

**SISTEM PENGADUAN KERUSAKAN KOMPUTER DAN
REKOMENDASI SOLUSI MENGGUNAKAN ALGORITMA COSINE
SIMILARITY**

SKRIPSI



**LUH GEDE AYU CANDRAWATI
NIM. 1408605018**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA
BUKIT JIMBARAN
2018**

SURAT PERNYATAAN KEASLIAN KARYA ILMIAH

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

SISTEM PENGADUAN KERUSAKAN KOMPUTER DAN REKOMENDASI SOLUSI MENGGUNAKAN ALGORITMA COSINE SIMILARITY

Nama : Luh Gede Ayu Candrawati

NIM : 1408605018

Program Studi : Teknik Informatika

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

Nomor telp/HP : 082188686362

Alamat : Jalan Palapa XIV Gang Ikan Sarden No. 6, Denpasar, Bali

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

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

Denpasar, 26 Juni 2018

Yang membuat pernyataan,

Luh Gede Ayu Candrawati

NIM. 1408605018

LEMBAR PENGESAHAN TUGAS AKHIR

Judul : Sistem Pengaduan Kerusakan Komputer Dan
Rekomendasi Solusi Menggunakan Algoritma *Cosine
Similarity*

Nama : Luh Gede Ayu Candrawati

NIM : 1408605018

Tanggal Disetujui : 26 Juni 2018

Reviewer I

Reviewer II

I Gede Santi Astawa, S.T, M.Cs
NIP. 198012062006041003

Ida Bagus Gede Dwidasmara, S.Kom., M.Cs.
NIP. 198503152010121007

Mengetahui,
Komisi Seminar dan Tugas Akhir
Jurusan Ilmu Komputer FMIPA UNUD
Ketua,

I Gede Arta Wibawa, S.T., M.Kom.
NIP. 19831022 200812 1 001

Judul : Siste 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

SURAT PERNYATAAN KEASLIAN KARYA ILMIAH.....	ii
LEMBAR PENGESAHAN TUGAS AKHIR	iii
ABSTRAK.....	iv
ABSTRACT.....	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian.....	3
1.6.1 Metode Pengumpulan Data.....	4
1.6.2 Metode Pengembangan Sistem	5
1.6.3 Analisis Kebutuhan.....	5
1.6.4 Perancangan Sistem	6
1.6.5 Implementasi Sistem.....	6
1.6.6 Pengujian Sistem.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Tinjauan Studi.....	7
2.2 Sistem Informasi Manajemen	10
2.3 Pengaduan	11
2.4 Preprocessing Data	11
2.5 TF-IDF (<i>Term Frequency- Inversed Document Frequency</i>)	12

2.6	<i>Cosine Similarity</i>	13
2.7	Model Pengembangan <i>Waterfall</i>	14
BAB III ANALISIS DAN PERANCANGAN SISTEM		17
3.1	Analisis Kebutuhan Sistem	17
3.1.1	Kebutuhan Fungsional	17
3.1.2	Kebutuhan Non Fungsional	20
3.2	Perancangan Sistem	21
3.2.1	<i>Use Case Diagram</i>	21
3.2.2	<i>Activity Diagram</i>	27
3.2.3	<i>Class Diagram</i>	40
3.2.4	<i>Sequence Diagram</i>	42
3.2.5	<i>Entity Relational Diagram</i>	53
3.2.6	<i>Preprocessing</i> , TF-IDF, <i>Cosine Similarity</i>	55
3.2.7	Perancangan Antar Muka.....	58
3.2.8	Skenario Pengujian Sistem	66
BAB IV HASIL DAN PEMBAHASAN		69
4.1	Gambaran Umum Sistem.....	69
4.2	Lingkungan Implementasi	69
4.3	Implementasi Basis Data	70
4.4	Implementasi Sistem.....	73
4.4.1	Implementasi Proses <i>Preprocessing</i> Pada Sistem.....	73
4.4.2	Implementasi Proses <i>Indexing</i> Pada Sistem.....	75
4.4.3	Implementasi Proses Perhitungan TF-IDF dan Bobot <i>Query</i> Pada Sistem.....	76
4.4.4	Implementasi Pencarian Solusi Kerusakan Menggunakan Algoritma <i>Cosine Similarity</i>	77
4.5	Antar Muka Sistem	79
4.6	Pengujian Sistem.....	100
4.6.1	Black Box Testing	100
4.6.2	Stress Testing	102
4.6.3	Pengujian Terhadap Respon Pengguna.....	107

BAB V KESIMPULAN	109
5.1 Kesimpulan	109
5.2 Saran	109
DAFTAR PUSTAKA	110
LAMPIRAN A	112
LAMPIRAN B	132
LAMPIRAN C	134

DAFTAR TABEL

Tabel 3. 1 Kebutuhan Sistem Berdasarkan Analisa	17
Tabel 3. 2 Kebutuhan Fungsional Sistem	18
Tabel 3. 3 Kebutuhan Non Fungsional Sistem	21
Tabel 3. 4 Definisi Aktor Pada Use Case Diagram	22
Tabel 3. 5 Definisi Use Case pada Use Case Diagram Pegawai	23
Tabel 3. 6 Definisi Use Case pada Use Case Diagram Technical Support	23
Tabel 3. 7 Definisi Use Case pada Use Case Diagram Admin	25
Tabel 3. 8 Definisi Use Case pada Use Case Diagram Kepala Dinas	26
Tabel 3. 9 Hubungan Class dan Use Case Diagram	41
Tabel 3. 10 Skenario Black Box Testing	67
Tabel 3. 11 Skenario Pengujian Pada Performance Test	67
Tabel 4. 1 Kode Preprocessing Teks	74
Tabel 4. 2 Kode Indexing	75
Tabel 4. 3 Kode Proses Perhitungan TF-IDF dan Bobot Query	76
Tabel 4. 4 Proses Perhitungan Algoritma Cosine Similarity	77
Tabel 4. 5 Checklist Pengujian Black Box	101
Tabel 4. 6 Pengujian Menggunakan 1 Term	103
Tabel 4. 7 Pengujian Menggunakan 2 Term	103
Tabel 4. 8 Pengujian Menggunakan 3 Term	104
Tabel 4. 9 Rata-rata Waktu Hasil Pengujian	104
Tabel 4. 10 Bobot Nilai Kuisioner	108
Tabel 4. 11 Bobot Nilai Kuisioner	108
Tabel Lampiran A. 1 Pengujian Validasi Input Pada Proses Login	113
Tabel Lampiran A. 2 Pengujian Hak Akses Proses Login	113
Tabel Lampiran A. 3 Pengujian Proses Sign Out	114
Tabel Lampiran A. 4 Pengujian Proses Pengaduan	115
Tabel Lampiran A. 5 Pengujian Proses Melihat History Pengaduan	115
Tabel Lampiran A. 6 Pengujian Proses Melihat Status Pengaduan	116
Tabel Lampiran A. 7 Pengujian Proses Mendapatkan Solusi	116
Tabel Lampiran A. 8 Pengujian Proses Menambah Kerusakan	116
Tabel Lampiran A. 9 Pengujian Proses Melihat Daftar Kerusakan	117
Tabel Lampiran A. 10 Pengujian Proses Mengedit Kerusakan	117
Tabel Lampiran A. 11 Pengujian Proses Menghapus Kerusakan	118
Tabel Lampiran A. 12 Pengujian Proses Menambah Solusi	119
Tabel Lampiran A. 13 Pengujian Proses Melihat List Solusi	119
Tabel Lampiran A. 14 Pengujian Proses Menghapus Solusi	120
Tabel Lampiran A. 15 Pengujian Proses Menambah Data Pengaduan	121
Tabel Lampiran A. 16 Pengujian Proses Melihat Pengaduan	122

Tabel Lampiran A. 17 Pengujian Proses Mengedit Pengaduan	122
Tabel Lampiran A. 18 Pengujian Proses Menghapus Pengaduan.....	123
Tabel Lampiran A. 19 Pengujian Proses Mengirim Pengaduan	123
Tabel Lampiran A. 20 Pengujian Proses Mencetak Pengaduan	124
Tabel Lampiran A. 21 Pengujian Proses <i>Search</i> Pengaduan	124
Tabel Lampiran A. 22 Pengujian Proses Melihat Grafik Pengaduan.....	125
Tabel Lampiran A. 23 Pengujian Proses Menambah Daftar User	125
Tabel Lampiran A. 24 Pengujian Proses Melihat Daftar User.....	126
Tabel Lampiran A. 25 Pengujian Proses Mengedit Daftar User	126
Tabel Lampiran A. 26 Pengujian Proses Menghapus Daftar User	127
Tabel Lampiran A. 27 Pengujian Proses Menambah Daftar Instansi	127
Tabel Lampiran A. 28 Pengujian Proses Melihat Daftar Instansi	128
Tabel Lampiran A. 29 Pengujian Proses Mengedit Daftar Instansi.....	128
Tabel Lampiran A. 30 Pengujian Proses Menghapus Daftar Instansi.....	129
Tabel Lampiran A. 31 Pengujian Proses Mendapatkan Informasi Dan Cetak Pengaduan	129
Tabel Lampiran A. 32 Pengujian Proses Melihat Grafik Pengaduan.....	130
Tabel Lampiran A. 33 Pengujian Proses Mengubah Nama Profile	130
Tabel Lampiran A. 34 Pengujian Proses Mengubah Alamat <i>Email</i> Profile.....	131
Tabel Lampiran A. 35 Pengujian Proses Mengubah Foto Profile.....	131

DAFTAR GAMBAR

Gambar 1. 1 Diagram <i>Fish Bone</i> Penelitian	4
Gambar 1. 2 Metode <i>Waterfall</i> (Pressman, 2015)	5
Gambar 2. 1 Metode <i>Waterfall</i> (Pressman, 2015)	16
Gambar 3. 1 <i>Use Case Diagram</i> Pegawai dalam Sistem Pengaduan.....	22
Gambar 3. 2 <i>Use Case Diagram Technical Support</i> dalam Sistem Pengaduan	23
Gambar 3. 3 <i>Use Case Diagram Admin</i> dalam Sistem Pengaduan	24
Gambar 3. 4 <i>Use Case Diagram Kepala Dinas</i> dalam Sistem Pengaduan.....	26
Gambar 3. 5 <i>Activity Diagram</i> Autentikasi	27
Gambar 3. 6 <i>Activity Diagram</i> Melakukan Pengaduan.....	28
Gambar 3. 7 <i>Activity Diagram</i> Melihat <i>History</i> Pengaduan	29
Gambar 3. 8 <i>Activity Diagram</i> Melihat Status Pengaduan	29
Gambar 3. 9 <i>Activity Diagram</i> Mendapatkan Solusi Kerusakan	30
Gambar 3. 10 <i>Activity Diagram</i> Melihat Daftar Kerusakan	31
Gambar 3. 11 <i>Activity Diagram</i> Menambah Daftar Kerusakan	32
Gambar 3. 12 <i>Activity Diagram</i> Edit Daftar Kerusakan	33
Gambar 3. 13 <i>Activity Diagram</i> Hapus Daftar Kerusakan.....	34
Gambar 3. 14 <i>Activity Diagram Search</i> Data Pengaduan	35
Gambar 3. 15 <i>Activity Diagram</i> Kirim Email Data Pengaduan	36
Gambar 3. 16 <i>Activity Diagram</i> Melihat Profile Admin.....	36
Gambar 3. 17 <i>Activity Diagram</i> Mengedit Profile Admin.....	37
Gambar 3. 18 <i>Activity Diagram</i> Mencetak Data Pengaduan	38
Gambar 3. 19 <i>Activity Diagram</i> Melihat Grafik	39
Gambar 3. 20 <i>Class Diagram</i>	40
Gambar 3. 21 <i>Sequence Diagram</i> Autentikasi	42
Gambar 3. 22 <i>Sequence Diagram</i> Melakukan Pengaduan.....	43
Gambar 3. 23 <i>Sequence Diagram</i> Melihat <i>History</i> Pengaduan	44
Gambar 3. 24 <i>Sequence Diagram</i> Melihat Status Pengaduan.....	44
Gambar 3. 25 <i>Sequence Diagram</i> Mendapatkan Solusi Pengaduan	45
Gambar 3. 26 <i>Sequence Diagram</i> Melihat Daftar Kerusakan.....	46
Gambar 3. 27 <i>Sequence Diagram</i> Menambah Daftar Kerusakan	47
Gambar 3. 28 <i>Sequence Diagram</i> Edit Kerusakan	48
Gambar 3. 29 <i>Sequence Diagram</i> Hapus Kerusakan.....	49
Gambar 3. 30 <i>Sequence Diagram Search</i> Pengaduan	49
Gambar 3. 31 <i>Sequence Diagram</i> Kirim Pengaduan	50
Gambar 3. 32 <i>Sequence Diagram</i> Melihat <i>Profile</i> Admin.....	51
Gambar 3. 33 <i>Sequence Diagram</i> Mengedit <i>Profile</i> Admin.....	51
Gambar 3. 34 <i>Sequence Diagram</i> Cetak Pengaduan	52
Gambar 3. 35 <i>Sequence Diagram</i> Melihat Grafik Pengaduan	53

