# SKRIPSI

**PERANCANGAN SISTEM INFORMASI PENDATAAN WARGA PERUMAHAN DUTA ASRI CIAKAR BERBASIS *WEBSITE* MENGGUNAKAN METODE *WATERFALL***

******

**Disusun oleh :**

|  |  |  |
| --- | --- | --- |
| **NAMA** | **:** | **ADITIA INDRA PERMANA** |
| **NPM** | **:** | **2021804087** |
| **PROGRAM STUDI** | **:** | **SISTEM INFORMASI** |

**Untuk Memenuhi Sebagian Dari Syarat- syarat**

**Guna Mencapai Gelar Sarjana Komputer**

UNIVERSITAS INSAN PEMBANGUNAN INDONESIA

**Jl. Raya Serang Km. 10 Bitung - Tangerang**

***Website* :** http://www.unipem.ac.id

**Email :** [info@unipem.ac.id](mailto:info@unipem.ac.id)

**Telp . (021) 59492836**

**Fax. (021) 594928837**

**Th. Akademik 2023/2024**

# DAFTAR ISI

Halaman

[SKRIPSI i](#_Toc201333759)

[DAFTAR ISI i](#_Toc201333760)

[DAFTAR TABEL iii](#_Toc201333761)

[DAFTAR GAMBAR v](#_Toc201333762)

[BAB I PENDAHULUAN 1](#_Toc201333763)

[1.1 Latar Belakang Masalah 1](#_Toc201333764)

[1.2 Identifikasi Masalah 4](#_Toc201333765)

[1.3 Ruang lingkup / Batasan Masalah 4](#_Toc201333766)

[1.4 Perumusan Masalah 4](#_Toc201333767)

[1.5 Tujuan Penelitian 5](#_Toc201333768)

[1.6 Manfaat Penelitian 5](#_Toc201333769)

[BAB II LANDASAN TEORI 7](#_Toc201333770)

[2.1 Tinjauan Pustaka 7](#_Toc201333772)

[2.1.1 Pengertian Perancangan 7](#_Toc201333773)

[2.1.2 Pengertian Sistem 8](#_Toc201333774)

[2.1.3 Definisi Sistem Informasi 14](#_Toc201333775)

[2.1.4 Definisi Pendataan 17](#_Toc201333776)

[2.1.5 Definisi Warga 18](#_Toc201333777)

[2.1.6 Sejarah *Website* 18](#_Toc201333778)

[2.1.7 Definisi *Website* 19](#_Toc201333779)

[2.1.8 Jenis-jenis *Website* 20](#_Toc201333780)

[2.2 Penelitian Sebelumnya / Relevan 21](#_Toc201333781)

[2.3 *Tools* yang digunakan 22](#_Toc201333782)

[2.3.1 UML (*Unified Modelling Language)* 22](#_Toc201333785)

[2.3.2 PHP (*Hypertext Preprocessor)* 30](#_Toc201333786)

[2.3.3 *HTML* 31](#_Toc201333787)

[2.3.4 *MySQL* 31](#_Toc201333788)

[2.3.5 *XAMPP* 32](#_Toc201333789)

[2.3.6 Black box Testing 33](#_Toc201333790)

[2.4 Kerangka Kerja Teoritis 35](#_Toc201333791)

[BAB III METODOLOGI PENELITIAN 36](#_Toc201333792)

[3.1 Desain Penelitian 36](#_Toc201333794)

[3.1.1 Tahap Penelitian 37](#_Toc201333795)

[3.2 Data dan Sumber Data 38](#_Toc201333796)

[3.2.1 Jenis Data 38](#_Toc201333800)

[3.2.2 Sumber Data 38](#_Toc201333801)

[3.2.3 Tempat Penelitian 39](#_Toc201333802)

[3.2.4 Waktu Penelitian 39](#_Toc201333803)

[3.3 Metode Pengumpulan Data 39](#_Toc201333804)

[3.3.1 Metode Observasi 39](#_Toc201333806)

[3.3.2 Metode Wawancara 40](#_Toc201333807)

[3.3.3 Metode Studi Pustaka 41](#_Toc201333808)

[3.4 Metode Analisis Data 41](#_Toc201333809)

[3.5 Metode Perancangan / Pengembangan Sistem 42](#_Toc201333810)

[3.7.1 Analisis Kebutuhan Sistem 43](#_Toc201333813)

[3.7.2 Desain Diagram yang Digunakan 44](#_Toc201333814)

[3.7.3 Software yang Digunakan 46](#_Toc201333815)

[3.7.4 Metode Pengujian Sistem yang Digunakan 46](#_Toc201333816)

[BAB 4 47](#_Toc201333817)

[PEMBAHASAN 47](#_Toc201333818)

[4.1 Analisa Sistem Yang Bejalan 47](#_Toc201333821)

[4.1.1 Latar Belakang Objek Penlitian 47](#_Toc201333822)

[4.1.2 Visi Dan Misi 47](#_Toc201333823)

[4.1.3 Struktur Organisasi dan Uraian Tugas 48](#_Toc201333824)

[4.1.4 Analisa Sistem Yang Berjalan 50](#_Toc201333825)

[4.1.5 Masalah Yang Terjadi 53](#_Toc201333826)

[4.1.6 Aternatif Pemecahan Masalah 54](#_Toc201333827)

[4.2 Rancangan Sistem yang Diusulkan 55](#_Toc201333828)

[4.2.1 *Use Case Diagram* yang Diusulkan 55](#_Toc201333830)

[4.2.2 *Activity Diagram* yang Diusulkan 63](#_Toc201333831)

[4.2.3 *Sequence Diagram* yang Diusulkan 69](#_Toc201333832)

[4.2.4 *Class Diagram* yang Diusulkan 76](#_Toc201333833)

[4.2.5 Rancangan Basis Data 77](#_Toc201333834)

# DAFTAR TABEL

[TABEL 2. 1 Penelitian Sebelumnya/Relevan 21](#_Toc199705953)

[TABEL 2. 2 Simbol Use Case Diagram 23](#_Toc199705954)

[TABEL 2. 3 Simbol-Simbol Activity Diagram 24](#_Toc199705955)

[TABEL 2. 4 Simbol Squence Diagram 27](#_Toc199705956)

[TABEL 2. 5. Simbol Class Diagram 29](#_Toc199705957)

[TABEL 2. 6 Simbol Indicator Relation Class Diagram 29](#_Toc199705958)

[TABEL 3. 1 Tahap Penelitian 37](#_Toc198932235)

[TABEL 3. 2 Waktu Penelitian 2025 39](#_Toc198932236)

[TABEL 4. 1 Skenario Use Case Diagram Yang Berjalan 51](#_Toc201328103)

[TABEL 4. 2 Definisi Use Case 57](#_Toc201328104)

[TABEL 4. 3 Skenario Use Case Login 59](#_Toc201328105)

[TABEL 4. 4 Skenario Use Case Tambah Kas 60](#_Toc201328106)

[TABEL 4. 5 Skenario Use Case Ubah Kas 60](#_Toc201328107)

[TABEL 4. 6 Skenario Use Case Hapus Kas 61](#_Toc201328108)

[TABEL 4. 7 Skenario Use Case Tambah Warga 61](#_Toc201328109)

[TABEL 4. 8 Skenario Use Case Ubah Warga 61](#_Toc201328110)

[TABEL 4. 9 Skenario Use Case Hapus Warga 62](#_Toc201328111)

[TABEL 4. 10 Skenario Use Case Akses User 62](#_Toc201328112)

[TABEL 4. 11 Skenario Use Case Laporan Kas 63](#_Toc201328113)

[TABEL 4. 12 Skenario Use Case Laporan Sampah 63](#_Toc201328114)

# DAFTAR GAMBAR

[GAMBAR 2. 1 Sistem Terbuka 9](#_Toc201327387)

[GAMBAR 2. 2 Sistem Tertutup 9](#_Toc201327388)

[GAMBAR 2. 3 Karakteristik Sistem 11](#_Toc201327389)

[GAMBAR 2. 4 Siklus Pengolahan Data 13](#_Toc201327390)

[GAMBAR 2. 5 Kerangka Kerja Teoritis 35](#_Toc201327391)

[GAMBAR 4. 1 Struktur Organisasi 48](#_Toc201333835)

[GAMBAR 4. 2 Activity Diagram Yang Sedang Berjalan 50](#_Toc201333836)

[GAMBAR 4. 3 Activity Diagram Sistem Pendataan Warga Yang Berjalan 53](#_Toc201333837)

[GAMBAR 4. 4 Use Case Diagram yang Diusulkan 55](#_Toc201333838)

[GAMBAR 4. 5 Activity Diagram Login 63](#_Toc201333839)

[GAMBAR 4. 6 Activity Diagram Tambah Warga 64](#_Toc201333840)

[GAMBAR 4. 7 Activity Diagram Ubah Warga 65](#_Toc201333841)

[GAMBAR 4. 8 *Activity Diagram* Hapus Warga 66](#_Toc201333842)

[GAMBAR 4. 9 *Activity Diagram* Laporan Kas 66](#_Toc201333843)

[GAMBAR 4. 10 *Activity Diagram* Ubah Kas 67](#_Toc201333844)

[GAMBAR 4. 11 *Activity Diagram* Tambah Kas 68](#_Toc201333845)

[GAMBAR 4. 12 *Activity Diagram* Hapus Kas 68](#_Toc201333846)

[GAMBAR 4. 13 *Sequence Diagram Login* 69](#_Toc201333847)

[GAMBAR 4. 14 *Sequence Diagram* Tambah Warga 70](#_Toc201333848)

[GAMBAR 4. 15 *Sequence Diagram Ubah Warga* 71](#_Toc201333849)

[GAMBAR 4. 16 *Sequence Diagram* Hapus Warga 72](#_Toc201333850)

[GAMBAR 4. 17 *Sequence Diagram* Tambah Kas 74](#_Toc201333851)

[GAMBAR 4. 18 *Sequence Diagram* Ubah Kas 75](#_Toc201333852)

[GAMBAR 4. 19 *Sequence Diagram* Hapus Kas 76](#_Toc201333853)

[GAMBAR 4. 20 *Class Diagram* yang Diusulkan 76](#_Toc201333854)

# BAB I PENDAHULUAN

## Latar Belakang Masalah

Teknologi terus menjadi pendorong utama transformasi dalam kehidupan modern. Penelitian teknologi fokus pada konsep-konsep, desain, dan implementasi inovasi yang mempercepat kemajuan masyarakat. Salah satu sektor yang secara signifikan terpengaruh oleh perkembangan teknologi adalah teknologi informasi. Dengan menggunakan teknologi informasi, organisasi dapat meningkatkan efisiensi dan efektivitas operasional, serta mengurangi biaya, waktu, dan energi yang dibutuhkan. Perkembangan teknologi informasi juga dapat membantu perkembangan teknologi di bidang lain, termasuk teknologi yang diterapkan dalam bidang administrasi.

Teknologi informasi dan komunikasi telah menyebar ke berbagai bidang, baik yang disadari maupun tidak oleh masyarakat. Seluruh kalangan masyarakat, termasuk petugas administrasi perumahan seperti RT, RW, atau ketua komplek, selalu menuntut kemudahan dan efisiensi dalam menyelesaikan pekerjaan. Penggunaan teknologi dapat meningkatkan efektivitas dan efisiensi dalam pekerjaan petugas administrasi perumahan. Organisasi juga membutuhkan teknologi informasi dan komunikasi untuk memudahkan kinerja dan membuat sistem bekerja dengan lebih mudah dan cepat.

Kemajuan teknologi informasi dan komunikasi telah membawa dampak yang cukup signifikan dalam berbagai aspek kehidupan, termasuk dalam pendataan warga komplek perumahan. Di era digital ini, pentingnya efisiensi dan kecepatan dalam mengelola informasi telah menjadi suatu kebutuhan yang tak terelakkan. Salah satu aspek yang menjadi perhatian dalam pengelolaan perumahan adalah sistem pendataan warga perumahan.

Pengelolaan iuran warga merupakan aspek penting dalam keberlangsungan dan kesejahteraan lingkungan tempat tinggal. Namun, dalam praktiknya, masih banyak kendala yang dihadapi, seperti kesulitan dalam pencatatan manual yang rentan terhadap kesalahan. Oleh karena itu, perancangan sebuah sistem informasi pendataan warga perumahan Duta Asri Ciakar melalui *Website* menjadi solusi yang relevan untuk mengatasi kendala tersebut.

Saat ini, ada lebih dari 100 kartu keluarga yang dimiliki warga Perumahan Duta Asri Ciakar, dan setiap rumah pasti mempunyai data sendiri. Pada Perumahan Duta Asri Ciakar, pendataan warga dilakukan secara langsung yang dilakukan oleh ketua RT, yang kemudian dicatat dan disimpan. Akibatnya, proses pengumpulan dan pendataan warga di Perumahan Duta Asri Ciakar membutuhkan waktu lebih dari satu hari, yang menyebabkan proses pengolahan data menjadi lebih lama.

Dari permasalahan tersebut dibutuhkan solusi yaitu sistem pendataan warga. Sistem pendataan warga adalah prosedur yang digunakan oleh pengelola perumahan untuk mengatur, mengawasi, dan mengelola segala sesuatu yang berkaitan dengan data warga perumahan. Sistem ini melibatkan pembuatan, pemeliharaan, dan pembaharuan informasi mengenai pendataan yang terkait dengan warga perumahan.

Dalam konteks tersebut, pengembangan sebuah sistem pendataan warga melalui *Website* menjadi solusi yang relevan dan penting. Sistem ini diharapkan dapat memberikan kemudahan akses, serta efisiensi dalam proses pengumpulan dan pendataan warga.

Dengan mengikuti tahapan secara berurutan seperti ini, diharapkan penelitian dapat mencapai hasil yang terukur dan dapat diandalkan dalam meningkatkan efisiensi administrasi komplek perumahan serta memperkuat partisipasi aktif warga dalam pengelolaan lingkungan tempat tinggal mereka. Dan dapat dikembangkan suatu model atau sistem yang mampu mengintegrasikan proses pendataan warga perumahan secara komprehensif melalui *Website*. Metode *Waterfall* juga memungkinkan untuk melakukan evaluasi secara sistematis terhadap progres penelitian dan mengidentifikasi perbaikan yang diperlukan pada setiap tahapan pengembangan sistem, sehingga dapat meningkatkan kualitas manajemen komplek perumahan serta memperkuat partisipasi aktif warga dalam pengelolaan lingkungan tempat tinggal mereka.

Berdasarkan permasalahan diatas, maka penulis melakukan penelitian dengan mengambil judul skripsi “Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar Berbasis *Website* Menggunakan Metode *Waterfall*”.

## Identifikasi Masalah

Berdasarkan latar belakang permasalahan maka identifikasi masalahnya sebagai berikut:

* + 1. Pendataan warga baru membutuhkan waktu yang lama.
    2. Berkas yang menumpuk menyebabkan penyimpanan berkas penuh dan pencarian data warga membutuhkan waktu yang lama.
    3. Risiko kesalahan data dalam pencatatan manual.
    4. Kurang aktifnya partisipasi warga.

## Ruang lingkup / Batasan Masalah

Berikut adalah batasan masalah dalam penelitian ini:

* + 1. Rancangan sistem yang sedang berjalan hanya untuk mendata warga
    2. Rancangan sistem ini hanya dilakukan pada Perumahan Duta Asri Ciakar.
    3. Membuat laporan data warga dan administrasi keuangan warga

## Perumusan Masalah

Berdasarkan permasalahan yang ada, perumusan masalah dalam penelitian ini adalah sebagai berikut:

* + 1. Bagaimana cara merancang sistem pendataan warga Perumahan Duta Asri Ciakar yang sedang berjalan saat ini?
    2. Bagaimana merancang sistem pengelolaan data warga Perumahan Duta Asri Ciakar secara efektif melalui *Website*?
    3. Bagaimana cara warga Perumahan Duta Asri Ciakar menambahkan data di *Website*?

## Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah sebagai berikut :

* + 1. Mengidentifikasi dan mengevaluasi mekanisme pendataan warga yang saat ini digunakan di Perumahan Duta Asri Ciakar, guna memahami kekuatan dan kelemahannya sebagai dasar dalam merancang sistem pendataan yang lebih efektif, efisien.
    2. Memastikan akses yang mudah bagi pengguna dalam proses pendataan warga Perumahan Duta Asri Ciakar melalui *Website*.
    3. Mengurangi kesalahan penulisan dan penumpukan berkas dalam pendataan warga melalui *Website*.

## Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

* + 1. Bagi Penulis
       1. Memperoleh pemahaman mendalam tentang pengembangan sistem pendataan warga melalui *Website*.
       2. Meningkatkan keterampilan dalam menerapkan metodologi pengembangan perangkat lunak, khususnya metode *Waterfall*, dalam konteks pendataan warga perumahan.
       3. Menambah pengalaman praktis dalam merancang, mengembangkan, dan menguji sistem informasi berbasis *web*.
       4. Membuka peluang untuk berkontribusi pada peningkatan efisiensi dan efektivitas administrasi komplek perumahan.
       5. Mempersiapkan diri untuk karier di bidang teknologi informasi atau manajemen perumahan dengan memiliki keterampilan yang relevan dalam pengembangan sistem dan pengelolaan data.
    2. Bagi Universitas Insan Pembangunan
       1. Sebagai bahan referensi bagi pembaca maupun peneliti selanjutnya dengan topik permasalahan yang sama,
       2. Dengan penelitian ini diharapkan menjadi tambahan pengetahuan bagi para pembacanya
       3. Memperluas jangkauan akademik kampus dengan menerapkan penelitian yang relevan dengan kebutuhan masyarakat dan industri di bidang teknologi informasi.
    3. Bagi Warga
       1. Memungkinkan warga memperoleh informasi penting secara cepat dan terpercaya, seperti pengumuman pemerintahan, program bantuan sosial, hingga jadwal vaksinasi.
       2. Membantu warga untuk membangun jaringan, mempromosikan UMKM, mengadakan kegiatan sosial, dan memperkuat solidaritas antarwarga.
       3. *Website* menyediakan ruang untuk laporan keuangan, proyek pembangunan, dan agenda publik.

