**Sistem Pengaduan Kerusakan Komputer dan rekomendasi solusi menggunakan algoritma cosine similarity**

**SKRIPSI**



**LUH GEDE AYU CANDRAWATI**

**NIM. 1408605018**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS UDAYANA**

**BUKIT JIMBARAN**

**2018**

**SURAT PERNYATAAN KEASLIAN KARYA ILMIAH**

Yang bertanda tangan di bawah ini menyatakan bahwa naskah Skripsi dengan judul:

**Sistem Pengaduan Kerusakan Komputer dan rekomendasi solusi menggunakan algoritma cosine similarity**

Nama : Luh Gede Ayu Candrawati

NIM : 1408605018

Program Studi : Teknik Informatika

E-mail : ayu.candrawati@mhs.unud.ac.id

Nomor telp/HP: 082188686362

Alamat : Jalan palapa XIV gang ikan sardin, Denpasar, Bali

Belum Pernah dipublikasikan dalam dokumen skripsi, jurnal nasional maupun internasional atau dalam prosiding manapun, dan tidak sedang atau akan diajukan untuk publikasi di jurnal atau prosiding manapun. Apabila di kemudian hari terbukti terdapat pelanggaran kaidah-kaidah akademik pada karya ilmiah saya, maka saya bersedia menanggung sanksi-sanksi yang dijatuhkan karena kesalahan tersebut, sebagaimana diatur oleh Peraturan Menteri Pendidikan Nasional Nomor 17 Tahun 2010 tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi.

Demikian Surat Pernyataan ini saya buat dengan sesungguhnya untuk dapat dipergunakan bilamana diperlukan

Denpasar, ……….. 2018

Yang membuat pernyataan

Luh Gede Ayu Candrawati

NIM. 1408605018

# LEMBAR PENGESAHAN TUGAS AKHIR

Judul :

Nama : Luh Gede Ayu Candrawati

NIM : 1408605018

Pembimbing I : I Gede Santi Astawa, S.T., M.Cs

Pembimbing II : Ida Bagus Gede Dwidasmara, S.Kom., M.Cs

# ABSTRAK

**Kata kunci:**

Title :

Name : Luh Gede Ayu Candrawati

Registration : 1408605018

First Supervisor : I Gede Santi Astawa, S.T., M.Cs

Second Supervisor : Ida Bagus Gede Dwidasmara, S.Kom., M.Cs

# ABSTRACT

**Keywords:**

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat rahmat Beliaulah Tugas Akhir yang berjudul “Sistem Pengaduan Kerusakan Komputer Dan Rekomendasi Solusi Menggunakan Algoritma Cosine Similarity” ini dapat diselesaikan sesuai harapan penulis.

Sehubungan dengan telah terselesaikannya tugas akhir ini, maka diucapkan terima kasih dan penghargaan kepada berbagai pihak yang telah membantu penulis, antara lain:

1. Bapak I Gede Santi Astawa, S.T, M.Cs, sebagai Reviewer I yang telah banyak membimbing dan membantu menyempurnakan tugas akhir ini;
2. Bapak Ida Bagus Gede Dwidasmara, S.Kom, M.Cs sebagai Reviewer II yang telah bersedia meluangkan waktunya untuk mengkritisi, mengkoreksi, dan membantu menyempurnakan tugas akhir ini;
3. Bapak – bapak dan Ibu – ibu dosen pengajar di Program Studi Teknik Informatika Fakultas MIPA Universitas Udayana yang telah meluangkan waktu untuk memberikan saran dan masukan dalam menyempurnakan tugas akhir ini;
4. Kawan – kawan di Jurusan Ilmu Komputer yang telah memberikan dukungan moral dalam proses penyelesaian tugas akhir ini.

Disadari pula bahwa sudah tentu tugas akhir ini masih mengandung kelemahan dan kekurangan. Memperhatikan hal ini, maka adanya masukan dan saran – saran penyempurnaan sangat diharapkan**.**

Bukit Jimbaran, 2018

Penulis

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# DAFTAR LAMPIRAN

# BAB I

**PENDAHULUAN**

## Latar Belakang

Dinas pemerintah kota Denpasar merupakan instansi pemerintah yang bertugas untuk melayani masyarakat yang ada di kota Denpasar. Dinas pemerintah kota Denpasar terbagi ke dalam banyak sub dinas dan bidang. Masing-masing bidang tersebut memiliki banyak pekerjaan yang mengharuskan pegawai untuk menggunakan komputer. Perangkat komputer yang digunakan pegawai sering mengalami kerusakan. Kerusakan tersebut meliputi kerusakan hardware, software maupun jaringan. Hal ini menjadi masalah ketika jumlah komputer yang rusak tidak sedikit dan tersebar di berbagai dinas.

Selama ini para pegawai melaporkan perangkat komputer yang mengalami masalah melalui sebuah mekanisme pengaduan kerusakan. Proses pengaduan biasanya dilakukan dengan membuat surat pengaduan yang diproses melalui beberapa tahapan yang rumit dan memakan waktu yang lama. Salah satu faktor penyebab lamanya proses tersebut adalah karena keterbatasan-keterbatasan yang dimiliki pihak technical support di kantor tersebut. Sehingga akan lebih baik apabila ada sebuah sistem yang dapat melakukan proses pengaduan secara online dan sekaligus mampu memberi informasi kerusakan awal ke technical support sehingga pekerjaan mereka menjadi lebih optimal. Hal tersebut tentunya akan membuat proses pengaduan menjadi lebih cepat, efektif dan efisien.

Dalam tujuan untuk membantu pekerjaan pegawai dan technical support maka diperlukan sebuah sistem pengaduan kerusakan komputer yang didalamnya berisi fitur rekomendasi solusi cepat dari kerusakan komputer. Fitur rekomendasi solusi cepat dari kerusakan komputer dibuat dengan menggunakan metode perbandingan kalimat keluhan dengan basisdata histori kerusakan. Beberapa penelitian mengenai pencocokan dua buah teks telah dilakukan. Nurdiana dkk. (2016) melakukan pebandingan antara metode cosine similarity dengan metode jaccard similarity dan tambahan k-nearest neighbor (K-NN) untuk mendukung pencocokan kata yang lebih akurat dalam terjemah Al-Qur’an. Selanjutnya, Firdaus dkk. (2014) melakukan penelitian mengenai pendeteksi kemiripan pada dokumen teks menggunakan algoritma Nazief & Adriani dan metode Cosine Similarity. Wahyuni dkk. (2014) melakukan penerapan algoritma cosine similarity dan pembobotan TF-IDF pada sistem klasifikasi dokumen skripsi.

Pada penelitian ini algoritma cosine similarity akan digunakan pada fitur rekomendasi solusi bagi technical support. Algoritma cosine similarity digunakan pada penelitian ini karena, berdasarkan penelitian yang sudah dilakukan oleh Nurdiana dkk. (2016) algoritma cosine similarity lebih unggul dibanding metode lainnya. Melalui aplikasi ini diharapkan pegawai dan technical support dinas kota Denpasar terbantu, sehingga proses pengaduan kerusakan komputer menjadi lebih cepat, efektif, dan efisien dan kinerja pegawai di dinas Kota Denpasar semakin optimal.

## Rumusan Masalah

Berdasarkan latar belakang tersebut, beberapa rumusan masalah dari penelitian ini adalah:

1. Bagaimana sistem yang dibangun dapat mendukung pekerjaan *technical support* untuk mendapatkan rekomendasi solusi kerusakan komputer yang ada di dinas kota Denpasar
2. Bagaimana implementasi algoritma *cosine similarity* dalam menentukan rekomendasi kerusakan computer

## Tujuan Penelitian

Berdasarkan rumusan masalah tersebut, tujuan penelitian ini adalah sebagai berikut.

1. Membangun sistem pengaduan yang dapat merekomendasi solusi kerusakan komputer yang ada di dinas kota Denpasar
2. Mengimplementasi algoritma *cosine similarity* dalam menentukan rekomendasi kerusakan computer

## Batasan Masalah

Beberapa batasan masalah yang akan dijadikan acuan dalam penelitian ini adalah sebagai berikut.

1. Rekomendasi solusi dari kerusakan komputer menggunakan metode *text similarity* dan algoritma *cosine similarity*
2. Sistem yang dibangun merupakan sistem berbasis web *responsive*.
3. Data solusi kerusakan yang digunakan pada sistem ini yaitu mencakup kerusakan *hardware, software,* dan jaringan.

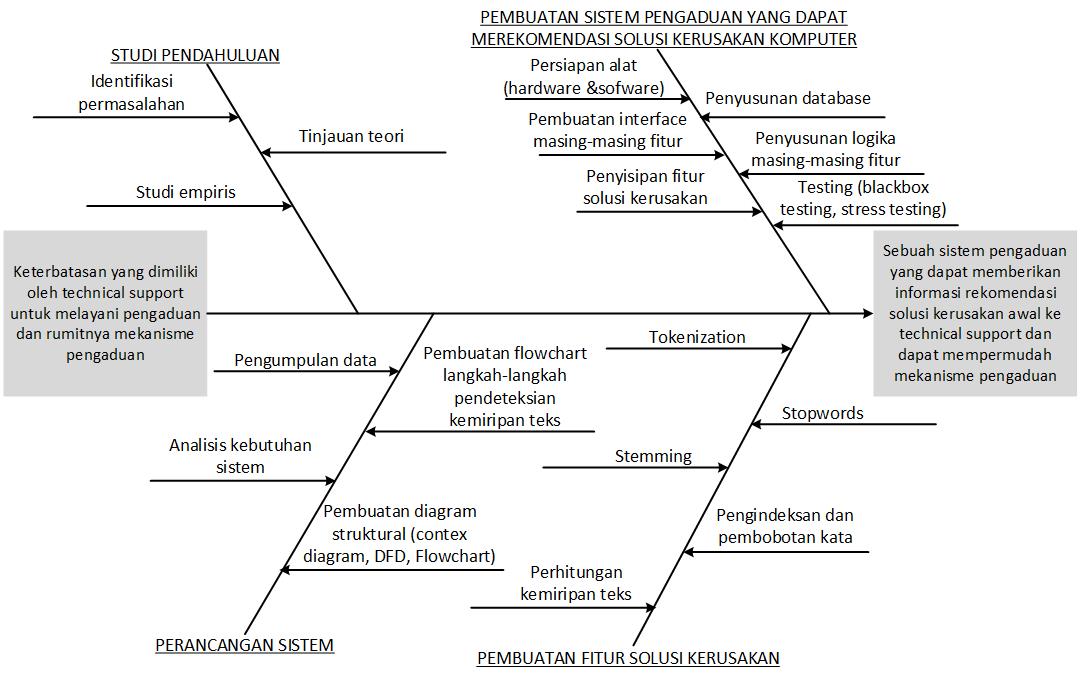
## Manfaat Penelitian

Beberapa manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Bagi *technical support* dinas kota Denpasar, membantu mendapatkan solusi cepat dari kerusakan komputer. Sehingga dapat mempercepat pekerjaan *technical support.*
2. Bagi pegawai dinas kota Denpasar, membantu melakukan pengaduan kerusakan komputer secara *online* agar pekerjaan pegawai menjadi lebih cepat dan efisien.
3. Bagi penulis, diharapkan nantinya dapat dijadikan pedoman dan pertimbangan apabila terdapat pihak lain yang berminat untuk mengembangkan penelitian lebih lanjut mengenai sistem pengaduan kerusakan komputer dan rekomendasi solusi menggunakan algoritma *cosine similarity.*

## Metodologi Penelitian

Langkah-langkah yang akan dilakukan dalam melakukan penelitian Sistem Pengaduan Kerusakan Komputer dan Rekomendasi Solusi Menggunakan Algoritma *Cosine Similarity.* Langkah-langkah tersebut akan digambarkan pada diagram *fishbone* seperti pada Gambar x.x



Gambar x. x Diagram *Fish Bone* Penelitian

### Metode Pengumpulan Data

Tahap ini pengumpulan data diperoleh dari metode studi pustaka atau data sekunder bersumber pada buku dan juga data primer dari para ahli yang dikumpulkan oleh peneliti sendiri. Data yang digunakan pada penelitian ini adalah berupa data dari berbagai macam solusi permasalahan mengenai kerusakan komputer. Data yang bersumber dari buku adalah buku mengenai solusi kerusakan komputer yang berisi gejala, kemungkinan penyebab, dan solusi dari berbagai jenis kerusakan komputer baik dari segi *hardware, software,* dan jaringan. Data ini berguna bagi bank data solusi kerusakan komputer. Data yang berasal dari para ahli dilakukan dengan cara mewawancara teknisi komputer, kemudian hasil dari wawancara tersebut diinputkan ke bank data solusi kerusakan komputer. Semua data yang digunakan pada penelitian ini berbahasa Indonesia. Pada sistem pengaduan kerusakan komputer, data yang ada pada tabelsolusi kerusakan komputer terdapat atribut gejala, atribut kemungkinan penyebab, dan atribut solusi. Atribut gejala berisi ciri-ciri dari masing-masing kerusakan, atribut penyebab berisi penyebab-penyebab dari masing-masing kerusakan, dan atribut solusi berisi solusi dari kerusakan komputer.

### Metode Pengembangan Sistem

Metode yang digunakan pada penelitian ini adalah metode pengembangan perangkat lunak model *waterfall.* Model *waterfall* dipilih karena model ini melakukan pendekatan secara sistematis dan berurutan. Model ini juga memungkinkan untuk departementalisasi dan kontrol. *One by one* adalah proses pengembangan model ini, sehingga dapat meminimalisir kesalahan-kesalahan yang mungkin terjadi.