Gambar 3. 36 <i>Entity Relational Diagram</i>	54
Gambar 3. 37 <i>Flowchart Preprocessing</i>	56
Gambar 3. 38 <i>Flowchart TF-IDF</i>	57
Gambar 3. 39 <i>Flowchart</i> Perhitungan Kemiripan Teks.....	58
Gambar 3. 40 Desain Tampilan Login	59
Gambar 3. 41 Desain Tampilan Halaman Awal Pegawai.....	60
Gambar 3. 42 Desain Tampilan Halaman Posting.....	60
Gambar 3. 43 Desain Tampilan Halaman Awal Admin	61
Gambar 3. 44 Desain Tampilan Halaman Daftar User	62
Gambar 3. 45 Desain Tampilan Halaman Profile	62
Gambar 3. 46 Desain Tampilan Halaman Awal <i>Technical Support</i>	63
Gambar 3. 47 Desain Tampilan Halaman Daftar Kerusakan.....	64
Gambar 3. 48 Desain Tampilan Halaman List Solusi.....	65
Gambar 3. 49 Desain Tampilan Halaman Kepala Dinas	66
Gambar 4. 1 Implementasi Basis Data	70
Gambar 4. 2 Tampilan Antar Muka Halaman <i>Login</i>	79
Gambar 4. 3 Tampilan Antar Muka Halaman Utama Pegawai	80
Gambar 4. 4 Tampilan Antar Muka Halaman Profil	80
Gambar 4. 5 Tampilan Antar Muka Halaman Alur	81
Gambar 4. 6 Tampilan Antar Muka Halaman Form.....	82
Gambar 4. 7 Tampilan Antar Muka Halaman Posting	82
Gambar 4. 8 Tampilan Antar Muka Halaman Kontak Kami	82
Gambar 4. 9 Tampilan Antar Muka Halaman List Pengaduan <i>Technical Support</i>	83
Gambar 4. 10 Tampilan Antar Muka Halaman Solusi	83
Gambar 4. 11 Tampilan Antar Muka Halaman Daftar Kerusakan	84
Gambar 4. 12 Tampilan Antar Muka Halaman Tambah Kerusakan.....	85
Gambar 4. 13 Tampilan Antar Muka Halaman Edit Kerusakan	85
Gambar 4. 14 Tampilan Antar Muka Halaman List Solusi	86
Gambar 4. 15 Tampilan Antar Muka Halaman Tambah Solusi.....	87
Gambar 4. 16 Tampilan Antar Muka Halaman Edit Solusi	88
Gambar 4. 17 Tampilan Antar Muka Halaman Profile.....	89
Gambar 4. 18 Tampilan Antar Muka Halaman Utama Admin	89
Gambar 4. 19 Tampilan Antar Muka Halaman Tambah Pengaduan	90
Gambar 4. 20 Tampilan Antar Muka Halaman Cetak Pengaduan	91
Gambar 4. 21 Tampilan Antar Muka <i>Search</i> Pengaduan	91
Gambar 4. 22 Tampilan Antar Muka Halaman Edit Pengaduan.....	92
Gambar 4. 23 Tampilan Antar Muka Halaman List User.....	93
Gambar 4. 24 Tampilan Antar Muka Halaman Tambah User	94
Gambar 4. 25 Tampilan Antar Muka Halaman Edit User	94
Gambar 4. 26 Tampilan Antar Muka Halaman List Instansi	95
Gambar 4. 27 Tampilan Antar Muka Halaman Tambah Instansi	96

Gambar 4. 28 Tampilan Antar Muka Halaman Tambah Edit Instansi.....	96
Gambar 4. 29 Tampilan Antar Muka Halaman List <i>Email</i>	97
Gambar 4. 30 Tampilan Antar Muka Halaman Tambah List <i>Email</i>	97
Gambar 4. 31 Tampilan Antar Muka Halaman Edit List <i>Email</i>	98
Gambar 4. 32 Tampilan Antar Muka Halaman Profile Admin.....	98
Gambar 4. 33 Tampilan Antar Muka Halaman Utama Kepala Dinas.....	99
Gambar 4. 34 Tampilan Antar Muka Halaman Profile Kepala Dinas	100
Gambar 4. 35 Grafik Peningkatan Waktu Proses Pencarian Solusi	105
Gambar 4.36 Skenario <i>Stress Test</i> Sistem	106
Gambar 4. 37 Grafik <i>Stress Test</i> Sistem.....	107

DAFTAR LAMPIRAN

LAMPIRAN A

LAMPIRAN B

LAMPIRAN C

BAB I

PENDAHULUAN

1.1 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. Akan lebih baik apabila ada dibuat sebuah sistem yang dapat melakukan proses pengaduan secara *online* sehingga dapat mempercepat proses pengaduan. Selain itu berdasarkan hasil wawancara juga diperlukan fungsi yang dapat menduga masalah yang terjadi pada kerusakan komputer yaitu dengan memberikan informasi kerusakan awal kepada *technical support*. Dalam tujuan untuk membantu *technical support* dan pegawai maka diperlukan sebuah sistem pengaduan kerusakan komputer yang didalamnya berisi fitur rekomendasi solusi cepat dari kerusakan komputer. Salah satu metode yang dapat menduga berbasis kasus kerusakan yaitu dengan *text similarity*. Pada sistem yang dirancang, pegawai menginputkan gejala dalam bentuk *text* dengan mencocokkan antara *query user* dalam bentuk *text* dengan *database*. Beberapa penelitian mengenai pencocokan dua buah teks telah dilakukan. Nurdiana dkk. (2016) melakukan perbandingan 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.

1.2 Rumusan Masalah

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

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

1.3 Tujuan Penelitian

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

- a. Membangun sistem pengaduan yang dapat merekomendasi solusi kerusakan komputer yang ada di dinas kota Denpasar
- b. Mengetahui waktu yang dibutuhkan oleh sistem dalam memproses *query* hingga merekomendasikan solusi yang didapatkan oleh sistem
- c. Mengetahui respon pengguna terhadap sistem pengaduan kerusakan komputer

1.4 Batasan Masalah

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

- a. Rekomendasi solusi dari kerusakan komputer menggunakan metode *text similarity* dengan algoritma *cosine similarity*
- b. Sistem yang dibangun merupakan sistem berbasis web *responsive*.

1.5 Manfaat Penelitian

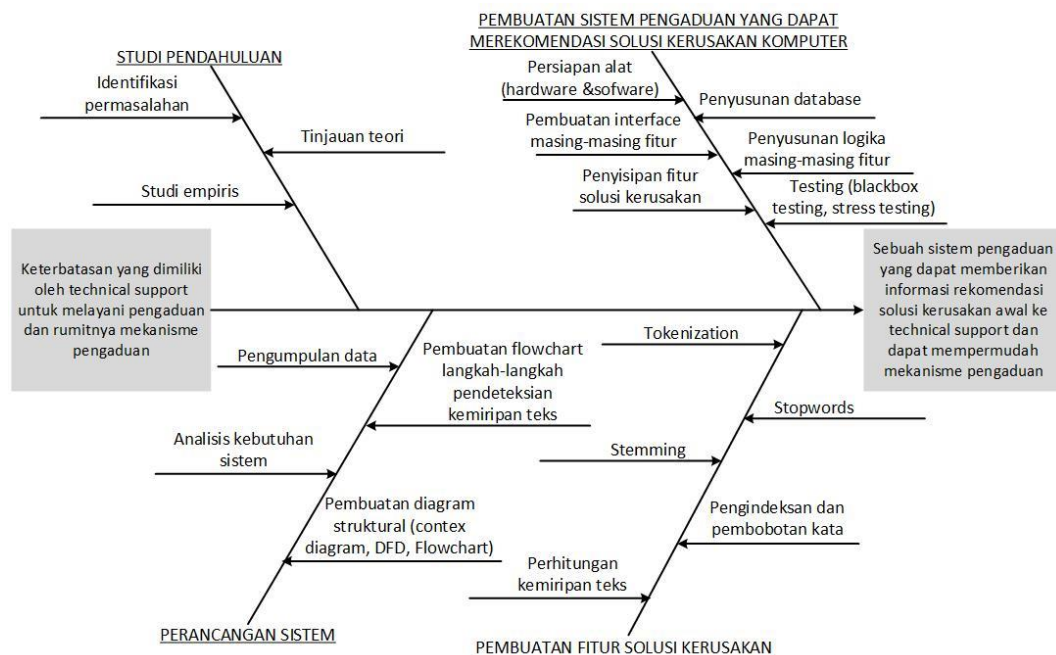
Beberapa manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut:

- a. Bagi *technical support* dinas kota Denpasar, membantu mendapatkan solusi cepat dari kerusakan komputer. Sehingga dapat mempercepat pekerjaan *technical support*.
- b. Bagi pegawai dinas kota Denpasar, membantu melakukan pengaduan kerusakan komputer secara *online* agar pekerjaan pegawai menjadi lebih cepat dan efisien.
- c. 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*.

1.6 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

1.6.1 Metode Pengumpulan Data

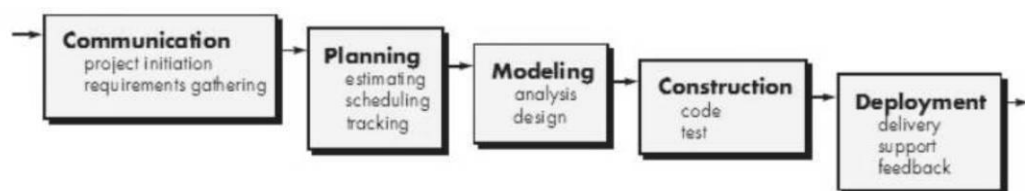
Tahap ini pengumpulan data diperoleh dari metode studi pustaka atau data sekunder bersumber pada buku dan juga data primer dari para ahli yang dikumpulkan oleh peneliti sendiri. Data yang digunakan pada penelitian ini adalah berupa data dari berbagai macam solusi permasalahan mengenai kerusakan komputer. 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 mewawancarai 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 tabel solusi 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.

1.6.2 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)

1.6.3 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan dengan mencari informasi mengenai kebutuhan sistem dan data yang akan dibangun. Dalam mencari informasi, dilakukan dengan mewawancarai 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.

1.6.4 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.

1.6.5 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*

1.6.6 Pengujian Sistem

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

BAB II

TINJAUAN PUSTAKA

2.1 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.

2. *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 mencocokkan 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* :

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

(Azhar Firdaus, Ernawati, dan Arie Vatesia, 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%.

4. **Perbandingan Metode *Cosine Similarity* Dengan Metode *Jaccard Similarity* Pada Aplikasi 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 neighbor (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 parameter K (jumlah dari tetangga terdekat). Oleh karena itu metode *cosine similarity* dapat menjadi alternatif untuk mencari kemiripan *text mining*.

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

(Rizki Tri Wahyuni, Dhidik Prastiyanto, dan Eko Suprptono, 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, *cosine similarity* dapat menjadi solusi agar dapat mempermudah proses menghitung kemiripan kata yang ada dalam dokumen.

2.2 Sistem Informasi Manajemen

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 sistem informasi manajemen 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.

2.3 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 memberikan solusi kerusakan. Apabila *technical support* tidak sanggup memperbaiki maka admin akan mengirim pengaduan ke dinas pusat.

2.4 Preprocessing Data

Preprocessing adalah proses pengubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai kebutuhan, untuk proses *text mining* yang lebih lanjut (Fauzi, 2014). Tahap preprocessing terdiri dari *casefolding*, *tokenization*, *filtering*, dan *stemming*.

- a. *Casefolding* adalah proses menghilangkan karakter-karakter seperti tanda baca dan mengubah semua huruf kapital menjadi huruf kecil (*lowercase*).
- b. *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.

- c. *Filtering* adalah proses menghapus kata-kata tidak penting dari hasil *tokenization* dengan cara mencocokkan kata tersebut dengan kata yang terdapat pada daftar kata tidak penting (*stopwords*).
- d. *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)

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

TF-IDF merupakan metode pembobotan *term* yang banyak digunakan sebagai metode pembandingan terhadap metode pembobotan baru. Pada metode ini, perhitungan bobot *term t* dalam sebuah dokumen dilakukan dengan mengalikan nilai *Term Frequency* dengan *Inverse Document Frequency*. TF murni (*raw tf*) merupakan nilai *tf* diberikan berdasarkan jumlah kemunculan suatu kata pada dokumen. *Inverse Document Frequency* (*idf*) dihitung dengan menggunakan formula (Mandala, 2004).

$$idf_j = \log \frac{D}{df_j} \quad (1)$$

Dimana

D adalah jumlah semua dokumen dalam koleksi

df_j adalah jumlah dokumen yang mengandung *term t_j*

Jenis formula yang akan digunakan untuk perhitungan *term frequency* (*tf*) yaitu *tf* murni (*raw tf*). Dengan demikian rumus umum untuk TF-IDF adalah penggabungan dari formula perhitungan *raw tf* dengan formula *idf* dengan cara mengalikan nilai *term frequency* (*tf*) dengan nilai *inverse document frequency* (*idf*) (Robertson, 2004).

$$W_{ij} = tf_{ij} * idf_j \quad (2)$$

Keterangan :

W_{ij} bobot dokumen ke-d terhadap *term* ke-t

- tf_{ij} adalah jumlah kemunculan *term* t_j dalam dokumen d_i
 D adalah jumlah semua yang ada dalaman *database*
 df_j adalah jumlah dokumen yang mengandung *term* t_j

2.6 Cosine Similarity

Menurut Manning, Raghavan, dan Schutze (2008), *cosine similarity* digunakan untuk mengukur kedekatan antara dua vektor. *Cosine similarity* merupakan hasil *dot product* kedua vektor tersebut yang dinormalisasikan dengan dibagi dengan *Euclidean Distance* antara kedua vektor tersebut. *Cosine Similarity* 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 penentuan nilai yang diperoleh dari *Cosine Similarity* (Firdaus, 2014).

