

LAPORAN PRAKTEK KERJA LAPANGAN

**RANCANG BANGUN SISTEM INFORMASI MANAJEMEN
ASET PADA BADAN PENGELOLAAN KEUANGAN DAN
ASET DAERAH PROVINSI NUSA TENGGARA BARAT**



Disusun oleh:

DITHA NURCAHYA AVIANTY

F1D 017 016

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2020

LEMBAR PENGESAHAN
LAPORAN PRAKTEK KERJA LAPANGAN

RANCANG BANGUN SISTEM INFORMASI MANAJEMEN
ASET PADA BADAN PENGELOLAAN KEUANGAN DAN
ASET DAERAH PROVINSI NUSA TENGGARA BARAT

Disusun oleh:

DITHA NURCAHYA AVIANTY

F1D 017 016

Telah disetujui oleh :

Tanggal :

1. Dosen Pembimbing

1. 16 Juni 2020



Andy Hidayat Jatmika, ST., M.Kom.
NIP. 19831209 201212 1 001

2. Pembimbing Lapangan

2. 17 Juni 2020



Lalu Miftahul Ulum, S.T.
NIP: 19820312 200501 1007

Mengetahui,
Sekretaris Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram



Andy Hidayat Jatmika, ST., M.Kom.
NIP. 19831209 201212 1 001

KATA PENGANTAR

Puji syukur penulis haturkan atas rahmat Allah SWT, Tuhan semesta alam, karena dengan berkat, rahmat, dan limpahan karunia-Nya, penulis dapat menyelesaikan Laporan Praktek Kerja Lapangan ini sebagaimana mestinya.

Adapun Praktek Kerja Lapangan (PKL) ini penulis laksanakan di Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat dengan judul “Sistem Informasi Manajemen Aset Pada Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat”. Sistem Informasi Manajemen Aset ini diperlukan oleh Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat sebagai aplikasi untuk mempermudah dalam pengelolaan aset di kantor. Dalam pembuatan laporan ini penulis berpedoman pada bahan kuliah, petunjuk dari pembimbing lapangan, dosen pembimbing, referensi dan literatur yang terkait dengan penulisan laporan.

Penulis menyadari laporan ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik serta saran yang bersifat membangun agar dapat menghasilkan karya yang lebih baik dimasa mendatang. Semoga laporan ini dapat bermanfaat bagi penulis dan pembaca sekalian.

Mataram, 17 Juni 2020



Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
BAB II TINJAUAN INSTANSI TEMPAT PKL	3
2.1 Profil Singkat BPKAD Provinsi Nusa Tenggara Barat.....	3
2.2 Visi dan Misi BPKAD Provinsi Nusa Tenggara Barat	4
2.3 Struktur Organisasi BPKAD Provinsi Nusa Tenggara Barat.....	5
BAB III DASAR TEORI	9
3.1 Sistem Informasi	9
3.2 Manajemen Aset.....	9
3.3 <i>Website</i>	9
3.4 UML.....	10
3.5 <i>Use Case Diagram</i>	11
3.6 <i>Activity Diagram</i>	11
3.7 <i>Class Diagram</i>	13
3.8 <i>Sequence Diagram</i>	13
3.9 ERD.....	15
3.10 <i>Framework</i> Laravel	16
3.11 Konsep MVC.....	16
3.12 Visual Studio Code.....	17
3.13 MySql.....	18
BAB IV PEMBAHASAN.....	19
4.1 Metode Pengembangan Sistem	19
4.2 Analisa kebutuhan	19

4.3	Desain Sistem.....	19
4.3.1	<i>Use Case Diagram</i>	20
4.3.2	<i>Activity Diagram</i>	22
4.3.3	<i>Class Diagram</i>	28
4.3.4	<i>Sequence Diagram</i>	29
4.3.5	ERD.....	35
4.3.6	Desain Tampilan Sistem.....	38
4.4	Implementasi Sistem.....	47
4.3.1	Implementasi <i>Database</i> Sistem.....	47
4.3.2	Implementasi Tampilan Sistem.....	50
4.3.3	Pengujian.....	59
BAB V KESIMPULAN DAN SARAN.....		62
5.1	Kesimpulan.....	62
5.2	Saran.....	62
DAFTAR PUSTAKA.....		63

DAFTAR GAMBAR

Gambar 2.1 Struktur organisasi BPKAD Provinsi NTB	5
Gambar 2.2 Struktur organisasi BPKAD Provinsi NTB bidang sekretariat	5
Gambar 2.3 Struktur organisasi BPKAD Provinsi NTB bidang anggaran	6
Gambar 2.4 Struktur organisasi BPKAD Provinsi NTB bidang perbendaharaan ..	6
Gambar 2.5 Struktur organisasi BPKAD Provinsi NTB bidang akuntansi dan pelaporan	7
Gambar 2.6 Struktur organisasi BPKAD Provinsi NTB bidang pengelolaan BMD	7
Gambar 2.7 Struktur organisasi BPKAD Provinsi NTB bidang Islamic Center	8
Gambar 2.8 Struktur organisasi BPKAD Provinsi NTB bidang UPTB	8
Gambar 3.1 Konsep MVC	17
Gambar 4.1 Metode <i>waterfall</i>	19
Gambar 4.2 <i>Use case diagram</i>	20
Gambar 4.3 <i>Activity diagram Login</i>	22
Gambar 4.4 <i>Activity diagram</i> tambah pengguna oleh admin	23
Gambar 4.5 <i>Activity diagram</i> ubah pengguna oleh admin	23
Gambar 4.6 <i>Activity diagram</i> hapus pengguna oleh admin	24
Gambar 4.7 <i>Activity diagram</i> tambah data	25
Gambar 4.8 <i>Activity diagram</i> ubah data	25
Gambar 4.9 <i>Activity diagram</i> hapus data	26
Gambar 4.10 <i>Activity diagram</i> cetak laporan persediaan oleh admin	26
Gambar 4.11 <i>Activity diagram</i> cetak <i>stock of name</i> oleh admin	27
Gambar 4.12 <i>Activity diagram</i> ubah profil	27
Gambar 4.13 <i>Class diagram</i>	28
Gambar 4.14 <i>Sequence diagram login</i>	29
Gambar 4.15 <i>Sequence diagram</i> tambah pengguna	29
Gambar 4.16 <i>Sequence diagram</i> ubah pengguna	30
Gambar 4.17 <i>Sequence diagram</i> hapus pengguna	31
Gambar 4.18 <i>Sequence diagram</i> tambah data	31
Gambar 4.19 <i>Sequence diagram</i> ubah data	32
Gambar 4.20 <i>Sequence diagram</i> hapus data	32
Gambar 4.21 <i>Sequence diagram</i> cetak laporan persediaan	33
Gambar 4.22 <i>Sequence diagram</i> cetak <i>stock of name</i>	33
Gambar 4.23 <i>Sequence diagram</i> ubah profil	34
Gambar 4.24 <i>ER Diagram</i>	35
Gambar 4.25 Desain tampilan halaman <i>login</i>	38
Gambar 4.26 Desain tampilan halaman beranda admin	39
Gambar 4.27 Desain tampilan halaman daftar barang admin	39
Gambar 4.28 Desain tampilan halaman isi data daftar barang admin	40
Gambar 4.29 Desain tampilan halaman daftar barang masuk admin	40
Gambar 4.30 Desain tampilan halaman isi data barang masuk admin	41
Gambar 4.31 Desain tampilan halaman daftar barang keluar admin	41

Gambar 4.32 Desain tampilan halaman isi data barang keluar admin.....	42
Gambar 4.33 Desain tampilan halaman daftar barang terpakai admin	42
Gambar 4.34 Desain tampilan halaman laporan persediaan admin	43
Gambar 4.35 Desain tampilan halaman <i>stock of name</i> admin	43
Gambar 4.36 Desain tampilan halaman kelola pengguna admin.....	44
Gambar 4.37 Desain tampilan halaman isi data pengguna admin	44
Gambar 4.38 Desain tampilan halaman beranda pegawai bidang	45
Gambar 4.39 Desain tampilan halaman daftar barang terpakai pegawai bidang..	45
Gambar 4.40 Desain tampilan halaman isi data barang terpakai pegawai bidang	46
Gambar 4.41 Desain tampilan halaman ubah profil pegawai bidang	46
Gambar 4.42 Implementasi tabel <i>user</i>	47
Gambar 4.43 Implementasi tabel barang	47
Gambar 4.44 Implementasi tabel barang masuk	48
Gambar 4.45 Implementasi tabel barang keluar	48
Gambar 4.46 Implementasi tabel barang terpakai	49
Gambar 4.47 Implementasi tabel laporan	49
Gambar 4.48 Implementasi tampilan halaman <i>login</i>	50
Gambar 4.49 Implementasi tampilan halaman beranda admin.....	50
Gambar 4.50 Implementasi tampilan halaman daftar barang admin	51
Gambar 4.51 Implementasi tampilan halaman isi data barang admin	51
Gambar 4.52 Implementasi tampilan halaman daftar barang masuk admin	52
Gambar 4.53 Implementasi tampilan halaman isi data barang masuk admin.....	52
Gambar 4.54 Implementasi tampilan halaman daftar barang keluar admin	53
Gambar 4.55 Implementasi tampilan halaman isi data barang keluar admin	53
Gambar 4.56 Implementasi tampilan halaman daftar barang terpakai admin	54
Gambar 4.57 Implementasi tampilan halaman laporan persediaan admin	54
Gambar 4.58 Implementasi cetak laporan persediaan	55
Gambar 4.59 Implementasi tampilan halaman <i>stock of name</i> admin	55
Gambar 4.60 Implementasi cetak <i>stock of name</i>	56
Gambar 4.61 Implementasi tampilan halaman kelola pengguna admin	56
Gambar 4.62 Implementasi tampilan halaman isi data pengguna admin	57
Gambar 4.63 Implementasi tampilan halaman beranda pegawai bidang.....	57
Gambar 4.64 Implementasi tampilan halaman daftar barang terpakai pegawai bidang.....	58
Gambar 4.65 Implementasi tampilan halaman isi data barang terpakai pegawai bidang.....	58
Gambar 4.66 Implementasi tampilan halaman ubah profil pegawai bidang.....	59
Gambar 4.67 Grafik hasil jawaban responden	61

DAFTAR TABEL

Tabel 3.1 Simbol-simbol <i>use case diagram</i>	11
Tabel 3.2 Simbol-simbol <i>activity diagram</i>	12
Tabel 3.3 Simbol-simbol <i>class diagram</i>	13
Tabel 3.4 Simbol-simbol <i>sequence diagram</i>	14
Tabel 3.5 Simbol ERD	15
Tabel 4.1 Entitas <i>User</i>	35
Tabel 4.2 Entitas Barang.....	36
Tabel 4.3 Entitas Barang Masuk.....	36
Tabel 4.4 Entitas Barang Keluar	37
Tabel 4.5 Entitas Barang Terpakai.....	37
Tabel 4.6 Entitas Laporan	37
Tabel 4.7 Bobot nilai MOS	59
Tabel 4.8 Hasil Pengujian MOS	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini, perkembangan teknologi sangat pesat dan banyak digunakan oleh banyak bidang seperti perkantoran, pendidikan, kesehatan, instansi pemerintahan dan lain lain. Dalam instansi pemerintah, beberapa aktifitas sudah menggunakan komputer namun beberapa aktivitas seperti pengelolaan data aset barang keluar masuk dilakukan secara konvensional dengan media kertas.

Dalam perkembangannya, komputer saat ini banyak digunakan untuk membantu pekerjaan manusia. Komputer digunakan untuk mengolah, menyimpan dan mencari data dengan cepat dan efisien. Dalam hal tersebut akan mengurangi terjadinya *human error* dan informasi yang dihasilkan akan lebih tepat sehingga meningkatkan kualitas kerja suatu perusahaan maupun instansi pemerintahan.

Badan Pengelolaan Keuangan dan Aset Daerah Provinsi (BPKAD) Nusa Tenggara Barat (NTB) adalah suatu badan/institusi yang merupakan gabungan dari sub Bagian Perlengkapan pada Biro Umum Sekretariat Daerah (Setda) Provinsi Nusa Tenggara Barat dan eks Biro Keuangan Setda Provinsi Nusa Tenggara Barat. BPKAD Provinsi NTB bertugas untuk memudahkan dalam menyelesaikan permasalahan aset maupun keuangan yang kompleks. Terlebih beberapa urusan di Kabupaten/Kota akan diserahkan urusannya kepada Pemerintah Provinsi dalam pengelolaan keuangan dan aset.

Dalam pengelolaan aset di BPKAD Provinsi NTB sendiri, proses pelaporan pengelolaan, dan *monitoring* stok barang dari tahun ke tahun masih dilakukan secara manual dan kurang efektif. Hal ini berpengaruh terhadap kelengkapan dari data-data yang memiliki kemungkinan terdapat data barang yang lupa untuk didata dan hal lain yang dapat mempengaruhi kelengkapan data seperti data barang yang telah terpakai atau sisa stok barang yang ada. Dengan menggunakan sistem yang manual, akan menyulitkan pegawai dalam mengelola data-data tersebut, dimana data barang tersebut harus dicari satu persatu sehingga akan membutuhkan waktu yang sangat lama. Dengan adanya permasalahan tersebut, maka dibuatlah “Sistem Informasi Manajemen Aset Pada Badan Pengelolaan Keuangan dan Aset Daerah

Provinsi Nusa Tenggara Barat” yang diharapkan dapat membantu pegawai BPKAD Provinsi NTB mengelola data-data tersebut agar lebih efisien.

1.2 Rumusan Masalah

Dari latar belakang yang telah dipaparkan dapat dirumuskan masalah sebagai berikut: “Bagaimana cara membuat Sistem Informasi Manajemen Aset Pada Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat agar dapat mempermudah pegawai Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat dalam mengelola aset barang?”

1.3 Batasan Masalah

Batasan masalah dari Sistem Informasi Manajemen Aset ini adalah:

1. Sistem informasi yang dibuat merupakan sistem informasi berbasis *web* yang dapat mencatat pengelolaan aset.
2. Aplikasi ini ditujukan untuk pegawai Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat.

1.4 Tujuan

Tujuan pembuatan Sistem Informasi Manajemen Aset yaitu membangun sistem informasi yang dapat mempermudah pencatatan pengelolaan aset barang di Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat

1.5 Manfaat

Manfaat dari pembuatan Sistem Informasi Manajemen Aset ini adalah:

1. Bagi Mahasiswa
 - a. Memberikan pengalaman kerja terhadap mahasiswa.
 - b. Mahasiswa harus dapat menyelesaikan masalah yang diberikan berdasarkan ilmu-ilmu yang telah dipelajari pada saat kuliah.
2. Bagi Instansi
 - a. Dapat mempermudah pegawai Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat dalam pencatatan pengelolaan aset.
 - b. Dapat menghemat ruang dari segi tempat maupun biaya.

BAB II

TINJAUAN INSTANSI TEMPAT PKL

2.1 Profil Singkat BPKAD Provinsi Nusa Tenggara Barat

BPKAD Provinsi NTB merupakan Satuan Kerja Perangkat Daerah baru yang dibentuk berdasarkan Peraturan Gubernur Nomor 22 Tahun 2015 tentang Perubahan Kedua Atas Peraturan Gubernur Nomor 22 Tahun 2008 tentang Rincian Tugas, Fungsi dan Tata Kerja Inspektorat, bappeda dan Lembaga Teknis Daerah Provinsi Nusa Tenggara Barat dimana BPKAD Provinsi NTB merupakan bagian daripada Lembaga Teknis Daerah.

Pada awal tahun 2017 BPKAD dikukuhkan kembali dengan nomenklatur yang baru berdasarkan PP 18 Tahun 2016 tentang Perangkat Daerah dan Perda Nomor 11 Tahun 2016 tentang Pembentukan dan Susunan Perangkat Daerah Provinsi Nusa Tenggara Barat sehingga Gubernur Nusa Tenggara Barat mengeluarkan Pergub Nomor 51 Tahun 2016 tentang Kedudukan, Susunan Organisasi, Tugas dan Fungsi Serta Tata Kerja Badan-Badan Daerah Provinsi Nusa Tenggara Barat yang didalamnya terdapat Susunan Organisasi BPKAD Provinsi NTB.

BPKAD Provinsi NTB dalam ketentuan yang baru juga membawahi dua buah Unit Pelayanan Terpadu Badan (UPTB) dengan dikeluarkannya Pergub Nomor 53 Tahun 2016 tentang Pembentukan, Kedudukan, Susunan Organisasi, Tugas dan Fungsi Serta Tata Kerja Unit Pelaksana Teknis Daerah (UPTD) pada Dinas-Dinas Daerah dan UPTB pada Badan-Badan Daerah Provinsi Nusa Tenggara Barat.

Badan Pengelolaan Keuangan Daerah Provinsi Nusa Tenggara Barat atau yang lebih dikenal dengan BPKAD Provinsi NTB merupakan gabungan dari eks Bagian Perlengkapan pada Biro Umum Setda Provinsi Nusa Tenggara Barat dan eks Biro Keuangan Setda Provinsi Nusa Tenggara Barat. Diharapkan dengan penggabungan tersebut dapat lebih memudahkan dalam menyelesaikan permasalahan aset maupun keuangan yang sangat kompleks. Terlebih beberapa urusan di Kabupaten/Kota akan diserahkan urusannya kepada Pemerintah Provinsi dalam pengelolaan keuangan dan aset.

2.2 Visi dan Misi BPKAD Provinsi Nusa Tenggara Barat

Adapun visi dan misi dari Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat (BPKAD Provinsi NTB) adalah sebagai berikut:

Visi:

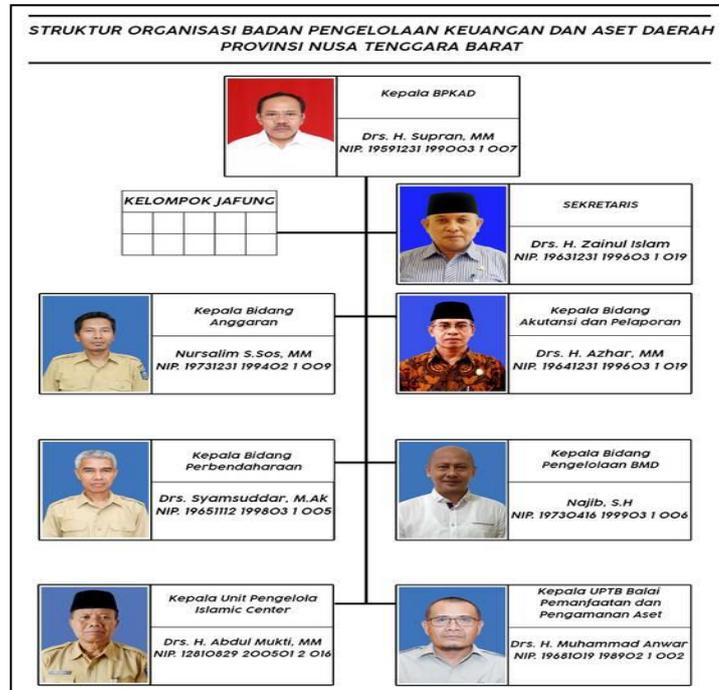
“Menjadi Institusi Pengelola Keuangan dan Aset Daerah Terbaik.”

Misi:

1. Mengikhtiarkan tetap WTP Laporan Keuangan Pemerintah Provinsi NTB setiap tahunnya.
2. Melaksanakan fungsi sebagai PPKD dengan penuh Amanah dan Profesional.
3. Melaksanakan fungsi pengelolaan BMD dengan tertib, dan optimal dalam peningkatan PAD.
4. Mengefektifkan pelaksanaan APBD untuk pencapaian Visi NTB yang Beriman, Berbudaya, Berdaya Saing dan Sejahtera.
5. Penguatan institusi dan sumberdaya manusia BPKAD yang berkualitas dan melayani.
6. Menyelaraskan dan mensinergikan pelaksanaan APBN, APBD Provinsi dan APBD Kabupaten/Kota se-NTB.
7. Melakukan percepatan penyelesaian tindak lanjut laporan hasil pemeriksaan APBD.

2.3 Struktur Organisasi BPKAD Provinsi Nusa Tenggara Barat

Berikut merupakan struktur organisasi BPKAD Provinsi Nusa Tenggara Barat:



Gambar 2.1 Struktur organisasi BPKAD Provinsi NTB

Gambar 2.1 menunjukkan struktur organisasi BPKAD Provinsi NTB. Struktur organisasi terdiri dari Kepala BPKAD, Sekretaris, dan 6 Kepala Bidang.



Gambar 2.2 Struktur organisasi BPKAD Provinsi NTB bidang sekretariat

Gambar 2.2 menunjukkan struktur organisasi BPKAD Provinsi NTB bidang sekretariat. Struktur organisasi terdiri dari Sekretaris dengan 3 bawahan Kepala Sub Bagian.



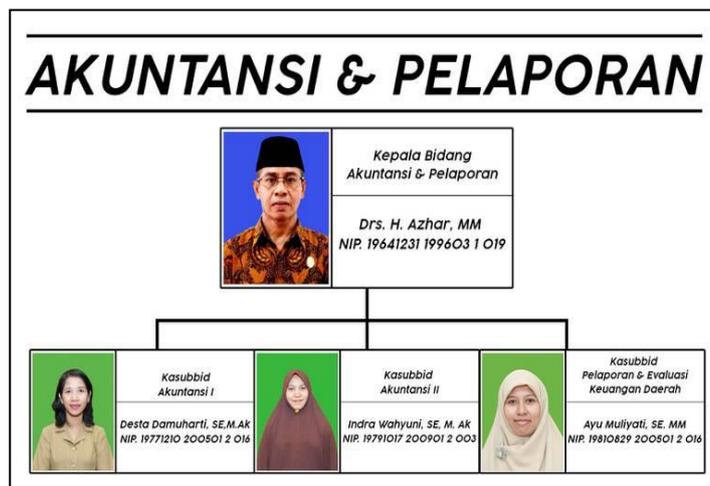
Gambar 2.3 Struktur organisasi BPKAD Provinsi NTB bidang anggaran

Gambar 2.3 menunjukkan struktur organisasi BPKAD Provinsi NTB bidang Anggaran. Struktur organisasi terdiri dari Kepala Bidang Anggaran dengan 3 bawahan Kepala Sub Bidang Anggaran.



