

Belajar Membangun Sistem Informasi

Penyimpanan Barang
Milik Daerah Berbasis Web

Perkembangan Teknologi Informasi dan Komunikasi (TIK) pada saat ini sudah sangatlah pesat, tidak terkecuali di instansi pemerintahan baik pusat, provinsi, kabupaten/ kota, ataupun Kecamatan berbagai inovasi teknologi terus dikembangkan dalam meningkatkan pelayanan terhadap masyarakat. Pada tingkat pemerintah daerah, dalam proses pelayanan kepada masyarakat yang optimal tidak terlepas dari tersedianya Barang milik daerah (BMD), Barang milik daerah merupakan salah satu aset yang paling penting yang dimiliki daerah guna menunjang operasional jalannya pemerintahan daerah. dengan adanya barang milik daerah maka pencapaian pembangunan daerah dapat terlaksana guna kesejahteraan masyarakat.

Barang Milik Daerah adalah semua barang yang dibeli atau diperoleh atas beban Anggaran Pendapatan dan Belanja Daerah atau berasal dari perolehan lainnya yang sah. Barang milik daerah meliputi aset berwujud dan aset tidak berwujud. Aset berwujud dibagi menjadi dua, yaitu aset tetap dan aset tidak tetap. Aset tetap adalah semua aset tidak cepat habis yang memiliki waktu ekonomis lebih dari satu tahun atau dalam kurun waktu yang lama. Aset tetap tersebut dikategorikan menjadi dua, yaitu aset bergerak dan aset tidak bergerak. Aset bergerak, contohnya ; mobil, motor, truk, dan lain-lain, sedangkan aset tidak bergerak ; gedung, jalan, tanah, irigasi, dan lain-lain.



Anggota IKAPI
Ds. Kalianyar RT. 003/ RW. 002, Kec. Ngronggot, Kab. Nganjuk Jatim

www.dewapublishing.com
publishingdewa@gmail.com
dewapublishing
0877-7141-5004



DEWA
PUBLISHING

Belajar Membangun Sistem Informasi Penyimpanan Barang Milik Daerah Berbasis Web

Rohmat Taufiq, ST., M.Kom, dkk.

Belajar Membangun Sistem Informasi

Penyimpanan Barang
Milik Daerah Berbasis Web

Rohmat Taufiq, ST., M.Kom
Yeni Daniarti, S.Pd., M.Pd
Muhamad Fhurqon Febriansyah, S.Kom
Angga Aditya Permana, M.Kom

DEWA
PUBLISHING

**BELAJAR MEMBANGUN SISTEM INFORMASI
PENYIMPANAN BARANG MILIK DAERAH BERBASIS
WEB**

Rohmat Taufiq, ST., M.Kom

Yeni Daniarti, S.Pd., M.Pd

Muhamad Fhurqon Febriansyah, S.Kom

Angga Aditya Permana, M.Kom

**BELAJAR MEMBANGUN SISTEM INFORMASI
PENYIMPANAN BARANG MILIK DAERAH BERBASIS
WEB**

ISBN : 978-623-8016-98-3

Penulis : Rohmat Taufiq, ST., M.Kom
Yeni Daniarti, S.Pd., M.Pd
Muhamad Fhurqon Febriansyah, S.Kom
Angga Aditya Permana, M.Kom

Editor : Achmad Wahdi, S.Kep.,Ns.,M.Tr.Kep.

Desain Cover : Redaksi Dewa Publishing

Layout : Irsyadul Mubin

15,5 cm x 23 cm

102 Halaman

Cetakan Pertama, Desember 2022

Penerbit CV. Dewa Publishing

Redaksi:

Desa Kalianyar RT 003/RW 002, Kec. Ngronggot,
Kab. Nganjuk, Jawa Timur

Email : publishingdewa@gmail.com

Website : www.dewapublishing.com

Phone : 0819-1810-0313

Anggota IKAPI 341/ JTI/ 2022

@Hak Cipta Dilindungi Undang-Undang

Dilarang memperbanyak, mencetak atau menerbitkan sebagian isi atau seluruh buku dengan cara apapun juga tanpa seizin editor dan penerbit.

KATA PENGANTAR

Puji syukur penyusun panjatkan kehadirat Allah SWT, karena hanya atas rahmat dan hidayah – Nya penulisan buku ini dapat tersusun hingga selesai.

Dalam melakukan penulisan ini, penyusun telah melibatkan berbagai pihak, untuk itu penyusun ucapkan banyak terimakasih kepada keluarga, rekan kerja dan semua pihak yang telah memberikan semangat dan membantu kami sehingga penulisan buku ini dapat terselesaikan. Kami juga menyadari didalam penulisan ini masih belum terlalu sempurna. Untuk itu masukan yang bersifat membangun sangat kami kami butuhkan untuk menyempurnakan penulisan buku ini.

Pada akhirnya kami berharap dengan adanya buku ini, akan memberikan wawasan bagi pembaca dan membantu para pembelajar.

Tangerang, 30 November 2022

Rohmat Taufiq, ST., M.Kom

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
BAB II LANDASAN TEORI	4
2.1 Perancangan	4
2.2 Sistem	4
2.3 Informasi	10
2.4. Sistem Informasi	12
2.5 PHP (Hypertext Preprocessor)	14
2.6 MySQL.....	17
2.7 XAMPP	20
2.8 Basis Data.....	20
2.9 Relationship.....	23
2.10 SDLC (Sistems Development Life Cycle)	26
2.11 Unified Modelling Language (UML).....	27
BAB III ANALISIS DAN PEMBAHASAN.....	44
3.1 Analisis Sistem.....	44
3.2 Rancangan Program	48
3.3 Testing.....	91
3.4 Pemeliharaan	95

BAB IV PENUTUP	97
DAFTAR PUSTAKA	98
TENTANG PENULIS.....	100

DAFTAR GAMBAR

Gambar 3. 1 Use Case Diagram Sistem yang sedang Berjalan .	45
Gambar 3. 2 Use Case Diagram Sistem Penerimaan Barang Milik Daerah.....	49
Gambar 3. 3 Activity Diagram Login User	71
Gambar 3. 4 Activity Diagram Pengelolaan data Lokasi	72
Gambar 3. 5 Activity Diagram Pengelolaan data Pegawai.....	73
Gambar 3. 6 Activity Diagram Pengelolaan data Barang.....	74
Gambar 3. 7 Activity Diagram Pengelolaan data Jenis Perolehan	75
Gambar 3. 8 Activity Diagram Penerimaan Barang	76
Gambar 3. 9 Activity Diagram Laporan	77
Gambar 3. 10 Class diagram sistem informasi penerimaan barang milik daerah	78
Gambar 3. 11 Rancangan antarmuka Form Login.....	79
Gambar 3. 12 Rancangan antarmuka halaman Dashboard	80
Gambar 3. 13 Rancangan antarmuka halaman Data Pegawai	81
Gambar 3. 14 Rancangan antarmuka halaman Data Barang	81
Gambar 3. 15 Rancangan antarmuka halaman Penerimaan Barang	82
Gambar 3. 16 Rancangan antarmuka halaman Jenis Perolehan .	83
Gambar 3. 17 Rancangan antarmuka halaman Data Lokasi	83
Gambar 3. 18 Form Login	84
Gambar 3. 19 Dashboard	85
Gambar 3. 20 Halaman Pegawai.....	85

Gambar 3. 21 Halaman Tambah Data Pegawai	86
Gambar 3. 22 Halaman Data Barang	86
Gambar 3. 23 Halaman Tambah data barang	87
Gambar 3. 24 Halaman Penerimaan Barang	87
Gambar 3. 25 Halaman Penerimaan Barang	88
Gambar 3. 26 Print Berita Acara Penerimaan	88
Gambar 3. 27 Halaman Jenis Perolehan.....	89
Gambar 3. 28 Halaman Tambah Jenis Perolehan	89
Gambar 3. 29 Halaman Data Lokasi	90
Gambar 3. 30 Halaman Tambah data Lokasi	90

DAFTAR TABEL

Tabel 3. 1 Penjelasan Usecase sistem Penerimaan barang milik daerah yang sedang berjalan.....	45
Tabel 3. 2 Penjelasan Aktor <i>Use Case</i> Sistem Informasi Penerimaan Barang Milik Daerah	50
Tabel 3. 3 Deskripsi Use Case Login.....	50
Tabel 3. 4 Penjelasan <i>Use Case</i> Tambah Data Barang	51
Tabel 3. 5 Penjelasan <i>Use Case</i> Ubah Data Barang	52
Tabel 3. 6 Penjelasan <i>Use Case</i> Hapus Data Barang.....	53
Tabel 3. 7 Penjelasan <i>Use Case</i> Tambah Data Pegawai	55
Tabel 3. 8 Penjelasan <i>Use Case</i> Ubah Data Pegawai	56
Tabel 3. 9 Penjelasan <i>Use Case</i> Hapus Data Pegawai.....	57
Tabel 3. 10 Penjelasan <i>Use Case</i> Tambah Data Lokasi Barang .	58
Tabel 3. 11 Penjelasan <i>Use Case</i> Ubah Data Lokasi Barang	59
Tabel 3. 12 Penjelasan <i>Use Case</i> Hapus Data Lokasi Barang....	60
Tabel 3. 13 Penjelasan <i>Use Case</i> Tambah Data Jenis Perolehan Barang	61
Tabel 3. 14 Penjelasan <i>Use Case</i> Ubah Data Jenis Perolehan Barang	62
Tabel 3. 15 Penjelasan <i>Use Case</i> Hapus Data Jenis Perolehan Barang	64
Tabel 3. 16 Penjelasan <i>Use Case</i> Tambah Data Penerimaan Barang	65
Tabel 3. 17 Penjelasan <i>Use Case</i> Ubah Data Penerimaan Barang	66

Tabel 3. 18 Penjelasan <i>Use Case</i> Hapus Data Penerimaan Barang	67
Tabel 3. 19 Penjelasan <i>Use Case</i> Print Berita Acara Penerimaan Barang.....	68
Tabel 3. 20 Penjelasan <i>Use Case</i> Laporan	70
Tabel 3. 21 Hasil Pengujian Dengan Pendekatan Black-Box	91

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan Teknologi Informasi dan Komunikasi (TIK) pada saat ini sudah sangatlah pesat, tidak terkecuali di instansi pemerintahan baik pusat, provinsi, kabupaten/ kota, ataupun Kecamatan berbagai inovasi teknologi terus dikembangkan dalam meningkatkan pelayanan terhadap masyarakat.

Pada tingkat pemerintah daerah, dalam proses pelayanan kepada masyarakat yang optimal tidak terlepas dari tersedianya Barang milik daerah (BMD), Barang milik daerah merupakan salah satu aset yang paling penting yang dimiliki daerah guna menunjang operasional jalannya pemerintahan daerah. dengan adanya barang milik daerah maka pencapaian pembangunan daerah dapat terlaksana guna kesejahteraan masyarakat.

Barang Milik Daerah adalah semua barang yang dibeli atau diperoleh atas beban Anggaran Pendapatan dan Belanja Daerah atau berasal dari perolehan lainnya yang sah. Barang milik daerah meliputi aset berwujud dan aset tidak berwujud. Aset berwujud dibagi menjadi dua, yaitu aset tetap dan aset tidak tetap. Aset tetap adalah semua aset tidak cepat habis yang memiliki waktu ekonomis lebih dari satu tahun atau dalam kurun waktu yang lama. Aset tetap tersebut dikategorikan menjadi dua, yaitu aset

bergerak dan aset tidak bergerak. Aset bergerak, contohnya ; mobil, motor, truk, dan lain-lain, sedangkan aset tidak bergerak ; gedung, jalan, tanah, irigasi, dan lain-lain.

Mengingat pentingnya pengelolaan barang milik daerah (BMD) maka diperlukan sebuah pengelolaan dan inventarisasi yang baik. berdasarkan peraturan Menteri dalam negeri Nomor 54 tahun 2010 tentang Pedoman Teknis Pengelolaan Barang Milik Negara/Daerah dapat dipahami bahwa inventarisasi adalah kegiatan untuk melakukan pendataan, pencatatan, dan pelaporan hasil pendataan barang. Sesuai peraturan tersebut perlu adanya proses inventarisasi yang baik dan benar. sehingga terwujud Pengelolaan Barang Milik Negara/Daerah yang transparan, efisien, akuntabel, ekonomis serta menjamin adanya kepastian nilai.

Kecamatan Tangerang sebagai bagian dari perangkat daerah kota Tangerang sebagai pelaksana teknis kewilayahan memiliki tugas dalam melaksanakan administrasi dan pengelolaan inventaris barang milik daerah. Dalam pelaksanaannya berdasarkan hasil observasi dilapangan, proses pengelolaan data penerimaan barang milik daerah yang ada dikecamatan Tangerang masih di lakukan dengan proses manual dengan *worksheet*, dan Adapun untuk proses administrasi dan pembuatan surat berita acara serah terima barang mengguna aplikasi Microsoft word.

Dari dari poses tersebut penulis menemukan beberapa permasalahan dimana terjadi kesulitan untuk mengelola data dengan jumlah yang banyak, pembuat surat berita acara penerimaan barang yang masih manual membuat satu persatu. Serta pelaporan data barang yang belum optimal.

BAB II

LANDASAN TEORI

2.1 Perancangan

2.1.1 Pengertian Perancangan

Perancangan adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah dan suatu kesatuan yang utuh dan berfungsi (Yakub, 2012)..

Menurut John W Satzinger, Robert B Jackson, dan Stephen D Burd (2012:5), perancangan sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Hal itu bertujuan untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan pengguna. basis komputer yang telah dipilih selama tahap analisis.

Secara umum perancangan bertujuan untuk memberikan gambaran jelas dari sistem yang dirancang dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan lengkap pada programmer komputer serta ahli teknik yang terlibat didalamnya. Berikut ini beberapa pendapat tentang perancangan sistem.

2.2 Sistem

2.2.1 Pengertian Sistem

Menurut Sucipto (2011 : 1), Sistem adalah kumpulan elemen-elemen yang saling berkaitan dan bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran

(*output*). Proses yang terjadi disini dapat terdiri dari bermacam-macam tipe yaitu :

1. Proses Fisik
2. Proses Prosedural

2.2.2 Konsep Dasar Sistem

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Prosedur (*Procedure*) didefinisikan oleh Richard F. Neuschel sebagai berikut.

Suatu prosedur adalah suatu urutan-urutan operasi klerikal (tulis menulis), biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi.

Prosedur menurut **Jerry Fitz Gerald, Ardra F. FitzGerald** dan **Warren D. Stallings, Jr.** Definisinya adalah sebagai berikut.