Berikut merupakan rumus untuk *cosine similarity* (Manning et al, 2008)

$$Sim(d_j, q) = \frac{d_j \cdot q}{||d_j|| ||q||} \quad (3)$$

$$Sim(d_i, q) = \frac{\sum_{i=1}^N w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^N w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^N w_{i,q}^2}} \quad (4)$$

Keterangan :

- d_j bobot dokumen j
 q_N *query* dokumen
 $\sum_{i=1}^N w_{i,j}$ jumlah bobot kata i pada dokumen j
 $\sum_{i=1}^N w_{i,q}$ jumlah bobot kata i pada dokumen *query*

Contoh perhitungan dari *cosine similarity* adalah sebagai berikut :

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)

$$Cosim = \frac{(0,707 * 0,578) * (0,707 * 0,578) * (0 * 0,578) * (0 * 0)}{\sqrt{(0,707^2 + 0,707^2 + 0^2 + 0^2) * (0,578^2 + 0,578^2 + 0,578^2 + 0^2)}}$$

$Cosim = 0,166$

Cosim (Dokumen 2, Query)

$$Cosim = \frac{(0,707 * 0) * (0,707 * 0) * (0 * 0) * (0 * 0,548)}{\sqrt{(0,707^2 + 0,707^2 + 0^2 + 0^2) * (0^2 + 0^2 + 0^2 + 0,548^2)}}$$

$Cosim = 0$

Hasil : mengembalikan dokumen 1

2.7 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). Analisis kebutuhan dilakukan dengan mengumpulkan masalah yang terjadi di lapangan. Masalah dikumpulkan dengan cara berkomunikasi dengan *user* sistem. Beberapa masalah yang ditemukan adalah sebagai berikut:

- a. Rumit dan lamanya proses pengaduan
- b. Keterbatasan *technical support*

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

- a. Sistem yang mampu memproses pengaduan sehingga lebih mudah dan cepat
- b. Sistem yang mampu memberikan solusi kerusakan komputer berdasarkan pengaduan yang diinput oleh *user*

2. *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.

3. *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*.

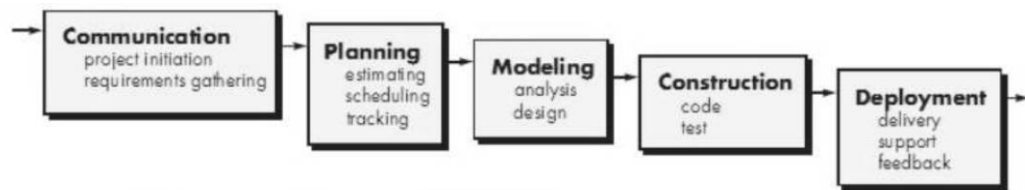
4. *Construction (Code & Test)*

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 kemampuan maksimum sistem.

5. *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

3.1 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem dilakukan dengan mencari informasi mengenai kebutuhan sistem dan data yang akan dibangun. Pada pencarian informasi, dilakukan dengan mewawancarai 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 <i>online</i> dengan fitur form pengaduan
2	Sistem dapat mengirimkan pengaduan berupa surat melalui <i>email</i> dengan fitur kirim pengaduan
3	<i>Technical support</i> 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 <i>Cosine Similarity</i>

3.1.1 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, <i>technical support</i> , admin, kepala dinas	Sistem dapat melakukan proses login dan logout dengan menggunakan <i>username</i> dan <i>password</i> yang dimiliki
KF2	Pegawai	Sistem dapat melakukan pengaduan secara <i>online</i> dengan menyediakan form pengaduan
KF3	Pegawai	Pegawai mendapatkan informasi mengenai <i>history</i> pengaduan yang pernah dilakukan
KF4	Pegawai	Pegawai mendapatkan informasi mengenai status pengaduan, apakah pengaduan belum diproses atau sudah diproses
KF5	<i>Technical support</i>	<i>Technical support</i> mendapatkan solusi kerusakan komputer berdasarkan data pengaduan yang diinput oleh pegawai pada halaman list pengaduan
KF6	<i>Technical support</i>	<p><i>Technical support</i> dapat mengelola daftar kerusakan dengan rincian :</p> <ul style="list-style-type: none"> a. <i>Technical support</i> dapat menambah kerusakan b. <i>Technical support</i> dapat melihat daftar kerusakan c. <i>Technical support</i> dapat mengedit kerusakan d. <i>Technical support</i> dapat menghapus kerusakan

Tabel 3. 2 Kebutuhan Fungsional Sistem (lanjutan)

KF7	<i>Technical support</i>	<p><i>Technical support</i> dapat mengelola list solusi dengan rincian :</p> <ul style="list-style-type: none"> a. <i>Technical support</i> dapat menambah solusi b. <i>Technical support</i> dapat melihat list solusi c. <i>Technical support</i> dapat mengedit solusi d. <i>Technical support</i> dapat menghapus solusi
KF8	Admin	<p>Admin dapat mengelola data pengaduan dengan rincian :</p> <ul style="list-style-type: none"> a. Admin dapat menambah pengaduan b. Admin dapat melihat informasi pengaduan c. Admin dapat melakukan <i>update</i> pengaduan dengan cara mengedit pengaduan d. Admin dapat melakukan hapus pengaduan e. Admin dapat mengirim pengaduan berupa surat melalui <i>email</i> f. Admin dapat mencetak pengaduan g. Admin dapat <i>search</i> pengaduan h. Admin dapat melihat grafik pengaduan
KF9	Admin	<p>Admin dapat mengelola daftar user dengan rincian :</p> <ul style="list-style-type: none"> a. Admin dapat menambah user b. Admin dapat melihat daftar user c. Admin dapat mengedit user d. Admin dapat menghapus user

Tabel 3. 2 Kebutuhan Fungsional Sistem (lanjutan)

KF10	Admin	Admin dapat mengelola daftar instansi dengan rincian : <ul style="list-style-type: none"> a. Admin dapat menambah instansi b. Admin dapat melihat daftar instansi c. Admin dapat mengedit instansi d. Admin dapat menghapus instansi
KF11	Kepala dinas	Kepala dinas mendapatkan laporan rekap pengaduan dengan rincian : <ul style="list-style-type: none"> a. Kepala dinas mendapatkan informasi pengaduan berdasarkan tahun dan bulan yang dipilih dan kepala dinas dapat mencetak pengaduan berdasarkan tahun dan bulan yang dipilih b. Kepala dinas dapat melihat grafik pengaduan berdasarkan instansi dinas, kategori kerusakan, dan status pengaduan
KF12	<i>Technical support</i> , admin, kepala dinas	Sistem dapat mengelola profile dengan rincian: <ul style="list-style-type: none"> a. Sistem dapat mengubah nama profile b. Sistem dapat mengubah alamat email profile c. Sistem dapat mengganti foto profile

3.1.2 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	<i>User Friendly</i>	Tampilan antar muka sistem harus mudah dimengerti oleh pegawai, <i>technical support</i> , 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.

3.2 Perancangan Sistem

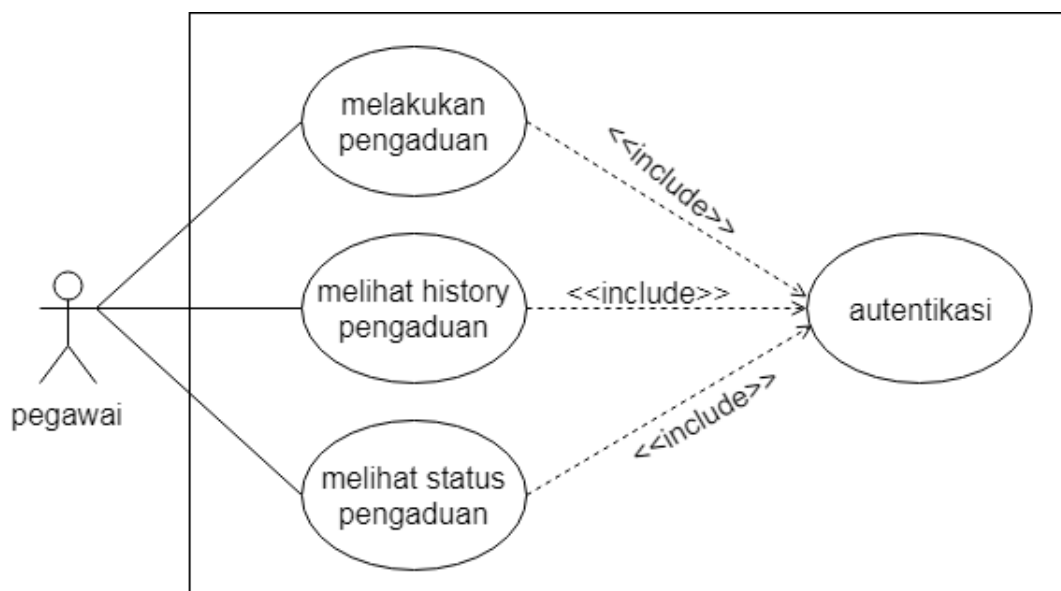
Sistem pengaduan dibangun dengan berbasis berorientasi objek sehingga sistem dirancang menggunakan diagram UML (*Unified Modeling Language*) antara lain *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*. Pada penelitian ini untuk diagram UML dengan menggunakan bantuan aplikasi draw.io

3.2.1 *Use Case Diagram*

Usecase diagram menampilkan hubungan antara aktor (pelaku) dengan hal-hal yang terlibat pada sistem. Definisi aktor pada *use case diagram* yang terdapat pada gambar 3.1, gambar 3.2, gambar 3.3, dan gambar 3.4 akan di jelaskan pada tabel 3.4.

Tabel 3. 4 Definisi Aktor Pada *Use Case Diagram*

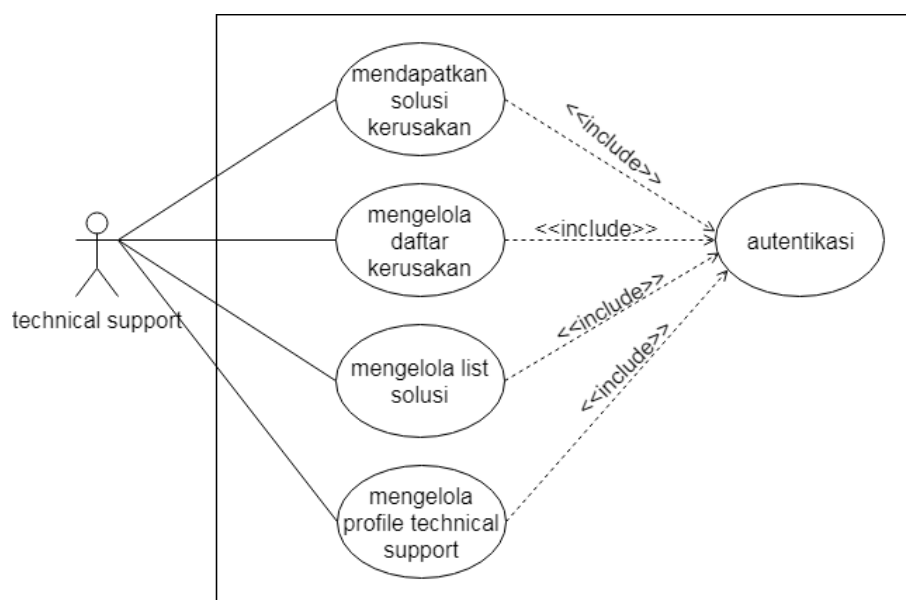
No	Aktor	Deskripsi
1	Pegawai	Orang yang akan melakukan pengaduan pada sistem pengaduan
2	<i>Technical Support</i>	Orang yang akan mendapatkan solusi kerusakan dari pengaduan yang dilakukan oleh pegawai pada sistem pengaduan
3	Admin	Orang yang bertugas mengelola data pengaduan yang terdapat pada sistem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

**Gambar 3. 1 *Use Case Diagram* Pegawai dalam Sistem 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	<i>Use Case</i>	Deskripsi
1	Autentikasi	Proses pengecekan dan validasi oleh sistem kepada pengguna sistem
2	Melakukan pengaduan	Proses untuk melakukan pengaduan pada sistem
3	Melihat <i>history</i> pengaduan	Proses untuk melihat <i>history</i> pengaduan yang ada pada sistem
4	Melihat status pengaduan	Proses untuk melihat status pengaduan yang ada pada sistem.

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

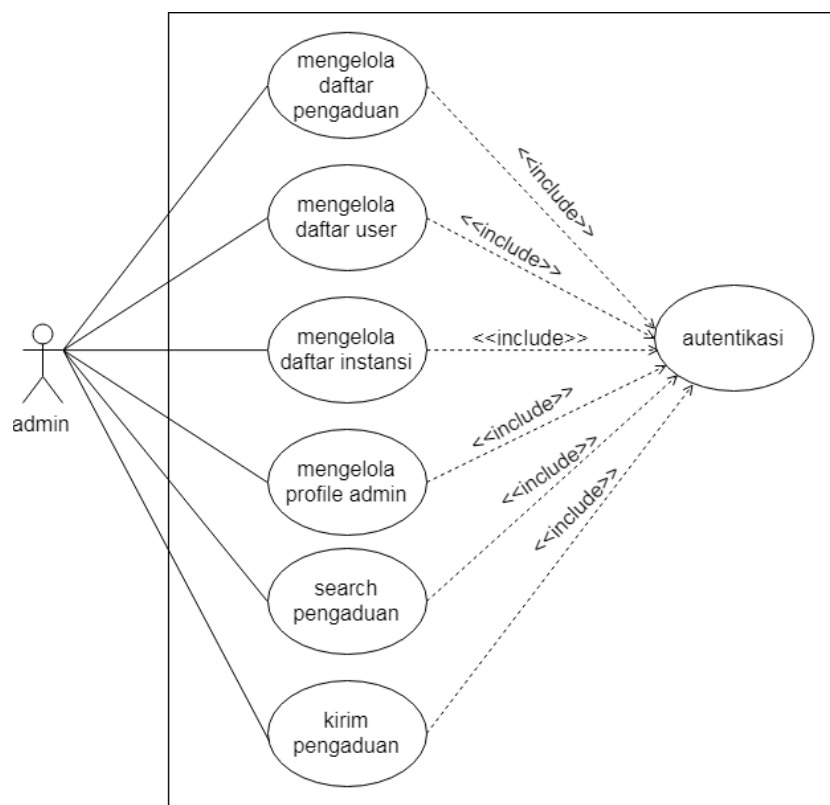
Definisi Use Case pada Use Case Diagram *Technical Support* Sistem Pengaduan ditujukan pada Tabel 3.6.