Model *waterfall* terdiri dari 5 tahapan yang dijelaskan pada Gambar x.x yaitu communication (Project Initiation & Requirements Gathering), planning (Estimating, Scheduling, Tracking), modeling (Analysis & Design), construction (Code & Test), deployment (Delivery, Support, Feedback). Berikut merupakan gambaran umum pengembangan sistem menggunakan model *waterfall.*



Gambar x. x Metode *Waterfall* (Pressman, 2015)

### Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan dengan mencari informasi mengenai kebutuhan sistem dan data yang akan dibangun. Dalam mencari informasi, dilakukan dengan mewawancara pegawai dinas yang berperan sebagai user pada sistem pengaduan. Hal tersebut dilakukan untuk mengetahui kebutuhan pegawai dinas dalam melakukan pengaduan dan pengelolaan sistem pengaduan yang diperankan oleh admin, *technical support* untuk mendapatkan solusi kerusakan, dan kepala dinas untuk rekap pengaduan. Dari hasil pencarian informasi yang dilakukan dengan wawancara, maka didapatkan hasil analisis kebutuhan yang dibedakan menjadi kebutuhan fungsional dan kebutuhan non fungsional.

### Perancangan Sistem

Tahap perancangan sistem bertujuan untuk menerapkan solusi dari permasalahan yang terdapat pada analisis sistem. Sistem yang dibangun berdasarkan teknik pemrograman berorientasi objek sehingga sistem dirancang dengan diagram UML *(Unified Modeling Language).* Diagram UML terdiri dari *Use Case Diagram, Activity Diagram, Class Diagram,* dan *Sequence Diagram.* Untuk menggambarkan hubungan antar entitas dan basis data sistem menggunakan diagram ERD *(Entity Relationship Diagram).* Tahap perancangan sistem juga dilakukan perancangan tampilan antarmuka atau *interface*  dari sistem.

### Implementasi Sistem

Tahap ini dilakukan dengan mengimplementasi hasil dari sistem yang telah dirancang sebelumnya kedalam bentuk kode program. Adapun komponen pendukung dalam mengimplementasi sistem adalah :

1. Bahasa pemrograman yang digunakan yaitu HTML, CSS, *Java Script,* PHP, dan menggunakan *framework* CodeIgniter
2. Server yang digunakan berbasis lokal yaitu menggunakan Localhost
3. Aplikasi SQLyog digunakan dalam mengolah *database*

### Pengujian Sistem

Tahap pengujian pada sistem ini yaitu dilakukan dengan cara menguji sesuai atau tidaknya proses *requirement* dengan kebutuhan*.* Terdapat tiga tahap pengujianyang dilakukan pada sistem ini yaitu dengan menggunakan metode *Black Box Testing*, *Stress Testing,* dan respon pengguna.

# BAB II

**TINJAUAN PUSTAKA**

## Tinjauan Studi

Dalam melakukan penelitian ini, penulis menggunakan beberapa tinjauan studi yang merupakan penelitian terdahulu yang pernah dilakukan sebelumnya yang terkait dengan sistem pengaduan dan algoritma *cosine similarity*.

1. **Sistem Pengaduan Masyarakat Pada Divisi Humas Polri Berbasis Web**

(Fajar Masya, Elvina, Fitri Maria Simanjuntak, 2012)

Masya dkk. (2012) membangun sebuah sistem pengaduan masyarakat pada Divisi Humas Polri berbasis web. Sistem ini dibuat untuk mempermudah masyarakat untuk menyampaikan pengaduan dan permohonan informasi, serta mempercepat pihak Divisi Humas Polri untuk merespon setiap pengaduan dan permohonan informasi tersebut. Hasil akhir dari penelitian ini adalah rancangan dan implementasi Sistem Pelayanan Pengaduan Masyarakat Berbasis Web. Dalam web ini masyarakat dapat mengisi langsung form pengaduan, melihat apa yang menjadi atensi dari pimpinan Polri, mengakses berita serta foto terbaru yang berkaitan dengan Polri. Selain itu juga, masyarakat bisa meminta langsung informasi yang berkaitan dengan Polri seperti kasus-kasus yang ditangani oleh Polri dan melihat langsung jawaban dari informasi apa yang diminta. Masyarakat juga bisa melihat langsung pengumuman yang ada di Polri seperti pengumuman penerimaan PNS, dan lain-lain.

1. **Prototype Aplikasi Layanan Pengaduan Masyarakat Berbasis Android Dan Web Service**

(Andi Jumardi, Achmad Solichin, 2016)

Jumardi dkk. (2016) melakukan penelitian mengenai prototype aplikasi layanan pengaduan masyarakat berbasis android dan web service. Berdasarkan penelitian, aplikasi ini dapat digunakan masyarakat untuk menyampaikan keluhan-keluhannya terhadap masalah sampah yang ada di lingkungan sekitarnya. Hasil pengujian dengan aplikasi stress tool versi 8 untuk menguji ketahanan dan kekuatan sistem menunjukkan hasil yang cukup baik dengan kecepatan akses terhadap web service dengan rata-rata sebesar 819 ms.

Setelah dijabarkan mengenai penelitian sebelumnya yang terkait dengan sistem pengaduan, bedanya dengan penelitian ini adalah sistem pengaduan yang ditangani merupakan pengaduan untuk kerusakan komputer untuk Dinas Kota Denpasar. Dimana pengaduan kerusakan komputer dilakukan oleh pegawai dinas, yang nantinya pengaduan tersebut dikelola oleh admin. Salah satu fitur yang terdapat pada sistem ini yaitu fitur rekomendasi solusi kerusakan komputer. Fitur ini dibuat dengan mencocokan kemiripan kata. Proses pencocokan kata dilakukan dengan algoritma *cosine similarity.* Pencocokan kata akan dihitung dari kata yang diinputkan *user* dengan kata yang ada pada tabel solusi kerusakan komputer. Masing-masing dari solusi kerusakan komputer yang ada pada tabel akan menghasilkan nilai persentase dan hasil *output* solusi kerusakan komputer diurutkan berdasarkan persentase yang tertinggi. Adapun penelitian yang terkait dengan algoritma *cosine similarity* :

1. **Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani Dan Metode Cosine Similarity**

(Azhar Firdaus, Ernawati, dan Arie Vatresia, 2014)

Firdaus dkk. (2014) melakukan penelitian mengenai pendeteksi kemiripan pada dokumen teks. Pada penelitian ini menghasilkan aplikasi yang dapat mendeteksi kemiripan pada dokumen teks menggunakan algoritma Nazief & Adriani dan metode Cosine Similarity. Dengan menggunakan algoritma Nazief & Adriani dan metode Cosine Similarity aplikasi dapat membandingkan berkas berbeda ekstensi dan membandingkan lebih dari dua dokumen secara bersamaan. Untuk mengoptimasi nilai kemiripan yang diperoleh dari metode Cosine Simiarity digunakan algoritma Nazief & Adriani. Berdasarkan pengujian, hasil Perhitungan tanpa menggunakan algoritma Nazief & Adriani sebesar 87,83%. Dan hasil Perhitungan dengan menggunakan algoritma Nazief & Adriani adalah sebesar 93,81%.

1. **Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Apliksi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia**

(Ogie Nurdiana, Jumadi, dan Dian Nursantika 2016)

Nurdiana, dkk. (2016) mengimplementasikan *text mining* menggunakan perbandingan metode *cosine similarity* dengan *metode jaccard similarity* dan tambahan k-nearest neighbor (K-NN) untuk mendukung pencocokan kata yang lebih akurat dalam terjemah Al-Qur’an. Pada penelitian ini proses klasifikasi dengan hasil akhir dari percobaan 33 kali dengan *key* yang berbeda dan total 6326 dokumen. Metode *cosine similarity* mendapatkan nilai kemiripan yang tertinggi yaitu 44%. Sedangkan metode *jaccard similarity* mendapatkan nilai 19% dan *k-nearest neaighbor (KNN)* mendapatkan nilai 40%. Metode *cosine similarity* lebih unggul dibanding metode lainnya karena pada metode *cosine similarity* mempunyai konsep normalisasi panjang vektor data dengan membandingkan N-gram yang sejajar satu sama lain dari 2 pembanding. Sedangkan pada metode jaccard hanya membandingkan isi N-gram dengan eksak dan hanya melihat apakah ada suatu N-gram tertentu pada pembanding tanpa melihat posisi penulisan yang berbeda. Pada euclidean distance yang diterapkan di metode k-nearest neighbor (K-NN) tidak mempunyai konsep normalisasi panjang vektor data, sehingga nilai akurasi metode dipengaruhi oleh panjang 2 data pembanding dan harus menentukan nilai dari pamameter K (jumlah dari tetangga terdekat). Oleh karena itu metode *cosine similarity* dapat menjadi altenatif untuk mencari kemiripan *text mining*.

1. **Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi**

(Rizki Tri Wahyuni, Dhidik Prastiyanto, dan Eko Supraptono, 2017)

Wahyuni dkk. (2017) melakukan penerapan algoritma *cosine similarity* dan pembobotan TF-IDF pada sistem klasifikasi dokumen skripsi. Penelitian ini dapat melakukan klasifikasi dokumen secara otomatis untuk sistem *collecting file* skripsi. Hasil pengujian kelayakan sistem mendapatkan hasil persentase kelayakan sebesar 88.3%. Dan berdasarkan hasil klasifikasi yang dilakukan oleh sistem adalah sebesar 98%.

Berdasarkan hasil studi empiris yang terdapat pada penelitian sebelumnya, didapatkan bahwa sistem pengaduan online dapat membantu mempermudah seseorang dalam penyampaian keluhan. Untuk mendeteksi kemiripan kata, c*osine similarity* dapat menjadi solusi agar dapat mempermudah proses menghitung kemiripan kata yang ada dalam dokumen.

## Sistem Informasi Menajemen

Manajemen adalah proses atau kegiatan yang dilakukan oleh seseorang atau pemimpin atau manajer di dalam organisasi untuk mencapai tujuan bersama. Secara operasional dapat didefinisikan sebagai proses mengkoordinasikan, mengintegrasikan, menyederhanakan dan mensinkronisasikan sumber daya manusia, material dan metode *(Men, Material, Methods)* dengan mengaplikasikan fungsi-fungsi manajemen seperti, perencanaan, pengorganisasian, penggiatan, pengawasan dan lain-lain agar tujuan organisasi dapat tercapai secara efektif dan efisien (Winarno, 2004). Menurut (Abdul Kadir, 2003) definisi sistem informasi manajemen adalah sistem informasi yang digunakan untuk menyajikan informasi yang digunakan untuk mendukung operasi, manajemen, dan pengambilan keputusan dalam sebuah organisasi. Biasanya SIM menghasilkan informasi untuk memantau kerja, memelihara koordinasi, dan menyediakan informasi untuk operasional organisasi yang dilakukan secara komputer misalnya perminggu, perbulan dan pertahun, tidak secara aktivitas per hari.

## Pengaduan

Menurut (Kamus Besar Bahasa Indonesia, 2008) makna dari pengaduan adalah ungkapan rasa tidak senang atau tidak puas akan hal-hal yang tidak begitu penting, tetapi perlu diperhatikan. Pengaduan bisa terjadi dimana saja dan kapan saja. Biasanya proses pengaduan pada suatu instansi adalah melalui beberapa tahapan yang rumit dan memakan waktu yang cukup lama. Lain hal nya dengan pengaduan yang dilakukan secara *online.* Melalui pengaduan *online* proses pengaduan dapat menjadi lebih mudah dan cepat.

Pada sistem ini pengaduan *online* diterapkan pada kasus pengaduan kerusakan komputer. Penanganan pengaduan pada sistem ini adalah dengan cara mengirim pengaduan ke dinas pusat atau *technical support* memperbaiki kerusakan dengan meminta sistem untuk merekomendasikan solusi dari kerusakan komputer. Apabila *user* ingin melakukan pengaduan caranya adalah dengan mengisi *form* pengaduan yang disediakan oleh sistem. Kemudian sistem akan menyimpan pengaduan pada *database* sistem. *Technical support* akan menangani kerusakan komputer dengan meminta sistem meberikan solusi kerusakan. Apabila *technical support* tidak sanggup memperbaiki maka admin akan mengirim pengaduan ke dinas pusat.

## Preprocessing Data

*Preprocessing* adalah proses pengubahan bentuk data yang terstruktur sembarang menjadi data yang terstruktur sesuai kebutuhan untuk proses dalam t*ext mining*. Tahap preprocessing terdiri dari *tokenizing, stopwords,* dan *stemming*. (Kurniawan, 2014)

1. *Tokenization* adalah proses memecah kalimat yang ada pada dokumen menjadi kata sehingga mendapatkan hasil akhir berupa *tokens* dan jumlah frekuensi *tokens* yang terdapat pada suatu dokumen. Proses ini juga menghilangkan karakter-karakter seperti tanda baca dan mengubah semua *tokens* mengjadi huruf kecil.
2. *Stopwords* adalah proses menghapus kata-kata tidak penting dari hasil *tokenization* dengan cara mencocokan kata tersebut dengan kata yang terdapat pada daftar kata tidak penting (*stoplist*).
3. *Stemming* adalah proses menghilangkan imbuhan, awalan, dan akhiran sehingga menjadi kata dasar dengan menggunakan aturan-aturan tertentu. Dalam penelitian ini, stemming akan dilakukan dengan algoritma Nazief-Adriani. (Agusta, 2009)

## TF-IDF *(Term Frequency- Inversed Document Frequency)*

Menurut (Fitri, 2013) TF-IDF yaitu perhitungan yang menggambarkan seberapa pentingnya kata *(term)* dalam sebuah dokumen. Proses ini digunakan untuk menilai bobot relevansi *term* dari sebuah dokumen terhadap seluruh dokumen. *Term frequency* adalah ukuran seringnya kemunculan sebuah *term* dalam sebuah dokumen. IDF merupakan banyaknya istilah tertentu dalam keseluruhan dokumen. Metode TF-IDF menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut (Nurjanah, 2013).