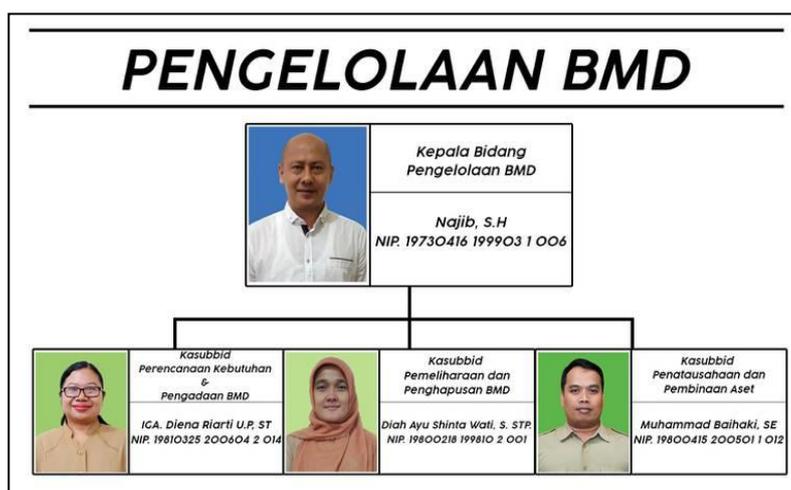
Gambar 2.4 Struktur organisasi BPKAD Provinsi NTB bidang perbendaharaan

Gambar 2.4 menunjukkan struktur organisasi BPKAD Provinsi NTB bidang Perbendaharaan. Struktur organisasi terdiri dari Kepala Bidang Perbendaharaan dengan 3 bawahan Kepala Sub Bidang Perbendaharaan.



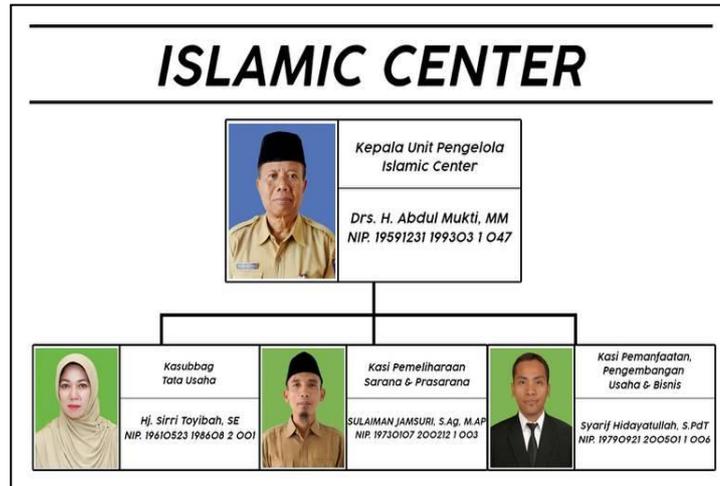
Gambar 2.5 Struktur organisasi BPKAD Provinsi NTB bidang akuntansi dan pelaporan

Gambar 2.5 menunjukkan struktur organisasi BPKAD Provinsi NTB bidang Akuntansi dan Pelaporan. Struktur organisasi terdiri dari Kepala Bidang Akuntansi dan Pelaporan dengan 3 bawahan Kepala Sub Bidang.



Gambar 2.6 Struktur organisasi BPKAD Provinsi NTB bidang pengelolaan BMD

Gambar 2.6 menunjukkan struktur organisasi BPKAD Provinsi NTB bidang Pengelolaan BMD. Struktur organisasi terdiri dari Kepala Bidang Pengelolaan BMD dengan 3 bawahan Kepala Sub Bidang.



Gambar 2.7 Struktur organisasi BPKAD Provinsi NTB bidang Islamic Center

Gambar 2.7 menunjukkan struktur organisasi BPKAD Provinsi NTB bidang Islamic Center. Struktur organisasi terdiri dari Kepala Bidang Islamic Center dengan 1 bawahan Kepala Sub Bagian dan 2 bawahan Kepala Seksi.



Gambar 2.8 Struktur organisasi BPKAD Provinsi NTB bidang UPTB

Gambar 2.8 menunjukkan struktur organisasi BPKAD Provinsi NTB bidang UPTB Balai Pemanfaatan dan Pengamanan Aset. Struktur organisasi terdiri dari Kepala Bidang UPTB Balai Pemanfaatan dan Pengamanan Aset dengan 1 bawahan Kepala Sub Bagian dan 2 bawahan Kepala Seksi.

BAB III

DASAR TEORI

3.1 Sistem Informasi

Sistem informasi adalah sistem yang disusun oleh satu atau beberapa sistem pengolahan data meliputi data, informasi, pengetahuan, dan kebijaksanaan [1]. Dalam sistem informasi terdapat proses atau aktivitas aktivitas yang akan mendukung aktivitas usaha dalam sistem organisasi. Proses dari sistem informasi antara lain mengumpulkan data (*data gathering*), mengolah data yang tersimpan, menyebarkan informasi.

Di dalam sistem informasi, manusia berinteraksi dengan manusia, manusia dengan komputer, dan komputer berinteraksi dengan komputer lain dimana data, informasi dan/atau pengetahuan mengalir dibawah media elektronik jaringan komputer. Keberadaan sistem informasi diperlukan dalam organisasi untuk mendampingi proses-proses bisnis dari organisasi.

3.2 Manajemen Aset

Manajemen aset didefinisikan sebagai sebuah proses pengelolaan aset (kekayaan) baik berwujud dan tidak berwujud yang memiliki nilai ekonomis, nilai komersial, dan nilai tukar, serta mampu mendorong tercapainya tujuan. Secara umum, siklus pengelolaan aset adalah tahapan yang harus dilalui dalam manajemen aset [2].

Setiap organisasi swasta maupun organisasi pemerintah tentu memiliki aset, baik yang berwujud (*tangible*) maupun tidak berwujud (*intangible*). Setiap aset yang dimiliki harus dikelola dengan efektif dan efisien sehingga aset tersebut dapat memberikan manfaat bagi sektor swasta maupun publik. Peran penting manajemen aset, baik di sektor swasta maupun publik, semakin diakui dan tidak kalah dengan peran manajemen keuangan dan manajemen sumber daya manusia [2].

3.3 Website

Website adalah kumpulan halaman-halaman yang digunakan untuk mempublikasikan informasi berupa teks, gambar dan program multimedia lainnya berupa animasi (gambar gerak, tulisan gerak), suara dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu

rangkaian bangunan yang saling terkait antara satu *page* dengan *page* yang lain yang sering disebut sebagai *hyperlink* [3]. *Website* biasanya terangkum dalam sebuah domain atau subdomain yang tempatnya berada di dalam *World Wide Web* (www) di internet. Sebuah *Web Page* adalah dokumen yang ditulis dalam format HTML yang hampir selalu bisa diakses melalui HTTP, yaitu *protocol* yang bisa menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Halaman-halaman dari *website* akan bisa diakses melalui sebuah URL yang bisa disebut *Homepage*. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun *hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan. Beberapa *website* membutuhkan subskripsi (data masukan) agar para *user* bisa mengakses sebagian atau keseluruhan isi *web* tersebut. Contohnya, ada beberapa situs-situs bisnis, situs-situs *email* gratisan yang membutuhkan subskripsi agar kita bisa mengakses situs tersebut [4].

3.4 UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang *Model* sebuah sistem [4].

Dengan menggunakan UML kita dapat membuat *Model* untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *Modeling* aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantik*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat

dikombinasikan. Notasi UML terutama diturunkan dari 3(tiga) notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

3.5 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [5]. Tabel 3.1 menunjukkan simbol-simbol yang biasa digunakan untuk membuat *use case diagram*.

Tabel 3.1 Simbol-simbol *use case diagram*

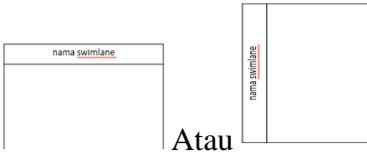
Simbol	Nama	Keterangan
	<i>Actor</i>	<i>Actor</i> tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
	<i>Use Case</i>	Gambaran fungsionalitas dari suatu sistem, sehingga <i>customer</i> atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
	<i>Association</i>	Menghubungkan <i>link</i> antar <i>element</i> .
	<i>Include</i>	Kejadian yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
	<i>Generalization</i>	Disebut juga <i>inheritance</i> (pewarisan).

3.6 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin

terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah aksi dan sebagian besar transisi dibangkitkan oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum [5]. Tabel 3.2 menunjukkan simbol-simbol yang biasa digunakan untuk membuat *activity diagram*.

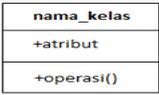
Tabel 3.2 Simbol-simbol *activity diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>Decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>Join</i> 	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas harus memiliki status akhir.
<i>Swimlane</i>  Atau	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

3.7 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain [5]. Tabel 3.3 menunjukkan simbol-simbol yang biasa digunakan untuk membuat *class diagram*.

Tabel 3.3 Simbol-simbol *class diagram*

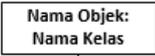
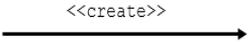
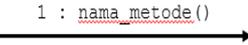
Simbol	Nama	Deskripsi
	<i>Class</i>	Kelas pada struktur sistem.
	Asosiasi / <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Asosiasi berarah / <i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
	Kebergantungan / <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
	Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

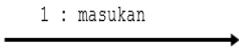
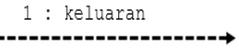
3.8 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa

digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang membangkitkan aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan [5]. Tabel 3.4 menunjukkan simbol-simbol yang biasa digunakan untuk membuat *sequence diagram*.

Tabel 3.4 Simbol-simbol *sequence diagram*

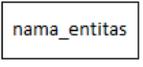
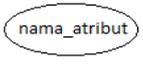
Simbol	Deskripsi
<p>Aktor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di bawa frase nama aktor.</p>
<p><i>Lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>
<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>

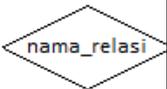
<p>Pesan tipe <i>send</i></p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>

3.9 ERD

ERD (*Entity Relation Diagram*) merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data. ERD (*Entity Relationship Diagram*) berguna untuk menggambarkan gambaran dari dunia nyata yang akan diterapkan pada suatu *database* sebuah sistem. ERD melihat objek nyata dapat sebagai sebuah entitas - entitas yang memiliki relasi antara entitas yang satu ataupun yang lain [6]. Dengan ERD sendiri dapat membantu mengurangi – kesalahan dalam melakukan perancangan *database* dari gambaran dunia nyata dan struktur *database* seperti redundansi data, hubungan – hubungan antara entitas dan lain sebagainya. Tabel 3.5 menunjukkan simbol-simbol yang biasa digunakan untuk membuat ERD.

Tabel 3.5 Simbol ERD

Simbol	Nama	Keterangan
	Entitas / <i>Entity</i>	Merupakan data inti yang akan disimpan; bakal tabel pada basis data; biasanya mengacu pada benda yang terlibat dalam aplikasi yang akan dibuat.
	Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam satu entitas.
	Atribut kunci primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas sebagai kunci akses <i>record</i> yang diinginkan.

	Relasi	Penghubung antar entitas; biasanya diawali dengan kata kerja.
	Asosiasi / <i>Association</i>	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian.

3.10 Framework Laravel

Framework adalah kumpulan instruksi-instruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam pemanggilannya tanpa harus menuliskan *syntax* program yang sama berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang agar tercipta nya *source code* yang bersih dan terstruktur.

Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*Model View Controller*). Laravel adalah pengembangan website berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu *Model*, *View* dan *Controller*.

3.11 Konsep MVC

MVC merupakan singkatan dari *Model*, *View*, dan *Controller*. MVC adalah suatu metode yang memisahkan *data logic* (*Model*) dari *presentation logic* (*View*) dan *process logic* (*Controller*) atau secara sederhana adalah memisahkan antara desain *interface*, data, dan proses. Berikut adalah penjelasan dari ketiga komponen MVC [7]:

1. *Model*

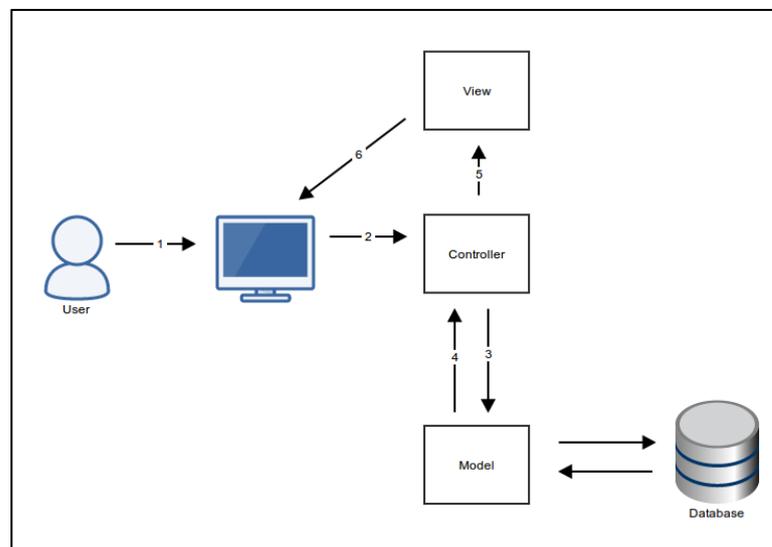
Model merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi *database*. seperti misalnya mengambil data dari *database*, meng-*input* dan pengolahan *database* lainnya. semua instruksi yang berhubungan dengan pengolahan *database* diletakkan di dalam *model* [7].

2. *View*

View merupakan bagian yang menangani halaman *user interface* atau halaman yang muncul pada *user*. tampilan dari *user interface* dikumpulkan pada *view* untuk memisahkannya dengan *controller* dan *model* sehingga memudahkan *web designer* dalam melakukan pengembangan tampilan halaman *website* [7].

3. *Controller*

Controller merupakan kumpulan instruksi aksi yang menghubungkan *model* dan *view*, jadi *user* tidak akan berhubungan dengan *model* secara langsung, intinya dari *view* kemudian *controller* yang mengolah instruksi [7].



Gambar 3.1 Konsep MVC

Gambar 3.1 menunjukkan arsitektur dari konsep MVC. Dimana *user* mengirim perintah yang akan masuk ke *Controller* lalu dari *Controller* data masuk ke *Model* untuk melalui pengecekan di *database* lalu data kembali ke *Controller* untuk menuju *View* sebelum ditampilkan ke *user*.

3.12 Visual Studio Code

Visual Studio Code adalah sebuah *software* aplikasi/*editor text* untuk Bahasa pemrograman multi Bahasa termasuk pemrograman PHP dan Javascript yang akan digunakan untuk mengembangkan sistem, Visual Studio Code mendukung banyak bahasa pemrograman dan bahasa *markup* serta fungsinya dapat ditambah dengan *plugin*.

Kelebihan Visual Studio Code yaitu:

- a. *Multiple selection*

- b. *Command palletete*
- c. *Distraction free mode*
- d. *Find in project*
- e. *Multi platform*

3.13 MySql

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. SQL juga dapat diartikan sebagai antarmuka standar untuk sistem manajemen basis data relasional, termasuk sistem yang beroperasi pada komputer pribadi [6]. SQL memungkinkan seorang pengguna untuk mengakses informasi tanpa mengetahui di mana lokasinya atau bagaimana informasi tersebut disusun.

MySQL merupakan DBMS *open source* (kode programnya dapat di-*download* gratis) dan *freeware* (gratis dipakai), walau gratis MySQL tidak bisa dikatakan DBMS jelek karena gratis, hal ini terbukti dengan adanya sebuah operator telepon seluler yang menggunakan DBMS ini dan memiliki sistem yang andal dengan optimasi. Sampai saat ini MySQL masih menjadi DBMS *open source* paling banyak digunakan di dunia. MySQL terkenal dengan kecepatan dalam pengaksesan datanya. MySQL saat ini dimiliki oleh perusahaan *Oracle*.

SQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multi thread*, dan *multi-user*. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

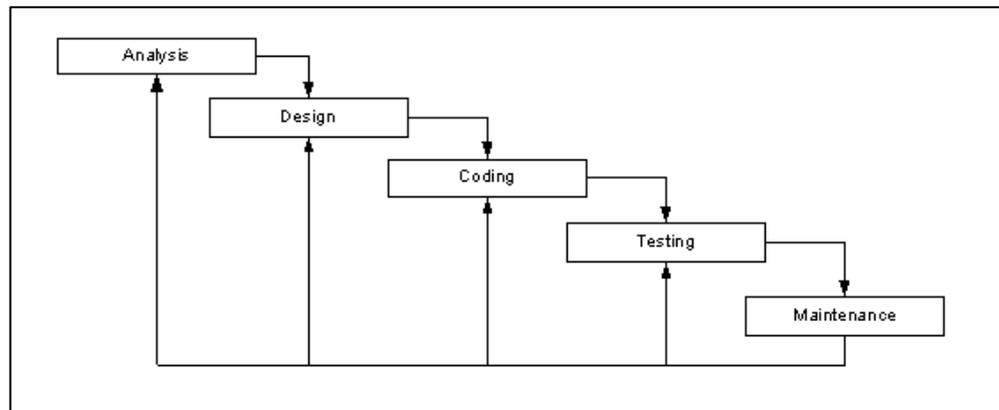
MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*), dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial.

BAB IV

PEMBAHASAN

4.1 Metode Pengembangan Sistem

Pengembangan perangkat lunak dilakukan menggunakan kaidah *software engineering* (rekayasa perangkat lunak). Dalam teori *software engineering* terdapat beberapa macam *model* proses pengembangan perangkat lunak. Penelitian ini menggunakan *model waterfall*. *Model waterfall* atau yang sering disebut *model classic life cycle* menunjukkan pengembangan perangkat lunak secara berurutan dan sistematis, seperti Gambar 4.1.



Gambar 4.1 Metode *waterfall*

4.2 Analisa kebutuhan

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini menggunakan metode wawancara, diskusi dan survei langsung. Kemudian informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

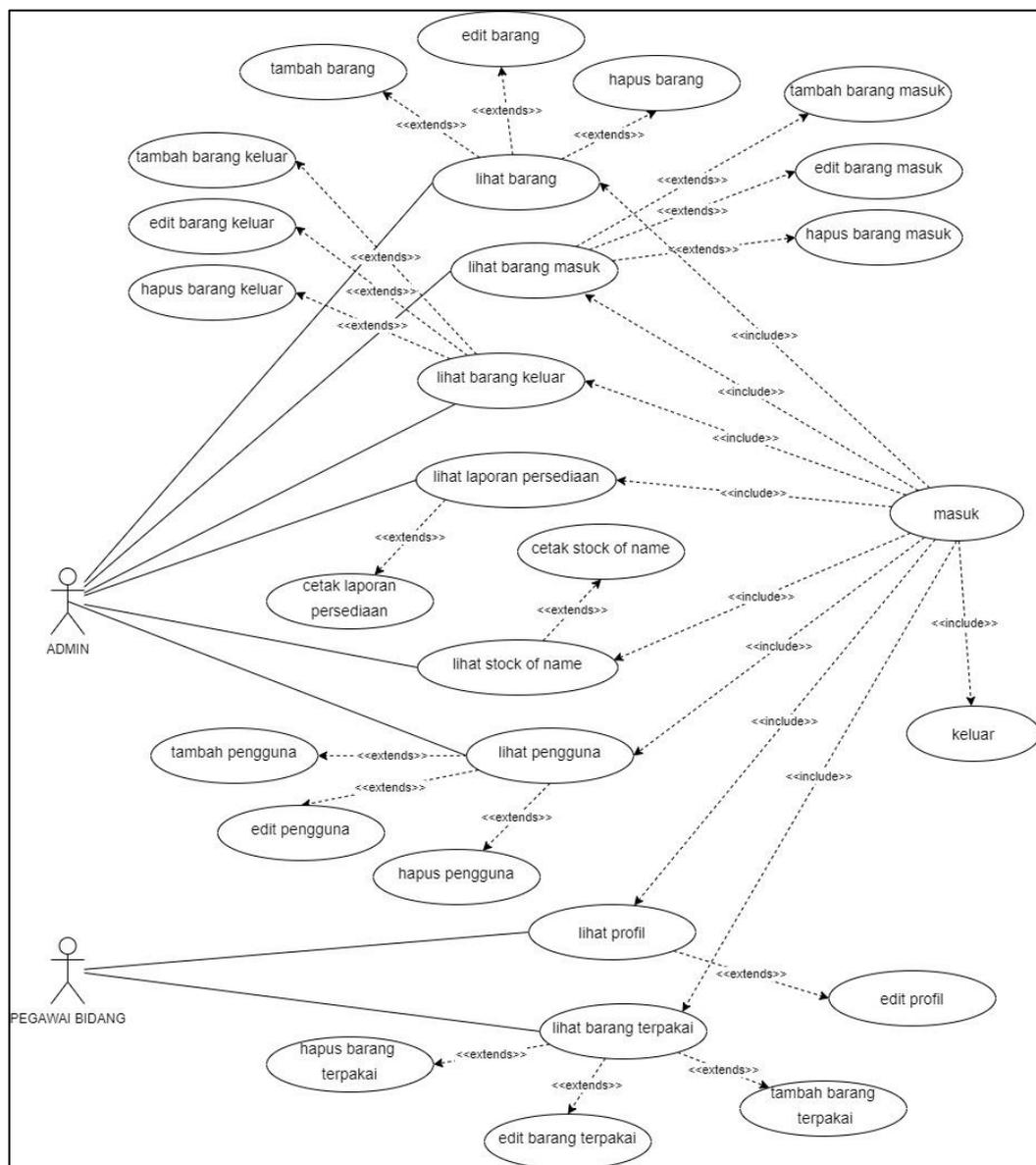
4.3 Desain Sistem

Dalam sebuah *project* yang bersifat OOP (*Object Oriented Programming*) perancangan sistem yang seperti *class diagram*, *use case diagram*, *activity diagram* dan *sequence diagram* harus ada. Untuk membuat perancangan sistem dilakukan analisis terhadap sistem yang akan dibuat. Hal yang pertama kali dilakukan adalah menganalisis *Use case diagram* untuk interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri. Selanjutnya menganalisis *activity diagram* pada sebuah sistem yang akan dirancang agar alur dari kerja suatu sistem dari awal hingga akhir

dapat dimengerti oleh perancang sistem. Setelah itu buat perancangan untuk *sequence* diagram yang harus sesuai dengan *use case* yang ada pada sistem. Kemudian merancang *class* diagram untuk menjelaskan struktur dari program yang akan dibuat, biasanya *class* diagram dirancang pada sistem yang menggunakan konsep OOP. Perancangan dari segi *database* menggunakan penggambaran ERD juga perlu, untuk menggambarkan data-data yang ada dalam sistem.