Tabel 3. 6 Definisi *Use Case* pada *Use Case Diagram* Technical Support

No	<i>Use Case</i>	Deskripsi
1	Autentikasi	Proses pengecekan dan validasi oleh sistem kepada pengguna sistem

Tabel 3. 6 Definisi *Use Case* pada *Use Case Diagram Technical Support* (lanjutan)

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 <i>technical support</i>	Proses untuk mengelola profile <i>technical support</i> pada sistem pengaduan seperti mengganti nama, email, dan foto <i>profile</i>

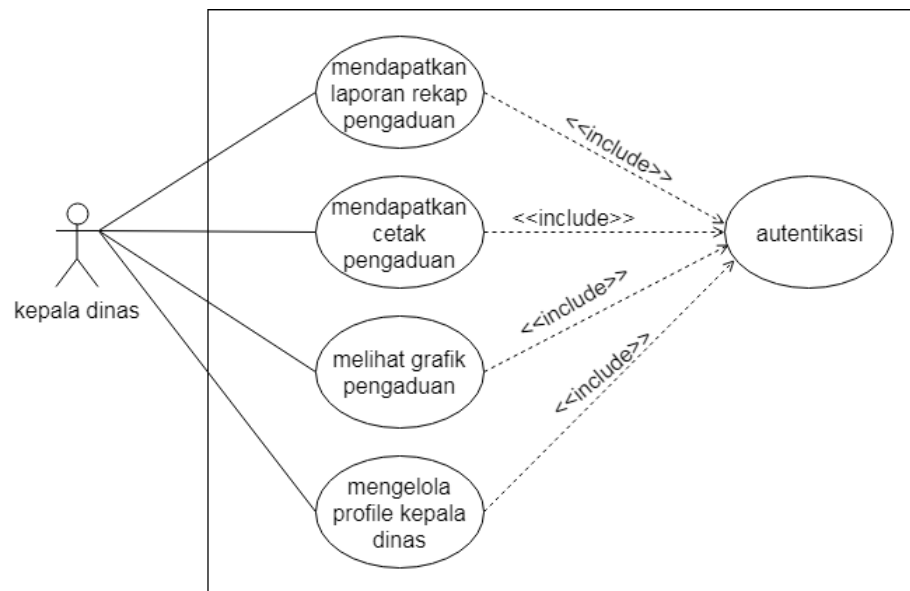


Gambar 3. 3 *Use Case Diagram* Admin dalam Sistem Pengaduan

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	<i>Use Case</i>	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 data pengaduan, edit data pengaduan, dan hapus data pengaduan
3	Mengelola daftar user	Proses untuk mengelola daftar user yang terdapat pada sistem pengaduan seperti tambah data pengaduan, edit data pengaduan, dan hapus data
4	Mengelola daftar instansi	Proses untuk mengelola daftar instansi yang terdapat pada sistem pengaduan seperti tambah data pengaduan, edit data pengaduan, dan hapus data pengaduan
5	Mengelola profile admin	Proses untuk mengelola profile admin seperti mengganti nama, email, dan foto <i>profile</i> di bagian <i>user panel</i> yang terdapat pada sistem pengaduan
6	Search pengaduan	Proses untuk mencari pengaduan berdasarkan nomor surat atau instansi yang ada pada sistem pengaduan



Gambar 3. 4 Use Case Diagram Kepala Dinas dalam Sistem Pengaduan

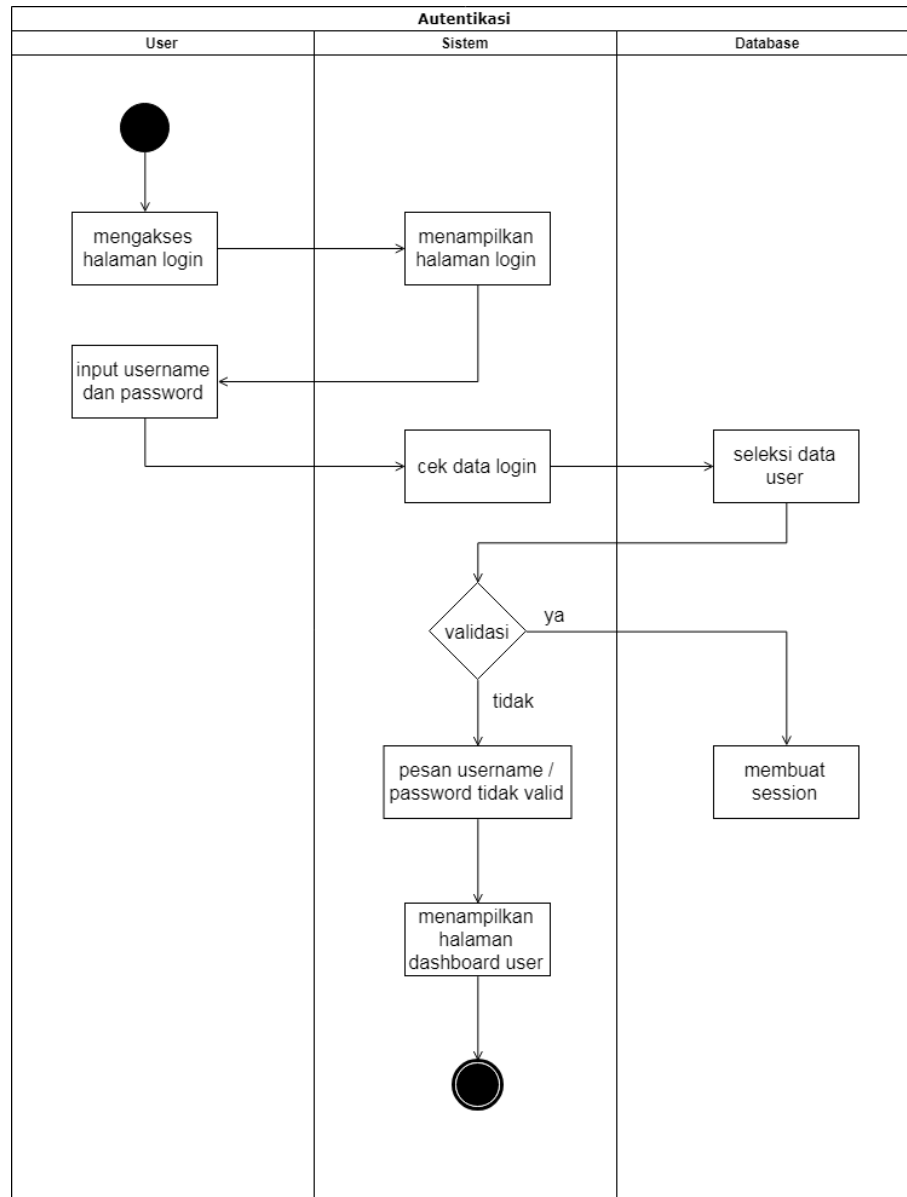
Definisi Use Case pada Use Case Diagram Kepala Dinas Sistem Pengaduan ditujukan pada Tabel 3.8.

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 <i>profile</i>

3.2.2 Activity Diagram

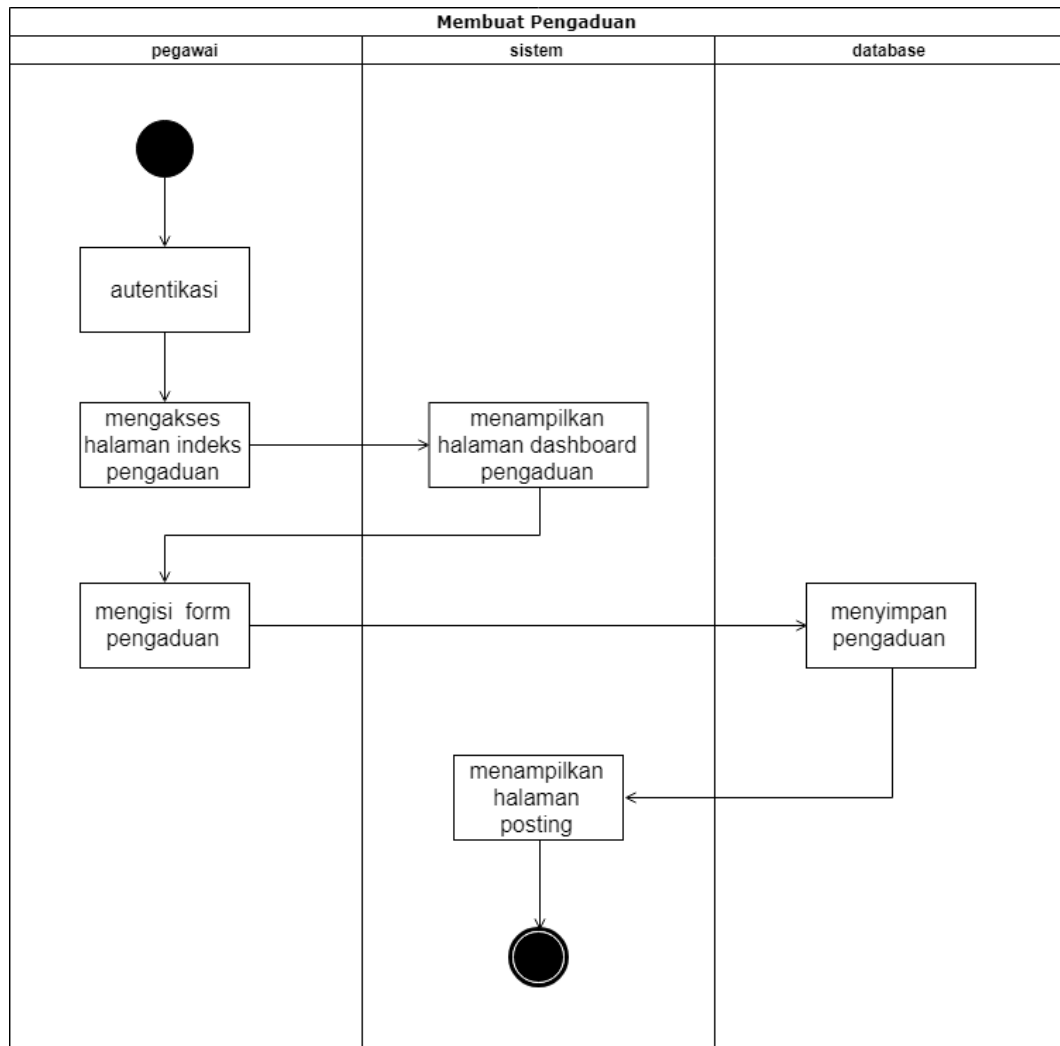
Activity diagram akan menjabarkan aktifitas yang terdapat pada *use case* pada setiap *use case diagram*.



Gambar 3. 5 Activity Diagram Autentikasi

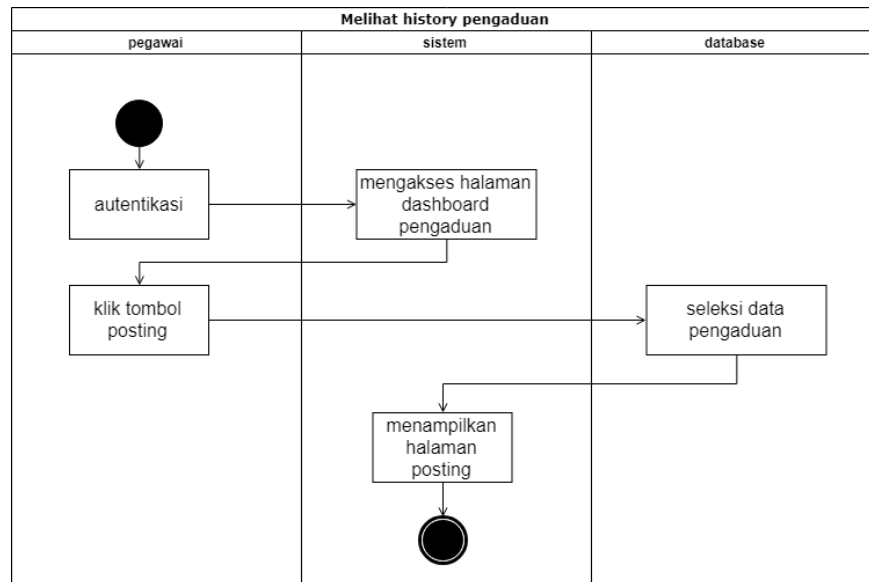
Gambar 3.5 menjelaskan alur aktivitas yang terdapat pada proses Autentikasi. Pertama pengguna sistem memasukkan *username* dan *password*. Jika data login yang terdapat pada *database* dan *username* dengan *password* sesuai, maka sistem akan membuat *session* yang dimiliki oleh pengguna yaitu sebagai

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.



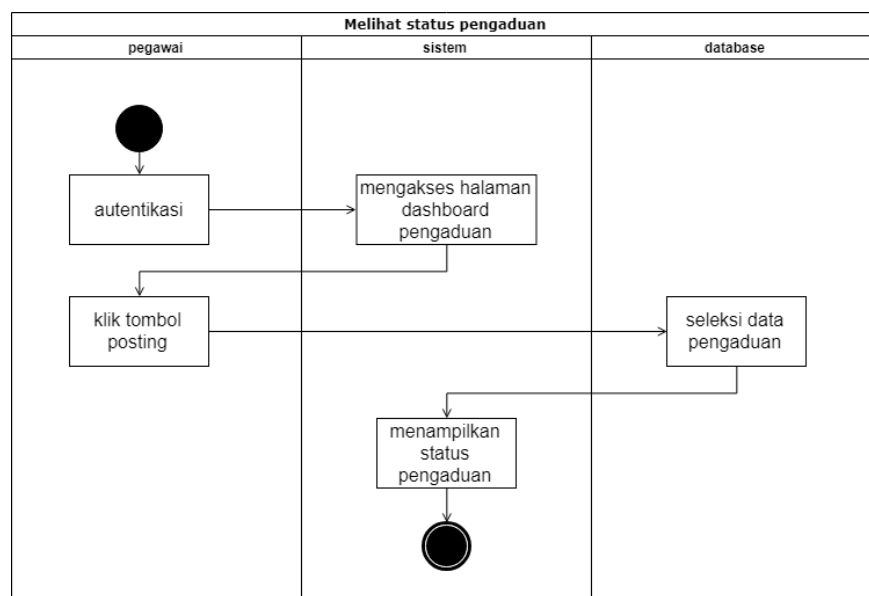
Gambar 3. 6 Activity Diagram Melakukan Pengaduan

Gambar 3.6 menjelaskan alur aktivitas yang terdapat pada proses melakukan pengaduan. 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.



