**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**

# LEMBAR PENGESAHAN TUGAS AKHIR

Judul : Sistem Pengaduan Kerusakan Komputer Dan

Rekomendasi Solusi Menggunakan Algoritma *Cosine*

*Similarity*

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 : Sistem Pengaduan Kerusakan Komputer Dan

Rekomendasi Solusi Menggunakan Algoritma *Cosine*

*Similarity*

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.

Berdasarkan wawancara yang telah dilakukan, selama ini para pegawai melaporkan perangkat komputer yang mengalami masalah melalui sebuah mekanisme pengaduan kerusakan secara *offline*. Proses pengaduan biasanya dilakukan dengan membuat surat pengaduan yang diproses melalui tahapan yang sesuai dengan SOP (Standar Operasional Prosedur) sehingga memakan waktu yang lama. Salah satu faktor penyebab lamanya proses tersebut adalah karena keterbatasan pengetahuan 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 *history* 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. Mengetahui waktu yang dibutuhkan oleh sistem dalam memproses *query* hingga merekomendasikan solusi yang didapatkan oleh sistem
3. Mengetahui respon pengguna terhadap sistem pengaduan kerusakan komputer

## 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.

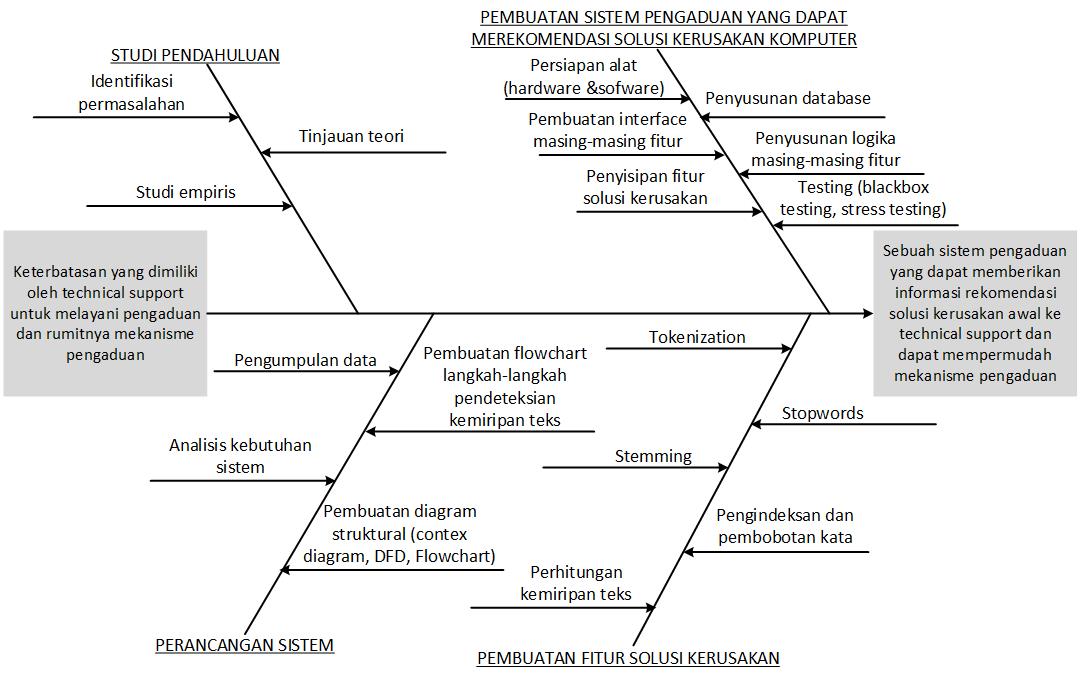
## Manfaat Penelitian

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 1.1



Gambar 1. 1 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 1.2 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 1. 2 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. Hasil *output* yang muncul pada sistem berdasarkan hasil perhitungan dengan nilai 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 belum terstruktur menjadi data yang terstruktur sesuai kebutuhan, untuk proses t*ext mining* yang lebih lanjut (Fauzi, 2014). Tahap preprocessing terdiri dari *casefolding, tokenization, filtering,* dan *stemming*.

1. *Casefolding* adalah proses menghilangkan karakter-karakter seperti tanda baca dan mengubah semua huruf kapital menjadi huruf kecil *(lowercase)*.
2. *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.
3. *Filtering* adalah proses menghapus kata-kata tidak penting dari hasil *tokenization* dengan cara mencocokan kata tersebut dengan kata yang terdapat pada daftar kata tidak penting (*stopwords*).
4. *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

Contoh perhitungan dari *cosine similarity* adalah sebagai berukut :

Diketahui :

Query = (0,707 ; 0,070 ; 0 ; 0)

Dokumen 1 = (0,578 ; 0,578 ; 0,578 ; 0)

Dokumen 2 = (0 ; 0 ; 0 ; 0,548)

Perhitungan :

Cosim (Dokumen 1, Query)

Cosim (Dokumen 2, Query)

Hasil : mengembalikan dokumen 1

## 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 model *Waterfall*:

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 2. 1 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 3.1. Berdasarkan wawancara tersebut, maka didapatkan hasil untuk analisis kebutuhan sistem yang dibedakan menjadi kebutuhan fungsional dan kebutuhan non fungsional.

Tabel 3. 1 Kebutuhan Sistem Berdasarkan Analisa

|  |  |
| --- | --- |
| No | Kebutuhan |
| 1 | Sistem pengaduan berbasis web dapat melakukan pengaduan secara *onlilne* dengan fitur form pengaduan |
| 2 | Sistem dapat mengirimkan pengaduan berupa surat melaui *email* dengan fitur kirim pengaduan |
| 3 | *Technical support* mendapatkan solusi kerusakan komputer dengan fitur solusi kerusakan |
| 4 | Kepala dinas mendapatkan laporan rekap pengaduan dengan fitur cetak 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 A. Adapun rincian dari kebutuhan fungsional sistem yaitu :

Tabel 3. 2 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 pegawai 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 |

Tabel 3. 2 Kebutuhan Fungsional Sistem (lanjutan)

|  |  |  |
| --- | --- | --- |
| 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 *search* pengaduan 8. Admin dapat melihat grafik pengaduan |
| KF9 | Admin | Admin dapat mengelola daftar user dengan rincian :   1. Admin dapat menambah user 2. Admin dapat melihat daftar user 3. Admin dapat mengedit user 4. Admin dapat menghapus user |

Tabel 3. 2 Kebutuhan Fungsional Sistem (lanjutan)

|  |  |  |
| --- | --- | --- |
| KF10 | Admin | Admin dapat mengelola daftar instansi dengan rincian :   1. Admin dapat menambah instansi 2. Admin dapat melihat daftar instansi 3. Admin dapat mengedit instansi 4. Admin dapat menghapus instansi |
| KF11 | Kepala dinas | Kepala dinas mendapatkan laporan rekap pengaduan dengan rincian :   1. Kepala dinas mendapatkan informasi pengaduan berdasarkan tahun dan bulan yang dipilih dan kepala dinas dapat mencetak pengaduan berdasarkan tahun dan bulan yang dipilih 2. Kepala dinas dapat melihat grafik pengaduan berdasarkan instansi dinas, kategori kerusakan, dan status pengaduan |
| KF12 | *Technical support*, admin, kepala dinas | Sistem dapat mengelola profile dengan rincian:   * 1. Sistem dapat mengubah nama profile   2. Sistem dapat mengubah alamat email profile   3. Sistem dapat mengganti foto profile |

### Kebutuhan Non Fungsional

Kebutuhan non fungsional sistem terdiri dari segi *user friendly,* kecepatan dan keamanan yang dijelaskan pada Tabel 3.3.

Tabel 3. 3 Kebutuhan Non Fungsional Sistem

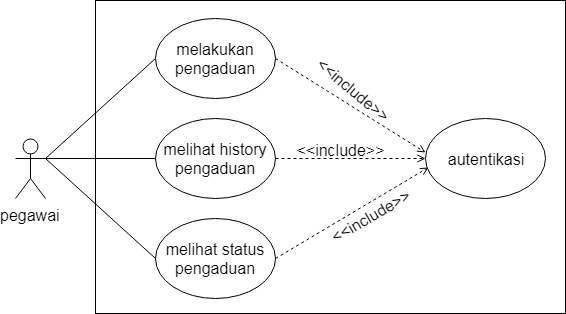
|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan Non Fungsional** | **Deskripsi** |
| 1 | *User Friendly* | Tampilan antar muka sistem harus mudah dimengerti oleh pegawai, *technical support*, admin, dan kepala dinas sebagai pengguna sistem. Penempatan fitur-fitur yang ada pada sistem diatur agar 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

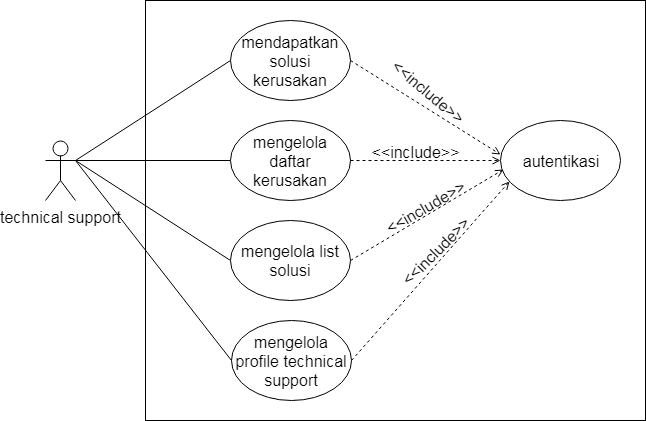
Sistem pengaduan dibangun dengan berbasis berorientasi objek sehingga sistem dirancang menggunakan diagram UML *(Unified Modeling Language)* antara lain *Use Case Diagram, Activity Diagram, Class Diagram,* dan *Sequence Diagram.*

### Use Case Diagram

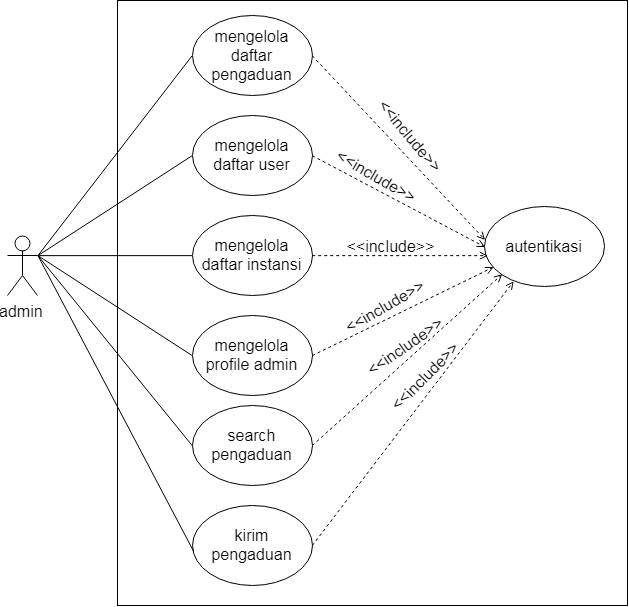
Usecase diagram menampilkan hubungan antara aktor (pelaku) dengan hal-hal yang terlibat pada sistem.



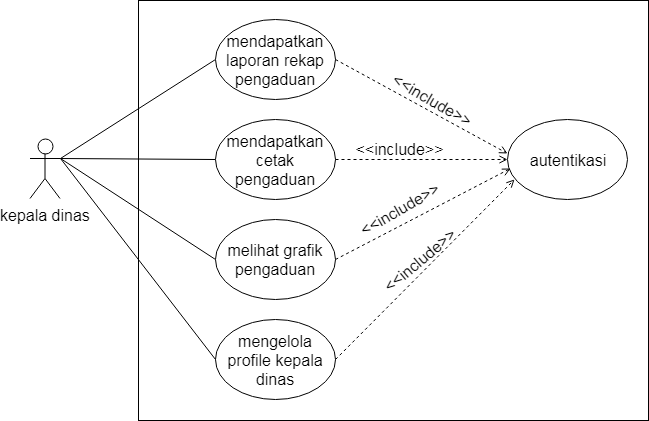
Gambar 3. 1 *Use Case Diagram* Pegawai dalam Sistem Pengaduan



Gambar 3. 2 *Use Case Diagram Technical Support* dalam Sistem Pengaduan

****

Gambar 3. 3 *Use Case Diagram* Admin dalam Sistem Pengaduan

****

Gambar 3. 4 *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 3.4.

Tabel 3. 4 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 3.5.

Tabel 3. 5 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 3.6.

Tabel 3. 6 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 3.7.

Tabel 3. 7 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 |

Tabel 3. 7 Definisi *Use Case* pada *Use Case* Diagram Admin (lanjutan)

|  |  |  |
| --- | --- | --- |
| 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 |

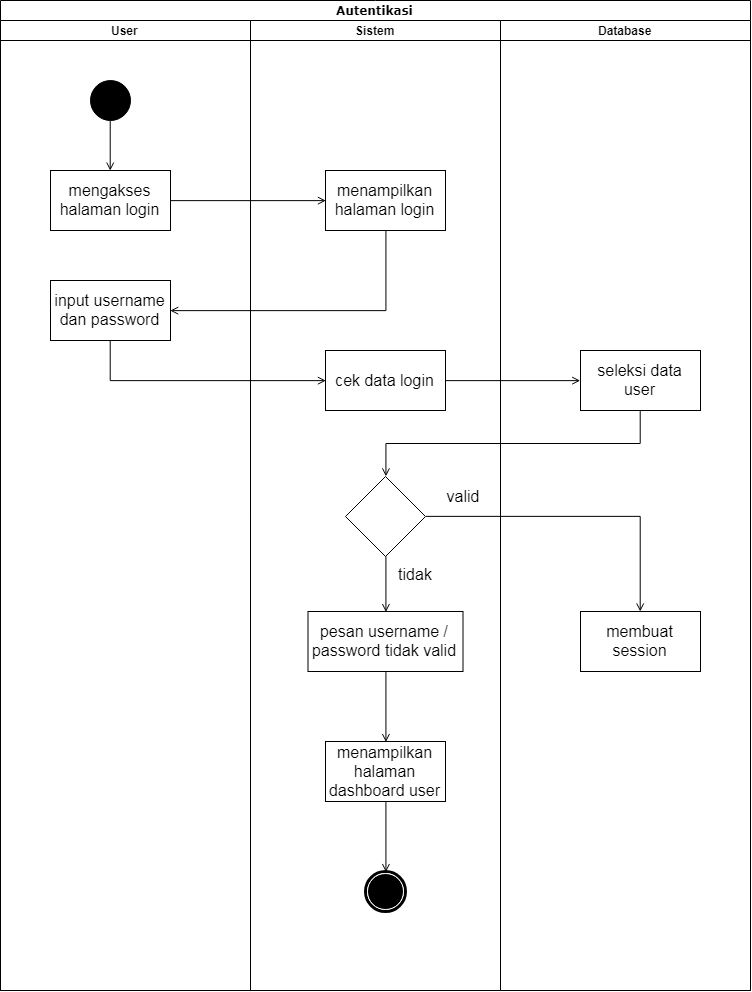
Definisi Use Case pada Use Case Diagram Kepala Dinas Sistem Pengaduan ditujukan pada Tabel 3.8.

Tabel 3. 8 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* |

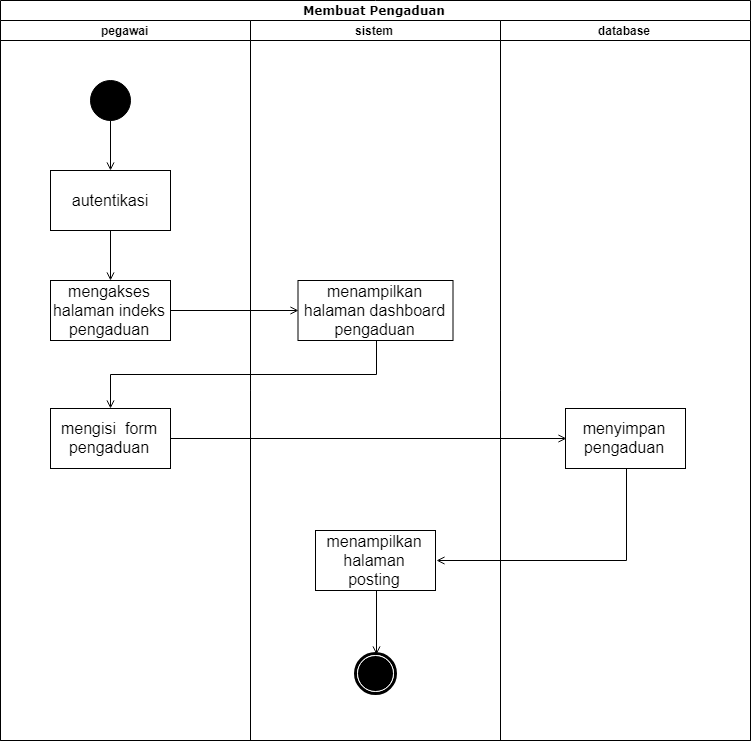
### *Activity* *Diagram*

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



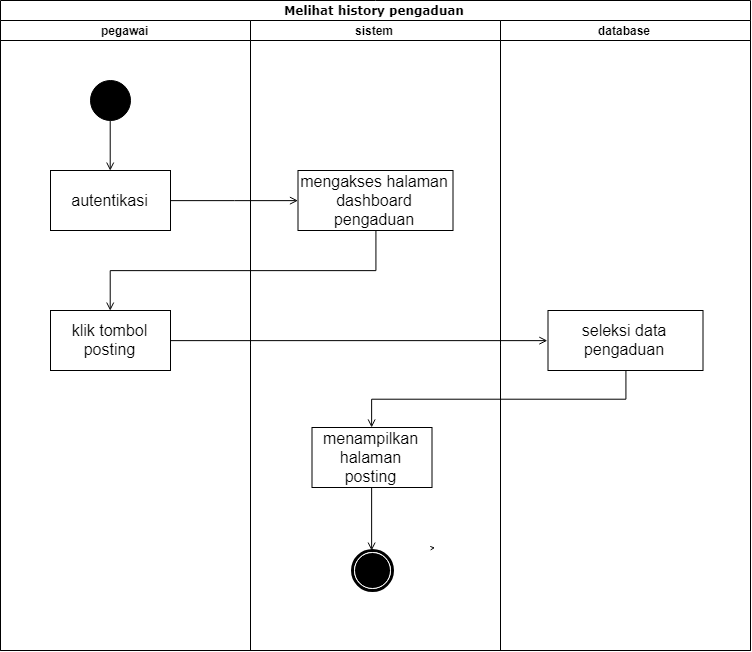
Gambar 3. 5 *Activity Diagram* Autentikasi

Gambar 3.5 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.



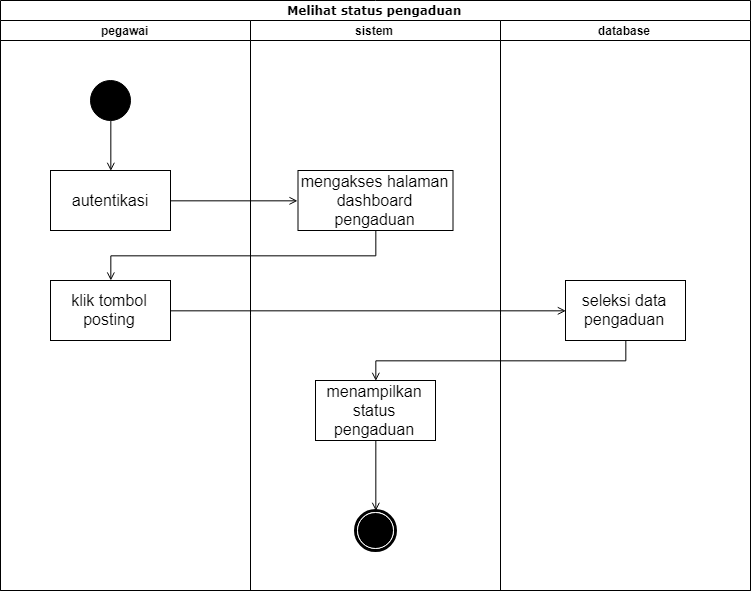
Gambar 3. 6 *Activity Diagram* Melakukan Pengaduan

Gambar 3.6 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.



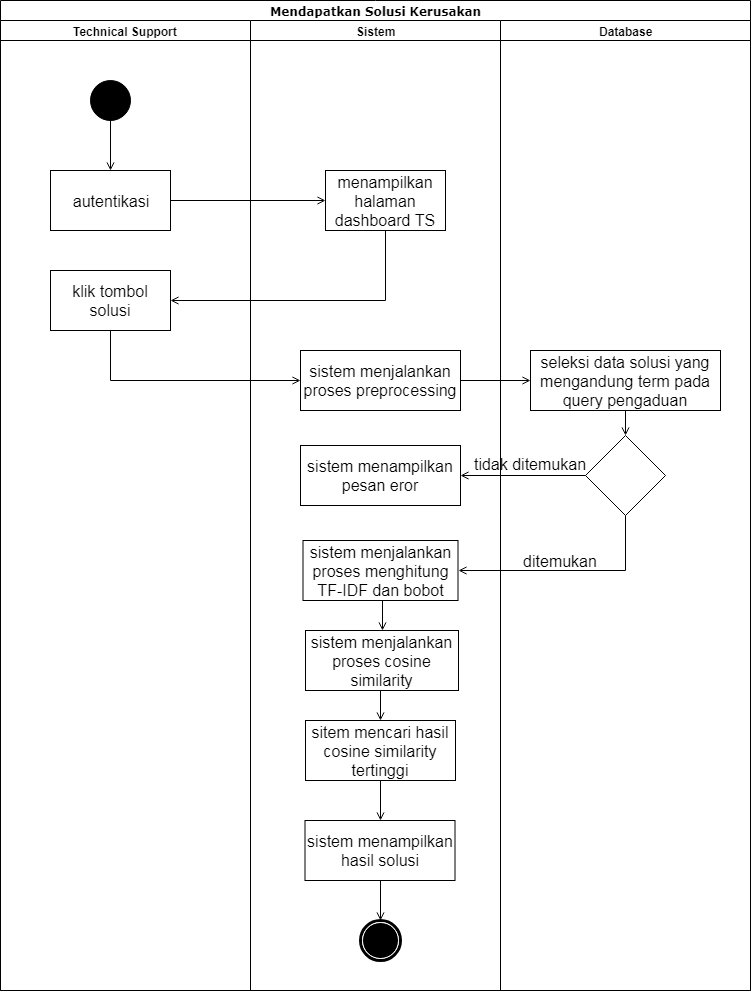
Gambar 3. 7 *Activity Diagram* Melihat *History* Pengaduan

Gambar 3.7 menjelaskan alur aktivitas yang terdapat pada proses melihat *history* pengaduan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian sistem mengakses halaman dashboard pengaduan. *Technical support* selaku *user* klik tombol posting yang terdapat pada bagian navigasi. *Database* menyeleksi data pengaduan, kemudian sistem menampilkan halaman posting.



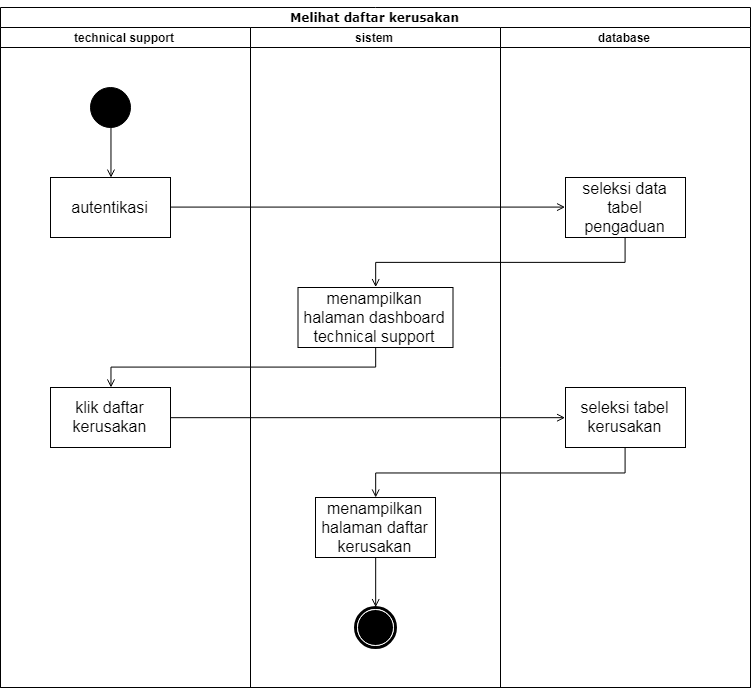
Gambar 3. 8 *Activity Diagram* Melihat Status Pengaduan

Gambar 3.8 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.