Rumus untuk TF-IDF :

(1)

(2)

(3)

Keterangan:

D = dokumen ke-d

t = *term* ke-t dari dokumen

W = bobot dokumen ke-d terhadap *term* ke-t

tf = banyaknya *term* i pada sebuah dokumen

idf = *Inversed Document Frequency*

df = banyak dokumen yang mengandung *term* i

## Cosine Similarity

Menurut (Pradnyana, 2012) *cosine similarity* adalah salah satu metode dalam menentukan nilai kemiripan antar dua objek, yang dinyatakan dalam dua buah vektor dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. *Cosine Similarit*y dapat diterapkan dalam menentukan nilai kemiripan pada dua berkas dokumen teks. Parameter yang digunakan adalah jumlah kata-kata pada dua dokumen teks yang dibandingkan. *Cosine Similarity* menggunakan dua vektor yang mempresentasikan dua dokumen teks dimana nilai sudut kosinus dari kedua vektor tersebut adalah nilai kemiripan dari dua dokumen teks tersebut. Batasan nilai yang dihasilkan mulai dari 0 sampai dengan 1. Komposisi kata pada dokumen teks merupakan penentukan nilai yang diperoleh dari *Cosine Similarity* (Firdaus, 2014).

Rumus untuk *cosine similarity* :

(4)

Keterangan :

= bobot istilah *j* pada dokumen

= bobot istilah *j* pada dokumen

## Model Pengembangan Waterfall

Metode yang digunakan pada penelitian ini adalah metode pengembangan perangkat lunak model *Waterfall.* Model *waterfall* dipilih karena model ini melakukan pendekatan secara sistematis dan berurutan. Menurut Pressman (2002), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Diberi nama waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Tahapan dalam Waterfall Model:

1. Communication (Project Initiation & Requirements Gathering)

Tahap ini adalah melakukan komunikasi dengan user untuk memahami dan mencapai tujuan yang ingin dicapai. Setelah berkomunikasi dengan user maka akan menghasilkan inisialisasi proyek seperti menganalisis kebutuhan user dan mengumpulkan data yang diperlukan (data gathering). Analilsis kebutuhan dilakukan dengan mengumpulkan masalah yang terjadi di lapangan. Masalah dikumpulkan dengan cara berkomunikasi dengan *user* sistem. Beberapa masalah yang ditemukan adalah sebagai berikut:

* 1. Rumit dan lamanya proses pengaduan
  2. Keterbatasan *technical support*

Dari beberapa masalah yang sudah didapat, dilakukan analisis sehingga menghasilkan beberapa kebutuhan fungsional :

1. Sistem yang mampu memproses pengaduan sehingga lebih mudah dan cepat
2. Sistem yang mampu memberikan solusi kerusakan komputer berdasarkan pengaduan yang diinput oleh *user*
3. Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko- resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

1. Modeling (Analysis & Design)

Tahap ini adalah merancang dan pemodelan arsitektur sistem untuk memahami gambaran mengenai sistem yang akan dibangun. Dalam merancang dan pemodelan arsitektur sistem dapat dilakukan dengan membuat *Contex Diagram, Data Flow Diagram, Entity Relationship Diagram,* dan *Model Database Diagram*.

1. Construction (Code & Test)

Tahapan ini adalah proses membangun sistem yang sesuai dengan rancangan yang dibuat pada tahap sebelumnya. Dalam proses pembuatan sistem digunakan bahasa berbasis web seperti PHP, MySQL, dan Javascript. Setelah sistem selesai dibangun maka dilakukan pengujian terhadap sistem. Sistem diuji dengan cara *black box testing* untuk kebutuhan fungsional dan *stress testing* untuk mengetahui kemapuan maksimum sistem.

1. Deployment (Delivery, Support, Feedback)

Tahapan deployment merupakan tahapan implementasi sistem dimana dengan *user* menggunakan sistem agar dapat menilai kesesuaian sistem terhadap kebutuhan mereka. Pemeliharaan meliputi perbaikan sistem ketika ditemukannya kesalahan dalam pengujian sistem.



Gambar 7. 4 Metode *Waterfall* (Pressman, 2015)

# BAB III

**ANALISIS DAN PERANCANGAN**

## Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem dilakukan dengan mencari informasi mengenai kebutuhan sistem dan data yang akan dibangun. Pada pencarian informasi, dilakukan dengan mewawancara pegawai dinas yang nantinya akan menjadi user pada sistem untuk mengetahui permasalahan dalam melakukan pengaduan, pengelolaan pengaduan, rekap pengaduan, dan solusi kerusakan komputer yang dirujuk pada tabel x.x. Berdasarkan wawancara tersebut, maka didapatkan hasil untuk analisis kebutuhan sistem yang dibedakan menjadi kebutuhan fungsional dan kebutuhan non fungsional.

**Tabel x.x Analisis Kebutuhan Sistem**

|  |  |
| --- | --- |
| No | Kebutuhan |
| 1 | Sistem pengaduan berbasis web dapat melakukan pengaduan secara *onlilne* |
| 2 | Sistem dapat mengirimkan pengaduan berupa surat melaui *email* |
| 3 | *Technical support* mendapatkan solusi kerusakan komputer |
| 4 | Kepala dinas mendapatkan laporan rekap pengaduan |
| 5 | Fitur solusi kerusakan komputer menggunakan algoritma *Cosine Similarity* |

### Kebutuhan Fungsional

Kebutuhan fungsional berisi mengenai kebutuhan-kebutuhan yang akan dipenuhi oleh sistem. Kebutuhan fungsional akan digunakan sebagai acuan dalam melakukan proses *black box testing* yang terdapat pada Lampiran x. Adapun rincian dari kebutuhan fungsional sistem yaitu :

**Tabel x.x Kebutuhan Fungsional Sistem**

|  |  |  |
| --- | --- | --- |
| **Kode** | **Pengguna** | **Deskripsi Kebutuhan** |
| KF1 | Pegawai, *technical support*, admin, kepala dinas | Sistem dapat melakukan proses login dan logout dengan menggunakan *username* dan *password* yang dimiliki |
| KF2 | Pegawai | Sistem dapat melakukan pengaduan secara *online* dengan menyediakan form pengaduan |
| KF3 | Pegawai | Pegawai mendapatkan informasi mengenai *history* pengaduan yang pernah dilakukan |
| KF4 | Pegawai | Pegawai mendapatkan informasi mengenai status pengaduan, apakah pengaduan belum diproses atau sudah diproses |
| KF5 | *Technical support* | *Technical support* mendapatkan solusi kerusakan komputer berdasarkan data pengaduan yang diinput oleh user pada halaman list pengaduan |
| KF6 | *Technical support* | *Technical support* dapat mengelola daftar kerusakan dengan rincian :   * 1. *Technical support* dapat menambah kerusakan   2. *Technical support* dapat melihat daftar kerusakan   3. *Technical support* dapat mengedit kerusakan   4. *Technical support* dapat menghapus kerusakan |
| KF7 | *Technical support* | *Technical support* dapat mengelola list solusi dengan rincian :   1. *Technical support* dapat menambah solusi 2. *Technical support* dapat melihat list solusi 3. *Technical support* dapat mengedit solusi 4. *Technical support* dapat menghapus solusi |
| KF8 | Admin | Admin dapat mengelola data pengaduan dengan rincian :   1. Admin dapat menambah pengaduan 2. Admin dapat melihat informasi pengaduan 3. Admin dapat melakukan *update* pengaduan dengan cara mengedit pengaduan 4. Admin dapat melakukan hapus pengaduan 5. Admin dapat mengirim pengaduanberupa surat melalui *email* 6. Admin dapat mencetak pengaduan 7. Admin dapat menari pengaduan 8. Admin dapat melihat grafik pengaduan |
| KF9 | Admin | Admin dapat mengelola daftar user dengan rincian :   1. Admin dapat melihat daftar user 2. Admin dapat menambah user 3. Admin dapat mengedit user 4. Admin dapat menghapus user |
| KF10 | Admin | Admin dapat mengelola daftar instansi dengan rincian :   1. Admin dapat melihat daftar instansi 2. Admin dapat menambah instansi 3. Admin dapat mengedit instansi 4. Admin dapat menghapus instansi |
| KF11 | Admin | Admin dapat mengelola profile admin dengan rincian :   * 1. Admin dapat mengubah nama profile   2. Admin dapat mengubah alamat email profile   3. Admin dapat mengganti foto profile |
| KF12 | Kepala dinas | Kepala dinas mendapatkan laporan rekap pengaduan dengan rincian :   1. Kepala dinas mendapatkan informasi pengaduan berdasarkan tahun dan bulan yang dipilih 2. Kepala dinas dapat mencetak pengaduan berdasarkan tahun dan bulan yang dipilih 3. Kepala dinas dapat melihat grafik pengaduan berdasarkan instansi dinas, kategori kerusakan, dan status pengaduan |

### Kebutuhan Non Fungsional

Kebutuhan non fungsional sistem terdiri dari segi *user friendly,* kecepatan dan keamanan yang dijelaskan pada Tabel x.x

**Tabel x.x Kebutuhan Non Fungsional Sistem**

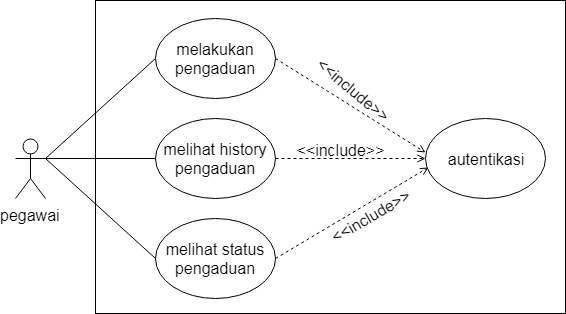
|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan Non Fungsional** | **Deskripsi** |
| 1 | *User Friendly* | Tampilan antar muka sistem harus mudah dimengerti oleh pegawai, *technical support*, admin, dan kepala dinas sebagai pengguna sistem. Penempatan fitur-fitur yang ada pada sistem diatur agar penampilan sistem menjadi lebih menarik. |
| 2 | Kecepatan | Sistem yang dibuat harus dapat memberikan kecepatan pada setiap proses-proses yang ada pada sistem, sehingga dapat memudahkan pengguna sistem. |
| 3 | Keamanan | Sistem yang dibuat harus dapat menjaga keamanan data yang ada pada database, sehingga pada sistem terdapat autentikasi sebelum mengakses halaman pada sistem. |

## Perancangan Sistem

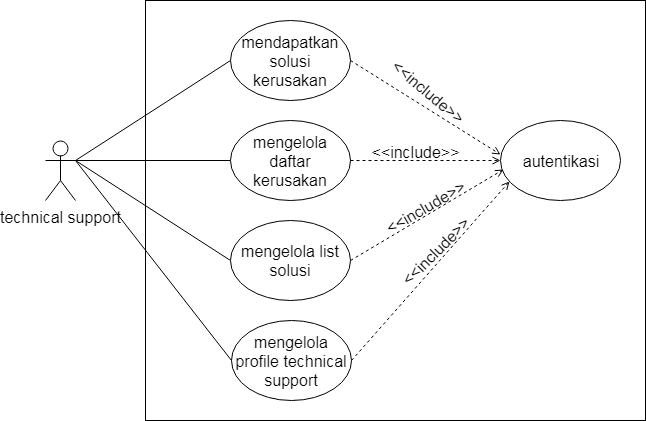
Sistem pengaduan dibangun dengan berbasis berorientasi objek sehingga sistem dirancang menggunakan diagram UML *(Unified Modeling Language)* antara lain *Use Case Diagram, Activity Diagram, Class Diagram,* dan *Sequence Diagram.*

### Use Case Diagram

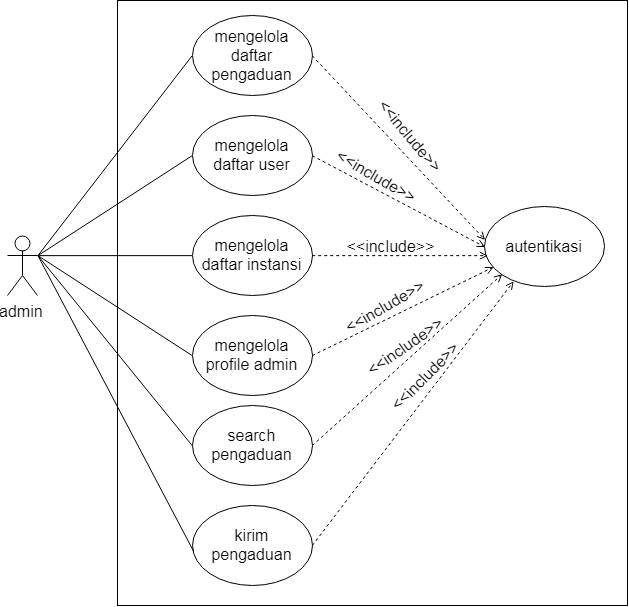
Usecase diagram menampilkan hubungan antara aktor (pelaku) dengan hal-hal yang terlibat pada sistem.



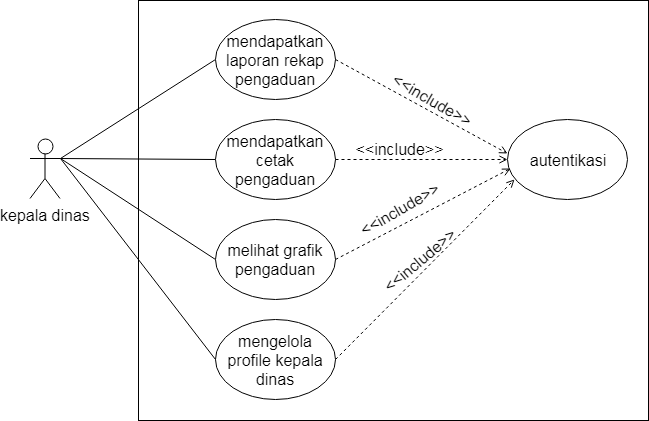
**Gambar x.x *Use Case Diagram* Pegawai dalam Sistem Pengaduan**



**Gambar x.x *Use Case Diagram Technical Support* dalam Sistem Pengaduan**



**Gambar x.x *Use Case Diagram* Admin dalam Sistem Pengaduan**

****

**Gambar x.x *Use Case Diagram* Kepala Dinas dalam Sistem Pengaduan**

Definisi aktor pada Use Case Diagram Gambar 3.1, 3.2, 3.3, 3.4 akan di jelaskan pada Tabel x.x.