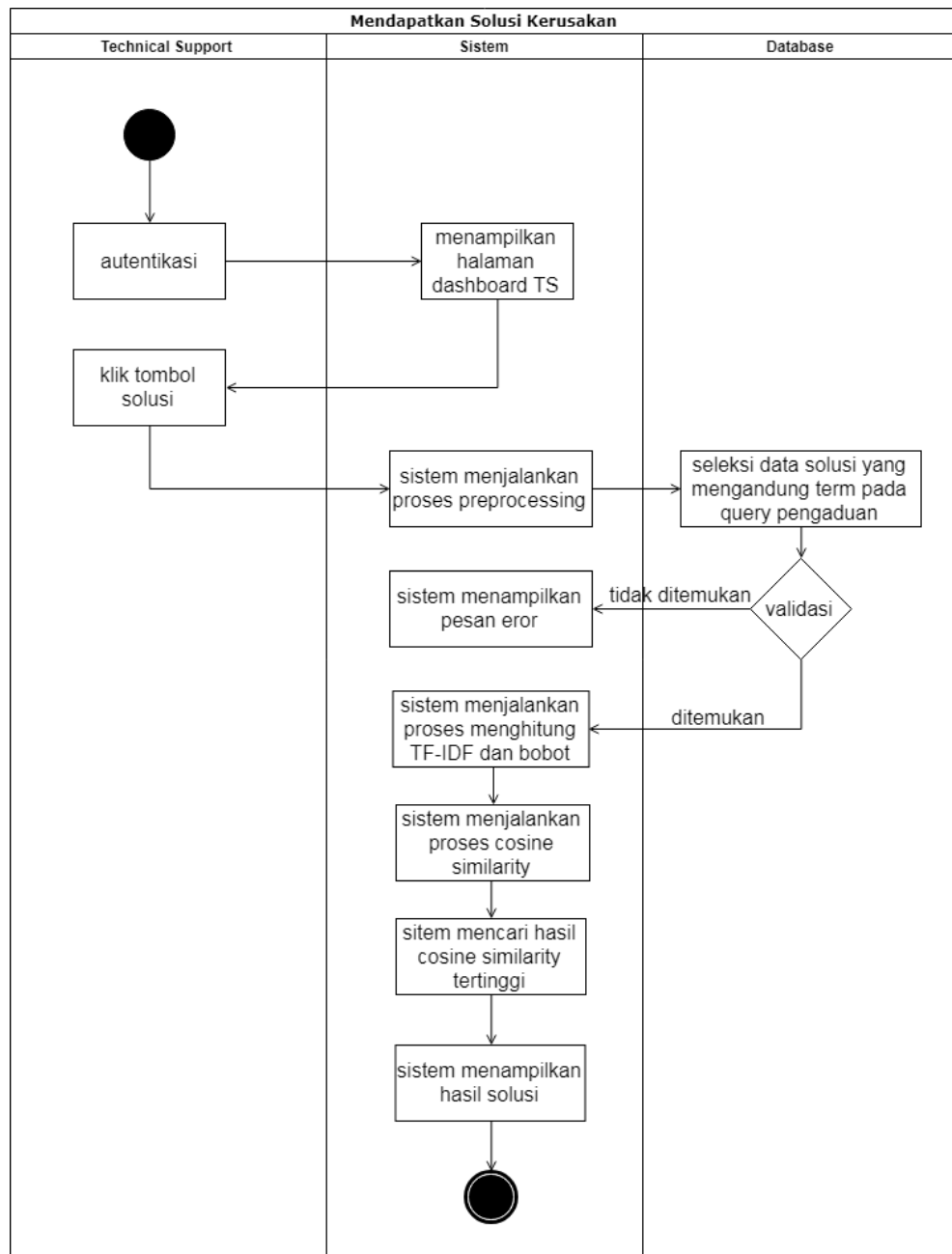
Gambar 3.7 Activity Diagram Melihat History Pengaduan

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.



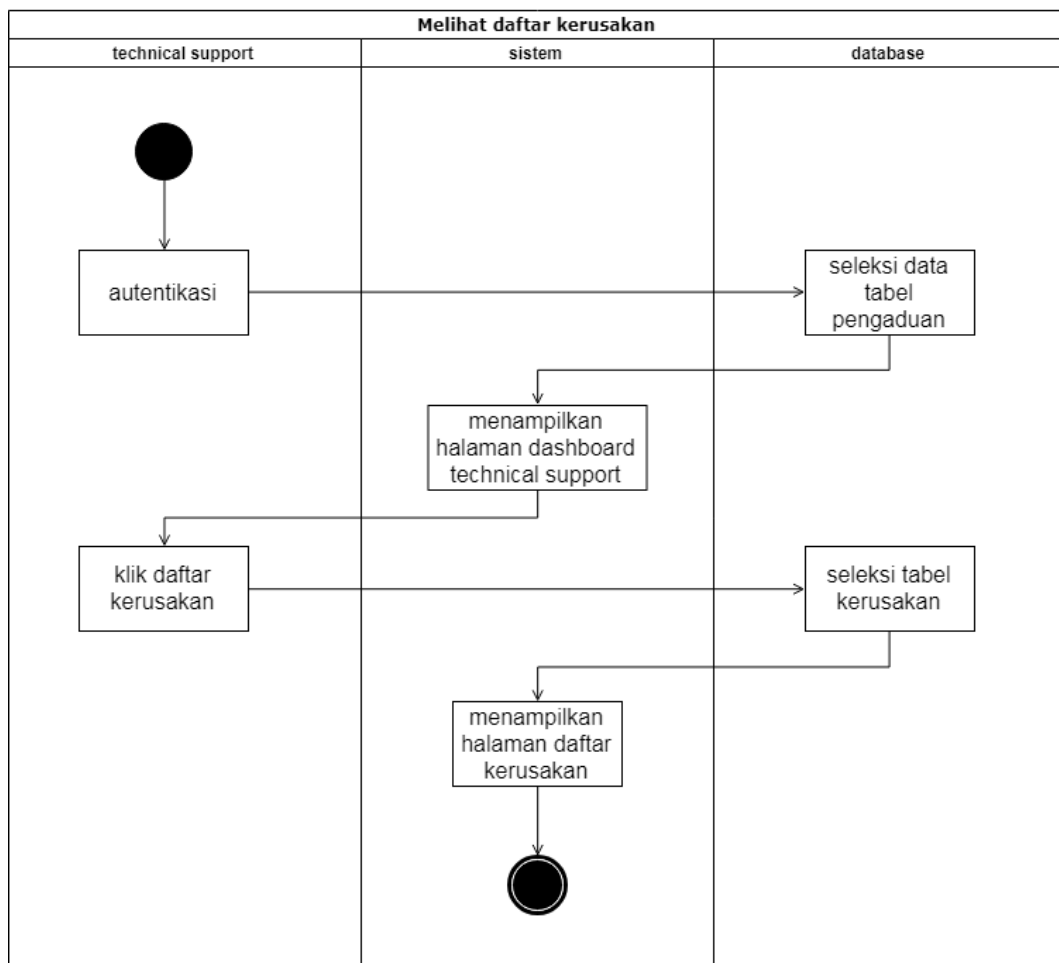
Gambar 3.8 Activity Diagram Melihat Status Pengaduan

Gambar 3.8 menjelaskan alur aktivitas yang terdapat pada proses melihat status pengaduan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian mengakses halaman dashboard pengaduan. Klik tombol posting yang terdapat pada bagian navigasi. Database menyeleksi data pengaduan, kemudian sistem menampilkan status pengaduan yang terdapat pada halaman posting.



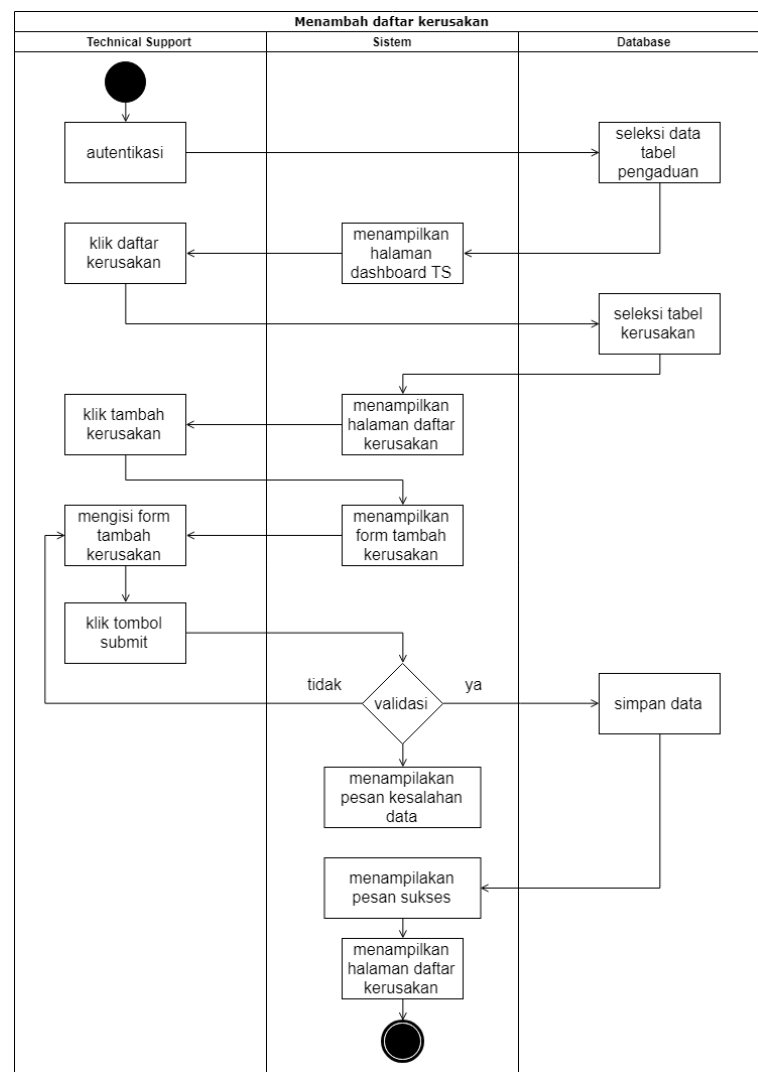
Gambar 3. 9 Activity Diagram Mendapatkan Solusi Kerusakan

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 memunculkan pesan error, dan bila ditemukan maka sistem menjalankan proses menghitung TF-IDF dan bobot, proses *cosine similarity*, dan pencarian hasil *cosine similarity* tertinggi. Setelah semua proses dijalankan kemudian sistem menampilkan hasil solusi kerusakan berdasarkan hasil *cosine* tertinggi.



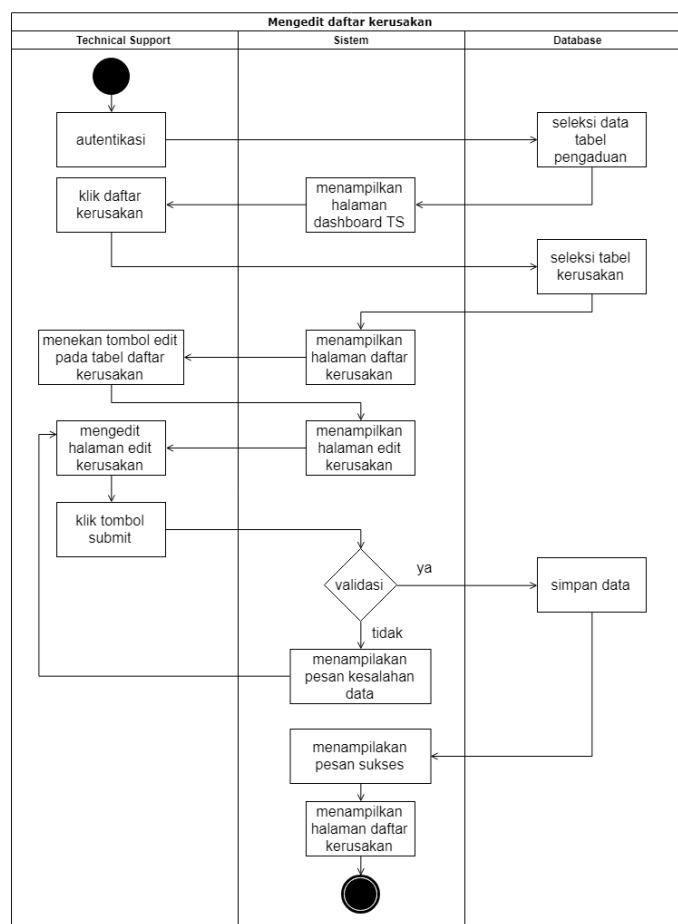
Gambar 3. 10 Activity Diagram Melihat Daftar Kerusakan

Gambar 3.10 menjelaskan alur aktivitas yang terdapat pada proses melihat daftar kerusakan. Pertama dilakukan proses autentikasi, lalu database menyeleksi data tabel pengaduan. Sistem menampilkan halaman dashboard. *Technical support* klik daftar kerusakan. Database menyeleksi tabel kerusakan, kemudian sistem menampilkan halaman daftar kerusakan. *Activity diagram* pada *use case* lihat list solusi, lihat data pengaduan, lihat daftar user, dan lihat daftar instansi memiliki karakteristik yang sama dengan *activity diagram* melihat daftar kerusakan. Sehingga hanya dibuat *activity diagram* untuk *usecase* melihat daftar kerusakan saja.