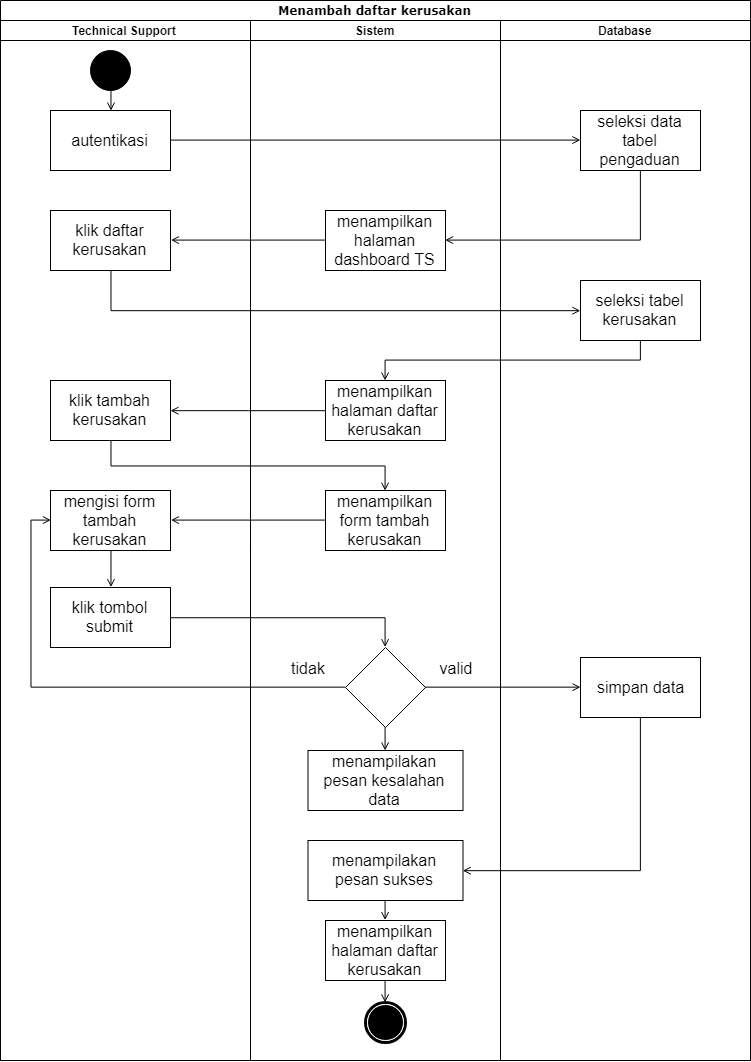
Gambar 3. 9 *Activity Diagram* Mendapatkan Solusi Kerusakan

Gambar 3.9 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. Kemudian sistem melakukan proses *preprocessing*. Database menyeleksi data solusi yang mengandung *term* pada *query* pengaduan. Apabila *term* tidak ditemukan maka sistem akan memuculkan pesan error, dan bila ditemukan maka sistem menjalankan proses menghitung TF-IDF dan bobot, proses *cosine similarity,* dan perncarian hasil *cosine similarity tertinggi*. Setelah semua proses dijalankan kemudian sistem menampilkan hasil solusi kerusakan berdasarkan hasil *cosine* tertinggi.



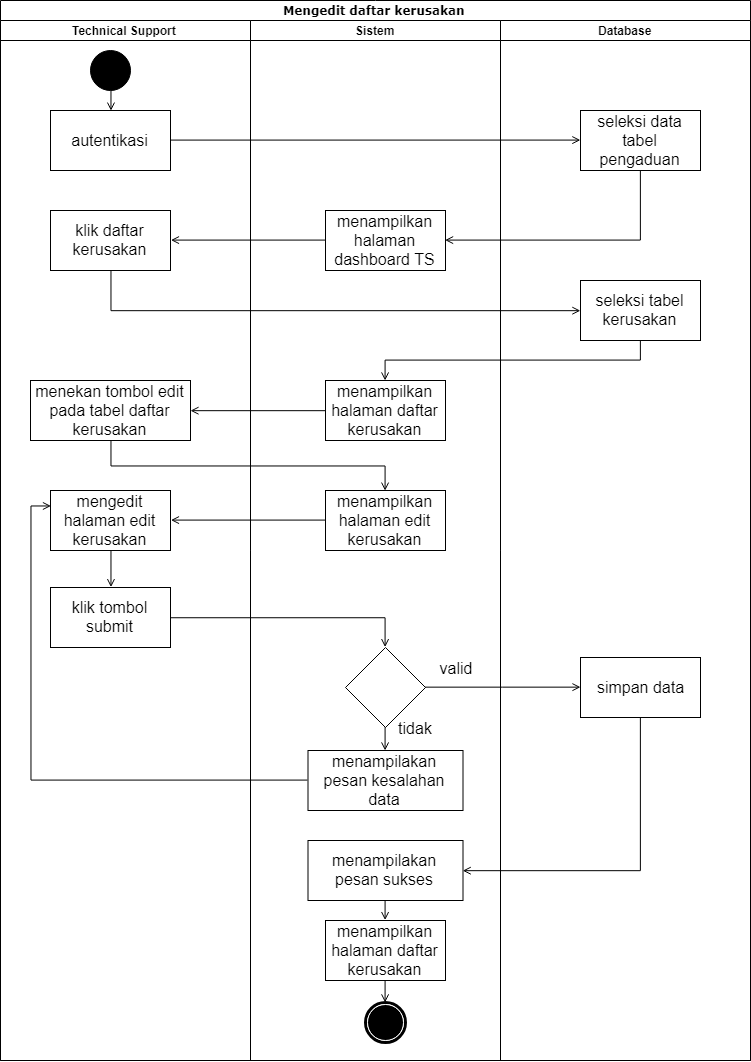
Gambar 3. 10 *Activity Diagram* Melihat Daftar Kerusakan

Gambar 3.10 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.



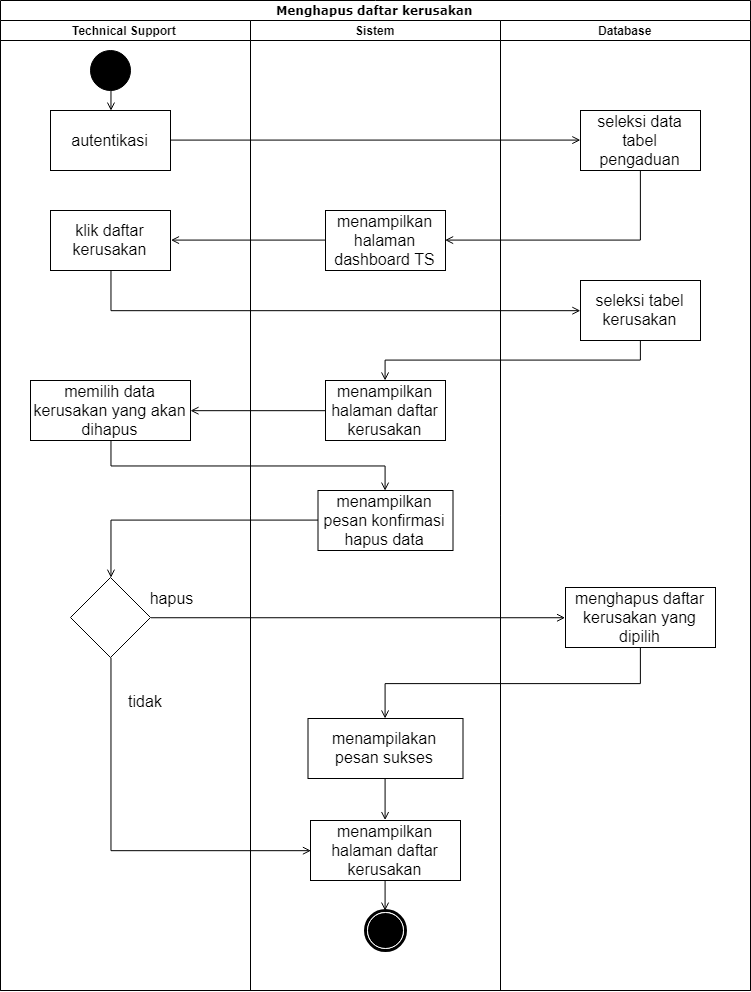
Gambar 3. 11 *Activity Diagram* Menambah Daftar Kerusakan

Gambar 3.11 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.



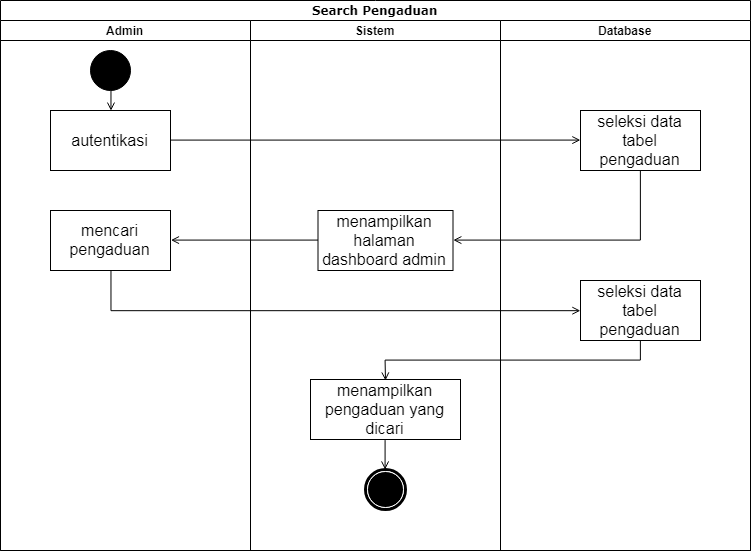
Gambar 3. 12 *Activity Diagram* Edit Daftar Kerusakan

Gambar 3.12 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.



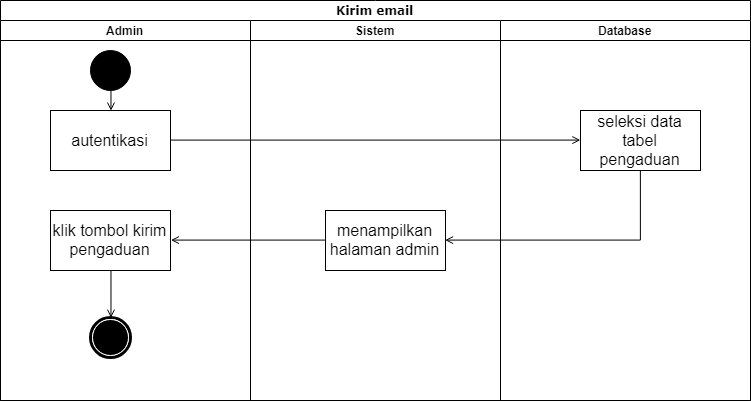
Gambar 3. 13 *Activity Diagram* Hapus Daftar Kerusakan

Gambar 3.13 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.



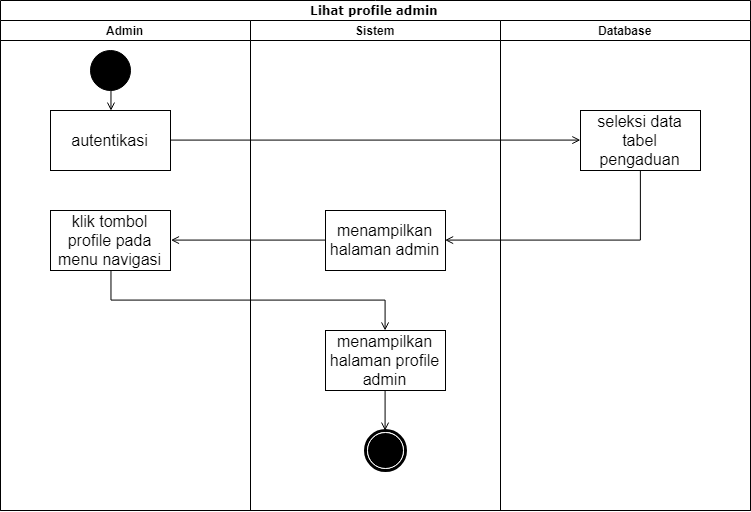
Gambar 3. 14 *Activity Diagram* *Search* Data Pengaduan

Gambar 3.14 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.



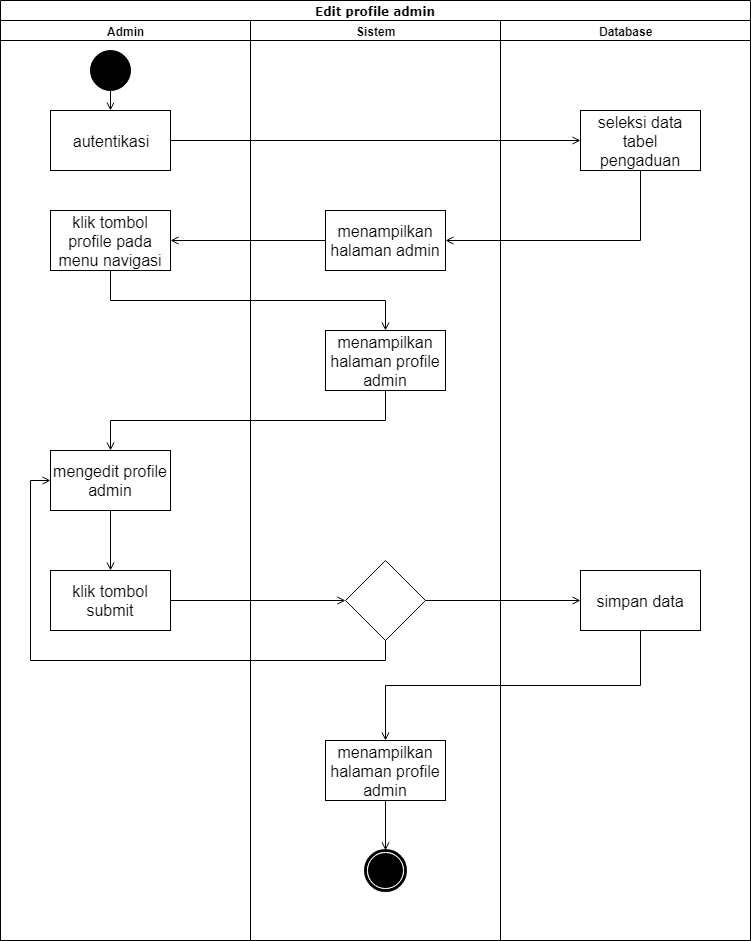
Gambar 3. 15 *Activity Diagram* Kirim Email Data Pengaduan

Gambar 3.15 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.



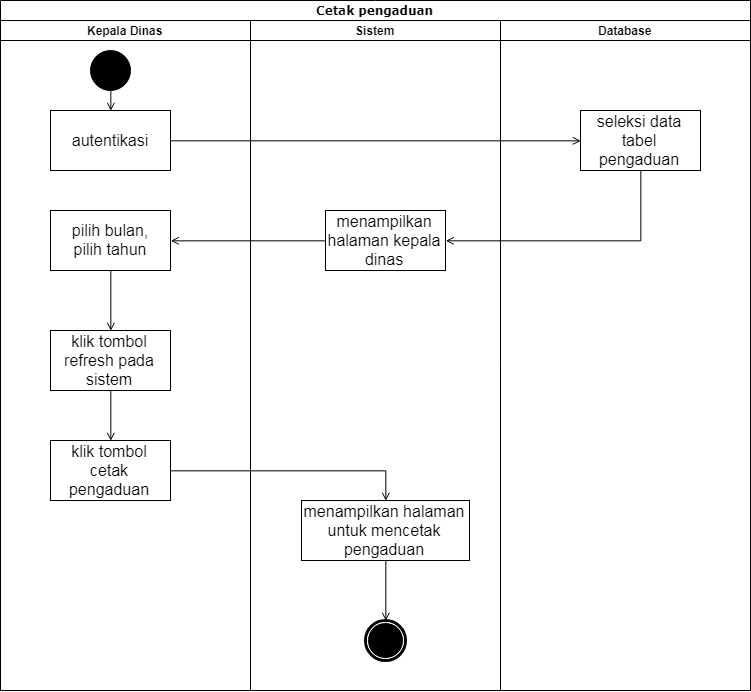
Gambar 3. 16 *Activity Diagram* Melihat Profile Admin

Gambar 3.16 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.



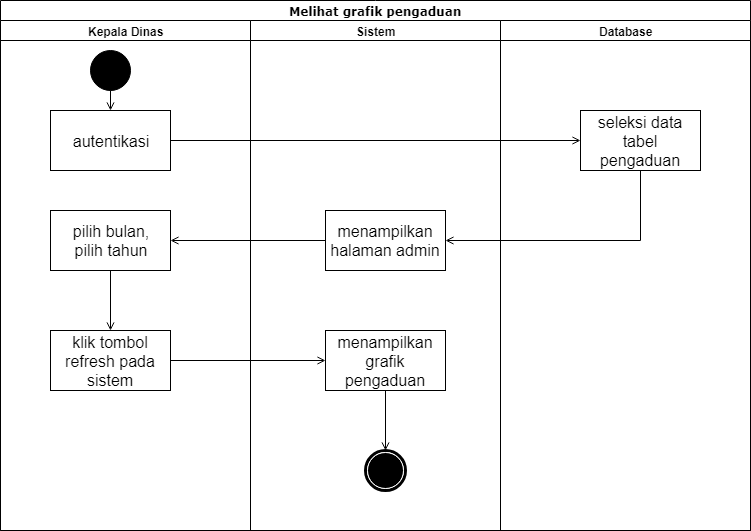
Gambar 3. 17 *Activity Diagram* Mengedit Profile Admin

Gambar 3.17 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 3. 18 *Activity Diagram* Mencetak Data Pengaduan

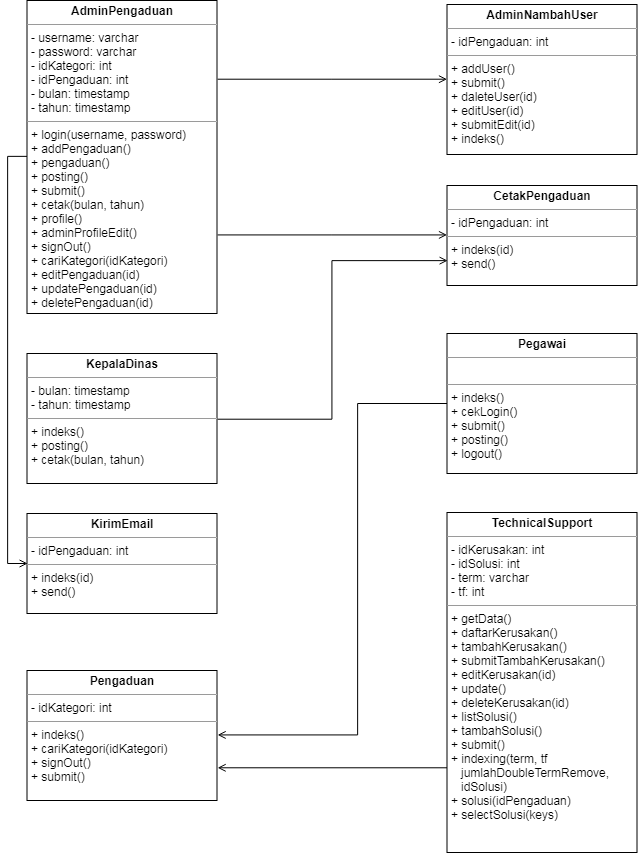
Gambar 3.18 menjelaskan alur aktivitas yang terdapat pada proses mencetak data pengaduan. Hal pertama yang dilakukan adalah proses autentikasi, kemudian database menyeleksi data pada tabel pengaduan. Sistem menampilkan halaman kepala dinas. Kepala dinas memilih bulan dan tahun pada menu yang disediakan, lalu klik tombol refresh pada sistem. Setelah itu kepala dinas klik tombol cetak pengaduan. Kemudian sistem menampilkan halaman untuk mencetak pengaduan.



Gambar 3. 19 *Activity Diagram* Melihat Grafik

Gambar 3.19 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 3. 20 *Class Diagram*

Gambar3.20Merupakan 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 3.8.

Tabel 3. 9 Hubungan *Class* dan *Use Case* Diagram

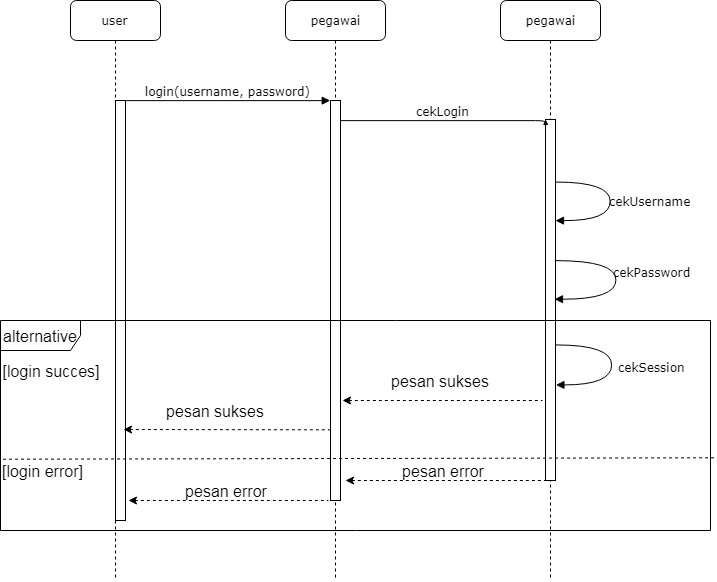
|  |  |  |
| --- | --- | --- |
| **Kelas** | ***Use Case*** | **Fungsi** |
| AdminPengaduan | Autentikasi | Menangani operasi pengelolaan dan akses data admin pada sistem |
| Mengelola data pengaduan |
| Mengelola daftar user |
| Mengelola daftar instansi |
| Mengelola profile admin |
| *Search* pengaduan |
| 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 |

**Tabel 3.9 Hubungan *Class* dan *Use Case* Diagram (lanjutan)**

|  |  |  |
| --- | --- | --- |
| 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* |

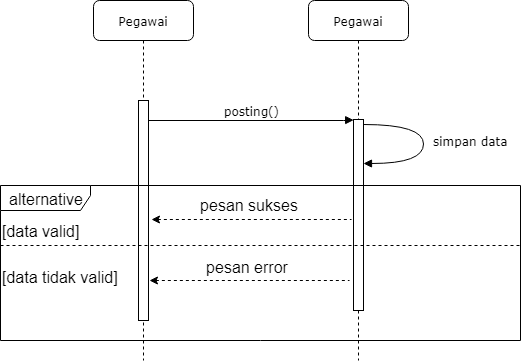
### *Sequence* *Diagram*

*Sequence diagram* menggambarkan perilaku *class* pada *use case* dengan mendeksripsikan waktu hidup objek dan pesan yang dikirim dan diterima antar *class.*



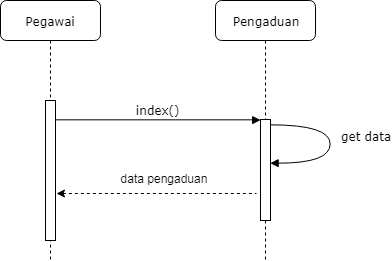
Gambar 3. 21 *Sequence Diagram* Autentikasi

Pada Gambar 3.21 menjelaskan perilaku dan waktu yang digunakan *class* pada proses autentikasi*.* Pertama, *user* mengakses halaman *login* melalui panggilan fungsi “*login()*” dengan parameter *username* dan *password* pada kelas “*Pegawai()*”. Kemudian setelah *user* klik tombol *login* maka fungsi “*cekLogin()*”dijalankan pada kelas “*Pegawai()*”. Sistem mengecek kesesuaian *username* dan *password* pada database sistem. Apabila sesuai maka sistem membuat *session* dan *user* berhasil login dan bila tidak sesuai maka *user* tidak dapat login.



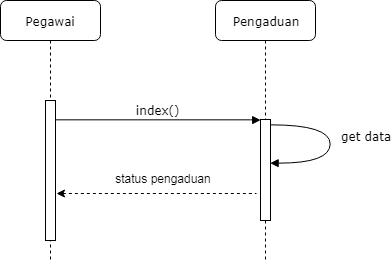
Gambar 3. 22 *Sequence Diagram* Melakukan Pengaduan

*Sequence diagram* pada Gambar 3.22 menjelaskan perilaku dan waktu yang digunakan *class* pada proses melakukan pengaduan*.* Ketika Pegawai sudah selesai mengisi form pengaduan maka akan dijalankan fungsi “*posting()*” pada kelas “*Pegawai()*”. Kemudian data disimpan. Apabila data valid maka akan dikirim pesan sukses dan bila tidak valid makan akan dikirim pesan error.



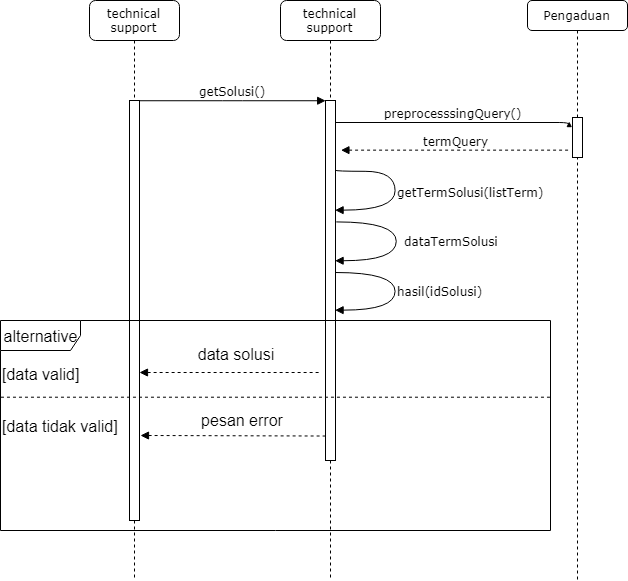
Gambar 3. 23 *Sequence Diagram* Melihat *History* Pengaduan

*Sequence diagram* pada Gambar 3.23 menjelaskan perilaku dan waktu yang digunakan *class* pada proses melihat *history* pengaduan*.* Ketika Pegawai ingin melihat *history* pengaduan maka akan dijalankan fungsi “*index()*” pada kelas “*Pengaduan()*”. Kemudian dijalankan fungsi “*getData()*” yang akan mengambil seluruh data pengaduan yang ada pada database, selanjutnya data pengaduan ditampilkan.



