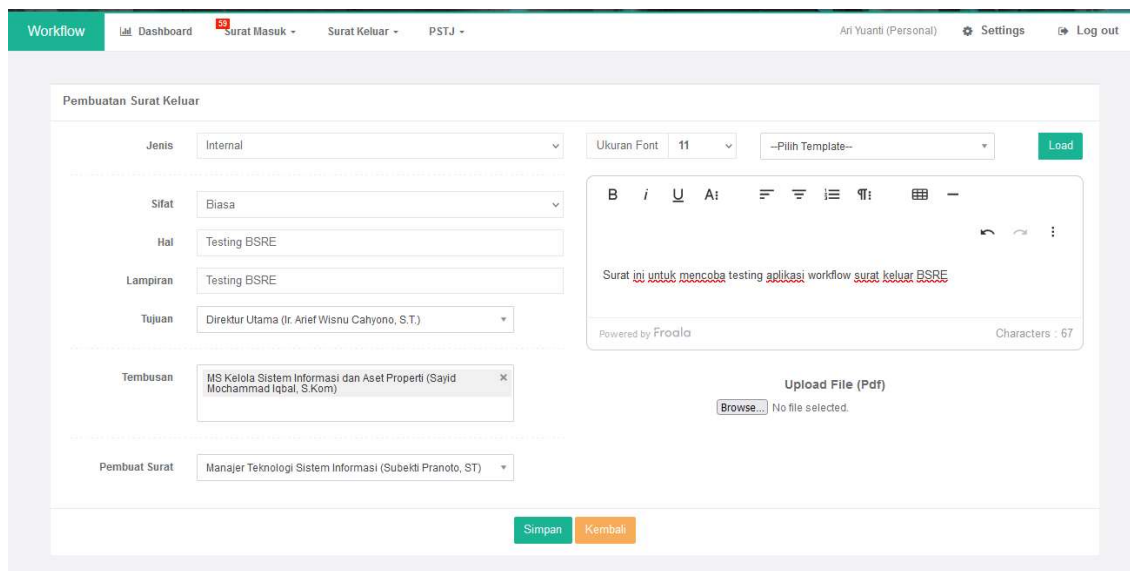


Gambar 72 Popup Master Buat Role

5.4.3 Aplikasi Workflow



Gambar 73 Membuat Surat Baru



Pembuatan Surat Keluar

Jenis: Internal | Ukuran Font: 11 | --Pilih Template-- | Load

Sifat: Biasa

Hal: Testing BSRE

Lampiran: Testing BSRE

Tujuan: Direktur Utama (Ir. Arief Wisnu Cahyono, S.T.)

Tembusan: MS Kelola Sistem Informasi dan Aset Properti (Sayid Mochammad Iqbal, S.Kom)

Pembuat Surat: Manajer Teknologi Sistem Informasi (Subekti Pranoto, ST)

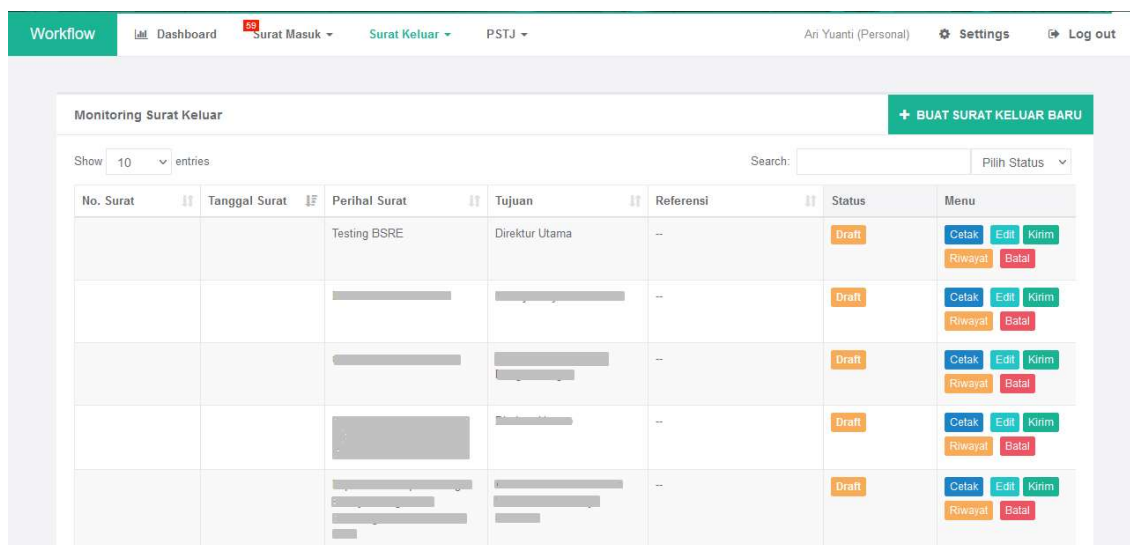
Surat ini untuk mencoba testing aplikasi workflow surat keluar BSRE

Powered by Froala | Characters: 67

Upload File (Pdf) | Browse... | No file selected.