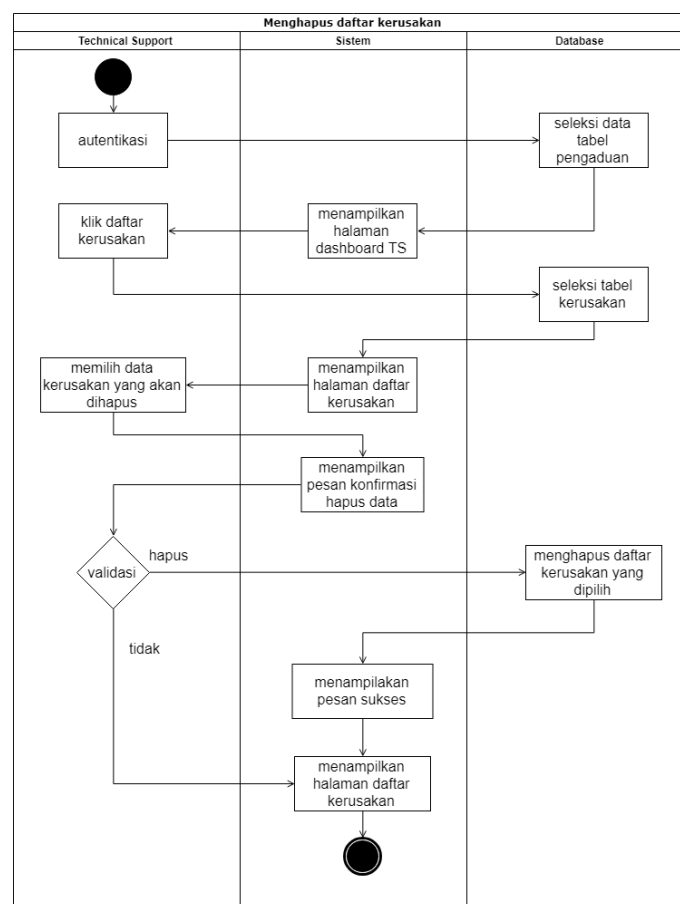
Gambar 3. 11 Activity Diagram Menambah Daftar Kerusakan

Gambar 3.11 menjelaskan alur aktivitas yang terdapat pada proses menambah daftar kerusakan. Pertama dilakukan proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan halaman dashboard. *Technical support* klik daftar kerusakan. 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 menampilkan pesan kesalahan data dan apabila data valid maka data akan disimpan. *Activity diagram* pada *use case* tambah list solusi, tambah data pengaduan, tambah daftar user, dan tambah daftar instansi memiliki karakteristik yang sama, sehingga hanya dibuat *activity diagram* untuk *use case* tambah daftar kerusakan saja.



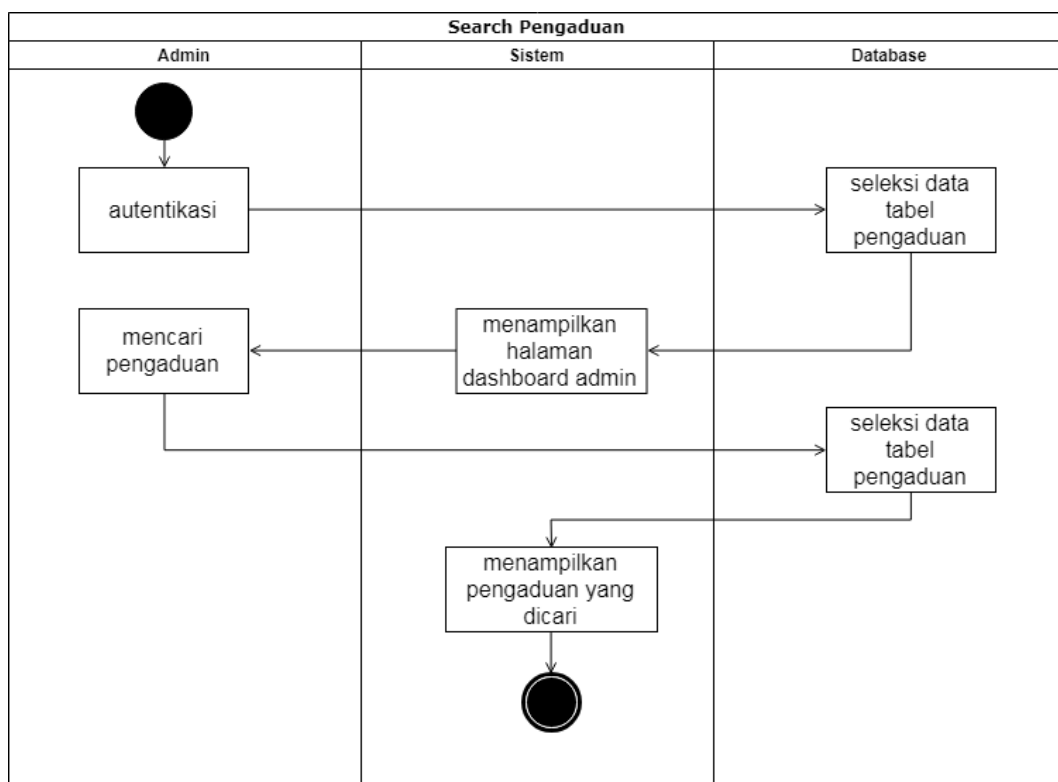
Gambar 3. 12 Activity Diagram Edit Daftar Kerusakan

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. 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, edit data pengaduan, edit daftar user, dan edit daftar instansi memiliki karakteristik yang sama dengan *activity diagram* edit daftar kerusakan. Sehingga hanya dibuat *activity diagram* untuk *use case* edit daftar kerusakan saja.



Gambar 3. 13 Activity Diagram Hapus Daftar Kerusakan

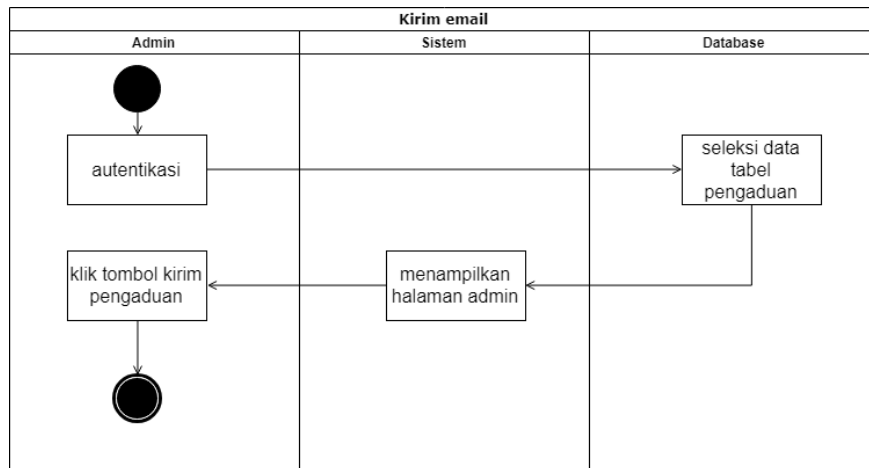
Gambar 3.13 menjelaskan alur aktivitas yang terdapat pada proses menghapus daftar kerusakan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan halaman dashboard *technical support*. Klik daftar kerusakan yang terdapat pada 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

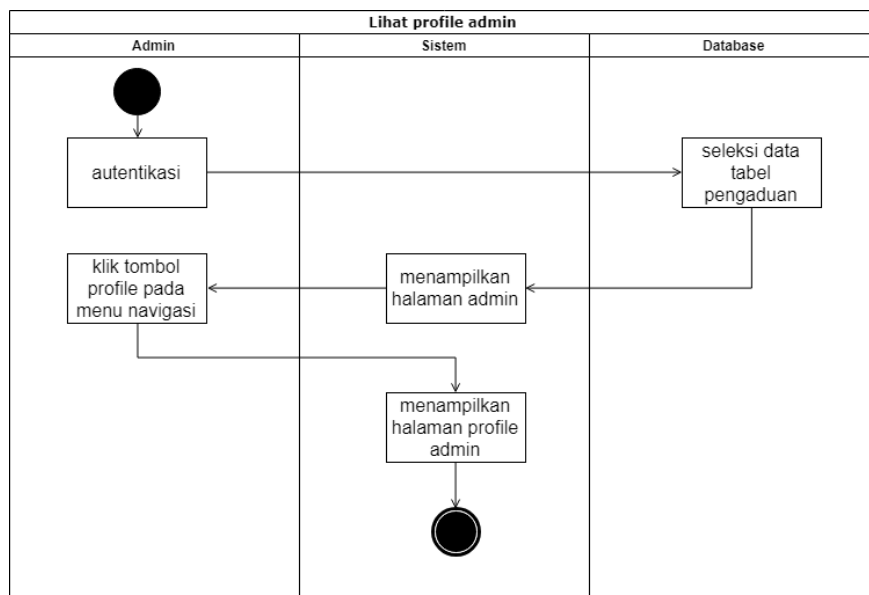
Gambar 3.14 menjelaskan alur aktivitas yang terdapat pada proses *search* data pengaduan. Hal pertama yang dilakukan adalah melalui proses autentikasi, kemudian database menyeleksi data tabel pengaduan. Sistem menampilkan

halaman admin. Admin mencari pengaduan dengan mengetikkan nama instansi atau nomor surat. Database menyeleksi data pada tabel pengaduan, setelah itu sistem menampilkan pengaduan yang dicari.



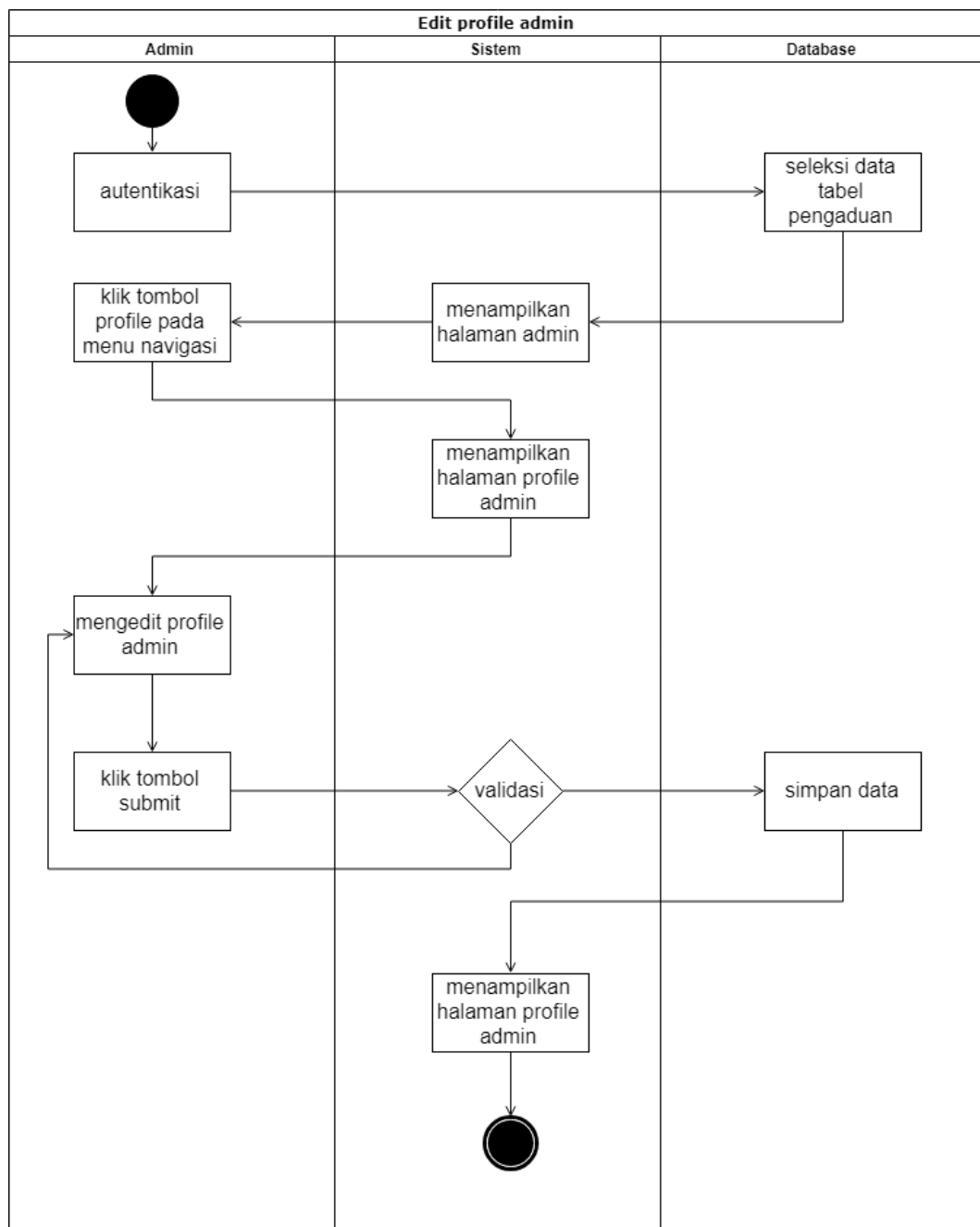
Gambar 3. 15 Activity Diagram Kirim Email Data Pengaduan

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.