Suatu prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan,

- 1) *Apa (what) yang harus dikerjakan,*
- 2) *Siapa (who) yang mengerjakannya,*
- 3) *Kapan (when) dikerjakan,*
- 4) *Bagaimana (how) mengerjakannya.*

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Kesimpulan ada 2 pendekatan :

1. Secara komponen
2. Secara prosedur

2.2.4 Karakteristik Sistem

Sistem yang memiliki sifat-sifat atau karakteristik. Sistem mempunyai beberapa komponen, yaitu :

1. Komponen Sistem

Suatu sistem dapat terdiri dari beberapa komponen yang saling berhubungan satu dengan yang lain dan bekerja sama membentuk satu kesatuan, sistem dapat terdiri dari bagian-bagian yang mempunyai fungsi yang berbeda dan saling terhubung.

2. Batas Sistem

Setiap sistem terdiri dari sub-sub sistem dan setiap sistem atau sub sistem mempunyai tata kerja yang berbeda antara satu dengan yang lain, sistem mempunyai kemampuan yang terbatas dan ruang lingkup yang terbatas. Batasan,

merupakan batasan-batasan yang ada dalam mencapai tujuan dari sistem, dimana batasan ini dapat berupa :

- a. Prosedur
- b. Biaya-biaya
- c. Personel
- d. Peralatan
- e. Mekanisme pengolahan

3. Lingkungan Luar Sistem

Baik dan buruknya kinerja sistem dapat dipengaruhi oleh lingkungan luar sistem.

4. Penghubung Sistem

Untuk dapat berkomunikasi satu sub sistem dengan sub sistem yang lain didukung adanya penghubung, *input* suatu sistem dapat berasal dari *output* sub sistem yang lain.

5. Masukan Sistem

Sesuatu yang dimasukkan dalam sistem sehingga sistem dapat melakukan proses. *Input* merupakan bagian dari sistem yang bertugas untuk menerima data masukan, dimana data dapat berupa :

- a. Asal masukan
- b. Frekuensi pemasukan data
- c. Jenis pemasukan data

6. Keluaran Sistem

Hasil dari inputan yang telah dimanipulasi menjadi bentuk yang berbeda. *Output*, Merupakan keluaran atau tujuan akhir dari sistem, *output* dapat berupa:

- a. Laporan
- b. Grafik

7. Pengolah Sistem

Mekanisme manipulasi yang disebabkan dari masukan dengan menggunakan model atau metode tertentu. Proses, merupakan bagian yang memproses masukan data menjadi informasi sesuai dengan keinginan penerima:

- a. Klasifikasi
- b. Peringkasan
- c. Pencaharian

8. Sasaran Sistem

Sesuatu yang dapat dicapai dari perpaduan *input*, proses, dan *output* yang berupa sasaran (*Objective*) atau tujuan (*goal*). Tujuan, merupakan tujuan dari sistem tersebut yang dapat berupa :

- a. Tujuan usaha
- b. Kebutuhan
- c. Masalah
- d. Prosedur pencapaian tujuan

9. Kontrol Sistem

Merupakan pengawas dari pelaksanaan pencapaian tujuan sistem, yang dapat berupa :

- a. Kontrol pemasukan data (*input*)
- b. Kontrol pengeluaran data (*output*)
- c. Kontrol pengoperasian.

2.2.5 Klasifikasi Sistem

Untuk dapat memudahkan identifikasi sistem dapat diklasifikasikan menjadi beberapa kelompok sebagai berikut.

1. Sistem abstrak (*abstract sistem*) dan sistem fisik (*physical sistem*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem Alamiah (*natural sistem*) dan sistem buatan (*human made sistem*)

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia melibatkan interaksi antara manusia dan mesin.

3. Sistem tertentu (*deterministic sistem*) dan sistem tidak tentu probabilistik (*probabilistic sistem*)

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tidak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem tertutup (*closed sistem*) dan sistem terbuka (*open sistem*)

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Sebaliknya, sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.3 Informasi

2.3.1 Pengertian Informasi

Menurut Sucipto (2011:7), Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Dalam dunia bisnis, kejadian-kejadian nyata yang sering terjadi adalah perubahan dari suatu nilai yang disebut dengan transaksi.

2.3.2 Siklus Informasi

Data ditangkap, diinputkan dengan menggunakan metode tertentu, diproses/olah dengan menggunakan model pengolahan data tertentu sehingga menjadi informasi yang berupa tabel atau grafik yang didistribusikan kepada penerima yang dijadikan acuan dalam pengambilan keputusan dan mengambil tindakan

dapat dijadikan data kembali dan terus melalui tahapan-tahapan dalam siklus informasi.

Data bersumber dari bagian Perkuliahan dan Ujian, diberikan kepada masing-masing kepala jurusan atau program studi, hasil rekap diberikan kepada Asisten Akademik, dilaporkan kepada Direktur atau Rektor untuk diambil keputusan. Hasil keputusan dapat dijadikan data kembali.

2.3.3 Kualitas Informasi

Kualitas suatu informasi (*quality of information*) tergantung dari tiga hal :

1. Bebas dari kesalahan (*accurate*) informasi berkualitas jika informasi tersebut menunjukkan kebenaran.
2. Tepat pada waktunya (*up date*) informasi berkualitas jika informasi tersebut disampaikan tepat waktu.
3. Bermanfaat (*relevan*) informasi berkualitas jika informasi tersebut memiliki nilai bagi yang menerimanya.

2.3.4 Nilai Informasi

Nilai dari suatu informasi ditentukan dari dua hal, yaitu :

1. Manfaat
2. Biaya

Suatu informasi akan bernilai jika manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* atau *cost benefit*.

2.4. Sistem Informasi

2.4.1 Pengertian Sistem Informasi

Menurut Sucipto (2011:10), Informasi dapat diperoleh dari sistem informasi (*information sistem*) atau disebut juga dengan *processing – generating sistem*. Kesimpulannya Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan :

1. Pengolahan transaksi harian,
2. Mendukung operasi,
3. Bersifat menajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan.

2.4.2 Komponen Sistem Informasi

Komponen sistem informasi yang saling terkait.

1. Blok Masukan (*Input Block*)

Blok masukan akan berhubungan dengan hal – hal yang berkaitan dengan *input*, dapat meliputi bagaimana sebuah transaksi diinputkan dan diproses oleh sistem. Input dapat terdiri dari transaksi – transaksi atau data – data / database. Metode input data yang dapat digunakan dapat terdiri dari dua macam cara yaitu:

a. Batch

Data tersebut dikumpulkan terlebih dahulu kemudian inputkan.

b. Direct

Data dapat diinputkan secara langsung.

2. Blok Model (*Model Block*)

Merupakan gabungan prosedur logika dan model matematik yang akan memanipulasi data input dan menghasilkan informasi.

3. Blok Keluaran (*Output Block*)

Hasil dari proses yang berupa informasi yang dapat membantu yang menerimanya dalam pengambilan keputusan.

4. Blok Teknologi (*Technology Block*)

Yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data.

5. Blok Basis Data (*Database Block*)

Kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat komputer dan digunakan perangkat lunak untuk memanipulasi data.

6. Blok Kendali (*Control Block*)

Blok yang berfungsi untuk menghindari kesalahan dan terjadinya penyimpangan *input control*.

2.5 PHP (Hypertext Preprocessor)

2.5.1 Pengertian PHP

Furi Rusliantoro (2016), PHP (*Hypertext Preprocessor*) adalah salah satu *server side scripting* yang didesain khusus untuk aplikasi web. PHP dapat disisipkan diantara bahasa HTML dan karena *server side scripting*, maka skrip PHP akan dieksekusi di server sehingga yang dikirimkan ke *browser* adalah “hasil jadi“ dalam bentuk HTML, dan kode PHP tidak akan terlihat.

2.5.2 Sejarah PHP

PHP pertama kali dibuat pada musim gugur tahun 1994 oleh Rasmus Lerdoff, yang digunakan oleh websitenya untuk mencatat siapa saja yang berkunjung dan melihat biodatanya. Versi pertama yang *release* tersedia pada awal tahun 1995. Dikenal sebagai *tool Personal Home Page*, yang terdiri atas *engine parser* yang sangat sederhana yang hanya mengerti beberapa makro khusus dan sejumlah utilitas yang sering digunakan pada halaman-halaman web, seperti buku tamu, *counter* pengunjung dan lainnya. Parser diprogram ulang pada pertengahan 1995 dan diberi nama PHP/FI *verse* 2.0. FI berasal dari paket Rasmus lainnya yang ditulis untuk menginterpretasikan data dari form, yang kemudian dikombinasikan dengan *tool Personal Home Page* dan ditambahkan dukungan untuk database mSQL (*mini SQL*).

Pada tahun 1996 pengguna PHP/FI diperkirakan mencapai 15.000 *website* di seluruh dunia. Dan pada pertengahan 1997

mencapai 50.000 situs. Pada saat itu juga terdapat perubahan dalam pengembangan PHP. PHP berubah dari proyek pribadi Rasmus menjadi sebuah tim yang lebih terorganisasi. Parsenya dikembangkan oleh Zeev Suraski dan Andi Gutsman yang kemudian menjadi dasar untuk versi 3, dan banyak utilitas tambahan yang diprogram untuk menambah kemampuan dari versi sebelumnya.

2.5.3 Pendekatan Cara PHP

PHP menawarkan solusi yang lebih luwes. Dengan PHP, *developer* tidak perlu lagi berurusan dengan dua buah file terpisah. *Browser* web mengacu secara langsung ke file yang dituju, yang lalu dibaca oleh server sebagaimana file HTML statis biasa. Bedanya sebelum dikirim balik ke *browser*, server memeriksa isi file dan menentukan apakah ada kode di dalam file tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya dimasukkan ke dalam dokumen yang sama. Server bekerja secara langsung terhadap file yang bersangkutan, tidak memanggil skrip terpisah seperti pada metode CGI. Seluruh kode dieksekusi di server, oleh karena itu disebut *server side scripting*.

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1. *Script* terintegrasi dengan file HTML, sehingga *developer* bisa berkonsentrasi langsung pada penampilan dokumen webnya.
2. Tidak ada proses *compiling* dan *linking*.
3. Berorientasi Objek (*object oriented*).
4. Sintaksis pemrogramannya mudah dipelajari.
5. Integrasi yang sangat luas ke berbagai server *database*. Menulis web yang terhubung ke *database* menjadi sangat sederhana. *Database* yang didukung oleh PHP adalah : Oracle, Sybase, mSQL, MySQL, Solid, ODBC, PostreSQL, Adabas D, FilePro, Velocis, Informix, dBase, UNIX dbm dan lain-lain.

2.5.4 Penulisan PHP

Sebelum pemrograman PHP, yang harus diperhatikan adalah apakah web server dan skrip PHP sudah berjalan dengan baik atau tidak. Fungsi-fungsi yang ada dalam PHP tidak *case sensitive* tetapi variabelnya *case sensitive* (membedakan huruf besar dan kecil).

Ada tiga cara untuk penulisan skrip PHP, yaitu :

1. <?

Script PHP

?>

2. <?php

Script PHP

?>

3. <SCRIPT LANGUAGE =” php”>

Script PHP

<SCRIPT>

Pemisah antar instruksi adalah titik koma (;) dan untuk menambah komentar, standar penulisannya adalah /*komentar*/, //komentar dan #komentar.

2.6 MySQL

2.6.1 Pengertian MySQL

Furi Rusliantoro (2016), MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). MySQL merupakan turunan salah satu turunan SQL (*Structure Query Language*) dimana sebuah konsep pengoperasian *database* terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TcX Data Konsult AB sekitar tahun 1944-1995. MySQL merupakan multiuser database yang menggunakan bahasa *Structured Query Language* (SQL). SQL (dibaca “*ess-que-el*”) adalah bahasa standar ANSI (*American National Standarts Institute*), yang

digunakan untuk mengakses server database atau lebih dikenal dengan *relational database management systems* (RDBMS). Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan basis data perusahaan-perusahaan skala menengah dan kecil. Istilah seperti tabel, baris, dan kolom tetap digunakan dalam MySQL.

MySQL mempunyai tampilan *client* yang mempermudah kita dalam mengakses database dengan kata sandi untuk mengijinkan proses yang boleh dilakukan . Sedangkan PHP MyAdmin merupakan halaman yang terdapat pada web server.

Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti *update* data pada database, atau menampilkan data dari database. Beberapa *software* RDBMS dan dapat menggunakan SQL, seperti Oracle, Sybase, Microsoft SQL Server, Microsoft Access, Ingres, dsb. Setiap software database mempunyai bahasa perintah / sintaks yang berbeda, namun pada prinsipnya mempunyai arti fungsi yang sama.

Perintah-perintah tersebut antara lain “*Select*”, “*Insert*”, “*Update*”, “*Delete*”, “*Create*”, dan “*Drop*”. Yang dapat digunakan untuk mengerjakan hampir semua kebutuhan untuk memanipulasi sebuah database. MySQL mempunyai beberapa keunggulan, antara lain :

1. MySQL merupakan program yang *multi-threaded*, sehingga dapat dipasang pada server yang memiliki multi-CPU.
2. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, TCL, APIs.
3. Bekerja pada berbagai *platform*. (tersedia berbagai versi untuk berbagai sistem operasi).
4. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem database.
5. Memiliki sistem sekuriti yang cukup baik dengan verifikasi host.
6. Mendukung ODBC untuk sistem operasi Microsoft Windows.
7. Mendukung record yang memiliki kolom dengan panjang tetap atau panjang bervariasi dan masih banyak keunggulan lainnya.
8. MySQL merupakan *software* yang free, dan bisa didownload di www.mysql.com. Sedangkan *software* database lainnya seperti ORACLE merupakan software yang harus dibeli.
9. MySQL dan PHP saling terintegrasi. Maksudnya adalah pembuatan *database* dengan menggunakan sintak PHP dapat di buat. Sedangkan input yang dimasukkan melalui aplikasi web yang menggunakan *script server-side* seperti PHP dapat langsung dimasukkan ke *database* MySQL yang ada di server dan tentunya web tersebut berada di sebuah web server.

2.7 XAMPP

XAMPP adalah program aplikasi yang berguna untuk pengembangan *website* berbasis PHP dan MySQL. Versi terbaru program ini adalah XAMPP 1.8.3. *Software* XAMPP dibuat dan dikembangkan oleh *Apache Friends*. Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai *server web apache* untuk simulasi pengembangan *website*. *Tool* pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl.

Melalui program ini, *programmer web* dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu koneksi ke internet. XAMPP juga dilengkapi fitur manajemen basis data PHP MyAdmin seperti pada *server hosting* sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis *database* secara mudah. Program XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman web.