Simpan | Kembali

Gambar 74 Mengisi data surat keluar



Monitoring Surat Keluar | + BUAT SURAT KELUAR BARU

Show: 10 entries | Search: | Pilih Status

No. Surat	Tanggal Surat	Perihal Surat	Tujuan	Referensi	Status	Menu
		Testing BSRE	Direktur Utama	--	Draft	Cetak Edit Kirim Riwayat Batal
				--	Draft	Cetak Edit Kirim Riwayat Batal
				--	Draft	Cetak Edit Kirim Riwayat Batal
				--	Draft	Cetak Edit Kirim Riwayat Batal
				--	Draft	Cetak Edit Kirim Riwayat Batal

Gambar 75 Data Surat Keluar Tampil Setelah Disimpan

Surabaya,

Nomor :
Sifat : Biasa
Lampiran : Testing BSRE
Hal : Testing BSRE

Yth. **Direktur Utama**
PDAM Surya Sembada
Kota Surabaya

Surat ini untuk mencoba testing aplikasi workflow surat keluar BSRE

Disetujui Oleh



Telah ditandatangani secara elektronik
oleh:
**MS Kelola Sistem Informasi dan Aset
Properti**
Sayid Mochammad Iqbal, S.Kom
NIP. 1.05.01321

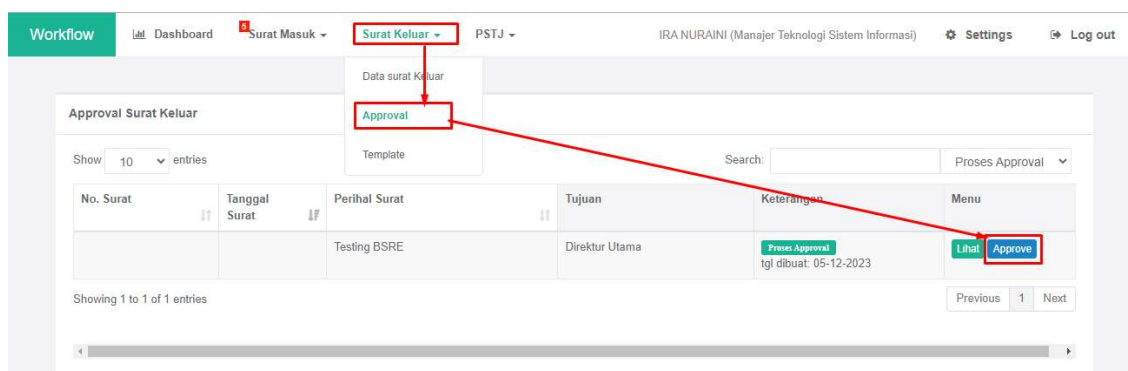
Dibuat Oleh



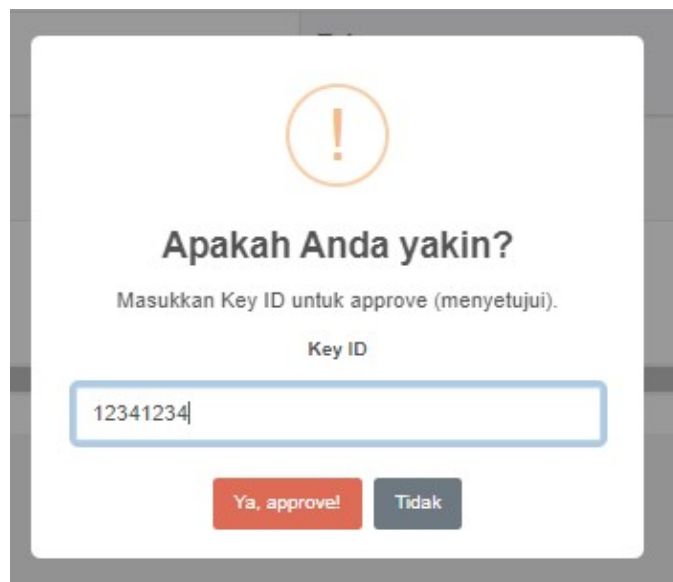
Telah ditandatangani secara elektronik
oleh:
**Plh. Manajer Teknologi Sistem
Informasi**
Ira Nuraini
NIP. 1.06.01390

Tembusan :
MS Kelola Sistem Informasi dan Aset Properti
PDAM Surya Sembada Kota Surabaya

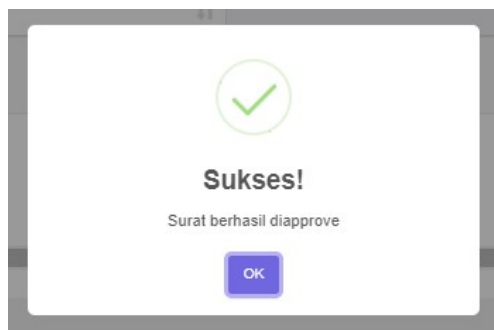
Gambar 76 Bentuk Cetak Pertama Setelah Pembuatan Dan Belum Di Approve