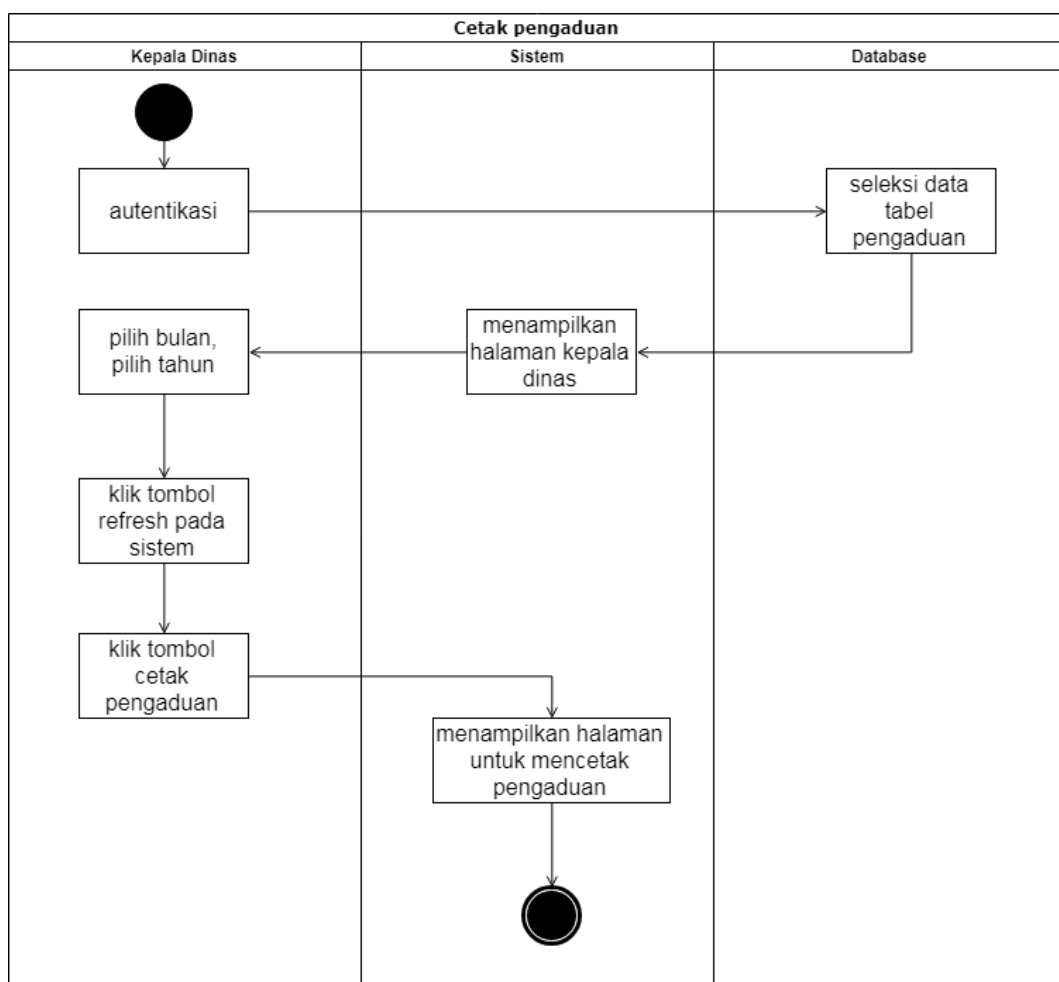
Gambar 3. 16 Activity Diagram Melihat Profile Admin

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. Setelah itu admin mengedit profile admin. Setelah itu admin klik tombol submit. Sistem melakukan validasi. Jika validasi berhasil, sistem menyimpan data ke database. Setelah itu sistem menampilkan halaman profile admin kembali. Jika validasi gagal, sistem menampilkan halaman profile admin kembali.



Gambar 3. 17 Activity Diagram Mengedit Profile Admin

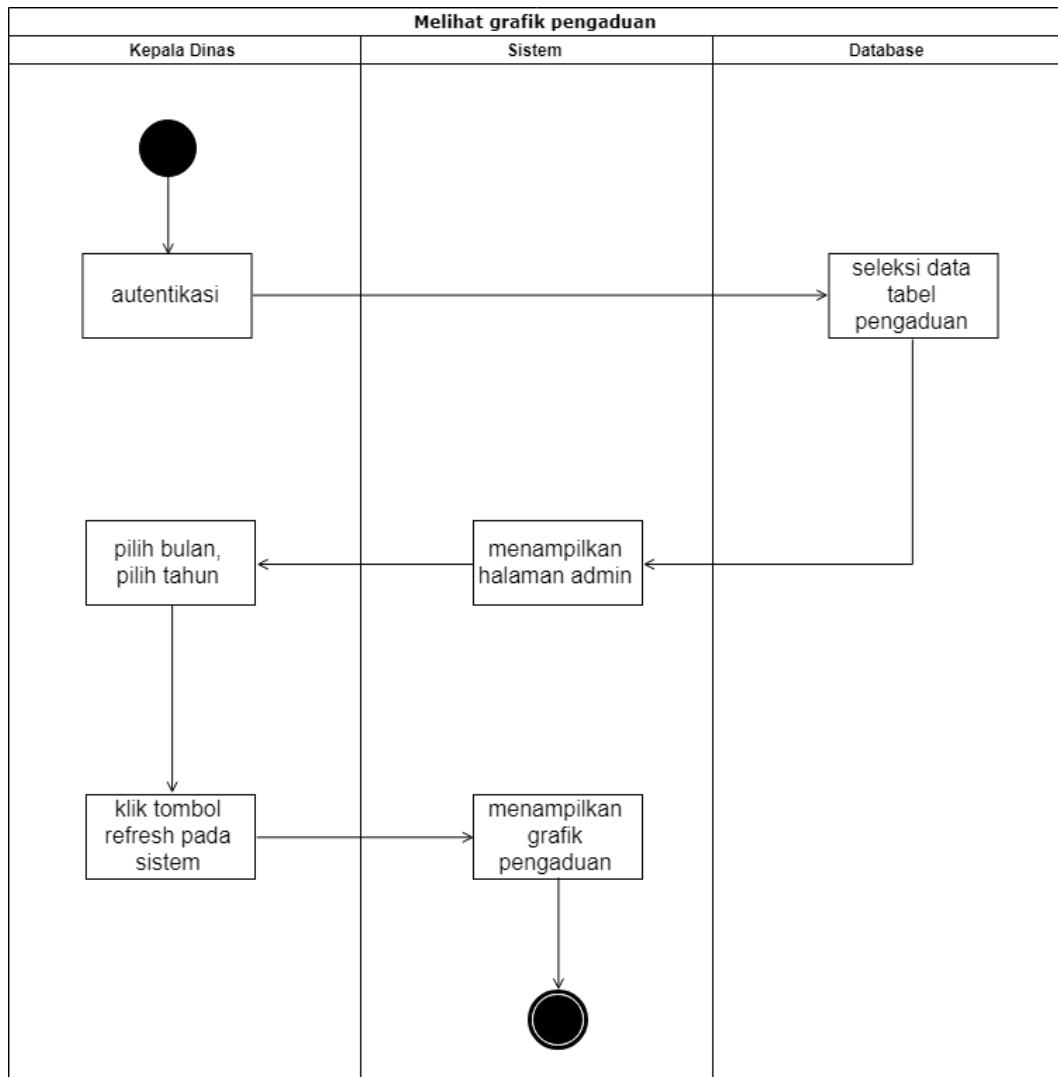
Gambar 3.17 menjelaskan alur aktivitas yang terdapat pada proses mengedit profile admin. Hal pertama yang dilakukan adalah proses autentikasi. Kemudian database menyeleksi data pada tabel pengaduan. Sistem menampilkan halaman admin. Setelah itu admin klik tombol profile pada menu navigasi. Sistem menampilkan 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,

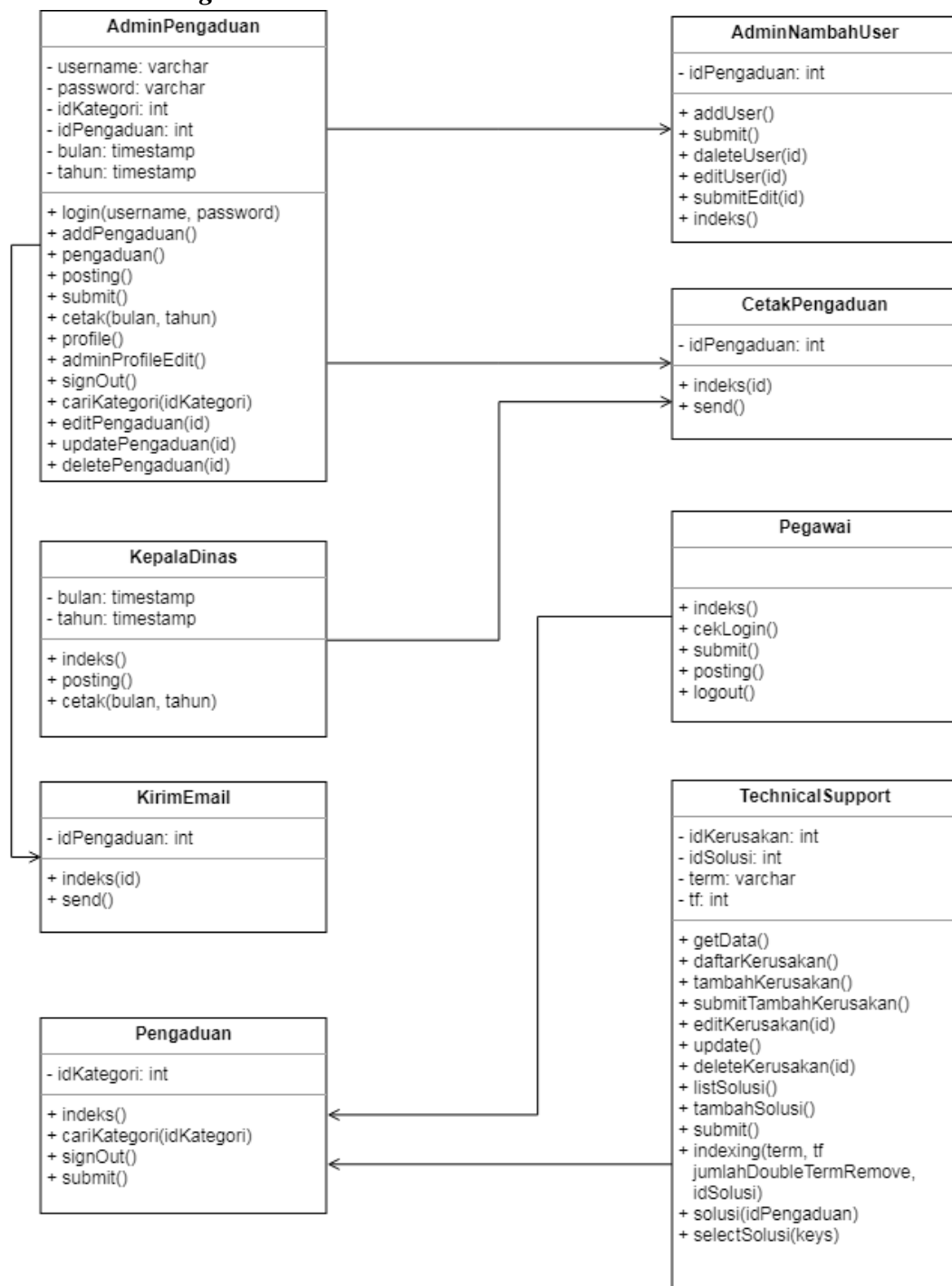
lalu klik tombol refresh pada sistem. Setelah itu kepala dinas klik tombol cetak pengaduan. Kemudian sistem menampilkan halaman untuk mencetak pengaduan.



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 menyeleksi 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 sistem. Sistem menampilkan grafik pengaduan.

3.2.3 Class Diagram



Gambar 3. 20 Class Diagram

Gambar 3.20 Merupakan 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

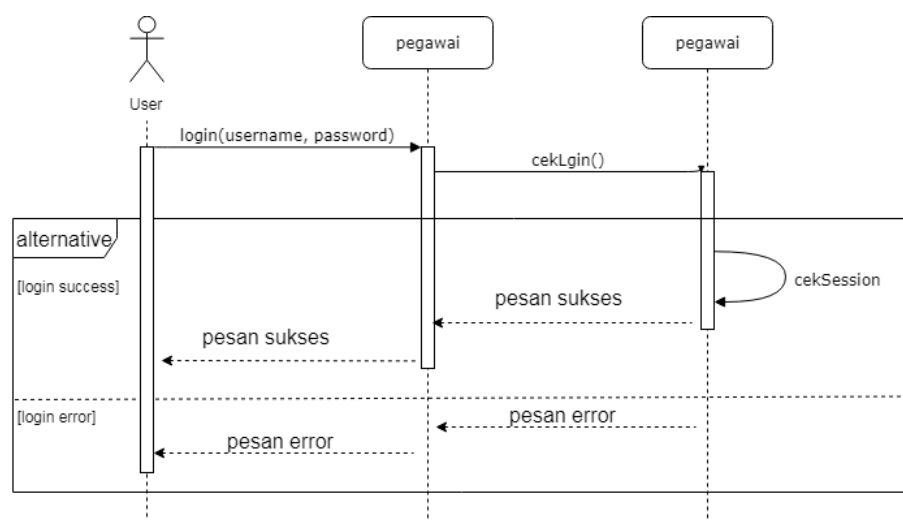
Kelas	<i>Use Case</i>	Fungsi
AdminPengaduan	Autentikasi	Menangani operasi pengelolaan dan akses data admin pada sistem
	Mengelola data pengaduan	
	Mengelola daftar user	
	Mengelola daftar instansi	
	Mengelola profile admin	
	<i>Search</i> pengaduan	
KepalaDinas	Autentikasi	Menangani proses rekap pengaduan, 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