4.3.1 Use Case Diagram

Berikut merupakan *use case diagram* dari Sistem Informasi Manajemen Aset BPKAD:



Gambar 4.2 Use case diagram

Pada Gambar 4.2, terdapat 2 (dua) jenis *user* yang menggunakan sistem informasi ini, diantaranya adalah:

1. Admin

Berfungsi sebagai *admin* yang dapat melakukan:

- a. Melihat data barang
- b. Menambah data barang
- c. Mengubah data barang
- d. Menghapus data barang
- e. Melihat data barang masuk
- f. Menambah data barang masuk
- g. Mengubah data barang masuk
- h. Menghapus data barang masuk
- i. Melihat data barang keluar
- j. Menambah data barang keluar
- k. Mengubah data barang keluar
- l. Menghapus data barang keluar
- m. Melihat data barang terpakai
- n. Mengubah data barang terpakai
- o. Menghapus data barang terpakai
- p. Melihat laporan persediaan
- q. Mencetak laporan persediaan
- r. Melihat *stock of name*
- s. Mencetak *stock of name*
- t. Melihat data pengguna
- u. Menambah data pengguna
- v. Mengubah data pengguna
- w. Menghapus data pengguna

2. Pegawai Bagian Umum merupakan *user* yang dapat melakukan:

- a. Melihat data barang terpakai
- b. Menambah data barang terpakai
- c. Mengubah data barang terpakai
- d. Menghapus data barang terpakai

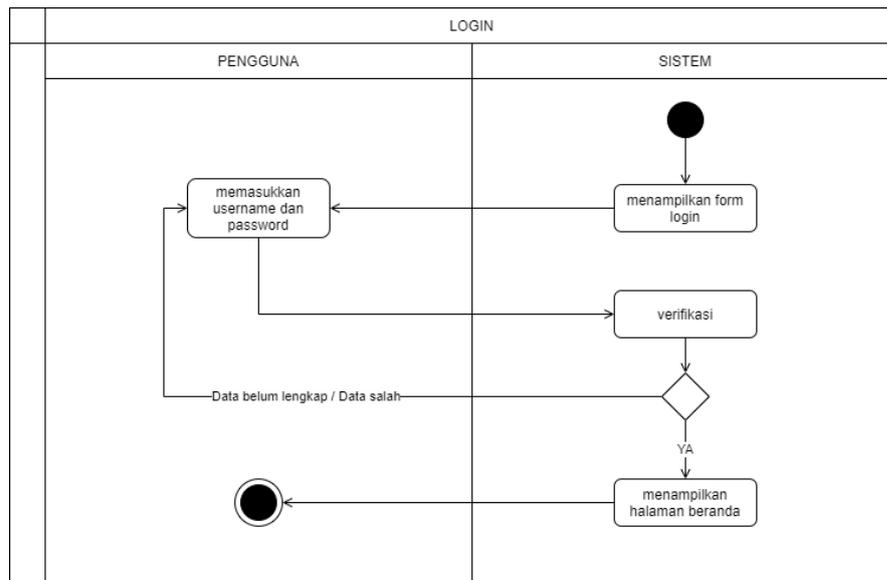
- e. Melihat profil
- f. Mengubah profil

Sebelum semua *user* dapat menggunakan fitur tersebut harus melakukan masuk ke dalam sistem dengan memasukkan *username* dan *password*.

4.3.2 Activity Diagram

Berikut merupakan *activity diagram* dari Sistem Manajemen Aset BPKAD:

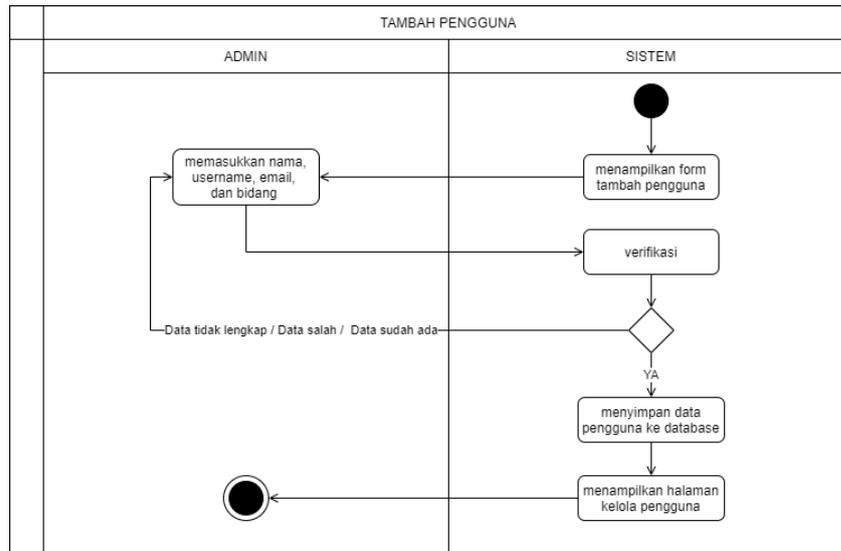
1. Proses Login



Gambar 4.3 Activity diagram Login

Gambar 4.3 merupakan *activity diagram* dari proses *login* yang dimulai dari sistem menampilkan *form login*. Kemudian pengguna memasukkan *username* dan *password*. Setelah itu, sistem memverifikasi data yang telah dimasukkan, apabila data sudah lengkap dan sudah valid maka sistem akan menampilkan halaman beranda. Apabila data tidak ditemukan atau data yang diisi salah maka pengguna harus memasukkan ulang data *username* dan *password* yang sesuai.

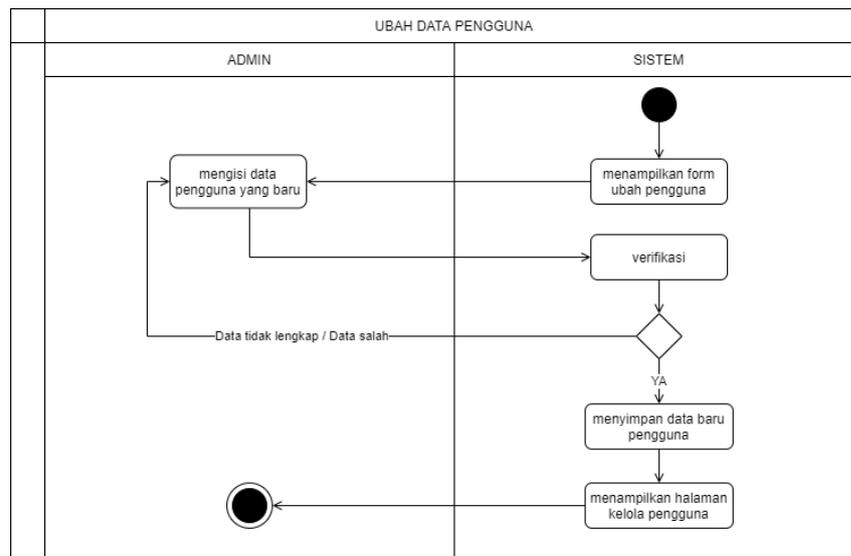
2. Proses menambahkan pengguna oleh admin



Gambar 4.4 Activity diagram tambah pengguna oleh admin

Gambar 4.4 merupakan *activity diagram* dari proses tambah pengguna. Proses ini berawal dari sistem yang menampilkan *form* tambah pengguna kemudian admin mengisi data pengguna yaitu nama, *username*, *email*, dan bidang. Data yang telah dimasukkan akan diverifikasi. Jika tidak terjadi duplikasi data, data telah benar dan terisi semua maka data pengguna disimpan ke dalam *database* dan jika tidak, maka admin harus melengkapi data pengguna atau memperbaiki data tersebut. Setelah menyimpan data pengguna ke *database*, selanjutnya sistem menampilkan halaman kelola pengguna.

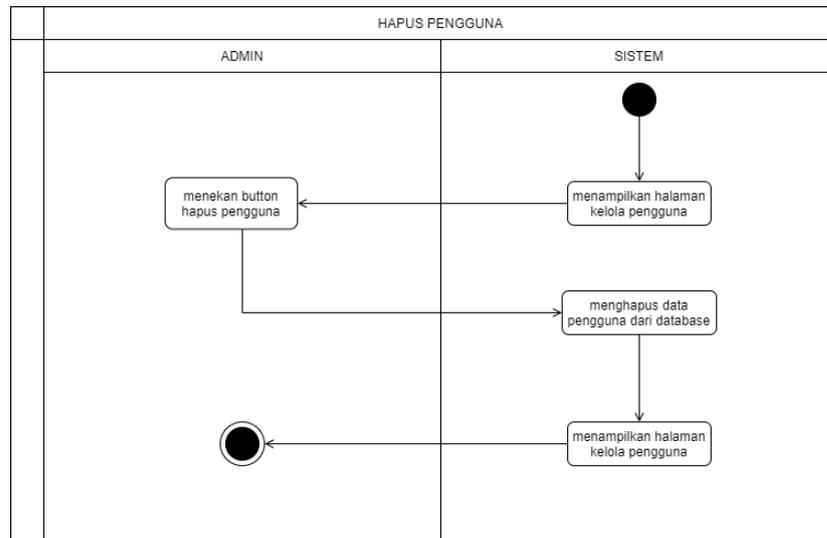
3. Proses mengubah pengguna oleh admin



Gambar 4.5 Activity diagram ubah pengguna oleh admin

Gambar 4.5 merupakan *activity diagram* dari proses ubah data pengguna. Proses ini berawal dari sistem menampilkan *form* ubah pengguna, lalu admin mengisi data baru pengguna yang ingin diubah. Data yang telah diubah diverifikasi, apabila data tersebut telah lengkap dan benar, maka data disimpan ke *database* dan sistem menampilkan halaman kelola pengguna. Apabila data tidak lengkap atau salah, maka admin kembali memasukkan data pengguna yang benar dan lengkap.

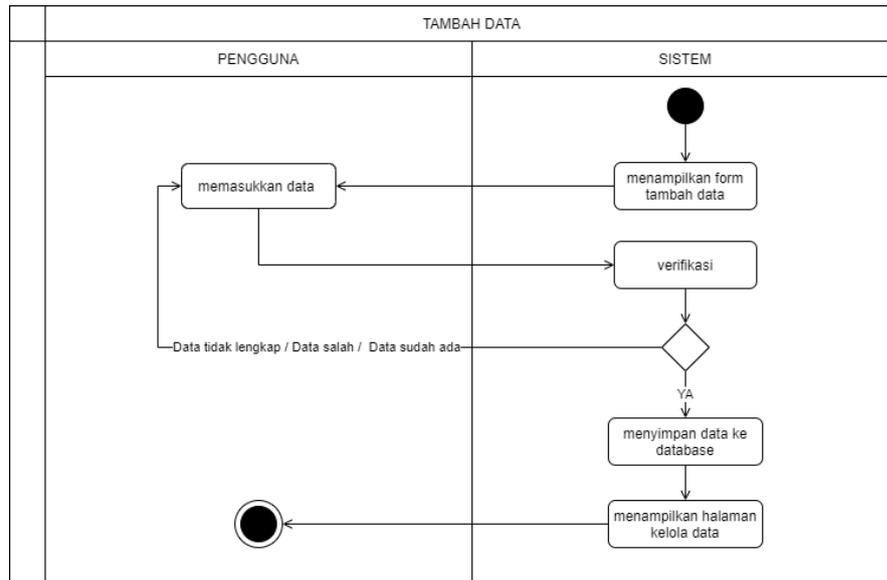
4. Proses menghapus pengguna oleh admin



Gambar 4.6 Activity diagram hapus pengguna oleh admin

Gambar 4.6 merupakan *activity diagram* dari proses menghapus pengguna. Proses ini berawal dari sistem menampilkan halaman kelola pengguna. Admin menekan tombol hapus pada pengguna yang dipilih di halaman kelola pengguna. Setelah itu, data pengguna akan dihapus dari *database* dan sistem menampilkan halaman kelola pengguna.

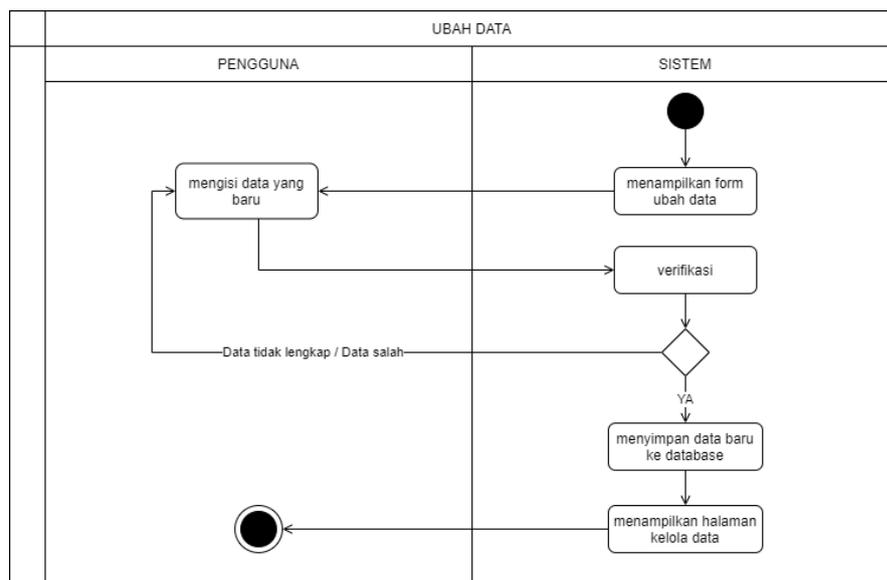
5. Proses menambahkan data



Gambar 4.7 Activity diagram tambah data

Gambar 4.7 merupakan *activity diagram* dari proses tambah pengguna. Proses ini berawal dari sistem yang menampilkan *form* tambah data, kemudian pengguna mengisi data sesuai yang dibutuhkan. Data yang telah dimasukkan akan diverifikasi. Jika tidak terjadi duplikasi data, data telah benar dan terisi semua maka data disimpan ke dalam *database* dan jika tidak, maka pengguna harus melengkapi data atau memperbaiki data tersebut. Setelah menyimpan data ke dalam *database*, selanjutnya sistem menampilkan halaman kelola data.

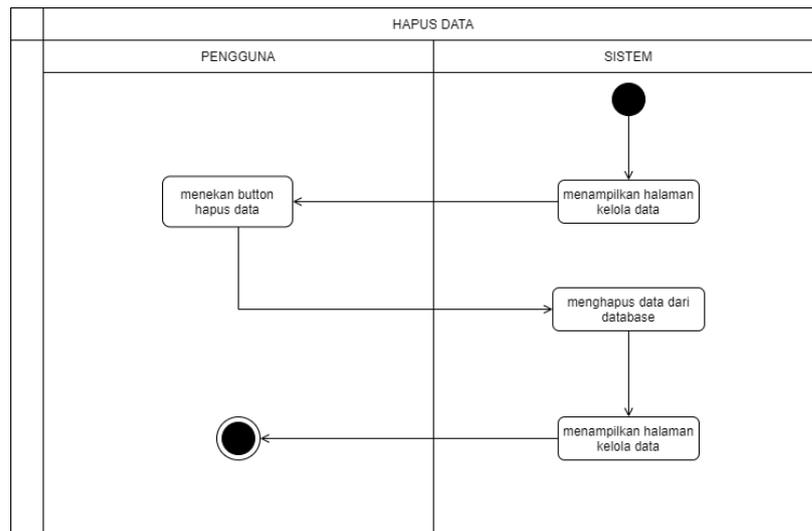
6. Proses mengubah data



Gambar 4.8 Activity diagram ubah data

Gambar 4.8 merupakan *activity diagram* dari proses ubah data. Proses ini berawal dari sistem menampilkan *form* ubah data, lalu pengguna mengisi data baru yang ingin diubah. Data yang telah diubah diverifikasi, apabila data tersebut telah lengkap dan benar, maka data disimpan ke dalam *database* dan sistem menampilkan halaman kelola data. Apabila data tidak lengkap atau salah, maka pengguna kembali memasukkan data yang benar dan lengkap.

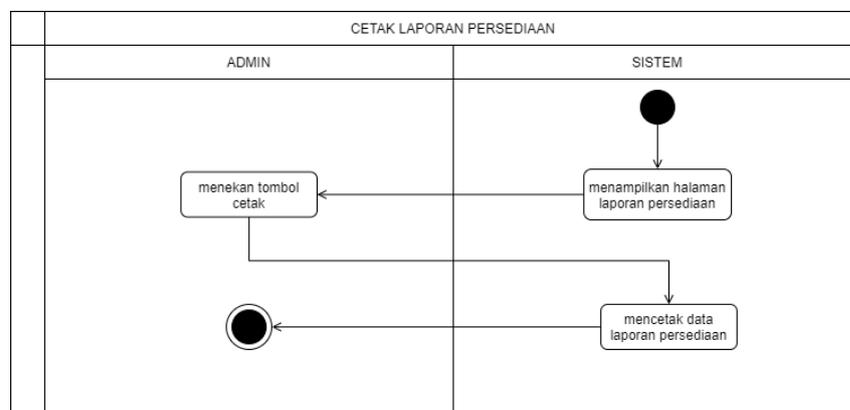
7. Proses menghapus data



Gambar 4.9 Activity diagram hapus data

Gambar 4.9 merupakan *activity diagram* dari proses menghapus data. Proses ini berawal dari sistem menampilkan halaman kelola data. Pengguna menekan tombol hapus pada data yang dipilih di halaman kelola data. Setelah itu, data yang terpilih akan dihapus dari *database* dan sistem menampilkan halaman kelola data.

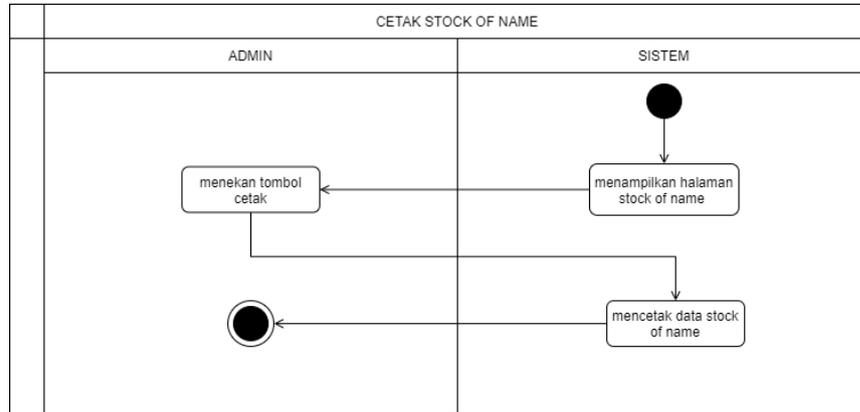
8. Proses mencetak laporan persediaan oleh admin



Gambar 4.10 Activity diagram cetak laporan persediaan oleh admin

Gambar 4.10 merupakan *activity diagram* dari proses mencetak laporan persediaan. Pengguna dapat mencetak laporan persediaan dengan menekan tombol cetak pada halaman tersebut, maka data laporan persediaan akan langsung tercetak.

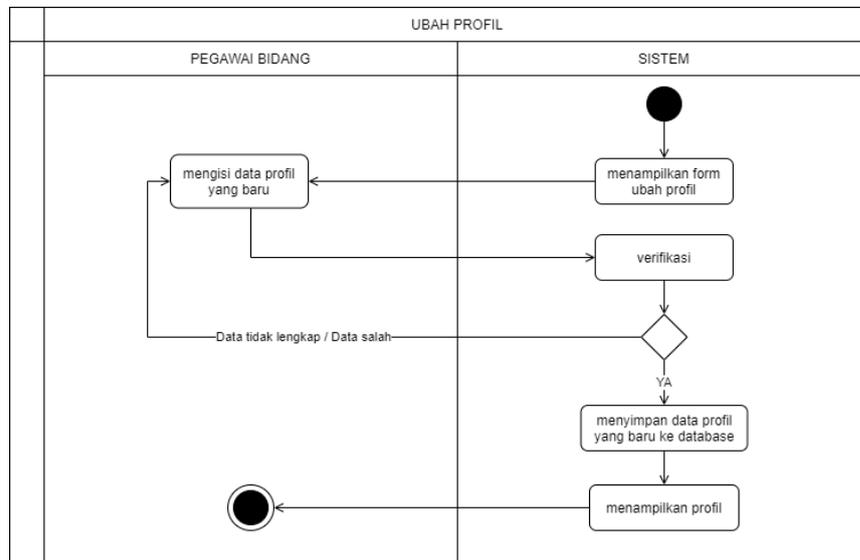
9. Proses mencetak *stock of name* oleh admin



Gambar 4.11 *Activity diagram* cetak *stock of name* oleh admin

Gambar 4.11 merupakan *activity diagram* dari proses mencetak *stock of name*. Pengguna dapat mencetak *stock of name* dengan menekan tombol cetak pada halaman tersebut, maka data *stock of name* akan langsung tercetak.

