**Rancang Bangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan**

**(Studi Kasus : Kantor Pelayanan dan Pengawasan Bea dan Cukai Tipe Madya B Tarakan, Kalimantan Utara)**

**TUGAS AKHIR**

Diajukan untuk memenuhi sebagian syarat-syarat memperoleh gelar

Ahli Madya

Oleh :

**Olga Christy Kinawan**

**193140714111013**



**BIDANG MINAT D-III TEKNOLOGI INFORMASI DAN KOMPUTER**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**PROGRAM PENDIDIKAN VOKASI**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2022**

# DAFTAR ISI

[DAFTAR ISI ii](#_Toc64971158)

[DAFTAR GAMBAR v](#_Toc64971159)

[DAFTAR TABEL vi](#_Toc64971160)

[BAB I PENDAHULUAN 1](#_Toc64971161)

[1.1 Latar Belakang 1](#_Toc64971162)

[1.2 Rumusan Masalah 5](#_Toc64971163)

[1.3 Batasan Masalah 5](#_Toc64971164)

[1.4 Tujuan 6](#_Toc64971165)

[1.5 Manfaat 6](#_Toc64971166)

# BAB I PENDAHULUAN

## Latar Belakang

Di zaman sekarang peranan teknologi dirasa penting karena seiring dengan perkembangan dan kemajuan teknologi yang jauh lebih maju. Tidak dapat dipungkiri, dengan adanya perkembangan teknologi dapat memudahkan masyarakat dalam kehidupannya. Tak terkecuali dalam pelayanan di kantor pelayanan dan penagwasan bea dan cukai Tarakan, keberadaan teknologi sangat dibutuhkan dalam mengatasi masalah dengan cepat dan akurat. (Ramadhani, 2020)

Lembaga Bea dan Cukai ini bukan sebuah istilah yang memiliki satu pengertian, melainkan dua istilah yang berbeda. Bea sendiri adalah suatu Tindakan pungutan dari pemerintah terhadap barang ekspor dan impor, sedangkan cukai adalah pungutan negara kepada suatu barang yang sifat dan karakteristik yang sudah ditetapkan dalam Undang-Undang Cukai. Jadi bea cukai adalah pungutan pemerintah terhadap barang ekspor dan impor serta suata barang yang memiliki karakteristik khusus. Pada tiap bea cukai pasti memiliki perbaikan atau Redress Manifes. Redress Manifes merupakan perbaikan yang dilaukan atas kesalahan data BC 1.1 yang telah diberitahukan oleh pihak pengangkut pada saat kedatangan sarana pengangkut. BC 1.1 merupakan dokumen manifes yang memuat daftar barang niaga yang diangkut oleh sarana pengangkut memauli laut, udara dan darat pada saaat memasuki atau meninggalkan Kawasan pabean. Kesalahan data pada BC 1.1 akan mengakibatkan proses penyelesaian barang impor tidak dapat dilakukan. Pada Redress Manifes dilakukan suatu kegiatan yaitu pelaporan berkas Perbaikan yang biasa disebut dengan Redress Manifes. (Pasha, 2019)

Masalah pencatatan yang cukup rumit karena banyaknya berkas yang menumpuk, rentan terhadap data tidak akurat dan hilang, penyajian laporan memerlukan waktu yang banyak, dan lain sebagainya. Disamping itu, penyimpanan berkas dalam lemari juga menyebabkan terjadinya kesulitan dalam mencari data apablia diperlukan sewaktu-waktu.

Permasalahan yang sama juga dialami oleh Kantor Pengawasan dan pelayanan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan, Kalimantan Utara. Proses pelaporan redress manifes hanya dilakukan jika pengangkut datang ke kantor tiap Bea cukai masing-masing dan menyiapkan surat permohonan perbaikan manifes. Perbaikan redress manifes hanya diketik di kertas dan selanjutnya pengangkut yang bertanggung jawab akan menyerahkan ke kantor. Dalam kondisi ini sering menyebabkan persuratan tidak efisien, tidak akurat, sering terduplikasi, hilangnya data pengangkut apabila berkas tidak disimpan dengan baik, dan pendataan menjadi rumit karena banyaknya kertas yang menumpuk. Selain itu, dalam penyajian laporan juga rumit karena membutuhkan waktu lebih untuk menganalisis. Oleh karena itu, di Kantor Pengawasan dan pelayanan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan salah satu tugas dari Kepala kantor adalah membantu Sarana Pengangkut untuk melakukan perbaikan data adtau Redress manifes. (EMAS, 2015)

Dari penelitian terdahulu dan beberapa permasalahan yang dialami oleh Kantor Pengawasan dan pelayanan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan, maka diperlukan sebuah sistem. Kelebihan dengan menggunakan sistem dapat menghemat biaya dan menciptakan kinerja lebih cepat dan efisien. Oleh karena itu dengan adanya sistem harapannya mampu mempermudah Kantor Pengawasan dan pelayanan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan dalam perbaikan data atau redress manifes. Sistem tersebut dapat berjalan dengan baik sesuai dengan tujuan awal penelitian. Namun, penelitian ini hanya berisi fitur untuk mendata inventaris desa seperti tambah, edit, dan hapus data.

Berdasarkan uraian diatas, penulis tertarik untuk membuat tugas akhir dengan judul “Rancang Bangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan”. Bahasa pemrograman yang digunakan adalah PHP, dan MySQL sebagai *database*-nya. Dengan adanya sistem ini harapannya mampu menunjang kinerja pegawai dan sarana pengangkut dalam mengatasi berbagai permasalahan seperti kesulitan dalam melihat data karena diketik dikertas, data tidak akurat, data terduplikasi, data hilang, dan data tidak *realtime*.

## Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat disimpulkan rumusan masalah sebagai berikut:

1. Bagaimana merancang Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan?
2. Bagaimana membangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan?
3. Bagaimana cara melakukan pengolahan Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan?

## Batasan Masalah

Dalam pembuatan tugas akhir “Rancang Bangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan (Studi Kasus : Kantor Pelayanan dan Pengawasan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan)”, agar lebih terarah dan tidak mencakup terlalu luas, maka terdapat batasan-batasan sebagai berikut :

1. Sistem ini hanya dirancang untuk Kantor Pelayanan dan Pengawasan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan.
2. Sistem ini bisa diakses oleh admin, pegawai, pengguna jasa atau sarana pengangkut.
3. Sistem ini hanya dirancang dalam bentuk website.
4. Sistem ini hanya menggunakan bahasa pemrograman PHP dan MySQL sebagai *database*-nya.

## Tujuan

Tujuan dari “Rancang Bangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan (Studi Kasus : Kantor Pelayanan dan Pengawasan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan)” adalah sebagai berikut:

1. Mampu merancang Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan.
2. Mampu membangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan.
3. Mampu melakukan pengolahan Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan.

## Manfaat

Manfaat yang didapatkan dari “Rancang Bangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan (Studi Kasus : Kantor Pelayanan dan Pengawasan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan)” adalah sebagai berikut :

1. Bagi Instansi
2. Memudahkan dalam pengecekan perbaikan data (redress manifes) oleh sarana pengangkut.
3. Memudahkan dalam melakukan penyimpanan perbaikan data (sarana pengangkut) secara efisien.
4. Bagi Sarana Pengangkut
5. Memudahkan dalam perbaikan data secara efisien.
6. Memudahkan dalam melihat laporan perbaikan data (redress manifes).

**BAB II**

**KAJIAN PUSTAKA**

1. **Redress Inward dan Outward Manifes**

Redress atau perbaikan data BC 1.1 adalah Perbaikan yang dilakukan terhadap kesalahan pada data BC 1.1 yang telah dilaporkan pada saat kedatangan atau keberangkatan sarana pengangkut. Perbaikan data atau Redress ini diperlukan karena data yang telah masuk kedalam system Bea dan Cukai akan menjadi acuan bagi petugas Bea dan Cukai dalam proses pengeluaran barang impor. Data BC 1.1 yang ada di system PDE kepabeanan harus sesuai dengan kondisi sebenarnya dari barang tersebut. Kesalahan pada BC 1.1 akan mengakibatkan proses pengeluaran barang impor tidak dapat dilakukan. Oleh sebab itu diperlukan yang namanya perubahan data yang telah dilaporkan oleh pihak pengangkut menjadi data yang sebenarnya dengan cara pengajuan Perbaikan data BC 1.1 atau Redress. (EMAS, 2015)

1. **Website**

Menurut ahli (Sholechul Azis, 2013) website adalah halaman informasi yang disediakan melalui jalur internet sehingga bias diakses di seluruh dunia selama terkoneksi dengan jaringan internet. Dan website merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, dan animasi sehingga menarik untuk dikunjungi. (Laily, 2022)

1. **Bahasa yang Digunakan**

Pada saat merancang dan membuat system informasi perbaikan data inward dan outward manifest ini, bebagai Bahasa yang digunakan sebagai berikut :

1. **HTML (Hypertext Markup Language)**

HTML (Hypertext Markup Language) merupakan suatu Bahasa yang menggunakan tanda-tanda tertentu (tag) untuk menyatakan kode-kode yang harus ditafsirkan oleh browser agar halamn tersebut dapat ditampilkan secara benar. Fungsi HTML adalah untuk mengelola serangakain data dan informasi sehingga suatu dokmen dapat diakses dan ditampilkan di internet melalui layanan web. (Indonesia, 2021)

1. **CSS (Cascading Style Sheets)**

CSS (Cascading Style Sheets) merupakan salah satu bahasadesain web (style sheet language) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda. Biasanya CSS digunakan untuk mendesain sebuah halaman HTML, tetapi sekarang CSS dapat diaplikasikan untuk segala dokumen. CSS dibuat untuk memisahkan konten utama dengan tampilan dokumen yang meliputi layput, warna dan huruf. Fungsi uatama CSS adalah merancang, merubah, mendesain, membentuk halaman website. (educhannel, 2016)