2.8 Basis Data

2.8.1 Pengertian Basis Data

Basis Data (*database*) adalah suatu kumpulan data yang disusun dalam bentuk tabel-tabel yang saling berkaitan maupun berdiri sendiri dan disimpan secara bersama-sama pada suatu media. Basis data dapat digunakan oleh satu atau lebih program

aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya. Terdapat beberapa aturan yang harus dipatuhi pada *file* basis data agar dapat memenuhi kriteria sebagai suatu basis data. Beberapa aturan itu berhubungan dengan :

1. Kerangkapan data, yaitu munculnya data-data yang sama secara berulang-ulang pada *file* basis data.
2. Inkonsistensi data, yaitu munculnya data yang tidak konsisten pada *field* yang sama untuk beberapa *file* dengan kunci yang sama.
3. Data terisolasi, disebabkan oleh pemakaian beberapa *file* basis data. Program aplikasi tidak dapat mengakses *file*. Tertentu dalam *sistem* basis data tersebut, kecuali program aplikasi dirubah atau ditambah sehingga seolah-olah ada *file* yang terpisah atau terisolasi terhadap *file* yang lain.
4. Keamanan data, berhubungan dengan masalah kewanitaan data dalam sistem basis data. Pada prinsipnya *file* basis data hanya boleh digunakan oleh pemakai tertentu yang mempunyai wewenang untuk mengakses.
5. Integrasi data, berhubungan dengan unjuk kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh.

2.8.2 Bagian-Bagian Basis Data

Membangun basis data adalah langkah awal dari pembuatan sebuah aplikasi. Keberhasilan dalam membangun basis data akan menyebabkan program lebih mudah dibaca, mudah dikembangkan dan mudah mengikuti perkembangan perangkat lunak. Berikut ini diuraikan mengenai komponen-komponen yang terdapat dalam basis data :

1. Tabel

Tabel adalah kumpulan dari suatu *field* dan *record*. Dalam hal ini biasanya *field* ditunjukkan dalam bentuk kolom dan *record* ditunjukkan dalam bentuk baris.

2. Field

Field adalah sebutan untuk mewakili suatu *record*. Misalnya seorang pegawai dapat dilihat datanya melalui *field* yang diberikan padanya seperti nip, nama, alamat, dan lain-lain.

3. Record

Record adalah kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu isi data secara lengkap. Satu *record* mewakili satu data atau informasi tentang seseorang misalnya, nomor daftar, nama pendaftar, alamat, tanggal masuk.

4. Primary Key

Primary Key adalah suatu kolom (*field*) yang menjadi titik acuan pada sebuah tabel, bersifat unik dalam artian tidak ada satu nilai pun yang sama atau kembar dalam tabel tersebut, dan dalam satu tabel hanya boleh ada satu *primary key*.

5. Foreign Key

Foreign key atau disebut juga kunci relasi adalah suatu kolom dalam tabel yang digunakan sebagai “kaitan” untuk melengkapi satu hubungan yang didapati dari tabel induk, dan biasanya hubungan yang terjalin antar tabel adalah satu ke banyak (*one to many*).

6. Index

Index adalah struktur basis data secara fisik, yang digunakan untuk optimalisasi pemrosesan data dan mempercepat proses pencarian data.

2.9 Relationship

Basis data adalah kumpulan file yang saling berkaitan. Pada model data relasional hubungan antar *file* direlasikan dengan kunci relasi (*relation key*), yang merupakan kunci utama dari masing-masing *file*. Perancangan basis data yang tepat akan menyebabkan paket program relasional akan bekerja secara optimal.

Relasi antara dua *file* atau dua tabel dapat dikategorikan menjadi tiga macam. Demikian pula untuk membantu gambaran

relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan *attribute* dalam satu file.

1. *One to one relationship 2 file*

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu. Seperti pada pelajaran privat dimana satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru pula. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.

2. *One to many relationship 2 file*

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu. Seperti pada sistem pengajaran di sekolah dasar dimana satu guru mengajar banyak siswa dan siswa hanya diajar satu guru. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

3. *Many to many relationship 2 file*

Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di sekolah menengah dimana satu guru mengajar banyak siswa

dan siswa diajar oleh banyak guru pula. Hubungan tersebut dapat digambarkan dengan tanda kotak untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

4. *Relational one to one 2 attribute* dalam satu *file*

Hubungan antara satu *attribute* dengan *attribute* yang lain dalam satu *file* yang sama mempunyai hubungan satu lawan satu. Missal *attribute* nomor pegawai yang unik dan *attribute* nomor NIM mahasiswa tersebut mempunyai hubungan satu lawan satu. Satu nomor pegawai hanya satu nomor NIM, tidak ada yang berganda.

5. *Relational many to one 2 attribute* dalam satu *file*

Hubungan antara satu *attribute* dengan *attribute* yang lain dalam satu *file* yang sama mempunyai hubungan satu lawan banyak. Misalnya pada satu alamat pegawai terdapat puluhan pegawai kantor tersebut maka hubungan antara *attribute* alamat pegawai dengan nomor pegawai adalah satu alamat menunjukkan banyak pegawai.

6. *Relational Many to Many 2 attribute dalam 1 file*

Hubungan antara satu *attribute* dengan *attribute* yang lain dalam satu *file* yang sama mempunyai hubungan banyak lawan banyak. Misalnya pada satu alamat pegawai terdapat puluhan pegawai kantor tersebut dan beberapa pegawai punya nama yang sama mempunyai alamat berbeda. Maka hubungan antara *attribute* alamat pegawai dengan nama pegawai adalah beberapa alamat menunjukkan nama pegawai yang sama dan sebaliknya.

2.10 SDLC (Sistems Development Life Cycle)

SDLC (*Sistems Development Life Cycle*) dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap :

1. Rencana (*Planning*),
2. Analisis (*Analysis*),
3. Desain (*Design*),
4. Implementasi (*Implementation*),
5. Uji Coba (*Testing*),
6. Dan Pengelolaan (*Maintenance*).

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Terdapat tiga jenis metode siklus hidup sistem yang paling banyak digunakan, yakni siklus hidup sistem tradisional (*traditional sistem life cycle*), siklus hidup menggunakan *prototyping (life cycle using prototyping)*, dan siklus hidup sistem orientasi objek (*object-oriented sistem life cycle*).

SDLC juga merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain. Konsep dalam SDLC mendasari model pengembangan perangkat lunak tersebut antara lain *waterfall, prototype, iterative, spiral, rapid application development (RAD)* dan lainnya.

2.11 Unified Modelling Language (UML)

2.11.1 Pengertian *Unified Modelling Language*

Winda Anggraeni (2016), UML adalah sebuah bahasa pemodelan yang telah menjadi standar dalam industri *software* untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Hal ini disebabkan Karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang

sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku.

Object – Oriented Analysis and Design (OOAD), yang dalam bahasa Indonesia-nya adalah Analisis dan Desain Berorientasi Objek merupakan pendekatan rekayasa perangkat lunak di mana model-model didalamnya merupakan interaksi objek.

Setiap objek mewakili beberapa entitas kepentingan dalam sistem yang dimodelkan, ditandai dalam kelasnya, wilayah-wilayahnya (elemen data), dan perilakunya. Berbagai model dapat dibuat untuk menunjukkan struktur statis, perilaku dinamis, dan penyebaran objek yang berkolaborasi pada saat sistem dijalankan.

Berdasarkan beberapa pendapat yang dikemukakan diatas penulis dapat menarik kesimpulan bahwa “*Unified Modelling Language (UML)*” adalah sebuah bahasa yang berdasarkan grafik dan gambar memvisualisasikan, menspesifikasika, membangun, dan pendokumentasian dari sebuah sistem pengembangan piranti lunak berbasis OO (*Object Oriented*).

Winda Anggraeni (2016), *Unified Modelling Language (UML)* biasa digunakan untuk :

1. Menggunakan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagram*.
4. Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component* dan *development diagrams*.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembangan sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk:

1. Mengkomunikasikan ide.
2. Melahirkan ide-ide baru dan peluang-peluang baru.
3. Menguji ide dan membuat prediksi.
4. Memahami struktur dan relasi-relasi.

2.11.2 Langkah – langkah Penggunaan Unified Modelling Language

Winda Anggraeni (2016), langkah-langkah penggunaan *Unified Modelling Language* (UML), sebagai berikut :

1. Buatlah daftar bisnis proses dari level tertinggi untuk mendefinisikan aktivitas dan proses yang mungkin muncul.
2. Petakan *use case* untuk setiap bisnis proses untuk mendefinisikan dengan tepat fungsional yang harus disediakan oleh sistem, kemudian perhalus *use case* diagram dan lengkapi dengan *requirement*, *constraints*, dan catatan-catatan lain.
3. Buatlah *deployment diagram* secara kasar untuk mendefinisikan arsitektur fisik sistem.
4. Definisikan *requirement* lain non fungsional, *security*, dan sebagainya yang juga harus disediakan sistem.
5. Berdasarkan *use case diagram*, mulailah membuat *activity diagram*.

6. Definisikan objek-objek level atas *package* atau dominan dan buatlah *sequence* dan atau *collaboration* untuk alur pekerjaan, jika sebuah *use case* memiliki kemungkinan alur normal dan *error*, buat lagi satu diagram untuk masing-masing alur.
7. Buatlah rancangan *user interface model* yang menyediakan antar muka bagi pengguna untuk menjalankan skenario *use case*.
8. Berdasarkan model-model yang sudah ada, buatlah *class diagram*. Setiap *package* atau *domain* dipecah menjadi hirarki *class* lengkap *attribute* dan metodenya. Akan lebih baik jika untuk menguji fungsionalitas *class* dan interaksi dengan *class* lain.
9. Setelah *class diagram* dibuat, kita dapat melihat kemungkinan pengelompokkan *class* menjadi komponen-komponen karena itu buatlah *component diagram* pada tahap ini. Juga, definisikan tes integrasi untuk setiap komponen meyakinkan ia bereaksi dengan baik.
10. Perhalus *deployment diagram* yang sudah dibuat. Detilkan kemampuan dan *requirement* piranti lunak, sistem operasi, jaringan dan sebagainya, petakan komponen ke dalam node.
11. Mulailah membangun sistem. Ada dua pendekatan yang tepat digunakan :

- a. Pendekatan *use case* dengan mendesain setiap *use case* kepada tim pengembang tertentu untuk mengembangkan unit kode yang lengkap dengan test.
- b. Pendekatan komponen yaitu mengassign setiap komponen kepada tim pengembang tertentu.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang dipergunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use Case Diagram*

Use Case Diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan mengenali interaksi mereka dalam suatu sistem.

a. Aktor

Aktor mewakili siapapun atau apa saja yang harus berinteraksi dengan sistem. Aktor biasa didefinisikan sebagai berikut :

- Aktor hanya memberikan informasi kepada sistem.
- Aktor hanya menerima informasi dari sistem.
- Aktor memberikan dan menerima informasi ke dan dari sistem.

b. *Use Case*

Use Case model adalah dialog antara aktor dengan sistem yang akan menggambarkan fungsi yang diberikan oleh sistem.

- *Use Case Relationship*

Use Case Relationship adalah suatu hubungan, baik itu antara aktor dan *use case* atau antara *use case*. Hubungan antara aktor dan *use case* disebut dengan *communicate Association*.






- *Association / Directed Association*






Asosiasi yaitu hubungan statis antar elemen. Umumnya menggambarkan elemen yang memiliki atribut berupa elemen lain, atau elemen yang harus mengetahui eksistensi elemen lain. Tanda panah menunjukkan arah *query* antar elemen.

- *Generalization / Pewarisan*

Pewarisan merupakan hubungan hirarkis antar elemen. Eleme dapat diturunkan dari elemen lain dan mewarisi semua atribut dan metode elemen asalnya dan menambahkan fungsionalitas baru, sehingga disebut anak dari elemen yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

Tabel 2. 1 Simbol *Use Case Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|--|
| 1 |  | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
| 2 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |
| 3 |  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
| 4 |  | <i>Include</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> . |
| 5 |  | <i>Extend</i> | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |

| NO | GAMBAR | NAMA | KETERANGAN |
|-----------|--|----------------------|---|
| 6 |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | <i>Sistem</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | <i>Use Case</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

2. *Activity Diagram*

Activity Diagram menggambarkan rangkaian aliran dari aktivitas. Digunakan untuk aktivitas lainnya seperti *use case* atau interaksi. *Activity Diagram* berupa *flow chart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut :

a. *Activity*

Notasi yang menggambarkan pelaksanaan dari beberapa proses dari aliran pekerjaan.

b. *Transition*

Notasi yang digunakan untuk memperlihatkan jalan aliran kontrol dari *activity* ke *activity*.







c. *Decision*

Notasi yang menandakan kontrol cabang aliran berdasarkan *decision point*.

d. *Sychromization Bar*

Aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan (pararel).

Tabel 2. 2 Simbol *Activity Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|----------------------------|---|
| 1 |  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | <i>Line Connector</i> | Memperlihatkan arah perpindahan aktifitas |
| 3 |  | <i>Initial Node</i> | Bagaimana objek dibentuk atau diawali. |
| 4 |  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan diakhiri |
| 5 |  | <i>Fork Node</i> | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |
| 6 |  | Decision | Pilihan untuk mengambil keputusan |

3. *Sequence Diagram*


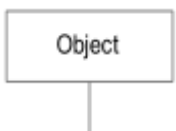
Sequence Diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence Diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *Sequence Diagram* adalah gambaran tahap demi tahap yang seharusnya


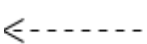
dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.





Dalam *sequence diagram* terdapat 2 model, yaitu :


- a. *Actor*, untuk menggambarkan pengguna sistem.
- b. *Lifeline*, untuk menggambarkan kelas dan objek.

Tabel 2. 3 Simbol *Sequence Diagram*

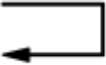
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|--|
| 1 |  | <i>Actor</i> | <i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. |
| 2 |  | <i>Object</i> | <i>Object</i> atau biasa juga disebut partisipan merupakan instance dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal.

Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma. |
| 3 |  | <i>Lifeline</i> | <i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah objek. |
| 4 |  | <i>Message Return</i> | Menggambarkan pesan/ hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi. |

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-------------------|---|
| 5 |  | <i>Activation</i> | <i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang melakukan suatu aksi |
| 6 |  | <i>Boundary</i> | Boundary terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori. |
| 7 |  | <i>Control</i> | <i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan. |
| 8 |  | <i>Entity</i> | <i>Entity</i> digunakan menangani informasi yang mungkin akan disimpan secara permanen.

<i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data. |
| 9 |  | <i>Message</i> | <i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> .




<i>Message</i> mengindikasikan Komunikasi antara <i>object-object</i> . |


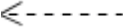
| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|---------------------|---|
| 10 |  | <i>Self-Message</i> | <i>Self-message</i> atau panggilan mandiri mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri. |

4. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu, antara lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class Diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Objek adalah nilai tertentu dari setiap *attribute* kelas *entity*. *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan orientasi objek. *Class* menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi).