# BAB II LANDASAN TEORI



## Tinjauan Pustaka

### Pengertian Perancangan

Menurut Mardian et al (2021:78) perancangan didefinisikan sebagai proses untuk mendefinisikan suatu model atau rancangan perangkat lunak dengan menggunakan teknik dan prinsip tertentu sedemikian sehingga modelatau rancangan tersebut dapat diwujudkan menjadi perangkat lunak.

Perancangan adalah proses merencanakan atau merancang sesuatu sebelum dilaksanakan. Ini dapat mencakup mengidentifikasi masalah atau kebutuhan, mengumpulkan informasi, menganalisis, membuat ide baru, dan membuat solusi yang efektif. Tujuannya adalah membuat sesuatu yang bermanfaat, efektif, dan menarik yang memenuhi kebutuhan pengguna atau pemangku kepentingan.

Perancangan merupakan tahapan krusial dalam pengembangan suatu produk atau sistem yang memerlukan pendekatan yang sistematis dan terencana. Proses perancangan melibatkan identifikasi kebutuhan, analisis, serta pemodelan untuk menciptakan solusi yang optimal sesuai dengan tujuan yang diinginkan. Dalam konteks sistem informasi, perancangan mencakup penentuan arsitektur sistem, pemilihan teknologi yang sesuai, dan pengembangan rencana implementasi yang efektif.

Selama proses perancangan, fokus utama adalah memahami kebutuhan pengguna dan memastikan bahwa solusi yang dihasilkan dapat memenuhi kebutuhan tersebut dengan baik. Ini melibatkan pembuatan model sistem yang jelas dan komprehensif, termasuk desain antarmuka pengguna yang intuitif dan fungsional. Selain itu, aspek keamanan dan skalabilitas juga diperhatikan dalam perancangan untuk memastikan bahwa sistem dapat beroperasi dengan aman dan dapat berkembang sesuai dengan kebutuhan masa depan.

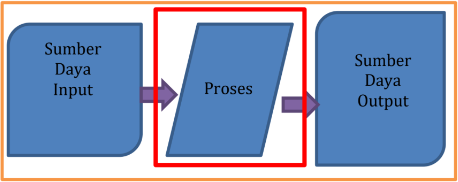
Hasil dari proses perancangan adalah rencana yang detail dan terstruktur untuk pengembangan dan implementasi sistem. Dengan perencanaan yang matang, diharapkan bahwa implementasi sistem akan berjalan lancar dan memberikan hasil yang sesuai dengan ekspektasi. Oleh karena itu, perancangan yang cermat dan terencana adalah langkah penting dalam menciptakan solusi yang efektif dan efisien bagi pengguna dan organisasi yang bersangkutan.

### Pengertian Sistem

Menurut Sitorus & Sakban, (2021:55) Sistem adalah serangkaian komponen yang saling berinteraksi dan bergantung satu sama lain untuk mencapai tujuan tertentu.

Menurut Nitami et al (2021:15) Sistem adalah sekumpulan objek yang bekerja bersama-sama untuk menghasilkan kesatuan metode, prosedur atau teknik yang digabungkan dan diatur sedemikian rupa sehingga menjadi satu kesatuan yang berfungsi untuk mencapai tujuan.

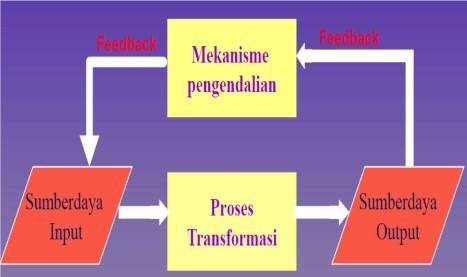
Menurut Dedy Rahman Prehanto et al.,( 2021:57) menyatakan bahwa “Pengelompokan sistem menjadi dua bagian dimana yang pertama sistem terbuka dan kedua sistem tertutup”.

Sistem terbuka merupakan hubungan proses sistem dengan lingkungan melalui arus sumber daya seperti pada gambar di bawah ini.

Sumber: Dedy Rahman Prehanto (2021:58)

GAMBAR 2. 2 Sistem Tertutup

Sistem tertutup adalah sistem yang tidak dipengaruhi oleh pihak luar dengan mekanisme pengendalian dapat dilihat pada gambar dibawah ini.



Sumber: Dedy Rahman Prehanto,(2021:58)

Selanjutnya ada beberapa karakteristik yang harus dimiliki sebuah sistem yaitu :

1. Komponen (*Component*)

Sistem terdapat komponen-komponen beberapa diantaranya melakukan interaksi dengan membentuk satu kesatuan dan saling bekerja sama yang terdiri dari berbagai cabang sistem.

1. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar pada sistem (*environment*) merupakan pengaruh operasi sistem oleh lingkungan luar sistem. Lingkungan luar sangat mempengaruhi yang bersifat merugikan tetap dijaga namun dikendalikan.

1. Batasan Sistem (*Boundry*)

Lingkup luar sistem yang dibatasi oleh ruang lingkup (*scope*) atau sistem dengan batas sistem lain yang sesuai bundaran daerahnya.

1. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan alat bantu yang menghubungkan antara satu subsistem ke subsistem lainnya.

1. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal,input*).

1. Keluaran Sistem (*Output*)

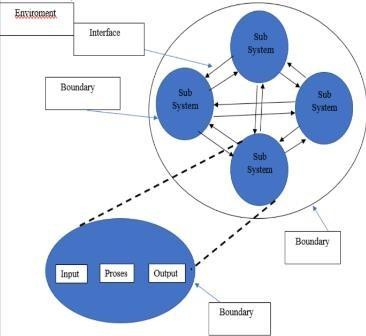
Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukkan untuk pengambilan keputusan.

1. Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

1. Sasaran Sistem (*Objective*)

Hasil Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *determinisfic*. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan. Berikut ini gambaran dari penjelasan sebuah karakteristik sistem diatas :



Sumber: Dedy Rahman Prehanto,(2021:59-30)

GAMBAR 2. 3 Karakteristik Sistem

Berdasarkan pengertian yang telah diuraikan, penulis dapat membuat kesimpulan bahwa sistem merupakan suatu entitas kompleks yang terdiri dari sejumlah komponen atau objek yang saling berinteraksi dan bergantung satu sama lain. Interaksi dan ketergantungan antar komponen tersebut diatur sedemikian rupa sehingga membentuk satu kesatuan yang memiliki metode, prosedur, atau teknik yang terpadu dan terkoordinasi.

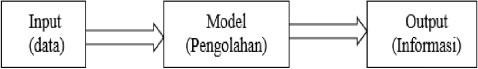
Tujuan dari sistem ini adalah untuk mencapai suatu hasil atau keadaan tertentu yang diinginkan. Dengan demikian, sistem dapat dipandang sebagai sebuah struktur organisasi yang kompleks dan terorganisir secara hierarkis, dimana setiap komponen memainkan peran penting dalam mencapai keseluruhan tujuan sistem tersebut.

Sistem di era digital adalah kumpulan perangkat lunak, perangkat keras, dan data yang saling terhubung dan berinteraksi secara kompleks untuk mencapai tujuan tertentu dalam lingkungan digital. Dalam konteks ini, sistem dapat mencakup berbagai hal, mulai dari jaringan komputer, platform perangkat lunak, aplikasi, hingga infrastruktur teknologi informasi yang mendukung operasi bisnis dan kegiatan sehari-hari. Sistem di era digital juga sering kali mencakup elemen - elemen seperti kecerdasan buatan, analisis data, dan konektivitas internet yang memungkinkan pertukaran informasi secara cepat dan efisien. Dengan adanya teknologi digital, sistem menjadi lebih fleksibel, dan adaptif untuk menghadapi berbagai tantangan dan kebutuhan yang muncul dalam dunia digital saat ini.

Istilah "informasi" berasal dari bahasa Latin *informatinem* yang berarti ide, kode, atau garis besar. Bentuk informasi dapat berupa tulisan, gambar, tabel, diagram, dan lain-lain.

Menurut Nitami et al (2021:33) Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Dengan kata lain sumber dari informasi adalah data.

Menurut Sitorus & Sakban (2021:12) Informasi adalah data yang diolah dari satu atau berbagai sumber yang memberikan nilai dan manfaat bagi penerima. Sumber dari informasi berupa data yang menggambarkan kejadian secara nyata yang telah terjadi pada saat tertentu.



GAMBAR 2. 4 Siklus Pengolahan Data

Sumber: Totok Haryanto (2021:19-20)

Berdasarkan pengertian yang telah diuraikan, penulis dapat membuat kesimpulan bahwa informasi adalah hasil dari pengolahan data yang menghasilkan suatu bentuk yang lebih berguna dan berarti bagi penerimanya. Tanpa informasi, suatu sistem tidak akan dapat berjalan dengan lancar dan akhirnya bisa mengalami kegagalan. Hal ini menunjukkan bahwa informasi memiliki peran yang sangat penting dalam operasi dan kelangsungan suatu sistem.

Selain itu, informasi juga dapat diperoleh dari berbagai sumber dan diolah menjadi bentuk yang memiliki nilai dan manfaat bagi penerimanya. Ini menekankan bahwa informasi bukan hanya sekedar data mentah, tetapi merupakan hasil dari proses pengolahan yang memberikan nilai tambah bagi penerima. Oleh karena itu, data merupakan sumber utama dari informasi, namun informasi memiliki nilai yang lebih tinggi karena telah mengalami proses pengolahan untuk menjadi lebih bermakna dan berguna.

Dengan demikian, kesimpulan dari ketiga pengertian tersebut adalah bahwa informasi merupakan hasil dari pengolahan data yang memiliki nilai dan manfaat bagi penerima, serta memainkan peran yang penting dalam kelancaran dan keberlangsungan suatu sistem.

### Definisi Sistem Informasi

Menurut Maydianto & M. Ridho (2021:52) menyatakan bahwa Sistem informasi merupakan sebuah kumpulan dari beberapa komponen yang mengelola data supaya data yang diolah dapat dijadikan sebagai informasi yang bermakna dan dapat membantu mencapai tujuan organisasi.

Menurut Dan et al (2022:87) menyatakan bahwa Sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan organisasi.

1. Manfaat Sistem Informasi

Beberapa manfaat sistem informasi adalah sebagai berikut :

* 1. Organisasi dapat menggunakan sistem informasi untuk mengurangi biaya dan menghasilkan produk atau pelayanan.
  2. Bank dapat menggunakan sistem informasi untuk mengolah cek-cek nasabah dan membuat berbagai laporan tentang transaksi dan rekening koran.
  3. Perusahaan dapat menggunakan sistem informasi untuk memantau stok persediaan secara *real time*.

1. Pemakai Sistem Informasi

Sebagian besar sistem informasi berbasis komputer berada di dalam suatu organisasi, dan ada berbagai jenis organisasi di sana. Semua orang dalam organisasi memanfaatkan informasi yang dihasilkan oleh sistem tersebut, termasuk manajer yang bertanggung jawab atas pembagian sumber daya untuk membantu pertumbuhan dan operasi bisnis.

1. Komponen Sistem Informasi
   1. *Hardware*

Terdiri dari *computer*, *peripheral (printer)* dan jaringan.

* 1. *Software*

Merupakan kumpulan fungsi dan perintah yang ditulis dengan aturan tertentu yang dimaksudkan untuk meminta komputer untuk melakukan tugas tertentu. *Software* dapat dikategorikan menjadi bahasa (3 GL dan 4 GL), sistem operasi (*Windows 95* dan NT), aplikasi (akuntansi), utilitas (anti virus, *speed disk*).

* 1. Data

Merupakan bagian utama informasi yang akan diproses lebih lanjut untuk menghasilkan informasi lainnya.

* 1. Prosedur

Buku penuntun operasional (aplikasi) dan teknis, serta dokumentasi prosedur dan proses sistem.

* 1. Manusia

Yang terlibat dalam elemen manusia seperti operator, pemimpin sistem informasi, dan sebagainya. Oleh karena itu, rincian tugas harus jelas.

1. Kegiatan Sistem Informasi
   1. *Input*

Menggambarkan cara mendapatkan data untuk diproses.

* 1. Proses

Menguraikan cara data diproses untuk menghasilkan informasi tambahan yang bernilai.

* 1. *Output*

Tugas untuk membuat laporan dari proses di atas.

* 1. Penyimpanan

Suatu tindakan yang melibatkan pengumpulan dan penyimpanan data. Tindakan untuk memastikan bahwa sistem informasi berjalan sesuai dengan yang diharapkan.

Dengan demikian, kesimpulan dari kedua pengertian tersebut bahwa ringkasannya, sistem informasi adalah kumpulan komponen yang mengelola data untuk menghasilkan informasi yang berguna bagi organisasi. Manfaatnya termasuk pengurangan biaya, peningkatan pendapatan, dan kemudahan pengolahan data. Penggunaannya melibatkan seluruh anggota organisasi, dengan komponen-komponen utamanya meliputi *hardware, software*, data, prosedur, dan manusia.

Kegiatan dalam sistem informasi mencakup *input*, proses*,* penyimpanan, dan kontrol. Dengan pemahaman yang baik tentang konsep dan komponen tersebut, organisasi dapat mengoptimalkan penggunaan sistem informasi untuk mencapai tujuan mereka.

### Definisi Pendataan

Menurut Zaitun, (2022:11) Pendataan adalah tahap dimana peserta memberikan informasi lengkap mengenai data pribadi serta program acara yang ingin diikuti.

Dapat disimpulkan bahwa pendataan dapat dipahami sebagai suatu proses yang melibatkan peserta untuk memberikan informasi secara komprehensif mengenai data pribadi mereka, seperti nama, alamat, nomor kontak, dan informasi lainnya yang relevan. Namun, lebih dari sekadar itu, pendataan juga mencakup pengumpulan informasi mengenai preferensi, kebutuhan, atau minat peserta terkait dengan program acara yang diadakan.

Ini mencakup detail tentang kehadiran peserta pada acara tertentu, minat mereka terhadap acara atau kegiatan yang diusulkan, dan preferensi lain yang mungkin mempengaruhi perencanaan dan penyelenggaraan acara tersebut.

Dengan demikian, pendataan di sini bukan hanya tentang pengumpulan informasi dasar, tetapi juga tentang memahami secara mendalam profil peserta dan memanfaatkannya untuk merencanakan kegiatan yang lebih sesuai dan memuaskan bagi seluruh peserta.

### Definisi Warga

Menurut Annas & Hazzar, (2023:56) yang dimaksud dengan warga negara adalah orang-orang bangsa Indonesia asli dan orang-orang bangsa lain yang disahkan dengan undang-undang sebagai warga negara.

Warga mengandung arti peserta, anggota atau warga dari suatu organisasi atau perkumpulan. Warga negara artinya warga atau anggota dari organisasi yang bernama negara .ada istilah rakyat, penduduk dan warga negara. Rakyat lebih merupakan konsep politis. Rakyat menunjuk pada orang-orang yang berada dibawah satu pemerintahan dan tunduk pada pemerintahan itu. Istilah rakyat umumnya dilawankan dengan penguasa. Penduduk adalah orang-orang yang bertempat tinggal di suatu wilayah negara dalam kurun waktu tertentu.

### Sejarah *Website*

Menurut Kusumawardani et al., (2023:43) *Website* pertama kali ditemukan oleh seorang ilmuan yang berasal dari negara inggris yang bernama Tim Berners-Lee.

Tim Berners- Lee juga merupakan anak dari seorang ilmuan komputer pada era awal dunia komputasi. Tim Berners-Lee membuat *Website* dengan tujuan agar memudahkan para peneliti untuk saling memberi dan menerima informasi di tempat kerjanya. Saat itu sebelum tahun 1990 an, Tim Berners-Lee menuliskan tentang tiga teknologi dasar yang ada pada *Website*, diantaranya adalah :

1. *Hyper Text Markup Language* (*HTML*)

Merupakan bahasa markup atau format untuk halaman *web*.

1. *Unifrom Resource Indetifier* (URI)

Merupakan sebuah alamat unik untuk membuka halaman situs. URI ini berfungsi untuk mengidentifikasi sumber daya yang ada pada *web*. URL saat ini sering di sebut dengan *Uniform Resource Locator* (URL).

1. *Hyper Text Transfer Protocol* (HTTP)

Merupakan sebuah protokol jaringan (*application layer*) yang dikembangkan dengan tujuan untuk memudahkan proses transfer antar komputer. Dengan protokol ini, pengguna bisa melakukan transfer file seperti dokumen, file, gambar, video, dan lain sebagainya.

### Definisi *Website*

Menurut Widia & Asriningtias, (2021:21) *Website* adalah kumpulan dokumen berupa halaman *web* yang berisi teks dalam format *Hyper Text Markup Language* (*HTML*).

Menurut Kusumawardani et al (2023:95) menyatakan bahwa *Website* merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara, dan video atau gabungan dari semuanya yang terkoneksi melalui internet, sehingga dapat di akses oleh seluruh ataupun siapapun orang yang bisa terkoneksi jaringan internet.