1. **Bootstrap**

Bootstrap merupakan sebuah library framework CSS yang telah dibuat khusus untuk mengembangkan front end sebuah website. Bootstrap juga dikenal sebagai salah satu framework CSS, HTML, Java Script yang begitu popular dikalangan pengembang website. Bootstrap juga memiliki fungsi yaitu, bias mempercepat waktu untuk memperoses pembuatan frontend, menampilkan sisi website yang lebih modern dan juga khas dan juga tanpilan bootstrap sudah sangat responsive sehingga sangat mendukung untuk segala jenis resolusi baik smartphone, tablet, PC dan laptop. (Nugroho, Apa Itu Bootstrap? Inilah Pengertian dan Cara Menggunakannya, 2019)

1. **Bahasa Pemrograman**

Bahasa pemrograman adalah instruksi standar untuk memerintah [komputer](https://id.wikipedia.org/wiki/Komputer). Bahasa pemrograman ini merupakan suatu himpunan dari aturan [sintaks](https://id.wikipedia.org/wiki/Sintaks) dan [semantik](https://id.wikipedia.org/wiki/Semantik) yang dipakai untuk mendefinisikan [program komputer](https://id.wikipedia.org/wiki/Program_komputer). Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis [langkah](https://id.wikipedia.org/wiki/Algoritma) apa yang akan diambil dalam berbagai situasi secara persis. (Wikipedia, 2022)

1. **PHP (Hypertext Preprocessor)**

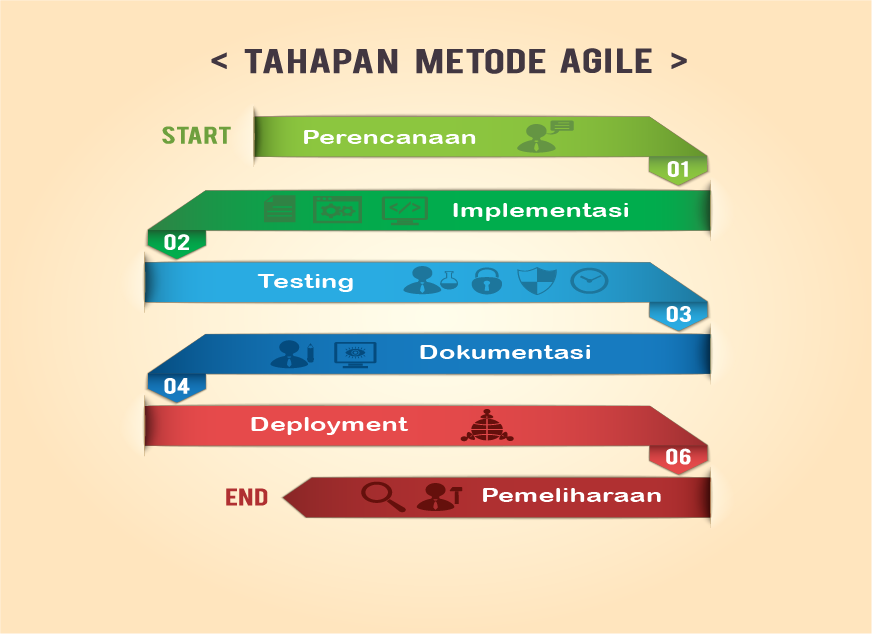
PHP (Hypertext Preprocessor)adalah bahasa scripting server-side, Bahasa pemrograman yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi Web. PHP singkatan dari Hypertext Pre-processor, yang sebelumnya disebut Personal Home Pages. Atau bisa diartikan sebagai bahasa pemrograman umum yang berarti dapat disematkan ke dalam kode HTML, atau dapat digunakan dalam kombinasi dengan berbagai sistem [templat web](https://www.jagoanhosting.com/blog/template-wordpress-gratis-2019/), sistem manajemen konten web, dan kerangka kerja web. (INTERMEDIA, 2019) Ada berbagai macam kelebihan yang dimiliki PHP jika dibandingkan dengan bahasa pemrograman lainnya, yaitu : (Laila, 2022)

1. Mampu membuat web menjadi lebih dinamis.
2. PHP berjalan secara Web Base ynag artinya seluruh Sistem Operasi bahkan HP yang memiliki Web Browser dapat menggunakan program PHP dan program yang dibuat menggunakan PHP bisa dijalankan oleh Semua Sistem Operasi.
3. PHP adalah open source, yang berarti bahwa siapa pun dapat menggunakannya secara gratis.
4. Jika dibandingkan dengan ASP atau Java, kini aplikasi PHP lebih cepat.
5. Mendukung berbagai macam paket Database, misalnya Oracle, MySQL, PostgrSQL dll.
6. Kompilasi / Compile tidak diperlukan dalam penggunaan bahasa pemrograman PHP.
7. Terdapat banyak server web yang mendukung PHP, salah satunya yaitu Lighttpd, Apache, IIS dan lainnya.
8. Mengembangkan aplikasi PHP mudah karena ada banyak dokumentasi, instruksi dan pengembang untuk mengembangkannya.
9. Banyak yang gratis dan mau menggunakan aplikasi dan program PHP seperti WordPress, Prestashop dan lainnya.
   1. **Metode perancangan yang digunakan**
10. **Agile Development Syestem**

Dalam membangun sistem ini menggunakan metode *agile development system*. *Agile Development System* adalah pendekatan khusus untuk manajemen proyek yang digunakan dalam pengembangan perangkat lunak. Metode ini membantu tim dalam merespon ketidakpastian pengembangan perangkat lunak dengan menggunakan incremental berulang yang umumnya disebut sprint. Dalam menjalankan metode agile atau yang lebih dikenal *Agile Development System* ini, dilakukan secara kolaboratif, terstruktur dan terorganisir yang menyangkut berbagai pihak yang bekerja sama. (Junaedi, Metode Agile Development: Pengertian, Tujuan, dan Keunggulan, 2022)

Berikut ini merupakan beberapa kelebihan dari metode Agile: (Adani, 2020)

1. Proses pengembangan perangkat lunak membutuhkan waktu yang relatif cepat dan tidak membutuhkan resources yang besar.
2. Perubahan dapat ditangani dengan cepat sesuai dengan kebutuhan client.
3. Client dapat memberikan feedback kepada tim pengembang dalam proses pembuatan program.

****

Gambar 2.1 Tahapan Agile

Berdasarkan gambar 2.1 dapat dijelaskan bahwa ada beberapa tahapan dalam pengembangan metode Agile, yaitu : (Musyaffa, 2020)

1. Perencanaan, pada langkah ini pengembang dan klien membuat rencana tentang kebutuhan dari perangkat lunak yang akan dibuat.
2. Implementasi, bagian dari proses dimana programmer melakukan pengkodean perangkat lunak.
3. Tes perangkat lunak, disini perangkat lunak yang telah dibuat di tes oleh bagian kontrol kualitas agar bug yang ditemukan bisa segera diperbaiki dan kualitas perangkat lunak terjaga.
4. Dokumentasi, setelah dilakukan tes perangkat lunak langkah selanjutnya yaitu proses dokumentasi perangkat lunak untuk mempermudah proses maintenanance kedepannya.
5. Deployment, yaitu proses yang dilakukan oleh penjamin kualitas untuk menguji kualitas sistem. Setelah sistem memenuhi syarat maka perangkat lunak siap dideployment.
6. Pemeliharaan, langkah terakhir yaitu pemeliharaan. Tidak ada perangkat lunak yang 100% bebas dari bug, oleh karena itu sangatlah penting agar perangkat lunak dipelihara secara berkala.
7. **UML (Unified Modeling Language)**

UML (Unified Modeling Language) adalah sebuah sebuaj metode pemodelan yang digunakan untuk memvisualkan sebuah perancangan berorientasi objek atau yang kita kenal dengan OOP. (Ramadhanti, 2021)

Berikut adalah beberapa tujuan dan fungsi dari UML : (Ramadhanti, 2021)

1. Sebuah bahasa pemodelan yang bisa digunakan dan dimanfaatkan di masa yang akan datang, baik oleh manusia maupun oleh mesin.
2. Mampu memberikan bahasa pemodelan secara visual kepada user dengan berbagai bahasa pemograman yang digunakan.
3. Memungkinkan untuk menyatukan praktek terbaik dalam suatu pemodelan.
4. UML digunakan untuk memodelkan sistem dengan menggunakan OPP, sehingga tidak hanya digunakan untuk pemodelan perangkat lunak saja.
5. Mampu menghasilkan model yang sangat baik, mudah untuk digunakan serta memungkinkan untuk dikerjakan secara terpisah.
6. Dapat dimanfaatkan sebagai *blue print*, karena pemodelan ini sangat rinci dan tertata.
   * 1. **Use Case Diagram**

Use Case diagram adalah diagram yang menggambarkan hubungan antara aktor dengan sistem. Use case diagram bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Use case diagram juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan  bisa juga mempresentasikan sebuah interaksi aktor dengan sistem. Komponen tersebut kemudian menjelaskan komunikasi antara aktor,  dengan sistem yang ada. Dengan demikian, use case dapat dipresentasikan dengan urutan yang sederhana, dan akan mudah dipahami oleh para konsumen. Manfaat dari use case sendiri adalah untuk memudahkan komunikasi dengan menggunakan domain expert dan juga end user, memberikan kepastian pemahaman yang pas tentang requirement atau juga kebutuhan sebuah sistem. (Hutauruk, 2017)