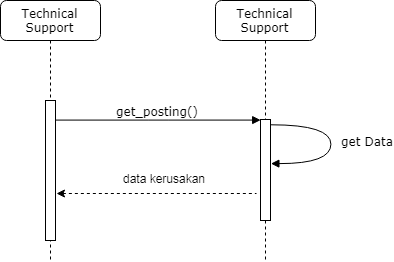
Gambar 3. 24 *Sequence Diagram* Melihat StatusPengaduan

*Sequence diagram* pada Gambar 3.24 menjelaskan perilaku dan waktu yang digunakan *class* pada proses melihat statuspengaduan*.* Ketika Pegawai ingin melihat statuspengaduan maka akan dijalankan fungsi “*index()*” pada kelas “*Pengaduan()*”. Kemudian dijalankan fungsi “*getData()*” yang akan mengambil data status pengaduan yang ada pada database, selanjutnya data status pengaduan ditampilkan.



Gambar 3. 25 *Sequence Diagram* Mendapatkan SolusiPengaduan

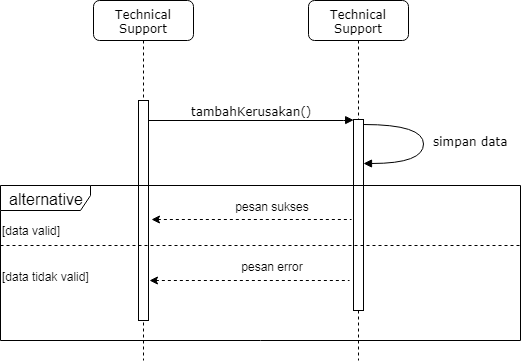
*Sequence diagram* pada Gambar 3.25 menjelaskan perilaku dan waktu yang digunakan *class* pada proses mendapatkan solusi pengaduan*.* Pertama *Technical Support* klik tombol solusi pada tabel pengaduan, kemudian dijalankan fungsi “*getSolusi()*” pada kelas “*Technical Support()*”. Kemudian sistem menjalankan *preprocessing* data pada *query* dengan menjalankan fungsi “*preprocessingQery()*” yang akan menghasilkan data listTerm dari *query*. Kemudian sistem mencari data solusi pada tabel *indexing* yang mengandung *term query* dengan fungsi “*getTermSolusi(listTerm)*”. Setiap *term* solusi yang sesuai dengan *query* akan dilakukan perhitungan bobot solusi dan hitung algoritma *cosine similarity* melalui fungsi “*hasil(idSolusi)*”. Apabila data valid maka akan dikirim data solusi dan bila tidak valid makan akan dikirim pesan error.

****

Gambar 3. 26 *Sequence Diagram* Melihat Daftar Kerusakan

*Sequence diagram* pada Gambar 3.26 menjelaskan perilaku dan waktu yang digunakan *class* pada proses melihat daftarkerusakan*.* Ketika *Technical Support* ingin melihat daftarkerusakan maka akan dijalankan fungsi “*get\_posting()*” pada kelas “*TechnicalSupport()*”. Kemudian mengambil data kerusakan yang ada pada database, selanjutnya data kerusakan ditampilkan.

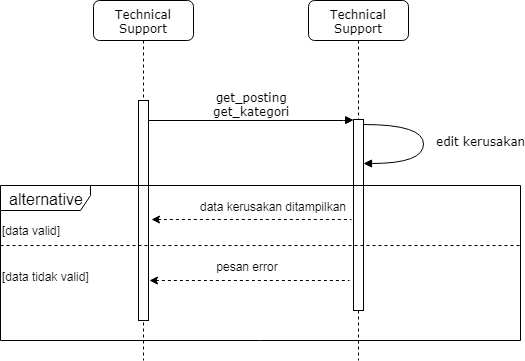
*Sequence diagram* pada *use case* lihat list solusi, *use case* lihat data pengaduan, *use case* lihat daftar user, dan *use case* lihat daftar instansi memiliki karakteristik yang sama dengan *sequence diagram* melihat daftar kerusakan. Sehingga hanya dibuat satu buah *sequence diagram* melihat daftar kerusakan saja.



Gambar 3. 27 *Sequence Diagram* Menambah Daftar Kerusakan

*Sequence diagram* pada Gambar 3.27 menjelaskan perilaku dan waktu yang digunakan *class* pada proses menambah daftar kerusakan*.* Ketika *Technical Support* ingin menambah daftar kerusakan maka akan dijalankan fungsi “*tambahKerusakan()*” pada kelas “*TechnicalSupport ()*”. Kemudian data disimpan. Apabila data valid maka akan dikirim pesan sukses dan bila tidak valid makan akan dikirim pesan error.

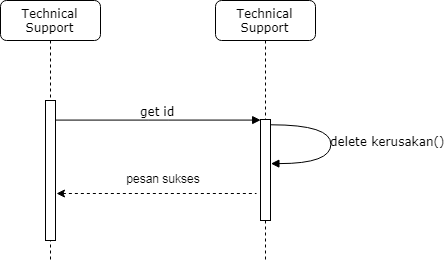
*Sequence diagram* pada *use case* tambah list solusi, *use case* tambah data pengaduan, *use case* tambah daftar user, dan *use case* tambah daftar instansi memiliki karakteristik yang sama dengan *sequence diagram* tambah daftar kerusakan. Sehingga hanya dibuat satu buah *sequence diagram* tambah daftar kerusakan saja.



Gambar 3. 28 *Sequence Diagram* Edit Kerusakan

*Sequence diagram* pada Gambar 3.28 menjelaskan perilaku dan waktu yang digunakan class pada proses edit kerusakan. Ketika *technical* *support* klik tombol “Edit” pada tabel kerusakan maka akan dilakukan pemanggilan fungsi “*editKerusakan()*” pada kelas “*TechnicalSupport()*” dengan parameter “id” kerusakan. Kemudian akan dijalankan fungsi “*get\_posting()*” dan “*get\_kategori()*” pada kelas “*TechnicalSupport()*” untuk mengambil data kerusakan dan kategori kerusakan berdasarkan parameter idKerusakan untuk ditampilkan pada form edit kerusakan. Setelah data pada form telah diedit dan *technical support* mengklik tombol “Submit” maka akan dijalankan fungsi “*update()*” pada kelas “*TechnicalSupport()*” untuk melakukan proses perubahan data. Sistem akan menyimpan perubahan data kerusakan. Ketika berhasil maka perubahan data akan disimpan dengan fungsi “*submit()*” dan sistem mengirimkan pesan sukses. Jika data tidak valid maka akan dikirimkan pesan error.

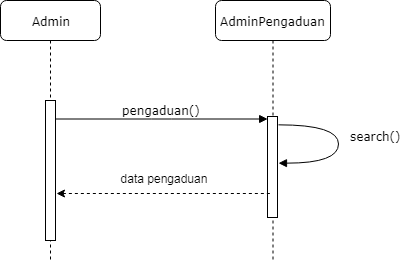
*Sequence diagram* pada *use case* edit list solusi, *use case* edit data pengaduan, *use case* edit daftar user, dan *use case* edit daftar instansi memiliki karakteristik yang sama dengan *sequence diagram* edit daftar kerusakan. Sehingga hanya dibuat satu buah *sequence diagram* edit daftar kerusakan saja.



Gambar 3. 29 *Sequence Diagram* Hapus Kerusakan

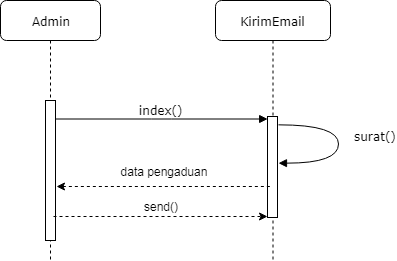
*Sequence diagram* pada Gambar 3.29 menjelaskan perilaku dan waktu yang digunakan class pada proses hapus kerusakan. Ketika *technical support* mengklik tombol “Hapus” pada tabel kerusakan maka akan dijalankan fungsi “*getId()*”pada kelas “*TechnicalSupport()*”. Kemudian sistem menjalankan fungsi “*deleteKerusakan()*”, lalu data kerusakan terhapus dan muncul pesan sukses.

*Sequence 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 *sequence diagram* hapus daftar kerusakan. Sehingga hanya dibuat satu buah *sequence diagram* hapus daftar kerusakan saja.



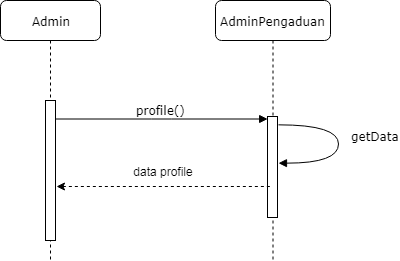
Gambar 3. 30 *Sequence Diagram Search* Pengaduan

*Sequence diagram* pada Gambar 3.30 menjelaskan perilaku dan waktu yang digunakan class pada proses *search* pengaduan. Ketika admin ingin mencari pengaduan, admin dapat mencari pengaduan berdasarkan nomor surat atau nama instansi. Pertama, akan dijalankan fungsi “*pengaduan()*” pada kelas “*AdminPengaduan()*”. Kemudian sistem menjalankan fungsi “*search()*”, lalu data pengaduan yang dicari ditampilkan oleh sistem.



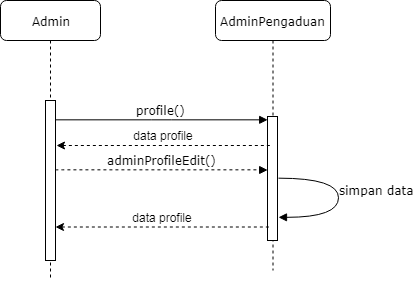
Gambar 3. 31 *Sequence Diagram* Kirim Pengaduan

*Sequence diagram* pada Gambar 3.31 menjelaskan perilaku dan waktu yang digunakan class pada proses kirim pengaduan. Proses mengirim pengaduan dilakukan melalui bantuan library fpdf dan dikirim melalui email. Pertama, akan dijalankan fungsi “*index()*” pada kelas “*KirimEmail()*”. Kemudian sistem menjalankan fungsi “*surat()*”, lalu admin mengirim data pengaduan yang diinginkan. Setelah itu dijalankan fungsi “*send()*”.



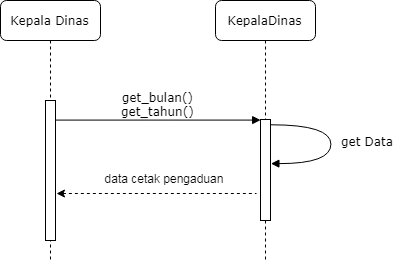
Gambar 3. 32 *Sequence Diagram* Melihat *Profile* Admin

*Sequence diagram* pada Gambar 3.32 menjelaskan perilaku dan waktu yang digunakan *class* pada proses melihat *profile* admin*.* Ketika admin ingin melihat *profile* admin maka akan dijalankan fungsi “*profile()*” pada kelas “*AdminPengaduan()*”. Kemudian mengambil data *profile* admin yang ada pada database, selanjutnya data *profile* ditampilkan.



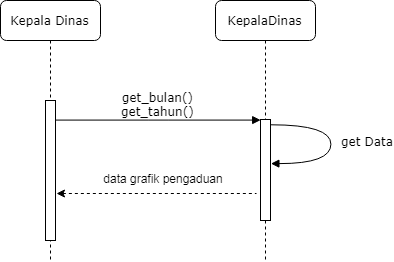
Gambar 3. 33 *Sequence Diagram* Mengedit *Profile* Admin

*Sequence diagram* pada Gambar 3.33 menjelaskan perilaku dan waktu yang digunakan class pada proses edit *profile* admin. Ketika admin klik tombol “Profile” pada menu navigasi maka akan dilakukan pemanggilan fungsi “*profile()*” pada kelas “*AdminPengaduan()*”. Kemudian data profile ditampilkan dan admin mengedit *profile* dengan dijalankan fungsi “*adminProfileEdit()*” pada kelas “*AdminPengaduan()*”. Setelah admin selesai mengedit *profile* lalu admin mengklik tombol “Submit” maka akan sistem akan menyimpan perubahan data *profile* admin. Kemudian data profile yang telah diedit ditampilkan oleh sistem.



Gambar 3. 34 *Sequence Diagram* Cetak Pengaduan

*Sequence diagram* pada Gambar 3.34 menjelaskan perilaku dan waktu yang digunakan *class* pada proses cetak pengaduan*.* Ketika kepala dinas ingin mencetak pengaduan maka akan dijalankan fungsi “*get\_bulan()*” dan “*get\_tahun()*” pada kelas “*KepalaDinas()*”. Kemudian mengambil data pengaduan berdasarkan bulan dan tahun yang dipilih oleh kepala dinas. Selanjutnya data cetak pengaduan ditampilkan.

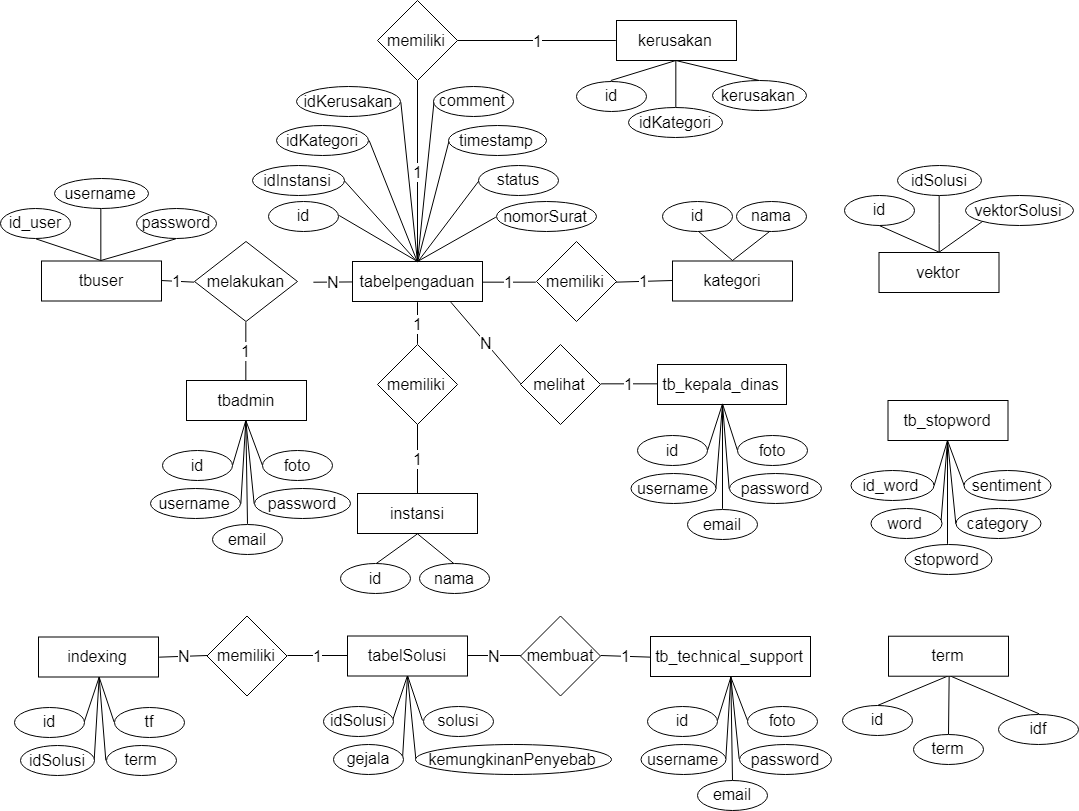


Gambar 3. 35 *Sequence Diagram* Melihat Grafik Pengaduan

*Sequence diagram* pada Gambar 3.35 menjelaskan perilaku dan waktu yang digunakan *class* pada proses melihat grafik pengaduan*.* Ketika kepala dinas ingin melihat grafik pengaduan maka akan dijalankan fungsi “*get\_bulan()*” dan “*get\_tahun()*” pada kelas “*KepalaDinas()*”. Kemudian mengambil data grafik pengaduan berdasarkan bulan dan tahun yang dipilih oleh kepala dinas. Selanjutnya data grafik pengaduan ditampilkan.

### Entity Relational Diagram

Berdasarkan hasil dari analisis kebutuhan dan rancangan *use case diagram* berikut adalah ERD dari sistem pengaduan untuk mendefinisikan entitas yang terlibat serta relasi antar entitas.



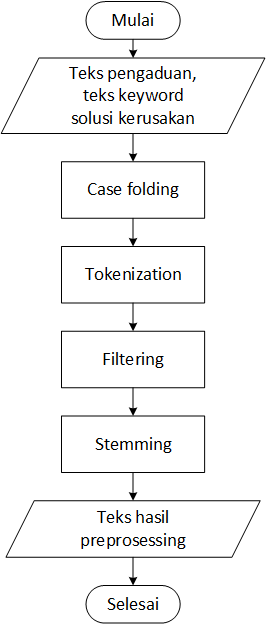
Gambar 3. 36 *Entity Relational Diagram*

ERD yang dirancang terdiri dari 13 entitas yang terlibat diantaranya entitas tbuser, tabelpengaduan, kategori, instansi, kerusakan, tbadmin, tb\_kepala\_dinas, indexing, tabelSolusi, tb\_technical\_support, term, tb\_stopword, dan vektor. Relasi yang terjadi antar entitas diatas yaitu:

1. Relasi antara tbuser dengan tabelpengaduan, memiliki sifat *one to many* (1:N), dimana setiap baris data pada tbuser dapat berhubungan dengan banyak baris data pada tabelpengaduan. Karena setiap user dapat membuat banyak pengaduan.
2. Relasi antara tbadmin dengan tabelpengaduan, memiliki sifat *one to many* (1:N), dimana setiap baris data pada tbadmin dapat berhubungan dengan banyak baris data pada tabelpengaduan. Karena seorang admin dapat membuat banyak pengaduan.
3. Relasi antara instansi dengan tabelpengaduan, memiliki sifat *one to one* (1:1), dimana setiap baris data pada tabelpengaduan dapat berhubungan dengan satu baris data pada instansi. Karena setiap satu instansi hanya dapat memiliki satu pengaduan pada setiap kali mengadu.
4. Relasi antara kerusakan dengan tabelpengaduan, memiliki sifat *one to one* (1:1), dimana setiap baris data pada tabelpengaduan dapat berhubungan dengan satu baris data pada kerusakan. Karena setiap satu kerusakan hanya dapat memiliki satu pengaduan pada setiap kali mengadu.
5. Relasi antara kategori dengan tabelpengaduan, memiliki sifat *one to one* (1:1), dimana setiap baris data pada tabelpengaduan dapat berhubungan dengan satu baris data pada kategori. Karena setiap satu kategori hanya dapat memiliki satu pengaduan pada setiap kali mengadu.
6. Relasi antara tb\_kepala\_dinas dengan tabelpengaduan, memiliki sifat *one to many* (1:N), dimana setiap baris data pada tb\_kepala\_dinas dapat berhubungan dengan banyak baris data pada tabelpengaduan. Karena seorang kepala dinas dapat melihat banyak pengaduan.
7. Relasi antara tabelSolusi dengan indexing, memiliki sifat *one to many* (1:N), dimana setiap baris data pada tabelSolusi dapat berhubungan dengan banyak baris data pada indexing. Karena setiap solusi memiliki banyak indexing.
8. Relasi antara tb\_technical\_support dengan tabelSolusi, memiliki sifat *one to many* (1:N), dimana setiap baris data pada tb\_technical\_support dapat berhubungan dengan banyak baris data pada tabelSolusi. Karena seorang *technical support* dapat membuat banyak solusi pengaduan.
9. Tabel term, stopword, dan vektor tidak memiliki relasi

### 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 :



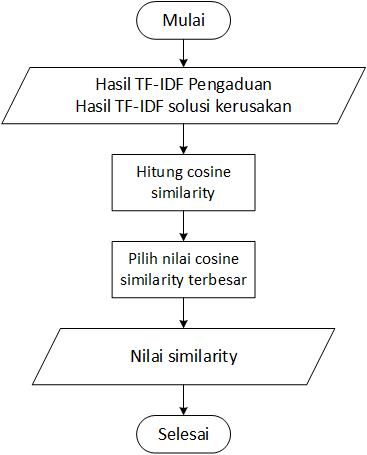
Gambar 3. 37 *Flowchart* *Preprocessing*

*Flowchart preprocessing* pada Gambar 3.37 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 menghilangkan karakter tanda baca dan mengubah semua huruf kapital menjadi huruf kecil, lalu memecah kalimat menjadi kata *(tokenization)*. Kata-kata yang tidak memiliki deskriptif dihilangkan melalui pengecekan kata yang terdapat dalam daftar kata pada yang tidak penting pada *stopwords (filtering)*. 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 *Solusi*  H:\..KULI AH\PROPOSAL ASTUNGKARA GASS\dfd\flowchart perhitungan TF-IDF solusi.jpg  (b) |

Gambar 3. 38 *Flowchart* TF-IDF

Gambar 3.38 (a) merupakan proses menghitung bobot TF-IDF untuk teks pengaduan dan Gambar 3.38 (b) merupakan proses menghitung bobot TF-IDF untuk teks solusi kerusakan. Inputan pada *flowchart* ini adalah teks pengaduan dan teks 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 3. 39 *Flowchart* Perhitungan Kemiripan Teks

*Flowchart* Perhitungan Kemiripan Teks pada Gambar 3.39 merupakan proses menghitung kemiripan antara teks pengaduan dan teks solusi kerusakan komputermenggunakan *cosine similarity.* Inputan pada *flowchart* ini adalah hasil TF-TDF Pengaduan dan 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 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 3.40.



Gambar 3. 40 Desain Tampilan Login

1. Tampilan Halaman Awal Pegawai

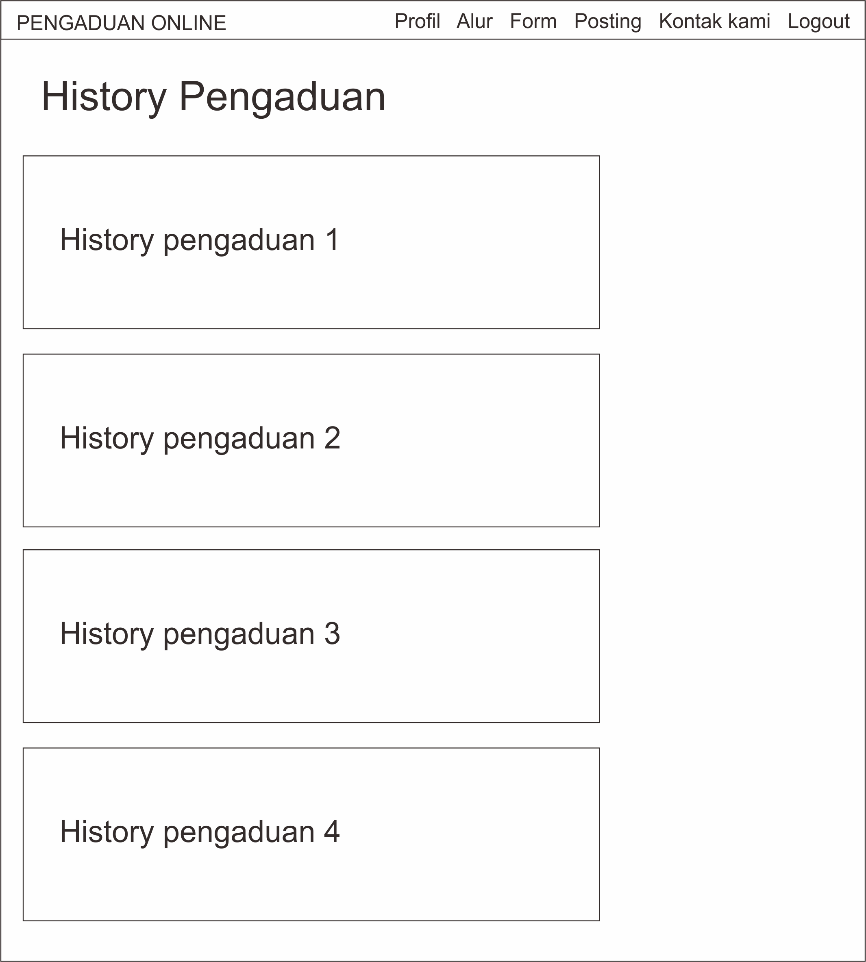
Rancangan tampilan halaman awal pegawai merupakan tampilan awal yang muncul setelah pegawai melalui proses login. Tampilan halaman awal pegawai terdapat menu navigasi (profil, alur, form, posting, kontak kami, dan logout), header, profil, alur, form, kontak kami. Desain tampilan halaman awal pegawai dapat dilihat seperti pada Gambar 3.41



Gambar 3. 41 Desain Tampilan Halaman Awal Pegawai

1. Tampilan Halaman Posting

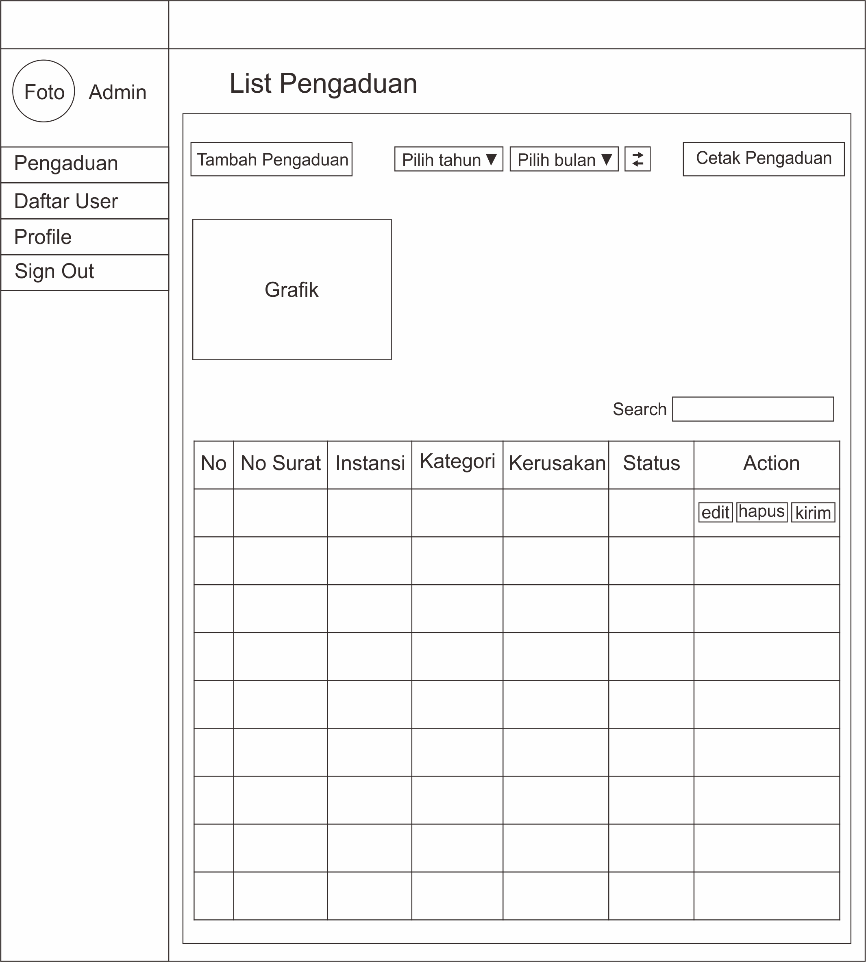
Rancangan tampilan halaman posting merupakan tampilan yang berisi history pengaduan yang dilakukan oleh pegawai. Pada halaman ini pegawai dapat melihat pengaduan yang pernah dilakukan sebelumnya dan pegawai dapat melihat status pengaduan. Status pengaduan ada dua jenis yaitu belum diproses dan sudah diproses. Desain tampilan halaman posting dapat dilihat seperti pada Gambar 3.42.