Dapat disimpulkan bahwa *Website* merupakan kumpulan halaman *web* atau dokumen digital yang terhubung melalui internet. Konten dalam sebuah *Website* dapat mencakup teks, animasi, gambar, suara, dan video, yang semuanya disusun dalam format *Hyper Text Markup Language* (*HTML*). Dengan demikian, sebuah *Website* tidak hanya terbatas pada teks dalam *HTML* tetapi juga melibatkan berbagai jenis konten multimedia yang membuatnya menjadi sumber informasi yang kaya dan dapat diakses oleh siapa saja yang terkoneksi dengan jaringan internet.

### Jenis-jenis *Website*

Menurut Kusumawardani et al., (2023:122) ditinjau dari aspek *content* atau isi ,*web* dapat dibagi menjadi 3 (tiga) jenis, yaitu *Website* statis, *Website* dinamis, dan *Website* interaktif.

Selain dari sisi content atau isi, *Website* statis, *Website* dinamis, dan *Website* interaktif dapat dilihat dari aspek teknologi yang digunakan untuk membuat jenis *web* tersebut. Adapun jenis-jenis *web*:

#### Website Statis

Merupakan *Website* yang mempunyai halaman yang tidak berubah. Perubahan suatu halaman dilakukan secara manual dengan mengedit kode yang menjadi struktur dari *Website* tersebut.

Atau istilah sederhananya adalah Ketika pemakai ingin merubah tampilan atau menginputkan sesuatu di *Website* tersebut maka harus membuka Kembali kode yang di tulis melalui aplikasi atau *text editor.*

#### Website Dinamis

Merupakan *Website* yang secara terstruktur di peruntukan untuk di update sesering mungkin. Atau *Website* tersebut mempunyai konten yang bisa di rubah secara berulang kali tanpa harus membuka kode dari *Website*nya. *Website* ini biasanya disediakan halaman *backend* untuk melakukan perubahan konten dari *Website* tersebut.

#### Website Interaktif

Merupakan *Website* yang berinteraksi antara penggunanya. Biasanya berupa forum diskusi maupun blog.

## Penelitian Sebelumnya / Relevan

Berikut ini merupakan tabel hasil penelitian-penelitian Sebelumnya yang di ambil dari berbagai sumber :

TABEL 2. 1 Penelitian Sebelumnya/Relevan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Nama & Tahun & Judul** | **Metode** | **Hasil Penelitian** | **Persamaan** | **Perbedaan** |
| 1 | **F Ardiansyah & P Rosyani (2021)**  Sistem Informasi Pendataan Masyarakat Kurang Mampu Berbasis *Web* di Kelurahan Cisalsak | *Waterfall*, UML | Sistem berhasil memetakan warga miskin berbasis *Website* dengan fitur data, pencarian, dan laporan. | Sama-sama menggunakan model *Waterfall* dan pengembangan berbasis *web*. | Fokusnya pada pendataan warga kurang mampu, bukan seluruh warga perumahan. |
| 2 | **DR Hidayat & MA Rosid (2022)**  Implementasi Framework CodeIgniter Dalam Pembuatan Sistem Informasi Pencatatan dan  Pendataan Penduduk Desa Berbasis *Web* | *Waterfall*, CodeIgniter | Sistem efektif dalam menyimpan dan menampilkan data penduduk lengkap dengan KK dan alamat. | Sama-sama berbasis *web* dan sistem informasi kependudukan. | Menggunakan framework CodeIgniter, bukan *HTML*/CSS native atau lainnya. |
| 3 | **FL Witi & S Soraya (2021)**  Rancang Bangun Sistem Informasi Pendataan  Penduduk di Kantor Kelurahan Kotaratu Kabupaten Ende | *Waterfall*, PHP,*MySQL* | Sistem dapat menggantikan proses manual dengan pencatatan digital oleh petugas kelurahan | Sama-sama bertujuan mengganti pendataan manual menjadi berbasis sistem | Fokusnya hanya pada pendataan warga saja |

## *Tools* yang digunakan



### UML (*Unified Modelling Language)*

Menurut Hasanah & Untari, (2021:85) Menyatakan bahwa UML merupakan salah satu standart bahasa yang banyak digunakan di dunia industry untuk mendefinisikan requirement, membuat analisis & desain,serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

1. *Use Case diagram*

*Use case diagram* adalah teknik untuk merekam persyaratan fungsional sebuah *system*, menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.

Berdasarkan definisi di atas, *Use Case Diagram* digunakan untuk menjelaskan interaksi antara *Aktor* (pengguna sistem) dan *Use Case* (Kasus Penggunaan), yang disesuaikan dengan langkah-langkah (Scenario) yang telah ditentukan. Fokus dari *Use Case Diagram* adalah "apa" yang dilakukan sistem, bukan "bagaimana" itu dilakukan.

TABEL 2. 2 Simbol Use Case Diagram

|  |  |  |
| --- | --- | --- |
| No | Simbol | Dekripsi |
| 1 | *Use Case* | fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. |
| 2 | Aktor/*Aktor*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (163).png | mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. |
| 3 | *Association* | menghubungkan *link* antar *element*. |
| 4 | Ekstensi / *Extend*  << extend >> | kelakuan yang hanya berjalan di bawah kondisi tertentu. |
| 5 | Generalisasi / *generalization*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (164).png | Sebuah elemen dapat merupakan spesialisasi dari elemen lainnya. |
| 6 | *Include / uses*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (165).png | kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah use case adalah bagian dari use case lainnya. |

Sumber: Hasanah & Untari, (2021:23-24)

1. *Activity diagram*

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di*-trigger* oleh selesainya state sebelumnya *(internal processing).*

*Activity Diagram* adalah suatu teknik yang digunakan untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja dalam suatu sistem. Dalam *Activity Diagram,* berbagai alur aktivitas dalam sistem yang sedang dirancang digambarkan, termasuk bagaimana masing-masing alur dimulai, keputusan yang mungkin terjadi, dan bagaimana alur tersebut berakhir. Selain itu, *Activity* Diagram juga dapat digunakan untuk menggambarkan proses paralel yang terjadi pada beberapa eksekusi. Dengan menggunakan *Activity Diagram*, pengembang sistem dapat memvisualisasikan alur kerja sistem secara jelas dan memudahkan dalam memahami dan merancang sistem yang efektif dan efisien.

TABEL 2. 3 Simbol-Simbol Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1 |  | Status Awal | Sebuah diagram aktivitas memiliki sebuah status  awal |
| 2 |  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3 |  | Percabangan/ *Decision* | Percabangan dimana ada pilihan aktivitas yang lebih dari satu |
| 4 |  | Penggabungan / *Join* | Penggabungan dimana yang lebih  Dari satu aktivitas lalu digabungkan jadi satu |
| 5 |  | Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir |
| 6 | C:\Users\ADMIN\Pictures\Screenshots\Screenshot (167).png | *Swimlane* | *Swimlane* memisahkan organisasi bisnis yang bertanggung jawab terhadap apa yang terjadi |

Sumber : Rasiban et al., (2024:33)

Dari tabel diatas, berikut penjelasan mengenai simbol-simbol yang terdapat pada *Activity Diagram* :

1. Titik Awal atau Status Awal (Titik Awal/Status Awal)

*Start Point* adalah lingkaran hitam kecil yang digunakan untuk menunjukkan status awal, tindakan awal, atau titik awal aktivitas dalam diagram aktivitas.

1. Aktivitas (*Activity*)

Aktivitas adalah tindakan atau kejadian yang terjadi di dalam sebuah sistem. Biasanya dimulai dengan kata kerja yang menggambarkan aktivitas yang dilakukan.

1. Keputusan atau Percabangan

Percabangan atau keputusan adalah titik yang menunjukkan kondisi di mana ada beberapa kemungkinan transisi. Hal ini diperlukan ketika sistem memiliki beberapa jalur alternatif.

1. Sinkronisasi

Sinkronisasi dibagi menjadi dua bagian, yaitu *fork* dan *join*.

1. *Fork*

digunakan untuk membagi perilaku menjadi beberapa aktivitas atau aksi yang paralel.

1. *Join*

digunakan untuk menghubungkan kembali aktivitas dengan tindakan paralel.

1. Penggabungan (*Merge*)

*Merge* menggabungkan *flow* yang telah dipecah menjadi beberapa bagian oleh sebuah *flow*.

1. *Swimlanes*

*Swimlanes* membagi diagram aktivitas ke dalam kolom dan baris untuk membagi tanggung jawab objek- objek yang melakukan suatu aktivitas.

1. *Squence Diagram*

Menurut Rasiban et al (2024:36) menyatakan bahwa *Squence Diagram* adalah jenis diagram dalam *Unified Modeling Language* (UML) yang menggambarkan interaksi antara objek dalam suatu sistem secara kronologis. Diagram ini menunjukkan bagaimana objek-objek berkomunikasi satu sama lain dan berurutan dalam eksekusi suatu skenario atau proses.

*Sequence* *diagram* adalah salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menjelaskan interaksi antar objek dalam sebuah sistem secara visual. Diagram ini menampilkan urutan tindakan dan komunikasi antara berbagai komponen sistem, serta dapat digunakan untuk menggambarkan serangkaian langkah yang dilakukan sebagai respons dari sebuah *event* atau *use case*. *Squence Diagram* berguna untuk memahami, merancang, dan memodelkan sistem yang kompleks.

TABEL 2. 4 Simbol Squence Diagram

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Deskripsi** |
| 1 | Aktor | Orang, Proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. |
| 2 | *Entity Class*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (168).png | bagian dari sebuah sistem yang terdiri dari sekumpulan kelas yang mewakili entitas-entitas yang membentuk kerangka awal sistem dan menjadi dasar untuk merancang basis data. |
| 3 | *Boundary Class*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (169).png | yang mencakup kumpulan kelas yang menjadi *interfaces* atau interaksi antara satu atau lebih Aktor dengan sistem, seperti tampilan *form entry* dan *form*  cetak. |
| 4 | *Control Class*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (170).png | merupakan sebuah objek yang mengandung logika aplikasi yang tidak terikat pada entitas, seperti kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
| 5 | *Recursive*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (172).png | merujuk pada proses pengiriman pesan yang ditujukan kembali kepada dirinya sendiri. |
| 6 | *Message*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (171).png | simbol mengirim pesan antar *class*. |
| 7 | *Activation*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (173).png | mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding dengan durasi aktivasi sebuah operasi. |
| 8 | *Lifeline*  C:\Users\ADMIN\Pictures\Screenshots\Screenshot (174).png | Garis yang menghubungkan titik- titik dengan objek, di mana terdapat *activation* sepanjang *lifeline*. |

Sumber: Rasiban et al., (2024:37 - 38)

1. *Class Diagram*

Menurut Informasi et al (2023:22) *Class Diagram* yaitu menggambarkan serta deskripsi dari *class*, atribut dan objek serta hubungan satu sama lain. *Class diagram* dapat memberikan pandangan global atas sebuah *system*. Hal tersebut tercermin dari *class* yang ada dan relasinya satu dengan yang lainnya.

TABEL 2. 5 Simbol *Class Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | *Class* | Himpunan objek-objek dari berbagai atribut yang me-miliki operasi yang sama. |
|  | *Association* | Relasi antar kelas dengan makna umum dan biasanya disertai  *multiplicity*. |
|  | *Directed Association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain. |
|  | *Aggregation* | Mengindikasikan keseluruhan bagian *relationship* disebut sebagai relasi. |
|  | *Composition* | Relasi *Composition* terhadap *class* tempat  dia bergantung. |
|  | *Dependency* | Menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. |

Sumber : Informasi et al., (2023:9)

TABEL 2. 6 Simbol Indicator Relation Class Diagram

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1 | 0..1 | Kosong atau satu |
| 2 | 0..\* | Lebih dari sama dengan kosong |
| 3 | 0..n | Lebih dari sama dengan n, dimana n lebih dari satu |
| 4 | 1 | Hanya satu |
| 5 | 1..\* | Lebih dari sama dengan satu dimana n lebih dari satu |
| 6 | 1..n | Lebih dari sama dengan satu dimana n lebih dari 1 |
| 7 | \* | Banyak atau *many* |
| 8 | N | Hanya n, dimana n lebih dari satu |
| 9 | n..\* | Lebih dari sama dengan nn dimana n lebih dari satu |
| 10 | n..m | Lebih dari sama dengan n dan kurang dari sama dengan m dan n lebih dari satu |

### PHP (*Hypertext Preprocessor)*

Menurut Sihombing,(2022:88) PHP merupakan bahasa pemrograman untuk membuat atau mengembangkan aplikasi berbasis *web* dan bersifat *open source* dan ditanamkan ke dalam *script HTML*.

Menurut Habibi & Sandi, (2020:67) PHP (*Hypertext Preprocessor*) adalah suatu bahasa pemrograman bersifat *server side* yang digunakan untuk menerjemahkan sejumlah baris kode (bisa ditambahkan ke dalam *HTML*) program menjadi kode mesin dimengerti oleh mesin komputer.

Dari penjelasan yang telah disampaikan, maka penulis dapat mengambil kesimpulan bahwa PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang bersifat *open source* dan berorientasi pada pengembangan aplikasi *web*. PHP beroperasi di sisi *server* (*server-side*) dan memungkinkan pembuat konten untuk menanamkan kode program ke dalam *HTML*.

Fungsi utama PHP adalah untuk menginterpretasikan kode-kode program menjadi instruksi yang dapat dipahami dan dijalankan oleh komputer, sehingga memudahkan pengembangan dan pengelolaan aplikasi *web* yang dinamis dan interaktif.

### *HTML*

Menurut Kusumawardani et al., (2023:49) menyatakan bahwa *HTML* merupakan sebuah bahasa pemrograman standar yang berfungsi untuk membuat halaman *Website* agar dapat diakses dan menampilkan berbagai jenis konten lewat perantara browser internet.

Menurut Menurut (Muhammad Yusril Helmi Setyawan, 2020) *HTML* merupakan suatu metode untuk mengimplementasikan konsep *hypertext* dalam suatu naskah atau dokumen.

Dari penjelasan yang telah disampaikan, maka penulis dapat mengambil kesimpulan bahwa *HTML* adalah bahasa pemrograman standar yang digunakan untuk membuat dan mendesain halaman *web*. *HTML* memungkinkan pengintegrasian berbagai jenis konten, seperti teks, gambar, video, dan animasi, ke dalam dokumen *web* yang dapat diakses melalui browser internet. *HTML* bekerja dengan konsep *hypertext*, yang memfasilitasi pembuatan *link* atau tautan antar dokumen, memungkinkan navigasi yang mudah dan dinamis di dalam internet.

### *MySQL*

Menurut Namruddin et al., (2023:22) *MySQL* adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server.*

Menurut Kom et al., (2023:28) *MySQL* adalah sistem manajemen basis data relasional (RDBMS) *open source* yang banyak digunakan di seluruh dunia.

Dari penjelasan yang telah disampaikan, maka penulis dapat mengambil kesimpulan bahwa *MySQL* adalah merupakan sistem manajemen basis data (*database management system* atau DBMS) yang bersifat *open source* dan relasional (RDBMS). *MySQL* menggunakan bahasa SQL (*Structured Query Language*) sebagai alat untuk berinteraksi dan mengelola data yang disimpan di dalam *database* *server*. Sebagai RDBMS, *MySQL* mendukung penyimpanan data dalam format tabel yang saling terkait, memudahkan penggunaan dan pengelolaan data secara efisien dan efektif. *MySQL* banyak digunakan di seluruh dunia karena kemampuannya yang kuat dalam pengelolaan *database* untuk aplikasi *web* dan sebagai bagian dari stack teknologi LAMP (Linux, Apache, *MySQL*, PHP/Python/Perl).

### *XAMPP*

Menurut Nurmufid et al (2023:29) *XAMPP* merupakan perangkat lunak berbasis *web* *server* yang bersifat *open source* (bebas), serta mendukung di berbagai sistem operasi, Windows, Linux, atau Mac OS.

Menurut Andre Andhara, (2022:77) *XAMPP* merupakan perangkat lunak (*free software*) bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program.

Dari penjelasan yang telah disampaikan, maka penulis dapat mengambil kesimpulan bahwa *XAMPP* adalah perangkat lunak *server* *web* yang bersifat *open source* dan mendukung berbagai sistem operasi seperti Windows, Linux, dan Mac OS.

*XAMPP* merupakan kumpulan atau kompilasi dari beberapa program yang dirancang untuk menyediakan lingkungan pengembangan *web* lokal yang lengkap dan mudah digunakan. Perangkat lunak ini biasanya digunakan untuk keperluan pengembangan dan pengujian aplikasi *web* secara lokal sebelum diimplementasikan ke *server* yang sebenarnya.

### *Black box Testing*

Menurut Pressman dalam Aini et al (2022:8648- 8649) “ *Black Box Testing* yaitu suatu pengujian yang hanya dilakukan pada kebutuhan fungsional sistemnya atau nama lainnya yaitu pengujian *behavior* (perilaku).

Menurut Mardzotillah dan Ridwan (2024:95) *Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program.

Penulis dapat mengambil kesimpulan bahwa metode pengujian perangkat lunak yang dikenal sebagai "*Black Box Testing*" berfokus pada fungsionalitas dan *output* tanpa memperhatikan cara perangkat lunak bekerja.