##### Tabel 2.1 Simbol Use Case Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Actor | Actor mewakili pengguna atau organisasi yang berinteraksi dengan use case pada sistem untuk mencapai suatu tujuan. |
|  | Sistem | Suatu sistem menunjukkan apapun yang dikembangkan. Dapat berupa *website*, komponen perangkat lunak, aplikasi, atau lainnya. |
|  | Use Case | Use Case berbentuk oval yang berfungsi untuk menggambarkan suatu proses yang terjadi pada sebuah sistem. |
|  | Association | Menunjukkan hubungan antara actor dengan use case. Digunakan juga untuk menghubungkan link antar element. |
|  | Generalization | Menunjukkan hubungan generalisasi antara actor dengan use case atau antara use case dengan use case. Dapat disebut juga *inheritance* (pewarisan). |
|  | Include | Menunjukkan bahwa suatu use case merupakan fungsionalitas dari use case lainnya. |
|  | Extend | Menunjukkan perluasan dari use case jika suatu kondisi terpenuhi. |

1. **Activity Diagram**

Activity Diagram adalah bentuk visual dari alur kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, atau pengulangan. Dalam Unified Modeling Language(UML), diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun alur aktivitas dalam organisasi. Selain itu diagram aktivitas juga menggambarkan alur kontrol secara garis besar. (Tanoto, 2020)

Activity diagram bisa juga dianggap sama seperti flowchart (diagram alur), namun meskipun diagram terlihat seperti sebuah diagram alur, tetapi sebenarnya berbeda. Diagram aktivitas menunjukkan aliran yang berbeda seperti paralel, bercabang, bersamaan dan tunggal. Simbol-simbol pada activity diagram dapat dilihat pada Tabel 2.2. Tabel 2.2 Simbol – simbol Activity Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Kondisi awal/Start point | Menunjukkan status awal dari sebuah activity diagram. |
|  | Kondisi akhir/End point | Menunjukkan status akhir dari sebuah activity diagram. |
|  | Activity | Menunjukkan aktivitas yang dilakukan sistem pada activity diagram. |
|  | Transition | Menunjukkan transisi dari aktivitas ke aktivitas lainnya. |
|  | Swimlane | Mengelompokkan aktivitas berdasarkan actor. |
|  | Decision | Menunjukkan pengecekan jika ada aktivitas lebih dari satu mengharuskan untuk mengambil suatu tindakan pada kondisi tertentu. |
|  | Fork (Percabangan) | Menunjukkan aktivitas yang dilakukan secara paralel. |
|  | Join (Penggabungan) | Menunjukkan aktivitas yang digabungkan. |

* 1. **Perangkat Lunak Yang di Gunakan**

Pada saat merancang dan membuat sistem informasi perbaikan data (Redress) Inward dan Outward Manifes ini, perangkat lunak yang digunakan sebagai berikut :

1. **Visual Studio Code**

**Visual Studio Code** merupakan editor kode sumber yang di buat dan diciptakan berbagai software seperti Microsoft untuk Windows, Linux, dan macOS. Fitur ini termasuk dukungan untuk debugging, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, pemfaktoran ulang kode, dan Git yang disematkan. Pengguna dapat mengubah tema, pintasan keyboard, preferensi, dan menginstal saham yang menambahkan fitur tambahan. Dalam Survei Pengembang Stack Overflow 2021, Visual Studio Code dinilai sebagai alat lingkungan pengembang paling populer, dan 71,06% dari 82.277 responden melaporkan menggunakannya. Visual Studio Code pertama kali diumumkan oleh Microsoft pada konferensi Build 2015 pada 29 April 2015, dan versi pratinjau dirilis. (GriyaWebsite, 2020)

1. **XAMPP**

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah cross platform sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. Program aplikasi XAMPP berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan. XAMPP bekerja secara offline layaknya web hosting biasa namun tidak bisa diakses oleh banyak orang. (Nugroho, Pengertian XAMPP , 2019)

1. **Draw.io**

Draw.io adalah sebuah website yang didesain khusus untuk menggambarkan diagram secara online. Semua fitur yang ada pada situs ini bisa kalian nikmati hanya dengan bermodalkan browser yang mendukung HTML 5. (Lathif, 2019)

1. **Database**

Database adalah sekumpulan data yang dikelola berdasarkan ketentuan tertentu yang saling berkaitan sehingga memudahkan dalam pengelolaannya. database atau basis data merupakan sekumpulan data atau informasi yang tersimpan secara sistematis. Database memiliki peran penting dalam perangkat untuk mengumpulkan informasi, data, atau file secara terintegrasi. Database berwujud tabel yang terdiri dari kolom dan baris yang memuat atribut dan nilai tertentu. Adapun jumlah kolom dan baris dalam suatu database tergantung pada jumlah kategori atau jenis informasi yang perlu disimpan. Fungsi database adalah untuk menghindari data ganda yang tersimpan. Suatu database management system (DBMS) dapat diatur supaya bisa mengenali duplikasi data ketika diinput. Namun selain untuk menghindari data ganda, database memiliki fungsi lainnya, antara lain: (Mulachela, 2021)

* 1. Mengelompokan data dan informasi. Memudahkan dalam identifikasi data.
  2. Memudahkan proses akses, menyimpan, pembaharuan, dan penghapusan data.
  3. Menjadi alternatif terkait masalah penyimpanan ruang dalam suatu aplikasi.
  4. Menjaga kualitas data yang diakses sesuai input. Menunjang kinerja aplikasi yang memerlukan penyimpanan data.
  5. Meminimalisasi redunsasi data atau munculnya banyak datadalam file yang berbeda.

1. **MySQL**  MySQL adalah tool yang digunakan khusus untuk mengolah SQL (Structured Query Language). SQL sendiri merupakan sebuah bahasa yang digunakan untuk mengakses baris data relasi. Mudahnya adalah untuk mengakses bahasa dalam komputer. Karena SQL dan MySQL sifatnya khusus, maka hanya orang-orang yang berkecimpung dalam dunia IT lah yang familiar dengan bahasa ini. Jika SQL adalah bahasa yang digunakan untuk mengakses data dalam database, maka MySQL adalah tool atau software atau alat yang digunakan untuk mengolah SQL. SQL adalah bahasanya, MySQL adalah alatnya. Yang perlu dipahami, MySQl bukanlah alat satu-satunya yang bisa mengolah SQL. Masih ada banyak alat lain. Sebut saja Microsoft Acces. Keduanya hanya berbeda pada cara penggunaan dan pengolahan databasenya saja. (litalia, 2019) Adapun kelebihan MySQl dalam penggunaanya dalam database adalah: (Arlina, 2016)
2. Gratis sehingga MySQL dapat dengan mudah untuk mendapatkannya
3. MySQL stabil dalam pengoprasiannya
4. MySQL mempunyai sistem keamanan yang cukup baik
5. Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas
6. Sangat fleksibel dengan berbagai macam program
7. Perkembangan dari MySQl sangat cepat
8. **ERD/Relasi DB**

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. vERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. (Mecha, 2014) (Admin, 2013)

Simbol-simbol dalam ERD (*Entity Relationship Diagram*) adalah sebagai berikut :

Tabel 2.4 Simbol – simbol ERD (*Entity Relationship Diagram*)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Komponen** | **Keterangan** |
|  | Entitas | Kumpulan objek dimana kita akan menyimpan data. |
|  | Atribut | Mendeskripsikan suatu entitas. |
|  | Relasi | Menunjukkan hubungan antar satu entitas atau lebih. |
|  | Link | Garis yang menghubungkan kumpulan entitas dengan relasi maupun relasi dengan atribut. |

1. **PhpMyAdmin**

PhpMyAdmin adalah aplikasi web untuk mengelola [database](https://www.termasmedia.com/lainnya/software/69-pengertian-database.html) MySQL dan database MariaDB dengan lebih mudah melalui antarmuka (interface)grafis. Aplikasi web ini ditulis menggunakan bahasa pemrograman PHP. Sebagaimana aplikasi-aplikasi lain untuk lingkungan web (aplikasi yang dibuka atau dijalankan menggunakan browser), phpMyAdmin juga mengandung unsur HTML/XHTML, CSS dan juga kode JavaScript. Aplikasi web ini ditujukan untuk memudahkan pengelolaan basis data MySQL dan MariaDB dengan penyajian antarmuka web yang lengkap dan menarik. Keuntungan dengan hadirnya phpMyAdmin tidak saja dapat dinikmati oleh penyedia web hosting, Anda juga bisa menginstal phpMyAdmin di server Anda sendiri (server lokal) asalkan syarat minimumnya (minimum requirenments) telah terpenuhi. (Admin, 2013)

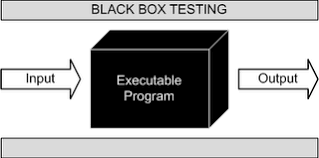
1. **Pengujian Black Box**

Pengujian Black Box adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu koatak hitam, kit hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian black box, mengevaluasi hanya dari tampilan luarnya(interface nya) , fungsionalitasnya.tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui input dan output). (tkjpnup, 2013)

Keuntungan dari Pengujian Black Box: (Ok, 2017)

* 1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
  2. Pengujian yang dilakukan berdasarkan sudut pandang *user* agar dapat mengungkapkan konsistensi atau ambiguitas dalam spesifikasi.
  3. Programmer dan tester memiliki ketergantungan satu sama lain.
  4. Efisien untuk segmen kode besar
  5. Akses kode tidak diperlukan.
  6. Pemisahan antara perspektif pengguna dan pengembang

Untuk sistem kerja Pengujian Black box dapat dilihat pada Gambar 2.2



Berdasarkan gambar 2.2 sistem kerja pengujian black box dapat dijelaskan sebagai berikut :

1. Input

Tester menginputkan data yang benar dan salah. Disini tester tidak perlu menguasai bahasa pemrograman yang digunakan dan tidak perlu mengetahui bagaimana sistem kerja internal dari perangkat lunak tersebut.