Gambar 3. 42 Desain Tampilan Halaman Posting

1. Halaman Awal Admin

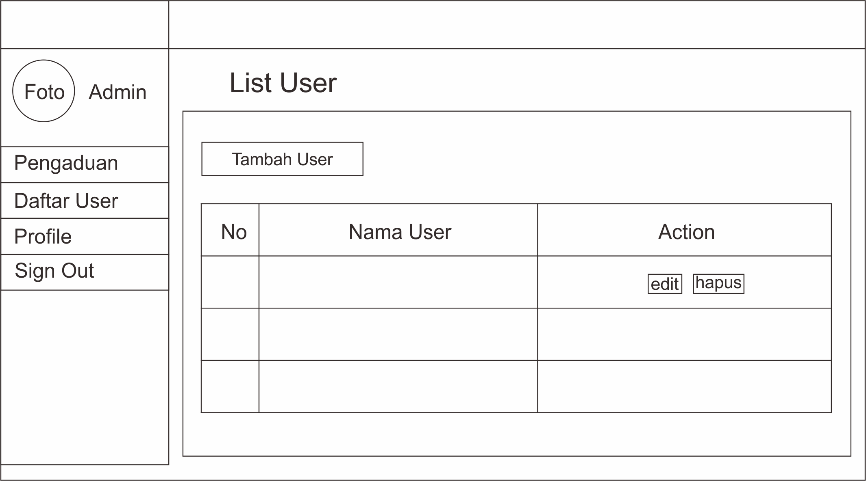
Rancangan tampilan halaman awal admin merupakan tampilan awal yang muncul setelah admin melalui proses login. Tampilan halaman awal admin berisi menu navigasi (pengaduan, daftar user, profile, dan sign out), tambah pengaduan, cetak pengaduan, pilih tahun dan bulan pengaduan yang berguna memunculkan grafik, cari pengaduan, dan tabel pengaduan. Desain tampilan halaman awal admin dapat dilihat seperti pada Gambar 3.43.



Gambar 3. 43 Desain Tampilan Halaman Awal Admin

1. Halaman Daftar User

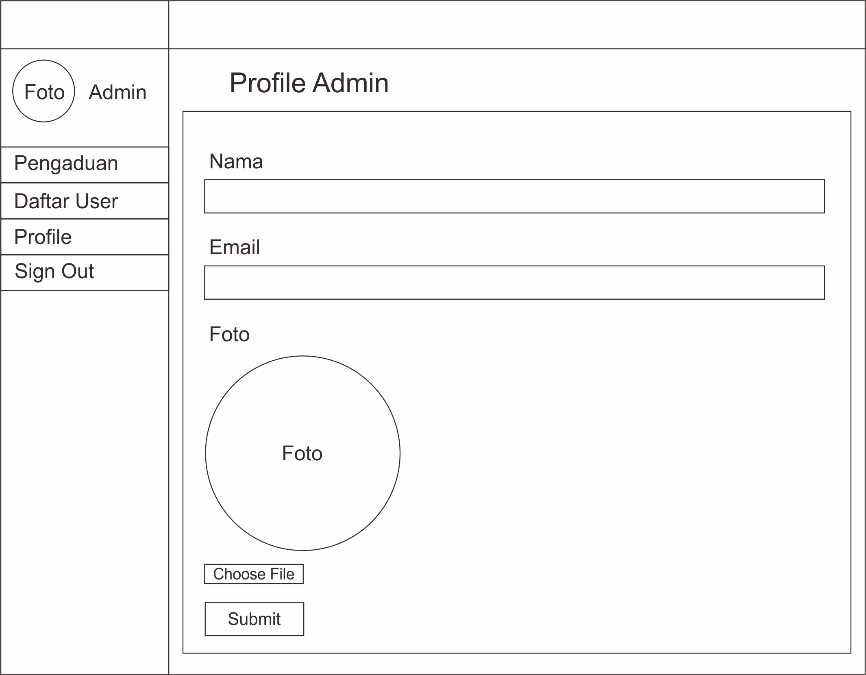
Halaman list user merupakan tampilan tempat mengelola daftar user untuk admin. Halaman ini berisi menu navigasi (pengaduan, daftar user, profile, dan sign out), tambah user, edit user, dan hapus user. Desain tampilan halaman daftar user dapat dilihat seperti pada Gambar 3.44.



Gambar 3. 44 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 3.45.



Gambar 3. 45 Desain Tampilan Halaman Profile

1. Halaman Awal *Technical Support*

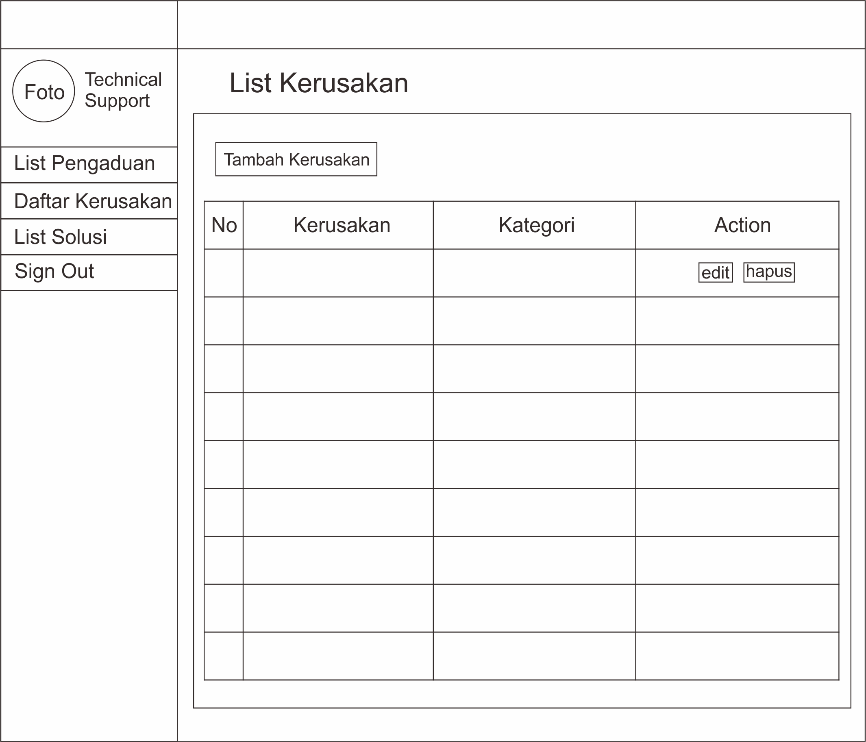
Rancangan tampilan halaman awal *technical support* merupakan tampilan awal yang muncul setelah *technical support* melalui proses login. Tampilan halaman awal *technical support* berisi menu navigasi (list pengaduan, daftar kerusakan, list solusi, dan sign out), tabel pengaduan, menu solusi. Desain tampilan halaman awal *technical support* dapat dilihat seperti pada Gambar 3.46.



Gambar 3. 46 Desain Tampilan Halaman Awal *Technical Support*

1. Halaman Daftar Kerusakan

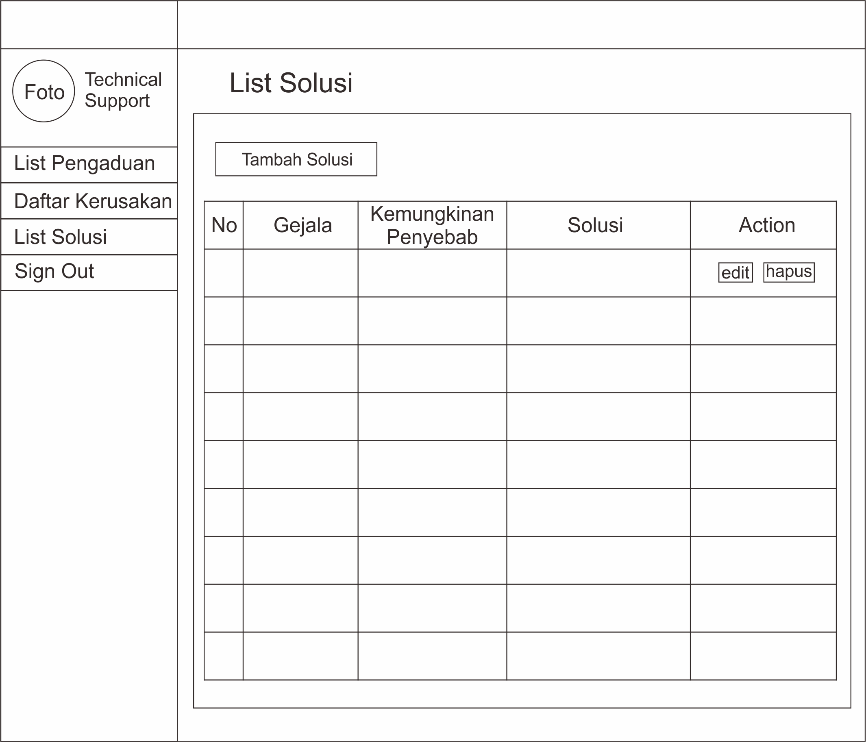
Halaman daftar kerusakan merupakan halaman yang digunakan *technical support* untuk melihat daftar kerusakan yang ada pada sistem. Halaman ini berisi menu navigasi (pengaduan, daftar user, profile, dan sign out). Pada halaman ini *technical support* dapat menambah, mengedit, dan menghapus kerusakan. Desain tampilan halaman daftar kerusakan dapat dilihat seperti pada Gambar 3.47.



Gambar 3. 47 Desain Tampilan Halaman Daftar Kerusakan

1. Halaman List Solusi

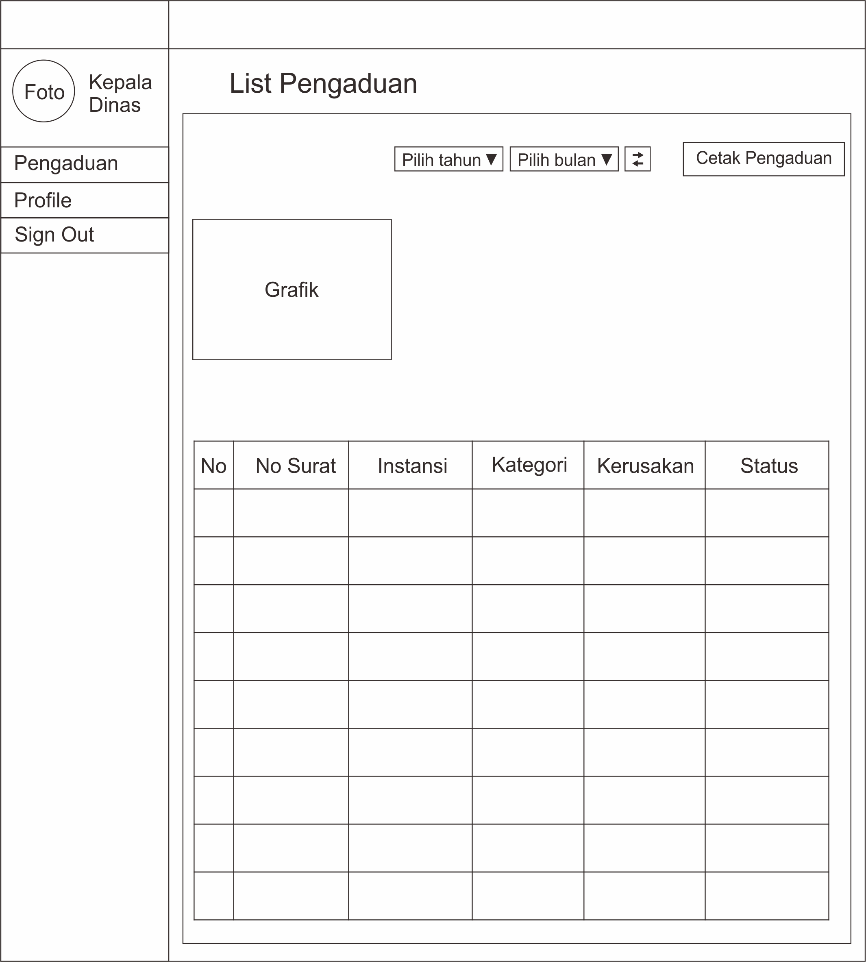
Halaman list solusi merupakan halaman yang digunakan *technical support* untuk melihat solusi yang ada pada sistem. Halaman ini berisi menu navigasi (pengaduan, daftar user, profile, dan sign out). Pada halaman ini *technical support* dapat menambah, mengedit, dan menghapus solusi. Desain tampilan halaman list solusi dapat dilihat seperti pada Gambar 3.48.



Gambar 3. 48 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 3.49.



Gambar 3. 49 Desain Tampilan Halaman Kepala Dinas

# BAB IV

**HASIL DAN PEMBAHASAN**

## Gambaran Umum Sistem

Sistem pengaduan kerusakan komputer dan rekomendasi solusi menggunakan algoritma *cosine similarity* dibangun dengan metode pengembangan *system waterfall*. Sistem ini dibangun dengan 4 sisi level user yaitu pegawai, technical support, admin, dan kepala dinas. Pada sisi pegawai, sistem dirancang agar dapat memberikan kemudahan dalam melakukan pengaduan secara *online*. Pada sisi *technical support*, sistem dirancang agar dapat memberikan kemudahan dalam memberikan informasi awal kerusakan komputer. Pada sisi admin, sistem dirancang agar dapat memberikan kemudahan dalam mengelola sistem. Kemudian pada sisi kepala dinas, sistem dirancang agar dapat memberikan kemudahan dalam memberikan informasi rekap pengaduan. Pada sistem terdapat fitur solusi yang dibuat dengan algoritma *cosine similarity*. Fitur solusi berguna bagi *technical support* untuk mendapatkan solusi dari kerusakan komputer.

## Lingkungan Implementasi

Pada proses implentasi sistem terdapat komponen pendukung untuk menunjang dan membantu dalam melakukan penelitian. Adapun komponen tersebut :

1. Kebutuhan Perangakat Keras *(Hardware)*

Spesifikasi minimal perangakat keras yang dibutuhkan untuk menunjang dan membantu dalam melakukan penelitian yaitu :

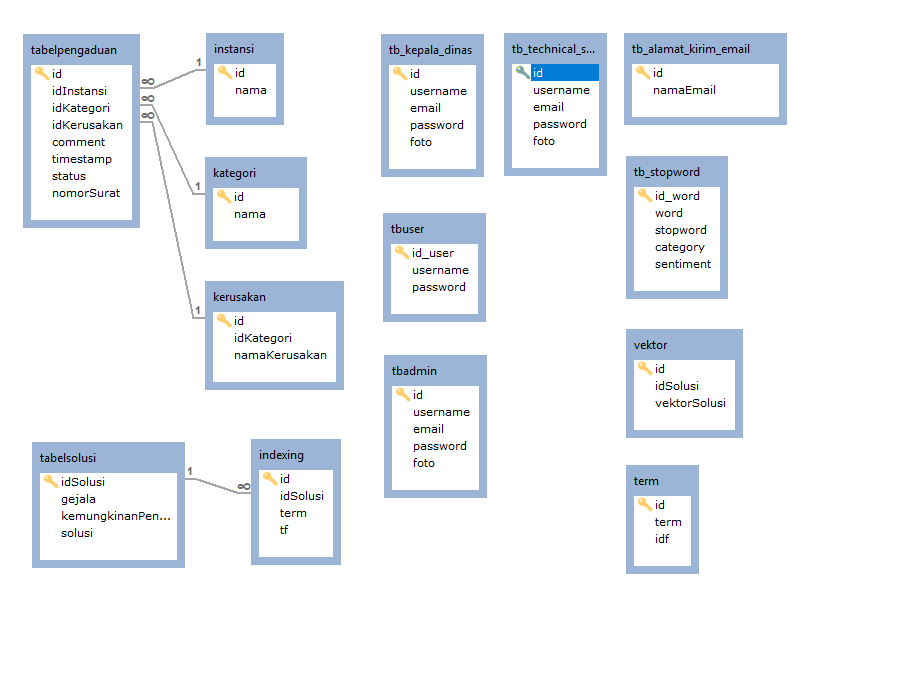
1. Laptop atau *notebook* dengan spesifikasi minimal :

* Processor intel(R) 2.0 GHz
* Memory DDR2 2 GB
* VGA 1 GB
* Harddisk 80 GB

1. Kebutuhan Perangakat Lunak *(Software)*

Kebutuhan perangakat lunak dari tahap penelitian hingga implementasi sistem dilakukan dengan bantuan beberapa aplikasi. Sistem ini dibangun dengan menggunakan teknologi berbasis *web responsive* menggunakan HTML dan CSS, menggunakan bahasa JavaScript dan PHP, dan menggunakan basis data SQLyog.

## Implementasi Basis Data



Gambar 4. 1 Implementasi Basis Data

Adapun deskripsi tabel yang dihasilkan dari skema basis data sistem berdasarkan gambar 4.1 adalah sebagai berikut :

* + 1. Tabel *TabelPengaduan*

Tabel t*abelPengaduan* merupakan tabel yang dihasilkan dari entitas t*abelPengaduan* yang memiliki atribut: id *(primary key)* yang bersifat unik, *idInstansi (foreign key)* digunakan untuk menampung idInstansi, *idKategori (foreign key)* digunakan untuk menampung idKategori, *idKerusakan (foreign key)* digunakan untuk menampung idKerusakan, *comment* digunakan untuk menampung deskripsi lebih detail dari kerusakan atau kerusakan baru yang tidak ada pada sistem, *timestamp* digunakan untuk menampung waktu pengaduan, *status* digunakan untuk menampung status pengaduan, dan *nomorSurat* digunakan untuk menampung nomor surat pengaduan.

* + 1. Tabel *Instansi*

Tabel *instansi* merupakan tabel yang dihasilkan dari entitas instansi yang memiliki atribut: id *(primary key)* yang bersifat unik, *nama* digunakan untuk menampung nama instansi seperti Sekretariat Daerah Kota Denpasar.

* + 1. Tabel *Kategori*

Tabel *kategori* merupakan tabel yang dihasilkan dari entitas kategori yang memiliki atribut: id *(primary key)* yang bersifat unik, *nama* digunakan untuk menampung nama kategori seperti *hardware, software,* dan jaringan.

* + 1. Tabel *Kerusakan*

Tabel *kerusakan* merupakan tabel yang dihasilkan dari entitas kerusakan yang memiliki atribut: id *(primary key)* yang bersifat unik, *idKategori* digunakan untuk menampung idKategori, *namaKerusakan* digunakan untuk menampung nama kerusakan seperti kinerja komputer lambat.

* + 1. Tabel *Tabelsolusi*

Tabel *solusi* merupakan tabel yang dihasilkan dari entitas tabelSolusi yang memiliki atribut: idSolusi *(primary key)* yang bersifat unik, *gejala* digunakan untuk menampung deskripsi mengenai gejala kerusakan komputer, *kemungkinanPenyebab* digunakan untuk menampung deskripsi mengenai kemungkinan penyebab kerusakan komputer, *solusi* digunakan untuk menampung deskripsi mengenai solusi kerusakan komputer.

* + 1. Tabel *Indexing*

Tabel *indexing* merupakan tabel yang dihasilkan dari entitas indexing yang memiliki atribut: id *(primary key)* yang bersifat unik, *idSolusi* digunakan untuk menampung idSolusi, *tf* digunakan untuk menampung jumlah kata x pada *query* dan solusi, *term* digunakan untuk menampung kata yang ada pada *query* dan solusi.

* + 1. Tabel *Tb\_kepala\_dinas*

Tabel *tb\_kepala\_dinas* merupakan tabel yang dihasilkan dari entitas tb­\_kepala\_dinas yang memiliki atribut: *id* *(primary key)* yang bersifat unik, *username* digunakan untuk menampung nama kepala dinas, *password* digunakan untuk menampung *password* kepala dinas, *email* digunakan untuk menampung *email* kepala dinas*, foto* digunakan untuk menampung foto kepala dinas.

* + 1. Tabel *Tbuser*

Tabel *tbuser* merupakan tabel yang dihasilkan dari entitas tbuser yang memiliki atribut: *id\_user* *(primary key)* yang bersifat unik, *username* digunakan untuk menampung nama pegawai, *password* digunakan untuk menampung *password* pegawai.

* + 1. Tabel *Tbadmin*

Tabel *tbadmin* merupakan tabel yang dihasilkan dari entitas tbadmin yang memiliki atribut: *id* *(primary key)* yang bersifat unik, *username* digunakan untuk menampung nama admin, *password* digunakan untuk menampung *password* admin, *email* digunakan untuk menampung *email* admin*, foto* digunakan untuk menampung foto admin.

* + 1. Tabel *Tb\_technical\_support*

Tabel *tb\_ technical\_support* merupakan tabel yang dihasilkan dari entitas tb­\_technical\_support yang memiliki atribut: *id* *(primary key)* yang bersifat unik, *username* digunakan untuk menampung nama *technical support*, *password* digunakan untuk menampung *password* *technical support*, *email* digunakan untuk menampung *email technical support, foto* digunakan untuk menampung foto *technical support*.

* + 1. Tabel *Tb\_alamat\_kirim\_email*

Tabel *tb\_alamat\_kirim\_email* merupakan tabel yang dihasilkan dari entitas *tb\_alamat\_kirim\_email* yang memiliki atribut: *id* *(primary key)* yang bersifat unik, *namaEmail* digunakan untuk menampung nama email.

* + 1. Tabel *Tb\_stopword*

Tabel *tb\_stopword* merupakan tabel yang dihasilkan dari entitas *tb\_stopword* yang memiliki atribut: *id\_word* *(primary key)* yang bersifat unik, *word* digunakan untuk menampung kata.

* + 1. Tabel Vektor

Tabel *vektor* merupakan tabel yang dihasilkan dari entitas vektor yang memiliki atribut: *id* *(primary key)* yang bersifat unik, *idSolusi* digunakan untuk menampung id solusi, *vektorSolusi* digunakan untuk menampung nilai vektor solusi.

* + 1. Tabel *Term*

Tabel *term* merupakan tabel yang dihasilkan dari entitas *term* yang memiliki atribut: *id* *(primary key)* yang bersifat unik, *term* digunakan sebagai variabel tempat menampung term, *idf* digunakan sebagai tempat menampung variabel nilai vektor solusi.

## Implementasi Sistem

Tahap implementasi sitem merupakan lanjutan dari tahap sebelumnya, dimana pada tahap ini akan dijabarkan mengenai implementasi Sistem Pengaduan Kerusakan Komputer Dan Rekomendasi Solusi Menggunakan Algoritma *Cosine Similarity*. Implementasi sistem dilakukan dengan menggunakan *framework* Codeigniter (CI). Struktur dari *framework* Codeigniter memiliki arsitektur MVC *(Model, View, Controller)* yang berguna agar pemrograman lebih terstruktur dan mudah dipelajari.