Tabel 2. 4 Simbol *Class Diagram*

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|--------------------|--|
| 1 |  | <i>Composition</i> | <p>Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah relationship</p> <p>composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.</p> |
| 2 |  | <i>Aggregation</i> | <p><i>Aggregation</i> mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”.</p> <p>Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.</p> |
| 3 |  | <i>Class</i> | <p><i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi objek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian.</p> <p>Bagian atas adalah bagian nama dari <i>class</i>. Bagian tengah mendefinisikan <i>property/atribut class</i>. Bagian</p> |

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|---|
| | | | akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> . |
| 4 | <u>1..n Owned by 1</u> | <i>Association</i> | Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>). |
| 5 |  | <i>Generalization</i> | Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi objek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas " <i>parent</i> "-nya/induknya. |
| 6 |  | <i>Dependency</i> | Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik- |

| NO | GAMBAR | NAMA | KETERANGAN |
|----|--------|------|------------|
| | | | titik. |

2.2. *Black Box Testing*

Menurut Rosa dan Salahuddin (2015:275) “Blackbox testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

Menurut Rizky (2011:264) “Blackbox testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya”.

Menurut Nidhra dan Dondeti (2012:1), *black box testing* juga disebut *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi

BAB III

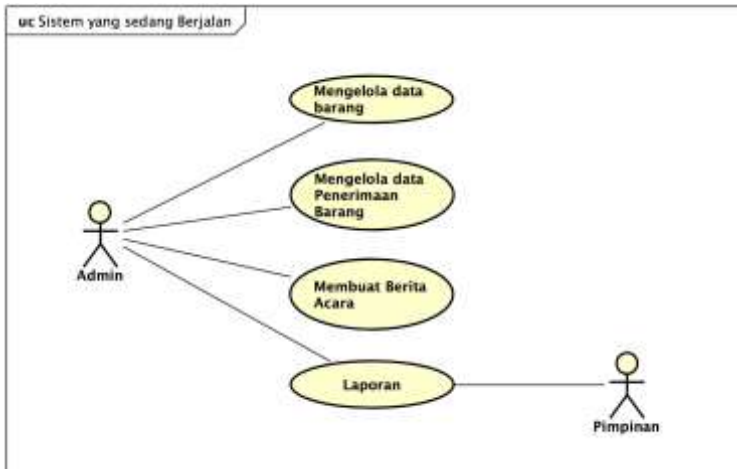
ANALISIS DAN PEMBAHASAN

3.1 Analisis Sistem

3.1.1 Analisis Sistem Berjalan

Pada saat proses Analisis sistem yang sedang berjalan pada proses penerimaan barang milik daerah pada Kecamatan Tangerang penulis mencatat bahwa proses penerimaan barang dan pembuatan berita acara penerimaan barang dari vendor ke pihak kecamatan masih menggunakan proses manual. Sistem penerimaan barang milik daerah yang saat ini sedang berjalan dirasa belum efektif, hal ini dikarenakan proses pengelolaan data masih menggunakan beberapa software yang terpisah (Ms. Excel, Ms. Word), dimana pihak admin atau staff kecamatan akan melakukan proses pengecekan kondisi barang dan penginputan data barang kedalam Microsoft Excel, lalu staff kecamatan membuat surat berita acara penerimaan barang dengan menggunakan Microsoft Word yang nantiya akan ditandatangani oleh pihak vendor dan pihak Kecamatan.

Berikut merupakan gambaran proses penerimaan barang dalam bentuk usecase diagram



Gambar 3. 1 Use Case Diagram Sistem yang sedang Berjalan

Tabel 3. 1 Penjelasan Usecase sistem Penerimaan barang milik daerah yang sedang berjalan

| Aktor | Sistem |
|-------------|---|
| Staff Admin | <ul style="list-style-type: none"> - Staff administrasi akan menginput data barang, dan data pemeliharaan barang dengan menggunakan Microsoft Excel. - Pada waktu yang ditentukan untuk melakukan pemeliharaan barang, staff administrasi akan menginfokan jadwal pemeliharaan barang kepada Petugas pemeliharaan barang. |
| Pimpinan | <ul style="list-style-type: none"> - Menerima laporan Pemeliharaan Barang |

3.1.2 Analisis Kebutuhan Sistem

Setelah melakukan tahap analisis sistem yang sedang berjalan, penulis melakukan proses analisis kebutuhan sistem, Adapun proses analisis kebutuhan pada proses penerimaan barang milik daerah di kecamatan Tangerang ini dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional kebutuhan yang berisi tentang proses - proses apa saja yang nantinya dilakukan oleh sistem serta informasi apa saja yang harus ada dan dihasilkan oleh sistem. Sedangkan kebutuhan non dilakukan untuk mengetahui spesifikasi kebutuhan diantaranya analisis perangkat keras/hardware, analisis perangkat lunak/software, analisis pengguna/user.

1. Kebutuhan fungsional

Dari hasil wawancara dengan pihak kecamatan dan hasil analisis dari latar belakang masalah yang ada di lapangan maka penulis menyimpulkan kebutuhan fungsional sistem sebagai berikut.

- a. Dibutuhkan sebuah sistem yang dapat mengelola data penerimaan barang milik daerah mulai dari menginput, merubah, menghapus dan mencari data.
- b. Dibutuhkan sistem yang dapat membuat dan mencetak berita acara secara langsung dari input yang tersedia.

- c. Dibutuhkan sistem berbasis web yang dapat menyimpan data dengan baik dan dapat diakses dan digunakan dilingkungan kecamatan Tangerang.

Berdasarkan kebutuhan sistem diatas penulis menyakini bahwa dengan mengimplementasikan sistem informasi penerimaan barang milik daerah berbasis web ini akan memberikan solusi dalam mempermudah dalam proses pengelolaan barang milik daerah di kecamatan Tangerang.

2. Kebutuhan Non Fungsional

a. Kebutuhan Software

Software yang dibutuhkan untuk dapat menjalankan sistem ini ialah, sebagai berikut:

- 1) Sistem Operasi minimal Windows XP atau versi keatas.
- 2) Web Browser, misalkan: Mozilla Firefox & Google Chrome terbaru.

b. Kebutuhan Hardware

Spesifikasi minimal hardware yang dibutuhkan untuk dapat menjalankan sistem tersebut, sebagai berikut:

- 1) Processor : Intel (R) Pentium IV 2,8 Ghz.
- 2) Harddisk : 80 Gb Seagate Rpm 7200.
- 3) Memory : DDR 1 GB Visipro PC 5300.
- 4) Optical Drive : DVD RW 52 X LG.

- 5) Monitor : Samsung 15”.
- 6) Motherboard : Intel Corporation D 945 GNT.
- 7) VGA : ATI Radeon HD 2600 512 MB.
- 8) Keyboard : Logitec.
- 9) Mouse : Digital.

3.2 Rancangan Program

Dalam merancang sistem, penulis menggunakan bahasa pemodelan UML (*Unified Modelling Language*). Hal ini karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku.

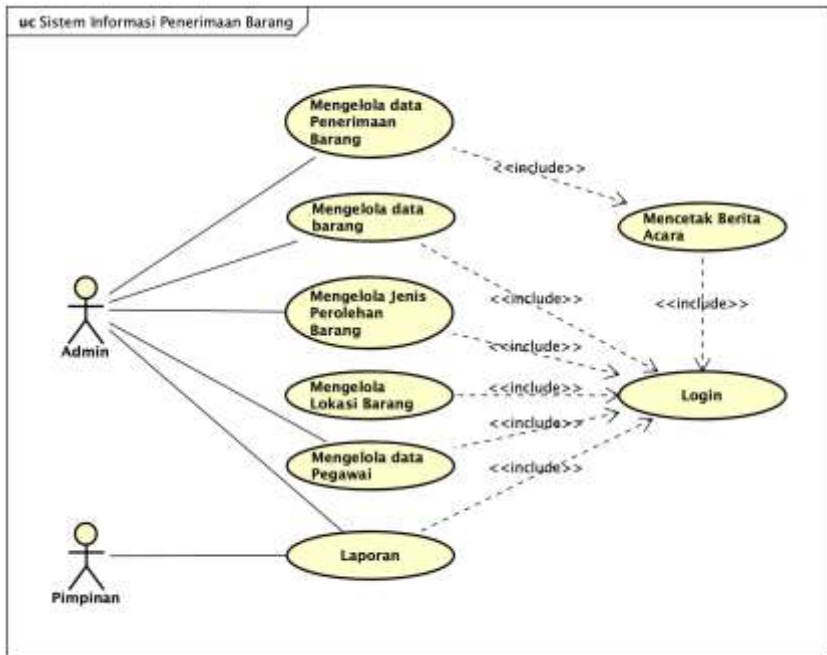
UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembangan sistem membuat *blue print* atas visinya dalam bentuk yang baku.

UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk:

1. Mengkomunikasikan ide.
2. Melahirkan ide-ide baru dan peluang-peluang baru.
3. Menguji ide dan membuat prediksi.
4. Memahami struktur dan relasi-relasi.

Berikut adalah rancangan program atau sistem yang akan penulis bangun:

3.2.1 Use Case Diagram



Gambar 3. 2 Use Case Diagram Sistem Penerimaan Barang Milik Daerah

**Tabel 3. 2 Penjelasan Aktor *Use Case* Sistem Informasi
Penerimaan Barang Milik Daerah**

| No | Aktor | Penjelasan |
|----|----------|--|
| 1. | Admin | Admin merupakan staff keamanan yang bertugas melaksanakan pengelolaan data penerimaan barang, dari mulai data barang, data jenis penerimaan barang, lokasi penyimpanan dan pembuatan berita acara penerimaan barang. |
| 2. | Pimpinan | Pimpinan dapat melihat dan mengelola laporan |

Berikut merupakan penjelasan use case sistem penerimaan barang milik daerah

1. Use case login

Tabel 3. 3 Deskripsi Use Case Login

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | <i>Login</i> | Tipe <i>Use Case</i> |
| Use Case Id: | 001 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan verifikasi masuk kedalam sistem informasi penerimaan barang milik daerah. | |

| | | |
|--------------------|---|---|
| Pre-Condition: | Admin membuka halaman sistem informasi penerimaan barang barang milik daerah | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | | Step1: Menampilkan <i>form login</i> |
| | Step2: Admin meng- <i>input</i> -kan <i>username & password</i> | Step3: Jika <i>Username & password</i> benar, maka akan menampilkan halaman utama sistem. |
| Alternate Courses: | Alt. Step1: Jika <i>username & password</i> tidak benar sistem akan menampilkan pesan kesalahan. | |
| Conclusion: | <i>Use Case</i> ini selesai jika <i>actor</i> berhasil <i>login</i> dan <i>Sistem</i> menampilkan halaman utama sistem. | |
| Post-Condition | - | |

2. Use case mengelola data barang

a. Tambah Data barang

Tabel 3. 4 Penjelasan *Use Case* Tambah Data Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|------------------|--------------------|----------------------|
| Use Case Name: | Tambah Data Barang | Tipe <i>Use Case</i> |
| Use Case Id: | 002 | Persyaratan Bisnis |
| Priority: | <i>High</i> | |
| Primary Business | Admin | |



| | | |
|--------------------|---|---|
| Actors | | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data user (Tambah Data). | |
| Pre-Condition: | Admin memilih form data Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin memilih tambah data Barang | Step2: Menampilkan <i>form</i> tambah data barang |
| | Step3: Admin meng- <i>input</i> -kan data baru | Step4: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak lengkap maka admin meng- <i>input</i> -kan ulang data | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menambahkan data | |
| Post-Condition | - | |

b. Ubah Data

Tabel 3. 5 Penjelasan Use Case Ubah Data Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|------------------|------------------|-------------------------------------|
| Use Case Name: | Ubah data Barang | Tipe Use Case |
| Use Case Id: | 003 | Persyaratan Bisnis |
| Priority: | <i>High</i> | |
| Primary Business | Admin | |
| | | <input checked="" type="checkbox"/> |

| | | |
|--------------------|--|---|
| Actors | | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Barang (Ubah Data). | |
| Pre-Condition: | Admin memilih form data barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Barang yang ingin dirubah. | Step2: Menampilkan data ke yang dicari. |
| | Step3: Admin merubah data Barang | Step4: Sistem memvalidasi data. |
| | Step5: Admin memilih simpan data. | Step6: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak valid, admin meng-inputkan ulang data Barang | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil merubah data Barang | |
| Post-Condition | - | |

c. Hapus Data Barang

Tabel 3. 6 Penjelasan *Use Case* Hapus Data Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|----------------|-------------------------------|----------------------|
| Use Case Name: | Olah Data Barang (Hapus Data) | Tipe <i>Use Case</i> |
| Use Case Id: | 004 | Persyaratan Bisnis |

| | | |
|-------------------------|--|---|
| Priority: | <i>High</i> | <input checked="" type="checkbox"/> |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Barang (Hapus Data). | |
| Pre-Condition: | Admin memilih form data Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Barang yang ingin dihapus. | Step2: Menampilkan data Barang yang dicari. |
| | Step3: Admin memilih data Barang yang ingin dihapus. | Step4: Sistem menampilkan pesan hapus atau batal. |
| | Step5: Admin memilih hapus. | Step6: Sistem menghapus data Barang. |
| Alternate Courses: | Alt. Step1: Jika admin batal untuk menghapus data, maka admin memilih batal hapus. | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menghapus data Barang. | |
| Post-Condition | - | |

3. Use case mengelola data pegawai

a. Tambah Data Pegawai

Tabel 3. 7 Penjelasan Use Case Tambah Data Pegawai

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | Tambah Data Pegawai | Tipe Use Case |
| Use Case Id: | 005 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | High | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data user (Tambah Data). | |
| Pre-Condition: | Admin memilih form data Pegawai | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin memilih tambah data Pegawai | Step2: Menampilkan form tambah data Pegawai |
| | Step3: Admin meng-inputkan data baru | Step4: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak lengkap maka admin menginputkan ulang data | |
| Conclusion: | Use Case ini selesai jika admin berhasil menambahkan data | |
| Post-Condition | - | |

b. Ubah Data Pegawai

Tabel 3. 8 Penjelasan Use Case Ubah Data Pegawai

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|---|---|
| Use Case Name: | Ubah data Pegawai | Tipe Use Case |
| Use Case Id: | 006 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | High | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Pegawai (Ubah Data). | |
| Pre-Condition: | Admin memilih form data Pegawai | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Pegawai yang ingin dirubah. | Step2: Menampilkan data ke yang dicari. |
| | Step3: Admin merubah data Pegawai | Step4: Sistem memvalidasi data. |
| | Step5: Admin memilih simpan data. | Step6: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak valid, admin meng-inputkan ulang data Pegawai | |
| Conclusion: | Use Case ini selesai jika admin berhasil merubah data | |
| Post-Condition | - | |

c. Hapus Data Pegawai

Tabel 3. 9 Penjelasan *Use Case* Hapus Data Pegawai

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | Olah Data Pegawai (Hapus Data) | Tipe <i>Use Case</i> |
| Use Case Id: | 007 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Pegawai (Hapus Data). | |
| Pre-Condition: | Admin memilih form data Pegawai | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Pegawai yang ingin dihapus. | Step2: Menampilkan data Pegawai yang dicari. |
| | Step3: Admin memilih data Pegawai yang ingin dihapus. | Step4: Sistem menampilkan pesan hapus atau batal. |
| | Step5: Admin memilih hapus. | Step6: Sistem menghapus data Pegawai. |
| Alternate Courses: | Alt. Step1: Jika admin batal untuk menghapus data, maka admin memilih batal hapus. | |

| | |
|----------------|---|
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menghapus data. |
| Post-Condition | - |