10. Proses mengubah profil oleh pegawai bidang



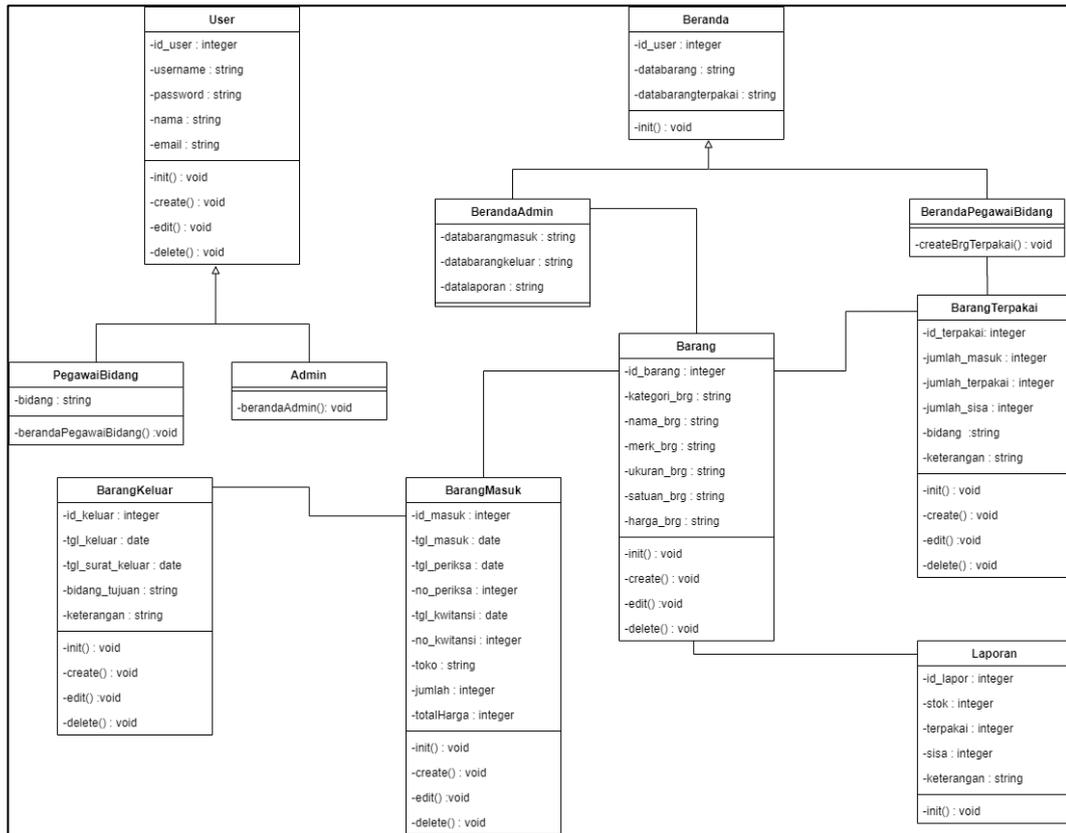
Gambar 4.12 *Activity diagram* ubah profil

Gambar 4.12 merupakan *activity diagram* dari proses mengubah profil. Aktivitas mengubah profil dapat dilakukan oleh pegawai bidang dengan sistem akan menampilkan form ubah profil. Pada form tersebut, pegawai bidang akan

mengisi data yang baru seperti nama, *email*, *username*, dan *password*. Setelah dilakukan verifikasi kesesuaian data maka profil pegawai bidang yang baru tersebut dapat disimpan dalam *database*.

4.3.3 Class Diagram

Class Diagram berikut ini menggambarkan hubungan antara relasi dari setiap kelas yang ada pada Sistem Informasi Manajemen Aset BPKAD.



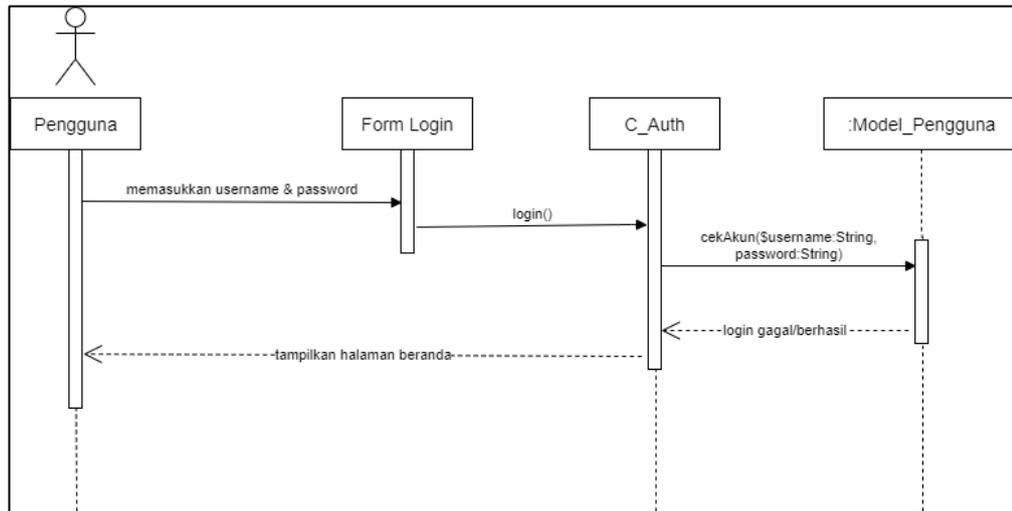
Gambar 4.13 *Class diagram*

Pada Gambar 4.13 dapat dilihat keterhubungan antar kelas yang terdapat pada Sistem Informasi Manajemen Aset BPKAD. Sistem ini menggunakan metode analisa dan perancangan berorientasi objek, sehingga dilakukan perancangan kelas – kelas yang ada pada sistem yang terdiri dari kelas-kelas yang ada.

4.3.4 Sequence Diagram

Berikut merupakan *sequence diagram* dari Sistem Informasi Manajemen Aset BPKAD :

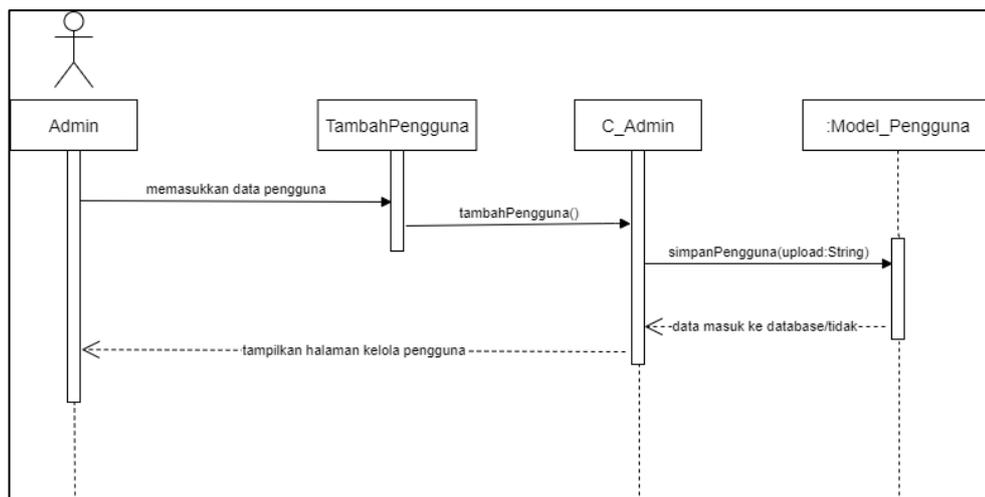
1. Login



Gambar 4.14 Sequence diagram login

Gambar 4.14 merupakan *sequence diagram* dari proses *login*. Proses ini diawali dari *form login*, pengguna memasukkan *username* dan *password* lalu memanggil fungsi `login()` pada *Controller Auth*. Kemudian memanggil fungsi `cekAkun()` pada *Model Model_Pengguna*. Dari *Model*, mengirimkan nilai *login* berhasil atau gagal. Apabila berhasil, *Controller* akan menampilkan halaman beranda.

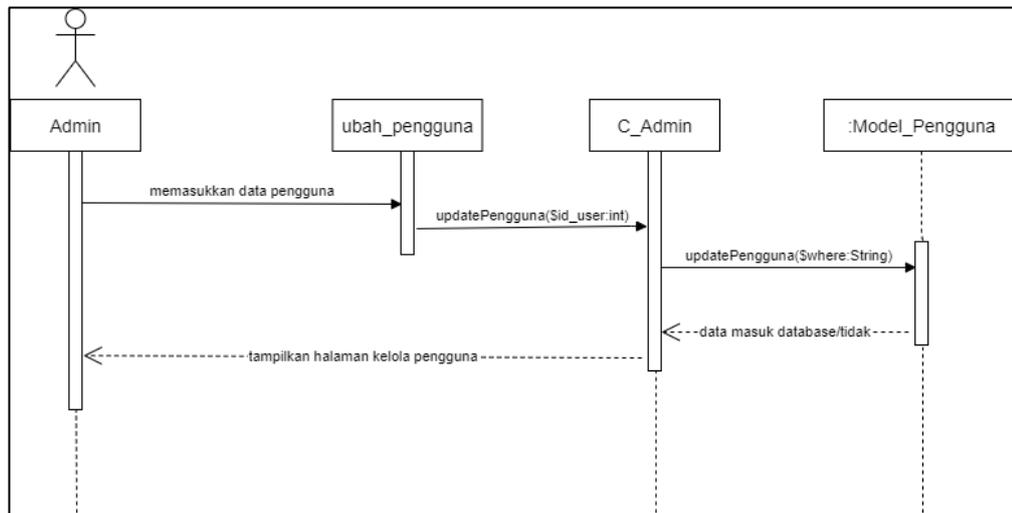
2. Menambahkan pengguna



Gambar 4.15 Sequence diagram tambah pengguna

Gambar 4.15 merupakan *sequence diagram* dari proses tambah pengguna. Proses ini diawali tampilan *form* tambah pengguna. Admin memasukkan data pengguna kemudian memanggil fungsi `tambahPengguna()` pada *Controller* Admin dan data yang dimasukkan dikirim pada fungsi `simpanPengguna()` pada *Model* `Model_Pengguna`. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola pengguna.

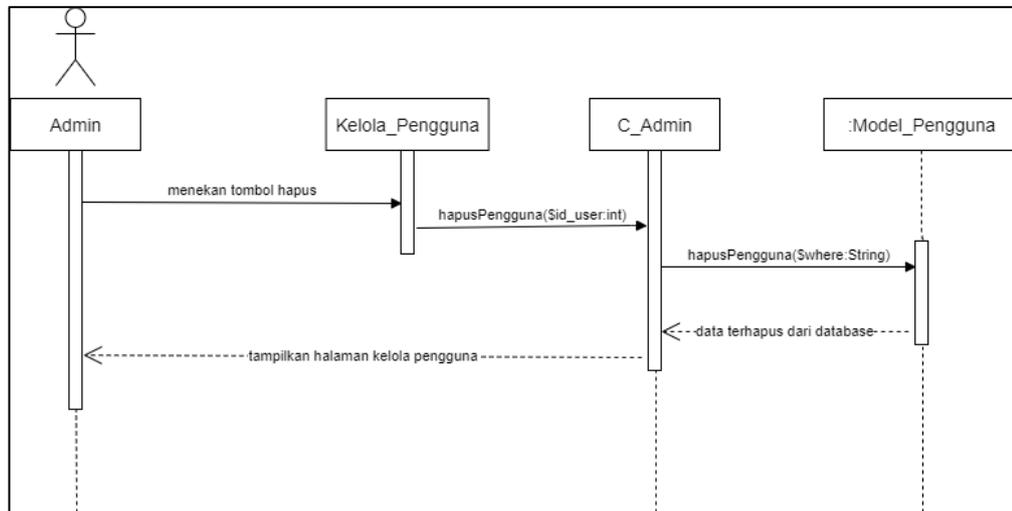
3. Mengubah data pengguna



Gambar 4.16 *Sequence diagram* ubah pengguna

Gambar 4.16 merupakan *sequence diagram* dari proses ubah pengguna. Proses ini diawali tampilan *form* Ubah_Pengguna. Admin memasukkan data pengguna yang baru kemudian memanggil fungsi `updatePengguna()` pada *Controller* Admin dan data yang dimasukkan dikirim pada fungsi `updatePengguna()` pada *Model* `Model_Pengguna`. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola pengguna.

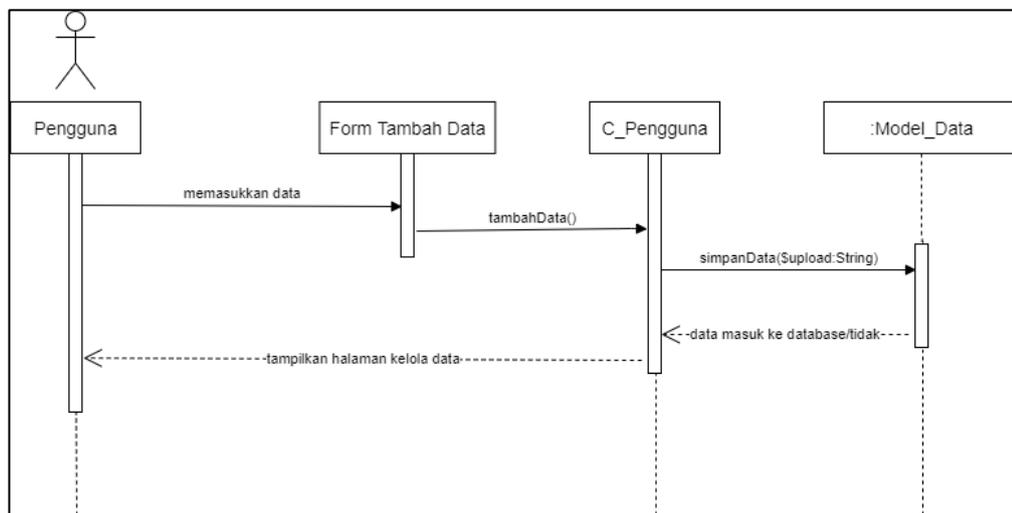
4. Menghapus pengguna



Gambar 4.17 Sequence diagram hapus pengguna

Gambar 4.17 merupakan *sequence diagram* menghapus pengguna. Berawal dari admin menekan tombol hapus kemudian memanggil fungsi `hapusPengguna()` pada *Controller* Admin. Data akan dihapus menggunakan fungsi `hapusPengguna()` pada *Model* `Model_Pengguna` dan data akan terhapus dari *database*.

5. Menambahkan data

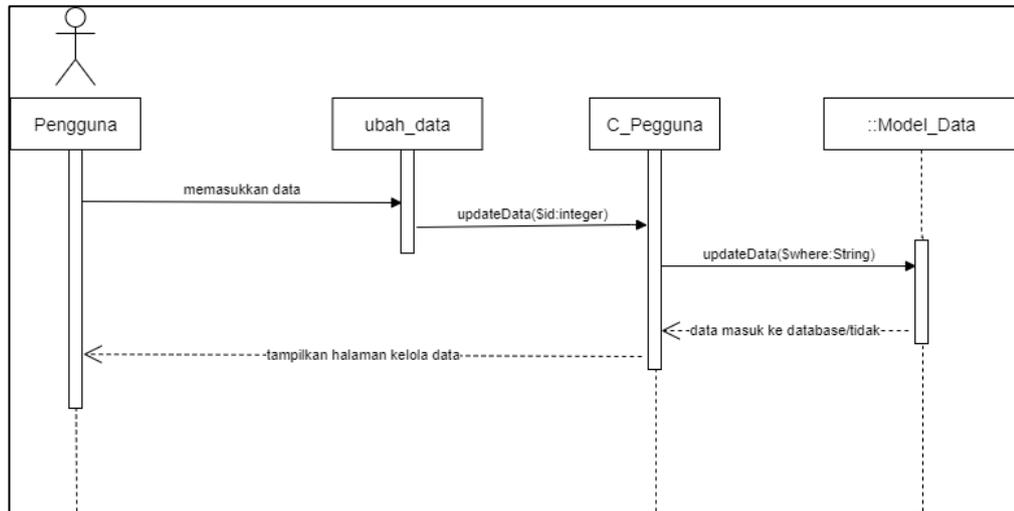


Gambar 4.18 Sequence diagram tambah data

Gambar 4.18 merupakan *sequence diagram* dari proses tambah data. Proses ini diawali tampilan *form* tambah data. Pengguna memasukkan data pemeliharaan kemudian memanggil fungsi `create()` pada *Controller*

Pengguna dan data yang dimasukkan dikirim pada fungsi `simpan()` dan akan tersimpan dalam *database*.

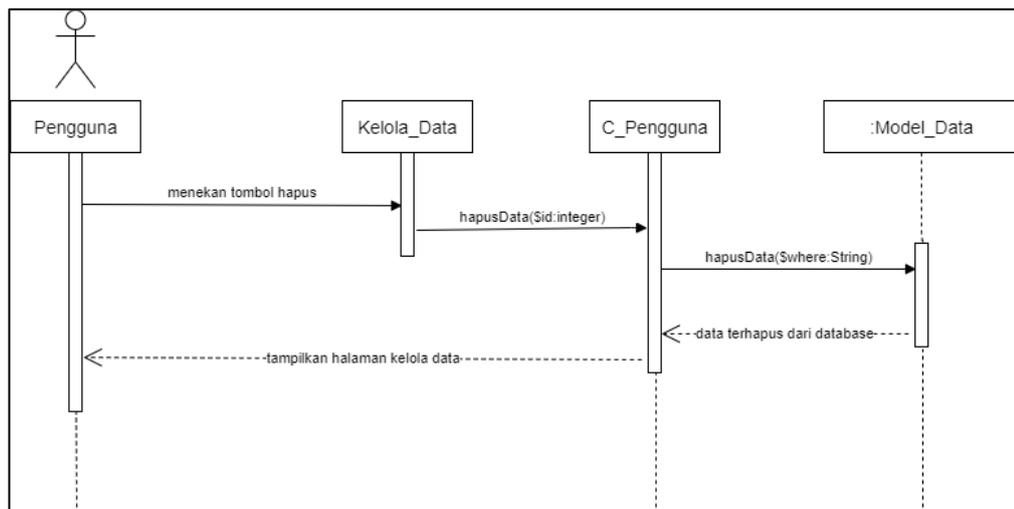
6. Mengubah data



Gambar 4.19 *Sequence diagram* ubah data

Gambar 4.19 merupakan *sequence diagram* dari proses ubah data. Proses ini diawali tampilan *form* `ubah_data`. Pengguna memasukkan data yang baru kemudian memanggil fungsi `update()` pada *Controller* Pengguna dan data yang dimasukkan dikirim pada fungsi `updateData()` pada *Model* `Model_Data`. Jika penyimpanan data berhasil, maka akan ditampilkan halaman kelola data.

7. Menghapus data

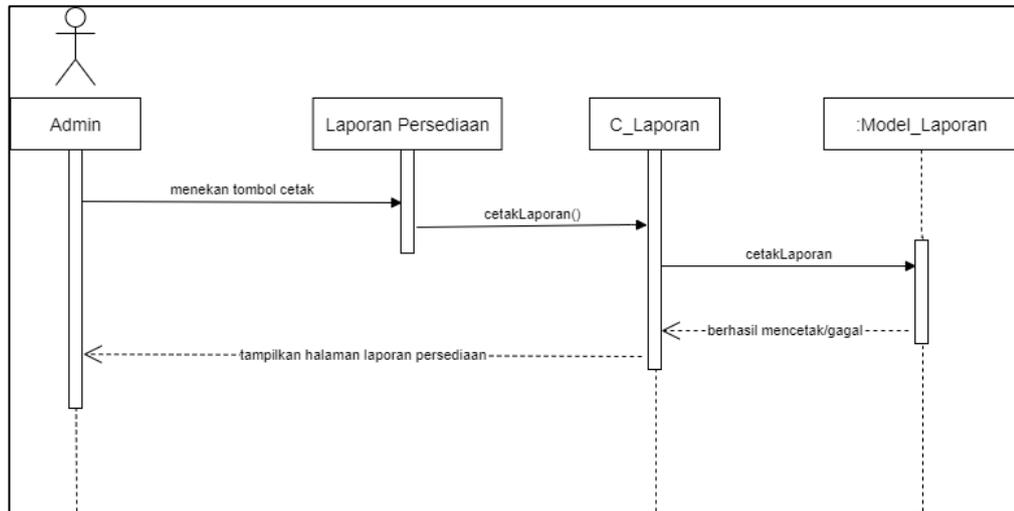


Gambar 4.20 *Sequence diagram* hapus data

Gambar 4.20 merupakan *sequence diagram* hapus data. Berawal dari pengguna menekan tombol hapus kemudian memanggil fungsi `hapus()` pada

Controller Pengguna. Data akan dihapus menggunakan fungsi `hapusData()` pada *Model Model_Data* dan data akan terhapus dari *database*.

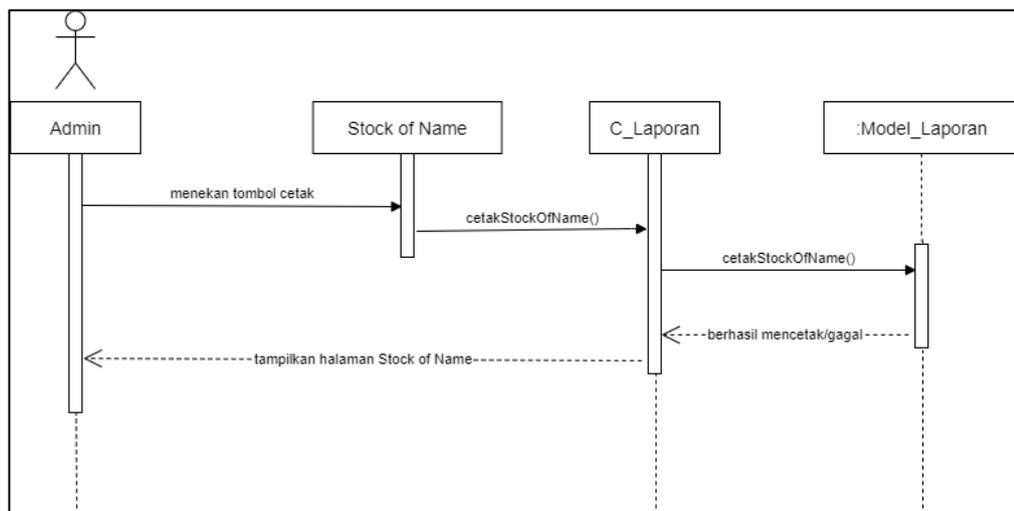
8. Mencetak laporan persediaan



Gambar 4.21 *Sequence diagram* cetak laporan persediaan

Gambar 4.21 merupakan *sequence diagram* proses mencetak laporan persediaan. Berawal dari Admin menekan tombol cetak pada halaman laporan persediaan kemudian memanggil fungsi `cetakLaporan()` pada *Controller* Laporan. Selanjutnya laporan persediaan akan tercetak dan sistem akan menampilkan kembali halaman laporan persediaan.