**Tabel x.x Definisi Aktor Pada *Use Case* Diagram**

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Pegawai | Orang yang akan melakukan pengaduan pada sistem pengaduan |
| 2 | *Technical Support* | Orang yang akan mendapatkan solusi kerusakan dari pengaduan yang dilakukan oleh pegawai pada sistem pengaduan |
| 3 | Admin | Orang yang bertugas mengelola data pengaduan yang terdapat pada sistem pengaduan seperti tambah, edit, dan hapus data |
| 4 | Kepala Dinas | Orang yang akan mendapatkan laporan rekap pengaduan |

Definisi Use Case pada Use Case Diagram Pegawai Sistem Pengaduan ditujukan pada Tabel x.x.

**Tabel x. x Definisi *Use Case* pada *Use Case* Diagram Pegawai**

|  |  |  |
| --- | --- | --- |
| **No** | ***Use Case*** | **Deskripsi** |
| 1 | Autentikasi | Proses pengecekan dan validasi oleh sistem kepada pengguna sistem |
| 2 | Melakukan pengaduan | Proses untuk melakukan pengaduan pada sistem |
| 3 | Melihat *history* pengaduan | Proses untuk melihat *history* pengaduan yang ada pada sistem |
| 4 | Melihat status pengaduan | Proses untuk melihat status pengaduan yang ada pada sistem |

Definisi Use Case pada Use Case Diagram *Technical Support* Sistem Pengaduan ditujukan pada Tabel x.x.

**Tabel x. x Definisi *Use Case* pada *Use Case* Diagram *Technical Support***

|  |  |  |
| --- | --- | --- |
| **No** | ***Use Case*** | **Deskripsi** |
| 1 | Autentikasi | Proses pengecekan dan validasi oleh sistem kepada pengguna sistem |
| 2 | Mendapatkan solusi kerusakan | Proses untuk mendapatkan solusi kerusakan komputer pada sistem |
| 3 | Mengelola daftar kerusakan | Proses untuk mengelola daftar kerusakan yang terdapat pada sistem pengaduan seperti tambah, edit, dan hapus data |
| 4 | Mengelola list solusi | Proses untuk mengelola list solusi yang terdapat pada sistem pengaduan seperti tambah, edit, dan hapus data |
| 5 | Mengelola profile *technical support* | Proses untuk mengelola profile *technical support* yang terdapat pada sistem pengaduan seperti mengganti nama, email, dan foto *profile* |

Definisi Use Case pada Use Case Diagram Admin Sistem Pengaduan ditujukan pada Tabel x.x.

**Tabel x. x Definisi *Use Case* pada *Use Case* Diagram Admin**

|  |  |  |
| --- | --- | --- |
| **No** | ***Use Case*** | **Deskripsi** |
| 1 | Autentikasi | Proses pengecekan dan validasi oleh sistem kepada pengguna sistem |
| 2 | Mengelola daftar pengaduan | Proses untuk mengelola daftar pengaduan yang terdapat pada sistem pengaduan seperti tambah, edit, dan hapus data |
| 3 | Mengelola daftar user | Proses untuk mengelola daftar user yang terdapat pada sistem pengaduan seperti tambah, edit, dan hapus data |
| 4 | Mengelola daftar instansi | Proses untuk mengelola dafar instansi yang terdapat pada sistem pengaduan seperti tambah, edit, dan hapus data |
| 5 | Mengelola profile admin | Proses untuk mengelola profile admin yang terdapat pada sistem pengaduan seperti mengganti nama, email, dan foto *profile* |
| 6 | Search pengaduan | Proses untuk mencari pengaduan berdasarkan nomor surat atau instansi yang ada pada sistem |

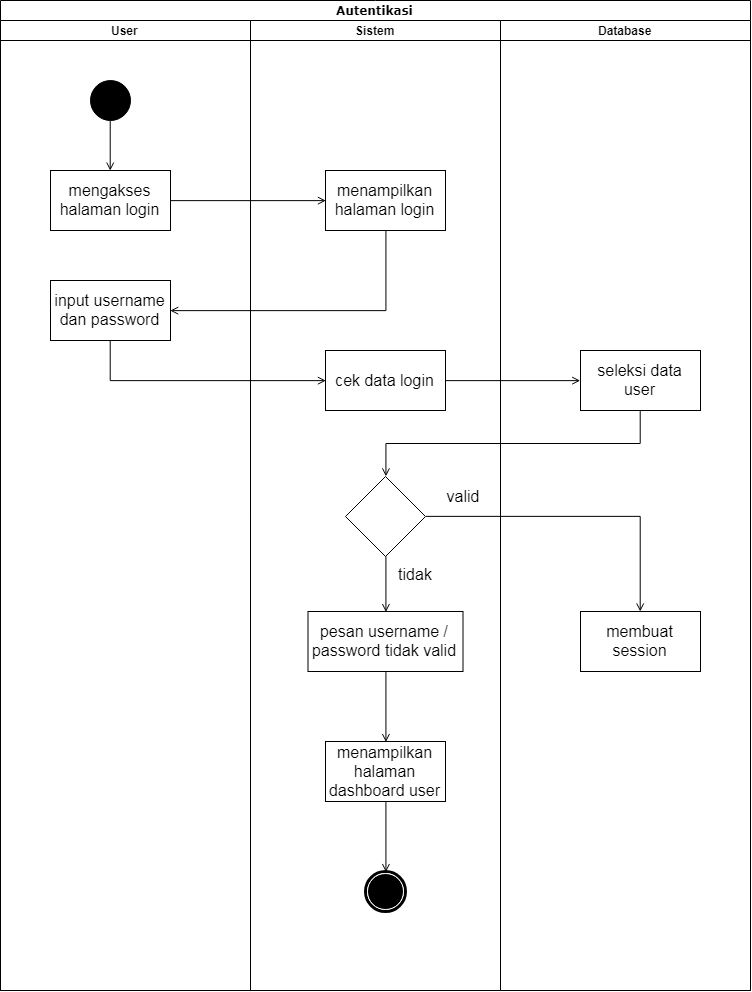
Definisi Use Case pada Use Case Diagram Kepala Dinas Sistem Pengaduan ditujukan pada Tabel x.x.

**Tabel x. x Definisi *Use Case* pada *Use Case* Diagram Kepala Dinas**

|  |  |  |
| --- | --- | --- |
| **No** | ***Use Case*** | **Deskripsi** |
| 1 | Autentikasi | Proses pengecekan dan validasi oleh sistem kepada pengguna sistem |
| 2 | Mendapatkan laporan rekap pengaduan | Proses untuk mendapatkan laporan rekap pengaduan yang ada pada sistem |
| 3 | Mendapatkan cetak pengaduan | Proses untuk mendapatkan cetak pengaduan yang ada pada sistem |
| 4 | Melihat grafik pengaduan | Proses untuk melihat grafik pengaduan yang ada pada sistem |
| 5 | Mengelola profile kepala dinas | Proses untuk mengelola profile kepala dinas yang terdapat pada sistem pengaduan seperti mengganti nama, email, dan foto *profile* |

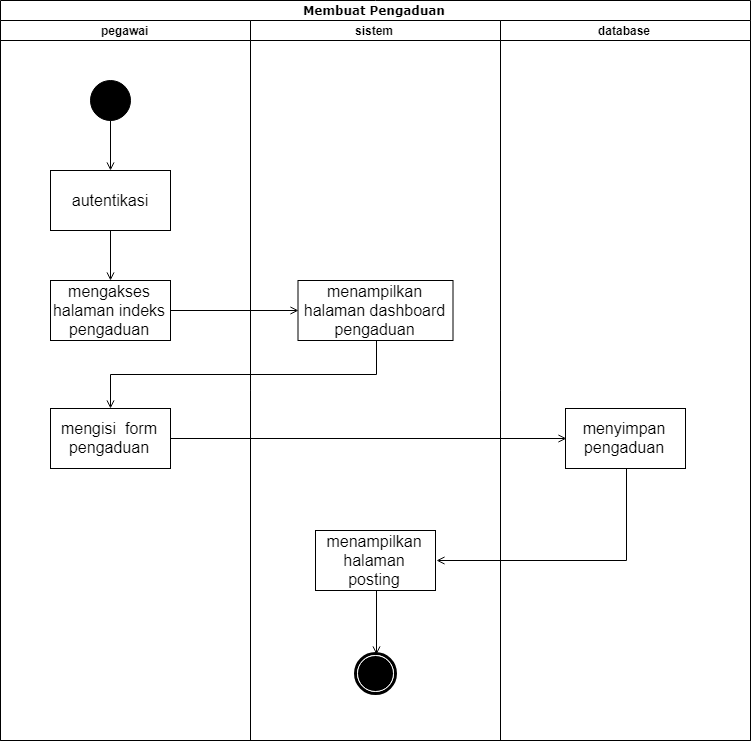
### *Activity* *Diagram*

*Activity diagram* akan menjabarkan aktifitas yang terdapat pada *use case* pada setiap *use case diagram*



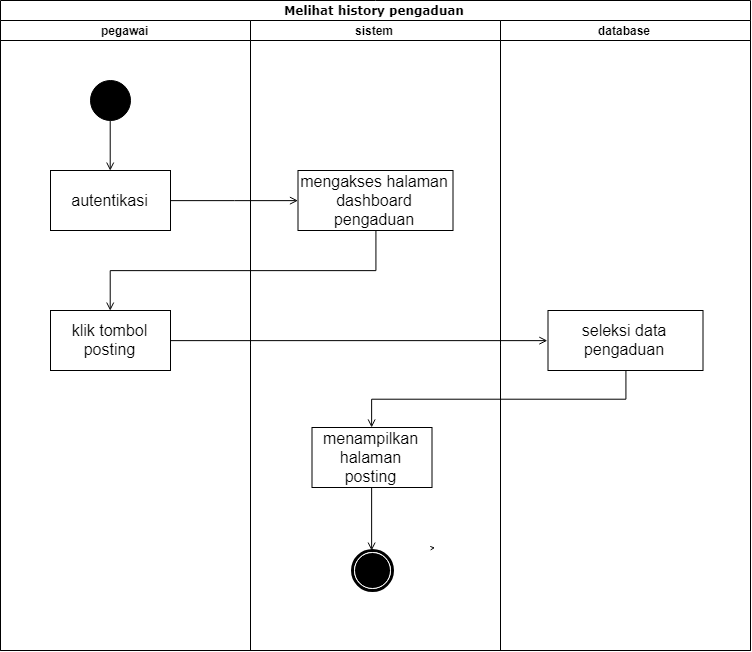
**Gambar x.x *Activity Diagram* Autentikasi**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses Auntentikasi. Pertama pengguna sistem memasukkan *username* dan *password*. Jika data login yang terdapat pada *database* dan *username* dengan *password* sesuai, maka sistem akan membuat *session* yang dimiliki oleh pengguna yaitu sebagai pegawai, *technical support*, admin, atau kepala dinas. Setelah data tersebut valid, maka pengguna berhak untuk mengakses halaman *dashboard*. Autentifikasi pada setiap proses dijelaskan pada diagram *activity* yang sama.



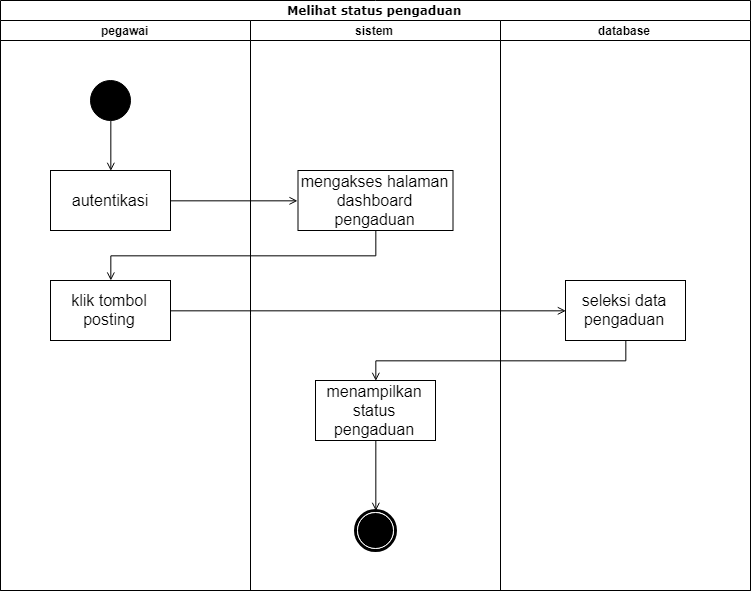
**Gambar x.x *Activity Diagram* Melakukan Pengaduan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses melakukanpengaduan. Hal pertama yang dilakukan adalah proses autentikasi, kemudian user mengakses halaman indeks pengaduan. Sistem menampilkan halaman dashboard. Setelah itu user mengisi form pengaduan kemudian disimpan ke database. Kemudian sistem menampilkan halaman posting.