### Implementasi Proses Preprocessing Pada Sistem

Implementasi proses *preprocessing* terdiri dari *case folding, tokenizing, filtering,* dan *stemming. Preproccessing* dilakukan ketika pegawai menginputkan data pengaduan *(query)* dan *technical support* menginputkan data baru saat tambah atau edit data solusi.Untuk implementasi proses *preprocessing* pada sistem diimplementasikan seperti pada Tabel 4.1.

Tabel 4. 1 Kode *Preprocessing* Teks

|  |  |
| --- | --- |
| Baris | Penggalan Code |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | //menghilangkan tanda baca  $namakerusakan = preg\_replace("/[[:punct:]]+/", " ", $query);  //menghubah menjadi huruf kecil  $namakerusakan = strtolower($namakerusakan);  //menjadikan array  $namakerusakan = explode(" ", $namakerusakan);    //pencocokan kata atau stopwords  $fileStopword= base\_url().'assets/preprocessing/stopword.txt';  $stopwords = file($fileStopword, FILE\_IGNORE\_NEW\_LINES);  $stopword = array\_values(array\_diff($gejala, $stopwords));    //hitung tf  $tf = array\_values(array\_count\_values($stopword));  //menghilangkan kata yg double  $doubleTermRemove = array\_values(array\_unique($stopword));  //Stemming  $jumlahDoubleTermRemove = count($doubleTermRemove);  $i=0;  while ($i<$jumlahDoubleTermRemove) {  $this->load->library('stemmingg');  $textAwal = $doubleTermRemove[$i];  $stemming = $this->stemmingg->stemming($textAwal);  if ($stemming=='') {  $term[$i] = $ doubleTermRemove[$i];  }else{  $term[$i] = $stemming;  }  $i++;  } |

Berdasarkan Tabel 4.1 proses pertama pada *preprocessing* adalah *case folding* yaitu menghilangkan tanda baca dan spasi seperti pada line 2 dan mengubah huruf kapital menjadi huruf kecil *(lowercase)* seperti pada line 5. Kemudian memecah teks kedalam bentuk kata *(term),* proses ini disebut dengan *tokenizing*, seperti pada line 8. Setelah proses *tokenizing* selesai dilanjutkan dengan menyimpan semua kata *(term)* kedalam bentuk *array,* kemudian dilakukan proses *filtering* membandingkan daftar kata yang ada di *stopword* dengan *term* dari hasil tokenisasi seperti pada line 13. Hasil *filtering* disimpan kedalam bentuk *array.*

### Implementasi Proses Indexing Pada Sistem

Implementasi proses *indexing* merupakan lanjutan dari tahapan *preprocessing.* Untuk implementasi proses *indexing* pada sistem diimplementasikan seperti pada Tabel 4.2.

Tabel 4. 2 Kode *Indexing*

|  |  |
| --- | --- |
| Baris | Penggalan Code |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | public function indexing($term, $tf, $jumlahDoubleTermRemove, $idSolusi) {  for ($i=0; $i<$jumlahDoubleTermRemove; $i++) {  $termMerge = $this->ModelTechnicalSupport->select\_AlltermIndex($idSolusi);  if (in\_array($term[$i], $termMerge)) {  $this->ModelTechnicalSupport->countTFIndex($idSolusi,$term[$i],$tf[$i]);  } else {  $dataTerm = array(  'idSolusi'=>$idSolusi,  'term' => $term[$i],  'tf' => $tf[$i],  );  $this->db->insert('indexing', $dataTerm);    }  }  } |

Pada proses *indexing, term* dan tf *(term frequency)* akan disimpan ke dalam tabel *indexing* yang ada pada database berdasarkan idSolusi. Pertama yaitu melakukan perulangan dengan menampung idSolusi pada variable termMerge seperti pada line 4. Kemudian mengecek kertersediaan *term* pada *database,* bila *term* sudah ada pada *database* maka jumlah tf *(term frequency)* akan bertambah dan bila tidak ada maka *term* akan disimpan ke dalam *database.*

### Implementasi Proses Perhitungan TF-IDF dan Bobot *Query* Pada Sistem

Implementasi proses perhitungan TF-IDF dan bobot merupakan lanjutan dari tahapan *indexing.* Untuk implementasi proses perhitungan TF-IDF pada sistem diimplementasikan seperti pada Tabel 4.3.

Tabel 4. 3 Kode Proses Perhitungan TF-IDF dan Bobot

|  |  |
| --- | --- |
| Baris | Penggalan Code |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | $x=0;  $bobotKuadratQuery=0;  while ( $x< $jumlahDoubleTermRemove) {  $df= $this->ModelTechnicalSupport->select\_df(); //cari df  $tf= $this->ModelTechnicalSupport->select\_tf($term[$x]); //cari tf  if ($tf==0) { //jika kata yang dicari tidak ada di dokumen  $x++  }else{  $idf=log10(($df+1)/($tf+1)); //hitung idf  $bobotQ[$term[$x]]=$tf\*$idf; //hitung bobot query  //rumus cosin bagian penyebut query  $bobotKuadratQuery=$bobotKuadratQuery + pow($bobotQ[$term[$x]], 2); //dikuadrat 2  $bobotQuery[$x]=array( $term[$x]=>$bobotQ[$term[$x]], ); // untuk mengetahui nilai bobot pada setiap term, digunakan saat menghitung cosine pd bagian penyebut  $termTerseleksi[$x]=$term[$x];  $x++;  }  } |

Pada proses perhitungan TF-IDF dan bobot langkah pertama yaitu, mencari jumlah df *(document frequency)* dan tf *(term frequency)* seperti pada line 5 dan 6. Kemudian menghitung IDF dengan cara log dari jumlah dokumen+1 (termasuk *qery*)dibagi dengan jumlah tf+1 (termasuk *qery*) seperti pada line 12. Setelah itu menghitung nilai bobot *query* dengan carat tf dikali IDF.

Perhitungan bagian perhitungan pembilang *(qery)* yang adapada rumus *cosine similarity* diselipkan gunanya untuk mengefisienkan kode. Proses perhitungannya adalah jumlah dari semua hasil kuadrat bobot *query* seperti pada line 15. Kemudian disimpan ke dalam bentuk *array* agar dapat dipanggil ketika perhitugan *cosine similarity.*

### Implementasi Pencarian Solusi Kerusakan Menggunakan Algoritma *Cosine Similarity*

Implementasi pencarian solusi kerusakan menggunakan algoritma *cosine similarity* merupakan lanjutan dari tahapan perhitungan TF-IDF dan bobot*.* Untuk implementasi pencarian solusi kerusakan menggunakan algoritma *cosine similarity* pada sistem diimplementasikan seperti pada Tabel 4.4.

Tabel 4. 4 Proses Perhitungan Algoritma *Cosine Similarity*

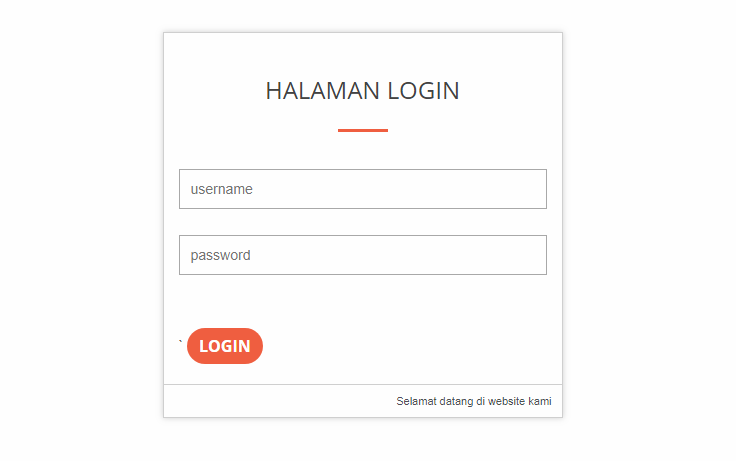
|  |  |
| --- | --- |
| Baris | Penggalan Code |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | //mencari solusi kerusan  $id=$this->ModelTechnicalSupport->select\_id(); // mencari id  $y=0;  foreach ($id->result() as $id) {  $idSolusi[$y]=$id->idSolusi; // id ubah ke array  $term=$this->ModelTechnicalSupport->selecttermIndex($idSolusi[$y]);// mencari term berdasarkan id dokumen  $z=0;  $bobotKuadrat=0;  $bobotKuadratSolusi=0;  $atas=0;    foreach ($term->result() as $term) {  if (in\_array($term->term, $termTerseleksi)) { //menghitung pembilang, kata x yang ada query dan dokumen  $df= $this->ModelTechnicalSupport->select\_df();  $tf= $this->ModelTechnicalSupport->select\_tf($term->term);  $idfSolusi=log10(($df+1)/($tf+1));  $bobot=$term->tf\* $idfSolusi;    $bobotKuadrat=$bobotKuadrat + pow($bobot, 2); //bobot pembilang  $atas= $atas+ $bobot\* $bobotQ[$term->term];  $z++;  }  else{  //menghitung penyebut solusi  $df= $this->ModelTechnicalSupport->select\_df();  $tf= $this->ModelTechnicalSupport->select\_tf($term->term);  $idfSolusi=log10(($df+1)/($tf));  $bobot=$term->tf\*$idfSolusi;  //rumus cosin penyebut  //bobotKuadratSolusi = perhitungan bobot solusi yg ada di penyebut, dan hanya bobot kata x yg belum terhitung dibagian pembilang. agar lebih efisien  $bobotKuadratSolusi=$bobotKuadratSolusi + pow($bobot, 2);  }  }  if ($z==0) { //perhitungan jika semua kata tidak ada yg sama di query dan dokumen  $y++;  } else{  //penyebutSolusi  $kuadratSolusi=$bobotKuadrat+$bobotKuadratSolusi;  $jumDocKaliQuery[$idSolusi[$y]]=$atas;  $jum\_akarkuadrat[$idSolusi[$y]]=sqrt($kuadratSolusi);  $jum\_akarkuadratQuery=sqrt($bobotKuadratQuery);  //unset($QuerykaliTerm);  unset($bobotKuadrat);  unset($bobotKuadratSolusi);  $hasil[$idSolusi[$y]]=$jumDocKaliQuery[$idSolusi[$y]]/($jum\_akarkuadrat[$idSolusi[$y]]\*$jum\_akarkuadratQuery);  $y++;  $max =max($hasil); //cari hasil cosine tertinggi  $key = array\_keys($hasil, $max);  $keys =$key[0];  }  } |

Pada proses pencarian solusi kerusakan menggunakan algoritma *cosine similarity* langkah pertama yaitu, mencari id solusi dan ubah ke dalam bentuk *array*. Kemudian mencari *term* berdasarkan id dokumen. Sebelum mencari bobot solusi yang ada pada bagian pembilang rumus *cosine similarity*, hal yang harus dilakukan adalah menghitung idfSolusi dengan cara log dari jumlah dokumen+1 (termasuk qery) dibagi dengan jumlah tf+1 (termasuk qery). Kemudian menghitung bobot solusi dengan cara tf dikali IDF, setelah mendapatkan hasil bobot solusi dikuadratkan. Langkah selanjutnya untuk mencari bobot yang ada pada bagian pembilang rumus *cosine similarity* adalah menjumlahkan bobot *query* dengan bobot solusi. Untuk menghitung bagian peyebut (solusi) yang ada dalam rumus *cosine similarity* adalah dengan menghitung bobot solusi kemudian dikuadratkan.

Kondisi jika tidak ada kata yang sama di *query* dan dokumen (solusi) maka kata btersebut diberi nilai 0. Jika ada kata yang sama maka dilakukan proses menghitung bagian penyebut (solusi) yang ada di dalam rumus *cosine similarity*. Setelah mendapatkan hasil bagian penyebut (solusi) maka disimpan kedalam bentuk *array*, lalu diakarkan. Hasil bagian penyebut (*query*) juga diakarkan. Setelah itu untuk mendapatkan solusi kerusakan menggunakan algoritma cosine similarity dilakukan perhitungan menggunakan sesuai rumus. Hasil yang ditampilkan pada sistem adalah berdasarkan nilai cosim tertinggi.

## Antar Muka Sistem

Implementasi antar muka sistem Pengaduan Kerusakan Komputer dan Rekomendasi Solusi Menggunakan Algoritma *Cosine Similarity* menggunakan *framework Codeigniter* dengan bahasa pemrograman PHP dan untuk tampilan sistem mengunakan HTML, CSS, dan Java Script.



Gambar 4. 2 Tampilan Antar Muka Halaman *Login*

Gambar 4.2 merupakan tampilan dari halaman login pada sistem pengaduan untuk semua *user*. Pada halaman *login*, *user* harus menginputkan *username* dan *password* ketika ingin masuk ke sistem pengaduan.



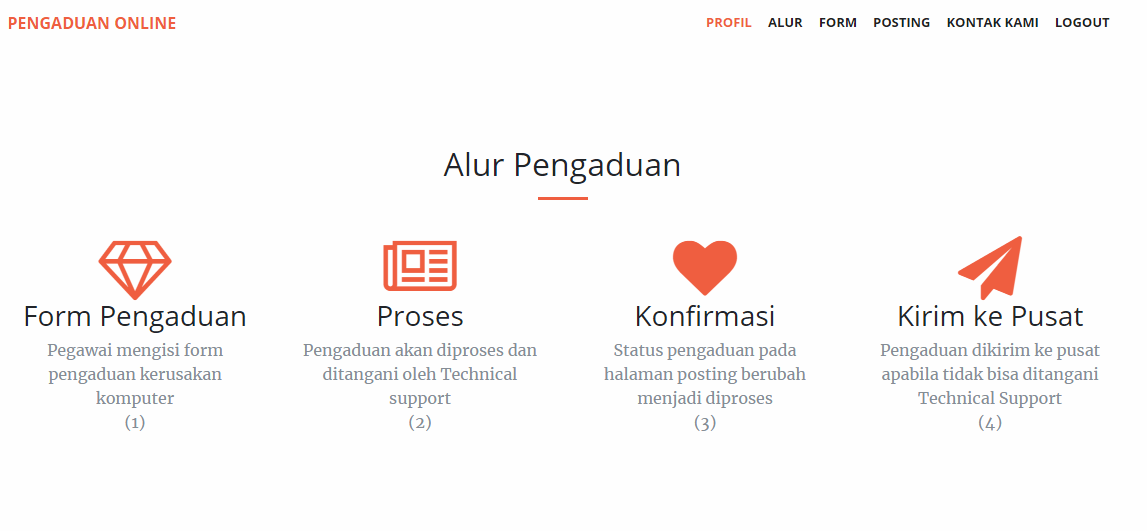
Gambar 4. 3 Tampilan Antar Muka Halaman Utama Pegawai

Gambar 4.3 merupakan tampilan dari halaman utama sistem pengaduan untuk pegawai. Pada menu navigasi sebelah kiri berisi menu profil, alur, form, kontak kami, dan logout. Pada menu navigasi sebelah kanan berisi menu pengaduan online yang ketika diklik oleh pegawai, maka halaman akan menuju bagian header. Pada bagian *header* berisi judul, penjelasan mengenai gambaran umum sistem, dan tombol “*Find Out More*” yang ketika diklik oleh pegawai, maka akan menuju halaman bagian *profile* seperti pada Gambar 4.4



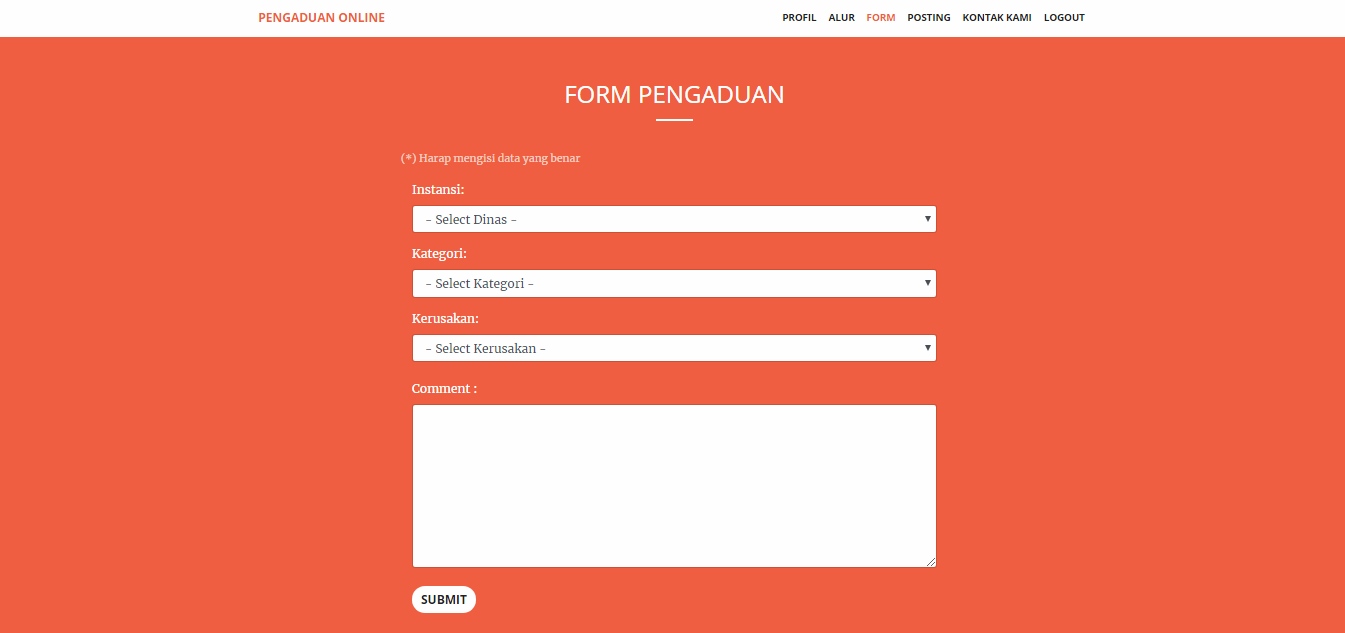
Gambar 4. 4 Tampilan Antar Muka Halaman Profil

Gambar 4.4 merupakan tampilan dari halaman profil pada sistem pengaduan untuk pegawai. Pada halaman profil terdapat tulisan “Selamat datang!” dan penjelasan lebih lengkap mengenai sistem pengaduan.



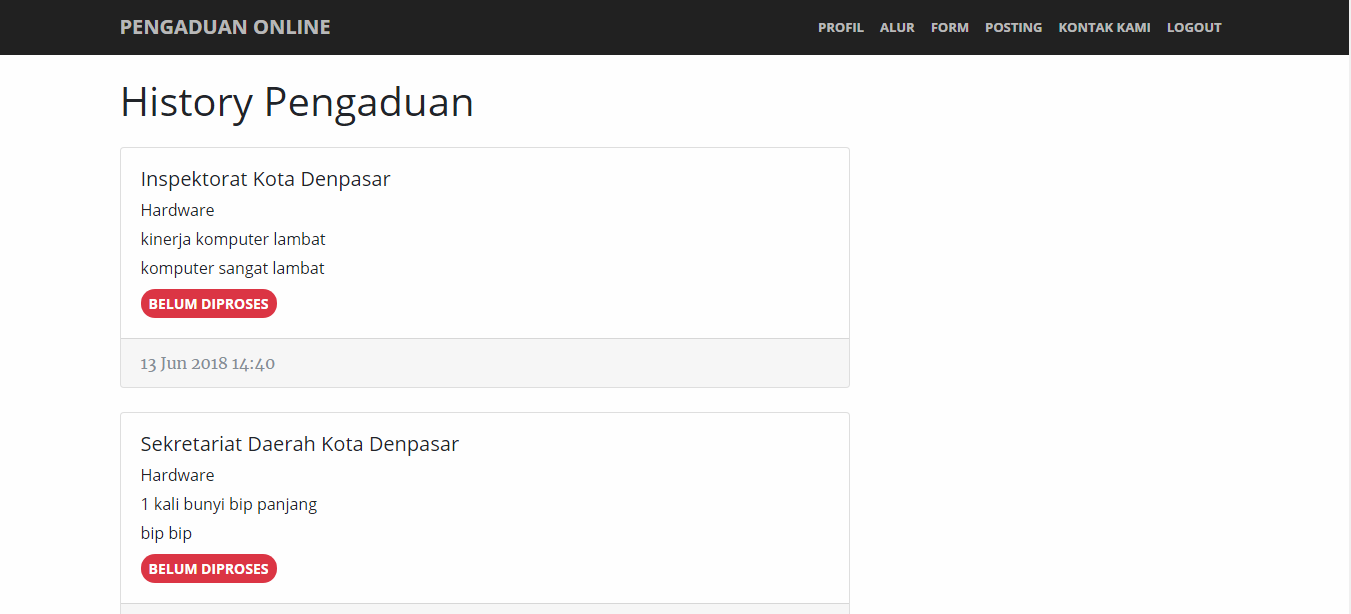
Gambar 4. 5 Tampilan Antar Muka Halaman Alur

Gambar 4.5 merupakan tampilan dari halaman alur pada sistem pengaduan untuk pegawai. Pada halaman alur berisi tahapan dari pengaduan yang diadu oleh pegawai. Dari tahap pertama yaitu mengisi form pengaduan, hingga tahap mengirim pengaduan ke pusat.



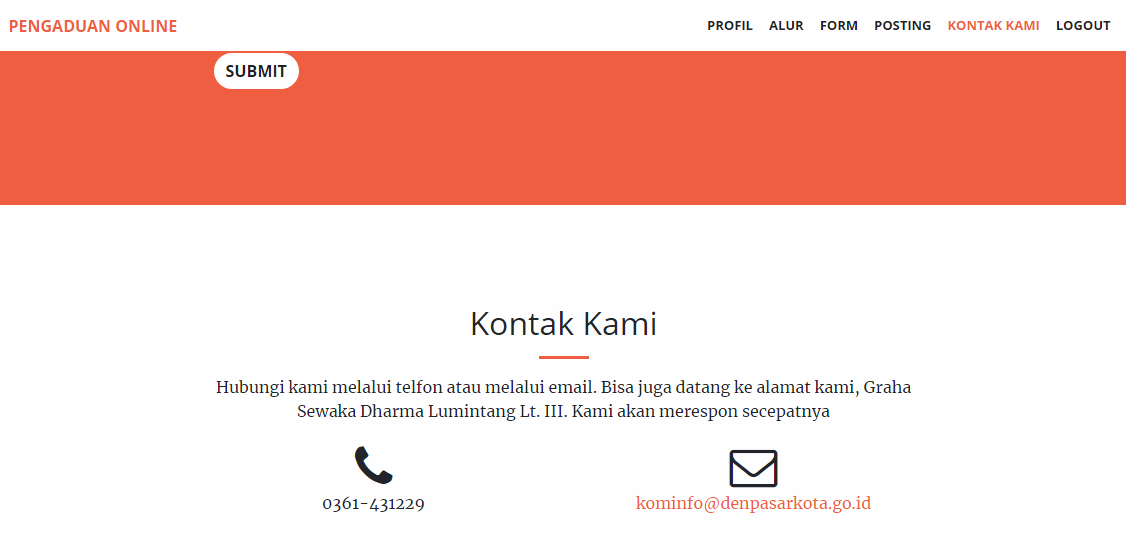
Gambar 4. 6 Tampilan Antar Muka Halaman Form

Gambar 4.6 merupakan tampilan dari halaman form pada sistem pengaduan untuk pegawai. Pada halaman form, ketika pegawai membuat pengaduan maka pegawai dapat memilih instansi yang mereka tempati, memilih kategori kerusakan *(hardware, software,* dan jaringan), memilih kerusakan komputer, dan menginputkan keterangan pada bagian *comment.* Setelah selesai mengisi form pegawai dapat menekan tombol submit untuk menyimpan data yang mereka inputkan, lalu halaman akan dialihkan ke halaman posting seperti Gambar 4.7



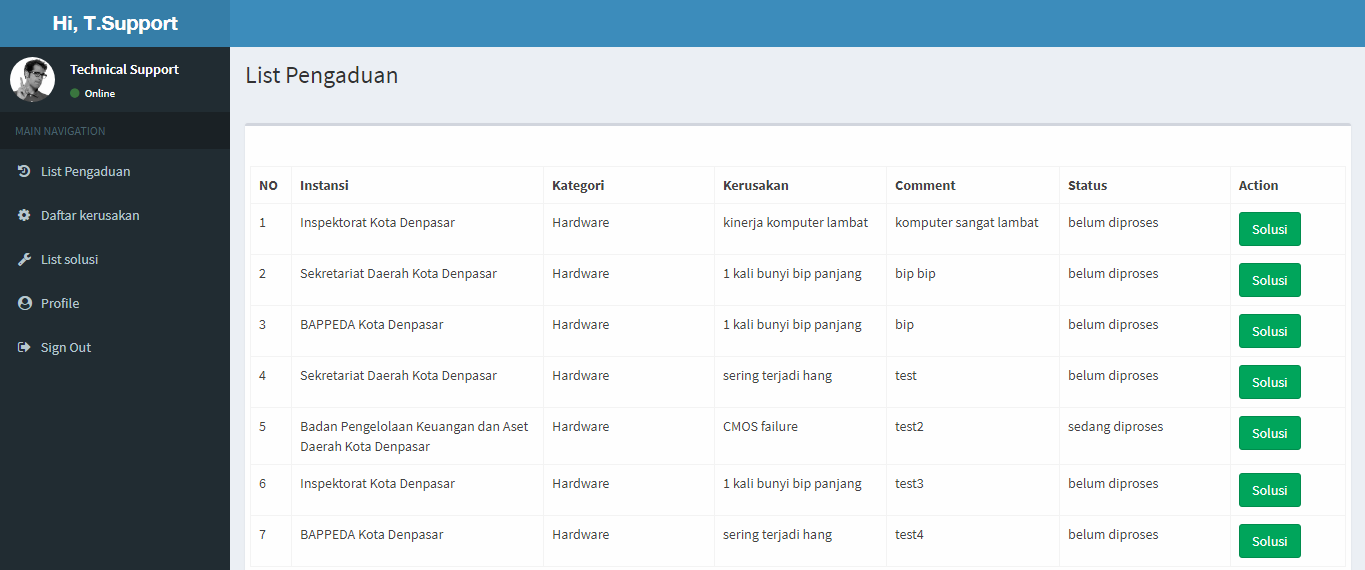
Gambar 4. 7 Tampilan Antar Muka Halaman Posting

Gambar 4.7 merupakan tampilan dari halaman posting pada sistem pengaduan untuk pegawai. Pada halaman posting berisi *history* dari pengaduan yang pernah diadu, berisi waktu ketika pengaduan dilakukan, dan juga berisi keterangan apakah pengaduan belum atau sudah diproses.