9. Mencetak *stock of name*

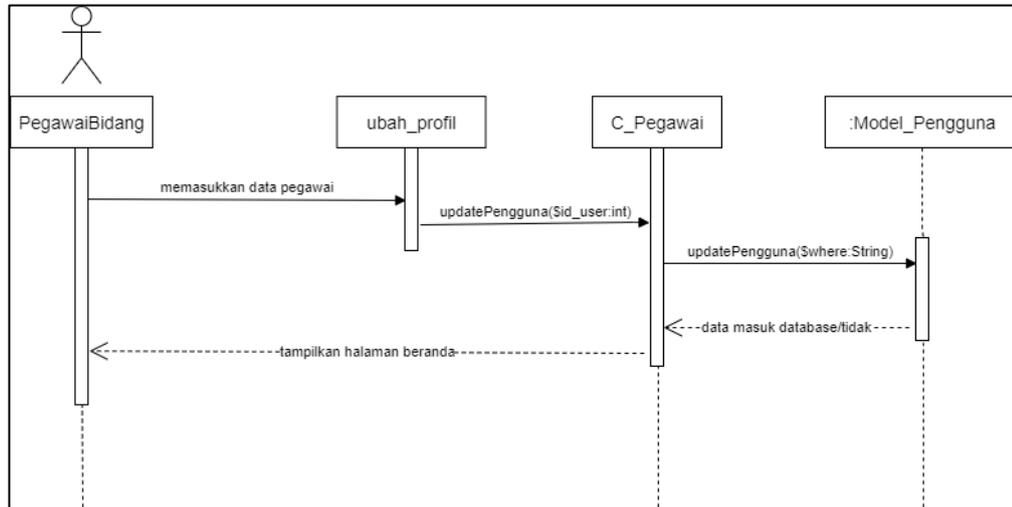


Gambar 4.22 *Sequence diagram* cetak *stock of name*

Gambar 4.22 merupakan *sequence diagram* proses mencetak *Stock of Name*. Berawal dari Admin menekan tombol cetak pada halaman *Stock of Name*

kemudian memanggil fungsi `cetakStockOfName()` pada *Controller* Laporan. Selanjutnya *Stock of Name* akan tercetak dan sistem akan menampilkan kembali halaman *Stock of Name*.

10. Mengubah profil

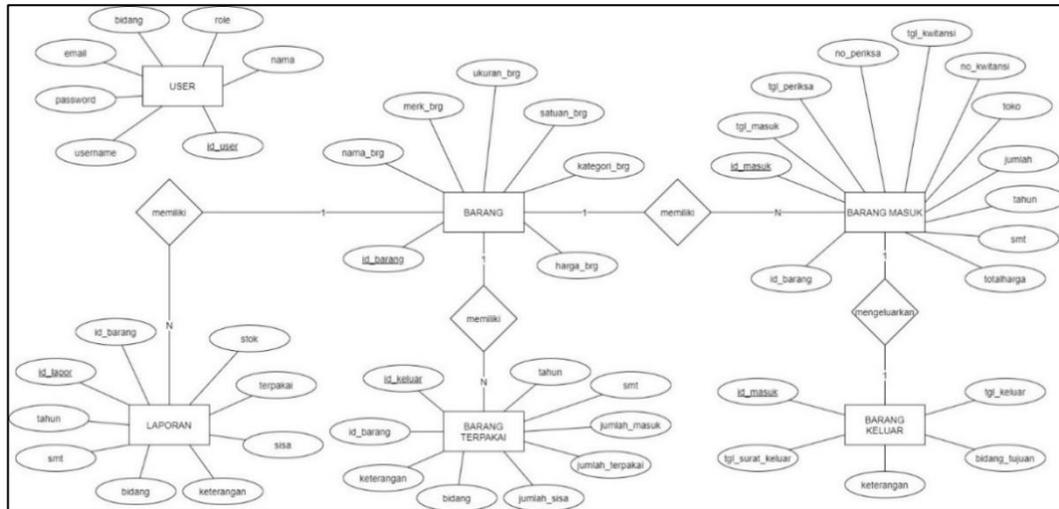


Gambar 4.23 *Sequence diagram* ubah profil

Gambar 4.23 merupakan *sequence diagram* proses ubah profil. Berawal dari Pegawai Bidang memasukkan data yang baru pada tampilan *form* ubah profil kemudian memanggil fungsi `updatePengguna()` pada *Controller* Pegawai dan data yang dimasukkan dikirim pada fungsi `updatePengguna()` pada *Model* *Model_Pengguna*. Jika penyimpanan data berhasil, maka akan ditampilkan halaman beranda pagi Pegawai Bidang.

4.3.5 ERD

Berikut merupakan *entity relationship diagram* dari Sistem Informasi Manajemen Aset BPKAD:



Gambar 4.24 ER Diagram

Gambar 4.24 merupakan *Entity Relationship Diagram* yang terdiri dari 6 (enam) entitas, yaitu *user*, *barang*, *barang masuk*, *barang keluar*, *barang pakai*, dan *laporan*. Berikut detail atribut dari entitas yang telah disebutkan:

Tabel 4.1 Entitas *User*

Field	Type	Null	Default
<u>id_user</u>	bigInt (20)	No	None
Username	Varchar(191)	No	None
Password	Varchar (191)	No	None
Nama	Varchar (191)	No	None
Email	Varchar (191)	No	None
Role	Varchar (191)	No	None
Bidang	Varchar (191)	No	None

Tabel 4.1 merupakan implementasi entitas *user* dalam bentuk tabel yang terdiri dari kolom nama atribut, tipe data, kondisi *Null*, dan *default*. Tabel ini nantinya akan diimplementasikan ke dalam *database*.

Tabel 4.2 Entitas Barang

Field	Type	Null	Default
<u>id_barang</u>	bigInt (20)	No	None
Nama_brg	Varchar(191)	No	None
Merk_brg	Varchar (191)	No	None
Ukuran_brg	Varchar (191)	No	None
Satuan_brg	Varchar (191)	No	None
Kategori_brg	Varchar (191)	No	None
Harga_brg	Varchar (191)	No	None

Tabel 4.2 merupakan implementasi entitas barang dalam bentuk tabel yang terdiri dari kolom nama atribut, tipe data, kondisi *Null*, dan *default*. Tabel ini nantinya akan diimplementasikan ke dalam *database*.

Tabel 4.3 Entitas Barang Masuk

Field	Type	Null	Default
<u>id_masuk</u>	bigInt (20)	No	None
Tgl_masuk	datetime	No	None
Tgl_periksa	Date	Yes	NULL
No_periksa	int (11)	Yes	NULL
Tgl_kwitansi	Date	Yes	NULL
No_kwitansi	int (11)	Yes	NULL
Toko	Varchar (191)	No	None
Jumlah	int (11)	No	None
Tahun	int (11)	No	None
Smt	int (11)	No	None
Totalharga	int (11)	Yes	NULL
Id_barang	bigInt (20)	No	None

Tabel 4.3 merupakan implementasi entitas barang masuk dalam bentuk tabel yang terdiri dari kolom nama atribut, tipe data, kondisi *Null*, dan *default*. Tabel ini nantinya akan diimplementasikan ke dalam *database*.

Tabel 4.4 Entitas Barang Keluar

Field	Type	Null	Default
Tgl_keluar	date	No	None
Tgl_surat_keluar	date	No	None
Bidang_tujuan	Varchar (191)	No	None
keterangan	text	Yes	NULL
Id_masuk	bigInt(20)	No	None

Tabel 4.4 merupakan implementasi entitas barang keluar dalam bentuk tabel yang terdiri dari kolom nama atribut, tipe data, kondisi *Null*, dan *default*. Tabel ini nantinya akan diimplementasikan ke dalam *database*.

Tabel 4.5 Entitas Barang Terpakai

Field	Type	Null	Default
<u>id_pakai</u>	bigInt (20)	No	None
Tahun	Int(11)	No	None
Smt	Int(11)	No	None
Jumlah_masuk	Int(11)	No	None
Jumlah_terpakai	Int(11)	No	None
Jumlah_sisa	Int(11)	Yes	NULL
Bidang	Varchar(191)	No	None
Keterangan	Text	Yes	NULL
Id_barang	bigInt(20)	No	None

Tabel 4.5 merupakan implementasi entitas barang terpakai dalam bentuk tabel yang terdiri dari kolom nama atribut, tipe data, kondisi *Null*, dan *default*. Tabel ini nantinya akan diimplementasikan ke dalam *database*.

Tabel 4.6 Entitas Laporan

Field	Type	Null	Default
<u>id_lapor</u>	bigInt (20)	No	None
Tahun	Int(11)	No	None
Smt	Int(11)	No	None

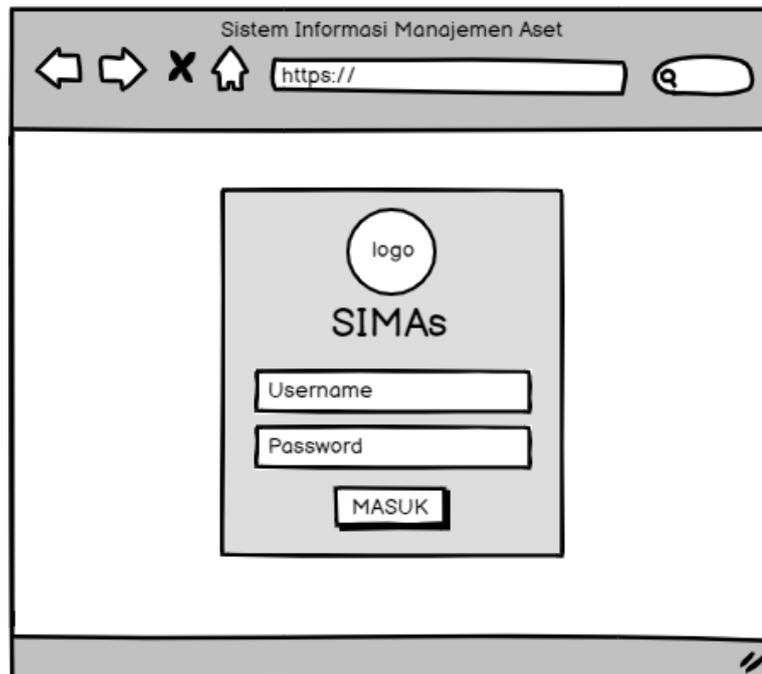
Stok	Int(11)	No	None
Terpakai	Int(11)	Yes	NULL
Sisa	Int(11)	Yes	NULL
Bidang	Varchar(191)	Yes	NULL
Keterangan	Text	Yes	NULL
Id_barang	bigInt(20)	No	None

Tabel 4.6 merupakan implementasi entitas laporan dalam bentuk tabel yang terdiri dari kolom nama atribut, tipe data, kondisi *Null*, dan *default*. Tabel ini nantinya akan diimplementasikan ke dalam *database*.

4.3.6 Desain Tampilan Sistem

Berikut merupakan desain tampilan dari Sistem Informasi Manajemen Aset BPKAD :

1. Halaman *Login*

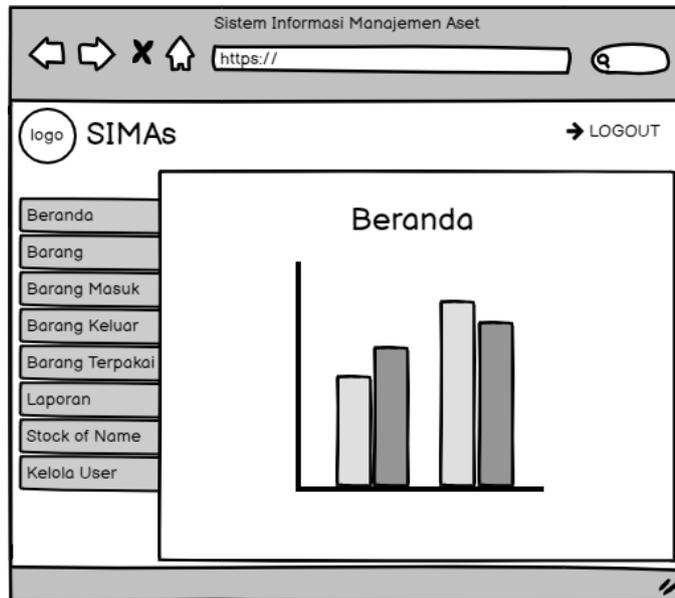


Gambar 4.25 Desain tampilan halaman *login*

Pada Gambar 4.25 merupakan implementasi dari halaman *login* yang digunakan *user* untuk masuk ke dalam sistem.

2. Desain Tampilan Sistem untuk Admin

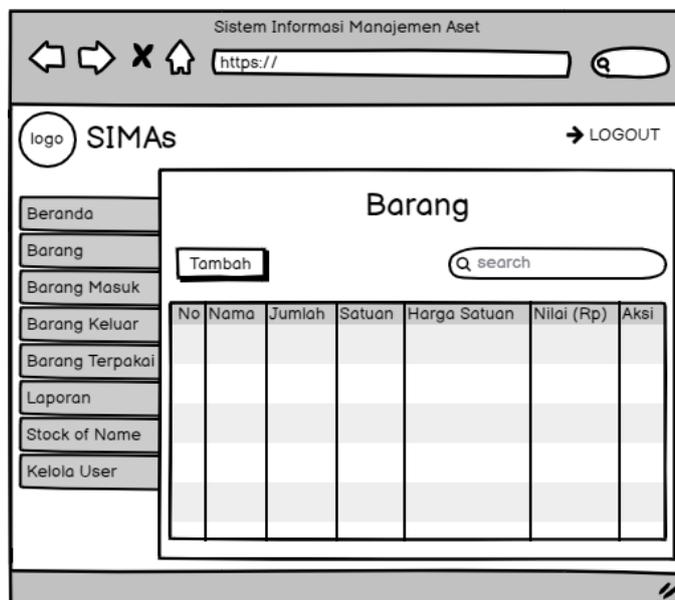
a. Halaman Beranda



Gambar 4.26 Desain tampilan halaman beranda admin

Pada Gambar 4.26 merupakan implementasi dari halaman beranda yang akan dijumpai admin ketika pertama kali masuk ke dalam sistem.

b. Halaman Daftar Barang

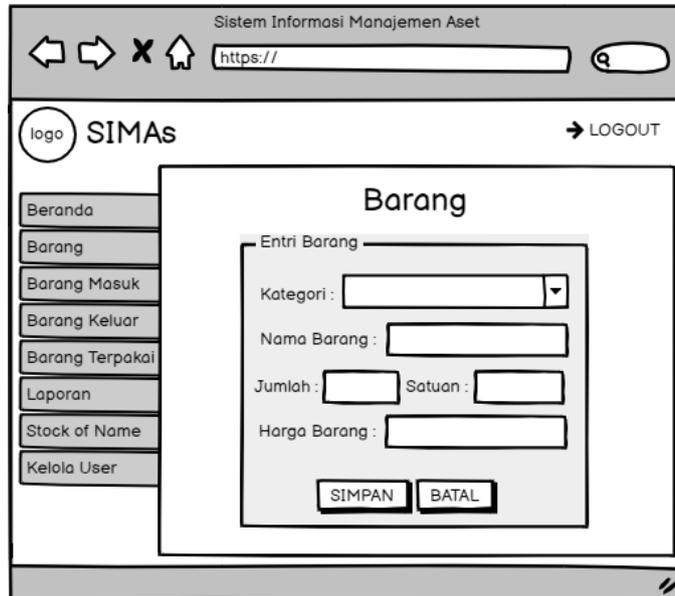


Gambar 4.27 Desain tampilan halaman daftar barang admin

Pada Gambar 4.27 merupakan implementasi dari halaman daftar barang yang berisi informasi tentang barang dan tombol cari untuk melakukan

pencarian barang. Pada halaman daftar barang juga dilengkapi dengan tombol tambah, hapus, dan edit data barang.

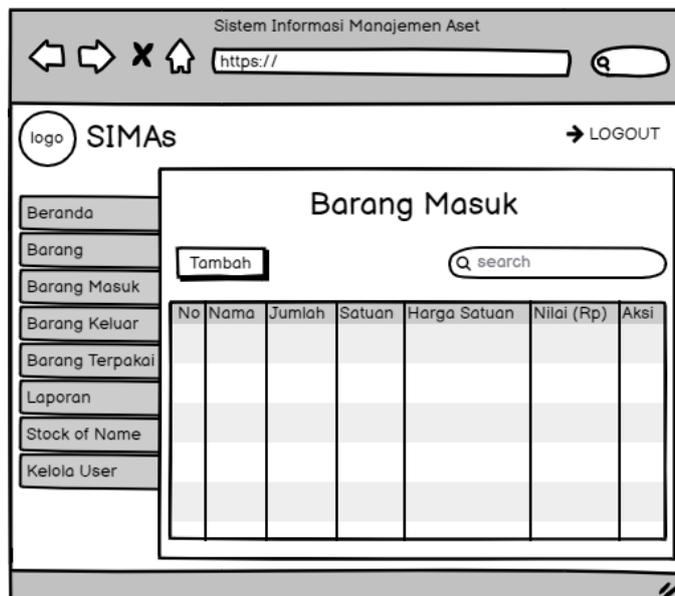
Tampilan jika pengguna ingin mengisi data pada tabel barang :



Gambar 4.28 Desain tampilan halaman isi data daftar barang admin

Pada Gambar 4.28 merupakan desain tampilan dari halaman isi data daftar barang bagi admin.

c. Halaman Daftar Barang Masuk

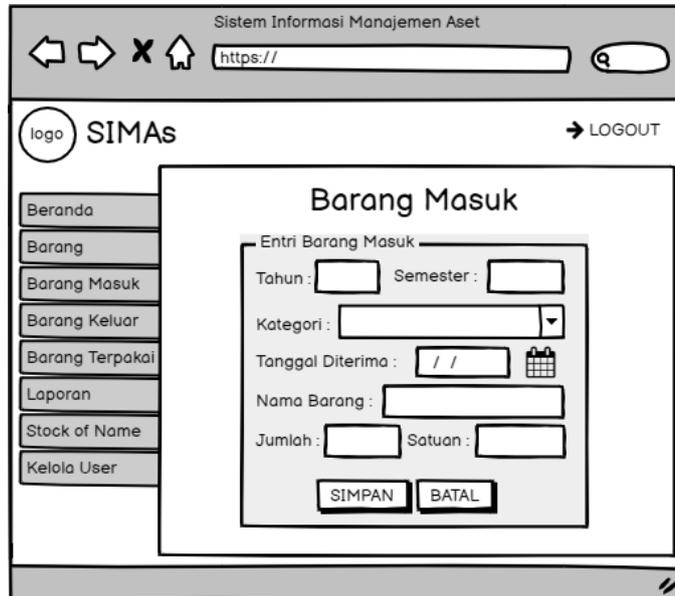


Gambar 4.29 Desain tampilan halaman daftar barang masuk admin

Pada Gambar 4.29 merupakan implementasi dari halaman daftar barang masuk yang berisi informasi tentang barang masuk dan tombol cari untuk

melakukan pencarian barang masuk. Pada halaman daftar barang masuk juga dilengkapi dengan tombol tambah, hapus, dan edit data barang masuk.

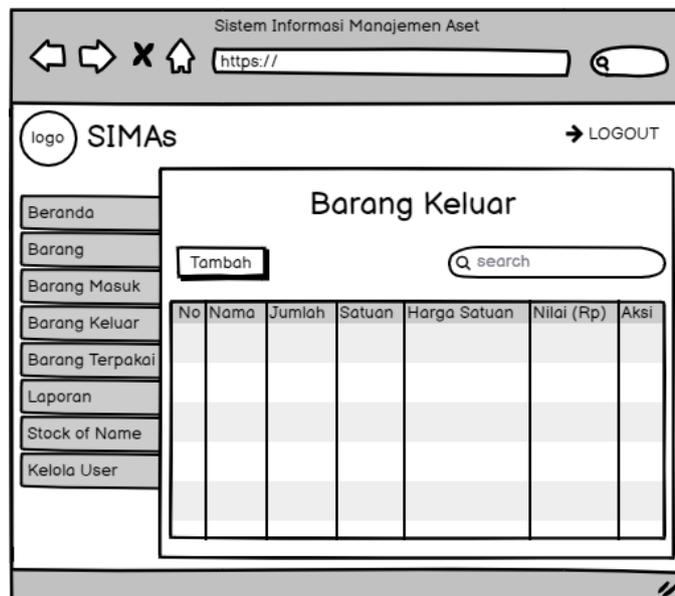
Tampilan jika pengguna ingin mengisi data pada tabel barang masuk :



Gambar 4.30 Desain tampilan halaman isi data barang masuk admin

Pada Gambar 4.30 merupakan desain tampilan dari halaman isi data daftar barang masuk bagi admin.

d. Halaman Daftar Barang Keluar



Gambar 4.31 Desain tampilan halaman daftar barang keluar admin

Pada Gambar 4.31 merupakan implementasi dari halaman daftar barang keluar yang berisi informasi tentang barang keluar dan tombol cari untuk

melakukan pencarian barang keluar. Pada halaman daftar barang keluar juga dilengkapi dengan tombol tambah, hapus, dan edit data barang.