1. Process

Sistem menerima inputan dan akan menganalisisnya.

1. Output

Hasil output keluar, tester akan memeriksa apakah output sesuain dengan harapan.

**BAB III**

**METODE PENYELESAIAN TUGAS AKHIR**

## Tempat dan Waktu Penelitian

Penulis melaksanakan penelitian selama 4 bulan yaitu dimulai pada bulan Juli 2021 sampai bulan November 2021 di Kantor Pengawasan dsan Pelayanan Bea dan Cukai kota Tarakan. Kantor Pengawasan dan Pelayanan Bea dan Cukai merupakan sebuah tempat yang digunakan oleh pemerintah Kota Tarakan untuk melaksanakan kegiatan Ekspor dan Impor.

## Sistematika Perancangan

Metode yang digunakan dalam membangun Sistem informasi Perbaikan Data (Redress) Inward dan Outward Manifes adalah metode *agile development system*. Untuk langkah-langkah dalam membangun sistemnya adalah sebagai berikut :

### Analisa Permasalahan

Kantor Pengawasan dsan Pelayanan Bea dan Cukai yang terletak di Kota Tarakan merupakan sebuah wadah untuk melaksanakan berbagai kegiatan. Salah satu tugas Kantor Pengawasan dan Pelayanan Bea dan Cukai adalah Perbaikan data BC 1.1 atau Redress. Selama ini dalam dan Pelayanan Bea dan Cukai adalah Perbaikan, Kantor Pengawasan dan Pelayanan Bea dan Cukai masih menggunakan cara manual, yaitu dating ke kantor dan mengisi form permohonan. Hal ini sering terjadi berbagai masalah seperti pembukuan yang tidak akurat, data hilang, dan data terduplikasi. Selain itu, menyulitkan pihak pegawai dalam membuat laporan karena membutuhkan waktu lebih untuk menganalisis. Proses pelaporan redress manifes hanya dilakukan jika pengangkut datang ke kantor tiap Bea cukai masing-masing dan menyiapkan surat permohonan perbaikan manifes. Perbaikan redress manifes hanya diketik di kertas dan selanjutnya pengangkut yang bertanggung jawab akan menyerahkan ke kantor. Dalam kondisi ini sering menyebabkan persuratan tidak efisien, tidak akurat, sering terduplikasi, hilangnya data pengangkut apabila berkas tidak disimpan dengan baik, dan pendataan menjadi rumit karena banyaknya kertas yang menumpuk. Selain itu, dalam penyajian laporan juga rumit karena membutuhkan waktu lebih untuk menganalisis. Oleh karena itu, di Kantor Pengawasan dan pelayanan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan salah satu tugas dari Kepala kantor adalah membantu Sarana Pengangkut untuk melakukan perbaikan data adtau Redress manifest.

Oleh sebab itu, maka dibuatlah sebuah website sistem informasi perbaikan data (Redress) Inward dan Outward Manifes, kebutuhan dasar yang harus bisa dipenuhi oleh website adalah sebagai berikut :

1. Mampu merancang Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan.
2. Mampu membangun Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan.
3. Mampu melakukan pengolahan Sistem Informasi Perbaikan Data (REDRESS) Pada Manifes Inward dan Outward Kepabeanan.

### Perencanaan Aplikasi

Setelah melakukan tahapan terkait analisa permasalahan, tahap selanjutnya yang dilakukan adalah perencanaan aplikasi mulai dari perancangan UML hingga perancangan database.

Langkah-langkah yang dilakukan pada tahap analisis kebutuhan sistem ini sebagai berikut :

1. Perancangan Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan secara singkat bagaimana sistem berjalan dan siapa saja yang menggunakan sistem tersebut. Pada perancangan ini diawali dengan menentukan aktor-aktor yang akan terlibat pada sistem yang akan dibangun. Lalu menentukan action-action bagaimana aktor-aktor tersebut akan menggunakan dan memanfaatkan sistem. Diagram ini menjelaskan bagaimana gambaran fungsionalitas dari sistem yang akan dibangun.

1. Perancangan Activity Diagram

Pada perancangan activity diagram dilakukan dengan menggambarkan event-event yang terjadi pada use case diagram untuk menjelaskan fungsionalitas atau proses logika yang terimplementasi oleh program.

1. Perancangan Database

Pada perancangan database diawali dengan merancang ERD (Entity Relationship Diagram). Kemudian membuat tabel-tabel beserta atributnya pada masing-masing tabel di dalam database.

1. Perancangan Desain Antarmuka

Pada perancangan desain antarmuka dilakukan dengan pembuatan mockup atau visualisasi konsep desain pada sistem.

### Implementasi

Tahap selanjutnya yang dilakukan yaitu implementasi sistem. Langkah-langkah yang dilakukan pada tahap implementasi antara lain :

1. Membuat tabel-tabel di dalam database berdasarkan rancangan ERD (Entity Relationship Diagram).
2. Membuat tampilan layout website berdasarkan desain antarmuka yang sudah dirancang dengan menggunakan script-script HTML dan CSS.
3. Membangun website yang dapat terhubung dengan database MySQL dengan menggunakan bahasa pemrograman PHP dan framework AngularJS berdasarkan use case diagram, activity diagram, sequence diagram, dan class diagram yang telah dirancang sebelumnya.

### Pengujian Website

Pengujian dilakukan dengan tujuan untuk memastikan bahwa sistem yang dibangun dapat berjalan dengan baik dan menampilkan sesuai dengan kebutuhan yang diharapkan. Pengujian pada sistem ini menggunakan metode pengujian *black box*. Metode pengujian black box dilakukan tanpa melihat struktur script sistem yang diuji. Dalam proses pengerjaannya dilakukan dengan menginputkan data pada sistem tanpa melihat script sistem atau internal pada sistem, kemudian sistem memproses inputan tersebut lalu mengeluarkan hasil keluaran (output). Sehingga dapat dilihat hasil keluaran (output) tersebut sesuai atau tidak dengan fungsi dan fitur yang sudah direncanakan.

### Pengaplikasian dan perawatan

Website yang selesai dibuat dapat diaplikasikan ke user. Selain itu, terdapat proses perawatan atau *maintenance* pada sistem yang dilakukan sebagai berikut :

1. Memastikan tampilan *website* selalu dalam kondisi rapi.
2. Penambahan fitur dapat disesuaikan dengan kebutuhan Kantor Bea dan Cukai.
3. Perubahan tampilan dapat disesuaikan dengan kebutuhan Kantor Bea dan Cukai .

## Teknik Pengambilan Data

Teknik pengambilan data yang digunakan penulis dalam menyelesaikan tugas akhir ini adalah dengan melakukan observasi dan wawancara.

* + 1. **Observasi**

Dilakukan observasi dengan melakukan pengamatan secara langsung pada Kantor Pengawasan dan Pelayanan Bea dan Cukai Kota Tarakan. Observasi ini dilakukan mulai bulan Juli 2021 hingga bulan November 2021. Berdasarkan observasi tersebut, penulis mendapatkan hasil sebagai berikut :

1. Pada bagian Perbaikan Data (Redress) masih dilakukan secara manual dating ke kantor untuk mengisi form perbaikan dan permohonan . Hal ini sering menyebabkan perbaikan tidak efisien, sering terjadinya keterlamabatan data antara sekretaris dan nagian Pengolahan data dan Administrasi Dokumen umum sering tidak sama karena perbaikan masih manual. Selain itu, perbaikan menjadi rumit karena banyaknya kertas yang menumpuk dan dalam penyajian laporan juga kurang efisien karena butuh waktu yang lebih lama dalam penyajiannya.
2. Masalah pencatatan yang cukup rumit karena banyaknya berkas yang menumpuk, rentan terhadap data tidak akurat dan hilang, penyajian laporan memerlukan waktu yang banyak, dan lain sebagainya. Disamping itu, penyimpanan berkas dalam lemari juga menyebabkan terjadinya kesulitan dalam mencari data apablia diperlukan sewaktu-waktu. Permasalahan yang sama juga dialami oleh Kantor Pengawasan dan pelayanan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan, Kalimantan Utara. Proses pelaporan redress manifes hanya dilakukan jika pengangkut datang ke kantor tiap Bea cukai masing-masing dan menyiapkan surat permohonan perbaikan manifes. Perbaikan redress manifes hanya diketik di kertas dan selanjutnya pengangkut yang bertanggung jawab akan menyerahkan ke kantor. Dalam kondisi ini sering menyebabkan persuratan tidak efisien, tidak akurat, sering terduplikasi, hilangnya data pengangkut apabila berkas tidak disimpan dengan baik, dan pendataan menjadi rumit karena banyaknya kertas yang menumpuk. Selain itu, dalam penyajian laporan juga rumit karena membutuhkan waktu lebih untuk menganalisis. Oleh karena itu, di Kantor Pengawasan dan pelayanan Bea dan Cukai Tipe Madya Pabean B Kota Tarakan salah satu tugas dari Kepala kantor adalah membantu Sarana Pengangkut untuk melakukan perbaikan data adtau Redress manifes.
   * 1. **Wawancara**

Pada tahap ini penulis melakukan wawancara langsung dengan Bapak Hendra Hutabarat sebagai Kepala Bagian Pengolahan Data dan Administrasi Dokumen (PDAD). Dari hasil wawancara tersebut penulis mendapat kesimpulan sebagai berikut :

1. Kantor Pengawasan dan Pelayanan Bea dan Cukai Kota Tarakan membutuhkan sebuah *website* untuk mempercepat kinerja perbaikan data (Redrerss) tersebut. Sehingga tidak dilakukan secara manual, karena hal itu sangat menyulitkan Pegawai kantor terutama dalam hal penyajian laporan yang membutuhkan waktu lebih karena harus menganalisis dari buku besar.
2. Kantor Pengawasan dan Pelayanan Bea dan Cukai Kota Tarakan membutuhkan sebuah *website* untuk Perbaikan Data (Redress). Sehingga membantu Pegawai dalam perbaikan data (Redress) Inward dan Outward Manifes Kota Tarakan.
3. Kantor Pengawasan dan Pelayanan Bea dan Cukai Kota Tarakan membutuhkan website yang dapat bisa langsung melakukan *import* data perbaikan data (Redress) yang dalam bentuk *Microsoft Excel* yang diberikan oleh pusat.
4. Kantor Pengawasan dan Pelayanan Bea dan Cukai Kota Tarakan membutuhkan *website* dengan fitur *print* dan *export* excel di dalam penyajian laporan.