Gambar 77 Approve Surat Melalui Pihak Pertama



Gambar 78 Masukkan KEY ID Untuk Melakukan Approve



Gambar 79 Approve Sukses

Surabaya,

Nomor :
Sifat : Biasa
Lampiran : Testing BSRE
Hal : Testing BSRE

Yth. **Direktur Utama**
PDAM Surya Sembada
Kota Surabaya

Surat ini untuk mencoba testing aplikasi workflow surat keluar BSRE

Disetujui Oleh



Telah ditandatangani secara elektronik
oleh:
**MS Kelola Sistem Informasi dan Aset
Properti**
Sayid Mochammad Iqbal, S.Kom
NIP. 1.05.01321

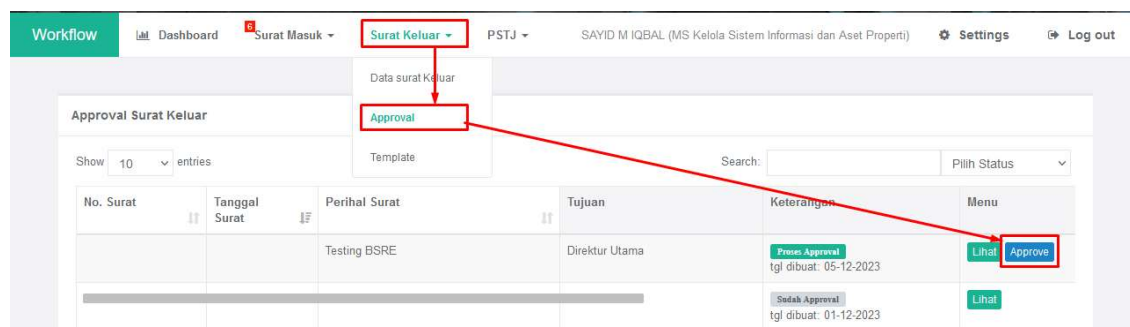
Dibuat Oleh



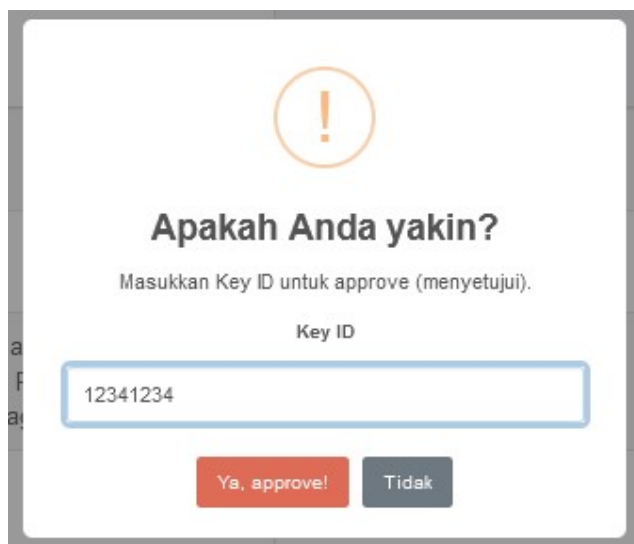
Telah ditandatangani secara elektronik
oleh:
**Plh. Manajer Teknologi Sistem
Informasi**
Ira Nuraini
NIP. 1.06.01390

Tembusan :
MS Kelola Sistem Informasi dan Aset Properti
PDAM Surya Sembada Kota Surabaya

Gambar 80 Sukses Approve dan KEY ID Sesuai, Maka Surat Telah Memiliki TTD Barcode



Gambar 81 Approve Surat Melalui Pihak Kedua



Apakah Anda yakin?

Masukkan Key ID untuk approve (menyetujui).

Key ID

12341234

Ya, approve! Tidak

Gambar 82 Memasukkan KEY ID

Workflow

Dashboard

Surat Masuk

Surat Keluar

PSTJ

SAYID M IQBAL (MS Kelola Sistem Informasi dan Aset Properti)

Settings

Log out

Approval Surat Keluar

Show10entries

Search:

Pilih Status

No. Surat	Tanggal Surat	Perihal Surat	Tujuan	Keterangan	Menu
151/TSI/2023	05-12-2023	Testing BSRE	Direktur Utama	Sudah Approval tgl dibuat: 05-12-2023	Lihat
				Sudah Approval tgl dibuat: 01-12-2023	Lihat

Gambar 83 File Tersimpan



Surabaya,

Nomor :
Sifat : Biasa
Lampiran : Testing BSRE
Hal : Testing BSRE

Yth. **Direktur Utama**
PDAM Surya Sembada
Kota Surabaya

Surat ini untuk mencoba testing aplikasi workflow surat keluar BSRE

Disetujui Oleh



Telah ditandatangani secara elektronik
oleh:
**MS Kelola Sistem Informasi dan Aset
Properti**
Sayid Mochammad Iqbal, S.Kom
NIP. 1.05.01321

Dibuat Oleh



Telah ditandatangani secara elektronik
oleh:
**PIh. Manajer Teknologi Sistem
Informasi**
Ira Nuraini
NIP. 1.06.01390

Tembusan :
MS Kelola Sistem Informasi dan Aset Properti
PDAM Surya Sembada Kota Surabaya

Gambar 84 Surat Setelah Approve Kedua