Tampilan jika pengguna ingin mengisi data pada tabel barang keluar :

Gambar 4.32 Desain tampilan halaman isi data barang keluar admin

Pada Gambar 4.28 merupakan desain tampilan dari halaman isi data daftar barang keluar bagi admin.

e. Halaman Daftar Barang Terpakai

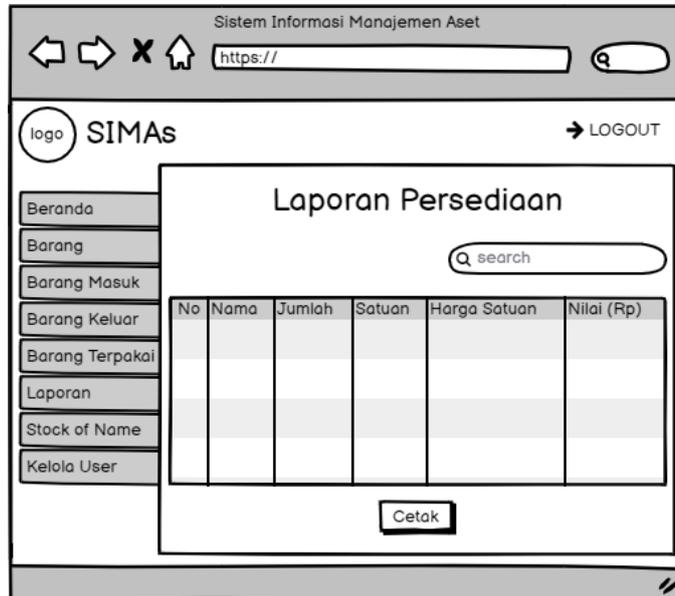
No	Nama	Jumlah	Satuan	Harga Satuan	Nilai (Rp)	Aksi

Gambar 4.33 Desain tampilan halaman daftar barang terpakai admin

Pada Gambar 4.33 merupakan implementasi dari halaman daftar barang terpakai yang berisi informasi tentang barang terpakai dan tombol cari untuk

melakukan pencarian barang terpakai. Pada halaman daftar barang terpakai bagi admin juga dilengkapi dengan tombol hapus, dan edit data barang.

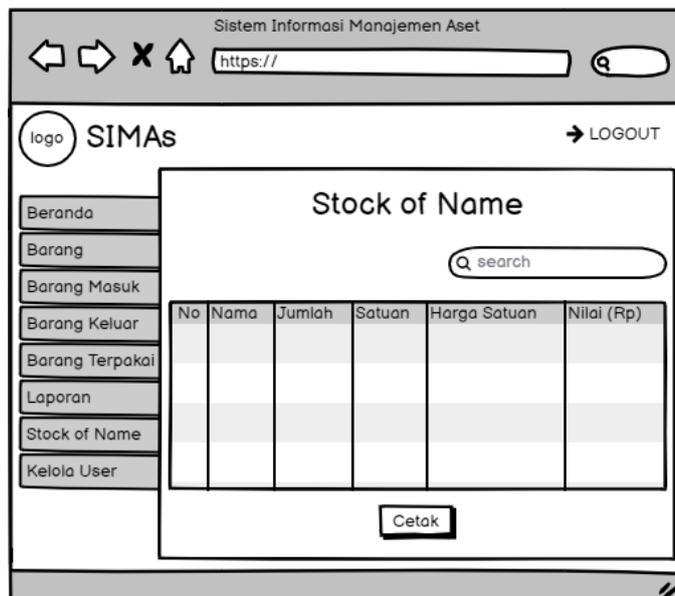
f. Halaman Laporan Persediaan



Gambar 4.34 Desain tampilan halaman laporan persediaan admin

Pada Gambar 4.34 merupakan implementasi dari halaman laporan persediaan dan tombol cari untuk melakukan pencarian.

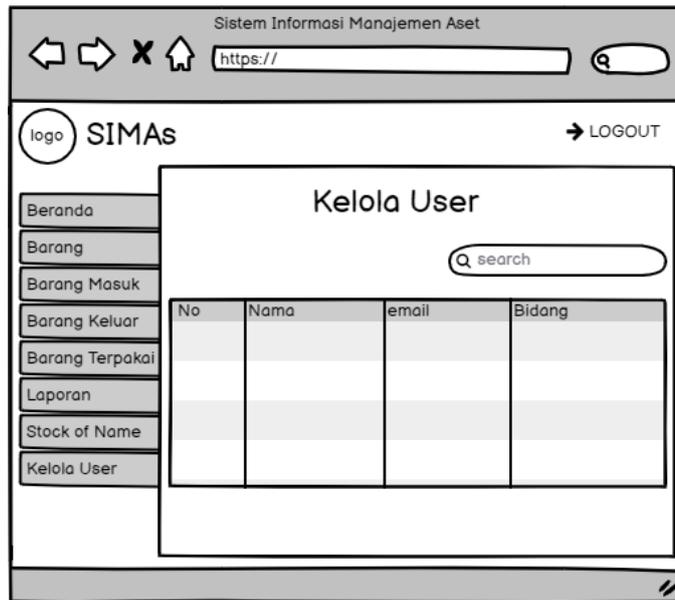
g. Halaman *Stock of Name*



Gambar 4.35 Desain tampilan halaman *stock of name* admin

Pada Gambar 4.35 merupakan implementasi dari halaman laporan *stock of name* dan tombol cari untuk melakukan pencarian.

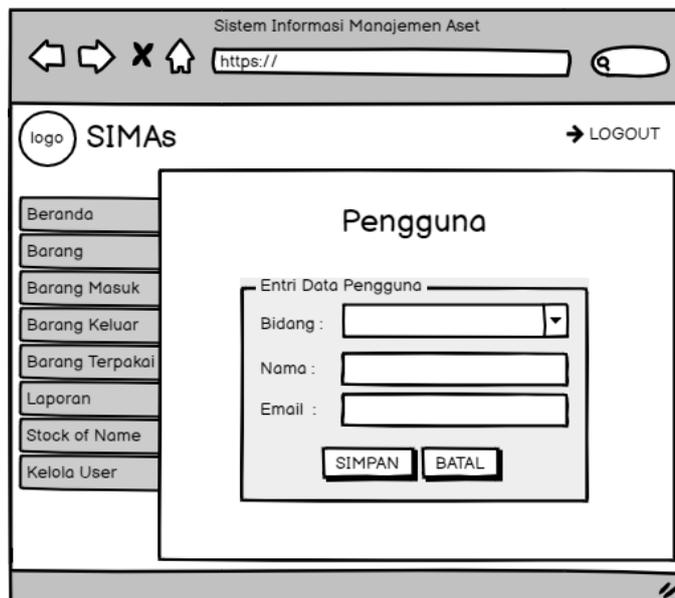
h. Halaman Kelola Pengguna



Gambar 4.36 Desain tampilan halaman kelola pengguna admin

Pada Gambar 4.36 merupakan implementasi dari halaman kelola pengguna yang berisi informasi tentang pengguna dan tombol cari untuk melakukan pencarian. Pada halaman kelola pengguna bagi admin juga dilengkapi dengan tombol hapus, dan edit data pengguna.

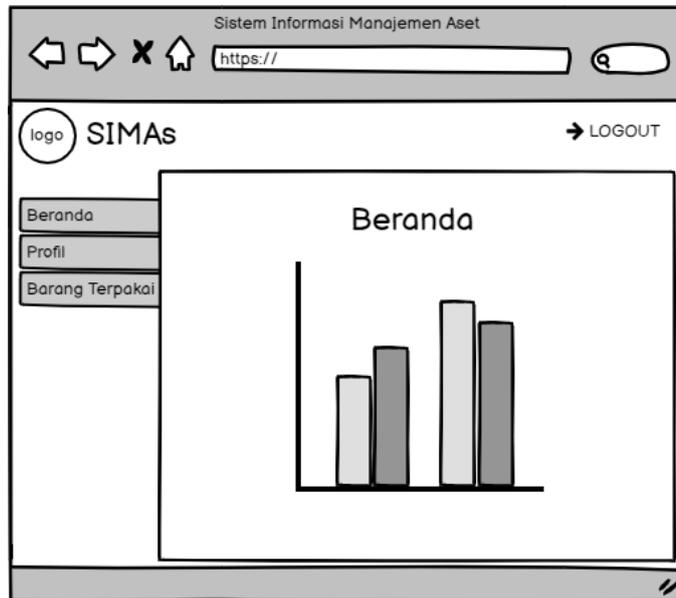
Tampilan jika pengguna ingin mengisi data pada tabel pengguna :



Gambar 4.37 Desain tampilan halaman isi data pengguna admin

Pada Gambar 4.37 merupakan desain tampilan dari halaman isi data pengguna bagi admin.

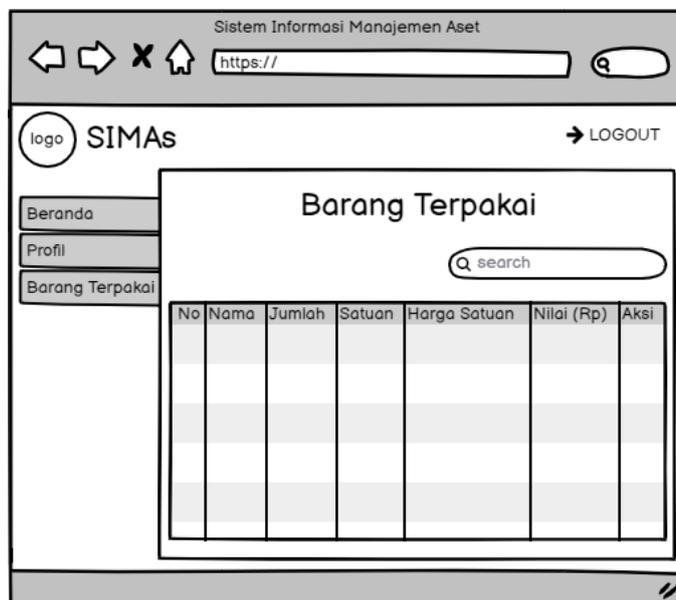
3. Desain Tampilan Sistem untuk Pegawai Bidang
 - a. Halaman Beranda



Gambar 4.38 Desain tampilan halaman beranda pegawai bidang

Pada Gambar 4.38 merupakan implementasi dari halaman beranda yang akan dijumpai pegawai bidang ketika pertama kali masuk ke dalam sistem.

- b. Halaman Daftar Barang Terpakai



Gambar 4.39 Desain tampilan halaman daftar barang terpakai pegawai bidang

Pada Gambar 4.39 merupakan implementasi dari halaman daftar barang terpakai keluar yang berisi informasi tentang barang keluar dan tombol cari untuk melakukan pencarian barang keluar. Pada halaman daftar barang bagi

pegawai bidang juga dilengkapi dengan tombol tambah, hapus, dan edit data barang terpakai.

Tampilan jika pengguna ingin mengisi data pada tabel barang terpakai :

Sistem Informasi Manajemen Aset

https://

logo SIMAs → LOGOUT

Beranda
Profil
Barang Terpakai

Barang Terpakai

Entri Barang Terpakai

Tahun : Semester :

Kategori :

Nama Barang :

Jumlah : Satuan :

SIMPAN BATAL

Gambar 4.40 Desain tampilan halaman isi data barang terpakai pegawai bidang

Pada Gambar 4.40 merupakan desain tampilan dari halaman isi data barang terpakai bagi pegawai bidang.

c. Halaman Ubah Profil

Sistem Informasi Manajemen Aset

https://

logo SIMAs → LOGOUT

Beranda
Profil
Barang Terpakai

Pengguna

Entri Data Pengguna

Bidang :

Nama :

Email :

SIMPAN BATAL

Gambar 4.41 Desain tampilan halaman ubah profil pegawai bidang

Pada Gambar 4.41 merupakan implementasi dari ubah profil bagi pegawai bidang untuk mengubah data profilnya.

4.4 Implementasi Sistem

4.3.1 Implementasi *Database* Sistem

Berikut merupakan implementasi *database* dari Sistem Informasi Manajemen Aset BPKAD :

1. Tabel *User*

Tabel *user* merupakan *database* Manajemen_Aset yang digunakan untuk menyimpan data dari pengguna. Adapun atribunya adalah id, name, email, username, password, role, dan bidang. Implementasi table *user* dapat dilihat pada Gambar 4.42.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id 🔑		bigint(20)	UNSIGNED	No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	name		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	3	email		varchar(191)	utf8mb4_unicode_ci	Yes	NULL	
<input type="checkbox"/>	4	password		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	5	username		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	6	role		varchar(191)	utf8mb4_unicode_ci	Yes	NULL	
<input type="checkbox"/>	7	bidang		varchar(191)	utf8mb4_unicode_ci	Yes	NULL	

Gambar 4.42 Implementasi tabel *user*

2. Tabel Barang

Tabel barang merupakan *database* Manajemen_Aset yang digunakan untuk menyimpan data dari Barang. Adapun atribunya adalah id, nama_brg, merk_brg, ukuran_brg, satuan_brg, kategori_brg, dan harga_brg. Implementasi tabel barang dapat dilihat pada Gambar 4.43.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id 🔑		bigint(20)	UNSIGNED	No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	nama_brg		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	3	merk_brg		varchar(191)	utf8mb4_unicode_ci	Yes	NULL	
<input type="checkbox"/>	4	ukuran_brg		varchar(191)	utf8mb4_unicode_ci	Yes	NULL	
<input type="checkbox"/>	5	satuan_brg		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	6	kategori_brg		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	7	harga_brg		int(11)		No	None	

Gambar 4.43 Implementasi tabel barang

3. Tabel Barang Masuk

Tabel barang masuk merupakan *database* Manajemen_Aset yang digunakan untuk menyimpan data dari barang masuk. Adapun atribunya adalah id, tgl_masuk, tgl_periksa, no_periksa, tgl_kwitansi, no_kwitansi, jumlah, toko,

tahun, smt, totalharga, id_barang. Implementasi tabel barang masuk dapat dilihat pada Gambar 4.44.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id		bigint(20)	UNSIGNED	No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	created_at		timestamp		Yes	NULL	
<input type="checkbox"/>	3	updated_at		timestamp		Yes	NULL	
<input type="checkbox"/>	4	tgl_masuk		datetime		No	None	
<input type="checkbox"/>	5	tgl_periksa		date		Yes	NULL	
<input type="checkbox"/>	6	no_periksa		int(11)		Yes	NULL	
<input type="checkbox"/>	7	jumlah		int(11)		No	None	
<input type="checkbox"/>	8	tgl_kwitansi		date		Yes	NULL	
<input type="checkbox"/>	9	no_kwitansi		int(11)		Yes	NULL	
<input type="checkbox"/>	10	toko		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	11	id_barang		bigint(20)	UNSIGNED	No	None	
<input type="checkbox"/>	12	tahun		int(11)		No	None	
<input type="checkbox"/>	13	smt		int(11)		No	None	
<input type="checkbox"/>	14	totalharga		int(11)		Yes	NULL	

Gambar 4.44 Implementasi tabel barang masuk

4. Tabel Barang Keluar

Tabel barang keluar merupakan *database* Manajemen_Aset yang digunakan untuk menyimpan data dari barang keluar. Adapun atribunya adalah tgl_keluar, tgl_surat_keluar, bidang_tujuan, keterangan, dan id_masuk. Implementasi tabel barang keluar dapat dilihat pada Gambar 4.45.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	created_at		timestamp		Yes	NULL	
<input type="checkbox"/>	2	updated_at		timestamp		Yes	NULL	
<input type="checkbox"/>	3	tgl_keluar		date		No	None	
<input type="checkbox"/>	4	tgl_surat_keluar		date		No	None	
<input type="checkbox"/>	5	bidang_tujuan		varchar(191)	utf8mb4_unicode_ci	No	None	
<input type="checkbox"/>	6	keterangan		text	utf8mb4_unicode_ci	Yes	NULL	
<input type="checkbox"/>	7	id_masuk		bigint(20)	UNSIGNED	No	None	

Gambar 4.45 Implementasi tabel barang keluar

5. Tabel Barang Terpakai

Tabel barang terpakai merupakan *database* Manajemen_Aset yang digunakan untuk menyimpan data dari barang terpakai. Adapun atribunya adalah id, tahun, smt, bidang, jumlah_masuk, jumlah_terpakai, jumlah_sisa, keterangan, dan id_barang. Implementasi tabel barang terpakai dapat dilihat pada Gambar 4.46.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	3 updated_at	timestamp			Yes	NULL		
<input type="checkbox"/>	4 tahun	int(11)			No	None		
<input type="checkbox"/>	5 smt	int(11)			No	None		
<input type="checkbox"/>	6 bidang	varchar(191)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	7 jumlah_masuk	int(11)			No	None		
<input type="checkbox"/>	8 jumlah_terpakai	int(11)			No	None		
<input type="checkbox"/>	9 jumlah_sisa	int(11)			Yes	NULL		
<input type="checkbox"/>	10 id_barang 🔑	bigint(20)		UNSIGNED	No	None		
<input type="checkbox"/>	11 keterangan	text	utf8mb4_unicode_ci		Yes	NULL		

Gambar 4.46 Implementasi tabel barang terpakai

6. Tabel Laporan

Tabel laporan merupakan *database* Manajemen_Aset yang digunakan untuk menyimpan data dari laporan. Adapun atribunya adalah id, tahun, smt, bidang, stok, terpakai, sisa, keterangan, dan id_barang. Implementasi tabel laporan dapat dilihat pada Gambar 4.47.

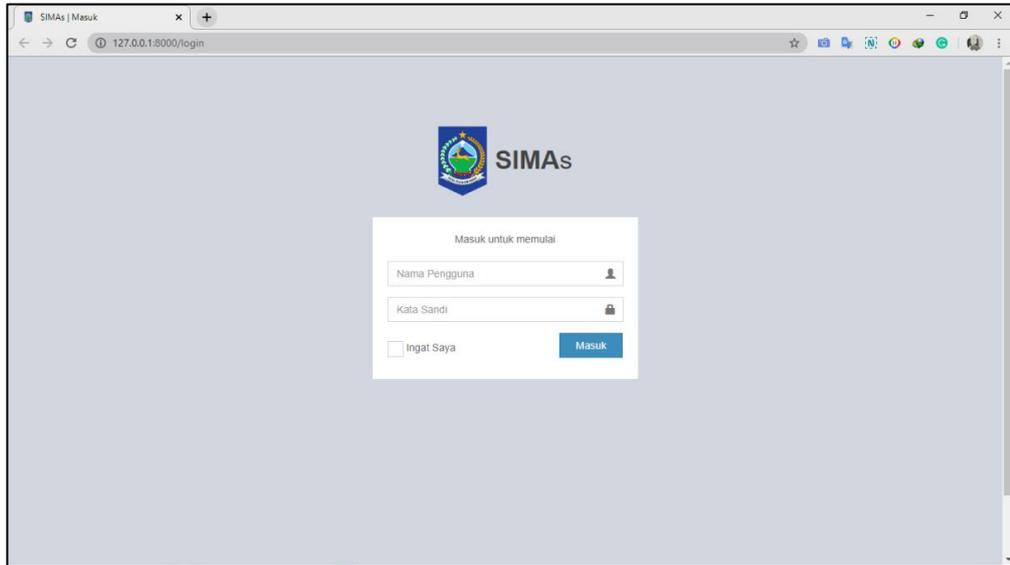
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 tahun	int(11)			No	None		
<input type="checkbox"/>	3 smt	int(11)			No	None		
<input type="checkbox"/>	4 bidang	varchar(191)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	5 stok	int(11)			No	None		
<input type="checkbox"/>	6 terpakai	int(11)			Yes	NULL		
<input type="checkbox"/>	7 sisa	int(11)			Yes	NULL		
<input type="checkbox"/>	8 id_barang 🔑	bigint(20)		UNSIGNED	No	None		
<input type="checkbox"/>	9 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	10 updated_at	timestamp			Yes	NULL		
<input type="checkbox"/>	11 keterangan	text	utf8mb4_unicode_ci		Yes	NULL		

Gambar 4.47 Implementasi tabel laporan

4.3.2 Implementasi Tampilan Sistem

Berikut merupakan implementasi tampilan dari Sistem Informasi Manajemen Aset BPKAD :

1. Halaman *Login*

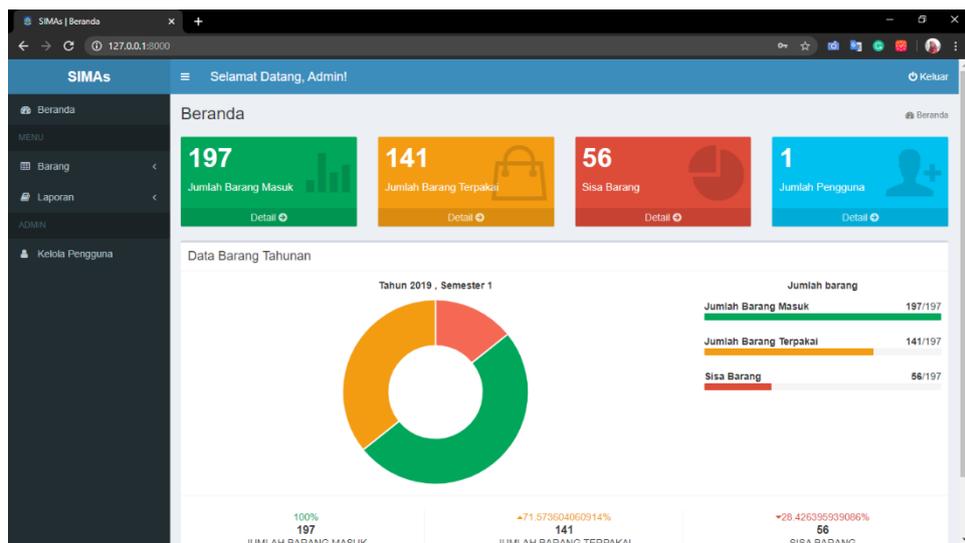


Gambar 4.48 Implementasi tampilan halaman *login*

Gambar 4.48 merupakan implementasi dari halaman *login* yang digunakan pengguna untuk masuk ke dalam sistem.