4. Use case Mengeola data Lokasi Barang

a. Tambah Data Lokasi Barang

Tabel 3. 10 Penjelasan *Use Case* Tambah Data Lokasi Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | Tambah Data Lokasi Barang | Tipe <i>Use Case</i> |
| Use Case Id: | 008 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data user (Tambah Data). | |
| Pre-Condition: | Admin memilih form data Lokasi Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin memilih tambah data Lokasi Barang | Step2: Menampilkan form tambah data Lokasi Barang |
| | Step3: Admin meng-input- | Step4: Sistem |

| | | |
|--------------------|---|-----------------|
| | kan data baru | menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak lengkap maka admin menginput-kan ulang data | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menambahkan data | |
| Post-Condition | - | |

b. Ubah Data Lokasi

Tabel 3. 11 Penjelasan *Use Case* Ubah Data Lokasi Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|---|---|
| Use Case Name: | Ubah data Lokasi Barang | Type <i>Use Case</i> |
| Use Case Id: | 009 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Lokasi Barang (Ubah Data). | |
| Pre-Condition: | Admin memilih form data Lokasi Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Lokasi Barang yang ingin dirubah. | Step2: Menampilkan data ke yang dicari. |

| | | |
|--------------------|---|---------------------------------|
| | Step3: Admin merubah data Lokasi Barang | Step4: Sistem memvalidasi data. |
| | Step5: Admin memilih simpan data. | Step6: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak valid, admin meng-inputkan ulang data Lokasi Barang | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil merubah data Lokasi Barang | |
| Post-Condition | - | |

c. Hapus Data Lokasi Barang

Tabel 3. 12 Penjelasan *Use Case* Hapus Data Lokasi Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | Olah Data Lokasi Barang (Hapus Data) | Tipe <i>Use Case</i> |
| Use Case Id: | 010 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Lokasi Barang (Hapus Data). | |
| Pre-Condition: | Admin memilih form data Lokasi Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |

| | | |
|--------------------|--|--|
| Of Events: | Step1: Admin mencari data Lokasi Barang yang ingin dihapus. | Step2: Menampilkan data Lokasi Barang yang dicari. |
| | Step3: Admin memilih data Lokasi Barang yang ingin dihapus. | Step4: Sistem menampilkan pesan hapus atau batal. |
| | Step5: Admin memilih hapus. | Step6: Sistem menghapus data Lokasi Barang. |
| Alternate Courses: | Alt. Step1: Jika admin batal untuk menghapus data, maka admin memilih batal hapus. | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menghapus data Lokasi Barang. | |
| Post-Condition | - | |

5. Use case Mengeola Jenis Perolehan Barang

a. Tambah Data Jenis Perolehan Barang

Tabel 3. 13 Penjelasan *Use Case* Tambah Data Jenis Perolehan Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|------------------------------------|---|
| Use Case Name: | Tambah Data Jenis Perolehan Barang | Tipe <i>Use Case</i> |
| Use Case Id: | 011 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |

| | | |
|--------------------|--|--|
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Jenis Perolehan Barang (Tambah Data). | |
| Pre-Condition: | Admin memilih form data Jenis Perolehan Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin memilih tambah data Jenis Perolehan Barang | Step2: Menampilkan form tambah data Jenis Perolehan Barang |
| | Step3: Admin meng-inputkan data baru | Step4: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak lengkap maka admin menginputkan ulang data | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menambahkan data | |
| Post-Condition | - | |

b. Ubah Data Jenis Perolehan Barang

Tabel 3. 14 Penjelasan Use Case Ubah Data Jenis Perolehan Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|------------------|----------------------------------|--------------------|
| Use Case Name: | Ubah data Jenis Perolehan Barang | Type Use Case |
| Use Case Id: | 012 | Persyaratan Bisnis |
| Priority: | High | |
| Primary Business | Admin | |



| | | |
|--------------------|--|---|
| Actors | | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Jenis Perolehan Barang (Ubah Data). | |
| Pre-Condition: | Admin memilih form data Jenis Perolehan Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Jenis Perolehan Barang yang ingin dirubah. | Step2: Menampilkan data ke yang dicari. |
| | Step3: Admin merubah data Jenis Perolehan Barang | Step4: Sistem memvalidasi data. |
| | Step5: Admin memilih simpan data. | Step6: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak valid, admin meng-inputkan ulang data | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil merubah data | |
| Post-Condition | - | |

c. Hapus Data Jenis Perolehan Barang

Tabel 3. 15 Penjelasan *Use Case* Hapus Data Jenis Perolehan Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|---|---|
| Use Case Name: | Olah Data Jenis Perolehan Barang (Hapus Data) | Type <i>Use Case</i> |
| Use Case Id: | 013 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Jenis Perolehan Barang (Hapus Data). | |
| Pre-Condition: | Admin memilih form data Jenis Perolehan Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Jenis Perolehan Barang yang ingin dihapus. | Step2: Menampilkan data Jenis Perolehan Barang yang dicari. |
| | Step3: Admin memilih data Jenis Perolehan Barang yang ingin dihapus. | Step4: Sistem menampilkan pesan hapus atau batal. |
| | Step5: Admin memilih hapus. | Step6: Sistem menghapus data. |
| Alternate Courses: | Alt. Step1: Jika admin batal untuk menghapus data, maka admin memilih batal hapus. | |

| | |
|----------------|---|
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menghapus data. |
| Post-Condition | - |

6. Use case Mengeola data Penerimaan Barang

a. Tambah Data Penerimaan Barang

Tabel 3. 16 Penjelasan *Use Case* Tambah Data Penerimaan Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | Tambah Data Penerimaan Barang | Tipe <i>Use Case</i> |
| Use Case Id: | 014 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Penerimaan Barang(Tambah Data). | |
| Pre-Condition: | Admin memilih form data Jenis Perolehan Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin memilih tambah data Jenis Perolehan Barang | Step2: Menampilkan <i>form</i> tambah data Jenis Perolehan Barang |
| | Step3: Admin meng- <i>input</i> - | Step4: Sistem |

| | | |
|--------------------|--|-----------------|
| | kan data baru | menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak lengkap maka admin menginputkan ulang data | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menambahkan data | |
| Post-Condition | - | |

b. Ubah Data Penerimaan Barang

Tabel 3. 17 Penjelasan *Use Case* Ubah Data Penerimaan Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | Ubah data Penerimaan Barang | Type <i>Use Case</i> |
| Use Case Id: | 015 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Penerimaan Barang(Ubah Data). | |
| Pre-Condition: | Admin memilih form data Jenis Perolehan Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Penerimaan Barangyang | Step2: Menampilkan data ke yang dicari. |

| | | |
|--------------------|---|---------------------------------|
| | ingin dirubah. | |
| | Step3: Admin merubah data Jenis Perolehan Barang | Step4: Sistem memvalidasi data. |
| | Step5: Admin memilih simpan data. | Step6: Sistem menyimpan data. |
| Alternate Courses: | Alt. Step1: Jika data tidak valid, admin meng-inputkan ulang data | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil merubah data | |
| Post-Condition | - | |

c. Hapus Data Jenis Perolehan Barang

Tabel 3. 18 Penjelasan *Use Case* Hapus Data Penerimaan Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|---|---|
| Use Case Name: | Olah Data Penerimaan Barang(Hapus Data) | Tipe <i>Use Case</i> |
| Use Case Id: | 016 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk melakukan pengolahan data Penerimaan Barang(Hapus Data). | |

| | | |
|--------------------|--|--|
| Pre-Condition: | Admin memilih form data Jenis Perolehan Barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Penerimaan Barang yang ingin dihapus. | Step2: Menampilkan data Penerimaan Barang yang dicari. |
| | Step3: Admin memilih data Penerimaan Barang yang ingin dihapus. | Step4: Sistem menampilkan pesan hapus atau batal. |
| | Step5: Admin memilih hapus. | Step6: Sistem menghapus data. |
| Alternate Courses: | Alt. Step1: Jika admin batal untuk menghapus data, maka admin memilih batal hapus. | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menghapus data. | |
| Post-Condition | - | |

7. Print Berita Acara Penerimaan Barang

Tabel 3. 19 Penjelasan *Use Case* Print Berita Acara Penerimaan Barang

| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|----------------|--------------------------------------|----------------------|
| Use Case Name: | Print Berita Acara Penerimaan Barang | Tipe <i>Use Case</i> |
| Use Case Id: | 017 | Persyaratan Bisnis |
| Priority: | <i>High</i> | |

| | | |
|-------------------------|---|--|
| Primary Business Actors | Admin | <input checked="" type="checkbox"/> |
| Description: | Use Case ini digunakan untuk Print Berita Acara Penerimaan Barang | |
| Pre-Condition: | Admin memilih form data penerimaan barang | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin mencari data Penerimaan Barang yang ingin di print | Step2: Menampilkan data Penerimaan Barang yang di print |
| | Step3: Admin memilih tombol print berita acara penerimaan barang | Step4: Sistem melakukan Print Berita Acara Penerimaan Barang |
| Alternate Courses: | - | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil Print Berita Acara Penerimaan Barang. | |
| Post-Condition | - | |

8. Use case Laporan

Tabel 3. 20 Penjelasan Use Case Laporan

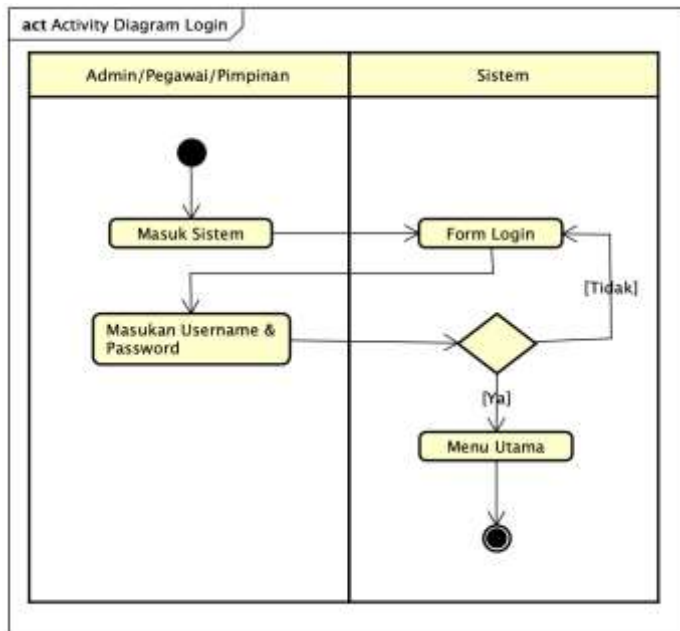
| | |
|--------------------------|-------------------|
| Tanggal: 26 Agustus 2022 | Author: M. Furqon |
| Versi: 1.0 | |

| | | |
|-------------------------|--|---|
| Use Case Name: | Use case Laporang | Tipe <i>Use Case</i> |
| Use Case Id: | 018 | Persyaratan Bisnis

<input checked="" type="checkbox"/> |
| Priority: | <i>High</i> | |
| Primary Business Actors | Admin | |
| Description: | Use Case ini digunakan untuk meliht laporan | |
| Pre-Condition: | Admin memilih menu laporan | |
| Typical Courses | <i>Actor Action</i> | <i>System Response</i> |
| Of Events: | Step1: Admin memilih menu laporan | Step2: Menampilkan halaman laporan |
| | Step5: Admin memilih hapus. | Step6: Sistem menghapus data. |
| Alternate Courses: | - | |
| Conclusion: | <i>Use Case</i> ini selesai jika admin berhasil menampilkan laporan. | |
| Post-Condition | - | |