## Teknologi yang Digunakan

Daftar teknologi yang digunakan sebagai berikut :

### Perangkat Keras (Hardware)

Kebutuhan perangkat keras yang dibutuhkan dalam membangun sistem ini dapat dilihat pada tabel 3.1.

##### Tabel 3.1 Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat Keras** | **Identifikasi** |
| 1. | Laptop | ASUS N4000   * Sistem Operasi : Windows 10 * RAM : 16384Mb * Processor : AMD A12-97020P RADEON R7 |
| 2. | Perangkat pendukung | Mouse |

### Perangkat Lunak (Software)

Kebutuhan perangkat keras yang dibutuhkan dalam membangun sistem ini dapat dilihat pada tabel 3.2.

##### Tabel 3.2 Kebutuhan Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat Lunak** | **Identifikasi** |
| 1. | Text Editor | Visual Studio Code |
| 2. | Browser | Google Chrome |
| 3. | Web Server | XAMPP v3.2.2 |
| 4. | Database | PHPMyAdmin for MySQL |
| 5. | Bahasa Pemrograman | PHP |

## Jadwal Kegiatan

Kegiatan ini direncanakan dan dibangun dalam waktu maksimal lima bulan (jika bisa kurang dari lima bulan) dengan rincian sebagai berikut :

##### Tabel 3.3 Jadwal Kegiatan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Kegiatan** | **Juli** | | | | **Agustus** | | | | **September** | | | | **Oktober** | | | | **November** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| 1. | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. | Analisis Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. | Desain Use Case Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. | Desain Activity Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. | Desain Sequence Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6. | Desain Class Diagram |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7. | Desain ERD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**BAB IV**

**HASIL DAN PEMBAHASAN**

1. **Deskripsi Project**

Sistem informasi perbaikan data (REDRESS) pada manifes inward dan outward kepabeanan merupakan sebuah sistem yang berbasis website. Sistem ini dibangun untuk membantu pegawai kantor dalam melakukan perbaikan data ekspor dan impor suatu perusahaan. Sehingga mempermudah tidak terjadinya dat terduplikasi, data hilang dan data tidak real time. Sistem ini mempunyai 4 pengguna atau actor, yaitu admin, pegawai (Perbendaharaan), sekretaris, dan pengguna jasa.

Admin memiliki akses untuk men gecek user yang mengunakan website dan juga mengecek atgau memantau berapa data yang sudah masuk, dikembaliakn, dan diterima oleh pembendaharaan. Perbendaharaan memiliki akses untuk mengecek kelengkapan dan menerima atau mengembalikan dokumen dari pengguna jasa. Sekertaris memiliki akses untuk melihat dokumen masuk, diterima, dan dikembalikan yang telah diberikan oleh perbendaharaaan dan sekertaris juga memiliki akses untuk menyampaikan atau mengirim pengumuman kepada pengguna jasa jika ada pengumuman. Pengguna jasa memiliki akses untuk memberikan atau mengirim kelengkapan dokumen ke website untuk perbaikan data (REDRESS).

1. **Identifikasi Masalah**

Identifikasi masalah dilakukan untuk menjelasksan kebutuhan dasar apa saja pada sistem yang dibuat. Kebutuhan dasar ini diperlukan dalam membangun sistem informasi perbaikan data (REDRESS) inward dan outward manifes kepabeanan adalah sebagai berikut:

1. Sistem mampu menyimpan data pengguna jasa.
2. Sistem mampu melakukan pencatatan perbaikan data.
3. Sistem mampu menampilkan data perbaikan data.
4. Sistem mampu menampilkan semua laporan atau perbaikan data dari pengguna jasa.
5. **Perancangan**

Pada tahap perancangan sistem ini menggunakan pendekatan oriented. Tahap perancangan sistem pada pendekatan object oriented dengan pemb uatan diagram perancangan seperti use case diagram, activity diagram, class diagram, relasi database, dan perancangan interface. Pada tahap ini tidak hanya dilakukan perancanagn namun mengidentifikasi pengguna atau actor yang ada pada sistem.

1. **Identifikasi pengguna**

Pada tahap ini digunakan untuk mengidentifikasi pengguna atau actor pada sistem. Terdapat 4 pengguna yang terlibat dalam penggunaan sistem. Daftar pengguna dapat dilihat pada Tabel 4.1.

Tabel 4.1 Identifikasi Pengguna

|  |  |  |
| --- | --- | --- |
| **No** | **Pengguna** | **Deskripsi Pengguna** |
| 1. | Admin | Admin memiliki akses untuk Login, melakukan dasbord, mengecek ddata user, menginput data satuan berat, melihat semua data, melihat dokumen yang dipproses, melihat dokumen yang di terima, melihat dokumen yang dikembalikan, dapat melakukan pencarian, dapat mengekspor data dalam bentuk CSV, EXCEL, dan PDF. |
| 2. | Perbendaharaan | Perbendaharaan memiliki akses untuk login, melihat dashboard, melihat data pembatalan, memeriksa dokumen pembatalan manifes, melihat data penambahan pos, mengisi status dokumen diterima atau dikembalikan,, dapat melakukan pencarian, dapat m,engekspor data menjadi CSV, Ecxel, dan PDF. |
| 3. | Sekertaris | Sekertaris memiliki akses untuk Login, melihat dashboard, melihat dokumen, mengirim pengumuman, ke pengguna jasa, dapat melakukan pencarian, dapat m,engekspor data menjadi CSV, Ecxel, dan PDF. |
| 4. | Pengguna Jasa | Pengguna jasa memiliki akses Melakukan register yang belum memiliki akun, Login, melihat dashboard, menginputkan data pembatalan nmanifes, menginput data penambahan pos, melihat pembatalan data manifes, melihat penambahan data manifes, dapat mengin put berkas pos baru, dapat melihatb details penambahan pos, dapat melakukan pencarian, mengirim data yang suda dinput, dapat melihat pengumuman dari sekertaris, dapat melakukan pencarian, dapat mendownload atau menyimpan pengumuman yang telah dikirim, dapat mengekspor data dalam bentuk CSV, Excel, dan PDF. |

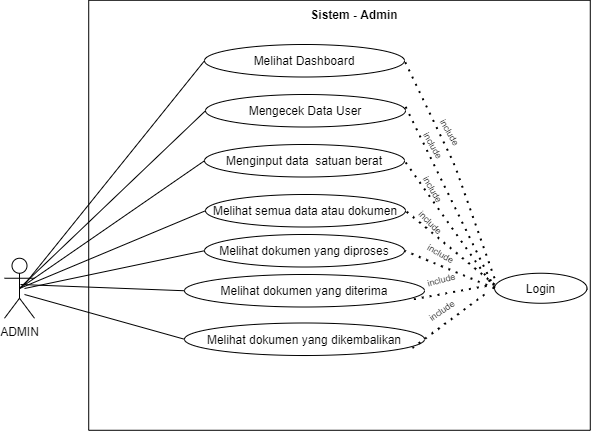
1. **Daftar Kebutuhan**

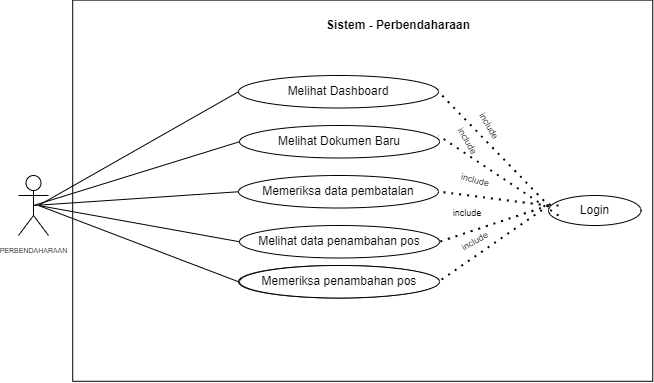
Daftar kebutuhan dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan

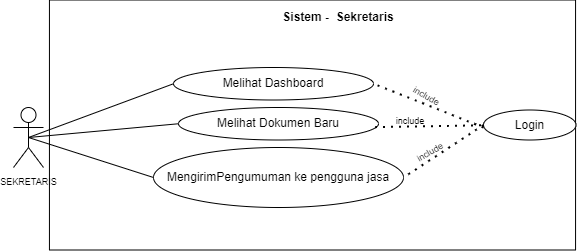
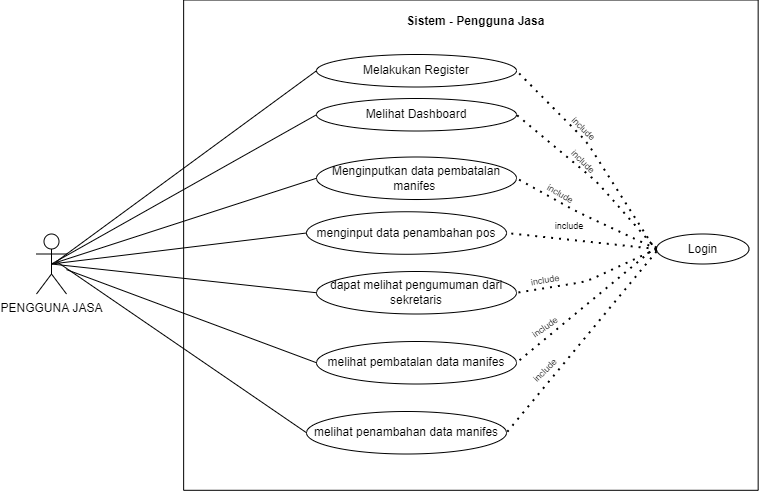
|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan** | **Pengguna** |
| 1. | User harus login terlebih dahulu untuk dapat login | Admin, Perbendaharaan, Sekertaris, dan Pengguna jasa. |
| 2. | User dapat melihat, mencari, menambah, mengedit, dan menghapus data | Admin, Perbendaharaan, Sekertaris, dan Pengguna jasa. |
| 3. | User dapat melihat, mencari, menambah, mengedit, menghapus, mendownload format import dalam bentuk Microsoft Excel, dan melakukan import data dalam bentuk Microsoft Excel data | Admin, Perbendaharaan, Sekertaris, dan Pengguna jasa. |

1. **Use Case Diagram** Sistem ini terdiri dari tiga pengguna yaitu admin 1 (sekretaris desa dan kepala urusan umum), admin 2 (kepala urusan keuangan dan kepala dusun), dan kepala desa. Untuk gambaran use case diagram pada sistem informasi inventaris desa dan pencatatan pajak bumi bangunan dapat dilihat pada gambar 4.1 sampai gambar 4.4.

****

Gambar 4.1 Use Case Admin

Gambar 4.2 Use Case Perbendaharaan

Gambar 4.3 Use Case SekertariS

Gambar 4.4 Use Case Pengguna Jasa

1. **Skenario Use Case Diagram**

Skenario use case diagram digunakan untuk menjelaskan secara jelas kebutuhan fungsional suatu sistem dari use case diagram. Skenario use case diagram nantinya terdapat nama use case, pengguna atau aktor, tujuan use case, kondisi awal yang harus terpenuhi, skenario utama use case, skenario alternatif, dan kondisi akhir yang diharapkan. Berikut skenario use case diagram dapat dilihat dari tabel 4.3 sampai tabel

Tabel 4.3 Skenario use case login pada sistem

|  |  |
| --- | --- |
| Skenario use case login pada sistem | |
| Nama | Login sistem. |
| Pengguna atau Aktor | * Admin * Perbendaharaan * Sekertaris * Pengguna Jasa |
| Tujuan | Menjelaskan bagaimana pengguna  melakukan login pada sistem |
| Kondisi Awal | Sistem menampilkan halaman login |
| Skenario Utama (basic flow) | 1. Sistem menampilkan form register untuk pengguna jasa yang belum memiliki akun 2. Sistem menampilkan halaman login untuk admin, perbendaharaan dan sekertaris 3. Pengguna mengisi username dan password 4. Pengguna klik tombol login 5. Sistem cek username dan password 6. Sistem akan menampilkan   halaman dashboard |
| Skenario Alternatif (alternative flow) | Jika gagal login sistem akan  menampilkan pesan kesalahan dan halaman login |
| Kondisi Akhir | Jika berhasil login sistem  menampilkan halaman dashboard |

Tabel 4.4 Skenario use case melihat dashboard

|  |  |
| --- | --- |
| Skenario use case melihat dashboard | |
| Nama | Melihat dashboard |
| Pengguna atau Aktor | * Admin * Perbendaharaan * Sekertaris * Pengguna Jasa |
| Tujuan | Melihat dashboard pada sistem |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman  dashboard  2. Pengguna dapat melihat ringkasan data atau dokumen dan jumlah dari setiap dokumen baik dokumen yang diproses, diterima, dikembalikan |
| Skenario Alternatif (alternative flow) | Jika sistem gagal menampilkan dahsboard, maka pengguna dapat  refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  dashboard. |

Tabel 4.5 Skenario use case Mengecek data users

|  |  |
| --- | --- |
| Skenario use case Mengecek data users | |
| Nama | Mengecek data users |
| Pengguna atau Aktor | Admin |
| Tujuan | Dapat melihat data users pegawai dan pengguna jasa yang login. Dapat menambahakan users, mengedit data users, menghapus data users dan melakukan pencarian data users. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman dashboard 2. Pengguna memilih users pada   Menu side bar   1. Sistem menampilkan halaman users, terdapat tabel yang berisi role, nama, username, dan aksi. 2. Untuk menambahkan users, klik “Tambah users” pilih salah sattu kategori pegawai atau pengguna jasa, sistem akan menampilkan form tambah users. Untuk menyimpan data users pengguna harus mengklik tombol “Simpan”. Jika ingin membatalkkan maka pengguna klik tombol “Kembali”. Jika data yang kita inputkan berhasil maka akan muncul pesan “Berhasil menambahkan users dengan username …..” 3. Untuk mengedit data users pengguna dapat mengklik tombol edit yang memiliki icon pen bewarna biru pada tabel aksi. Sistem akan menampilkamn halaman form edit users. Jika ingin menyimpan data yang telah diedit maka pengguna harus klik tombol “Simpan”. Jika ingin membatalkan. Maka penggunma mengklik tombol “Kembali”. \ 4. Untuk menghapus data users, pengguna mengklik icon sampah pada tabel aksi, maka sistem akan menampilkan pesan “Apakah anda yakin akan menghapus atas nama…..” lalu klik tombol hapus maka sistem akan menghapus data yang akan dihapus. 5. Untuk mencari data users pengguna mangklik tombol search, maka sistem akan menampilkan form pencarian diatas tabel users dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 6. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | 1. Jika pengguna gagal melihat data users, maka pengguna dapat refresh ulang browser. 2. Jika pengguna gagal menambahkan data pada halaman form users maka sistem akan menampilkan pesan “Username sudah digunakan! Silahkan gunakan username lain!” Maka pengguna dapat menginputkan ulang data serta pastikan kondisi internet stabil. Lalu teliti kembali apakah data telah terisi secara lengkap atau belum. 3. Jika pengguna gagal mengedit data pada halaman form users maka sistem akan menampilkan pesan “Please fill out this field”. Pastikan kembali kondisi internet stabil dan data yang akan diedit sudah terisi secara lengkap. |
| Kondisi Akhir | Sistem akan menampilkan halaman data users. Pada halaman tersebut terdapat tabel yang berisi data users , form “Search”, dan tombol “Tambah users”. |

Tabel 4.6 Skenario use case menginput data satuan berat

|  |  |
| --- | --- |
| Skenario use case menginput data satuan berat | |
| Nama | Menginput data satuan berat |
| Pengguna atau Aktor | Admin |
| Tujuan | Dapat melihat data satuan berat, dapat menambahkan satuan berat, mengedit satuan berat, menghapus satuan berat dan melakukan pencarian satuan berat. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih satuan pada   Menu side bar   1. Sistem menampilkan halaman satuan, terdapat tabel yang berisi no, satuan dan aksi. 2. Untuk menambahkan satuan, klik “Tambah satuan” sistem akan menampilkan form tambah satuan. Untuk menambah data satuan pengguna harus mengklik tombol “Tambah”. Jika ingin membatalkkan maka pengguna klik tombol “Batal”. Jika data yang kita inputkan berhasil maka akan muncul pesan “Berhasil menambah satuan!” 3. Untuk mengedit data satuan pengguna dapat mengklik tombol edit yang memiliki icon pen bewarna biru pada tabel aksi. Sistem akan menampilkamn halaman form edit satuan. Jika ingin menyimpan data yang telah diedit maka pengguna harus klik tombol “Simpan”. Jika ingin membatalkan. Maka penggunma mengklik tombol “Batal”. 4. Untuk menghapus data satuan, pengguna mengklik icon sampah pada tabel aksi, maka sistem akan menampilkan pesan “Apakah anda yakin akan menghapus satuan …..” lalu klik tombol hapus maka sistem akan menghapus data yang akan dihapus. 5. Untuk mencari data satuan pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel satuan dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 6. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data satuan, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Satuan. |

Tabel 4.7 Skenario use case melihat semua data dokumen

|  |  |
| --- | --- |
| Skenario use case melihat semua data dokumen | |
| Nama | Melihat semua data dokumen |
| Pengguna atau Aktor | Admin |
| Tujuan | Dapat melihat semua dokumen yang telah masuk. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Dokumen All pada   Menu side bar   1. Sistem menampilkan halaman dokumen all, terdapat tabel yang berisi no, no. manifes, tanggal, details dan keterangan. 2. Untuk melihat details data dokumen yang masuk, maka pengguna klik icon “show” yang ada di tabel details. 3. Untuk mencari data dokumen, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data semua dokumen dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 4. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Dokumen All. |