2. Desain Tampilan Sistem untuk Admin

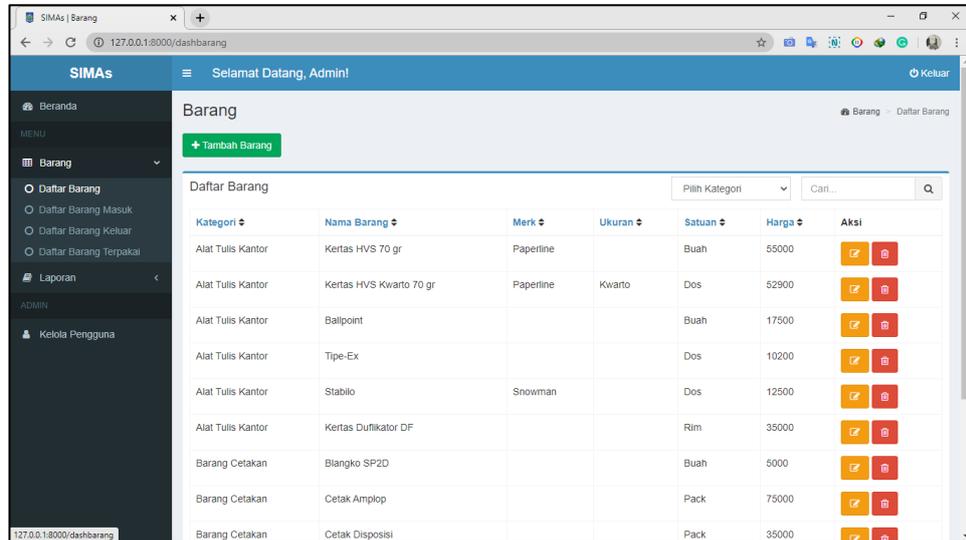
a. Halaman Beranda



Gambar 4.49 Implementasi tampilan halaman beranda admin

Pada Gambar 4.49 merupakan implementasi dari halaman beranda yang akan dijumpai admin ketika pertama kali masuk ke dalam sistem.

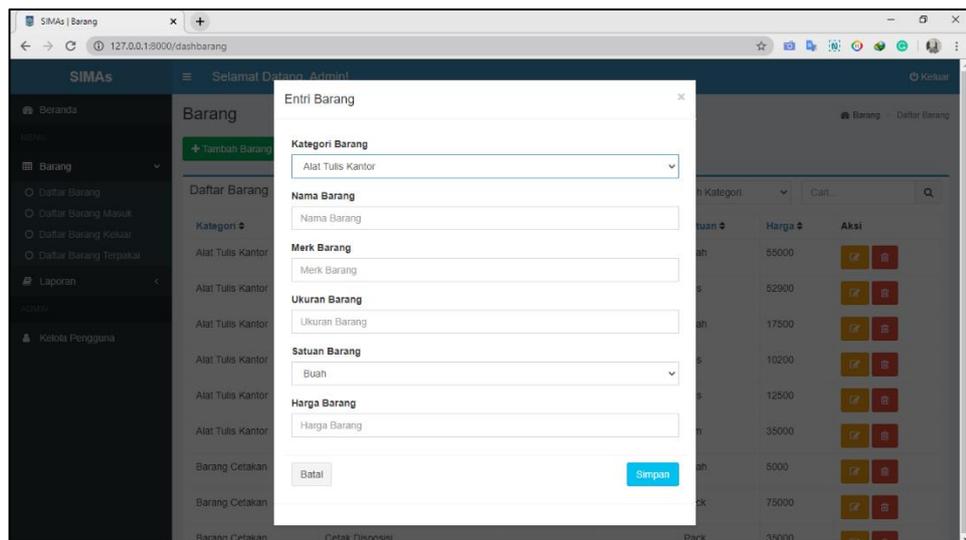
b. Halaman Daftar Barang



Gambar 4.50 Implementasi tampilan halaman daftar barang admin

Pada Gambar 4.50 merupakan implementasi dari halaman daftar barang yang berisi informasi tentang barang dan tombol cari untuk melakukan pencarian barang. Pada halaman daftar barang juga dilengkapi dengan tombol tambah, hapus, dan edit data barang.

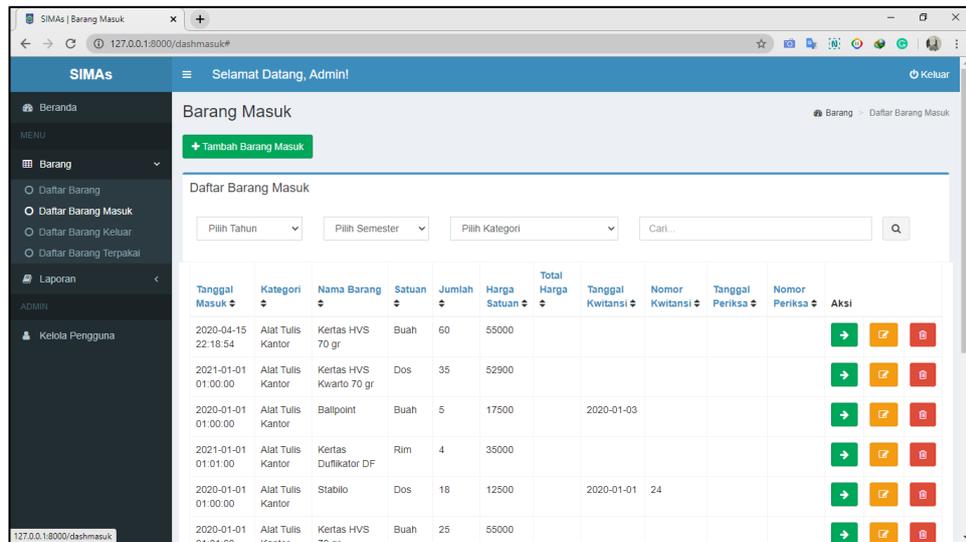
Tampilan jika pengguna ingin mengisi data pada tabel barang :



Gambar 4.51 Implementasi tampilan halaman isi data barang admin

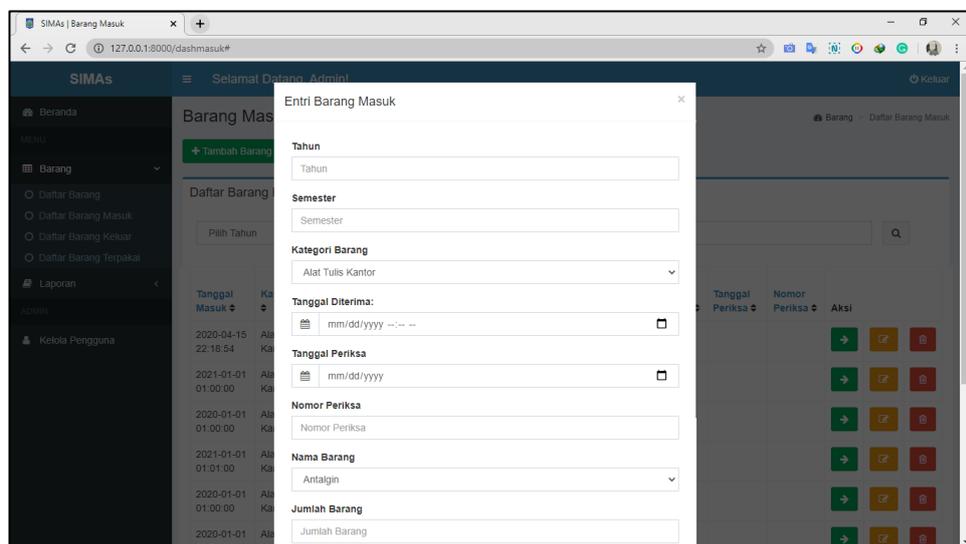
Pada Gambar 4.51 merupakan implementasi tampilan dari halaman isi data barang bagi admin.

c. Halaman Daftar Barang Masuk



Gambar 4.52 Implementasi tampilan halaman daftar barang masuk admin

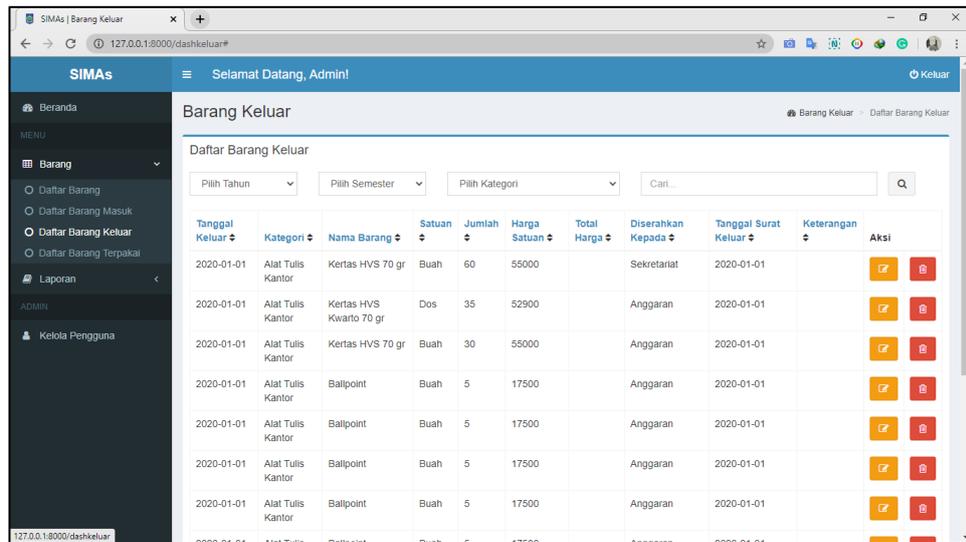
Pada Gambar 4.52 merupakan implementasi dari halaman daftar barang masuk yang berisi informasi tentang barang masuk dan tombol cari untuk melakukan pencarian barang masuk. Pada halaman daftar barang masuk juga dilengkapi dengan tombol tambah, hapus, dan edit data barang masuk. Tampilan jika pengguna ingin mengisi data pada tabel barang masuk :



Gambar 4.53 Implementasi tampilan halaman isi data barang masuk admin

Pada Gambar 4.53 merupakan implementasi dari halaman isi data barang masuk bagi admin.

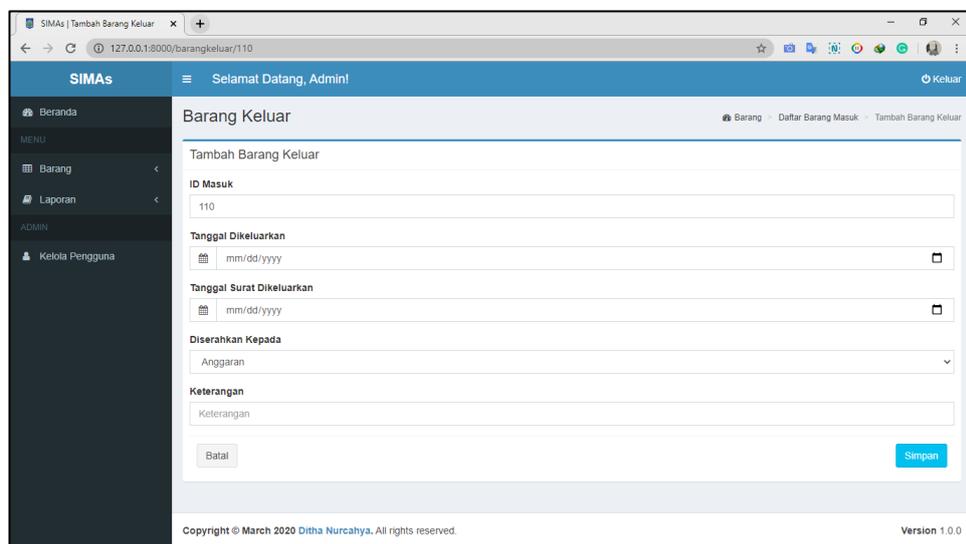
d. Halaman Daftar Barang Keluar



Gambar 4.54 Implementasi tampilan halaman daftar barang keluar admin

Pada Gambar 4.54 merupakan implementasi dari halaman daftar barang keluar yang berisi informasi tentang barang keluar dan tombol cari untuk melakukan pencarian barang keluar. Pada halaman daftar barang keluar juga dilengkapi dengan tombol tambah, hapus, dan edit data barang.

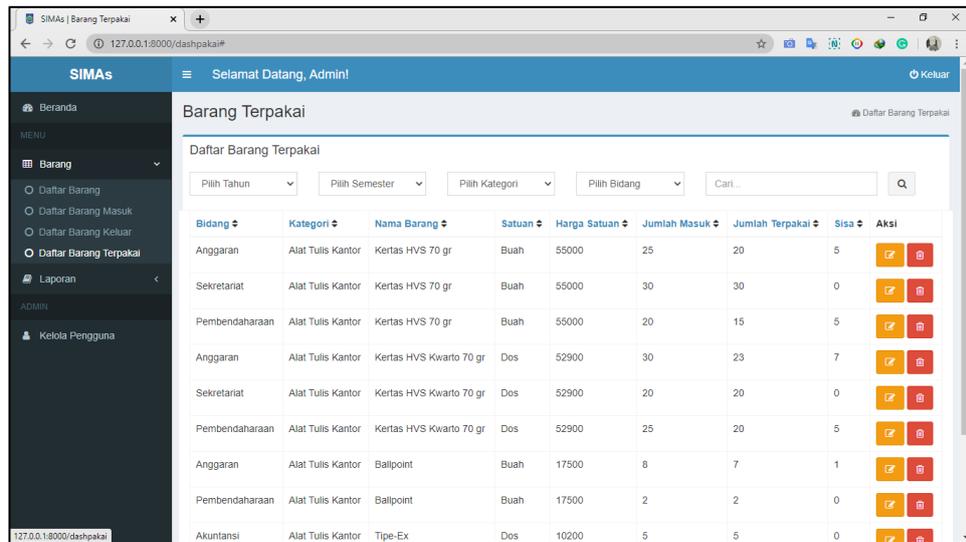
Tampilan jika pengguna ingin mengisi data pada tabel barang keluar :



Gambar 4.55 Implementasi tampilan halaman isi data barang keluar admin

Pada Gambar 4.55 merupakan implementasi dari halaman isi data daftar barang keluar bagi admin.

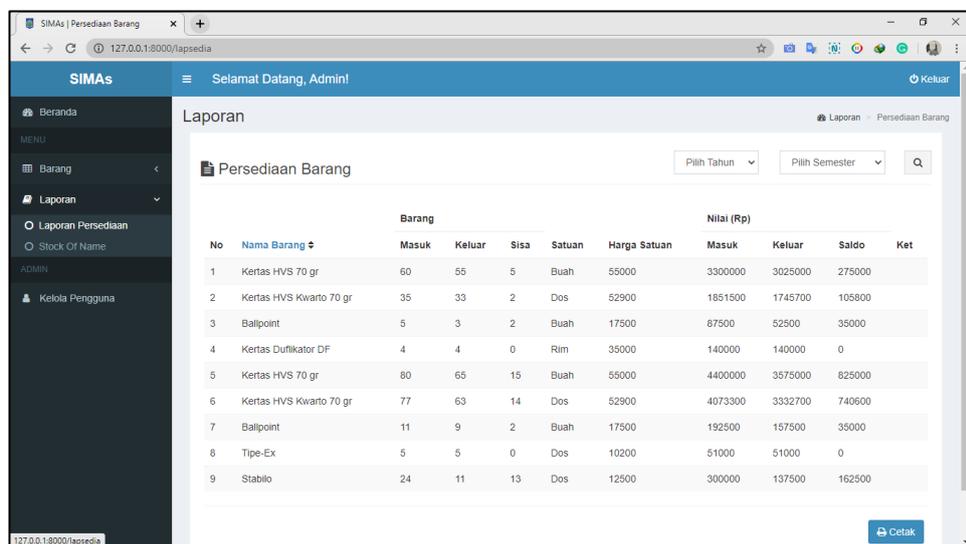
e. Halaman Daftar Barang Terpakai



Gambar 4.56 Implementasi tampilan halaman daftar barang terpakai admin

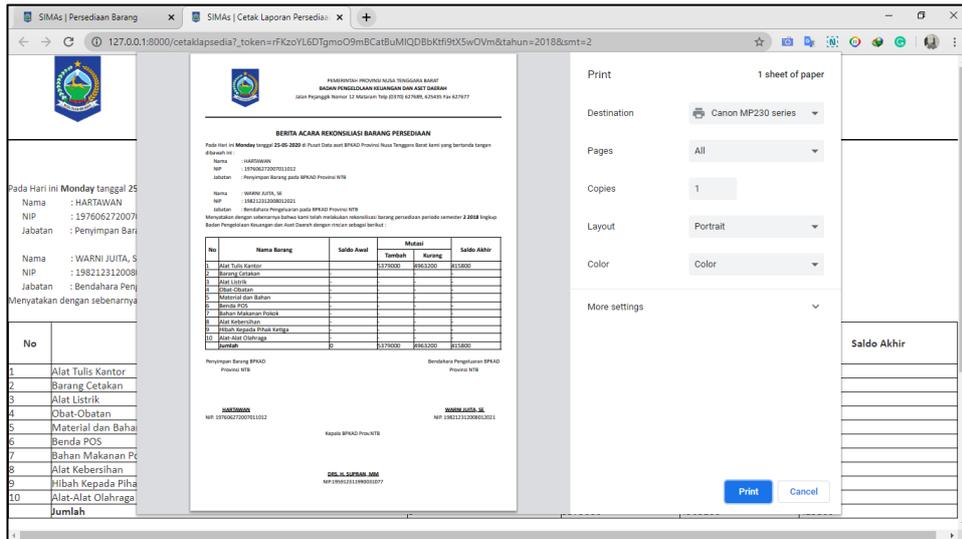
Pada Gambar 4.56 merupakan implementasi dari halaman daftar barang terpakai yang berisi informasi tentang barang terpakai dan tombol cari untuk melakukan pencarian barang terpakai. Pada halaman daftar barang terpakai bagi admin juga dilengkapi dengan tombol hapus, dan edit data barang.

f. Halaman Laporan Persediaan



Gambar 4.57 Implementasi tampilan halaman laporan persediaan admin

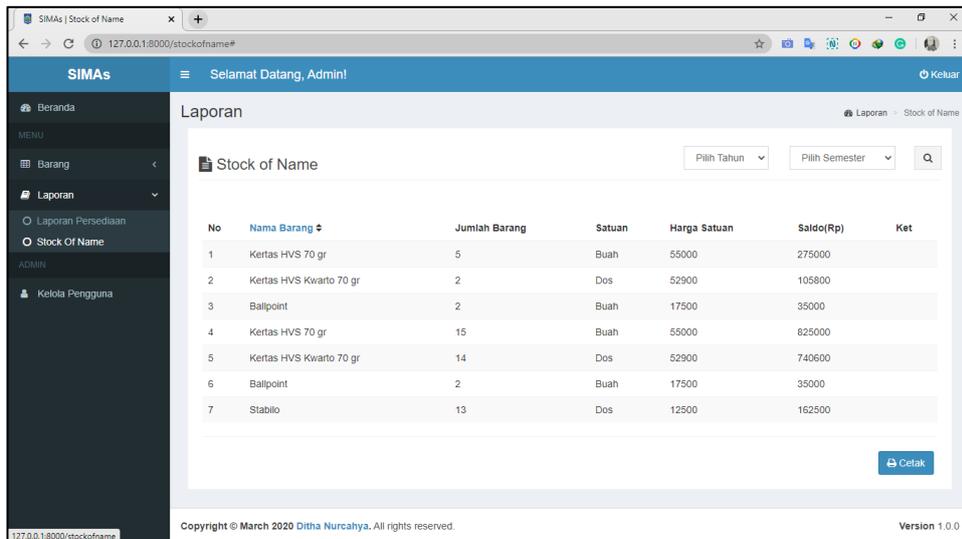
Pada Gambar 4.57 merupakan implementasi dari halaman laporan persediaan dan tombol cari untuk melakukan pencarian.