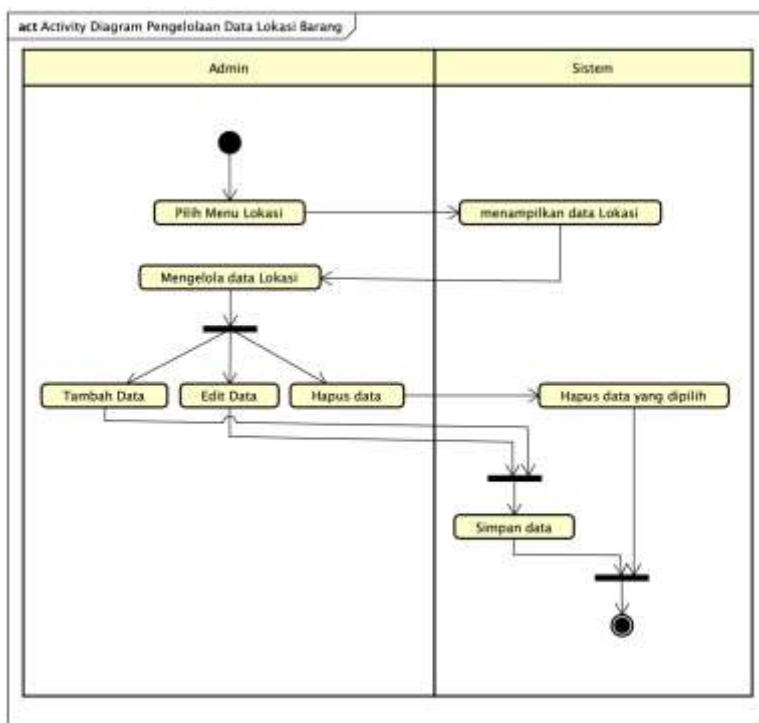
3.2.2 Activity Diagram

1. Activity diagram Login User



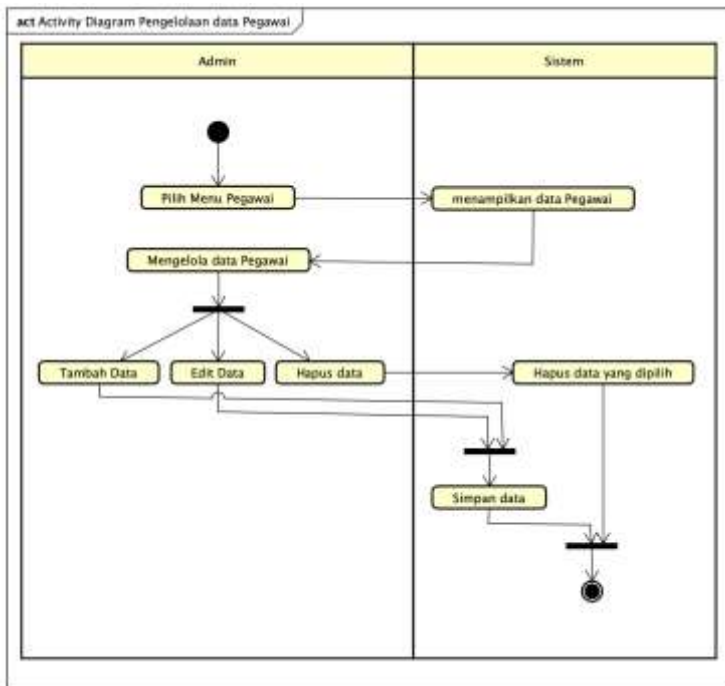
Gambar 3. 3 Activity Diagram Login User

2. Activity Diagram Pengelolaan data Lokasi



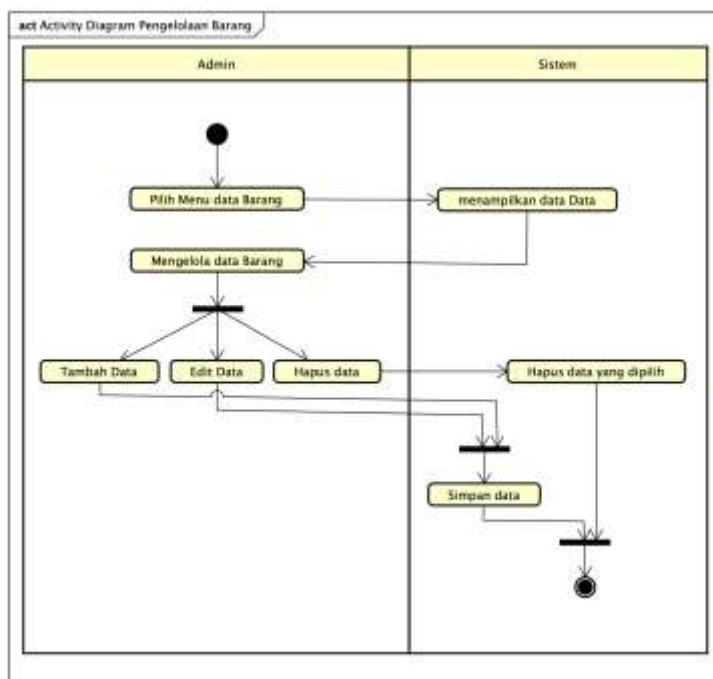
Gambar 3. 4 Activity Diagram Pengelolaan data Lokasi

3. Activity Diagram Pengelolaan data Pegawai



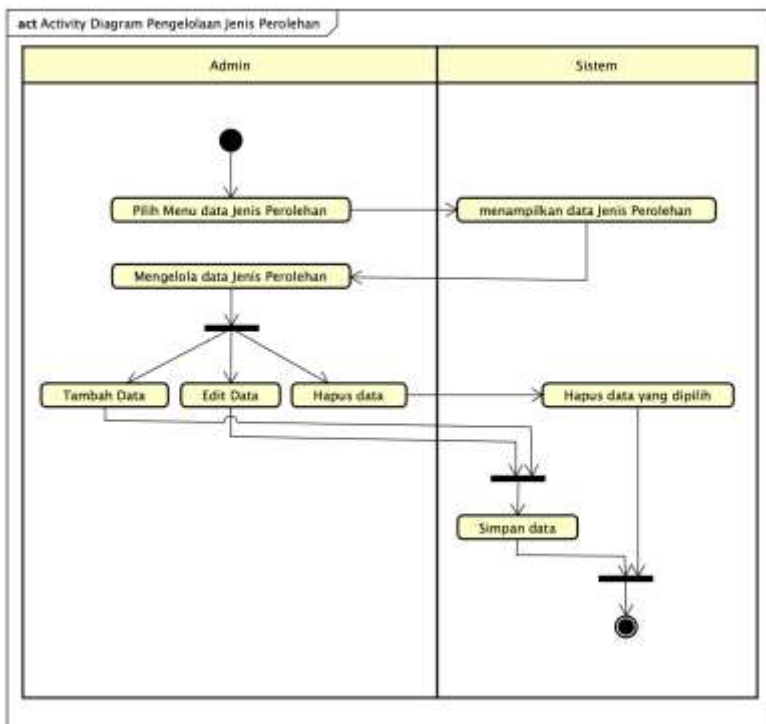
Gambar 3. 5 Activity Diagram Pengelolaan data Pegawai

4. Activity Diagram Pengelolaan data Barang



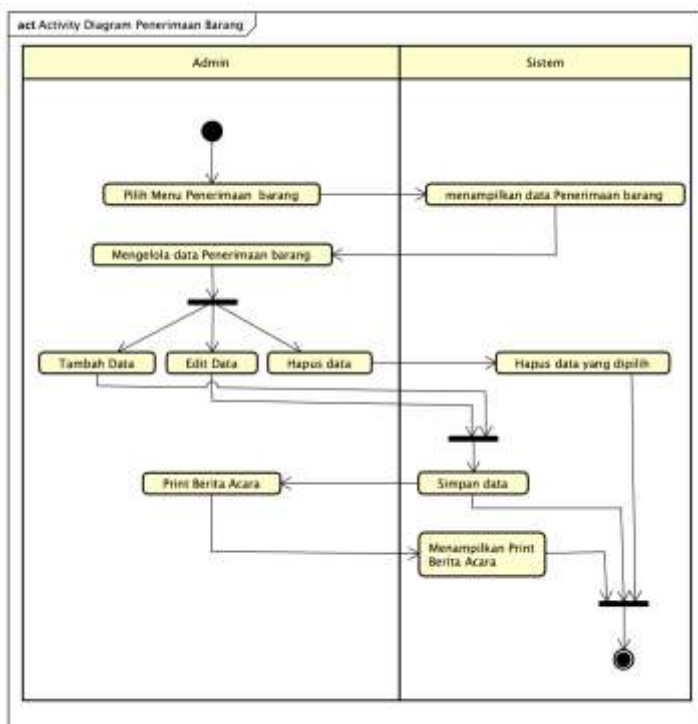
Gambar 3. 6 Activity Diagram Pengelolaan data Barang

5. Activity Diagram Pengelolaan data Jenis Perolehan



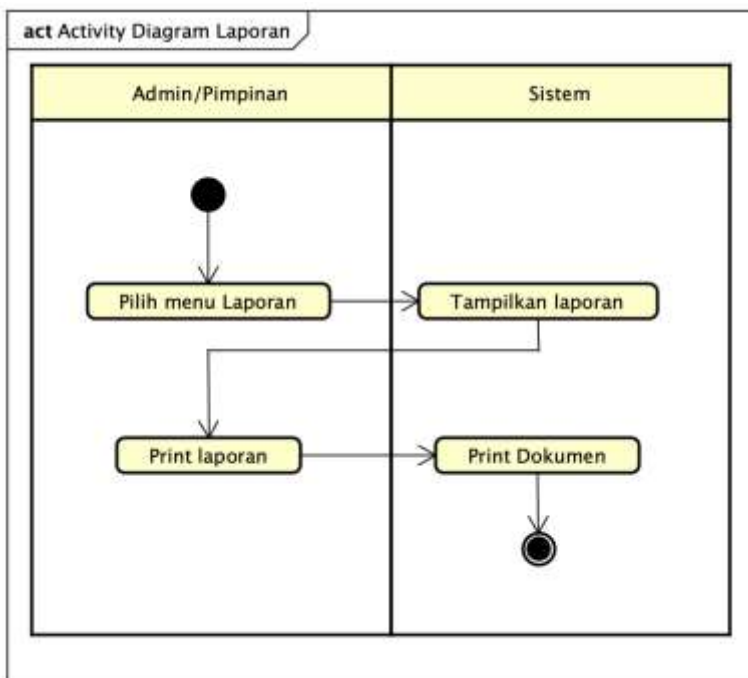
Gambar 3. 7 Activity Diagram Pengelolaan data Jenis Perolehan

6. Activity Diagram Penerimaan Barang



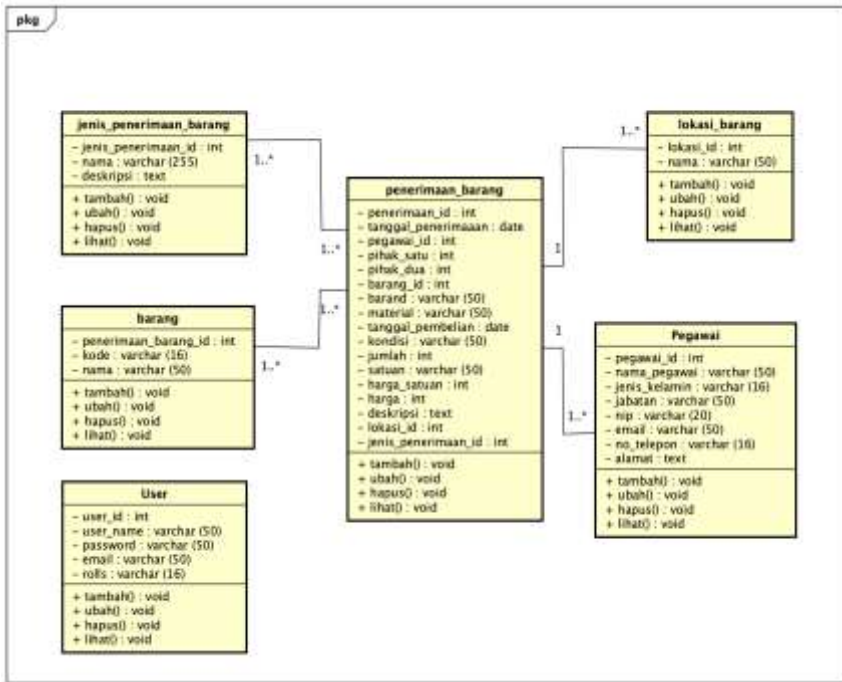
Gambar 3. 8 Activity Diagram Penerimaan Barang

7. Activity Diagram Laporan



Gambar 3. 9 Activity Diagram Laporan

8. Class Diagram sistem informasi penerimaan barang milik daerah



Gambar 3. 10 Class diagram sistem informasi penerimaan barang milik daerah

3.2.3 Perancangan Antarmuka

Proses perancangan antar muka atau user interface (UI) bertujuan untuk membuat tampilan perangkat lunak atau aplikasi yang menarik, dan mudah digunakan. Serta memberikan gambaran visual dari aplikasi yang akan dibuat, dan memudahkan programmer dalam pengembangan aplikasi.

Dalam proses perancangan antar muka atau user interface (UI) penulis menggunakan tools aplikasi figma, figma merupakan aplikasi populer yang digunakan dalam proses merancang User interface dan User Experience, Adapun hasil perancangan dapat di lihat pada link berikut:

<https://www.figma.com/proto/KEvMCqehwzPcw1dzBdX9nX/Untitled?page-id=0%3A1&node-id=2%3A2&viewport=153%2C151%2C0.07&scaling=min-zoom&starting-point-node-id=2%3A2>

dan berikut merupakan hasil dari rancangan antarmuka atau user interface yang di buat :

1. Form Login

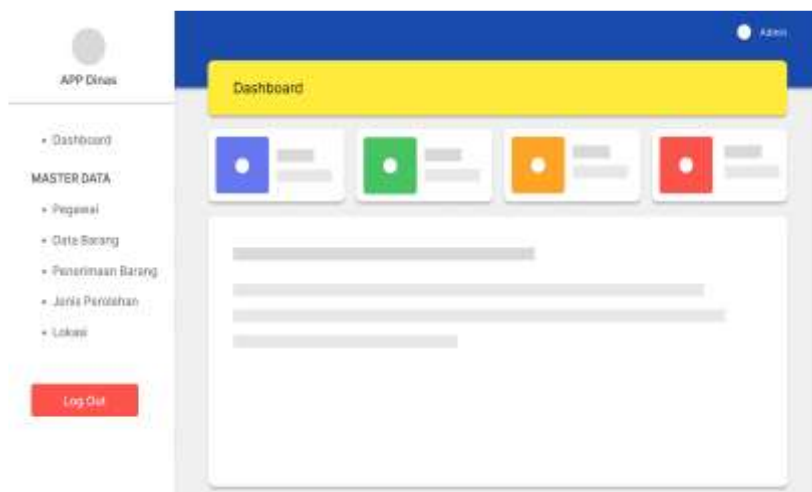
Dalam form login ini user harus masukan email dan password untuk dapat mengakses sistem



Gambar 3. 11 Rancangan antarmuka Form Login

2. Dashboard

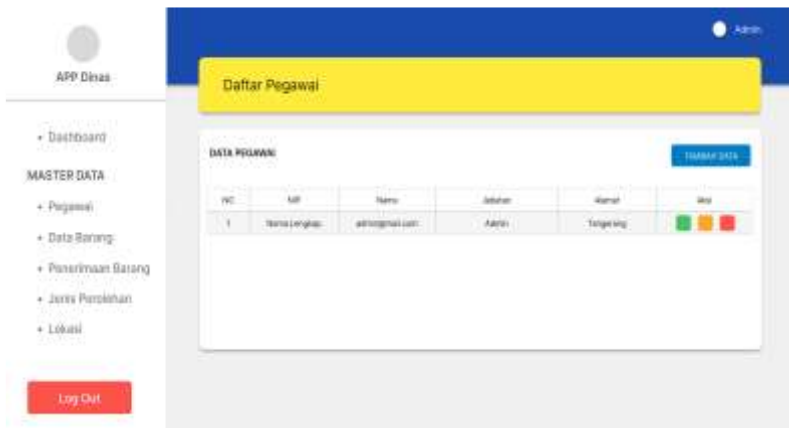
Halaman dashboard merupakan halaman utama yang nantinya akan berisi informasi utama dari sistem informasi ini.



Gambar 3. 12 Rancangan antarmuka halaman Dashboard

3. Halaman Data Pegawai

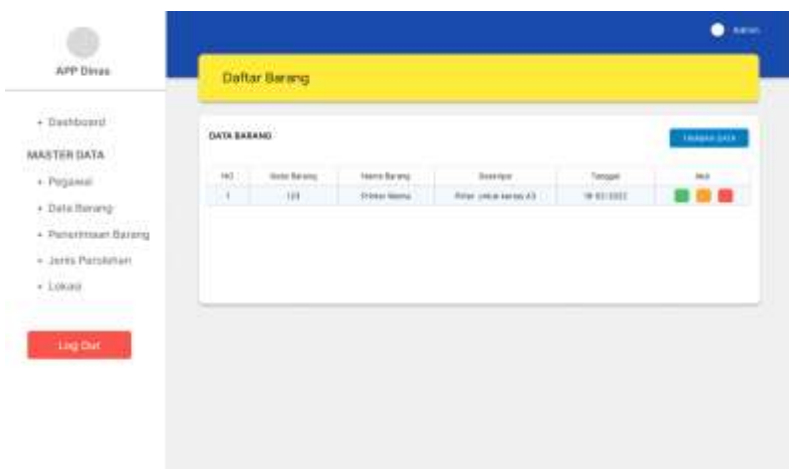
Halaman ini merupakan halaman untuk mengelola data pegawai, dalam hal ini user dapat menambah, mengubah, menghapus dan mencari data pegawai.