Tabel 4.8 Skenario use case melihat dokumen diproses

|  |  |
| --- | --- |
| Skenario use case melihat semua dokumen diproses | |
| Nama | Melihat dokumen diproses |
| Pengguna atau Aktor | Admin |
| Tujuan | Dapat melihat semua dokumen yang diproses. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Dokumen diproses pada Menu side bar 2. Sistem menampilkan halaman dokumen diproses, terdapat tabel yang berisi no, no. manifes, tanggal, details dan keterangan. 3. Untuk melihat details data dokumen yang masuk, maka pengguna klik icon “show” yang ada di tabel details. 4. Untuk mencari data dokumen, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data semua dokumen dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Dokumen diproses. |

Tabel 4.9 Skenario use case melihat dokumen diterima

|  |  |
| --- | --- |
| Skenario use case melihat semua dokumen diterima | |
| Nama | Melihat dokumen diterima |
| Pengguna atau Aktor | Admin |
| Tujuan | Dapat melihat semua dokumen yang diterima. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Dokumen diterima 2. Sistem menampilkan halaman dokumen diterima, terdapat tabel yang berisi no, no. manifes, tanggal, details dan keterangan. 3. Untuk melihat details data dokumen yang masuk, maka pengguna klik icon “show” yang ada di tabel details. 4. Untuk mencari data dokumen, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data semua dokumen dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Dokumen diterima. |

Tabel 4.10 Skenario use case melihat dokumen dikembalikan

|  |  |
| --- | --- |
| Skenario use case melihat semua dokumen dikembalikan | |
| Nama | Melihat dokumen dikembalikan |
| Pengguna atau Aktor | Admin |
| Tujuan | Dapat melihat semua dokumen yang dikembalikan. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Dokumen dikembalikan pada Menu side bar 2. Sistem menampilkan halaman dokumen dikembalikan, terdapat tabel yang berisi no, no. manifes, tanggal, details dan keterangan. 3. Untuk melihat details data dokumen yang masuk, maka pengguna klik icon “show” yang ada di tabel details. 4. Untuk mencari data dokumen, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data semua dokumen dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Dokumen dikembalikan. |

Tabel 4.11 Skenario use case dokumen baru

|  |  |
| --- | --- |
| Skenario use case melihat semua dokumen baru. | |
| Nama | Melihat dokumen baru |
| Pengguna atau Aktor | * Perbendaharaan * Sekertaris |
| Tujuan | Dapat melihat semua dokumen baru. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Dokumen baru pada Menu side bar 2. Sistem menampilkan halaman dokumen baru, terdapat tabel yang berisi no. manifes, tanggal, details dan keterangan. 3. Untuk melihat details data dokumen baru yang masuk, maka pengguna klik icon “show” yang ada di tabel details. 4. Untuk mencari data dokumen baru, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data dokumen baru dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Dokumen baru. |

Tabel 4.12 Skenario use case melihat data penambahan pos

|  |  |
| --- | --- |
| Skenario use case melihat data penambahan pos | |
| Nama | Melihat data penambahan pos |
| Pengguna atau Aktor | Perbendaharaan |
| Tujuan | Dapat melihat semua data di penambahan pos. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih penambahan pos pada Menu side bar 2. Sistem menampilkan halaman penambahan pos, terdapat tabel yang berisi no, nama perusahaan, no. manifes, tanggal, status dan aksi. 3. Untuk melihat aksi data penambahan pos, maka pengguna klik icon “mata yang ada di tabel aksi. 4. Untuk mencari data penambahan pos, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data penambahan pos dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Data penambahan pos. |

Tabel 4.14 Skenario use case memeriksa pembatalan pos

|  |  |
| --- | --- |
| Skenario use case memeriksa penambatalan pos | |
| Nama | Memeriksa penambahan pos |
| Pengguna atau Aktor | Perbendaharaan. |
| Tujuan | Dapat memeriksa penambahan pos yang sudah masuk. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih penambahan pos pada Menu side bar 2. Sistem menampilkan halaman penambahan pos, terdapat tabel yang berisi no, nama perusahaan, no.manifes, tanggal, status dan aksi. 3. Untuk melihat aksi data dokumen yang masuk, maka pengguna klik icon “mata” yang ada di tabel aksi, kemudian memilih status dokumen diterima yang ada di form detail pos manifes. 4. Untuk mencari data dokumen, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data semua dokumen dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Penambahan pos. |

Tabel 4.13 Skenario use case memeriksa penambahan pos

|  |  |
| --- | --- |
| Skenario use case memeriksa penambahan pos | |
| Nama | Memeriksa penambahan pos |
| Pengguna atau Aktor | Perbendaharaan |
| Tujuan | Dapat memeriksa penambahan pos yang sudah masuk. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih penambahan pos pada Menu side bar 2. Sistem menampilkan halaman penambahan pos, terdapat tabel yang berisi no, nama perusahaan, no.manifes, tanggal, status dan aksi. 3. Untuk melihat aksi data dokumen yang masuk, maka pengguna klik icon “mata” yang ada di tabel aksi, kemudian memilih status dokumen diterima yang ada di form detail pos manifes. 4. Untuk mencari data dokumen, pengguna klik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data semua dokumen dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel. |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data dokumen, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  Penambahan pos. |

Tabel 4.14 Skenario Use Case Mengirim Pengumuman Ke Pengguna Jasa

|  |  |
| --- | --- |
| Skenario use case Mengirim Pengumuman Ke Pengguna Jasa | |
| Nama | Mengirim Pengumuman Ke Pengguna Jasa |
| Pengguna atau Aktor | Sekertaris |
| Tujuan | Dapat mengirim pengumuman ke pengguna jasa |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Pengumuman pada Menu side bar 2. Sistem menampilkan halaman dokumen baru, terdapat tabel yang berisi no, judul, Tanggal, Lampiran, Aksi. 3. Untuk mengirim pengumuman pengguna bisa klik “Tambah Pengumuman” dan penguna mengisi form “Tambah Pengumuman” dan mengisi form yang di dalam berisikan Judul, Isi Pengumuman, Tipe Pengumuman,dan Lampiran 4. Untuk menyimpan data users pengguna harus mengklik tombol “Simpan”. Jika ingin membatalkkan maka pengguna klik tombol “Kembali”. Jika data yang kita inputkan berhasil maka akan muncul pesan “Berhasil menambahkan pengumuman” 5. Untuk mengedit data pengumumangklik tombol edit yang memiliki icon pen bewarna biru pada tabel aksi. Sistem akan menampilkamn halaman form edit pengumuman. Jika ingin menyimpan data yang telah diedit maka pengguna harus klik tombol “Simpan”. Jika ingin membatalkan. Maka penggunma mengklik tombol “Kembali”. 6. Untuk menghapus data users, pengguna mengklik icon sampah pada tabel aksi lalu klik tombol hapus maka sistem akan menghapus data yang akan dihapus. 7. Untuk mencari data pengumuman pengguna mengklik tombol search, maka sistem akan menampilkan form pencarian diatas tabel users dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 8. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel |
| Skenario Alternatif (alternative flow) | Jika pengguna gagal melihat data pengumuman, maka pengguna dapat refresh ulang browser. |
| Kondisi Akhir | Sistem menampilkan halaman  pengumuman. |

Tabel 4.15 Skenario Use Case Menginput data Pembatalan Manifes

|  |  |
| --- | --- |
| Skenario use case Menginput data Pembatalan Manifes | |
| Nama | Menginput data Pembatalan  Manifes |
| Pengguna atau Aktor | Pengguna Jasa |
| Tujuan | Dapat menginput pembatalan manifes |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Pembatalan manifes pada Menu side bar 2. Sistem menampilkan halaman pembatalan manifes yang berisi form input berkas yang berisi no.manifes, tanggal, lampiran dan persetujuan. 3. Untuk mengirim pembatalan manifes pengguna bias mengeklik “submit”. 4. Untuk mencari data pengumuman pengguna mengklik tombol search, maka sistem akan menampilkan form pencarian diatas tabel pembatalan manifes dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 5. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel |
| Kondisi Akhir | Sistem menampilkan halaman input berkas. |

Tabel 4.16 Skenario Use Case Menginput data Penambahan pos

|  |  |
| --- | --- |
| Skenario use case Menginput data Penambahan pos | |
| Nama | Menginput data Penambahan pos |
| Pengguna atau Aktor | Pengguna Jasa |
| Tujuan | Pengguna dapat mengirim penambahan pos |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Penambahan pos pada Menu side bar 2. Sistem menampilkan form data penambahan pos yang berisi tabel no, nama perusahaan, no.manifes, tanggal, status, dan aksi. 3. Untuk membuat penambahan pos, pengguna dapat klik “Tambah Manifes” yang berisi no.manifes, tanggal, lampiran dan persetujuan. 4. Untuk menginput pos baru, pengguna mengeklik icon “tambah” pada kolom aksi, kemudian sistem menampilkan form tambah pos baru yang berisi no.manifes, no.pos. no.BL, NPWP Shipper, nama&alamat consigne, nama&alamat consnsigne, uraian barang, jumlah jenis kemasan, satuan, POL/PON, no.BC 1.1 dan tanggal. 5. Untuk menginput data pos baru pengguna dapat mengeklik “submit” pada form tambah pos baru. 6. Jika pengguna telah mengisi tambah manifes dan menginput pos baru, pengguna dapat mengirim penambahan pos tersebut ke perbendaharaan dengan mengeklik icon pesawat di dalam kolom “akses”, kemudian sistem akan menampilkan pesan “sistem berhasil mengirim manifes” 7. Untuk melihat detail penambahan pos, pengguna bias mengeklik icon mata yang ada di kolom aksi, dan sistem akan menampilkan detail pos manifes yang masih dalam status diproses. 8. Untuk mencari data pengumuman pengguna mengklik tombol search, maka sistem akan menampilkan form pencarian diatas tabel data penambahan pos dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 9. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel |
| Kondisi Akhir | Sistem menampilkan halaman input pos baru. |