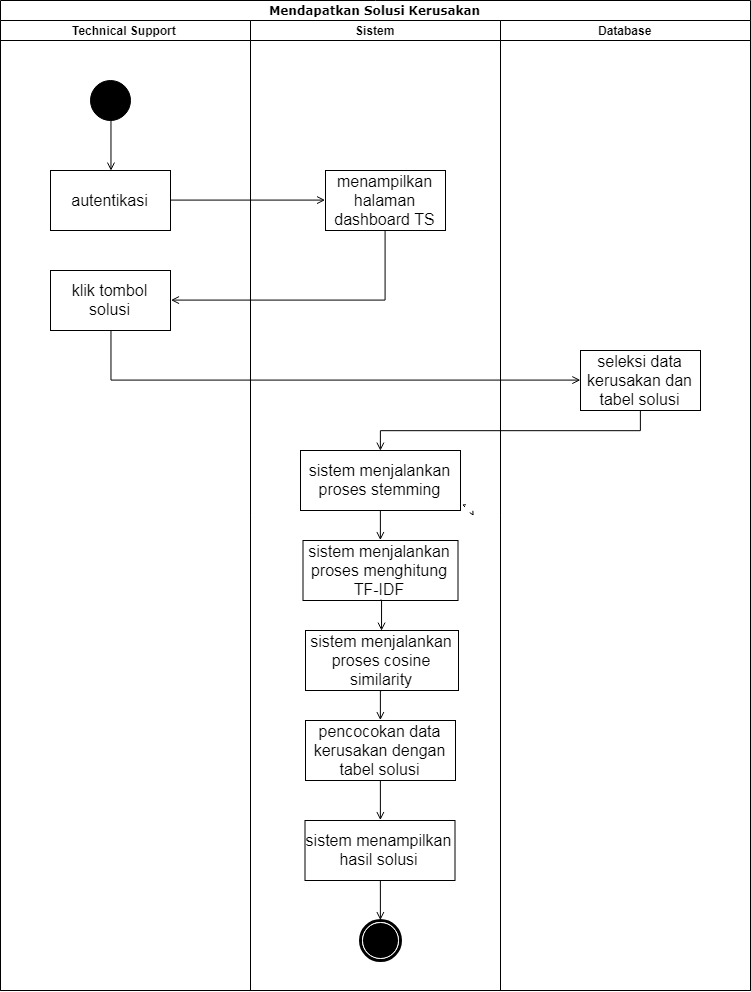
**Gambar x.x *Activity Diagram* Melihat *History* Pengaduan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses melihat *history* pengaduan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian mengakses halaman dashboard pengaduan. Klik tombol posting yang terdapat pada bagian navigasi. Database menyeleksi data pengaduan, kemudian sistem menampilkan halaman posting.



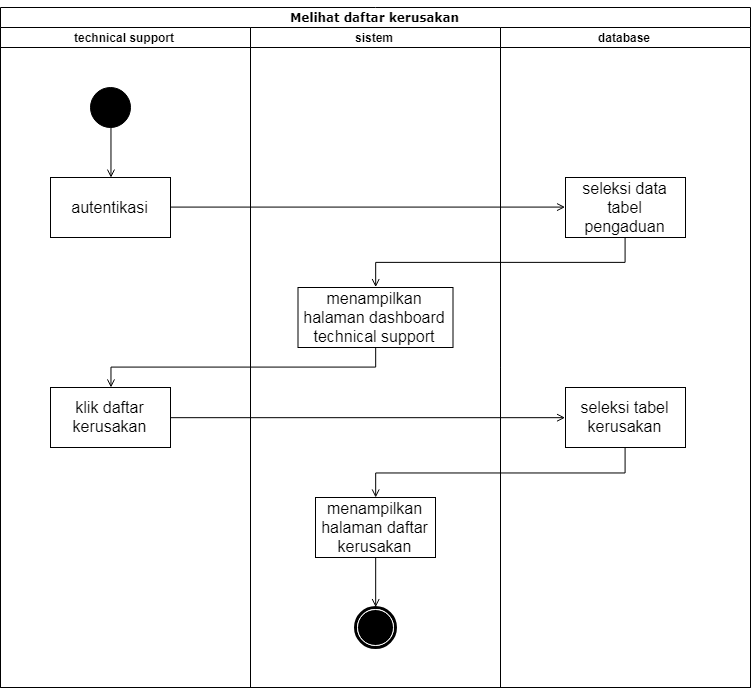
**Gambar x.x *Activity Diagram* Melihat Status Pengaduan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses melihat statuspengaduan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian mengakses halaman dashboard pengaduan. Klik tombol posting yang terdapat pada bagian navigasi. Database menyeleksi data pengaduan, kemudian sistem menampilkan status pengaduan yang terdapat pada halaman posting.



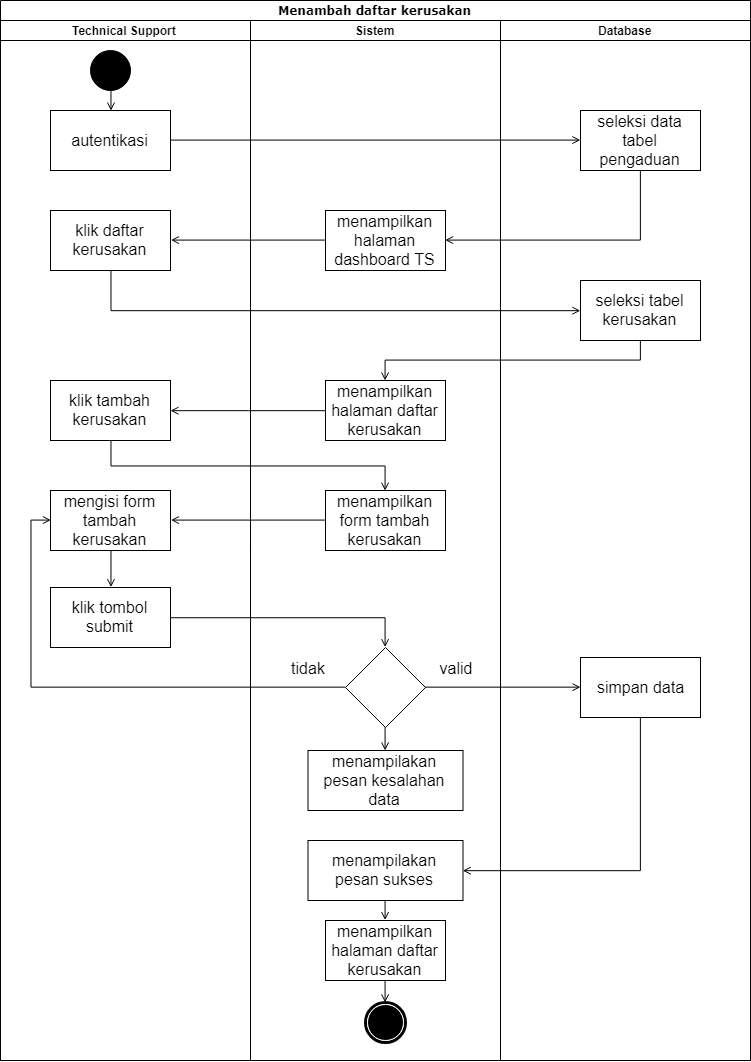
**Gambar x.x *Activity Diagram* Mendapatkan Solusi Kerusakan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses mendapatkan solusi kerusakan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian mengakses halaman dashboard *technical support*. Klik tombol solusi yang terdapat pada tabel pengaduan. Database menyeleksi data kerusakan dan tabel solusi, kemudian sistem menjalankan proses *stemming,* proses menghitung TF-IDF, proses *cosine similarity,* dan pencocokan data kerusakan dengan tabel solusi. Setelah semua proses dijalankan kemudian sistem menampilkan hasil solusi kerusakan.



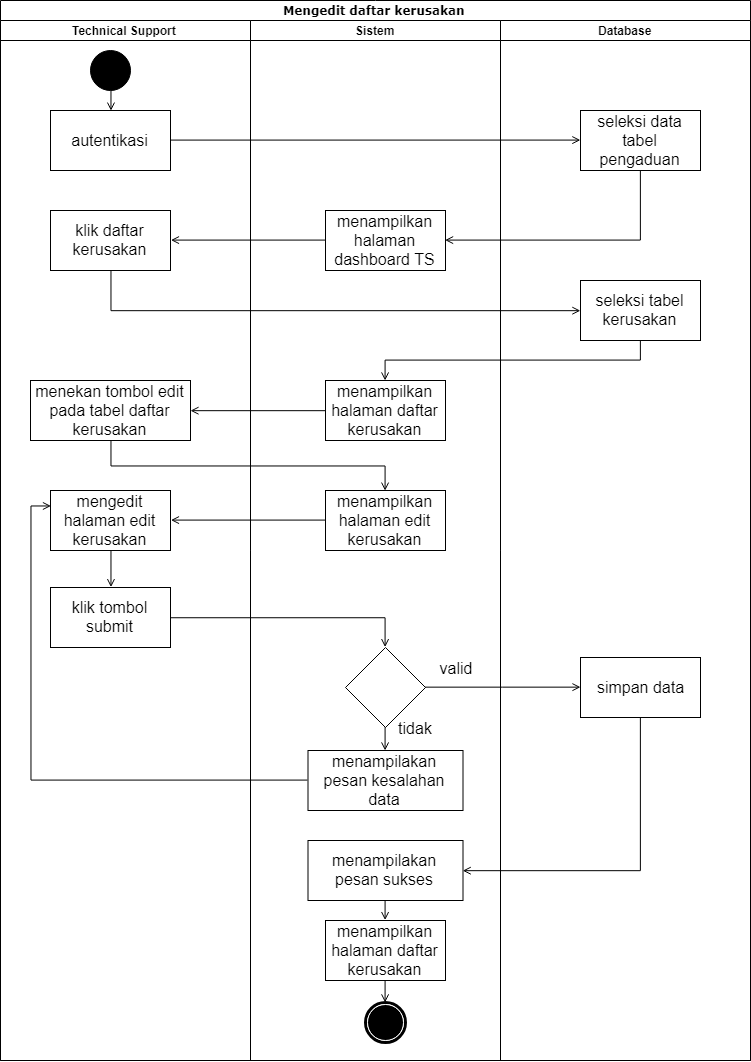
**Gambar x.x *Activity Diagram* Melihat Daftar Kerusakan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses melihat daftar kerusakan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan halaman dashboard *technical support*. Klik daftar kerusakan yang terdapat pada bagian navigasi. Database menyeleksi tabel kerusakan, kemudian sistem menampilkan halaman daftar kerusakan. *Activity diagram* pada *use case* lihat list solusi, *use case* lihat data pengaduan, *use case* lihat daftar user, dan *use case* lihat daftar instansi memiliki karakteristik yang sama dengan *activity diagram* melihat daftar kerusakan. Sehingga hanya dibuat satu buah *activity diagram* melihat daftar kerusakan saja.



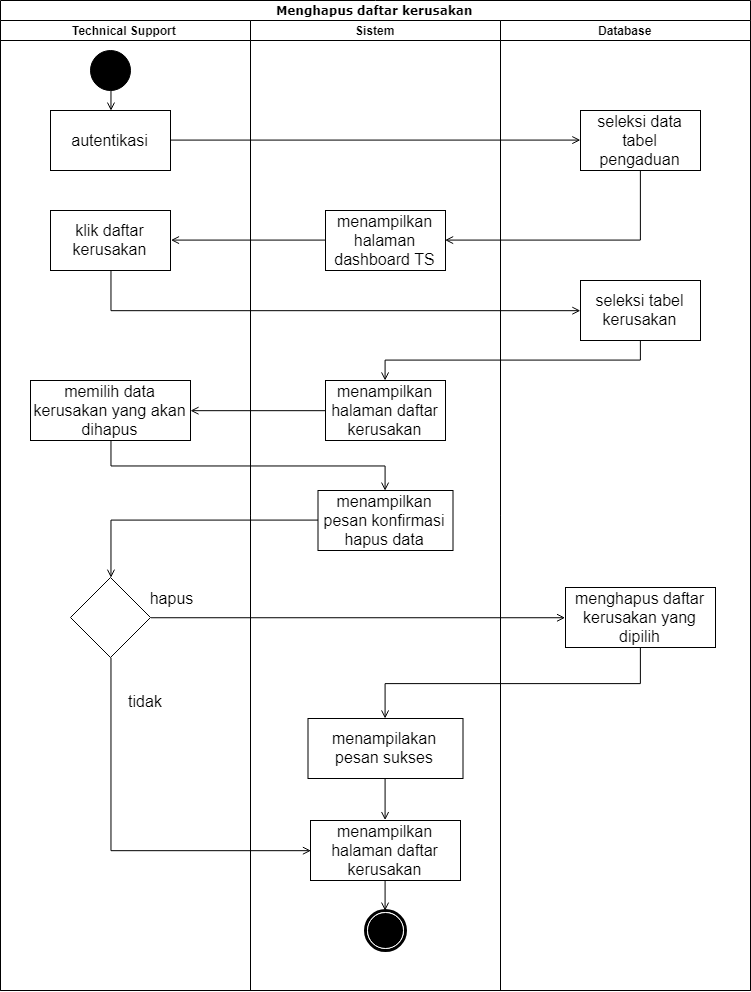
**Gambar x.x *Activity Diagram* Menambah Daftar Kerusakan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses menambah daftar kerusakan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan halaman dashboard *technical support*. Klik daftar kerusakan yang terdapat pada bagian navigasi. Database menyeleksi tabel kerusakan, kemudian sistem menampilkan halaman daftar kerusakan. Klik tombol tambah kerusakan, kemudian sistem menampilkan form tambah kerusakan. *Technical Support* mengisi form tambah kerusakan, lalu klik tombol submit. Apabila data tidak valid maka sistem akan menampilkan pesan kesalahan data dan apabila data valid maka data akan disimpan. *Activity diagram* pada *use case* tambah list solusi, *use case* tambah data pengaduan, *use case* tambah daftar user, dan *use case* tambah daftar instansi memiliki karakteristik yang sama dengan *activity diagram* tambah daftar kerusakan. Sehingga hanya dibuat satu buah *activity diagram* tambah daftar kerusakan saja.