Gambar 3. 13 Rancangan antarmuka halaman Data Pegawai

4. Halaman Data Barang

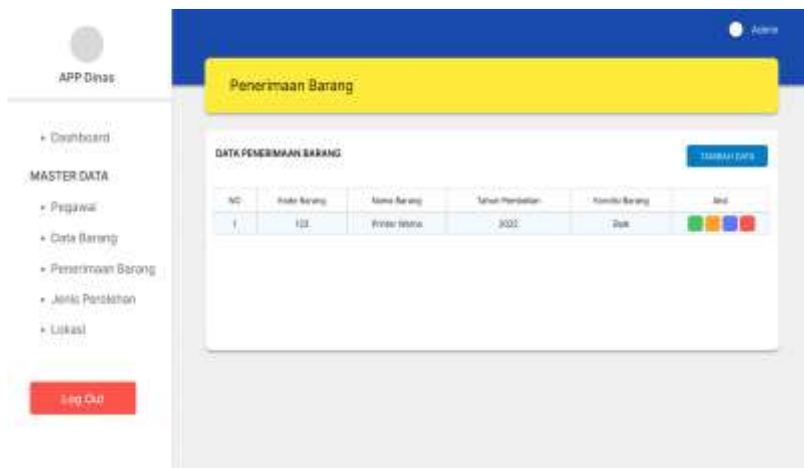
Halaman ini merupakan halaman untuk mengelola data Barang, user dapat menambah, mengubah, menghapus dan mencari data Barang.



Gambar 3. 14 Rancangan antarmuka halaman Data Barang

5. Halaman Penerimaan Barang

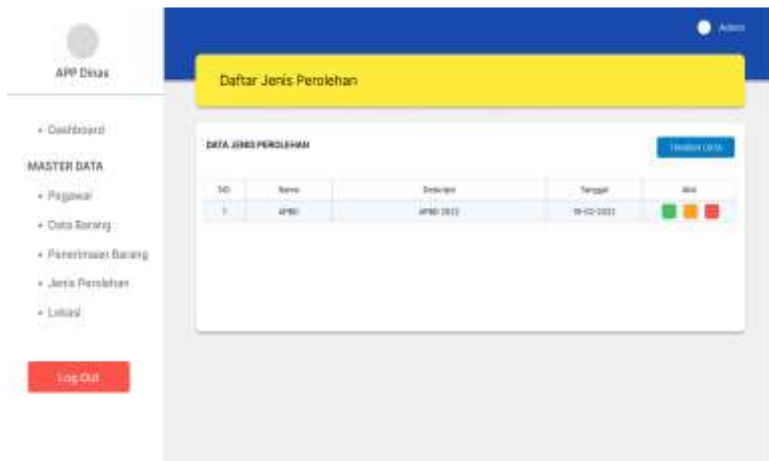
Halaman ini merupakan halaman untuk mengelola data penerimaan Barang, user dapat menambah, mengubah, menghapus dan mencari data Barang.



Gambar 3. 15 Rancangan antarmuka halaman Penerimaan Barang

6. Halaman data Jenis Perolehan

Halaman ini merupakan halaman untuk mengelola data Jenis Perolehan (Misal APBN/APBD), dalam halamin ini user dapat menambah, mengubah, menghapus dan mencari data Jenis Perolehan.



Gambar 3. 16 Rancangan antarmuka halaman Jenis Perolehan

7. Halaman Data Lokasi

Halaman ini merupakan halaman untuk mengelola data Lokasi, user dapat menambah, mengubah, menghapus dan mencari data Lokasi.

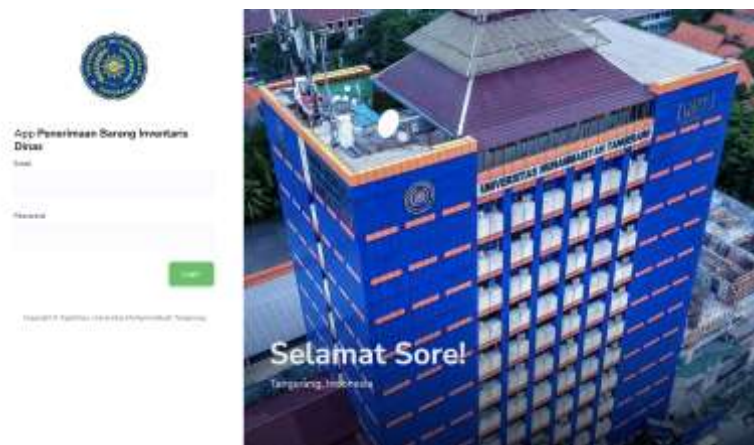


Gambar 3. 17 Rancangan antarmuka halaman Data Lokasi

3.3 Implementasi Antar Muka dan Pemrograman

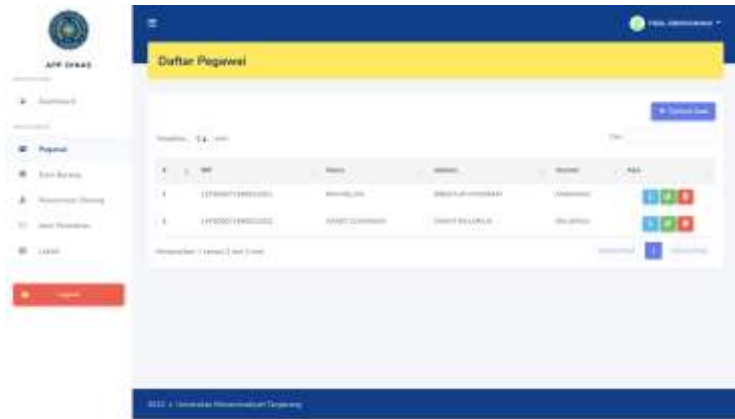
Dalam tahap implementasi dan pengembangan sistem informasi penerimaan barang milik daerah berbasis web ini penulis melakukan proses development sistem dengan menggunakan Bahasa pemrograman berbasis PHP dan MySQL. Berikut merupakan hasil user interface dari sistem informasi penerimaan barang milik daerah.

1. Form Login



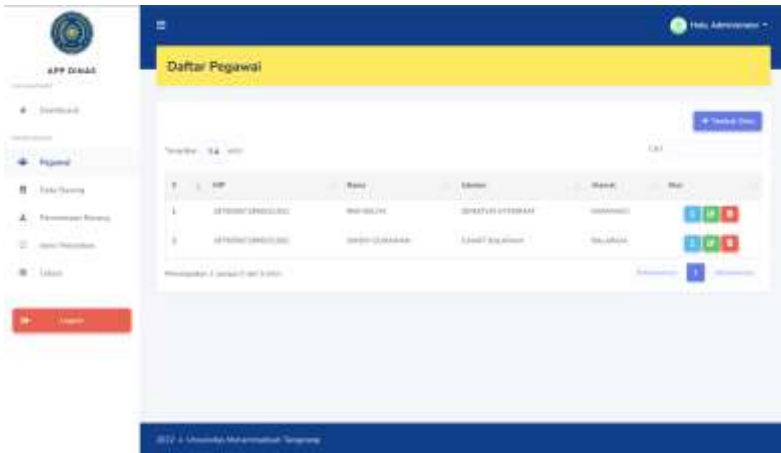
Gambar 3. 18 Form Login

2. Dashboard



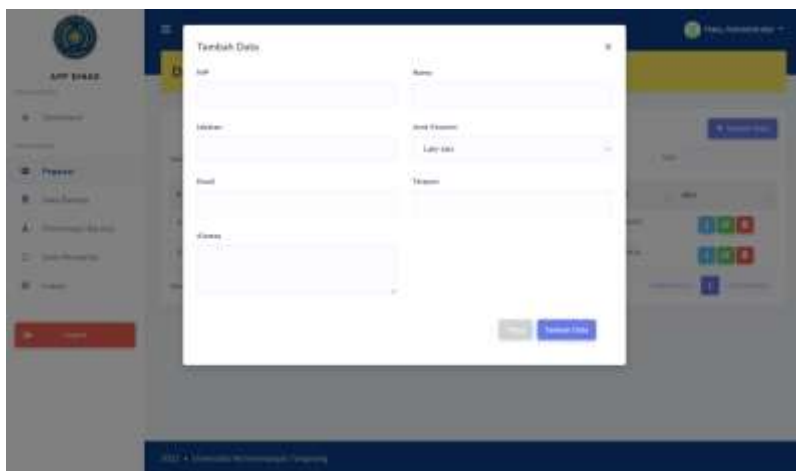
Gambar 3. 19 Dashboard

3. Halaman Pegawai



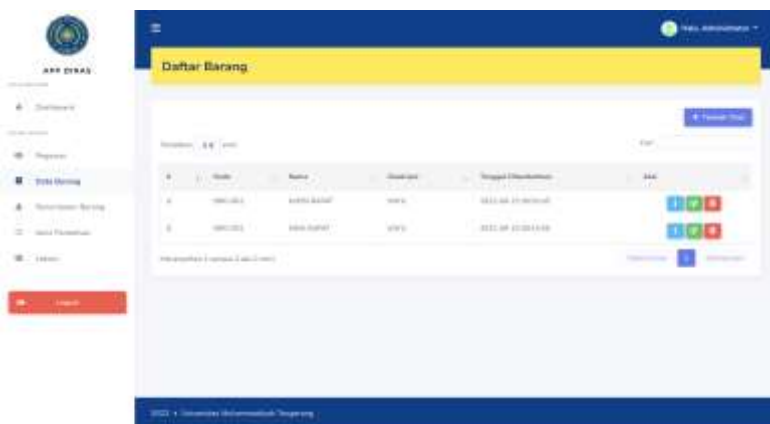
Gambar 3. 20 Halaman Pegawai

4. Halaman Tambah Data Pegawai



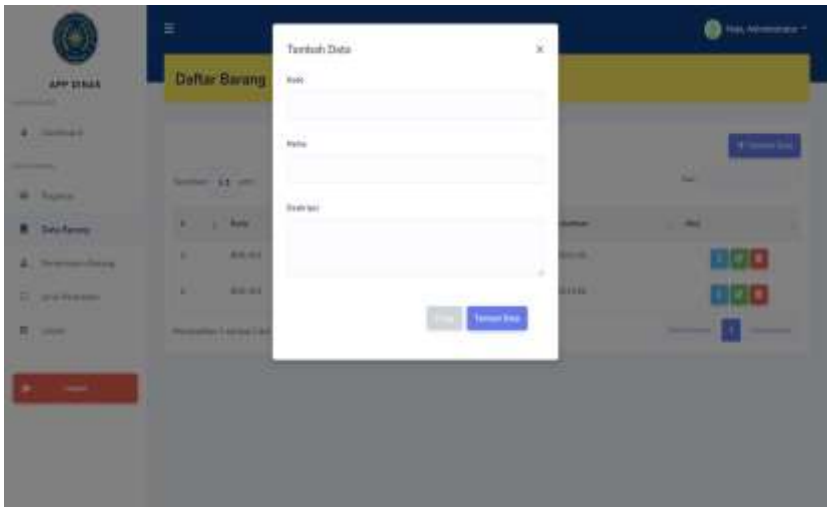
Gambar 3. 21 Halaman Tambah Data Pegawai