Tabel 4.17 Skenario Use Case Melihat Pengumuman dari Sekretaris

|  |  |
| --- | --- |
| Skenario use case Melihat Pengumuman dari Sekretaris | |
| Nama | Melihat Pengumuman dari  Sekretaris |
| Pengguna atau Aktor | Pengguna Jasa |
| Tujuan | Pengguna dapat melihat pengumuman dari sekretaris. |
| Kondisi Awal | Pengguna berhasil login dan sistem  menampilkan halaman dashboard |
| Skenario Utama (basic flow) | 1. Sistem menampilkan halaman   Dashboard   1. Pengguna memilih Pengumuman pada Menu side bar 2. Pengguna dapat mengeklik menu pengumuman pada menu side bar. 3. Sistem menampilkan tabel pengumuman yang berisi no, judul, isi, tanggal, dan lampiran. 4. Untuk dapat melihat lampiran, pengguna dapat mengeklik icon show pada kolom “lampiran” dan sistem akan menampilkan preview dokumen lampiran dan pengguna dapat mendownload lampiran tersebut dengan klik button download. 5. Untuk mencari data pengumuman pengguna mengklik tombol search, maka sistem akan menampilkan form data pengumuman dan pengguna dapat mencari data sesuai kebutuhan. Sistem akan menampilkan data yang akan dicari. 6. Untuk menyimpan data, pengguna dapat mengeksport data dalam bentuk csv, pdf dan excel |
| Kondisi Akhir | Sistem menampilkan halaman input pos baru. |

**BAB V**

**PENUTUP**

### Kesimpulan

Dari hasil analisa dan perancangan Sistem Informasi Perbaikan Data (Redress) Manifes Inward dan Outward Kepabeanan, maka dapat disimpulkan sebagai berikut :

* + 1. Sistem ini dirancang menggunakan metode agile development system, terdapat beberapa tahapan mulai dari analisis permasalahan, perencanaan sistem, implementasi, pengujian sistem, pengaplikasian dan perawatan. Kelebihan menggunakan metode ini yaitu tingkat kegagalan relatif rendah, apabila terjadi kegagalan nilai kerugian dari segi material maupun non-material tidak terlalu besar, dan pengembangan sistem lebih cepat.
    2. Sistem ini dibangun dengan bahasa pemrograman PHP sehingga sistem menjadi lebih dinamis dan mudah diletakkan didalam HTML. Sistem informasi Perbaikan data (Redress) Manifes juga dibangun menggunakan MySQL sebagai database-nya, kelebihan menggunakan MySQL salah satunya memiliki performa tinggi sehingga dapat diakses dengan dengan mudah dan kecepatan optimal.

### Saran

Adapun saran yang dapat diberikan agar sistem informasi inventaris desa dan pencatatan pajak bumi bangunan ini menjadi lebih baik dimasa mendatang adalah sebagai berikut :

* + 1. Pengembangan berikutnya sistem ini dapat dikembangkan di Kantor Bea dan Cukai Kota Tarakan saja untuk sekarang karena sistem ini tidak mudah langsung untuk diberikan ke pusat Dorektorat Bea dan Cukai.
    2. Perlu adanya pengembangan fitur lebih lanjut dari sistem, sehingga menghasilkan fitur-fitur terbaru yang semakin mempermudah Perbaikan data (Redress).

**DAFTAR PUSTAKA**

Adani, M. R. (2020, Agustus 8). *Penjelasan Lengkap Metode Agile dalam Pengembangan Perangkat Lunak*. Retrieved from Sekawan Media: https://www.sekawanmedia.co.id/blog/metode-agile-development/

Admin. (2013, January 21). *Mengenal phpMyAdmin Dan Kegunaannya*. Retrieved from termasmedia: https://www.termasmedia.com/database/phpmyadmin/131-mengenal-phpmyadmin-dan-kegunaannya.html

Arlina. (2016). *Pengertian Database MySQL*. Retrieved from IDMySQL: http://idmysql.blogspot.com/2016/07/pengertian-database-MySQL.html

educhannel. (2016, Juli 13). *Pengertian CSS*. Retrieved from educhannel: https://educhannel.id/blog/artikel/pengertian-css.html

EMAS, K. T. (2015, June 23). *Redress Inward/Outward Manifes*. Retrieved from KPPBC TMP TANJUNG EMAS: https://bctemas.beacukai.go.id/faq/redress-inwardoutward-manifes/

GriyaWebsite. (2020, Mei 3). *APA ITU VISUAL STUDIO CODE*. Retrieved from GriyaWebsite: https://www.griyawebsite.com/mengenal-apa-itu-visual-studio-code-dan-bahasa-pemrograman/

Hutauruk, M. K. (2017, Agustus 2). *UML Diagram : Use Case Diagram*. Retrieved from Bina Nusantara: https://socs.binus.ac.id/2019/11/26/uml-diagram-use-case-diagram/

Indonesia, L. S. (2021, June 14 ). *Fungsi HTML (HyperText Markup Language)*. Retrieved from LMS SPADA Indonesia: https://lmsspada.kemdikbud.go.id/mod/page/view.php?id=57743

INTERMEDIA, B. (2019, 14, 14). *Pengertian PHP, Fungsi, Syntax & Contoh Penulisan yang Benar*. Retrieved from Jagoan Hosting: https://www.jagoanhosting.com/blog/pengertian-php/

Junaedi, N. L. (2021, October 11). *Sequence diagram: Pengertian, manfaat, 14 komponen, dan contohnya*. Retrieved from Ekrut Media: https://www.ekrut.com/media/sequence-diagram-adalah

Junaedi, N. L. (2021, October 11). *Sequence diagram: Pengertian, manfaat, 14 komponen, dan contohnya*. Retrieved from Ekrut Media: https://www.ekrut.com/media/sequence-diagram-adalah

Junaedi, N. L. (2022, January January ). *Metode Agile Development: Pengertian, Tujuan, dan Keunggulan*. Retrieved from Ekrut Media: https://www.ekrut.com/media/metode-agile-adalah

Laila, G. (2022, Januari 26). *PHP – Pengertian, Fungsi, Kelebihan dan Kekurangannya*. Retrieved from 4winmobile: https://4winmobile.com/php/

Laily, I. N. (2022, Februari 7 ). *Pengertian Website Menurut Para Ahli, Beserta Jenis dan Fungsinya*. Retrieved from Dkatadata: https://katadata.co.id/safrezi/berita/6200a2a9697ec/pengertian-website-menurut-para-ahli-beserta-jenis-dan-fungsinya

Lathif, R. (2019, September 05). *Apa Itu Draw.io ? Ini Penjelasan Lengkapnya*. Retrieved from SURGATEKNO: https://surgatekno.com/tech-news/apa-itu-draw-io/

litalia. (2019, Maret 13). *Pengertian MySQL, Fungsi dan Sejarah MySQL Lengkap !!* Retrieved from Jurnalponsel: https://www.jurnalponsel.com/pengertian-mysql/#

Mecha, S. (2014, Agustus). *Pengertian ERD (Entity Relationship Diagram)*. Retrieved from Satria Mecha: http://satriamecha.blogspot.com/2014/08/pengertian-erd-entity-relationship.html

Mulachela, H. (2021, Desember 21). *Database Adalah: Pengertian dan Jenisnya*. Retrieved from DKATADATA: https://katadata.co.id/intan/digital/61c04e3f62f5b/database-adalah-pengertian-dan-jenisnya

Musyaffa, I. (2020, April 22). *Metode Pengembangan Agile : Pengertian, Tahapan, dan Keuntungan*. Retrieved from fittechinova: https://fittechinova.com/blog/detail/metode-pengembangan-agile-pengertian-tahapan-dan-keuntungan

Nugroho, A. (2019, December 8 ). *Apa Itu Bootstrap? Inilah Pengertian dan Cara Menggunakannya*. Retrieved from Qwords: https://qwords.com/blog/apa-itu-bootstrap/

Nugroho, A. (2019, September 19). *Pengertian XAMPP* . Retrieved from Qwords: https://qwords.com/blog/pengertian-xampp/

Ok, T. (2017, Januari 16). *Pengujian Black Box | Teknik Pengujian Sistem*. Retrieved from Gudang Ilmu Komputer: https://koleksibukukuliah.blogspot.com/2017/01/pengujian-black-box-teknik-pengujian.html

Pasha, A. R. (2019, Februari 26). *Bea Cukai: Pengertian, Fungsi dan Kebijakan yang Penting Diketahui*. Retrieved from cermati: https://www.cermati.com/artikel/bea-cukai-pengertian-fungsi-dan-kebijakan-yang-penting-diketahui

Ramadhani, N. (2020, febuari 28). *Ini Dampak Perkembangan Teknologi yang Dapat Dirasakan*. Retrieved from akseleran: https://www.akseleran.co.id/blog/perkembangan-teknologi/

Ramadhanti. (2021, November 1). *Pengertian UML*. Retrieved from pinhome: https://www.pinhome.id/blog/pengertian-uml/

Tanoto, U. (2020, Desember 29). *Activity Diagram: Pengertian, Fungsi, Contoh serta Cara Membuatnya*. Retrieved from Jojonomic: https://www.jojonomic.com/blog/activity-diagram/

tkjpnup. (2013, 12). *Black Box Testing Dan White box Testing*. Retrieved from tkjpnup: http://tkjpnup.blogspot.com/2013/12/black-box-testing-dan-white-box-testing.html

Wikipedia. (2022, Februari 1). *Bahasa pemrograman*. Retrieved from Wikipedia: https://id.wikipedia.org/wiki/Bahasa\_pemrograman