Berikut adalah beberapa kelebihan dari pengujian *Black box testing*:

* + - 1. Pengujian ini dapat dilakukan oleh orang-orang yang tidak memiliki pengetahuan teknis yang kuat karena tidak memerlukan pengetahuan bahasa pemrograman tertentu.
      2. Dilakukan dari perspektif pengguna, sehingga memperbaiki ambiguitas atau ketidak konsistenan dalam spesifikasi persyaratan.
      3. Membantu *programmer* dan *tester* bekerja sama untuk memastikan bahwa perangkat lunak berfungsi dengan baik dari sudut pandang pengguna.

Pengujian *black box testing* membuat pengujian perangkat lunak lebih mudah dan efektif karena tidak memerlukan pengetahuan teknis yang mendalam dan dapat membantu menemukan masalah dari perspektif pengguna. Selain itu, *black box testing* juga dapat membantu programmer dan tester bekerja sama untuk membuat perangkat lunak.

## Kerangka Kerja Teoritis

Latar Belakang Masalah

Penelitian ini akan merancang sebuah sistem informasi pendataan warga perumahan Duta Asri Ciakar melalui berbasis website. Tujuannya adalah meningkatkan efektivitas dan efisiensi pendataan warga perumahan dengan memanfaatkan teknologi informasi. Penelitian akan menggunakan metode pengembangan perangkat lunak SDLC dengan pendekatan waterfall, yang memungkinkan langkah-langkah yang terstruktur dan berurutan dalam pengembangan sistem. Diharapkan sistem ini dapat mengatasi kendala dalam pendataan manual yang rentan terhadap kesalahan, serta meningkatkan partisipasi aktif warga dalam pengelolaan lingkungan tempat tinggal mereka.

Rumusan Masalah

1. Bagaimana cara merancang sistem pendataan warga Perumahan Duta Asri Ciakar yang sedang berjalan saat ini?
2. Bagaimana merancang sistem pengelolaan data warga Perumahan Duta Asri Ciakar secara efektif melalui *Website*?
3. Bagaimana cara warga Perumahan Duta Asri Ciakar menambahkan data di *Website*?

Tujuan Masalah

* + 1. Mengidentifikasi dan mengevaluasi mekanisme pendataan warga yang saat ini digunakan di Perumahan Duta Asri Ciakar, guna memahami kekuatan dan kelemahannya sebagai dasar dalam merancang sistem pendataan yang lebih efektif, efisien.
    2. Memastikan akses yang mudah bagi pengguna dalam proses pendataan warga Perumahan Duta Asri Ciakar melalui *Website*.
    3. Mengurangi kesalahan penulisan dan penumpukan berkas dalam pendataan warga melalui *Website*.

Analisa

Metode Pengembangan

* *Waterfall*

Metode Pengumpulan Data

* Observasi
* Wawancara
* Studi Pustaka

Hasil Akhir

Menghasilkan *Website* Perancangan Sistem Informasi Pengelolaan Pendataan Warga Perumahan Duta Asri Ciakar Melalui *Website.*

GAMBAR 2. 5 Kerangka Kerja Teoritis

# BAB III METODOLOGI PENELITIAN



## Desain Penelitian

Menurut Sarie et al., (2023:26) Desain Penelitian adalah rencana atau strategi yang digunakan untuk mengatur Langkah- langkah dalam melakukan penelitian dengan tujuan mencapai tujuan penelitian yang telah di tetapkan

Untuk memastikan bahwa variabel yang diuji dikendalikan dengan baik dan hasilnya dapat diandalkan, maka penulis menggunakan penelitian eksperimental. Oleh karena itu, penelitian ini dilakukan dengan menggunakan metode pengujian yang terkontrol dan mengumpulkan data secara sistematis dan terstruktur.

Dengan desain penelitian eksperimental, penulis dapat menguji efektivitas sistem informasi pendataan warga Perumahan Duta Asri Ciakar melalui *Website* yang telah dirancang. Hasil penelitian ini dapat memberikan kontribusi dalam meningkatkan efisiensi administrasi perumahan dan dapat digunakan sebagai dasar untuk pengembangan lebih lanjut dalam penerapan teknologi informasi dalam pendataan warga.

### Tahap Penelitian

TABEL 3. 1. Tahap Penelitian



Analisis

Analisis

Analisis

Analisis

Desain

Desain

Desain

Desain

Pengodean

Pengodean

Pengodean

Pengodean

Pengujian

Pengujian

Pengujian

Pengujian

SDLC : *Waterfall*

Sistem

Sistem

Sistem

Sistem

**Studi Literatur**

Literatur pendukung penelitian

1. Jurnal
2. Sumber Buku
3. Internet

**Studi Literatur**

Literatur pendukung penelitian

1. Jurnal
2. Sumber Buku
3. Internet

**Studi Literatur**

Literatur pendukung penelitian

1. Jurnal
2. Sumber Buku
3. Internet

**Studi Literatur**

Literatur pendukung penelitian

1. Jurnal
2. Sumber Buku
3. Internet

**Pengumpulan Data**

1. Wawancara
2. Observasi
3. Studi Pustaka

**Pengumpulan Data**

1. Wawancara
2. Observasi
3. Studi Pustaka

**Pengumpulan Data**

1. Wawancara
2. Observasi
3. Studi Pustaka

**Pengumpulan Data**

1. Wawancara
2. Observasi
3. Studi Pustaka

**Tahap awal penelitian**

1. Menentukan kebutuhan data yang diperlukan
2. Mengumpulkan data yang diperlukan

**Tahap awal penelitian**

1. Menentukan kebutuhan data yang diperlukan
2. Mengumpulkan data yang diperlukan

**Tahap awal penelitian**

1. Menentukan kebutuhan data yang diperlukan
2. Mengumpulkan data yang diperlukan

**Tahap awal penelitian**

1. Menentukan kebutuhan data yang diperlukan
2. Mengumpulkan data yang diperlukan

## Data dan Sumber Data



### Jenis Data

Dalam penelitian "Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar Berbasis *Website* Menggunakan Metode *Waterfall*" data primer menjadi landasan utama. Data primer diperoleh melalui interaksi langsung dengan pihak terkait dalam pendataan warga, seperti pengurus perumahan, ketua RT, atau warga yang terlibat aktif dalam proses pendataan. Proses pengumpulan data primer dilakukan melalui wawancara yang terstruktur dan mendalam, yang memungkinkan penulis untuk memperoleh informasi yang relevan dan komprehensif tentang permasalahan yang dihadapi, kebutuhan pengguna, serta harapan terhadap sistem yang akan dikembangkan.

Selain wawancara, pengumpulan data primer juga dilengkapi dengan pencatatan tertulis untuk mencatat setiap detail dan aspek yang dibahas selama interaksi dengan informan. Catatan tertulis ini menjadi penting karena memastikan bahwa informasi yang diperoleh dapat dianalisis secara mendalam dan akurat dalam tahap selanjutnya dari penelitian. Dengan demikian, data primer yang diperoleh secara langsung dari sumber terpercaya menjadi pijakan yang kokoh dalam merancang, mengembangkan, dan mengevaluasi sistem informasi pendataan warga.

### Sumber Data

Sumber data yang dibutuhkan dalam penelitian ini terbagi menjadi beberapa bagian, di antaranya:

### Tempat Penelitian

Tempat penelitian yang dilakukan penulis dalam penelitian ini di Perumahan Duta Asri Ciakar, Kec. Legok, Kabupaten Tangerang, Banten 15820.

### Waktu Penelitian

Waktu penelitian adalah waktu yang dibutuhkan penulis dalam melakukan penelitian untuk pengumpulan data. Dalam penelitian ini penulis melakukan penelitian untuk pengumpulan data dari bulan Mei 2025 sampai dengan bulan Agustus 2025.

TABEL 3. 2 Waktu Penelitian 2025

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **2024 – 2025** | | | | | | | | | | | | | | | | | | | | | | | |
| **Maret** | | | | **April** | | | | **Mei** | | | | **Juni** | | | | **Juli** | | | | **Agustus** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| 1 | Perencanaan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Analisis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Desain |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | *Testing* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Dokumentasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Metode Pengumpulan Data

Dalam pengumpulan data yang digunakan dalam penelitian ini, penulis menggunakan beberapa metode pengumpulan data sebagai berikut :



### Metode Observasi

Dalam rangka "Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar Berbasis *Website*", penulis menggunakan metode pengamatan langsung terhadap proses pengelolaan iuran yang sedang berlangsung. Dengan pencatatan yang sistematis, penelitian ini bertujuan untuk memahami secara mendalam alur proses saat ini dalam pengumpulan dan pencatatan data warga. Proses pengamatan ini memungkinkan penulis untuk melihat secara langsung dinamika yang terjadi, identifikasi masalah yang muncul, dan menilai efektivitas serta efisiensi dari sistem yang sedang berjalan. Dengan demikian, data yang diperoleh dari metode pengamatan langsung ini akan menjadi dasar yang kuat dalam merancang solusi yang tepat dan efisien melalui pengembangan sistem informasi pendataan warga melalui *Website*.

### Metode Wawancara

Dalam konteks "Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar Berbasis *Website* Menggunakan Metode *Waterfall*", penulis melakukan wawancara dengan pihak-pihak terkait untuk mendapatkan pemahaman yang lebih mendalam tentang sistem pendataan yang sedang berjalan dan harapan mereka terhadap sistem yang diusulkan. Melalui pertanyaan-pertanyaan langsung, penulis bertujuan untuk menggali informasi tentang proses pengumpulan dan pencatatan data warga yang saat ini digunakan, serta mendapatkan masukan dan harapan dari pihak terkait untuk perbaikan dan pengembangan sistem yang lebih baik di masa mendatang. Dengan demikian, hasil wawancara tersebut menjadi landasan penting dalam merancang dan mengimplementasikan sistem informasi yang lebih efektif dan efisien untuk pendataan warga dalam lingkungan Perumahan.

### Metode Studi Pustaka

Dalam rangka "Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar Berbasis *Website* Menggunakan Metode *Waterfall*", penulis melakukan studi pustaka dengan menelaah buku- buku, jurnal, dan literatur terkait, termasuk tugas akhir yang relevan dengan penelitian ini. Studi pustaka menjadi bagian penting dalam pengumpulan data, yang digunakan untuk memperoleh pemahaman yang mendalam tentang konsep, teori, dan praktik terkait pengelolaan sistem informasi. Informasi yang diperoleh dari studi pustaka tersebut akan digunakan untuk memperkuat dan melengkapi data yang telah diperoleh dari observasi dan wawancara sebelumnya. Dengan demikian, studi pustaka menjadi salah satu metode yang penting dalam mendukung penyusunan sistem informasi pendataan warga dalam lingkungan Perumahan Duta Asri Ciakar melalui *Website* .

## Metode Analisis Data

Dalam konteks "Perancangan Sistem Informasi Pendataan Iuran Warga Perumahan Duta Asri Ciakar berbasis *Website* menggunakan metode *Waterfall*  penulis menggunakan teknik analisis data deskriptif untuk memproses data yang diperoleh dari hasil wawancara, catatan lapangan, dan dokumentasi. Proses analisis dilakukan dengan mengorganisasikan data ke dalam kategori- kategori yang relevan, menjelaskan hasil evaluasi sistem yang sudah ada, serta mengembangkan sistem baru yang lebih efektif.

Hasil dari analisis ini akan digunakan untuk membuat revisi yang diperlukan dalam *Website* yang dikembangkan. Dengan demikian, teknik analisis data deskriptif menjadi kunci dalam memahami, mengevaluasi, dan mengembangkan sistem informasi pendataan warga Perumahan Duta Asri Ciakar berbasis *Website* menggunakan metode *Waterfall*.

## Metode Perancangan / Pengembangan Sistem

Dalam konteks "Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar berbasis *Website* menggunakan metode *Waterfall*.", penulis memilih metode pengembangan sistem SDLC (Systems Development Life Cycle) dengan menerapkan metode *Waterfall*. Alasan pemilihan metode ini adalah karena kemudahannya dalam dipahami serta struktur yang baik. SDLC memastikan setiap tahap pengembangan sistem dilakukan secara terstruktur dan terukur, sehingga menghasilkan sistem yang berkualitas dan sesuai dengan kebutuhan pengguna. Metode *Waterfall*, yang merupakan salah satu tahapan dalam SDLC, memastikan bahwa setiap fase pengembangan sistem diselesaikan dengan baik sebelum memasuki fase berikutnya.

Dengan demikian, metode ini sangat sesuai untuk pengembangan *Website* yang membutuhkan tahapan yang terstruktur dan terukur, seperti dalam perancangan sistem informasi pendataan warga Perumahan Duta Asri Ciakar berbasis *Website* menggunakan metode *Waterfall*.



### Analisis Kebutuhan Sistem

Dalam konteks "Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar berbasis *Website* menggunakan metode *Waterfall*.", analisis kebutuhan sistem sangat penting untuk mendukung kinerja sistem yang sedang dirancang. Kebutuhan sistem yang teridentifikasi akan menjadi landasan untuk mencapai tujuan yang ditetapkan. Dalam mengidentifikasi data dan proses yang dibutuhkan pada rancangan sistem yang diusulkan, analisis kebutuhan sistem bertujuan untuk membantu dalam mengembangkan sistem yang efektif dan sesuai dengan kebutuhan. Dengan melakukan analisis kebutuhan sistem secara menyeluruh, penulis dapat menentukan kebutuhan sistem secara fungsional dan non- fungsional yang dibutuhkan dalam pengelolaan iuran warga perumahan, sehingga sistem yang dirancang dapat berjalan dengan optimal dan memberikan manfaat yang maksimal bagi pengguna.

* + - 1. Fungsional
         1. Sistem dapat melakukan proses pendaftaran pengguna, seperti warga perumahan, dan memberikan akses ke *Website*.
         2. Sistem menghasilkan laporan data secara berkala yang mencakup informasi tentang NIK, nama, nomor telepon, dan lain-lain.
         3. Sistem memungkinkan pengguna untuk memasukkan dan mengelola data secara mudah dan efisien.
      2. Non fungsional
         1. Sistem berbasis *Website* dirancang dengan antarmuka pengguna yang intuitif dan mudah digunakan, sehingga meminimalkan waktu belajar pengguna dan meningkatkan kepuasan pengguna.
         2. Sistem mampu mengakomodasi pertumbuhan jumlah pengguna dan volume data tanpa mengalami penurunan kinerja.
         3. Sistem dirancang dengan struktur yang mudah dipahami dan dimodifikasi, sehingga pemeliharaan dan pembaruan sistem dapat dilakukan dengan efisien.

### Desain Diagram yang Digunakan

Dalam penelitian "Perancangan Sistem Informasi Pendataan Warga Perumahan Duta Asri Ciakar berbasis *Website* m enggunakan metode *Waterfall*.", pembuatan diagram merupakan bagian dari implementasi hasil analisis kebutuhan sistem. Pada tahap ini, penulis memodelkan kebutuhan sistem ke dalam bentuk diagram menggunakan bahasa pemodelan *Unified Modeling Language* (UML).

UML digunakan sebagai sarana perancangan sistem berorientasi objek, yang memungkinkan penulis untuk secara visual menggambarkan struktur dan interaksi antara berbagai komponen sistem. Dalam konteks *Website* pendataan warga, UML digunakan untuk membuat diagram seperti *class diagram, use case diagram*, *Squence Diagram*, dan lainnya.

Diagram-diagram ini membantu dalam memvisualisasikan kebutuhan fungsional dan nonfungsional dari sistem yang dirancang, serta memudahkan proses pengembangan dan dokumentasi aplikasi.

Berikut *diagram-diagram* yang digunakan dalam penelitian ini, yaitu :

1. *Use case diagram*

Penelitian ini menggunakan *use case diagram* untuk mendekripsikan interaksi satu atau lebih Aktor dengan sistem yang akan dibuat atau diusulkan.

1. *Activiy diagram*

Diagram aktivitas, juga dikenal sebagai *activity diagram*, menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses. Dalam penelitian perancangan *Website* pendataan warga, diagram ini dapat digunakan untuk menggambarkan menu atau aliran kerja yang ada dalam perangkat lunak.

1. *Squence Diagram*

Untuk menunjukkan interaksi antar objek dalam sistem dalam penelitian ini, *Squence Diagram* digunakan. Diagram ini menunjukkan urutan tindakan dan komunikasi antara berbagai komponen sistem, seperti pengguna dan objeknya. Garis vertikal menunjukkan setiap komponen, dan panah horizontal menunjukkan pesan yang bergerak di antara mereka.

1. *Class diagram*

Hubungan antara kelas dan struktur data dalam sistem pengelolaan iuran warga dapat digambarkan melalui *class diagram* yang digunakan untuk menjelaskan kelas-kelas yang akan digunakan untuk membuat sistem yang dirancang.

### Software yang Digunakan

Dalam penelitian ini, software yang digunakan:

1. *Operation System* : *Windows 11 Home Single 22H2 , 64 bit*
2. Bahasa Pemograman : *PHP*
3. *Database Server* : *MYSQL*
4. *Text Editor* : *Visual Studio Code*
5. *Web Server* : *XAMPP*

### Metode Pengujian Sistem yang Digunakan

Teknik / Metode Pengujian Sistem adalah fase di mana penulis menguji desain dan kode dengan menggunakan metode *black box testing*. *Black Box Testing* adalah pengujian perangkat lunak tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukkan, dan keluaran perangkat lunak sesuai dengan spesifikasi. Pengujian ini dilakukan dengan membuat kasus uji yang mencoba semua fungsi perangkat lunak untuk melihat apakah sesuai dengan spesifikasi.