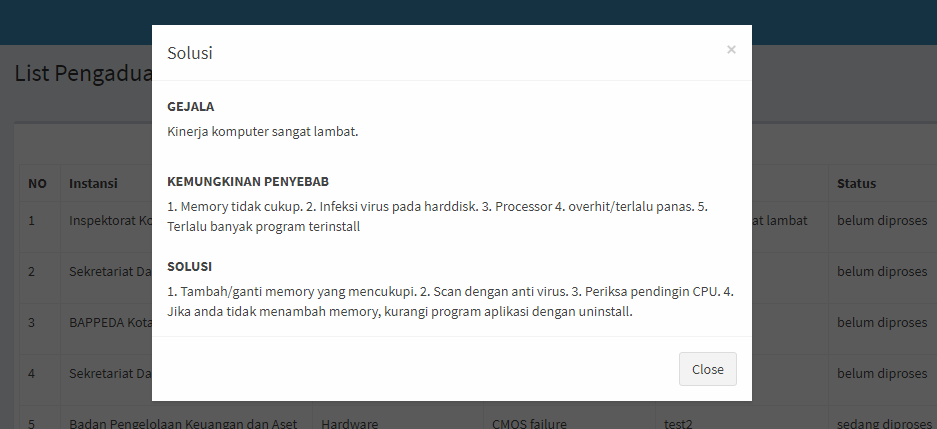
Gambar 4. 8 Tampilan Antar Muka Halaman Kontak Kami

Gambar 4.8 merupakan tampilan dari halaman kontak kami pada sistem pengaduan untuk pegawai. Pada halaman kontak kami berisi alamat, nomor telepon dan email yang dapat dihubungi.



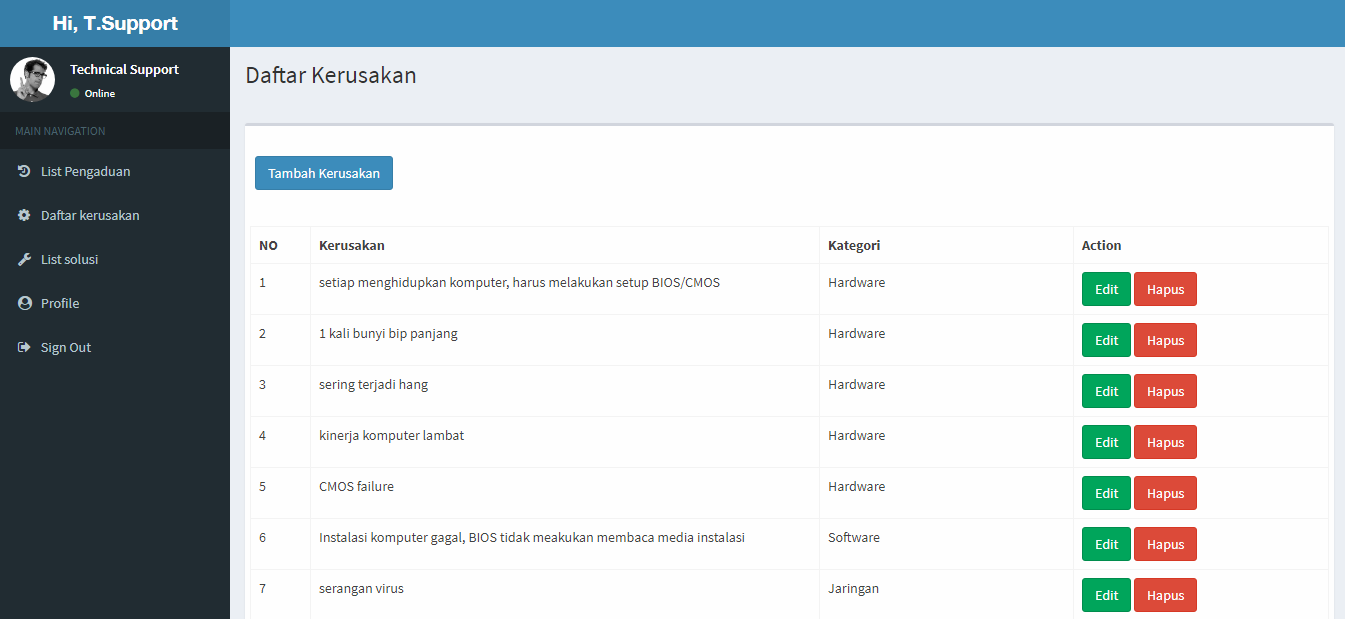
Gambar 4. 9 Tampilan Antar Muka Halaman List Pengaduan *Technical Support*

Gambar 4.9 merupakan tampilan dari halaman list pengaduan pada sistem untuk *technical support*. Pada bagian navigasi terdapat menu list pengaduan, daftar kerusakan, list solusi, *profile,* dan *sign out.* Pada halaman ini juga terdapat tabel list pengaduan yang berisi informasi mengenai data pengaduan seperti instansi, kategori, kerusakan, comment, status, dan *action.* Ketika *technical support* ingin mendapatkan solusi dari sistem mengenai kerusakan yang telah diadu oleh pegawai maka *technical support* dapat menekan tombol solusi yang terdapat pada tabel list pengaduan. Kemudian sistem akan menampilkan solusi dari kerusakan komputer yang telah diadu seperti pada Gambar 4.10



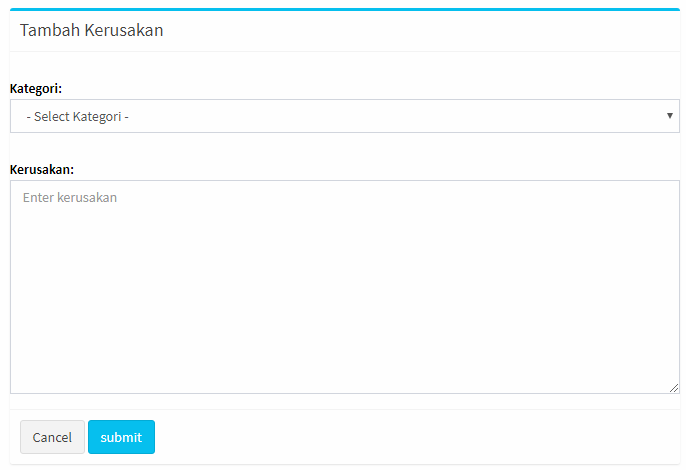
Gambar 4. 10 Tampilan Antar Muka Halaman Solusi

Gambar 4.10 merupakan tampilan dari halaman solusi pada sistem untuk *technical support*. Pada halaman solusi berisi informasi mengenai gejala, kemungkinan penyebab, dan solusi dari kerusakan komputer yang telah diadu oleh pegawai. Kemudian juga terdapat tombol *close* yang berguna ketika *technical support* ingin kembali ke halaman solusi.



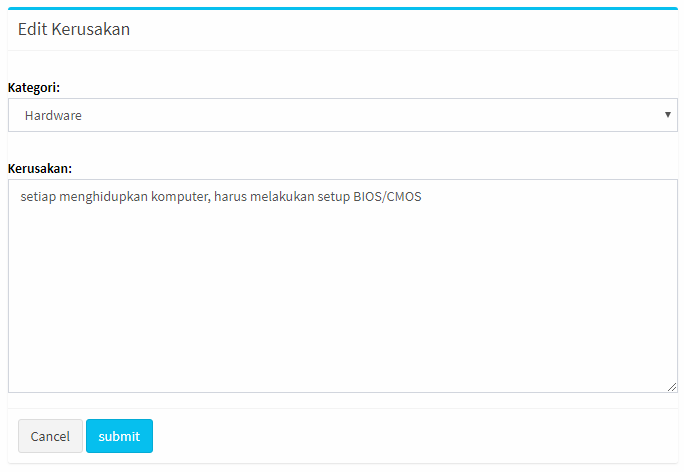
Gambar 4. 11 Tampilan Antar Muka Halaman Daftar Kerusakan

Gambar 4.11 merupakan tampilan dari halaman daftar kerusakan pada sistem untuk *technical support*. Pada halaman daftar kerusakan terdapat tabel daftar kerusakan yang berisi informasi mengenai data kerusakan seperti nama kerusakan, kategori, dan *action.* Ketika *technical support* ingin menambah kerusakan maka *technical support* dapat menekan tombol tambah kerusakan yang terdapat diatas tabel daftar kerusakan. Kemudian sistem akan menampilkan halaman tambah kerusakan seperti pada Gambar 4.12. Ketika *technical support* ingin mengedit kerusakan maka *technical support* dapat menekan tombol edit kerusakan yang terdapat pada tabel daftar kerusakan. Kemudian sistem akan menampilkan halaman edit kerusakan seperti pada Gambar 4.13. Ketika *technical support* ingin menghapus kerusakan maka *technical support* dapat menekan tombol hapus kerusakan yang terdapat pada tabel daftar kerusakan. Kemudian sistem akan menghapus data kerusakan yang ada pada sistem.



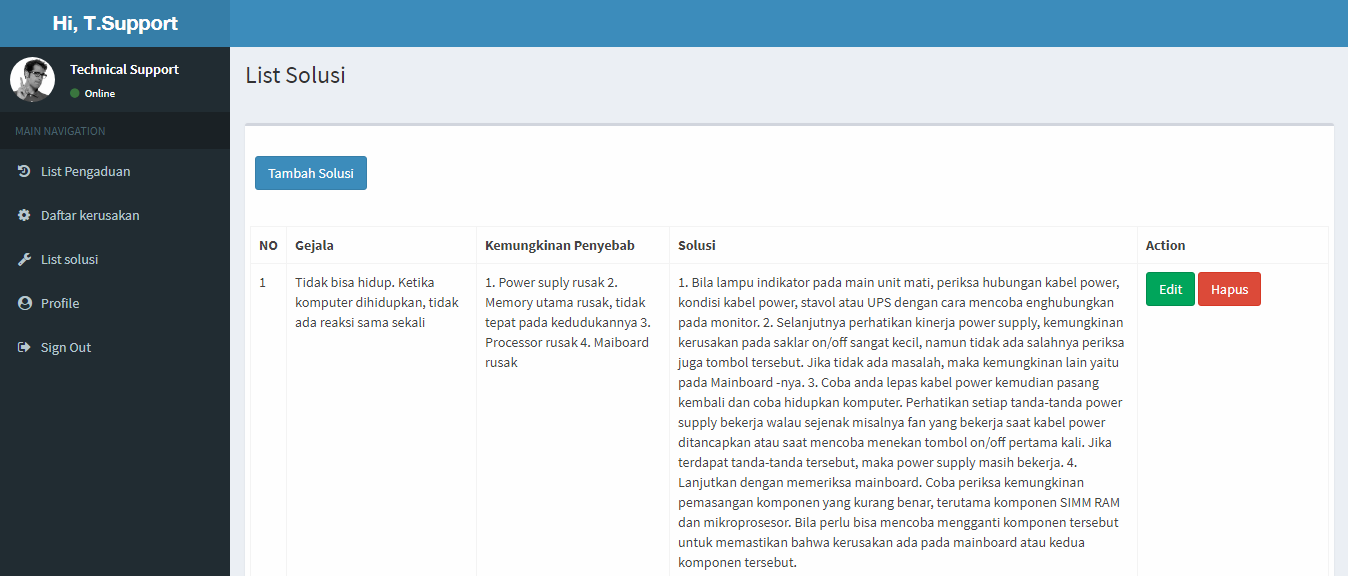
Gambar 4. 12 Tampilan Antar Muka Halaman Tambah Kerusakan

Gambar 4.12 merupakan tampilan dari halaman tambah kerusakan pada sistem untuk *technical support*. Ketika *technical support* ingin menambah kerusakan maka *technical support* memilih jenis kategori dan menginputkan nama kerusakan. Menu kategori ketika dipilih akan muncul pilihan *hardware*, *software,* dan jaringan. Setelah *technical support* selesai menginputkan data maka *technical support* akan mengklik tombol submit untuk menyimpan data.



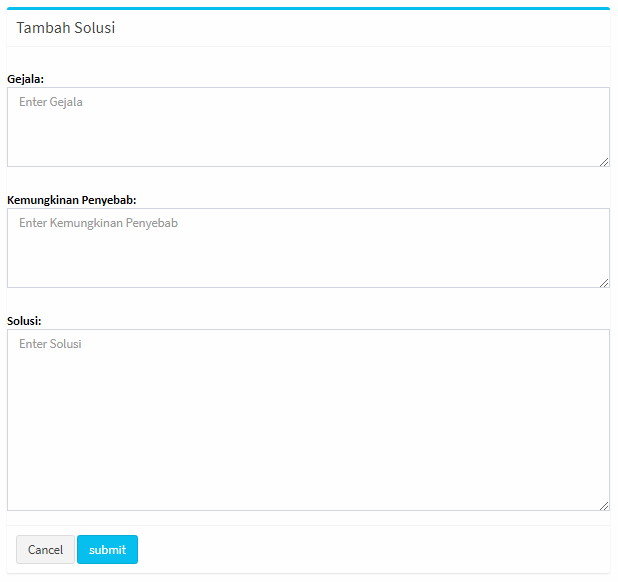
Gambar 4. 13 Tampilan Antar Muka Halaman Edit Kerusakan

Gambar 4.13 merupakan tampilan dari halaman edit kerusakan pada sistem pengaduan untuk *technical support*. Pada halaman edit kerusakan *technical support* dapat mengubah kategori dan kerusakan apabila terdapat kesalahan data.



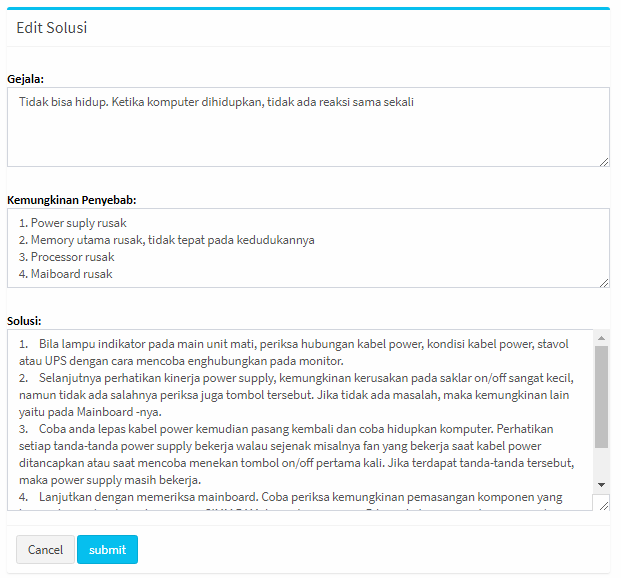
Gambar 4. 14 Tampilan Antar Muka Halaman List Solusi

Gambar 4.14 merupakan tampilan dari halaman list solusi pada sistem pengaduan untuk *technical support*. Pada halaman list solusi terdapat tabel list solusi yang berisi informasi mengenai data solusi seperti gejala, kemungkinan penyabab, solusi , dan *action.* Ketika *technical support* ingin menambah list solusi maka *technical support* dapat menekan tombol tambah solusi yang terdapat diatas tabel list solusi. Kemudian sistem akan menampilkan halaman tambah solusi seperti pada Gambar 4.15. Ketika *technical support* ingin mengedit solusi maka *technical support* dapat menekan tombol edit yang terdapat pada tabel daftar kerusakan. Kemudian sistem akan menampilkan halaman edit solusi seperti pada Gambar 4.16. Ketika *technical support* ingin menghapus solusi maka *technical support* dapat menekan tombol hapus yang terdapat pada tabel list solusi. Kemudian sistem akan menghapus data solusi yang ada pada sistem.



Gambar 4. 15 Tampilan Antar Muka Halaman Tambah Solusi

Gambar 4.15 merupakan tampilan dari halaman tambah solusi pada sistem untuk *technical support*. Ketika *technical support* ingin menambah solusi maka *technical support* mengisi *form* dengan menginputkan gejala, kemungkinan penyebab, dan solusi. Setelah *technical support* selesai menginputkan data solusi maka *technical support* akan mengklik tombol submit untuk menyimpan data.



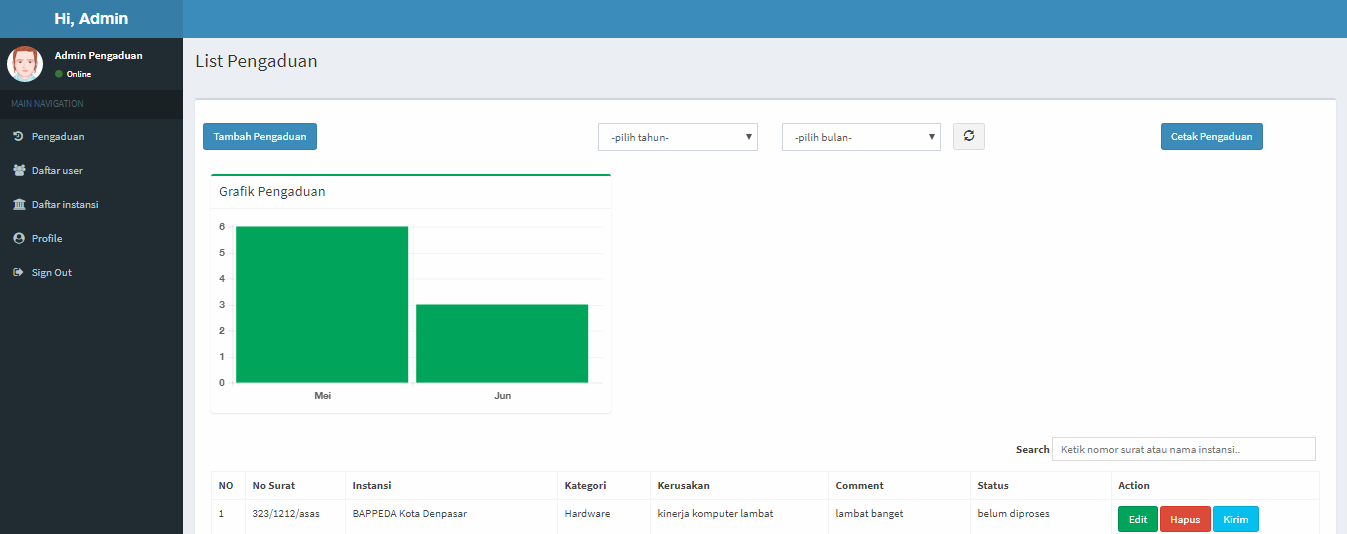
Gambar 4. 16 Tampilan Antar Muka Halaman Edit Solusi

Gambar 4.16 merupakan tampilan dari halaman edit solusi pada sistem pengaduan untuk *technical support*. Pada halaman edit solusi, *technical support* dapat mengubah gejala, kemungkinan penyebab, dan solusi pada *form* apabila terdapat kesalahan data.

-----------------------------------------BELUM ISI SS---------------------------------------

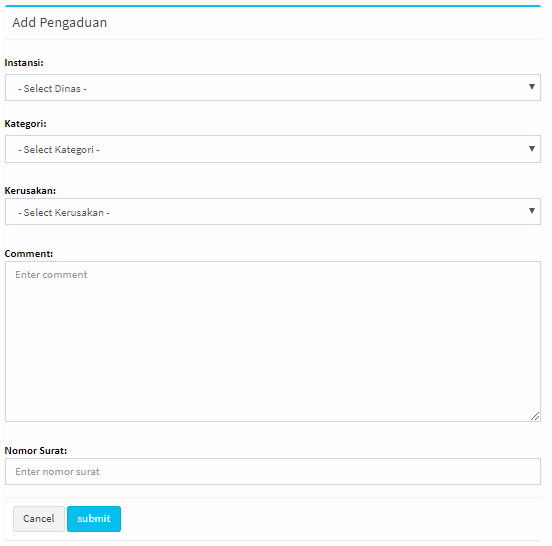
Gambar 4. 17 Tampilan Antar Muka Halaman Profile

Gambar 4.17 merupakan tampilan dari halaman profile pada sistem pengaduan untuk *technical support*. Pada halaman profile *technical support* dapat mengubah nama, email, dan foto profile sesuai keinginan *technical support.*



Gambar 4. 18 Tampilan Antar Muka Halaman Utama Admin

Gambar 4.18 merupakan tampilan dari halaman utama pada sistem pengaduan untuk admin. Pada halaman admin terdapat navigasi disebelah kiri yang berisi menu pengaduan, daftar user, daftar instansi, profile, dan *signout.* Pada halaman ini juga terdapat tombol tambah pengaduan yang berguna apabila admin ingin menambah pengaduan. Halaman tambah pengaduan dapat dilihat pada Gambar 4.19. Kemudian terdapat tombol cetak pengaduan yang berguna apabila admin ingin mencetak pengaduan. Menu pilih bulan dan pilih tahun yang berguna apabila admin ingin melihat data pengaduan berdasarkan memilih bulan dan tahun yang diinginkan. Menu *search* yang berguna apabila admin ingin mencari data pengaduan berdasarkan nomor surat atau nama instansi. Tabel pengaduan berisi informasi mengenai data pengaduan seperti nomor surat, instansi, kategori, kerusakan, comment, status, dan *action.* Ketika adminingin mengedit pengaduan maka admindapat menekan tombol edit yang terdapat pada tabel pengaduan. Kemudian sistem akan menampilkan halaman edit pengaduan seperti pada Gambar 4.20. Ketika admin ingin menghapus pengaduan maka admindapat menekan tombol hapus yang terdapat pada tabel pengaduan. Kemudian sistem akan menghapus data pengaduan yang ada pada sistem. Ketika admin ingin mengirim pengaduan, maka admin dapat klik tombol kirim pada tabel pengaduan. Proses kirim pengaduan dilakukan menggunakan fitur fpdf dan diterima oleh penerima melalui *email*.



Gambar 4. 19 Tampilan Antar Muka Halaman Tambah Pengaduan

Gambar 4.19 merupakan tampilan dari halaman tambah pengaduan pada sistem pengaduan untuk admin. Ketika adminingin menambah pengaduan maka admin mengisi *form* dengan menginputkan instansi, kategori, kerusakan, *comment,* dan nomor surat. Setelah adminselesai menginputkan data pengaduan maka adminklik tombol submit untuk menyimpan data pengaduan.

-----------------------------------------BELUM ISI SS---------------------------------------

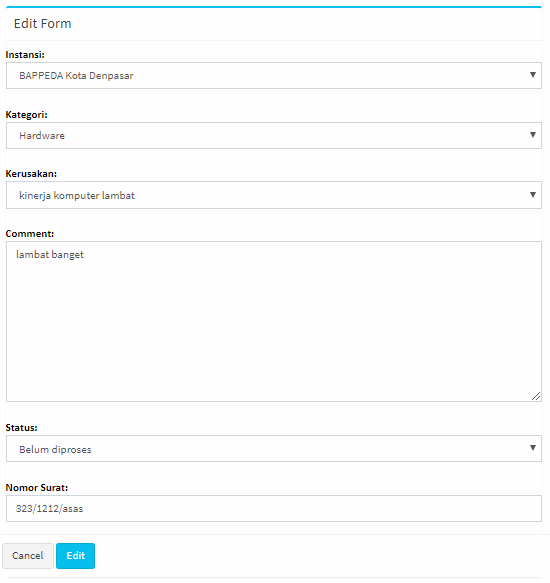
Gambar 4. 20 Tampilan Antar Muka Halaman Cetak Pengaduan

Gambar 4.20 merupakan tampilan dari halaman cetak pengaduan pada sistem pengaduan untuk admin.