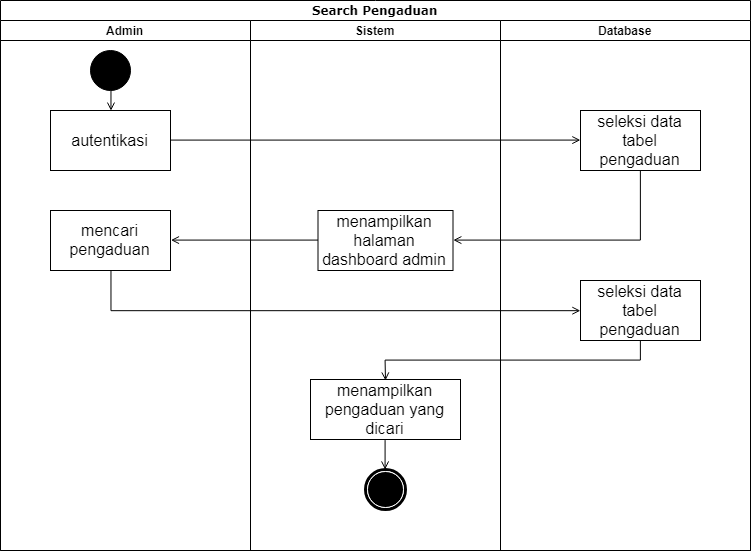
**Gambar x.x *Activity Diagram* Edit Daftar Kerusakan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses mengedit daftar kerusakan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan halaman dashboard *technical support*. Klik daftar kerusakan yang terdapat pada bagian navigasi. Database menyeleksi tabel kerusakan, kemudian sistem menampilkan halaman daftar kerusakan. Klik tombol edit kerusakan, kemudian sistem menampilkan form edit kerusakan. *Technical Support* mengedit form kerusakan, lalu klik tombol submit. Apabila data tidak valid maka sistem akan menampilkan pesan kesalahan data dan apabila data valid maka data akan disimpan. *Activity diagram* pada *use case* edit list solusi, *use case* edit data pengaduan, *use case* edit daftar user, dan *use case* edit daftar instansi memiliki karakteristik yang sama dengan *activity diagram* edit daftar kerusakan. Sehingga hanya dibuat satu buah *activity diagram* edit daftar kerusakan saja.



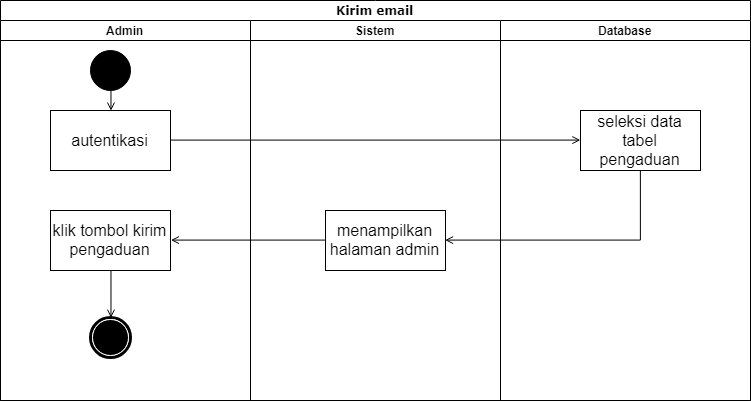
**Gambar x.x *Activity Diagram* Hapus Daftar Kerusakan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses menghapus daftar kerusakan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan halaman dashboard *technical support*. Klik daftar kerusakan yang terdapat pada bagian navigasi. Database menyeleksi tabel kerusakan, kemudian sistem menampilkan halaman daftar kerusakan. *Technical Support* memilih data kerusakan yang akan dihapus. Kemudian sistem menampilkan pesan konfirmasi hapus data. *Activity diagram* pada *use case* hapus list solusi, *use case* hapus data pengaduan, *use case* hapus daftar user, dan *use case* hapus daftar instansi memiliki karakteristik yang sama dengan *activity diagram* hapus daftar kerusakan. Sehingga hanya dibuat satu buah *activity diagram* hapus daftar kerusakan saja.



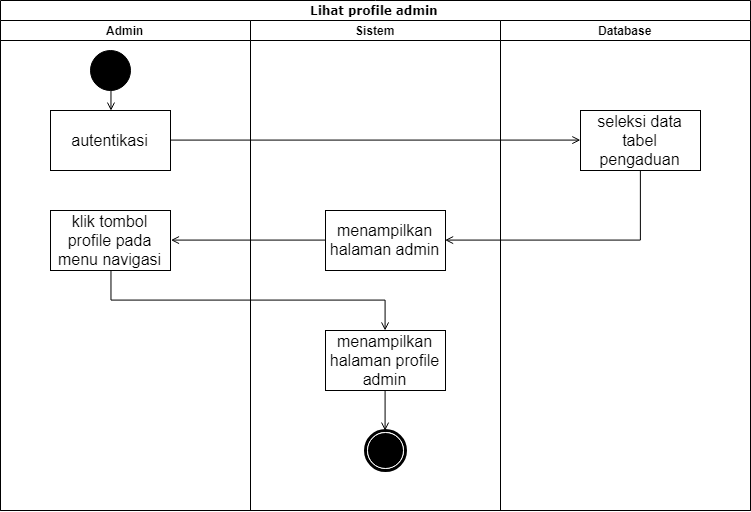
**Gambar x.x *Activity Diagram* *Search* Data Pengaduan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses *search* data pengaduan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan halaman admin. Admin mencari pengaduan dengan mengetikan nama instansi atau nomor surat. Database menyeleksi data pada tabel pengaduan, setelah itu sistem menampilkan pengaduan yang dicari.



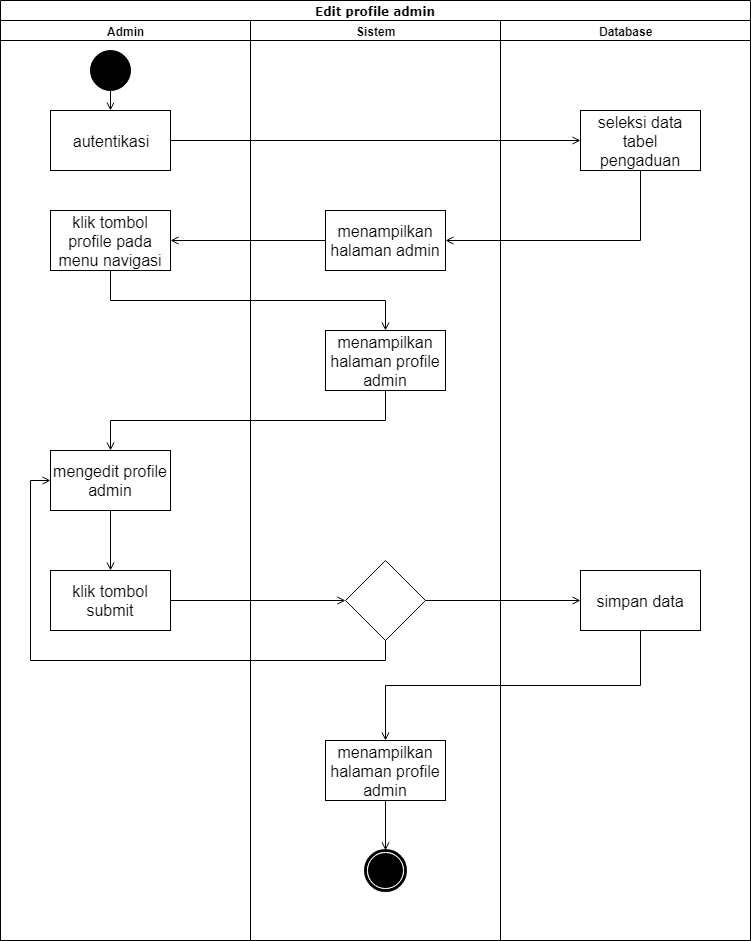
**Gambar x.x *Activity Diagram* Kirim Email Data Pengaduan**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses mengirim email data pengaduan. Hal pertama yang dilakukan adalah proses autentikasi. Kemudian database menyeleksi data pada tabel pengaduan. Sistem menampilkan halaman admin, lalu admin klik tombol kirim pengaduan.



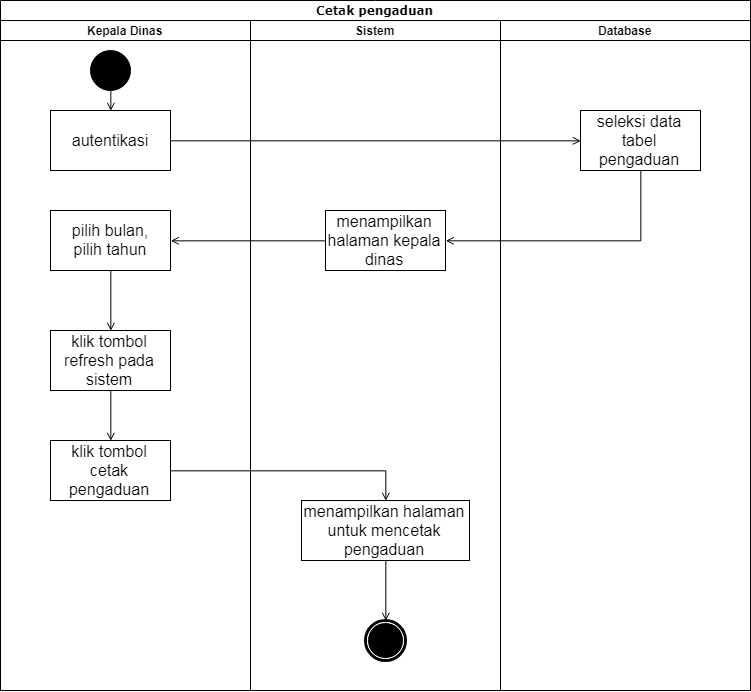
**Gambar x.x *Activity Diagram* Melihat Profile Admin**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses melihat profile admin. Hal pertama yang dilakukan adalah proses autentikasi. Kemudian database menyeleksi data pada tabel pengaduan. Sistem menampilkan halaman admin. Setelah itu admin klik tombol profile pada menu navigasi. Sistem menampilkan halaman profile admin.



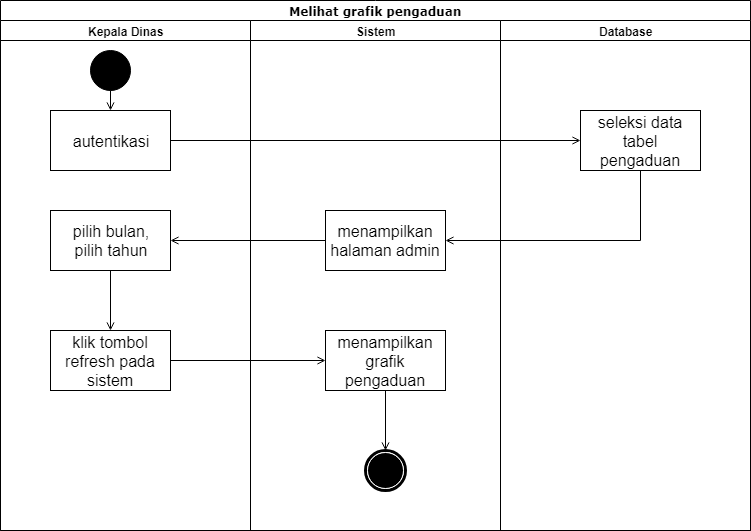
**Gambar x.x *Activity Diagram* Mengedit Profile Admin**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses mengedit profile admin. Hal pertama yang dilakukan adalah proses autentikasi. Kemudian database menyeleksi data pada tabel pengaduan. Sistem menampilkan halaman admin. Setelah itu admin klik tombol profile pada menu navigasi. Sistem menampilkan halaman profile admin. Admin mengedit profile admin, lalu klik tombol submit. Apabila data tidak sesuai maka data tidak disimpan, dan bila data sesuai maka data disimpan. Kemudian sistem menampilkan halaman profile admin.



**Gambar x.x *Activity Diagram* Mencetak Data Pengaduan**

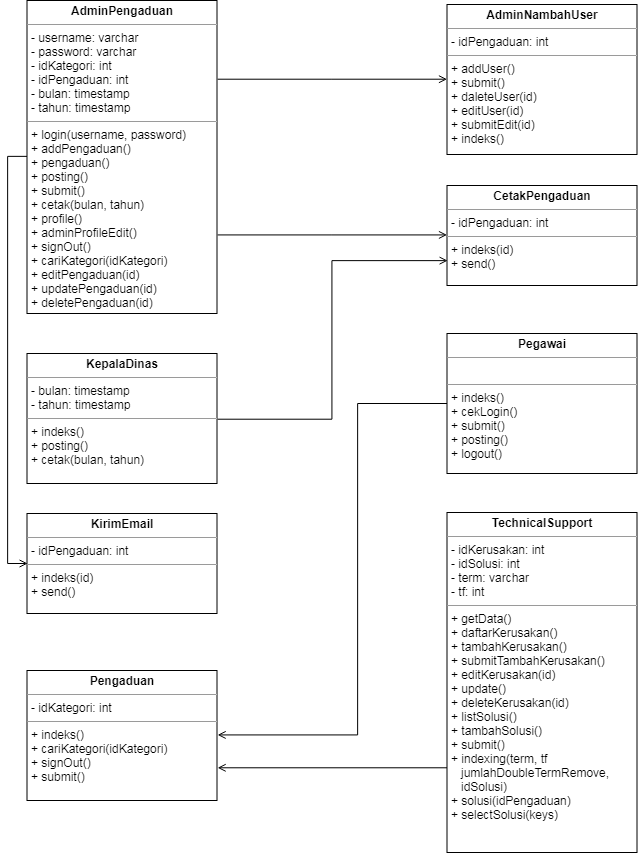
Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses mencetak data pengaduan. Hal pertama yang dilakukan adalah proses autentikasi, kemudian database menyeleksi data pada tabel pengaduan. Sistem menampilkan halaman kepala dinas. Kepala dinas memilih bulan dan tahun pada menu yang disediakan, lalu klik tombol refresh pada sistem. Setelah itu kepala dinas klik tombol cetak pengaduan. Kemudian sistem menampilkan halaman untuk mencetak pengaduan.