# BAB IV PEMBAHASAN



## Analisa Sistem Yang Bejalan

### Latar Belakang Objek Penlitian

Perumahan Duta Asri Ciakar adalah salah satu dari sedikit kawasan perumahan yang terletak di Kelurahan Serdang Wetan, Kecamatan Legok, Kabupaten Tangerang, Provinsi Banten. Menjadi vital *hub* dengan meningkatkan keinginan masyarakat terhadap hunian nyaman, aman, dan strategis di kawasan Tangerang. Dengan demikian, Perumahan Duta Asri Ciakar tidak hanya sekedar kawasan hunian, tetapi juga merupakan objek penelitian yang layak diaplikasikan dalam konteks membangun permukiman, maupun dalam konteks dinamika sosial ekonomi dan implikasi tata ruang kebijakan di Kabupaten Tangerang.

### Visi Dan Misi

Visi

Mewujudkan lingkungan Perumahan Duta Asri yang aman, nyaman, tertib dan harmonis melalui partisipasi warga serta pembangunan berbasis kebersamaan dan keberlanjutan.

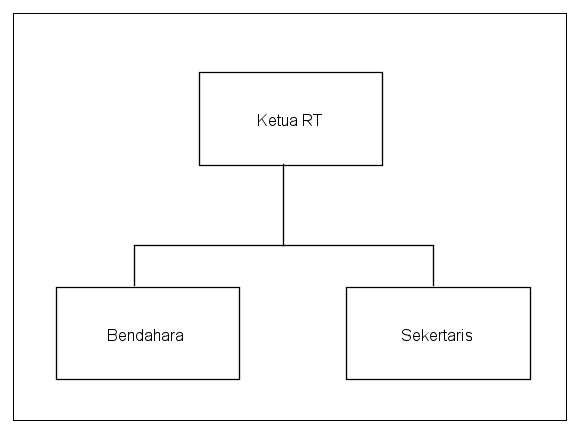
Misi

1. Menjalin kerja sama antar warga Masyarakat
2. Menciptakan aparatur yang bersih
3. Mengadakan kegiatan gotong royong
4. Mendukung dan memfasilitasi kegiatan di lingkungan

### Struktur Organisasi dan Uraian Tugas

* + 1. Struktur organisasi

Struktur organisasi merupakan suatu susunan dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan. Struktur organisasi menggambarkan dengan yang lainnya dan bagaimana hubungan aktivitas dan fungsi dibatasi untuk mencapai tujuan.



GAMBAR 4. 1 Struktur Organisasi

* + 1. Ketua RT

Ketua RT adalah seseorang yang di pilih dari hasil musyawarah warga serta memiliki tugas dan tanggung jawab meliputi:

1) Berwenang memberikan instruksi dan koman ke seluruh jajaran pengurus serta melakukan evaluasi hasil kinerjanya

2) Berwenang untuk memberikan sanksi kepada seluruh warga yang melanggar aturan yang telah disepakati bersama

3) Menjembatani hubungan antara sesama warga dan juga antara warga dengan pemerintah daerah

4) Menangani berbagai macam masalah-masalah yang terjadi di lingkungan masyarakat

5) Bertanggung jawab untuk menjalankan tugas pelayanan kepada masyarakat setempat.

6) Memelihara kerukunan hidup antar warga

7) Menyusun rencana dan melaksanakan pembangunan dengan mengembangkan aspirasi masyarakat

b) Bendahara

Bendahara adalah posisi dalam struktur organisasi RT yang memiliki tugas dan tanggung jawab meliputi:

1. Mencatat semua pemasukan dan pengeluaran uang dengan rapi dan jelas.
2. Menyimpan uang organisasi dengan aman.
3. Menyusun laporan keuangan secara berkala, misalnya tiap bulan atau tiap akhir kegiatan.
4. Menyiapkan dan mengatur dana untuk kegiatan organisasi.
5. Bertanggung jawab kalau ada audit atau pengecekan keuangan.
   * 1. Sekretaris

Sekretaris adalah posisi dalam struktur organisasi RT yang memiliki tugas dan tanggung jawab meliputi:

1. Menulis dan menyimpan notulen rapat (catatan hasil rapat).
2. Mengelola surat masuk dan keluar.
3. Menyusun jadwal kegiatan dan membantu koordinasi dengan pihak lain.
4. Menyimpan dokumen-dokumen penting organisasi.
5. Membantu ketua dalam menyusun laporan atau administrasi lainnya.

### Analisa Sistem Yang Berjalan

* 1. *Activity Diagram* proses yang berjalan

Berikut ini adalah Activity Diagram yang berjalan pada Perumahan Duta Asri Ciakar seperti yang tercantum pada gambar 4.2 di bawah ini.

GAMBAR 4. 2 Activity Diagram Yang Sedang Berjalan

Penjelasan dari *Activity Diagram* sistem berjalan sebagai berikut:

1. Ada 3 Aktor yaitu RT, Bendahara, Warga.
2. *Use Case* yaitu informasi pendataan warga, melakukan pembayaran kas atau sampah, mendata warga melaui *website,* mengelola data warga, dan menerima laporan keuangan.

Berikut Ini definisi Aktor pada sistem pendataan warga yang sedang berjalan.

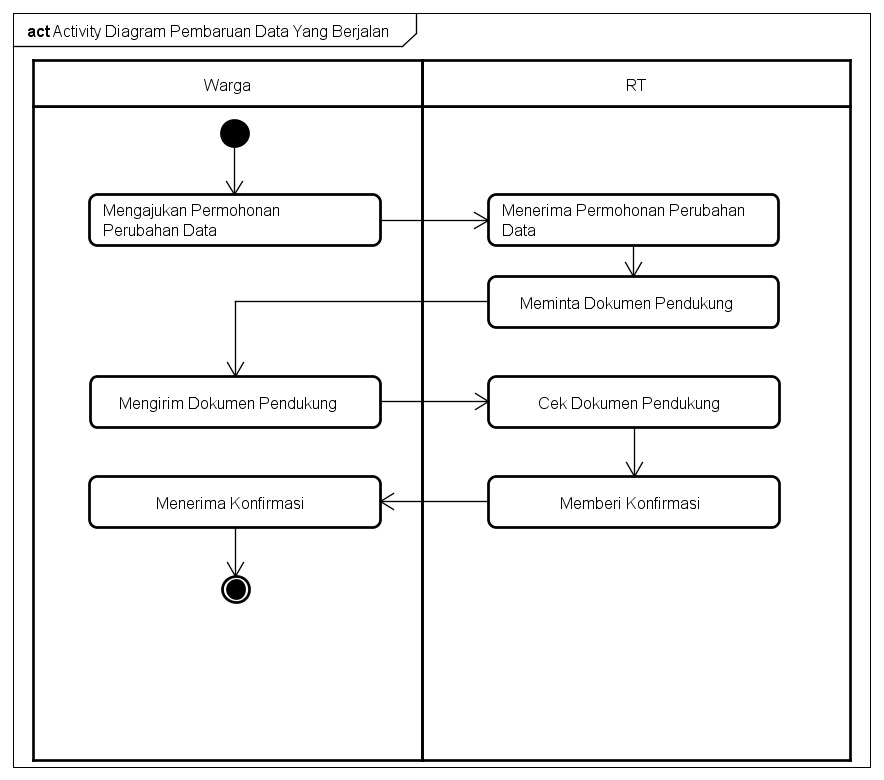
|  |  |  |
| --- | --- | --- |
| sa a. Nama Aktor | : | RT |
| sa Deskripsi | : | Aktor yang mengelola warga dan menerima hasil dari Bendahara. |
| sa b. Nama Aktor | : | Bendahara |
| sa Deskripsi | : | Aktor yang mgelola pembayaran kas dan sampah. |
| sa c. Nama Aktor | : | Warga |
| sa Deskripsi | : | Aktor yang bertugas untuk melakukan pengkinian atau pembaruan data |

Berikut ini adalah skenario *use case diagram* sistem yang sedang berjalan:

TABEL 4. 1 Skenario Use Case Diagram Yang Berjalan

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama *Use Case* | Nama Aktor | Deskripsi |
| 1 | Mencatat Data Warga | RT | RT mencatat data warga secara manual (buku catatan). |
| 2 | Meng-*update* Data Warga | RT | RT memperbarui data jika ada perubahan. |
| 3 | Menyimpan Arsip Data *Manual* | RT | RT menyimpan semua catatan dalam dokumen fisik. |
| 4 | Menagih Iuran Bulanan Ke Warga | Bendahara | Bendahara menagih uang dari rumah ke rumah. |
| 5 | Mencatat Pembayaran Iuran *Manual* | Bendahara | Bendahara mencatat pembayaran secara manual. |
| 6 | Membayar Iuran Bulanan | Warga | Warga memberikan uang secara langsung. |
| 7 | Meminta *Update* Data Keluarga | Warga | Warga datang ke RT untuk meminta perubahan data.pembayaran kas dari warga |

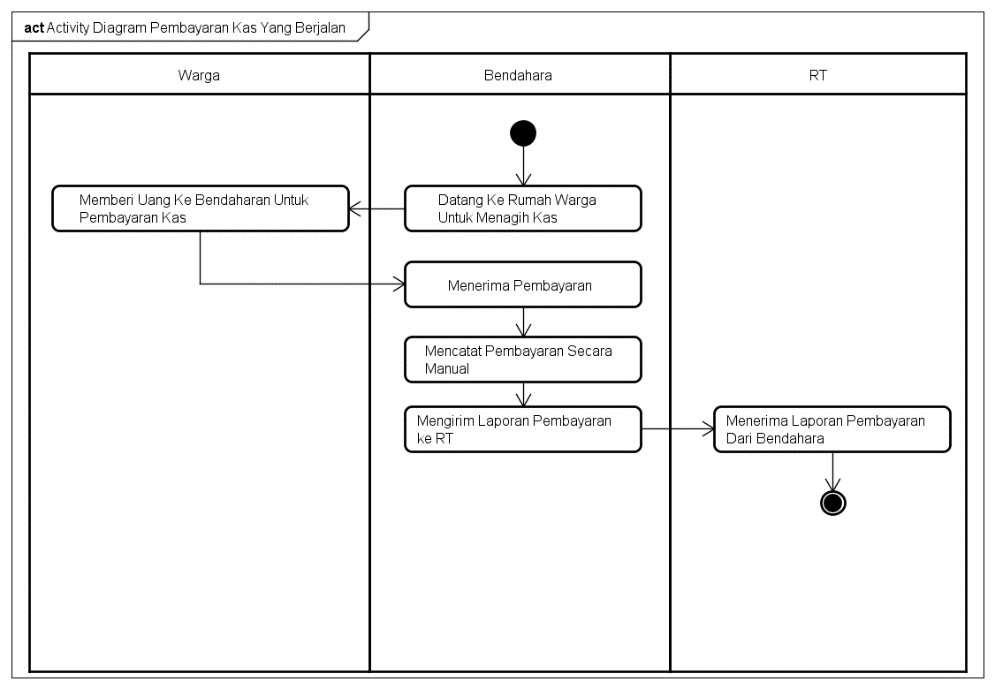
* 1. *Activity Diagram* Sistem Pendataan Warga yang Berjalan



GAMBAR 4. 3 Activity Diagram Pendataan Warga Yang Berjalan

Keterangan :

1. Mulai
2. Warga mengajukan permohonan perubahan data kepada RT
3. Rt menerima permohonan perubahan data dari warga
4. Rt meminta dokumen pendukung lainnya
5. Warga mengirimkan Dokumen pendukung kepada RT
6. RT mengecek dokumen warga
7. RT mengkonfirmasi
8. Selesai
   1. *Activity Diagram* Pembayaran Kas Yang Berjalan



GAMBAR 4. 4 Activity Diagram *Pembayaran Kas Yang Berjalan*

Keterangan :

1. Mulai
2. Bendahara datang ke rumah warga untuk menagih kas
3. Warga memberikan uang kepada bendahara untuk pembayaran kas
4. Bendahara menerima pembayaran
5. Bendahara mencatat pembayaran secara manual
6. Bendahara mengirim laporan pembayaran ke RT
7. RT menerima laporan pembayaran dari bendahara
8. Selesai

### Masalah Yang Terjadi

Berdasarkan hasil pengamatan dan penelitian dari sistem pendataan warga yang sedang berjalan di Perumahan Duta Asri, Penulis menemukan masalah – masalah sebagai berikut :

1. Pendataan warga masih dilakukan secara manual.
2. Pembayaran kas dan sampah masih dilakukan secara *Dor To Dor* sehingga membutuhkan waktu yang cukup lama untuk melakukan pendataannya
3. Berkas yang menumpuk menyebabkan penyimpanan berkas penuh dan pencarian data warga membutuhkan waktu yang lama.
4. Risiko kesalahan data dalam pencatatan manual.

### Aternatif Pemecahan Masalah

Setelah mengetahui masalah yang dihadapi oleh Pengurus Perumahan Duta Asri maka penulis mencoba memecahkan masalah tersebut yaitu :

* + 1. Dengan merancang dan membangun sebuah sistem pendataan warga dan pembayaran kas berbasis *website* dapat mengefisienkan waktu dan biaya.
    2. Dengan merancang dan membangun sebuah sistem pendataan warga dan pembayaran kas dapat memberikan informasi lebih cepat dan efisien.
    3. Dengan merancang dan membangun sebuah sistem pendataan warga dan pembayaran kas dapat lebih mudah memonitoring laporan data warga dan pembayaran kas warga.

## Rancangan Sistem yang Diusulkan



### *Use Case Diagram* yang Diusulkan

GAMBAR 4. 5 Use Case Diagram yang Diusulkan

Keterangan:

1. Definisi Aktor
   * + 1. RT

Aktor ini memiliki hak yang lebih luas. Selain memiliki kemampuan untuk *Login* dan *Logout*, RT memiliki kemampuan untuk mengakses pengguna", yang memungkinkan untuk mengelola, akun pengguna memiliki kemampuan untuk mengelola data warga, yang memungkinkan untuk menambah, mengubah, atau menghapus data warga; dan memiliki kemampuan untuk melihat laporan, yang dapat mencakup kemungkinan laporan keuangan atau laporan lainnya yang relevan dengan RT.

* + - 1. Warga

Aktor ini memiliki kemampuan untuk masuk dan keluar dari sistem. Selain itu, warga dapat melihat laporan, yang mungkin berisi laporan tentang iuran atau informasi lainnya, dan mengakses data warga, yang memungkinkan mereka untuk melihat data pribadi mereka atau data warga lain yang diizinkan.

* + - 1. Bendahara

Aktor ini lebih berkonsentrasi pada hal-hal yang berkaitan dengan uang. Bendahara memiliki kemampuan untuk *Login* dan *Logout*. tugas utamanya adalah mengelola kas menambahkan, mengubah, atau menghapus data kas, melakukan pembayaran (mencatat atau memproses pembayaran), dan melihat berbagai jenis laporan, termasuk laporan kas dan laporan sampah (kemungkinan terkait dengan iuran kebersihan atau retribusi sampah).