-----------------------------------------BELUM ISI SS---------------------------------------

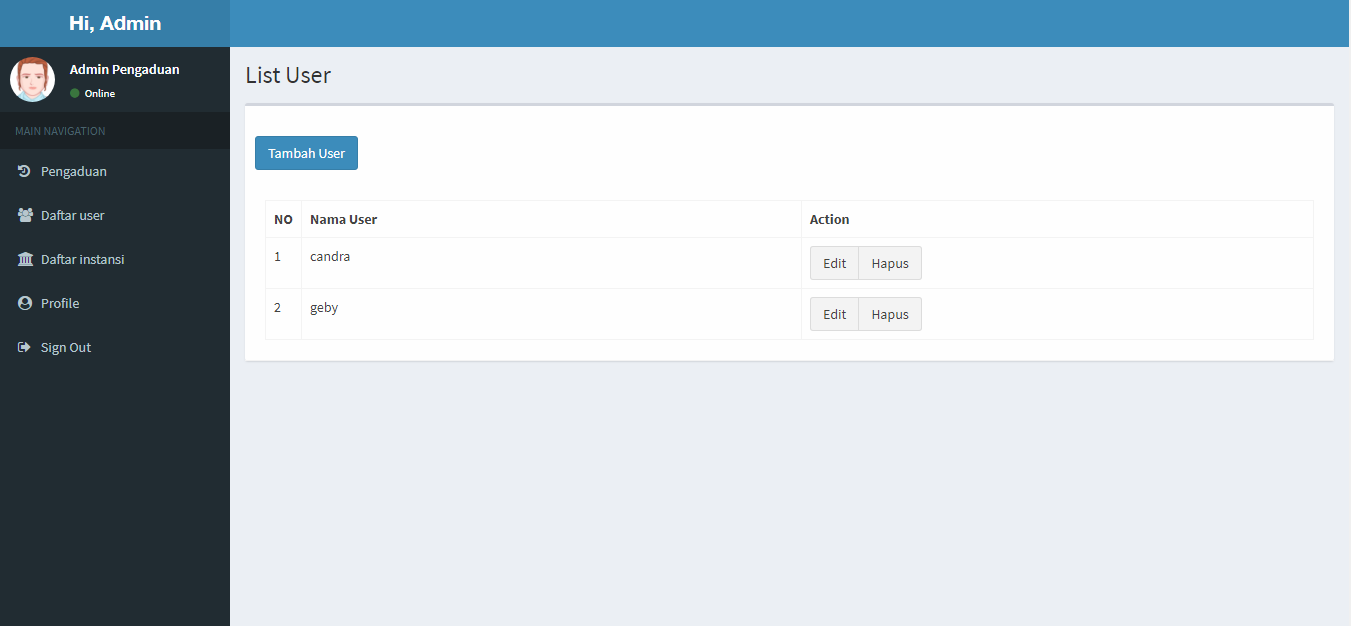
Gambar 4. 21 Tampilan Antar Muka *Search* Pengaduan

Gambar 4.21 merupakan tampilan dari *search* pengaduan pada sistem pengaduan untuk admin.



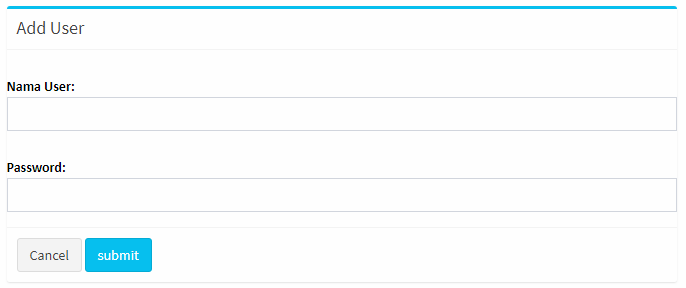
Gambar 4. 22 Tampilan Antar Muka Halaman Edit Pengaduan

Gambar 4.22 merupakan tampilan dari halaman edit pengaduan pada sistem pengaduan untuk admin. Pada halaman edit pengaduan, admindapat mengubah instansi, kategori, kerusakan, *comment,* dan nomor surat pada *form* apabila terdapat kesalahan data.



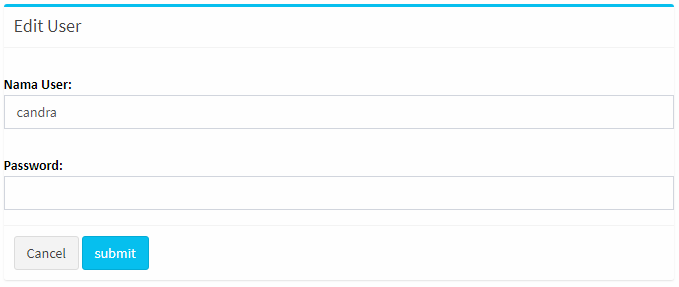
Gambar 4. 23 Tampilan Antar Muka Halaman List User

Gambar 4.23 merupakan tampilan dari halaman list user pada sistem pengaduan untuk admin. Pada halaman list user terdapat navigasi disebelah kiri yang berisi menu pengaduan, daftar user, daftar instansi, profile, dan *signout.* Ketika adminingin menambah list user maka admindapat menekan tombol tambah user yang terdapat diatas tabel list user. Kemudian sistem akan menampilkan halaman tambah user seperti pada Gambar 4.24. Ketika adminingin mengedit user maka admindapat menekan tombol edit yang terdapat pada tabel list user. Kemudian sistem akan menampilkan halaman edit user seperti pada Gambar 4.25. Ketika adminingin menghapus user maka admindapat menekan tombol hapus yang terdapat pada tabel list user. Kemudian sistem akan menghapus data user yang ada pada sistem.



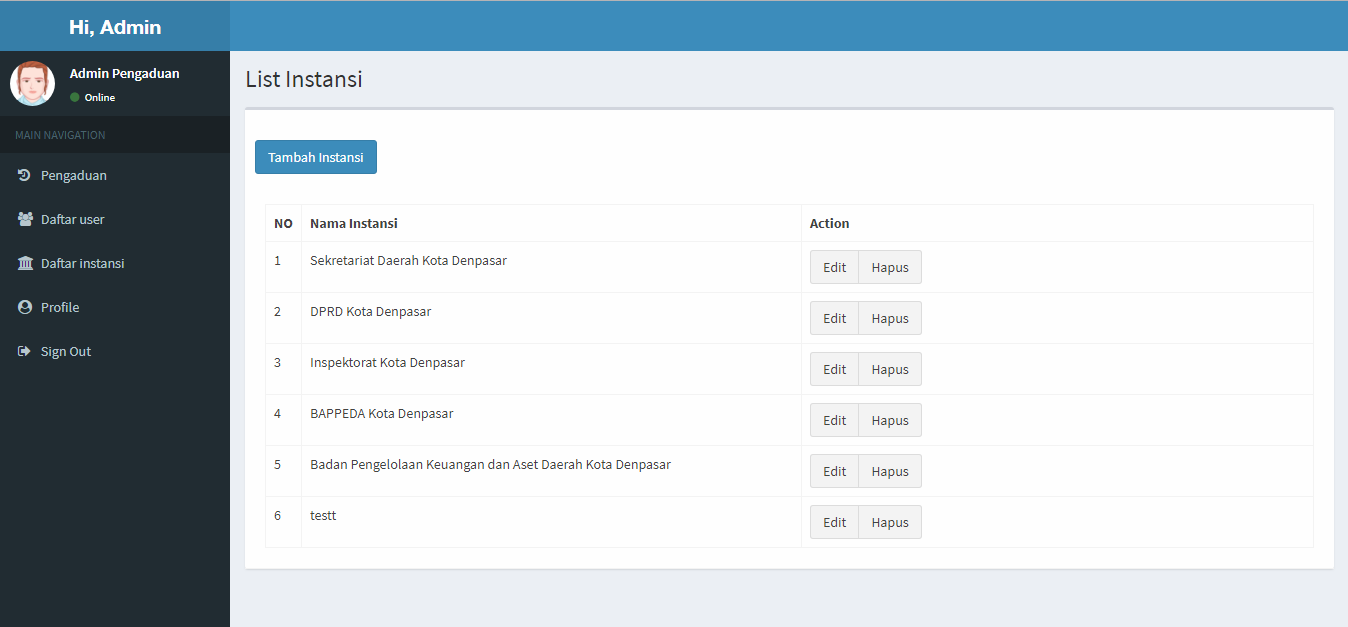
Gambar 4. 24 Tampilan Antar Muka Halaman Tambah User

Gambar 4.24 merupakan tampilan dari halaman tambah user pada sistem pengaduan untuk admin. Ketika adminingin menambah list user maka admin mengisi *form* dengan menginputkan *username* dan *password.* Setelah adminselesai menginputkan data user maka adminklik tombol submit untuk menyimpan data user.



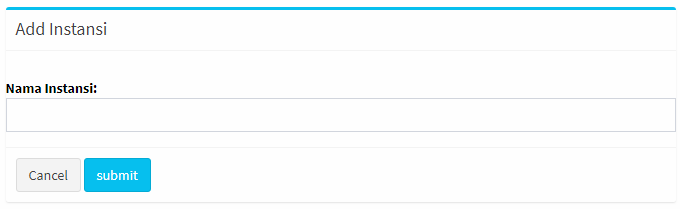
Gambar 4. 25 Tampilan Antar Muka Halaman Edit User

Gambar 4.25 merupakan tampilan dari halaman edit user pada sistem pengaduan untuk admin. Pada halaman edit user, admindapat mengubah *username* dan *password* pada *form* apabila terdapat kesalahan data.



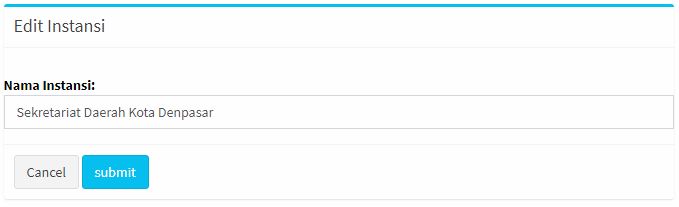
Gambar 4. 26 Tampilan Antar Muka Halaman List Instansi

Gambar 4.26 merupakan tampilan dari halaman list instansi pada sistem pengaduan untuk admin. Pada halaman list instansi terdapat navigasi disebelah kiri yang berisi menu pengaduan, daftar user, daftar instansi, profile, dan *signout.* Ketika adminingin menambah instansi maka admindapat menekan tombol tambah instansi yang terdapat diatas tabel list instansi. Kemudian sistem akan menampilkan halaman list instansi seperti pada Gambar 4.27. Ketika adminingin mengedit instansi maka admindapat menekan tombol edit yang terdapat pada tabel list instansi. Kemudian sistem akan menampilkan halaman edit instansi seperti pada Gambar 4.28. Ketika adminingin menghapus instansi maka admindapat menekan tombol hapus yang terdapat pada tabel list instansi. Kemudian sistem akan menghapus data instansi yang ada pada sistem.



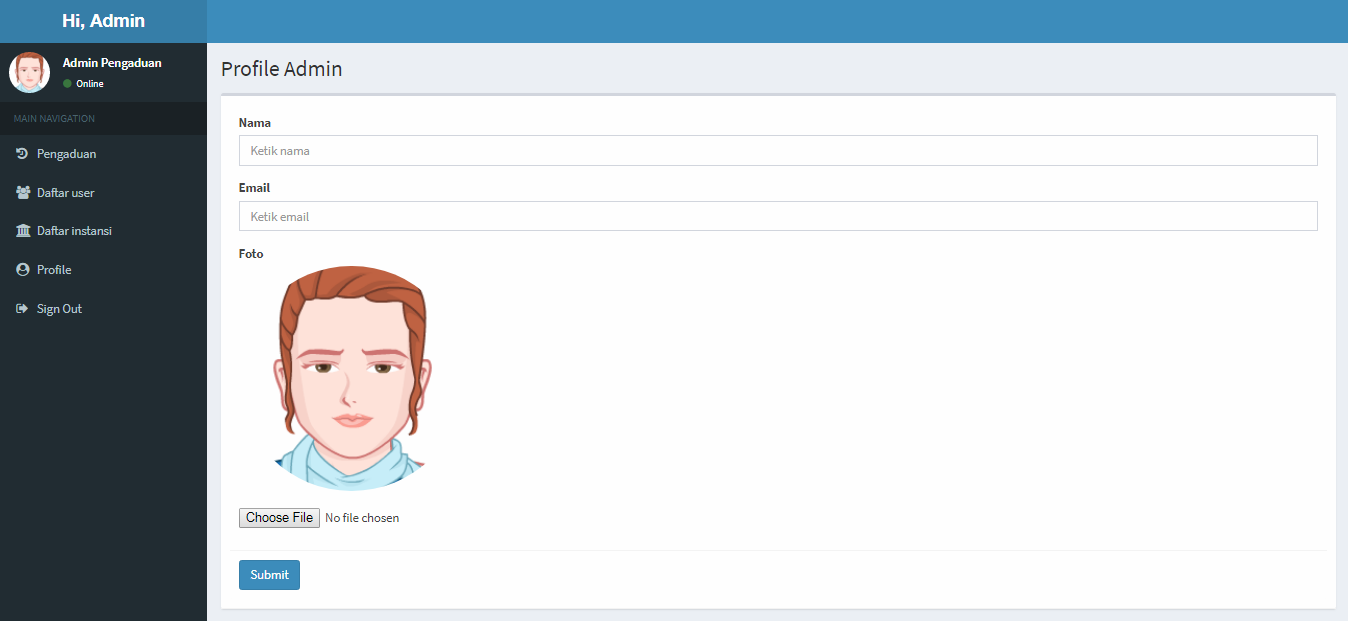
Gambar 4. 27 Tampilan Antar Muka Halaman Tambah Instansi

Gambar 4.27 merupakan tampilan dari halaman tambah instansi pada sistem pengaduan untuk admin. Ketika adminingin menambah instansi maka admin mengisi *form* dengan menginputkan nama instansi*.* Setelah adminselesai menginputkan data instansi maka adminklik tombol submit untuk menyimpan data instansi.



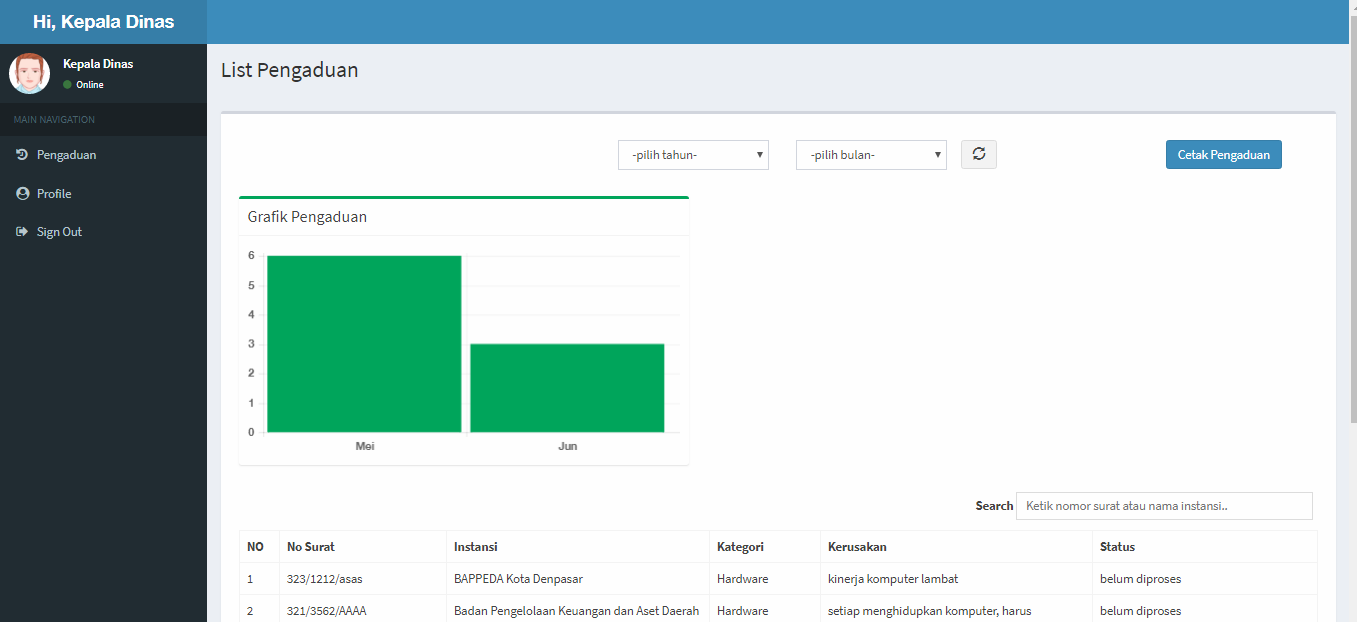
Gambar 4. 28 Tampilan Antar Muka Halaman Tambah Edit Instansi

Gambar 4.28 merupakan tampilan dari halaman edit instansi pada sistem pengaduan untuk admin. Pada halaman edit instansi, admindapat mengubah nama instansi pada *form* apabila terdapat kesalahan data.



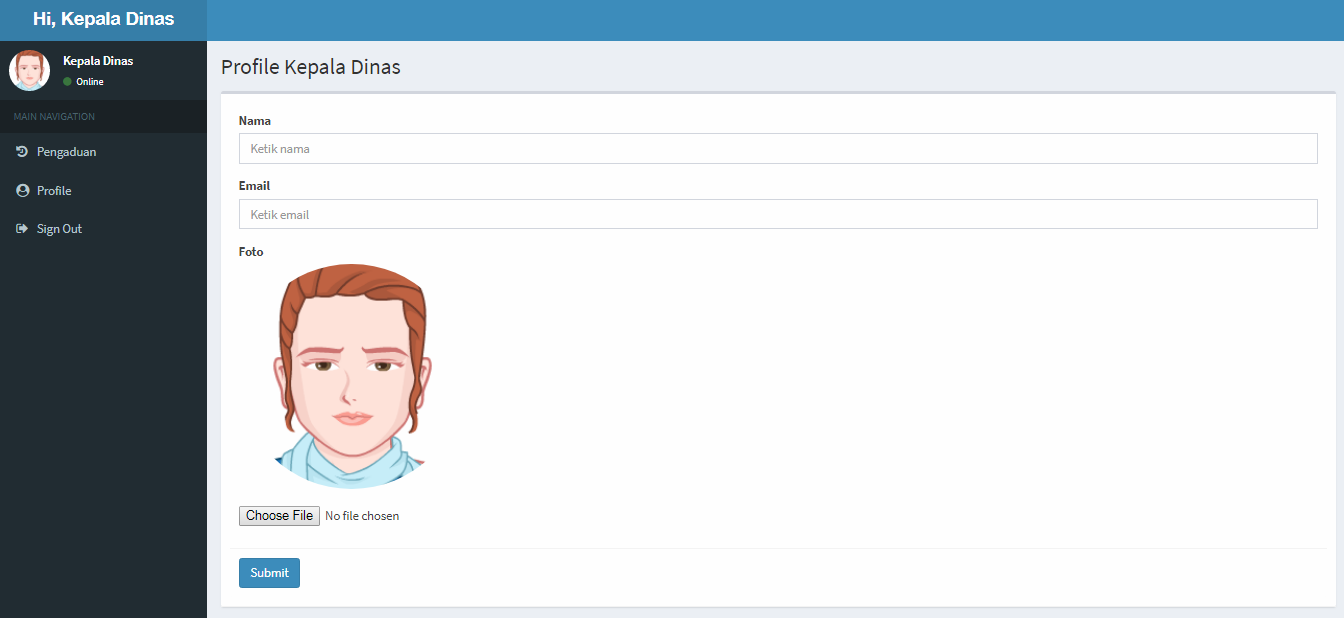
Gambar 4. 29 Tampilan Antar Muka Halaman Profile Admin

Gambar 4.29 merupakan tampilan dari halaman profile pada sistem pengaduan untuk admin. Pada halaman profile admindapat mengubah nama, email, dan foto profile sesuai keinginan admin.



Gambar 4. 30 Tampilan Antar Muka Halaman Utama Kepala Dinas

Gambar 4.30 merupakan tampilan dari halaman utama pada sistem pengaduan untuk kepala dinas. Pada halaman utama kepala dinas terdapat navigasi disebelah kiri yang berisi menu pengaduan, profile, dan *signout.* Pada halaman ini terdapat tombol cetak pengaduan yang berguna apabila kepala dinas ingin mencetak data pengaduan. Menu pilih bulan dan pilih tahun yang berguna apabila kepala dinas ingin melihat data pengaduan berdasarkan memilih bulan dan tahun yang diinginkan. Tabel pengaduan berisi informasi mengenai data pengaduan seperti nomor surat, instansi, kategori, kerusakan, comment, dan status*.*



Gambar 4. 31 Tampilan Antar Muka Halaman Profile Kepala Dinas

Gambar 4.31 merupakan tampilan dari halaman profile pada sistem pengaduan untuk kepala dinas. Pada halaman profile kepala dinasdapat mengubah nama, email, dan foto profile sesuai keinginan kepala dinas.

## Pengujian Sistem

Tahap pengujian sistem merupakan tahap menguji apakah sistem yang dibuat sudah sesuai dengan kebutuhan yang diinginkan. Pada penelitian ini tahap pengujian sistem tebagi menjadi tiga macam pengujian yaitu *black box testing, stress testing,* dan respon pengguna.

### Black Box Testing

Pengujian *black box* bertujuan untuk mengetahui apakah sistem sudah berjalan sesuai yang diinginkan atau belum. Bagian penting dari pengujian *black box* adalah fungsionalitas program yang dapat disebut *functional testing. Functional testing* merupakan sebuah metode pengujian yang berfokus pada proses eksekusi fungsi dalam program dan kesesuaian data *input* dan *output* berdasarkan kebutuhan sistem.

Tabel 4. 5 *Checklist* Pengujian *Black Box*

|  |  |  |
| --- | --- | --- |
| **Kode Kebutuhan** | **Pengguna** | **Hasil Pengujian** |
| **KF1** | Pegawai, *technical support,* admin, kepala dinas | Terpenuhi |
| **KF2** | Pegawai | Terpenuhi |
| **KF3** | Pegawai | Terpenuhi |
| **KF4** | Pegawai | Terpenuhi |
| **KF5** | *Technical support* | Terpenuhi |
| **KF6-A** | *Technical support* | Terpenuhi |
| **KF6-B** | *Technical support* | Terpenuhi |
| **KF6-C** | *Technical support* | Terpenuhi |
| **KF6-D** | *Technical support* | Terpenuhi |
| **KF7-A** | *Technical support* | Terpenuhi |
| **KF7-B** | *Technical support* | Terpenuhi |
| **KF7-C** | *Technical support* | Terpenuhi |
| **KF7-D** | *Technical support* | Terpenuhi |
| **KF8-A** | Admin | Terpenuhi |
| **KF8-B** | Admin | Terpenuhi |
| **KF8-C** | Admin | Terpenuhi |
| **KF8-D** | Admin | Terpenuhi |
| **KF8-E** | Admin | Terpenuhi |
| **KF8-F** | Admin | Terpenuhi |
| **KF8-G** | Admin | Terpenuhi |
| **KF8-H** | Admin | Terpenuhi |
| **KF9-A** | Admin | Terpenuhi |
| **KF9-B** | Admin | Terpenuhi |
| **KF9-C** | Admin | Terpenuhi |
| **KF9-D** | Admin | Terpenuhi |
| **KF10-A** | Admin | Terpenuhi |
| **KF10-B** | Admin | Terpenuhi |
| **KF10-C** | Admin | Terpenuhi |
| **KF10-D** | Admin | Terpenuhi |
| **KF11-A** | Admin | Terpenuhi |
| **KF11-B** | Admin | Terpenuhi |
| **KF11-C** | Admin | Terpenuhi |
| **KF12-A** | Kepala dinas | Terpenuhi |
| **KF12-B** | Kepala dinas | Terpenuhi |
| **KF12-C** | Kepala dinas | Terpenuhi |

Tabel x.x *Checklist* Pengujian *Black Box* merupakan pengujian *black box* yang dibuat berdasarkan analisis kebutuhan pada sub bab 3.1.1. Rincian dari hasil pengujian *black box* pada *checklist* pengujianTabel 4.5 akan dijelaskan lebih detail pada lampiran A.

### Stress Testing

Tujuan dari stress testing adalah untuk memastikan perangkat lunak tidak mengalami kerusakan dalam kondisi sumber daya yang tidak memadai. Pada stress testing sistem pengaduan terdapat dua buah skenario pengujian yaitu performance test dan stress test. Pada performance test objek penelitian yang akan dianalisis yaitu tingkat kestabilan web server, kestabilan akses dan loading time web saat diakses. Pengujian stress test dilakukan untuk mengetahui kemampuan perangkat lunak dalam memastikan aplikasi yang diluncurkan dapat menangani pengguna aplikasi sesuai jumlah yang ditargetkan.

1. Performance Test

### Pengujian Terhadap Respon Pengguna

Pengujian terhadap respon penggunabertujuan untuk mengetahui kenyamanan interaksi antara manusia dan komputer. Pada pengujian terhadap respon pengguna dilakukan dengan menyebar kuisioner terhadap sampel pengguna dari sistem pengaduan. Proses pengujian respon pengguna pada sistem pengaduan dilakukan dengan memberikan kuisioner kepada pengguna untuk mengetahui kemudahan penggunaan sistem. Pengujian dengan kuisioner dilakukan kepada 10 orang pengguna dengan form kuisioner terlampir pada lampiran B. Adapun hasil pengujian respon pengguna adalah sebagai berikut :

Tabel 4. 6 Bobot Nilai Kuisioner

|  |  |
| --- | --- |
| Keterangan | Bobot |
| Sangat Setuju | 4 |
| Setuju | 3 |
| Kurang Setuju | 2 |
| Tidak Setuju | 1 |

Tabel 4. 7 Bobot Nilai Kuisioner

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pertanyaan** | **Sangat Setuju**  **(4)** | **Setuju**  **(3)** | **Kurang Setuju**  **(2)** | **Tidak**  **Setuju**  **(1)** | **Skor** |
| 1 | 9 | 1 | 0 | 0 | 39 |
| 2 | 9 | 1 | 0 | 0 | 39 |
| 3 | 6 | 4 | 0 | 0 | 36 |
| 4 | 7 | 3 | 0 | 0 | 37 |
| 5 | 7 | 3 | 0 | 0 | 37 |
| 6 | 8 | 2 | 0 | 0 | 38 |
| 7 | 8 | 2 | 0 | 0 | 38 |
| 8 | 8 | 2 | 0 | 0 | 38 |
| 9 | 7 | 3 | 0 | 0 | 37 |
| 10 | 8 | 2 | 0 | 0 | 38 |
| **Jumlah** | 77 | 23 | 0 | 0 | 377 |
| **Presentase (%)** | 77% | 23% | 0% | 0% |  |

Dari hasil kuisioner oleh 10 pengguna dan setiap kuisioner terdiri dari 10 pertanyaan maka didapatkan hasil 77% pengguna merasa sangat setuju, 23% pengguna merasa setuju, 0% pengguna merasa kurang setuju, 0% pengguna merasa tidak setuju. Pada sistem yang dibangun dapat memberikan kemudahan, hal ini dapat dilihat dari pertanyaan nomer 1 dan nomer 2 yang memiliki skor tertinggi dengan nilai skor 39. Jadi dapat disimpulkan bahwa sistem pengaduan mampu memberikan kemudahan bagi pengguna dalam melakukan pengaduan secara *online* dan mempermudah *technical support* dalam mencari solusi kerusakan komputer.