5. Halaman Data Barang



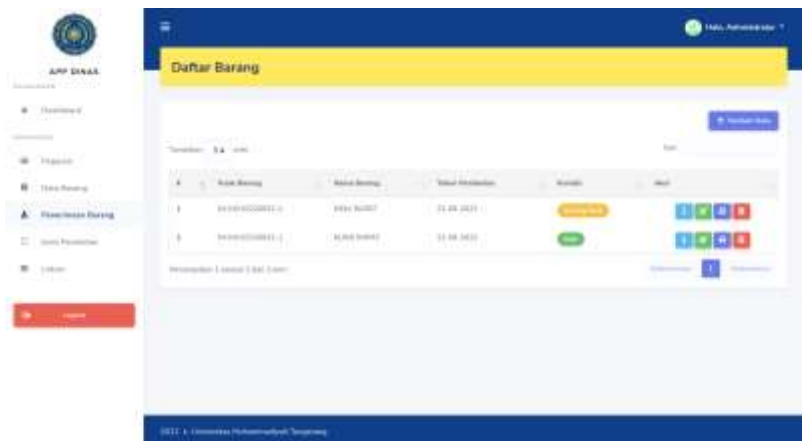
Gambar 3. 22 Halaman Data Barang

6. Tambah data barang



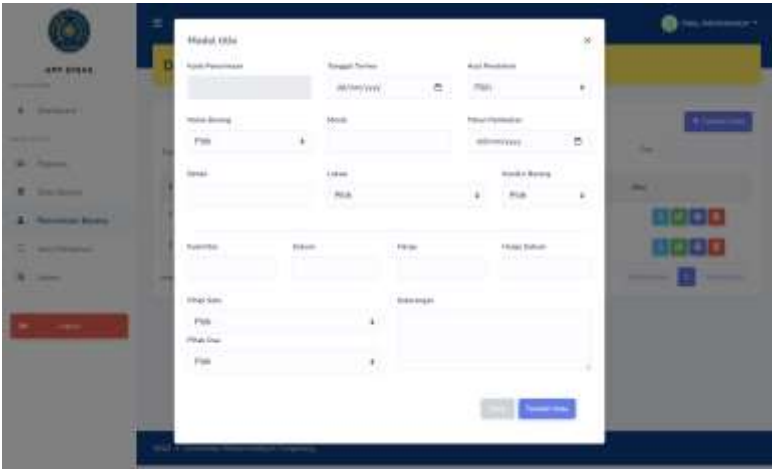
Gambar 3. 23 Halaman Tambah data barang

7. Halaman Penerimaan Barang



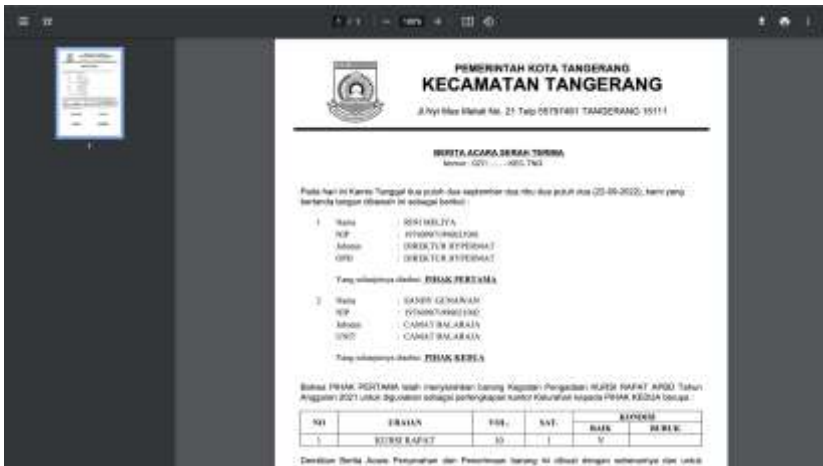
Gambar 3. 24 Halaman Penerimaan Barang

8. Tambah Data Penerimaan Barang



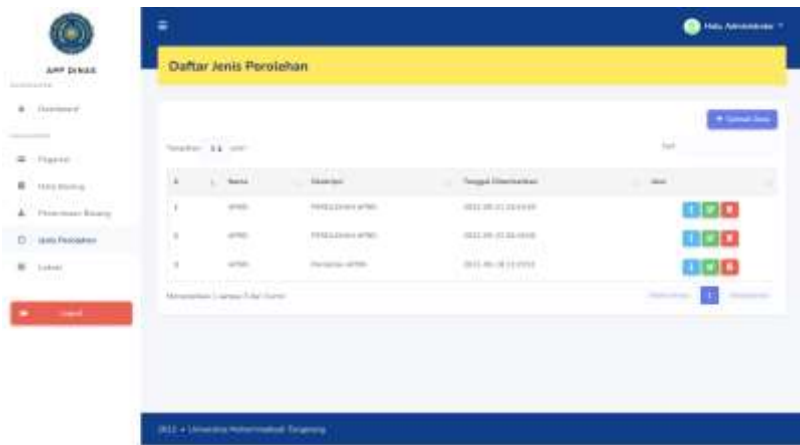
Gambar 3. 25 Halaman Penerimaan Barang

9. Print Berita Acara Penerimaan



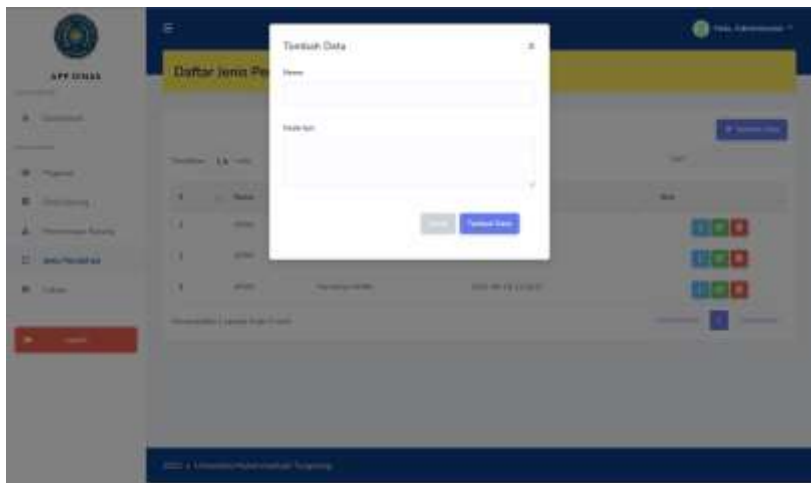
Gambar 3. 26 Print Berita Acara Penerimaan

10. Halaman Jenis Perolehan



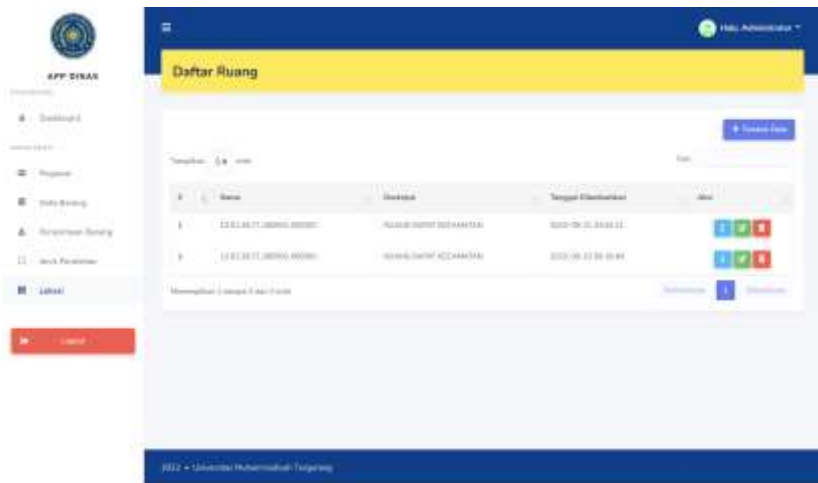
Gambar 3. 27 Halaman Jenis Perolehan

11. Halaman Tambah Jenis Perolehan



Gambar 3. 28 Halaman Tambah Jenis Perolehan

12. Halaman Data Lokasi



Gambar 3. 29 Halaman Data Lokasi

13. Tambah data Lokasi



Gambar 3. 30 Halaman Tambah data Lokasi






3.3 Testing





Setelah melakukan proses development Aplikasi, aplikasi harus melalui tahap pengujian baik itu dari pihak pembuat aplikasi dan juga nantinya pengujian oleh pihak user, pengujian dilakukan untuk meminimalisir sekecil mungkin adanya *bug* yang mungkin terjadi.


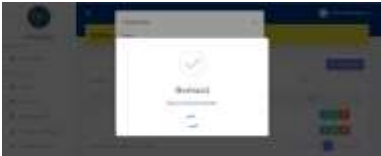


Adapun pengujian yang dilakukan adalah dengan menggunakan *Black-Box*, pengujian *Black-Box* adalah suatu pendekatan untuk menguji apakah setiap program dapat berjalan dengan benar. Berikut tabel hasil dari aplikasi yang dirancang oleh penulis:




Tabel 3. 21 Hasil Pengujian Dengan Pendekatan Black-Box

| NO | Proses | Keterangan |
|----|---|---|
| 1 | Login | Berhasil
 |
| 2 | Menu Barang
<ul style="list-style-type: none">- Tambah data barang- Merubah data barang | Berhasil
 |

| NO | Proses | Keterangan |
|--------------------------------------|---|--|
| | <ul style="list-style-type: none"> - Hapus data barang | 
 |
| <p style="text-align: center;">3</p> | <p style="text-align: center;">Menu User</p> <ul style="list-style-type: none"> - Tambah data user - Merubah data user - Hapus data user | <p style="text-align: center;">Berhasil</p> 

 |
| <p style="text-align: center;">4</p> | <p style="text-align: center;">Menu pegawai</p> <ul style="list-style-type: none"> - Tambah data | <p style="text-align: center;">Berhasil</p> |

| NO | Proses | Keterangan |
|----|---|--|
| | pegawai
- Merubah data pegawai
- Hapus data pegawai |    |
| 5. | Menu Jenis Perolehan
- Tambah data Jenis Perolehan
- Merubah data Jenis Perolehan
- Hapus data Jenis Perolehan | Berhasil
   |

| NO | Proses | Keterangan |
|----|--|---|
| 6 | <p data-bbox="259 374 406 439">Menu Lokasi Barang</p> <ul data-bbox="286 470 430 817" style="list-style-type: none"> <li data-bbox="286 470 430 565">- Tambah data lokasi <li data-bbox="286 597 430 692">- Merubah data lokasi <li data-bbox="286 723 430 817">- Hapus data lokasi | <p data-bbox="656 300 750 326">Berhasil</p>    |
| 7 | <p data-bbox="236 927 430 991">Menu Penyaluran Barang</p> <ul data-bbox="286 1022 430 1543" style="list-style-type: none"> <li data-bbox="286 1022 430 1187">- Tambah data Penyaluran Barang <li data-bbox="286 1218 430 1383">- Merubah data Penyaluran Barang <li data-bbox="286 1414 430 1543">- Hapus data Penyaluran | <p data-bbox="656 927 750 953">Berhasil</p>   |

| NO | Proses | Keterangan |
|----|--------------------|---|
| | Barang |  |
| 8 | Print Berita Acara | <p data-bbox="515 447 605 473">Berhasil</p>  |
| 9 | Laporan | <p data-bbox="695 843 785 869">Berhasil</p>  |

3.4 Pemeliharaan

Proses selanjutnya setelah tahap pengujian aplikasi adalah proses devloyment yaitu kegiatan yang bertujuan untuk menyebarkan aplikasi yang telah dikerjakan oleh pengembang. Pada tahap ini penulis menyerahkan kepada pihak kecamatan

tangerang dan setelah itu sistem perlu di maintenance atau melakukan penerimaan secara berkala terhadap sistem yang sudah dibangun untuk memastikan sistem dapat berjalan sebagaimana mestinya.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan hasil yang didapat dalam pembuatan Laporan Skripsi ini, disesuaikan dengan tujuannya, maka diperoleh beberapa kesimpulan sebagai berikut:

1. Dengan adanya sistem informasi penerimaan barang milik daerah berbasis web diharapkan dapat membantu kecamatan Tangerang dalam mengelola penerimaan barang milik daerah.
2. Dengan adanya sistem informasi penerimaan barang milik daerah diharapkan dapat menimalisir terjadinya kesalahan dalam penginputan data penerimaan barang milik daerah.
3. Dengan adanya sistem informasi pemeliharaan barang milik daerah diharapkan dapat meningkatkan efektifitas dan produktifitas kecamatan Tangerang dalam pengelolaan data penerimaan barang milik daerah.

DAFTAR PUSTAKA

- A.S Rosa dan Shalahuddin, M. 2013. UML, Use Case Diagram, Activity Diagram, Class Diagram. In Rekayasa Perangkat Lunak Terstruktur (p. undefined-undefined).
- Furi, R., 2016. Rancang Bangun Sistem Informasi Laporan Kerja Teknisi Komputer PT. Indomarco Prismatama Cabang Parung. Skripsi. Universitas Muhammadiyah Tangerang.
- F. Neuschel, Richard, 1960. Management by System. McGrawHill. New York
- Fitzgerald, Jerry., Fitzgerald, Ardra F., Stalling, Warren D. 1981. Fundamentals of Systems Analysis (2nd ed). New York: John Wiley & Sons.
- Nidhra, Srinivas dan Dondeti, Jagruthi. 2012. Black Box And White Box Testing Techniques – A Literature Review, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, 8-9.
- Rizky, Soetam. 2011. Pengujian Blackbox. Jakarta: Prestasi Pustaka. Sibero 2013 dalam Jurnal Bianglala Informatika Vol 3 No 1 Maret 2015 – lppm3.bsi.ac.id/jurnal

- Sucipto. 2011. Konsep dan Teknik Pengembangan Sistem Berbasis Teknologi Informasi. Banten: Dinas Pendidikan Provinsi Banten.
- Satzinger, Jackson, Burd. 2012. Systems Analysis and Design In A Changing World. Boston: Cengage Learning.
- Winda, A., 2016. Rancang Bangun Aplikasi Pembelajaran Matematika SD Kelas 6 Berbasis Android Pada SDN Cimone 1 Tangerang. Skripsi, Universitas Muhammadiyah Tangerang. Tangerang.
- Yakub. 2012. Pengantar Sistem Informasi. Edisi Pertama. Yogyakarta: Graha Ilmu.

TENTANG PENULIS



Rohmat Taufiq, ST., M.Kom lahir di Bojonegoro Jawa Timur pada tanggal 19 Maret 1980. Semasa kecilnya sekolah di Madrasah Ibtidaiyah Muhammadiyah 19 gunungsari. SMP Negeri 1 Baureno, Bokonegoro. SMU 1 Babad, Lamongan. D3

Jurusan Teknik Informatika di STT Mandala Bandung. S1 Jurusan Teknik Informatika di STT Mandala Bandung. Melanjutkan kuliah di S2 Magister Sistem Informasi (MSI) Program Sarjana Universitas Diponegoro (UNDIP) Semarang.

Sejak tahun 2009 penulis mengabdikan sebagai Dosen Tetap Di Program Studi Teknik Informatika, Fakultas Teknik UNIVERSITAS MUHAMMADIYAH TANGERANG yang diberi tugas tambahan sebagai WAKIL DEKAN I di Fakultas Teknik. Selain juga sebagai Dosen tidak tetap di beberapa Perguruan Tinggi yang berada di Jawa Barat.



Yeni Daniarti, S.Pd., M.Pd , lahir pada 6 Juli 1984 di Kota Tangerang, Banten, dari pasangan suami-istri, Bapak H. Hardani Saleh, SH dan Ibu Hj. Surati, S.Pd. Merupakan anak pertama dari 3 bersaudara. Penulis telah aktif mengajar pada SMP dan

SMK di Kota Tangerang sejak tahun 2010 dan meraih Sarjana

(Strata 1) Pendidikan Bahasa Inggris di STKIP Kusuma Negara pada tahun 2011. Lalu melanjutkan studi Pasca Sarjana pada tahun 2013 dan meraih Magister (Strata 2) Pendidikan Bahasa Inggris di UHAMKA-Jakarta pada tahun 2015. Sejak tahun 2017, penulis mulai aktif menulis jurnal penelitian tentang metode dan media pengajaran bahasa Inggris dan pada tahun 2020 penulis menerbitkan jurnal terindeks scopus pada *Journal of Physics: Conference Series*, dengan judul “*The Implementation of Teaching Reading Through Genre Based Approach for University Students*’. Menjadi salah satu penulis Chapter Book Media Pembelajaran Dalam *Student Centered Learning (SCL)* Jilid 2 pada tahun 2022 dan menjadi editor pada beberapa buku pembelajaran lainnya. Saat ini, aktivitas penulis lebih banyak dihabiskan untuk melaksanakan Tridarma Perguruan Tinggi dan telah menjadi dosen tetap di Universitas Muhammadiyah Tangerang sejak tahun 2017



Muhamad Fhurqon Febriansyah, S.Kom.

Lahir di Tangerang Provinsi Banten pada Tanggal 05 Februari 1998 semasa kecilnya menempuh Pendidikan Sekolah Dasar dan Menengah di SDN Merak III lulus tahun 2009, SMPN III Balaraja lulus tahun 2012,

SMAN 19 Kabupaten angeringg lulus tahun 2015 dan pada tahun 2018 melanjutkan ke jenjang S1 di Universitas Muhammadiyah Tangerang jurusan Teknik Informatika dan lulus tahun 2022



Angga Aditya Permana, M.Kom. Angga Aditya Permana (lahir pada Desember 1989 di Jakarta) seorang dosen full time di bidang Computer Science, fokus penelitian pada kriptografi, steganografi serta keamanan computer, namun pada tahun 2021 sedang mendalami topik bioinformatika dan network science. Angga juga memiliki hobi yaitu touring menggunakan motor dan juga bermain bulutangkis, saat ini sedang menjadi mahasiswa program doctoral pada IPB University. Memulai profesinya sebagai dosen pada 2013 di kampus swasta yang ada di Jakarta, pernah juga mengajar di kampus negeri yang berada di Jakarta dan Tangerang, juga pernah di undang menjadi dosen tamu untuk mengenalkan bioinformatika di kampus negeri di Jakarta. Terimakasih.