1. Definisi *Use Case*

TABEL 4. 2 Definisi Use Case

|  |  |  |
| --- | --- | --- |
| No | *Use Case* | Deskripsi |
| 1 | *Login* | *Use case* ini menggambarkan proses yang dilakukan semua pengguna sebelum mengakses menu yang dimana menu tersebut harus *login* terlebih dahulu. |
| 2 | Pembayaran | *Use case* ini menggambarkan proses yang dilakukan bendahara untuk mencatat dan melakukan pembayaran kas atau sampah. |
| 3 | Tambah Kas | *Use case* ini menggambarkan proses yang dilakukan oleh bendahara untuk menambahkan data kas baru. |
| 4 | Ubah Kas | *Use case* ini menggambarkan proses yang dilakukan oleh bendahara untuk mengedit data kas yang telah tercatat |
| 5 | Hapus Kas | *Use case* ini menggambarkan proses bendahara untuk menghapus data kas. |
| 6 | Laporan Kas | *Use case* ini menggambarkan proses yang dilakukan bendahara dan rt untuk melihat dan mencetak laporan keuangan kas. |
| 7 | Laporan | *Use case* ini menggambarkan proses yang dilakukan semua pengguna untuk mengakses berbagai jenis laporan. |
| 8 | Laporan Kas | *Use case* ini menggambarkan proses yang dilakukan seluruh pengguna untuk melihat laporan pengelolaan kas. |
| 8 | Laporan Sampah | *Use case* ini menggambarkan proses yang dilakukan seluruh pengguna untuk melihat laporan pengelolaan sampah. |
| 9 | Data Warga | *Use case* ini menggambarkan proses yang dilakukan seluruh pengguna untuk melihat dataa warga. |
| 10 | Tambah Warga | *Use case* ini menggambarkan proses yang dilakukan rt, dan warga untuk menambahkan data warga baru. |
| 11 | Ubah Warga | *Use case* ini menggambarkan proses yang dilakukan rt, dan warga untuk mengubah data warga yang sudah ada. |
| 12 | Hapus Warga | *Use case* ini menggambarkan proses yang dilakukan rt untuk menghapus data warga yang sudah tidak aktif. |
| 14 | Akses *User* | *Use case* ini menggambarkan proses yang dilakukan rt untuk mengatur hak akses seluruh pengguna. |
| 15 | Acc Pembaruan | *Use case* ini menggambarkan proses persetujan pembaruan data dari warga oleh RT |
| 16 | Pembaruan Data | *Use case* ini menggambarkan proses pembaruan atau pengkinian data yang dilakukan oleh warga atau bendahara |
| 17 | *Logout* | *Use case* ini menggambarkan proses yang dilakukan semua pengguna setelah melakukan *login.* |

1. Skenario *Use Case*
   * + 1. Nama *use case : Login*

Aktor : Semua Pengguna

TABEL 4. 3 Skenario Use Case Login

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| Skenario Normal | |
| 1. Melakukan Akses Aplikasi |  |
|  | 1. Menam­­­pilkan *form login* |
| 1. *Input Username* dan *Password* |  |
| 1. Klik tombol *Sign In* |  |
| ­­­ | 1. Verifi­­­kasi *Username* dan *Password* |
|  | 1. Bila *valid*, sistem akan menampilkan halaman *Dashboard* untuk masing - masing pengguna sesuai hak akses |
| Skenario alternatif | |
|  | 1. Bila tidak valid, sistem akan menampilkan pesan “*Username is not registered!*” |
| 1. *Input Username* dan *Password* yang valid |  |
| 1. Klik tombol *Sign In* |  |
|  | 1. Verifikasi *Username* dan *Password* |
|  | 1. Menampilkan halaman *dashboard* |

1. Nama *use case* : Pembayaran

Aktor : Bendahara

TABEL 4. 4 Skenario Use Case Pembayaran

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Pembayaran lalu Kas Masuk |  |
|  | 1. Menampilkan *submenu* pembayaran |

1. Nama use case : Tambah Kas

Aktor : Bendahara

TABEL 4. 5 Skenario Use Case Tambah Kas

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Pembayaran lalu Kas Masuk |  |
| 1. Klik tombol pemasukan |  |
|  | 1. Menampilkan *form* pembayaran kas masuk |
| 1. *Input* data data yang dimunculkan *form* |  |
| 1. Klik tombol *Confirm* |  |
|  | 1. Menyimpan data yang telah di-*input* oleh *user* |
|  | 1. Menampilkan halaman kas masuk |

1. Nama *use case* : Ubah Kas

Aktor *:* Bendahara

TABEL 4. 6 Skenario Use Case Ubah Kas

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Pembayaran lalu Kas Masuk |  |
| 1. Pilih data lalu klik ikon pensil |  |
|  | 1. Menampilkan *form* kas dengan data yang dipilih sebelumnya |
| 1. Ubah data yang ingin diubah datanya |  |
| 1. Klik tombol *Confirm* |  |
|  | 1. Menyimpan data yang telah diubah oleh *user* |
|  | 1. Menampilkan halaman kas dan memunculkan pesan “Data berhasil Diupdate” |

1. Nama *use case* : Hapus Kas

Aktor *:* Bendahara

TABEL 4. 7 Skenario Use Case Hapus Kas

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Pembayaran lalu Kas Masuk |  |
| 1. Pilih data lalu klik ikon hapus |  |
|  | 1. Menampilkan notif “Apakah Kamu Yakin Ingin menghapus Data Ini?” |
| 1. Klik Tombol Hapus |  |
|  | 1. Sistem akan menghapus data yang dipilih |
|  | 1. Menampilkan halaman kas dan memunculkan pesan “Data Berhasil Dihapus!” |

1. Nama *use case* : Laporan

Aktor: RT dan Bendahara

TABEL 4. 8 Skenario Use Case Laporan

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. User Memilih Menu Laporan |  |
|  | 1. Menampilkan Sub Menu Laporan |

1. Nama *use case* : Laporan Kas

Aktor: RT dan Bendahara

TABEL 4. 9 Skenario Use Case Laporan Kas

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. User Memilih Menu Laporan |  |
|  | 1. Menampilkan Sub Menu Laporan |
| 1. Memilih Laporan Kas |  |
|  | 1. Menampilkan laporan kas dengan urutan terbaru terlebih dahulu |

1. Nama *use case* : Laporan Sampah

Aktor: RT dan Bendahara

TABEL 4. 10 Skenario Use Case Laporan Sampah

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. User Memilih Menu Laporan |  |
|  | 1. Menampilkan Sub Menu Laporan |
| 1. Memilih Laporan Sampah |  |
|  | 1. Menampilkan Laporan Sampah dengan urutan terbaru terlebih dahulu |

1. Nama *use case* : Data Warga

Aktor *:* RT

TABEL 4. 11 Skenario Use Case Data Warga

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Menu Warga |  |
|  | 1. Menampilkan halaman warga dan data-data warga |
|  | 1. Menampilkan *form* tambah warga |

1. Nama *use case* : Tambah Warga

Aktor *:* RT

TABEL 4. 12 Skenario Use Case Tambah Warga

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Menu Warga |  |
| 1. Klik tombol (+ Warga) |  |
|  | 1. Menampilkan *form* tambah warga |
| 1. *Input* data data yang dimunculkan *form* |  |
| 1. Klik tombol *Confirm* |  |
|  | 1. Menyimpan data yang telah di-*input* oleh *user* |
|  | 1. Menampilkan halaman warga lalu memunculkan pesan “Data berhasil ditambahkan!” |

1. Nama *use case* : Ubah Warga

Aktor: RT

TABEL 4. 13 Skenario Use Case Ubah Warga

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Menu Warga |  |
| 1. Pilih data lalu klik ikon pensil |  |
|  | 1. Menampilkan *form* edit warga dengan data yang dipilih sebelumnya |
| 1. Ubah data yang ingin diubah datanya |  |
| 1. Klik tombol *Confirm* |  |
|  | 1. Menyimpan data yang telah diubah oleh *user* |
|  | 1. Menampilkan halaman warga dan memunculkan pesan “Data Berhasil Diubah!” |

1. Nama *use case* : Hapus Warga

Aktor : RT

TABEL 4. 14 Skenario Use Case Hapus Warga

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. Klik Menu Warga |  |
| 1. Pilih data lalu klik ikon hapus |  |
|  | 1. Menampilkan notif “Apakah Kamu Yakin Ingin Menghapus Data Ini?” |
| 1. Klik tombol Hapus |  |
|  | 1. Sistem akan menghapus data yang dipilih |
|  | 1. Menampilkan halaman warga dan memunculkan pesan “Data Berhasil Dihapus!” |

1. Nama *use case* : Akses User

Aktor : RT

TABEL 4. 15 Skenario Use Case Akses User

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. User Memilih Menu Akses User |  |
|  | 1. Menampilkan User Yang Terdaftar |

1. Nama *use case* : Acc Pembaruan

Aktor : RT

TABEL 4. 16 Skenario Use Case Acc Pembaruan

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. User Memilih Menu Akses Acc Pembaruan |  |
|  | 1. Menampilkan Data Yang Mengajukan Pembaruan |
| 1. Memilih Data Yang Ingin Disetujui atau Ditolak |  |
|  | 1. Sistem akan mengupdate data jika disetujui oleh user jika ditolak sistem akan meng-*update* status menjadi ditolak |

1. Nama *use case* : Pembaruan Data

Aktor : Warga dan Bendahara

TABEL 4. 17 Skenario Use Case Pembaruan Data

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. *User* Memilih Menu Akses Pembaruan Data |  |
|  | 1. Menampilkan *Form* data user |
| 1. *User* mengubah data yang ingin diubah |  |
|  | 1. Sistem akan menyimpan data sebagai data sementara yang akan disetujui atau ditolak oleh RT kemudian. |

1. Nama *use case* : Logout

Aktor: Semua pengguna

TABEL 4. 13 Skenario Use Case Laporan Sampah

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| 1. *User* klik *dropdown* disebelah profil |  |
|  | 1. Menampilkan Sub Menu |
| 1. Memilih *Logout* |  |
|  | 1. Sistem akan menampilkan halaman *Login* |

### *Activity Diagram* yang Diusulkan

1. *Activity Diagram Login*

GAMBAR 4. 6 Activity Diagram Login

Keterangan:

1. *User* membuka aplikasi

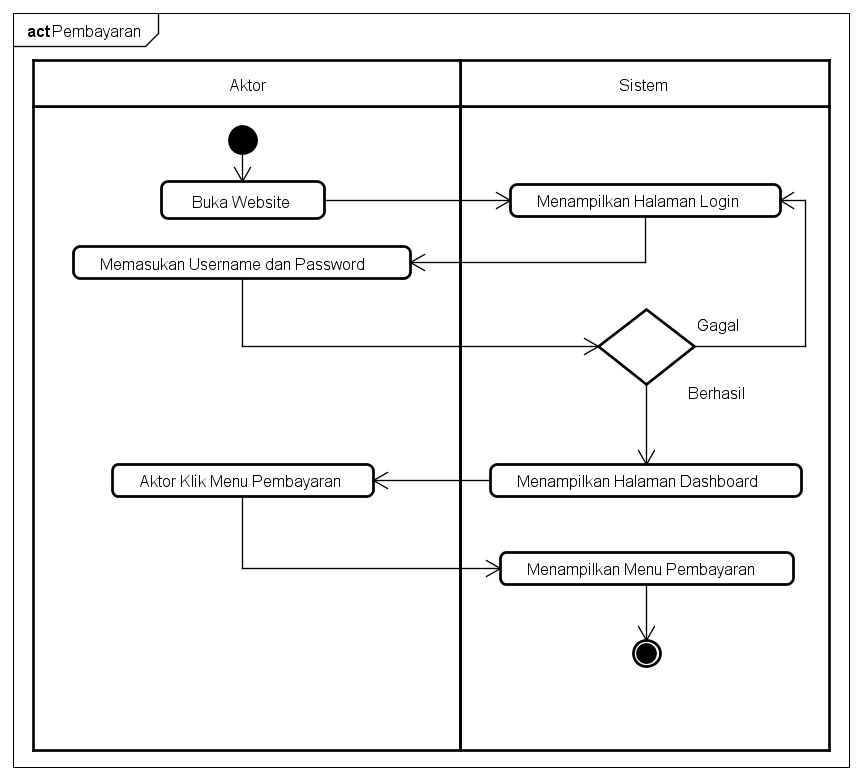
2. Sistem menampilkan form *login*

3. Input *username* dan *password*

4. Sistem mevalidasi apakah *username* dan *password* tersebut valid atau tidak, jika *valid* makan sistem akan menampilkan menu *dashboard* jika tidak kembali ke form *login*

5. Selesai

1. *Activity Diagram* Pembayaran



GAMBAR 4. 6 Use Case Diagram Pembayaran

Keterangan:

1. *User* membuka aplikasi
2. Sistem menampilkan *form login*
3. *User* menginput *username* dan *password*
4. Sistem memvalidasi *username* dan *password*, jika valid maka menampilkan *dashboard*, jika tidak kembali ke form login
5. *User* memilih menu pembayaran
6. Sistem menampilkan halaman/menu pembayaran
7. Selesai
8. *Activity Diagram* Tambah Kas

GAMBAR 4. 7 *Activity Diagram* Tambah Kas

Keterangan :

1. *User* membuka *website* lalu melakukan *login*

2. Memilih menu pembayaran lalu pilih kas

3. *User* klik tombol bayar lalu mengisi data kas masuk

4. Setelah semua terisi klik *confirm*

5. Sistem menyimpan data di *database* yang telah di *input* oleh *user*

6. Selesai

1. *Activity Diagram* Ubah Kas

GAMBAR 4. 8 *Activity Diagram* Ubah Kas

Keterangan :

1. *User* membuka *website* lalu melakukan *login*

2. Memilih menu pembayaran lalu pilih kas

3. *User* memilih kas yang ingin diubah lalu klik ikon pensil

4. Lalu ubah data yang ingin diubah lalu klik *confirm*

5. Sistem mengubah data di *database* yang telah diubah oleh *user*

6. Selesai

1. *Activity Diagram* Hapus Kas

GAMBAR 4. 9 *Activity Diagram* Hapus Kas

Keterangan:

1. *User* membuka *website* lalu melakukan *login*

2. Memilih menu pembayaran lalu pilih kas

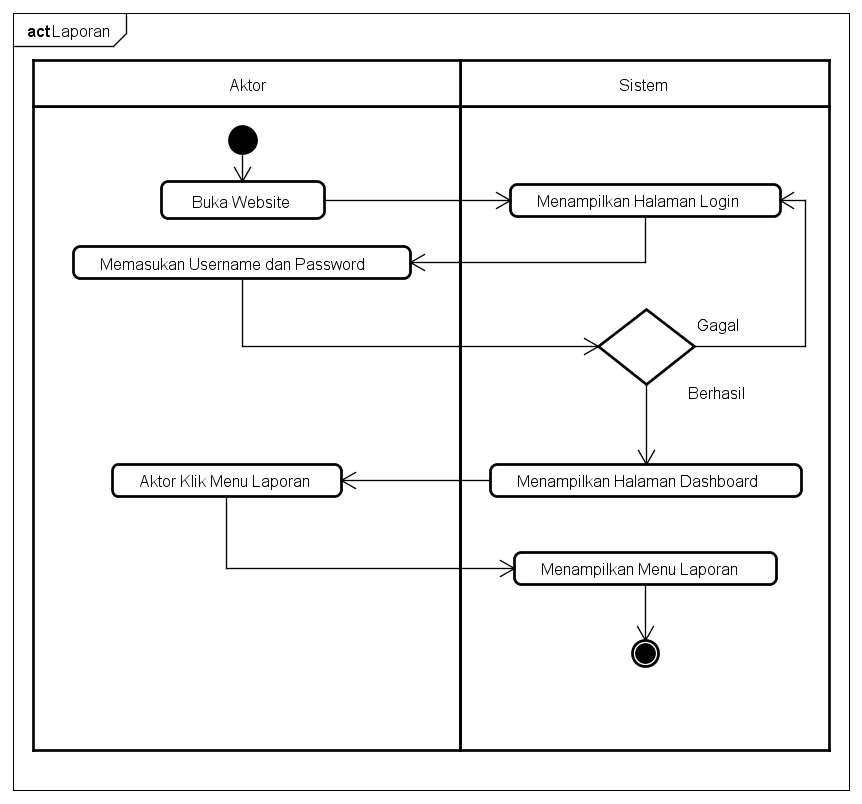
3. *User* memilih kas yang ingin dihapus lalu klik ikon sampah

4. Sistem akan menampilkan modal validasi dengan pesan "*Data will deleted permanently"*

5. Sistem akan menghapus data yang dipilih oleh *user* di *database*

6. Selesai

1. *Activity Diagram* Laporan

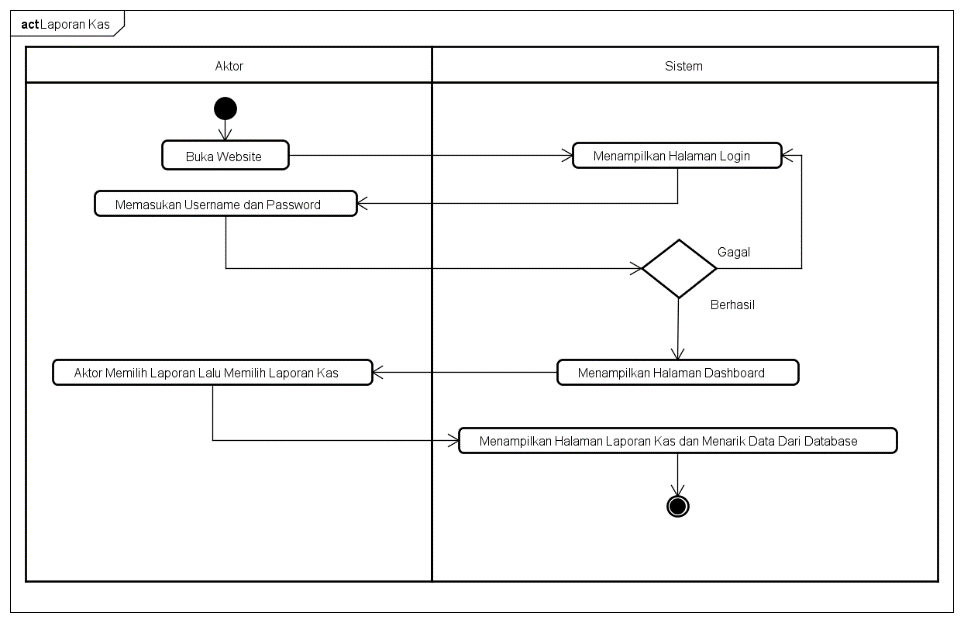


GAMBAR 4. 10 *Activity Diagram* Laporan

Keterangan:

* + 1. *User* membuka website
    2. Sistem menampilkan halaman *login*
    3. *User* meng-*input username* dan *password*
    4. Sistem memvalidasi *username* dan *password*, jika valid maka menampilkan *dashboard*, jika tidak kembali ke *form* *login*
    5. User klik menu laporan
    6. Sistem menampilkan menu laporan
    7. Selesai