**Gambar x.x *Activity Diagram* Melihat Grafik**

Gambar x.x menjelaskan alur aktivitas yang terdapat pada proses melihat grafik data pengaduan. Hal pertama yang dilakukan adalah proses autentikasi, kemudian database meyeleksi data pada tabel pengaduan. Sistem menampilkan halaman admin. Kepala dinas memilih bulan dan tahun pada menu yang disediakan. Kemudian kepala dinas klik tombol refresh pada sisem. Sistem menampilkan grafik pengaduan.

### Class Diagram



**Gambar x.x *Class Diagram***

Gambarx.xMerupakan rancangan dari sistem pengaduan dalam bentuk *class diagram.*  Diagram ini dibangun berdasarkan definisi dari *use case diagram,* dimana masing-masing fungsi pada tiap kelas dibuat untuk memenuhi kebutuhan proses yang terdapat pada *Use Case.* Penjelasan class diagram dijelaskan pada Tabel x.x.

|  |  |  |
| --- | --- | --- |
| **Kelas** | ***Use Case*** | **Fungsi** |
| AdminPengaduan | Autentikasi | Menangani operasi pengelolaan dan akses data admin pada sistem |
| Mengelola data pengaduan |
| Mengelola daftar user |
| Mengelola daftar instansi |
| Mengelola profile admin |
| *Search* pengaduan |
| KepalaDinas | Autentikasi | Menangani proses rekap pengaduan, cetak pengaduan, grafik pengaduan, mengelola profile kepala dinas, dan akses data kepala dinas pada sistem |
| Mendapatkan laporan rekap pengaduan |
| Mnedapatkan cetak pengaduan |
| Melihat grafik pengaduan |
| Mengelola profile kepala dinas |
| KirimEmail | Kirim pengaduan | Menangani proses kirim pengaduan melalui email |
| Pengaduan | Membuat pengaduan | Menangani proses membuat pengaduan |
| AdminNambahUser | Mengelola daftar user | Menangani proses admin untuk menambah user pada sistem pengaduan |
| CetakPengaduan | Mendapatkan cetak pengaduan | Menangani proses cetak pengaduan yang dilakukan oleh kepala dinas |
| Pegawai | Autentikasi | Menangani proses lihat *history* pengaduan, lihat status pengaduan, dan akses data pegawai pada sistem |
| Melihat *history* pengaduan |
| Melihat status pengaduan |
| TechnicalSupport | Autentikasi | Menangani proses mendapatkan solusi kerusakan, menangani proses operasi pengelolaan, dan akses data pegawai pada sistem |
| Mendapatkan solusi kerusakan |
| Mengelola daftar kerusakan |
| Mengelola list solusi |
| Mengelola profile *technical support* |

### Sequence Diagram

…………………………………………………………………………………………………………

### Entity Relational Diagram

…………………………………………………………………………………………………………

### Preprocessing, TF-IDF, Cosine Similarity

Dalam membangun sistem terdapat tahapan penting yang lakukan pada penelitian ini untuk mendapatkan output berupa solusi kerusakan komputer yaitu tahap preprocessing data, TF-IDF, dan cosine similarity untuk mendeteksi kemiripan teks. Adapun flowchart preprocessing, TF-IDF, dan cosine similarity yang digunakan pada penelitian ini adalah :



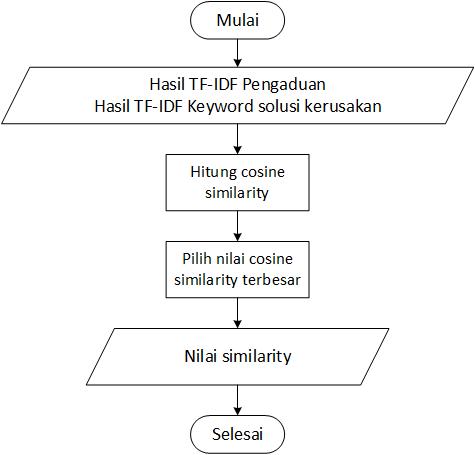
Gambar x.x *Flowchart* *Preprocessing*

*Flowchart preprocessing* pada Gambar 7.5 adalah suatu proses yang bertujuan mengolah data mentah menjadi data yang berkualitas. Pada *flowchart preprocessing* inputan berupa teks pengaduan dan teks *keyword* solusi kerusakan. Dokumen-dokumen diproses dengan memecah menjadi kata-kata *(tokenization)*, kemudian menghilangkan tanda baca, spasi, dan spesial karakter. Kata-kata yang tidak memiliki deskriptif dihilangkan melalui pengecekan kata yang terdapat dalam daftar kata yang tidak penting *(stopwords)*. Proses selanjutnya adalah menghilangkan imbuhan, awalan, dan akhiran menjadi kata dasar *(stemming)*.

|  |  |
| --- | --- |
| *Flowchart* TF-IDF Pengaduan  H:\..KULI AH\PROPOSAL ASTUNGKARA GASS\dfd\flowchart perhitungan TF-IDF.jpg  (a) | *Flowchart* TF-IDF *Keyword Solusi*  H:\..KULI AH\PROPOSAL ASTUNGKARA GASS\dfd\flowchart perhitungan TF-IDF keyword.jpg  (b) |

Gambar 7. 6 *Flowchart* TF-IDF

Gambar 7.6 (a) merupakan proses menghitung bobot TF-IDF untuk teks pengaduan dan Gambar 7.6 (b) merupakan proses menghitung bobot TF-IDF untuk teks *keyword* solusi kerusakan. Inputan pada *flowchart* ini adalah teks pengaduan dan teks *keyword* solusi kerusakan.Proses menghitung bobot TF-IDF dilakukan setelah melalui tahapan *preprocessing.* Cara menghitung TF-IDF adalah dengan mengalikan *term frequency* (TF) dengan *invers document frequency* (IDF)*.* Rumus dari *Term frequency* adalah . Dimana *tf* adalah banyaknya *term* i pada sebuah dokumen dan *max (tf)* adalah jumlah semua *term* i pada seluruh dokumen. Rumus dari *invers document frequency* adalah . Dimana *D* adalah dokumen ke-d dan *dft* adalahbanyak dokumen yang mengandung *term* i.



**Gambar 7. 7 *Flowchart* Perhitungan Kemiripan Teks**

*Flowchart* Perhitungan Kemiripan Teks pada Gambar 7.7 merupakan proses menghitung kemiripan antara teks pengaduan dan teks *keyword* solusi kerusakan komputermenggunakan *cosine similarity.* Inputan pada *flowchart* ini adalah hasil TF-TDF Pengaduan dan *keyword* solusi kerusakan. Setelah itu dilakukan proses menghitung *cosine similarity.* Cara menghitung dengan *cosine similarity* adalah dengan rumus . Dimana adalah bobot istilah *j* pada dokumen dan adalah bobot istilah *j* pada dokumen . Setelah mendapatkan hasil *cosine similarity,* maka proses selanjutnya adalah memilih nilai *cosine similarity* terbesar. Dimana nilai *cosine similarity* terbesar adalah nilai terbesar dari kecocokan antara teks pengaduan dengan seluruh teks *keyword* solusi kerusakan.

### Perancangan Antar Muka

Perancangan antar muka berguna untuk merancang desain awal sistem yang akan dibangun. Pada perancangan antar muka berisi konten serta menu-menu yang digunakan pada sistem. Adapun tampilan yang akan dirancang yaitu halaman pegawai, halaman *technical support,* halaman admin, dan halaman kepala dinas.

1. Tampilan Login

Rancangan tampilan login merupakan tampilan yang digunakan user untuk proses autentikasi. Tampilan ini diakses sebelum user ingin menuju halaman dashboard, dengan menginputkan *username* dan *password.* Desain tampilan login dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Login

1. Tampilan Halaman Awal Pegawai

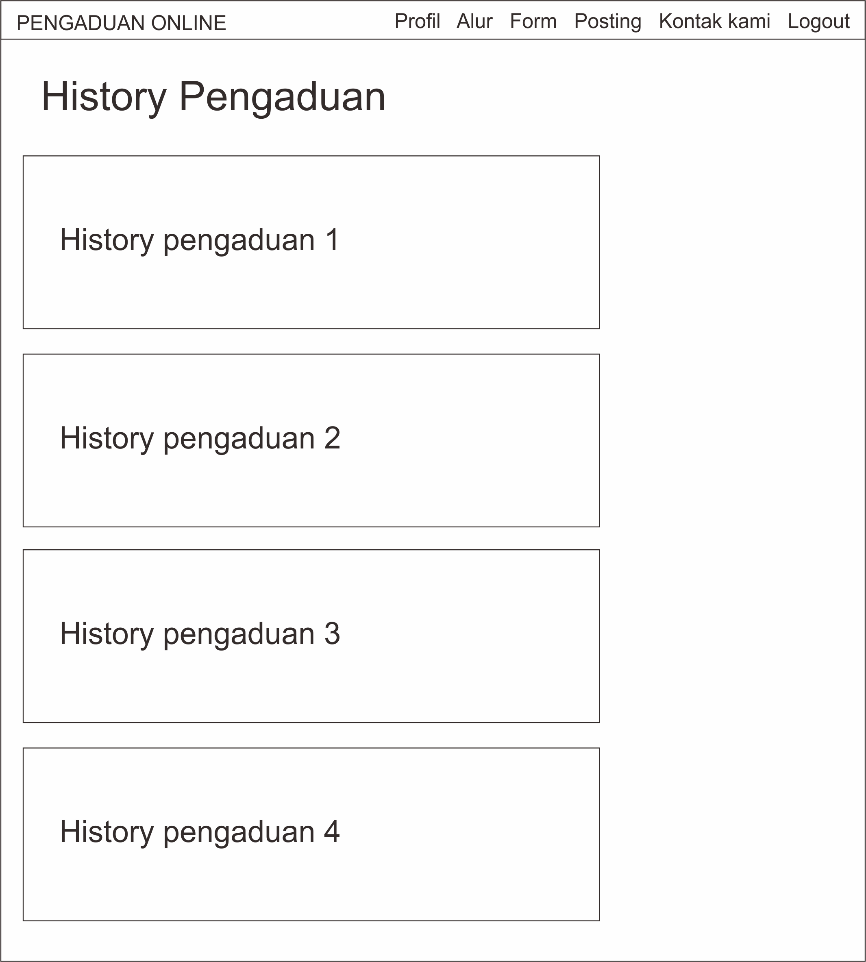
Rancangan tampilan halaman awal pegawai merupakan tampilan awal yang muncul setelah pegawai melalui proses login. Tampilan halaman awal pegawai terdapat menu navigasi (profil, alur, form, posting, kontak kami, dan logout), header, profil, alur, form, kontak kami. Desain tampilan halaman awal pegawai dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Awal Pegawai

1. Tampilan Halaman Posting

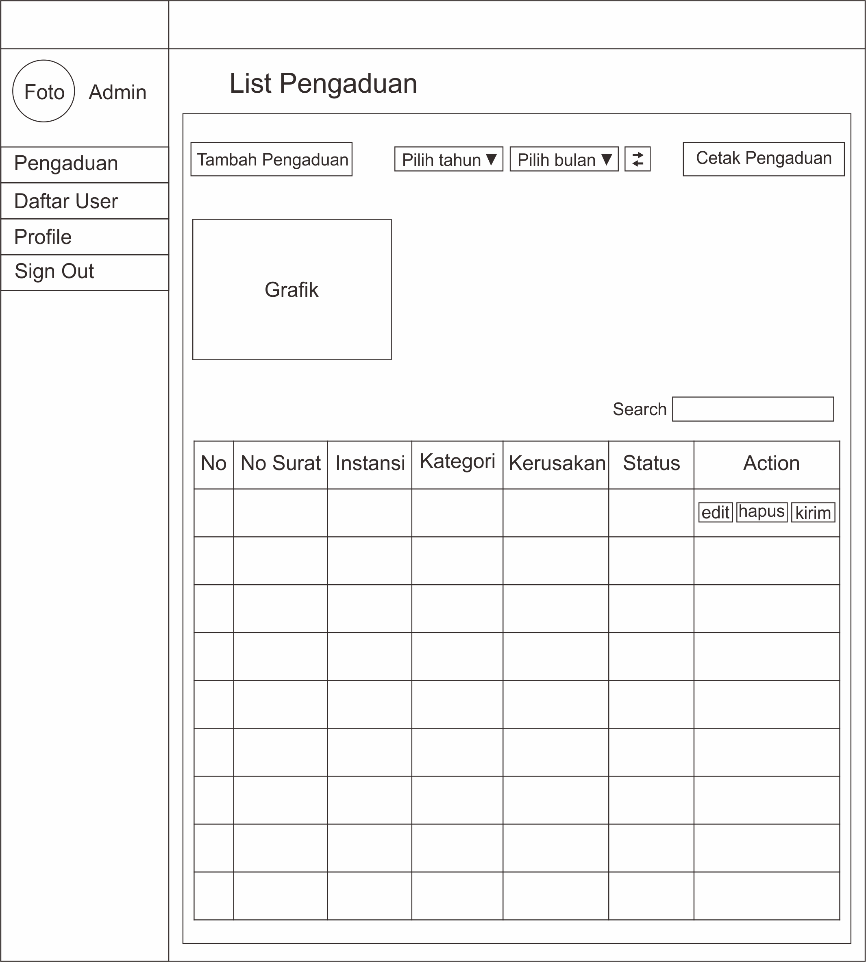
Rancangan tampilan halaman posting merupakan tampilan yang berisi history pengaduan yang dilakukan oleh pegawai. Pada halaman ini pegawai dapat melihat pengaduan yang pernah dilakukan sebelumnya dan pegawai dapat melihat status pengaduan. Status pengaduan ada dua jenis yaitu belum diproses dan sudah diproses. Desain tampilan halaman posting dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Posting