Gambar 4.58 Implementasi cetak laporan persediaan

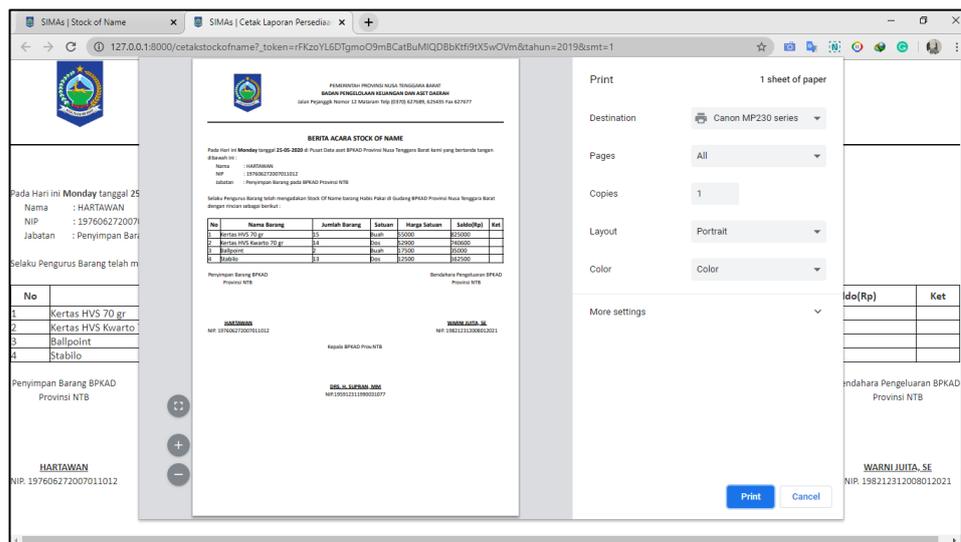
Pada Gambar 4.58 merupakan implementasi dari halaman laporan persediaan ketika dicetak.

g. Halaman *Stock of Name*



Gambar 4.59 Implementasi tampilan halaman *stock of name* admin

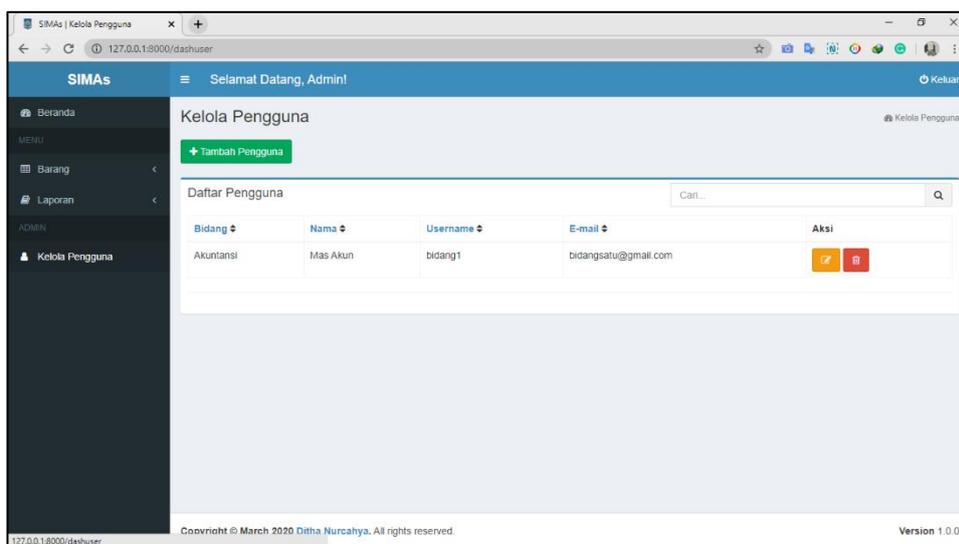
Pada Gambar 4.59 merupakan implementasi dari halaman laporan *stock of name* dan tombol cari untuk melakukan pencarian.



Gambar 4.60 Implementasi cetak *stock of name*

Pada Gambar 4.60 merupakan implementasi dari halaman laporan *stock of name* ketika dicetak.

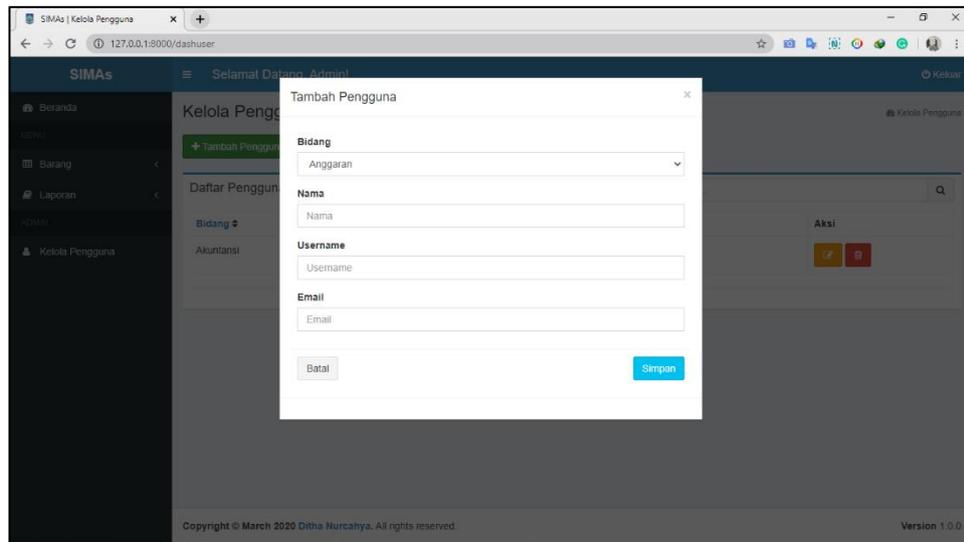
h. Halaman Kelola Pengguna



Gambar 4.61 Implementasi tampilan halaman kelola pengguna admin

Pada Gambar 4.61 merupakan implementasi dari halaman kelola pengguna yang berisi informasi tentang pengguna dan tombol cari untuk melakukan pencarian. Pada halaman kelola pengguna bagi admin juga dilengkapi dengan tombol hapus, dan edit data pengguna.

Tampilan jika pengguna ingin mengisi data pada tabel pengguna :

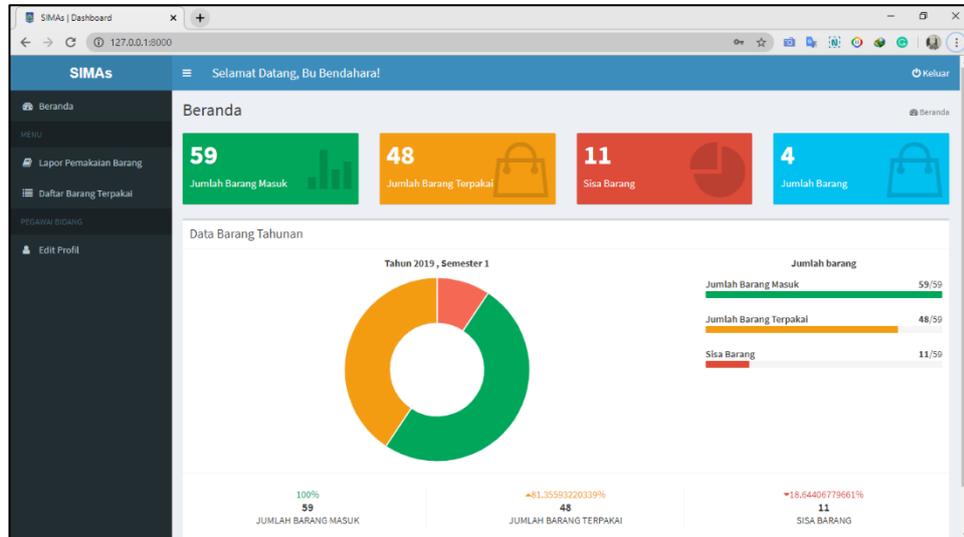


Gambar 4.62 Implementasi tampilan halaman isi data pengguna admin

Pada Gambar 4.62 merupakan implementasi dari halaman isi data pengguna bagi admin.

3. Desain Tampilan Sistem untuk Pegawai Bidang

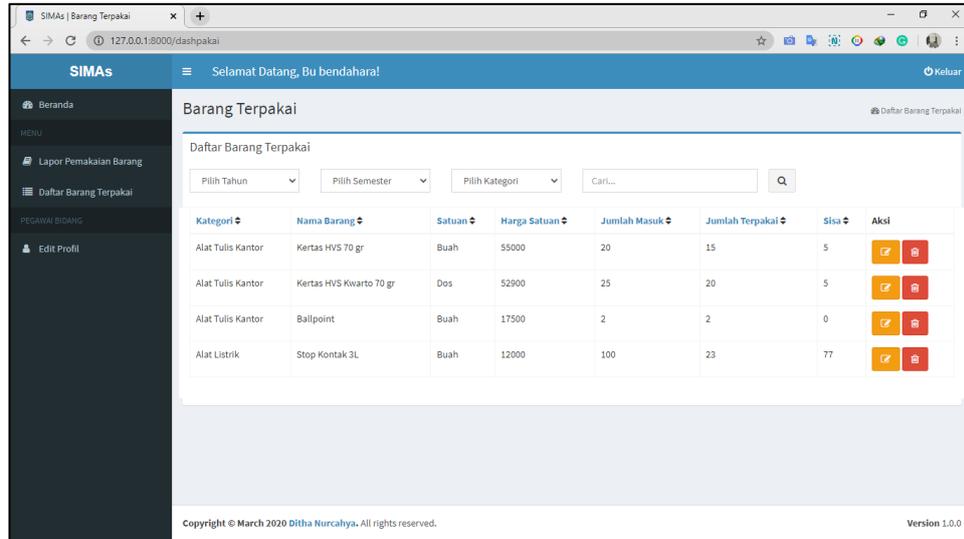
a. Halaman Beranda



Gambar 4.63 Implementasi tampilan halaman beranda pegawai bidang

Pada Gambar 4.63 merupakan implementasi dari halaman beranda yang akan dijumpai pegawai bidang ketika pertama kali masuk ke dalam sistem.

b. Halaman Daftar Barang Terpakai

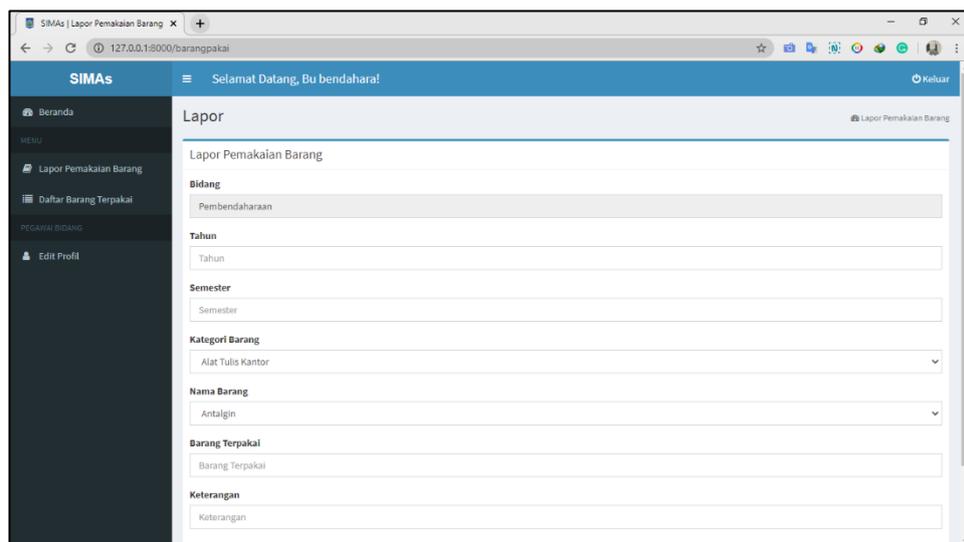


Kategori	Nama Barang	Satuan	Harga Satuan	Jumlah Masuk	Jumlah Terpakai	Sisa	Aksi
Alat Tulis Kantor	Kertas HVS 70 gr	Buah	55000	20	15	5	[Edit] [Hapus]
Alat Tulis Kantor	Kertas HVS Kwarto 70 gr	Dos	52900	25	20	5	[Edit] [Hapus]
Alat Tulis Kantor	Ballpoint	Buah	17500	2	2	0	[Edit] [Hapus]
Alat Listrik	Stop Kontak 3L	Buah	12000	100	23	77	[Edit] [Hapus]

Gambar 4.64 Implementasi tampilan halaman daftar barang terpakai pegawai bidang

Pada Gambar 4.64 merupakan implementasi dari halaman daftar barang terpakai keluar yang berisi informasi tentang barang keluar dan tombol cari untuk melakukan pencarian barang keluar. Pada halaman daftar barang bagi pegawai bidang juga dilengkapi dengan tombol tambah, hapus, dan edit data barang terpakai.

Tampilan jika pengguna ingin mengisi data pada tabel pengguna :



Laporan Pemakaian Barang

Bidang
Pembendaharaan

Tahun
Tahun

Semester
Semester

Kategori Barang
Alat Tulis Kantor

Nama Barang
Antalgin

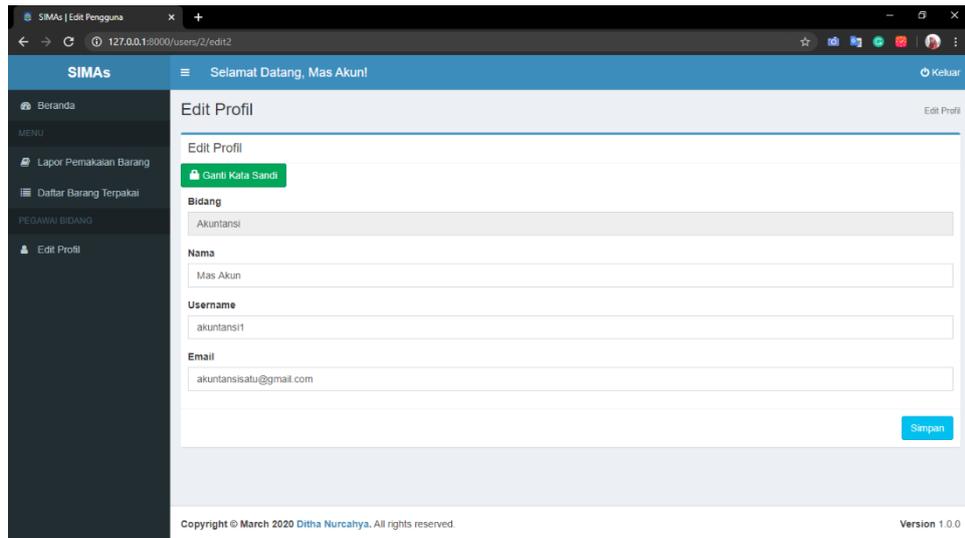
Barang Terpakai
Barang Terpakai

Keterangan
Keterangan

Gambar 4.65 Implementasi tampilan halaman isi data barang terpakai pegawai bidang

Pada Gambar 4.65 merupakan implementasi dari halaman isi data barang terpakai oleh pegawai bidang.

c. Halaman Ubah Profil



Gambar 4.66 Implementasi tampilan halaman ubah profil pegawai bidang

Pada Gambar 4.66 merupakan implementasi dari ubah profil bagi pegawai bidang untuk mengubah data profilnya.

4.3.3 Pengujian

Pengujian MOS (*Mean Opinion Scores*) digunakan untuk mengetahui respon penilaian secara subjektif dari responden terhadap sistem yang dibuat berdasarkan pengamatan hasil *survey* [8]. Adapun yang akan dinilai adalah dari segi kemudahan penggunaan sistem, dan tampilan sistem.

Pengujian dilakukan oleh 5 responden yang terdiri dari 6 pertanyaan, yaitu:

1. Sistem mudah digunakan?
2. Sistem memiliki kecocokan *font*, warna, dan *style* di setiap halaman?
3. Navigasi pada sistem cukup sederhana dan mudah digunakan?
4. Fungsi tiap fitur sudah sangat jelas?
5. Fitur-fitur pada sistem sudah sesuai dengan kebutuhan instansi?
6. Sistem mempermudah pekerjaan pegawai?

Dari pertanyaan tersebut, responden diminta untuk menjawab dengan mencentang sesuai dengan Tabel 4.7.

Tabel 4.7 Bobot nilai MOS

MOS	Keterangan	Bobot Nilai
SS	Sangat Setuju	5
S	Setuju	4

CS	Cukup Setuju	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Data diolah dengan cara mengalikan setiap poin jawaban dengan bobot yang sudah ditentukan sesuai dengan tabel bobot nilai jawaban. Dari hasil perhitungan dengan mengalikan setiap jawaban bobot yang sudah ditentukan maka didapat hasil nilai jumlah bobot, selanjutnya dicari rata rata jawaban dengan membagi jumlah bobot dengan jumlah responden. Setelah didapatkan nilai rata-rata, kemudian menentukan persentase hasil uji dengan membagi nilai rata-rata jawaban dengan banyaknya pilihan jawaban dikalikan dengan 100.

Berikut merupakan hasil pengujian yang dilakukan pada masing masing responden yang berjumlah 5 orang secara subjektif

Tabel 4.8 Hasil Pengujian MOS

No.	Pertanyaan	Jumlah Jawaban dan Nilai					Mean
		SS	S	CS	TS	STS	
1	Sistem mudah digunakan	2	3	0	0	0	88
2	Sistem memiliki kecocokan <i>font</i> , warna, dan <i>style</i> di setiap halaman	1	4	0	0	0	84
3	Navigasi pada sistem cukup sederhana dan mudah digunakan	1	4	0	0	0	84
4	Fungsi tiap fitur sudah sangat jelas	3	2	0	0	0	92
5	Fitur-fitur pada sistem sudah sesuai dengan kebutuhan instansi	0	5	0	0	0	80
6	Sistem mempermudah pekerjaan pegawai	3	2	0	0	0	92
MOS (Mean Opinion Score)							86.67

1. Pengujian *User Interface* dan *User Experience*

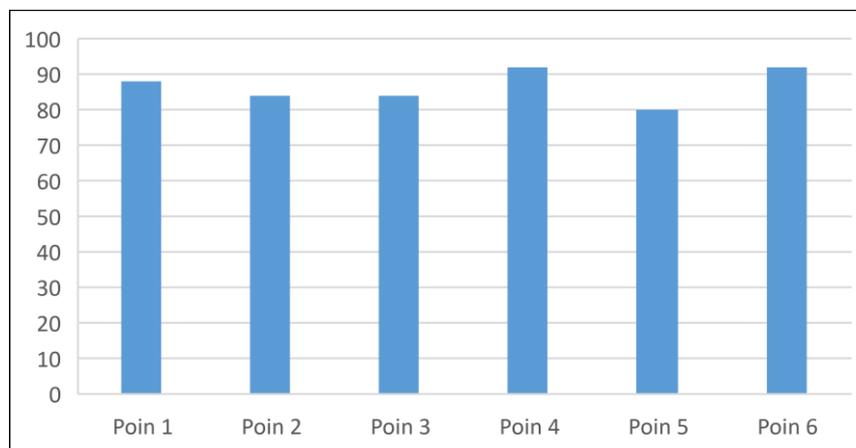
Pengujian *User Interface* dan *User Experience* pada sistem diwakilkan oleh pertanyaan poin 1-3 pada sistem. Dimana didapatkan nilai MOS = 88 dari interval 100 untuk kemudahan penggunaan sistem, nilai MOS = 84 dari interval 100 untuk kecocokan *font*, warna, dan *style* di setiap halaman, dan nilai MOS = 84 dari interval 100 untuk navigasi sistem yang sederhana dan mudah

digunakan. Maka dari itu, untuk menghitung hasil rata-rata dari pengujian *User Interface* dan *User Experience* dari sistem dilakukan penjumlahan hasil dari tiap poin pertanyaan lalu dibagi jumlah poin pertanyaan sehingga didapatkan hasil pengujian *User Interface* dan *User Experience* sebesar 85.34%.

2. Pengujian performa sistem

Pengujian performa pada sistem diwakili oleh pertanyaan poin 4-6 pada sistem. Dimana didapatkan nilai MOS = 92 dari interval 100 untuk kejelasan fungsi dari tiap fitur, nilai MOS = 80 dari interval 100 untuk kesesuaian fitur pada sistem dengan kebutuhan instansi, dan nilai MOS = 92 dari interval 100 untuk sistem yang memudahkan pekerjaan pegawai. Maka dari itu, untuk menghitung hasil rata-rata dari pengujian performa dari sistem dilakukan penjumlahan hasil dari tiap poin pertanyaan lalu dibagi jumlah poin pertanyaan sehingga didapatkan hasil pengujian performa sistem sebesar 88%.

Berdasarkan hasil pengujian MOS yang dilakukan oleh 5 orang responden, pengujian yang dilakukan pada pihak BPKAD Provinsi NTB menghasilkan nilai MOS = 86,67 dari interval 100, menunjukkan bahwa Sistem Informasi Manajemen Aset pada BPKAD Provinsi NTB telah berjalan dengan baik.



Gambar 4.67 Grafik hasil jawaban responden

Adapun grafik hasil perhitungan MOS digambarkan dengan grafik pada Gambar 4.67.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil praktek kerja lapangan yang telah dilakukan dapat disimpulkan bahwa:

1. Sistem Informasi Manajemen Aset pada BPKAD Provinsi NTB telah berjalan dengan baik berdasarkan hasil pengujian yang dilakukan pada pihak BPKAD yang menghasilkan nilai *Mean Opinion Score* sebesar 86,67%.
2. Pengujian proses yang ada dan performa dari Sistem Informasi Manajemen Aset pada BPKAD Provinsi NTB sudah sesuai dengan kebutuhan instansi dengan hasil persentase jawaban kuesioner sebesar 88%.
3. Pengujian *User Interface* dan *User Experience* dari Sistem Informasi Manajemen Aset pada BPKAD Provinsi NTB menunjukkan tampilan sistem menarik dan sistem mudah digunakan oleh pengguna dengan persentase jawaban kuesioner sebesar 85,34%.

5.2 Saran

Adapun saran yang dapat diberikan penulis agar Sistem Informasi Manajemen Aset ini menjadi lebih baik di masa yang akan datang yaitu dalam pengembangan selanjutnya, diharapkan sistem informasi dapat menerima data barang terpakai dari para pegawai setiap bidang, yang terintegrasi dengan sistem berbasis mobile untuk memudahkan pengisian data persediaan barang yang terdapat pada Badan Pengelolaan Keuangan dan Aset Daerah Provinsi Nusa Tenggara Barat.

DAFTAR PUSTAKA

- [1] Yakub, *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu, 2012.
- [2] M. R. Pratama, B. Pangayow, and M. Si, “Pengaruh Manajemen Aset Terhadap Optimalisasi,” vol. 11, no. November, pp. 33–51, 2016.
- [3] B. Barthos, *Manajemen Kearsipan*. Jakarta: Bumi Aksara, 2009.
- [4] H. Hidayat, *Cara Instan Menguasai Pemrograman Website Secara Otodidak*. Jakarta Barat: Agogos Publishing, 2016.
- [5] D. Widodo, *Menggunakan UML*. Bandung: Informatika, 2011.
- [6] B. Raharjo, *Belajar otodidak membuat database menggunakan MySQL*. Bandung: Informatika, 2011.
- [7] Prijambodo, *Out of The Box Koperasi tantangan perubahan kini dan masa depan*. Yogyakarta: Ahmed Ghoseen A, 2018.
- [8] Z. Rahmiati, I. G. P. S. Wijaya, and B. Irmawati, “The Analysis System Students Satisfaction Level Toward Teaching and Learning Process in PSTI Unram Using Service Quality Method,” *JTIKA*, vol. 1, no. 1, pp. 31–40, 2019.