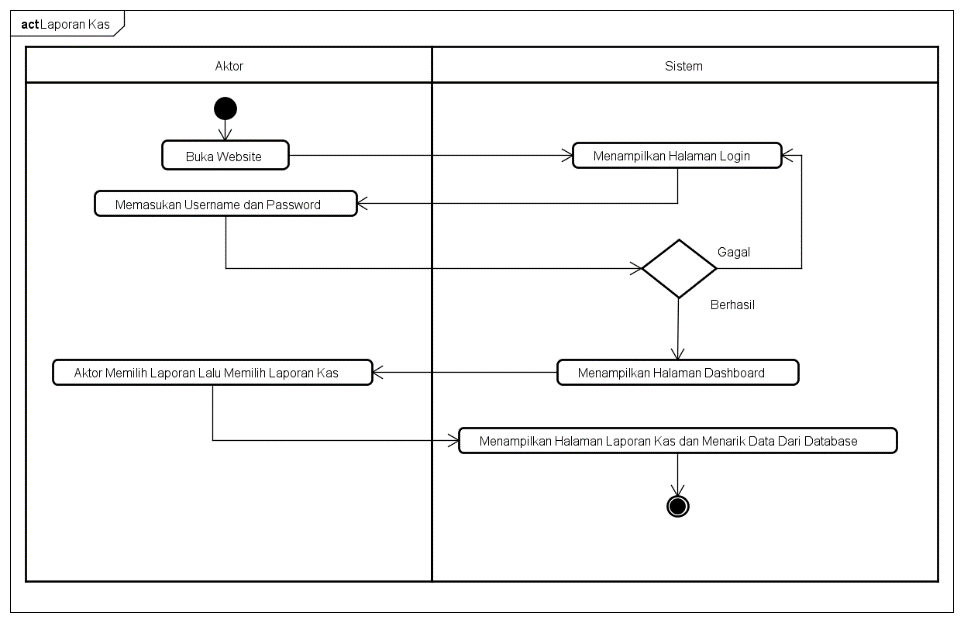
1. *Activity Diagram* Laporan Kas



GAMBAR 4. 11 *Activity Diagram* Laporan Kas

Keterangan:

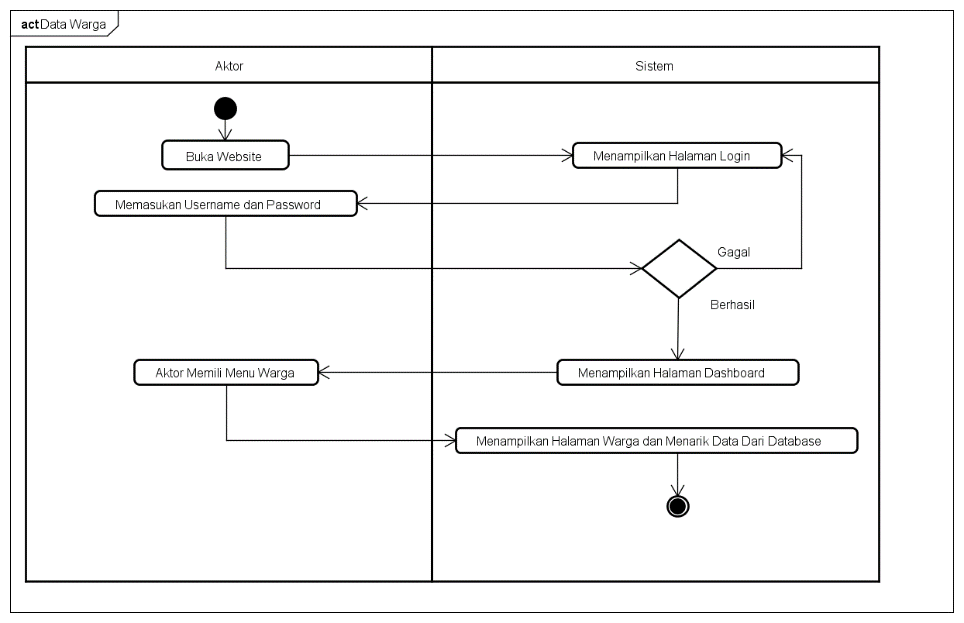
1. *User* membuka *website*.
2. Sistem menampilkan halaman *login*.
3. Sistem memverifikasi *login*. Jika gagal, kembali ke halaman *login*. Jika berhasil, lanjut.
4. Sistem menampilkan *dashboard*. *User* memilih menu laporan kas.
5. Sistem menampilkan halaman laporan kas dan mengambil data dari *database*.
6. Selesai.
7. *Activity Diagram* Laporan Sampah



GAMBAR 4. 12 *Activity Diagram* Laporan Sampah

Keterangan:

1. *User* membuka *website*.
2. Sistem menampilkan halaman *login*.
3. Sistem memverifikasi *login*. Jika gagal, kembali ke halaman *login*. Jika berhasil, lanjut.
4. Sistem menampilkan *dashboard*. *User* memilih menu laporan sampah.
5. Sistem menampilkan halaman laporan sampah dan mengambil data dari *database*.
6. Selesai.
7. A*ctivity Diagram* Data Warga



GAMBAR 4. 13 Activity Diagram Data Warga

Keterangan:

1. Pengguna membuka aplikasi

2. Sistem menampilkan form *login*

3. *Input* *username* dan *password*

4. Sistem memvalidasi apakah *username* dan *password* tersebut *valid* atau tidak, jika valid makan sistem akan menampilkan menu *dashboard* jika tidak kembali ke *form* *login.*

5. Selesai

1. *Activity Diagram* Tambah Warga

KkKkkk

KkKkkk

GAMBAR 4. 14 Activity Diagram Tambah Warga

Keterangan:

1. *User* membuka aplikasi

2. Sistem menampilkan form *login*

3. *Input* *username* dan *password*

4. Sistem memvalidasi apakah *username* dan *password* tersebut *valid* atau tidak, jika valid makan sistem akan menampilkan menu *dashboard* jika tidak kembali ke *form* *login.*

5. Selesai

1. *Activity Diagram* Ubah Warga

GAMBAR 4. 15 Activity Diagram Ubah Warga

Keterangan:

1. *User* membuka *website* lalu melakukan *login*

2. Memilih menu warga

3. Pengguna memilih kas yang ingin diubah lalu klik ikon pensil

4. Lalu ubah data yang ingin diubah lalu klik *confirm*

5. Sistem mengubah data di *database* yang telah diubah oleh pengguna

6. Selesai

1. *Activity Diagram* Hapus Warga

GAMBAR 4. 16 *Activity Diagram* Hapus Warga

Keterangan :

1. *User* membuka *website* lalu melakukan *login*

2. Memilih warga

3. Pengguna memilih data warga yang ingin dihapus lalu klik ikon sampah

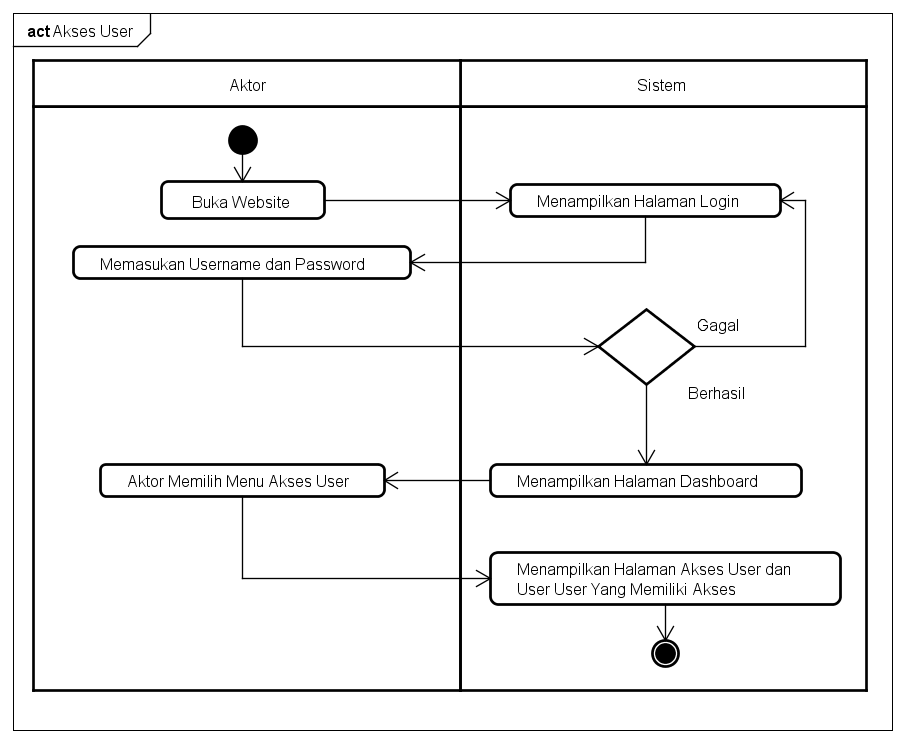
4. Sistem akan menampilkan modal validasi dengan pesan "*Data will deleted permanently"*

5. Sistem akan menghapus data yang dipilih oleh *user* di *database*

6. Selesai.

­­

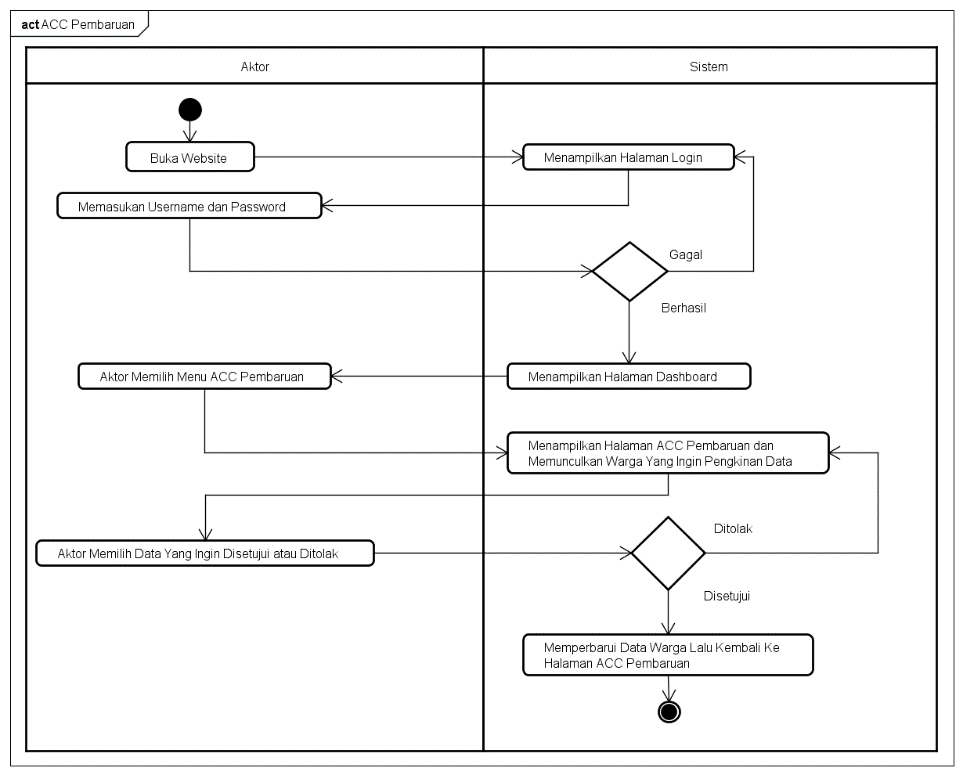
1. *Activity Diagram* Akses *User*



GAMBAR 4. 17 *Activity Diagram* Akses *User*

Keterangan :

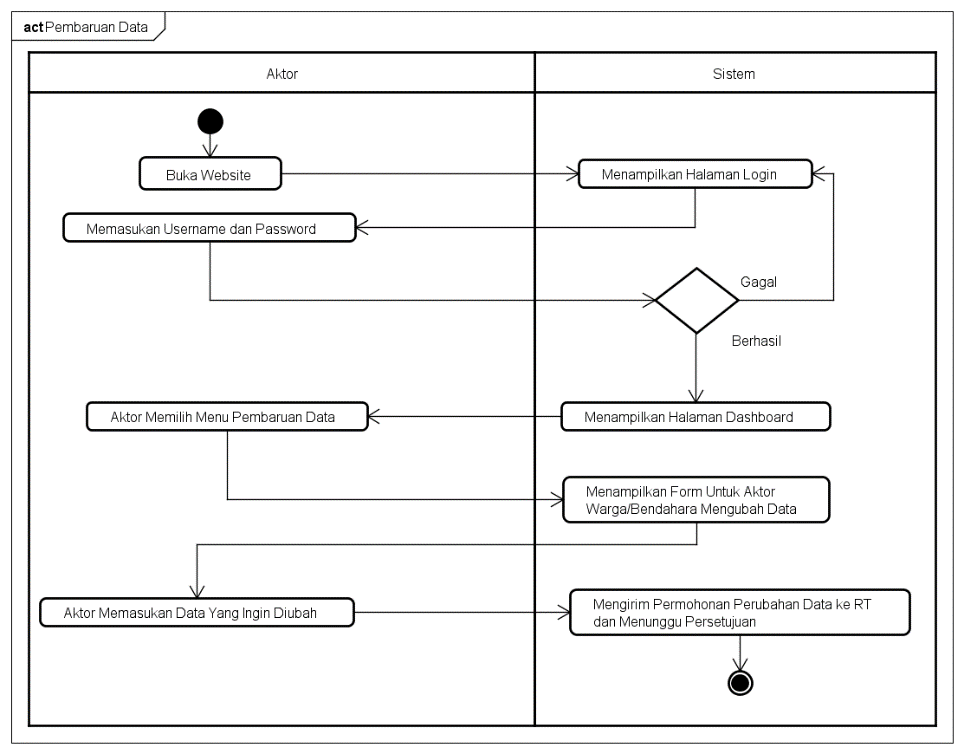
1. *User* membuka *website*, lalu sistem menampilkan halaman *login*.
2. *User* memasukkan *username* dan *password* ke *form login.*
3. Sistem melakukan validasi *login*. Jika data valid maka sistem menampilkan halaman *dashboard*, jika tidak valid maka sistem tetap menampilkan halaman *login*.
4. User memilih menu Akses User dari halaman *dashboard*.
5. Sistem menampilkan halaman Akses *User* beserta daftar user yang memiliki akses.
6. Proses selesai.
7. *Activity Diagram* ACC Pembaruan



GAMBAR 4. 18 *Activity Diagram* ACC Pembaruan

Keterangan :

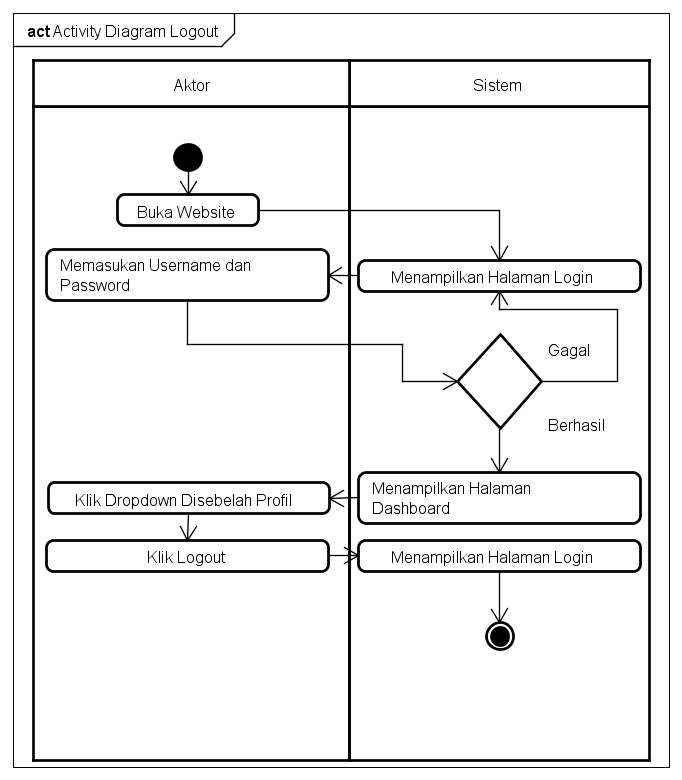
1. User membuka website dan sistem menampilkan halaman login.
2. User memasukkan username dan password ke form login.
3. Sistem memvalidasi login dan menampilkan halaman dashboard jika berhasil.
4. User memilih menu ACC Pembaruan.
5. Sistem menampilkan daftar warga yang mengajukan pembaruan data.
6. User memilih data yang ingin disetujui atau ditolak.
7. Jika disetujui, sistem memperbarui data warga dan kembali ke halaman ACC Pembaruan; jika ditolak, sistem tetap di halaman tersebut.
8. Proses selesai.
9. *Activity Diagram* Pembaruan Data



GAMBAR 4. 19 *Activity Diagram* Pembaruan Data

Keterangan :

1. *User* membuka *website* dan sistem menampilkan halaman *login*.
2. *User* memasukkan *username* dan *password* ke *form* *login*.
3. Sistem memvalidasi *login* dan menampilkan halaman *dashboard* jika berhasil, atau tetap menampilkan halaman *login* jika gagal.
4. *User* memilih menu Pembaruan Data dari halaman *dashboard*.
5. Sistem menampilkan *form* pembaruan data untuk diisi oleh *user* (warga/bendahara).
6. *User* memasukkan data yang ingin diperbarui melalui *form* tersebut.
7. Sistem mengirimkan permohonan pembaruan data ke RT dan menunggu persetujuan.
8. Proses selesai.
9. *Activity Diagram Logout*



GAMBAR 4. 20 *Activity Diagram* Logout

Keterangan :

1. Pengguna membuka *website* dan sistem menampilkan halaman *login*.

2. Pengguna memasukkan *username* dan *password* ke *form login.*

3. Sistem melakukan validasi *login*:

4. Pengguna klik *dropdown* di sebelah profil.

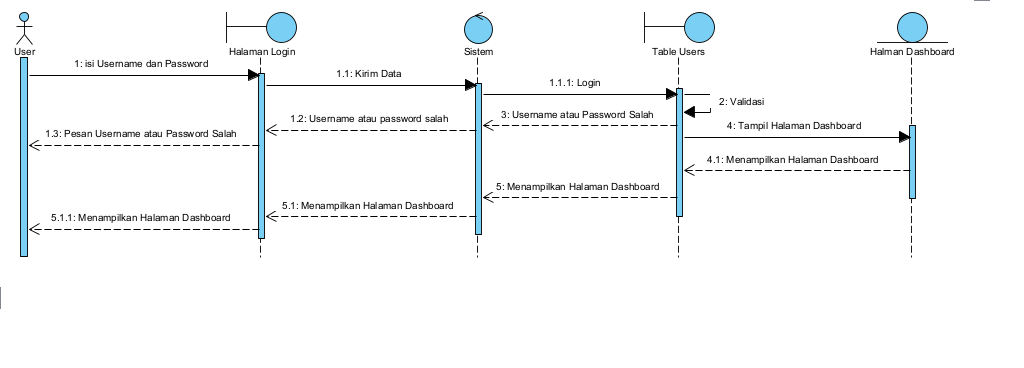
5. Pengguna klik *Logout* pada menu *dropdown*.