1. Halaman Awal Admin

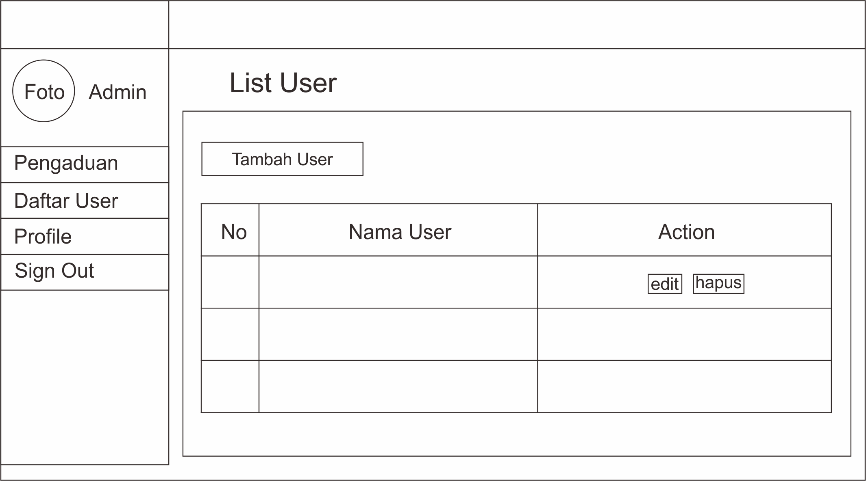
Rancangan tampilan halaman awal admin merupakan tampilan awal yang muncul setelah admin melalui proses login. Tampilan halaman awal admin berisi menu navigasi (pengaduan, daftar user, profile, dan sign out), tambah pengaduan, cetak pengaduan, pilih tahun dan bulan pengaduan yang berguna memunculkan grafik, cari pengaduan, dan tabel pengaduan. Desain tampilan halaman awal admin dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Awal Admin

1. Halaman Daftar User

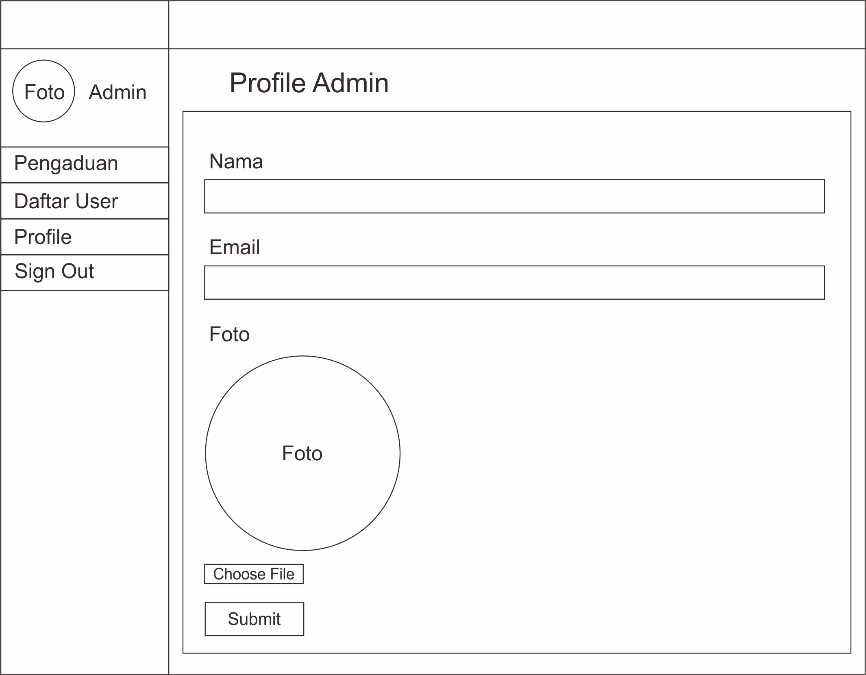
Halaman list user merupakan tampilan tempat mengelola daftar user untuk admin. Halaman ini berisi menu navigasi (pengaduan, daftar user, profile, dan sign out), tambah user, edit user, dan hapus user. Desain tampilan halaman daftar user dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Daftar User

1. Halaman Profile

Halaman profile merupakan halaman yang digunakan admin untuk update profile admin. Halaman ini berisi menu navigasi (pengaduan, daftar user, profile, dan sign out). Pada halaman ini admin dapat mengganti nama profile, email, dan foto profile admin. Desain tampilan halaman profile dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Profile

1. Halaman Awal *Technical Support*

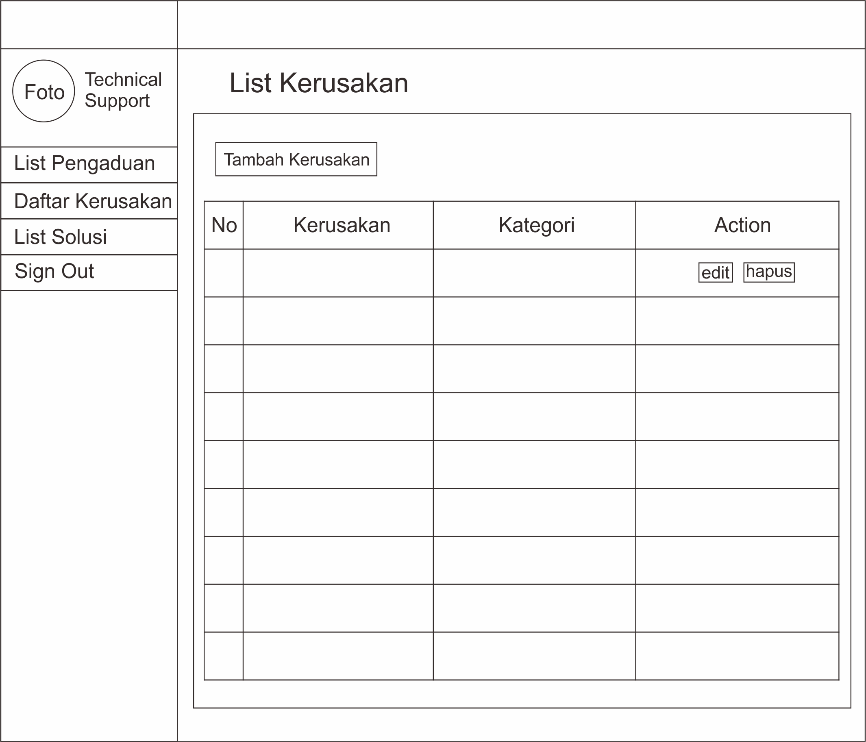
Rancangan tampilan halaman awal *technical support* merupakan tampilan awal yang muncul setelah *technical support* melalui proses login. Tampilan halaman awal *technical support* berisi menu navigasi (list pengaduan, daftar kerusakan, list solusi, dan sign out), tabel pengaduan, menu solusi. Desain tampilan halaman awal *technical support* dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Awal *Technical Support*

1. Halaman Daftar Kerusakan

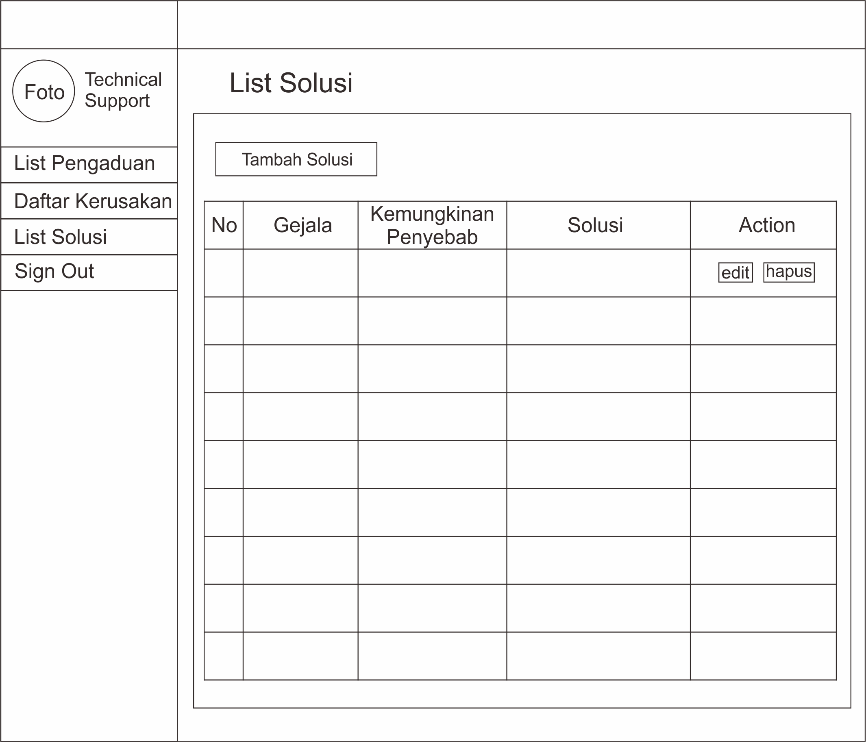
Halaman daftar kerusakan merupakan halaman yang digunakan *technical support* untuk melihat daftar kerusakan yang ada pada sistem. Halaman ini berisi menu navigasi (pengaduan, daftar user, profile, dan sign out). Pada halaman ini *technical support* dapat menambah, mengedit, dan menghapus kerusakan. Desain tampilan halaman daftar kerusakan dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Daftar Kerusakan

1. Halaman List Solusi

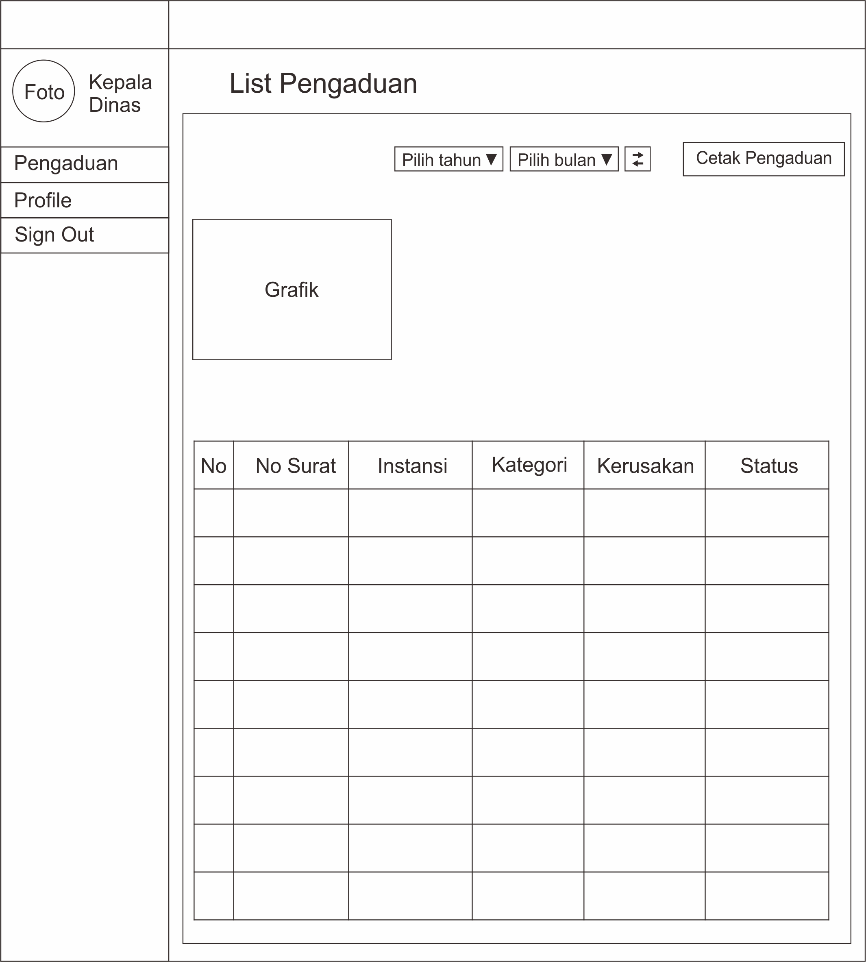
Halaman list solusi merupakan halaman yang digunakan *technical support* untuk melihat solusi yang ada pada sistem. Halaman ini berisi menu navigasi (pengaduan, daftar user, profile, dan sign out). Pada halaman ini *technical support* dapat menambah, mengedit, dan menghapus solusi. Desain tampilan halaman list solusi dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman List Solusi

1. Halaman Kepala Dinas

Halaman kepala dinas merupakan halaman yang digunakan kepala dinas untuk melihat rekap pegaduan yang ada pada sistem. Halaman ini berisi menu navigasi (pengaduan, profile, dan sign out). Pada halaman ini kepala dinas dapat melihat rekap pengaduan, mencari pengaduan, melihat grafik pengaduan, dan mencetak rekap pengaduan. Desain tampilan halaman kepala dinas dapat dilihat seperti pada Gambar x.x.



Gambar x.x Desain Tampilan Halaman Kepala Dinas

# BAB IV

**HASIL DAN PEMBAHASAN**

## Gambaran Umum Sistem

## Lingkungan Implementasi

## Implementasi Basis Data

## Implementasi Sistem

### Implementasi Fungsi-Fungsi Operasi Pada Sistem

### Implementasi Proses Stemming Pada Sistem

### Implementasi Proses Perhitungan TF-IDF Pada Sistem

### Implementasi Proses Algoritma Pada Sistem

## Antar Muka Sistem

## Pengujian Sistem

### Pengujian Black Box

### Stress Testing

### Pengujian Terhadap Respon Pengguna

# BAB V

**KESIMPULAN DAN SARAN**

## Kesimpulan

## Saran

# DAFTAR PUSTAKA