# BAB V

**KESIMPULAN DAN SARAN**

## Kesimpulan

## Saran

# DAFTAR PUSTAKA

# LAMPIRAN A

**BLACK BOX TESTING**

**Tabel Lampiran A. 1 Pengujian Validasi Input Pada Proses Login**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF1 | | | |
| Kasus:  *Melakukan pengujian validasi input pada proses login* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Pengguna memasukkan input *string username* dan *password* yang benar pada *form* dan menekan tombol *login* | Sistem mencocokan data yang dimasukkan dan mengarahkan pengguna ke halaman utama | Sesuai Harapan |
| 2 | Pengguna memasukkan imput *string username* dan *password* yang salah pada *form* dan menekan tombol *login* | Sistem mencocokkan data yang dimasukkan dan kembali ke halaman *login* | Sesuai Harapan |

**Tabel Lampiran A. 2 Pengujian Hak Akses Proses Login**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF1 | | | |
| Kasus:  *Melakukan pengujian hak akses proses login* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Pengguna melakukan *login* dengan memilih hak akses sebagai pegawai | Sistem memberi hak akses pengguna sebagai pegawai dan menuju halaman utama pegawai | Sesuai Harapan |
| 2 | Pengguna melakukan *login* dengan memilih hak akses sebagai *technical support* | Sistem memberi hak akses pengguna sebagai pegawai dan menuju halaman utama *technical support* | Sesuai Harapan |
| 3 | Pengguna melakukan *login* dengan memilih hak akses sebagai admin | Sistem memberi hak akses pengguna sebagai pegawai dan menuju halaman utama admin | Sesuai Harapan |
| 4 | Pengguna melakukan *login* dengan memilih hak akses sebagai kepala dinas | Sistem memberi hak akses pengguna sebagai pegawai dan menuju halaman utama kepala dinas | Sesuai Harapan |

**Tabel Lampiran A. 3 Pengujian Proses *Sign Out***

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF1 | | | |
| Kasus:  *Melakukan pengujian proses sign out* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Pengguna sebagai pegawai menekan tombol *sign out* pada menu navigasi | Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman *login* | Sesuai Harapan |
| 2 | Pengguna sebagai *technical support* menekan tombol *sign out* pada menu navigasi | Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman *login* | Sesuai Harapan |
| 3 | Pengguna sebagai admin menekan tombol *sign out* pada menu navigasi | Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman *login* | Sesuai Harapan |
| 4 | Pengguna sebagai kepala dinas menekan tombol *sign out* pada menu navigasi | Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman *login* | Sesuai Harapan |

**Tabel Lampiran A. 4 Pengujian Proses Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF2 | | | |
| Kasus:  *Melakukan pengujian proses pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Pegawai menekan tombol *form* pada menu navigasi yang ada pada sistem pengaduan | Sistem menampilkan *form* pengaduan | Sesuai Harapan |
| 2 | Pegawai mengisi semua *input field* yang ada pada *form* pengaduan dan menekan tombol submit | Sistem menyimpan inputan ke *database* dan mengakses halaman posting | Sesuai Harapan |
| 3 | Pegawai tidak mengisi semua *input field* yang ada pada *form* pengaduan dan menekan tombol submit | Sistem akan menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |

**Tabel Lampiran A. 5 Pengujian Proses Melihat *History* Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF3 | | | |
| Kasus:  *Melakukan pengujian proses melihat history pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Pegawai menekan tombol posting pada menu navigasi yang ada pada sistem pengaduan | Sistem menampilkan halaman posting | Sesuai Harapan |

**Tabel Lampiran A. 6 Pengujian Proses Melihat StatusPengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF4 | | | |
| Kasus:  *Melakukan pengujian proses melihat status pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Pegawai menekan tombol posting pada menu navigasi yang ada pada sistem pengaduan | Sistem menampilkan halaman posting | Sesuai Harapan |

**Tabel Lampiran A. 7 Pengujian Proses Mendapatkan Solusi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF5 | | | |
| Kasus:  *Melakukan pengujian proses mendapatkan solusi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan tombol solusi pada halaman list pengaduan | Sistem menampilkan solusi kerusakan komputer berdasarkan data pengaduan yang diinputkan oleh pegawai | Sesuai Harapan |

**Tabel Lampiran A. 8 Pengujian Proses Menambah Keruskan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF6-A | | | |
| Kasus:  *Melakukan pengujian proses menambah kerusakan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan tombol tambah kerusakan pada halaman daftar kerusakan | Sistem menampilkan *form* tambah kerusakan | Sesuai Harapan |
| 2 | *Technical support* mengisi semua *input field* yang ada pada *form* tambah kerusakan dan menekan tombol submit | Sistem menyimpan inputan ke *database* dan menampilkan halaman daftar kerusakan | Sesuai Harapan |
| 3 | *Technical support* tidak mengisi semua *input field* yang ada pada *form* tambah kerusakan dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | *Technical support* menekan tombol *cancel* pada *form* daftar kerusakan | Sistem menampilkan halaman daftar kerusakan | Sesuai Harapan |

**Tabel Lampiran A. 9 Pengujian Proses Melihat Daftar Keruskan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF6-B | | | |
| Kasus:  *Melakukan pengujian proses melihat daftar kerusakan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan daftar kerusakan pada menu navigasi yang ada pada sistem | Sistem menampilkan halaman daftar kerusakan | Sesuai Harapan |

**Tabel Lampiran A. 10 Pengujian Proses Mengedit Keruskan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF6-C | | | |
| Kasus:  *Melakukan pengujian proses mengedit kerusakan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan tombol edit pada halaman daftar kerusakan | Sistem menampilkan *form* edit kerusakan | Sesuai Harapan |
| 2 | *Technical support* mengedit semua *input field* yang ada pada *form* edit kerusakan dan menekan tombol submit | Sistem menyimpan inputan yang telah diubah ke *database* dan menampilkan halaman daftar kerusakan | Sesuai Harapan |
| 3 | *Technical support* mengosongkan semua *input field* yang ada pada *form* edit kerusakan dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | *Technical support* tidak mengubah data kerusakan dan menekan tombol submit | Sistem akan menyimpan data yang tidak dirubah dan menampilkan halaman daftar kerusakan | Sesuai Harapan |
| 5 | *Technical support* menekan tombol *cancel* pada *form* edit kerusakan | Sistem menampilkan halaman daftar kerusakan | Sesuai Harapan |

**Tabel Lampiran A. 11 Pengujian Proses Menghapus Keruskan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF6-D | | | |
| Kasus:  *Melakukan pengujian proses menghapus kerusakan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan tombol hapus pada halaman daftar kerusakan | Sistem menampilkan pesan untuk konfirmasi hapus data kerusakan | Sesuai Harapan |
| 2 | *Technical support* menekan tombol OK | Sistem menghapus data kerusakan dan menampilkan halaman daftar kerusakan | Sesuai Harapan |
| 3 | *Technical support* menekan tombol Cancel | Sistem akan menutup jendela konfirmasi hapus kerusakan dan menampilkan halaman daftar kerusakan | Sesuai Harapan |

**Tabel Lampiran A. 12 Pengujian Proses Menambah Solusi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF7-A | | | |
| Kasus:  *Melakukan pengujian proses menambah solusi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan tombol tambah solusi pada halaman list solusi | Sistem menampilkan *form* tambah solusi | Sesuai Harapan |
| 2 | *Technical support* mengisi semua *input field* yang ada pada *form* tambah solusi dan menekan tombol submit | Sistem menyimpan inputan ke *database* dan menampilkan halaman list solusi | Sesuai Harapan |
| 3 | *Technical support* tidak mengisi semua *input field* yang ada pada *form* tambah solusi dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | *Technical support* menekan tombol *cancel* pada *form* tambah solusi | Sistem menampilkan halaman list solusi | Sesuai Harapan |

**Tabel Lampiran A. 13 Pengujian Proses Melihat List Solusi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF7-B | | | |
| Kasus:  *Melakukan pengujian proses melihat list solusi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan list solusi pada menu navigasi yang ada pada sistem | Sistem menampilkan halaman list solusi | Sesuai Harapan |

**Tabel Lampiran A. 13 Pengujian Proses Mengedit Solusi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF7-C | | | |
| Kasus:  *Melakukan pengujian proses mengedit solusi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan tombol edit pada halaman list solusi | Sistem menampilkan *form* edit solusi | Sesuai Harapan |
| 2 | *Technical support* mengedit semua *input field* yang ada pada *form* edit solusi dan menekan tombol submit | Sistem menyimpan inputan yang telah diubah ke *database* dan menampilkan halaman list solusi | Sesuai Harapan |
| 3 | *Technical support* mengosongkan semua *input field* yang ada pada *form* edit solusi dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | *Technical support* tidak mengubah data solusi dan menekan tombol submit | Sistem akan menyimpan data yang tidak dirubah dan menampilkan halaman list solusi | Sesuai Harapan |
| 5 | *Technical support* menekan tombol *cancel* pada *form* edit solusi | Sistem menampilkan halaman list solusi | Sesuai Harapan |

**Tabel Lampiran A. 14 Pengujian Proses Menghapus Solusi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF7-D | | | |
| Kasus:  *Melakukan pengujian proses menghapus solusi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support* menekan tombol hapus pada halaman list solusi | Sistem menampilkan pesan untuk konfirmasi hapus data solusi | Sesuai Harapan |
| 2 | *Technical support* menekan tombol OK | Sistem menghapus data solusi dan menampilkan halaman list solusi | Sesuai Harapan |
| 3 | *Technical support* menekan tombol Cancel | Sistem akan menutup jendela konfirmasi hapus solusi dan menampilkan halaman list solusi | Sesuai Harapan |

**Tabel Lampiran A. 15 Pengujian Proses Menambah Data Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-A | | | |
| Kasus:  *Melakukan pengujian proses menambah pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol tambah pengaduan pada halaman pengaduan admin | Sistem menampilkan *form* tambah pengaduan | Sesuai Harapan |
| 2 | Admin mengisi semua *input field* yang ada pada *form* tambah pengaduan dan menekan tombol submit | Sistem menyimpan inputan ke *database* dan menampilkan halaman pengaduan admin | Sesuai Harapan |
| 3 | Admin tidak mengisi semua *input field* yang ada pada *form* tambah pengaduan dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | Admin menekan tombol *cancel* pada *form* tambah pengaduan | Sistem menampilkan halaman pengaduan admin | Sesuai Harapan |

**Tabel Lampiran A. 16 Pengujian Proses Melihat Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-B | | | |
| Kasus:  *Melakukan pengujian proses melihat daftar pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan menu pengaduan pada navigasi yang ada pada sistem | Sistem menampilkan halaman pengaduan admin | Sesuai Harapan |

**Tabel Lampiran A. 17 Pengujian Proses Mengedit Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-C | | | |
| Kasus:  *Melakukan pengujian proses mengedit pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol edit pada halaman pengaduan | Sistem menampilkan *form* edit pengaduan | Sesuai Harapan |
| 2 | Admin mengedit semua *input field* yang ada pada *form* edit pengaduan dan menekan tombol submit | Sistem menyimpan inputan yang telah diubah ke *database* dan menampilkan halaman pengaduan admin | Sesuai Harapan |
| 3 | Admin mengosongkan semua *input field* yang ada pada *form* edit pengaduan dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | Admin tidak mengubah data pengaduan dan menekan tombol submit | Sistem akan menyimpan data yang tidak dirubah dan menampilkan halaman pengaduan admin | Sesuai Harapan |
| 5 | Admin menekan tombol *cancel* pada *form* edit pengaduan | Sistem menampilkan halaman pengaduan admin | Sesuai Harapan |

**Tabel Lampiran A. 18 Pengujian Proses Menghapus Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-D | | | |
| Kasus:  *Melakukan pengujian proses menghapus pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol hapus pada halaman pengaduan | Sistem menampilkan pesan untuk konfirmasi hapus data pengaduan | Sesuai Harapan |
| 2 | Admiin menekan tombol OK | Sistem menghapus data pengaduan dan menampilkan halaman pengaduan admin | Sesuai Harapan |
| 3 | Admin menekan tombol Cancel | Sistem akan menutup jendela konfirmasi hapus pengaduan dan menampilkan halaman pengaduan admin | Sesuai Harapan |

**Tabel Lampiran A. 19 Pengujian Proses Mengirim Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-E | | | |
| Kasus:  *Melakukan pengujian proses mengirim pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol kirim pada halaman pengaduan admin | Sistem mengirim pengaduan ke alamat *email* yang sudah ditentukan pada sistem | Sesuai Harapan |

**Tabel Lampiran A. 20 Pengujian Proses Mencetak Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-F | | | |
| Kasus:  *Melakukan pengujian proses mencetak pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin memilih tahun dan bulan pada menu pilih tahun dan pilih bulan, lalu klik tombol *refresh* pada halaman pengaduan admin | Sistem dapat menampilkan data pengaduan sesuai dengan pilihan admin berdasarkan tahun dan bulan yang dipilih pada menu | Sesuai Harapan |
| 2 | Admin klik tombol cetak pada halaman pengaduan admin | Sistem menampilkan halaman pdf yang berisi data pengaduan yang telah dipilih berdasarkan tahun dan bulan yang sebelumnya dipilih oleh admin | Sesuai Harapan |

**Tabel Lampiran A. 21 Pengujian Proses *Search* Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-G | | | |
| Kasus:  *Melakukan pengujian proses mencari pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin mencari data pengaduan dengan mengetikan nomor surat atau nama instansi | Sistem dapat menampilkan data pengaduan berdasarkan yang dicari oleh admin | Sesuai Harapan |
| 2 | Admin klik tombol cetak pada halaman pengaduan admin | Sistem menampilkan halaman pdf yang berisi data pengaduan yang telah dipilih berdasarkan tahun dan bulan yang sebelumnya dipilih oleh admin | Sesuai Harapan |

**Tabel Lampiran A. 22 Pengujian Proses Melihat Grafik Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF8-H | | | |
| Kasus:  *Melakukan pengujian proses melihat grafik pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin melihat grafik pengaduan pada halaman pengaduan admin | Sistem dapat menampilkan grafik pengaduan pada halaman pengaduan admin | Sesuai Harapan |

**Tabel Lampiran A. 23 Pengujian Proses Menambah Daftar User**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF9-A | | | |
| Kasus:  *Melakukan pengujian proses menambah daftar user* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol tambah user pada halaman daftar user | Sistem menampilkan *form* tambah user | Sesuai Harapan |
| 2 | Admin mengisi semua *input field* yang ada pada *form* tambah user dan menekan tombol submit | Sistem menyimpan inputan ke *database* dan menampilkan halaman daftar user | Sesuai Harapan |
| 3 | Admin tidak mengisi semua *input field* yang ada pada *form* tambah user dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | Admin menekan tombol *cancel* pada *form* tambah user | Sistem menampilkan halaman daftar user | Sesuai Harapan |

**Tabel Lampiran A. 24 Pengujian Proses Melihat Daftar User**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF9-B | | | |
| Kasus:  *Melakukan pengujian proses melihat daftar user* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan menu daftar user pada navigasi yang ada pada sistem | Sistem menampilkan halaman daftar user | Sesuai Harapan |

**Tabel Lampiran A. 25 Pengujian Proses Mengedit Daftar User**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF9-C | | | |
| Kasus:  *Melakukan pengujian proses mengedit daftar user* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol edit pada halaman daftar user | Sistem menampilkan *form* edit user | Sesuai Harapan |
| 2 | Admin mengedit semua *input field* yang ada pada *form* edit user dan menekan tombol submit | Sistem menyimpan inputan yang telah diubah ke *database* dan menampilkan halaman daftar user | Sesuai Harapan |
| 3 | Admin mengosongkan semua *input field* yang ada pada *form* edit user dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | Admin tidak mengubah data user dan menekan tombol submit | Sistem akan menyimpan data yang tidak dirubah dan menampilkan halaman daftar user | Sesuai Harapan |
| 5 | Admin menekan tombol *cancel* pada *form* edit user | Sistem menampilkan halaman daftar user | Sesuai Harapan |

**Tabel Lampiran A. 26 Pengujian Proses Menghapus Daftar User**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF9-D | | | |
| Kasus:  *Melakukan pengujian proses menghapus daftar user* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol hapus pada halaman daftar user | Sistem menampilkan pesan untuk konfirmasi hapus data user | Sesuai Harapan |
| 2 | Admiin menekan tombol OK | Sistem menghapus data user dan menampilkan halaman daftar user | Sesuai Harapan |
| 3 | Admin menekan tombol Cancel | Sistem akan menutup jendela konfirmasi hapus user dan menampilkan halaman daftar user | Sesuai Harapan |

**Tabel Lampiran A. 27 Pengujian Proses Menambah Daftar Instansi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF10-A | | | |
| Kasus:  *Melakukan pengujian proses menambah daftar instansi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol tambah instansi pada halaman daftar instansi | Sistem menampilkan *form* tambah instansi | Sesuai Harapan |
| 2 | Admin mengisi semua *input field* yang ada pada *form* tambah instansi dan menekan tombol submit | Sistem menyimpan inputan ke *database* dan menampilkan halaman daftar instansi | Sesuai Harapan |
| 3 | Admin tidak mengisi semua *input field* yang ada pada *form* tambah instansi dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | Admin menekan tombol *cancel* pada *form* tambah instansi | Sistem menampilkan halaman daftar instansi | Sesuai Harapan |

**Tabel Lampiran A. 28 Pengujian Proses Melihat Daftar Instansi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF10-B | | | |
| Kasus:  *Melakukan pengujian proses melihat daftar instansi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan menu daftar instansi pada navigasi yang ada pada sistem | Sistem menampilkan halaman daftar instansi | Sesuai Harapan |

**Tabel Lampiran A. 29 Pengujian Proses Mengedit Daftar Instansi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF10-C | | | |
| Kasus:  *Melakukan pengujian proses mengedit daftar instansi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol edit pada halaman daftar instansi | Sistem menampilkan *form* edit instansi | Sesuai Harapan |
| 2 | Admin mengedit semua *input field* yang ada pada *form* edit instansi dan menekan tombol submit | Sistem menyimpan inputan yang telah diubah ke *database* dan menampilkan halaman daftar instansi | Sesuai Harapan |
| 3 | Admin mengosongkan semua *input field* yang ada pada *form* edit instansi dan menekan tombol submit | Sistem menampilkan pesan pada *form input field* bahwa *form input field* harus diisi inputan | Sesuai Harapan |
| 4 | Admin tidak mengubah data instansi dan menekan tombol submit | Sistem akan menyimpan data yang tidak dirubah dan menampilkan halaman daftar instansi | Sesuai Harapan |
| 5 | Admin menekan tombol *cancel* pada *form* edit instansi | Sistem menampilkan halaman daftar instansi | Sesuai Harapan |

**Tabel Lampiran A. 30 Pengujian Proses Menghapus Daftar Instansi**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF10-D | | | |
| Kasus:  *Melakukan pengujian proses menghapus daftar instansi* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Admin menekan tombol hapus pada halaman daftar instansi | Sistem menampilkan pesan untuk konfirmasi hapus data instansi | Sesuai Harapan |
| 2 | Admiin menekan tombol OK | Sistem menghapus data instansi dan menampilkan halaman daftar instansi | Sesuai Harapan |
| 3 | Admin menekan tombol Cancel | Sistem akan menutup jendela konfirmasi hapus instansi dan menampilkan halaman daftar instansi | Sesuai Harapan |

**Tabel Lampiran A. 32 Pengujian Proses Mendapatkan Informasi Dan Cetak Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF11-A | | | |
| Kasus:  *Melakukan pengujian proses mendapatkan informasi dan cetak pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Kepala dinas memilih tahun dan bulan pada menu pilih tahun dan pilih bulan, lalu klik tombol *refresh* pada halaman pengaduan kepala dinas | Sistem dapat menampilkan data pengaduan sesuai dengan pilihan kepala dinas berdasarkan tahun dan bulan yang dipilih pada menu | Sesuai Harapan |
| 2 | Kepala dinas klik tombol cetak pada halaman pengaduan kepala dinas | Sistem menampilkan halaman pdf yang berisi data pengaduan yang telah dipilih berdasarkan tahun dan bulan yang sebelumnya dipilih oleh kepala dinas | Sesuai Harapan |

**Tabel Lampiran A. 33 Pengujian Proses Melihat Grafik Pengaduan**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF11-B | | | |
| Kasus:  *Melakukan pengujian proses melihat grafik pengaduan* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | Kepala dinas melihat grafik pengaduan pada halaman pengaduan kepala dinas | Sistem dapat menampilkan grafik pengaduan pada halaman pengaduan kepala dinas | Sesuai Harapan |

**Tabel Lampiran A. 34 Pengujian Proses Mengubah Nama Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF12-A | | | |
| Kasus:  *Melakukan pengujian proses mengubah nama profile* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support,* kepala dinas, dan admin klik menu profile pada navigasi | Sistem menampilkan halaman profile masing-masing user | Sesuai Harapan |
| 2 | *Technical support,* kepala dinas, dan admin mengubah nama profile yang ada pada halaman profile masing-masing user dan klik tombol submit | Sistem dapat mengubah nama profile pada halaman profile masing-masing user | Sesuai Harapan |

**Tabel Lampiran A. 35 Pengujian Proses Mengubah Alamat *Email* Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF12-B | | | |
| Kasus:  *Melakukan pengujian proses mengubah alamat email profile* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support,* kepala dinas, dan admin mengubah alamat *email* profile yang ada pada halaman profile masing-masing user dan klik tombol submit | Sistem dapat mengubah alamat *email* profile pada halaman profile masing-masing user | Sesuai Harapan |

**Tabel Lampiran A. 36 Pengujian Proses Mengubah Foto Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| Kode Kebutuhan: KF12-C | | | |
| Kasus:  *Melakukan pengujian proses mengubah foto profile* | | | |
| **No.** | **Skenario Pengujian** | **Hasil yang Diharapakan** | **Hasil Pengujian** |
| 1 | *Technical support,* kepala dinas, dan admin mengubah foto profile yang ada pada halaman profile masing-masing user dan klik tombol submit | Sistem dapat mengubah foto profile pada halaman profile masing-masing user | Sesuai Harapan |

# LAMPIRAN B

**FORM KUISIONER PENGGUNA SISTEM**

**KUISIONER RESPON PENGGUNA**

**TERHADAP SISTEM PENGADUAN**

Berikut ini adalah kuisioner untuk mengetahui respon pengguna yang berkaitan dengan penelitian tentang Sistem Pengaduan Kerusakan Komputer Dan Rekomendasi Solusi Menggunakan Algoritma Cosine Similarity. Oleh karena itu disela-sela kesibukan anda, dimohon dengan hormat kesediaan anda untuk dapat mengisi kuisioner berikut ini.

**IDENTITAS RESPONDEN**

Nama Responden :

Pekerjaan :

**DAFTAR PERNYATAAN**

Bobot

SS = Sangat Setuju = 4

S = Setuju = 3

KS = Kurang Setuju = 2

TS = Tidak Setuju = 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Pernyataan | Respon | | | |
| SS | S | KS | TS |
| 1 | Sistem pengaduan kerusakan komputer dapat memberikan kemudahan dalam melakukan pengaduan secara *online* |  |  |  |  |
| 2 | Fitur solusi dari kerusakan komputer memudahkan *technical support* dalam mencari solusi kerusakan komputer |  |  |  |  |
| 3 | Penjelasan yang keluar ketika fitur solusi diklik sudah sesuai dengan masalah kerusakan komputer yang diadu |  |  |  |  |
| 4 | *Technical support* diberi kemudahan dalam mengelola daftar kerusakan |  |  |  |  |
| 5 | *Technical support* diberi kemudahan dalam mengelola list solusi |  |  |  |  |
| 6 | Admindiberi kemudahan dalam mengelola daftar pengaduan |  |  |  |  |
| 7 | Admindiberi kemudahan dalam mengelola daftar user |  |  |  |  |
| 8 | Admindiberi kemudahan dalam mengelola daftar instansi |  |  |  |  |
| 9 | Kepala dinas diberi kemudahan dalam rekap pengaduan |  |  |  |  |
| 10 | *Technical support,* admin, dan kepala dinas diberi kemudahan dalam mengelola profile masing-masing user |  |  |  |  |

**Catatan:**