6. Sistem akan menampilkan halaman *login* kembali.

7. Proses selesai.

### *Sequence Diagram* yang Diusulkan

1. *Sequence Diagram Login*



GAMBAR 4. 21 *Sequence Diagram Login*

Keterangan :

1. Pengguna membuka aplikasi dan mengakses form *login*. Pengguna memulai dengan membuka aplikasi dan menampilkan halaman *login*.

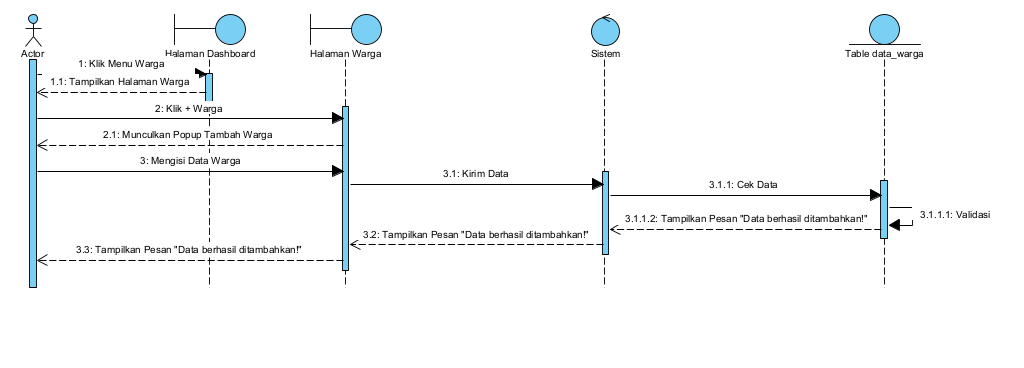
2. Pengguna memasukkan *username* dan *password*. Pengguna mengisi kolom *username* dan *password* lalu mengirimkannya ke sistem melalui halaman *login*.

3. Sistem menerima data dan melakukan proses *login*. Sistem menerima data login dan meneruskannya ke *database* (*Table Users)* untuk melakukan validasi terhadap *username* dan *password* yang dimasukkan.

4. Validasi data *login*, jika *username* atau *password* salah, sistem mengirimkan pesan kesalahan yang ditampilkan kembali ke pengguna, meminta untuk memasukkan ulang data *login*, jika *username* dan *password* benar, sistem melanjutkan proses ke langkah berikutnya.

5. Sistem menampilkan halaman *dashboard*. Setelah *login* berhasil, sistem memuat dan menampilkan halaman *dashboard* kepada pengguna sebagai tampilan utama dari sistem.

1. *Sequence Diagram* Tambah Warga



GAMBAR 4. 15 *Sequence Diagram* Tambah Warga

Keterangan :

1. Aktor memilih menu Warga dari halaman *dashboard*, kemudian sistem akan menampilkan halaman warga.

2. Aktor mengklik tombol + Warga untuk menambahkan data baru, lalu sistem menampilkan popup form tambah warga.

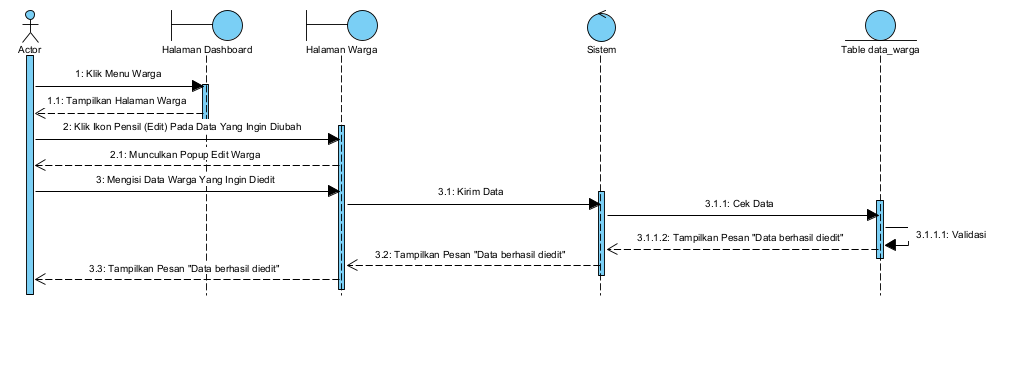
3. Aktor mengisi form data warga yang ingin ditambahkan dan mengirimkan data tersebut.

4. Sistem menerima data yang dikirim, lalu melakukan proses pengecekan dan validasi ke tabel data\_warga.

5. Jika data valid, sistem menyimpan data warga baru tersebut dan menampilkan pesan bahwa data berhasil ditambahkan.

6. Pesan keberhasilan kemudian ditampilkan kembali ke halaman warga sebagai notifikasi kepada aktor.

1. *Sequence Diagram* Ubah Warga



GAMBAR 4. 16 *Sequence Diagram Ubah Warga*

Keterangan :

1. Aktor membuka menu warga melalui halaman *dashboard*, kemudian sistem menampilkan halaman warga.

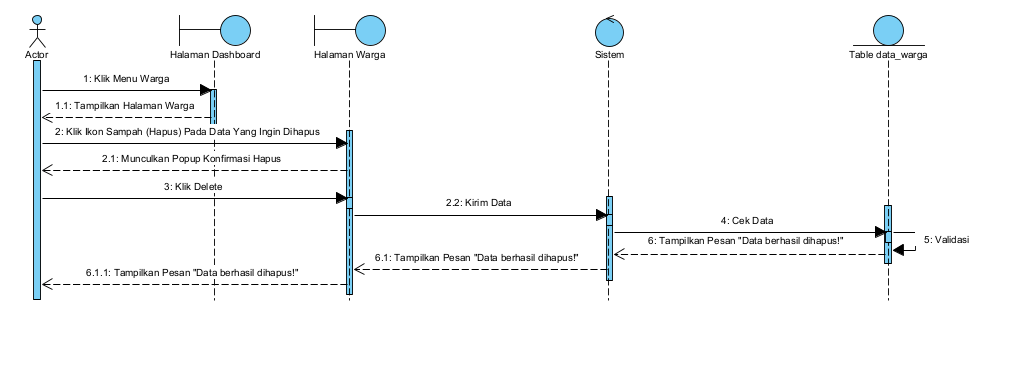
2. Aktor mengklik ikon pensil (*edit*) pada baris data warga yang ingin diubah, lalu sistem menampilkan pop up *form* edit data warga.

3. Aktor mengisi atau memperbarui data pada *form* sesuai data yang ingin diubah, lalu mengirimkan data tersebut ke sistem.

4. Sistem menerima data, kemudian melakukan pengecekan dan validasi ke tabel data\_warga, jika data valid maka sistem memperbarui data yang ada.

5. Sistem menampilkan pesan bahwa data berhasil diedit kepada aktor melalui halaman warga dan *dashboard* sebagai notifikasi keberhasilan.

1. Sequence Diagram Hapus Warga



GAMBAR 4. 17 *Sequence Diagram* Hapus Warga

Keterangan :

1. Aktor membuka menu warga melalui halaman *dashboard*, lalu sistem akan menampilkan halaman warga.

2. Aktor mengklik ikon tempat sampah (hapus) pada baris data warga yang ingin dihapus, sistem kemudian menampilkan pop up konfirmasi penghapusan.

3. Aktor menekan tombol *Delete* pada pop up konfirmasi untuk melanjutkan proses penghapusan data.

4. Sistem menerima perintah penghapusan, lalu melakukan pengecekan dan validasi terhadap data yang dipilih melalui tabel data\_warga.

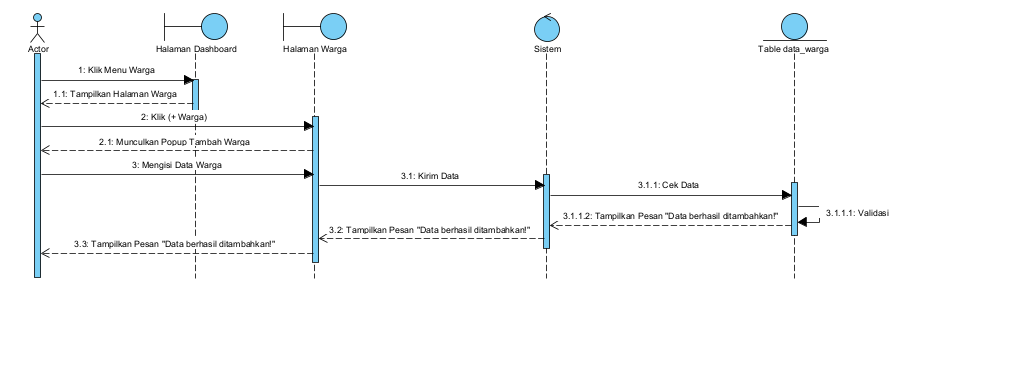
5. Jika valid, sistem menghapus data tersebut dari tabel dan menampilkan pesan bahwa data berhasil dihapus.

6. Sistem mengirimkan pesan tersebut ke halaman warga dan *dashboard* untuk ditampilkan kepada aktor sebagai notifikasi keberhasilan penghapusan.

memiliki kemampuan untuk memasukkan dan keluar dari sistem. Selain itu, warga dapat melihat "Laporan", yang mungkin berisi laporan tentang iuran atau informasi lainnya, dan mengakses "Data Warga", yang memungkinkan mereka untuk melihat data pribadi mereka atau data warga lain yang diizinkan.

berkonsentrasi pada hal-hal yang berkaitan dengan uang. Bendahara memiliki kemampuan untuk "*Login*" dan "*Logout*". Tugas utamanya adalah mengelola "Kas" (menambahkan, mengubah, atau menghapus data kas), melakukan "Pembayaran" (mencatat atau memproses pembayaran), dan melihat berbagai jenis "Laporan", termasuk "Laporan Kas" dan "Laporan Sampah" (kemungkinan terkait dengan iuran kebersihan atau retribusi sampah).

1. *Sequence Diagram* Tambah Kas



GAMBAR 4. 18 *Sequence Diagram* Tambah Kas

Keterangan :

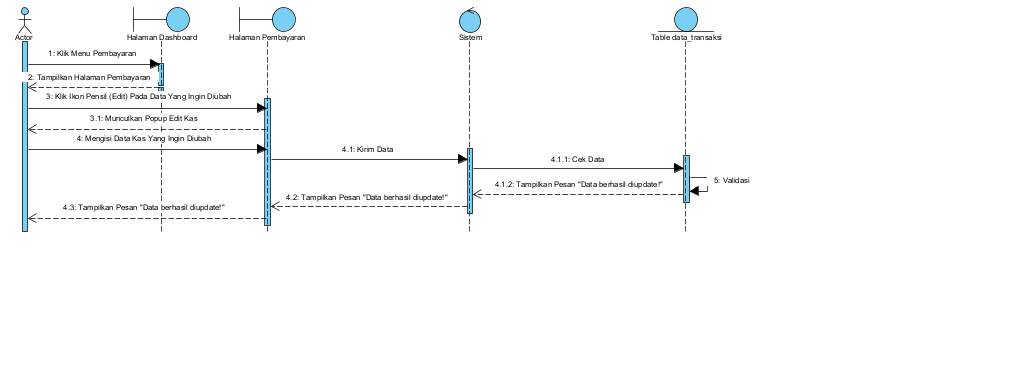
1. Aktor mengklik tombol tambah warga (+ Warga), lalu sistem menampilkan pop up untuk pengisian data warga.

2. Aktor mengisi form data warga dan mengirimkan data tersebut ke sistem.

3. Sistem menerima data, kemudian memeriksa dan memvalidasi data ke tabel data\_warga, jika valid maka data ditambahkan ke tabel.

4. Sistem menampilkan pesan bahwa data berhasil ditambahkan kepada aktor, dan pesan ini juga diteruskan dari halaman warga ke halaman *dashboard* sebagai notifikasi keberhasilan *dashboard*. Setelah login berhasil, sistem memuat dan menampilkan halaman *dashboard* kepada pengguna sebagai tampilan utama dari sistem.

1. *Sequence Diagram* Ubah Kas



GAMBAR 4. 19 *Sequence Diagram* Ubah Kas

Keterangan :

1. Aktor membuka menu pembayaran melalui halaman *dashboard*, kemudian sistem akan menampilkan halaman pembayaran.

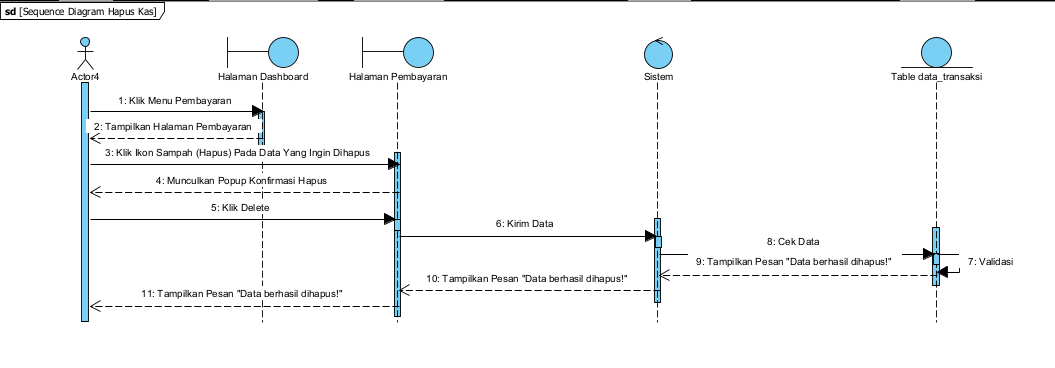
2. Aktor mengklik ikon pensil (edit) pada baris data kas yang ingin diedit, lalu sistem menampilkan pop up *form* edit kas.

3. Aktor mengisi atau memperbarui data kas pada *form* sesuai perubahan yang diinginkan.

4. Aktor mengirimkan data tersebut ke sistem, lalu sistem menerima data dan melakukan pengecekan serta validasi terhadap data tersebut pada tabel data\_transaksi, jika data valid maka sistem memperbarui data yang dimaksud.

5. Sistem menampilkan pesan bahwa data berhasil di *update* kepada aktor, baik melalui halaman pembayaran maupun halaman *dashboard*  sebagai notifikasi keberhasilan.

1. *Sequence Diagram* Hapus Kas



GAMBAR 4. 19 *Sequence Diagram* Hapus Kas

Keterangan :

1. Aktor klik menu Pembayaran.

2. Sistem menampilkan halaman pembayaran.

3. Aktor klik ikon Sampah (Hapus) pada data kas yang ingin dihapus.

4. Sistem menampilkan *popup* konfirmasi hapus.

5. Aktor klik tombol *Delete* pada *popup* konfirmasi.

6. Halaman pembayaran mengirimkan data penghapusan ke sistem.

7. Sistem menerima dan melakukan validasi data ke tabel data\_transaksi.

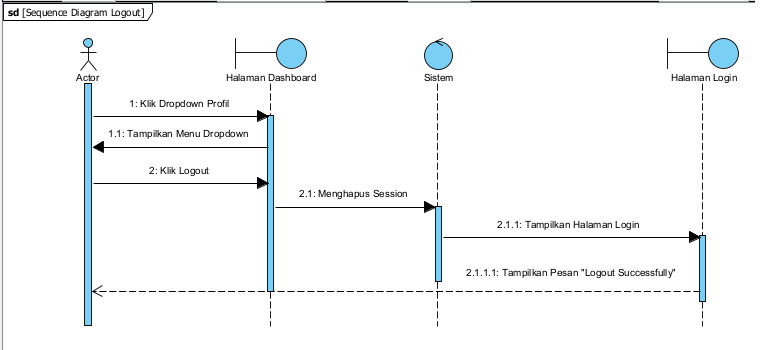
8. Sistem melakukan pemeriksaan data dan memproses penghapusan.

9. Sistem mengirimkan pesan "Data berhasil dihapus" ke halaman pembayaran.

10. Halaman pembayaran menampilkan pesan "Data berhasil dihapus" ke sistem.

11. Aktor melihat notifikasi bahwa data berhasil dihapus.

1. *Sequence Diagram Logout*



GAMBAR 4. 20 *Sequence Diagram Logout*

Keterangan :

1. Aktor membuka menu *drop down* profil.

2. Sistem (halaman *dashboard*) menampilkan menu *dropdown*.

3. Aktor memilih menu *Logout*.

4. Sistem menghapus *session user* yang sedang aktif.

5. Sistem menampilkan halaman *Login*.

6. Sistem menampilkan pesan "*Logout Successfully*" di halaman *Login*.

7. Halaman *Login* beserta pesan ditampilkan kepada aktor.

### *Class Diagram* yang Diusulkan

GAMBAR 4. 21 *Class Diagram* yang Diusulkan

### Rancangan Basis Data