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

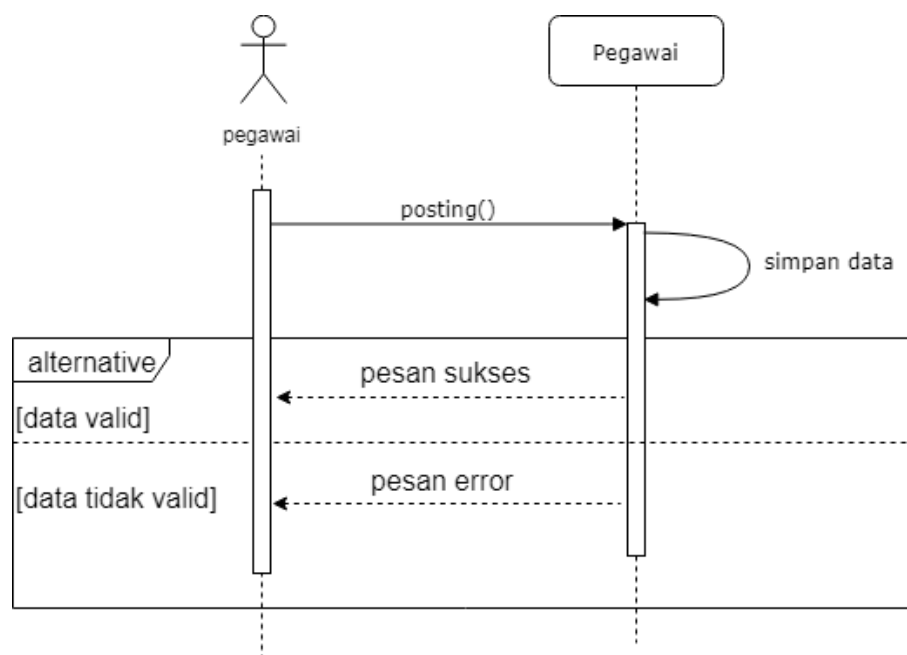
CetakPengaduan	Mendapatkan cetak pengaduan	Menangani proses cetak pengaduan yang dilakukan oleh kepala dinas
Pegawai	Autentikasi	Menangani proses lihat <i>history</i> pengaduan, lihat status pengaduan, dan akses data pegawai
	Melihat <i>history</i> pengaduan	
	Melihat status pengaduan	
TechnicalSupport	Autentikasi	Menangani proses mendapatkan solusi kerusakan, menangani proses operasi pengelolaan, dan akses data pegawai
	Mendapatkan solusi kerusakan	
	Mengelola daftar kerusakan	
	Mengelola list solusi	
	Mengelola profile <i>technical support</i>	

3.2.4 Sequence Diagram

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

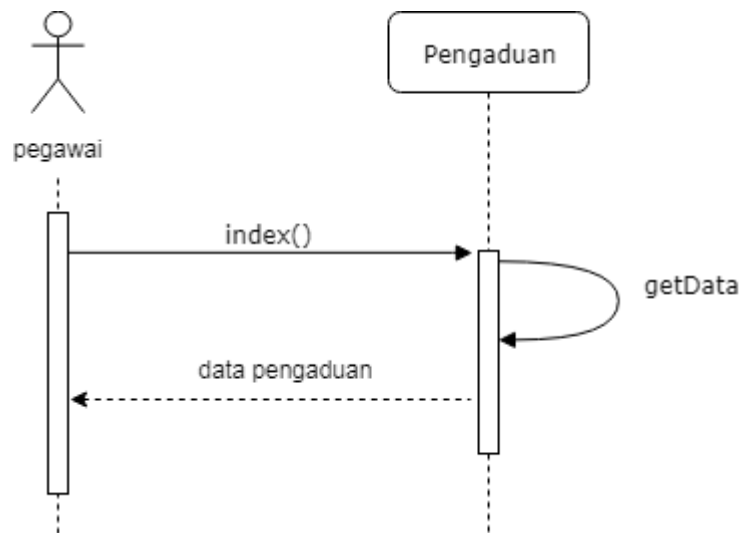
Gambar 3. 21 *Sequence Diagram* Autentikasi

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.



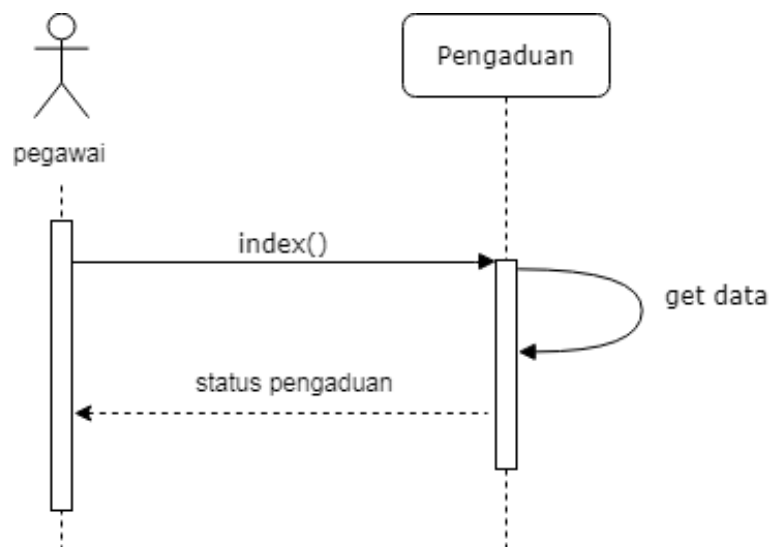
Gambar 3. 22 Sequence Diagram Melakukan Pengaduan

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 maka akan dikirim pesan error.



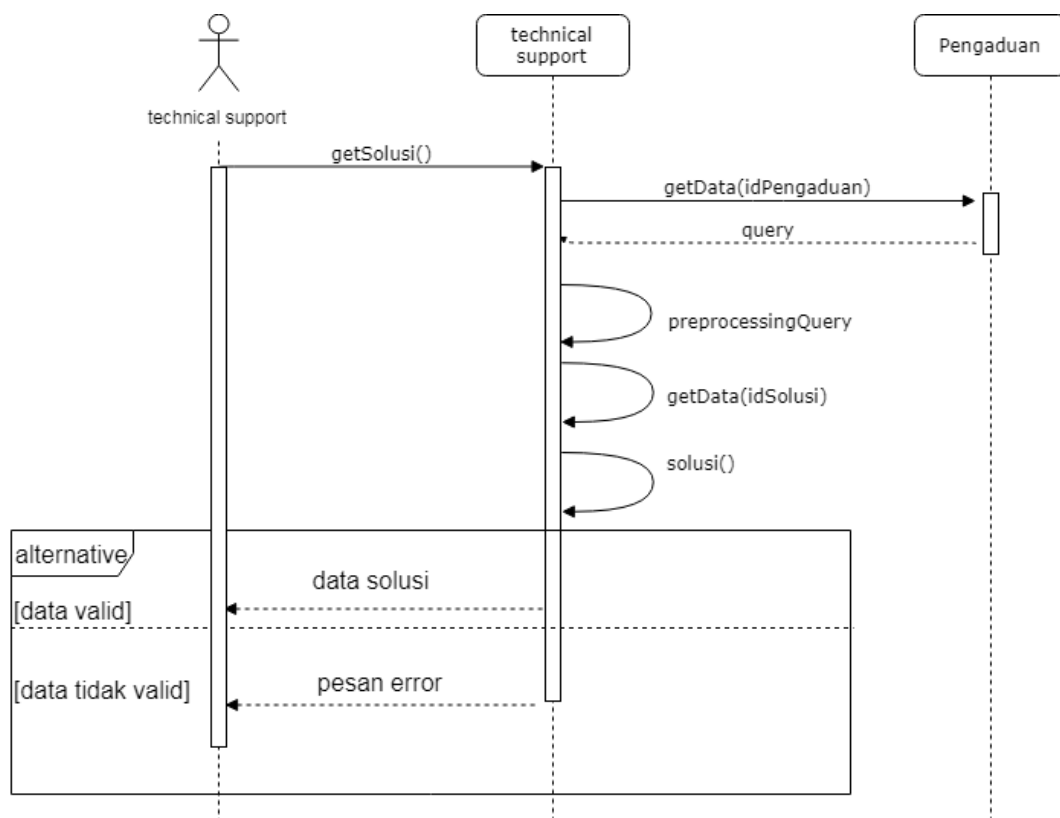
Gambar 3. 23 Sequence Diagram Melihat History Pengaduan

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 *query* “*getData*” yang akan mengambil seluruh data pengaduan yang ada pada database, selanjutnya data pengaduan ditampilkan.



Gambar 3. 24 Sequence Diagram Melihat Status Pengaduan

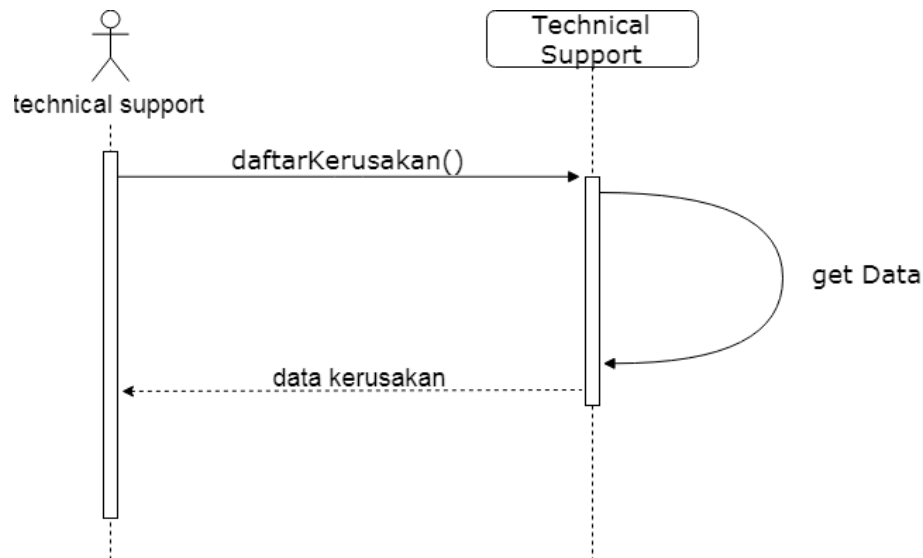
Sequence diagram pada Gambar 3.24 menjelaskan perilaku dan waktu yang digunakan *class* pada proses melihat status pengaduan. Ketika Pegawai ingin melihat status pengaduan maka akan dijalankan fungsi “*index()*” pada kelas “*Pengaduan()*”. Kemudian dijalankan fungsi “*getData()*” yang akan mengambil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



Gambar 3. 25 Sequence Diagram Mendapatkan Solusi Pengaduan

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()*”. Setelah itu dijalankan fungsi “*getData()*” dengan parameter *idPengaduan* pada kelas “*Pengaduan()*”. Kemudian sistem menjalankan *preprocessing* data pada *query* dengan menjalankan fungsi “*preprocessingQuery()*” yang akan menghasilkan data *listTerm* dari *query*. Kemudian sistem menjalankan fungsi “*solusi()*”. Dalam fungsi “*solusi()*” sistem

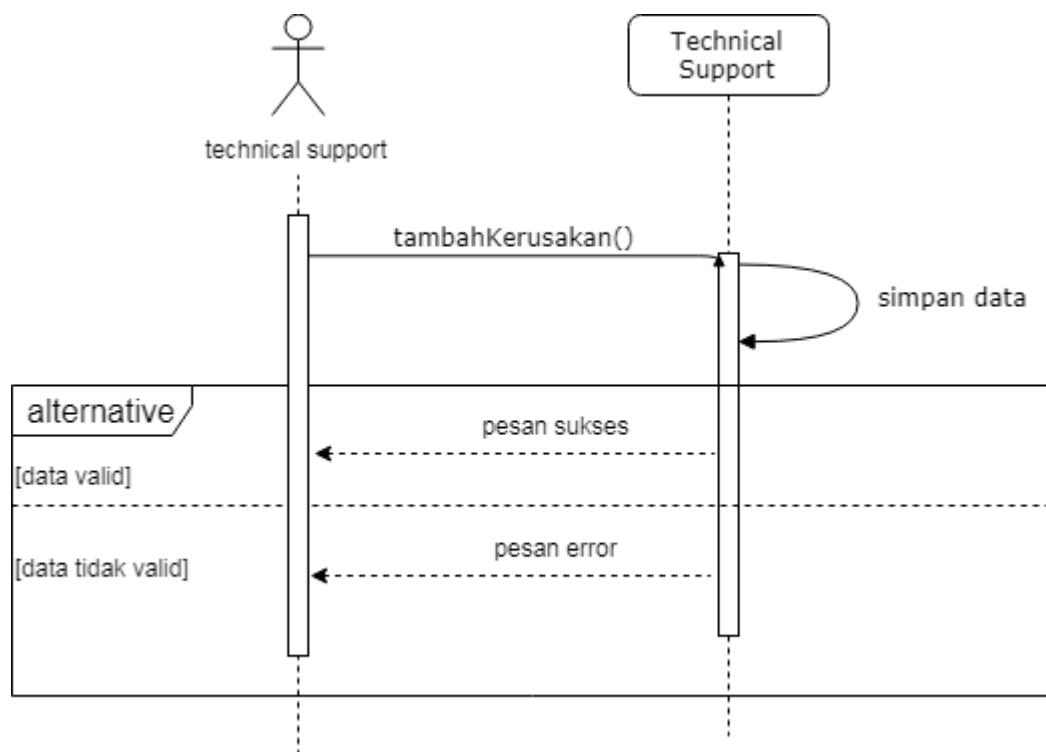
melakukan proses mencari data solusi pada tabel *indexing* yang mengandung *term query* dengan fungsi “*getTermSolusi(listTerm)*”. Setiap *term* solusi yang sesuai dengan *query* akan dilakukan perhitungan bobot solusi dan hitung algoritma *cosine similarity* melalui fungsi “*hasil(idSolusi)*”. Apabila data valid maka akan dikirim data solusi dan bila tidak valid maka 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 daftar kerusakan. Ketika *Technical Support* ingin melihat daftar kerusakan maka akan dijalankan fungsi “*daftarKerusakan()*” pada kelas “*TechnicalSupport()*”. Kemudian mengambil data kerusakan yang ada pada database, selanjutnya data kerusakan ditampilkan.

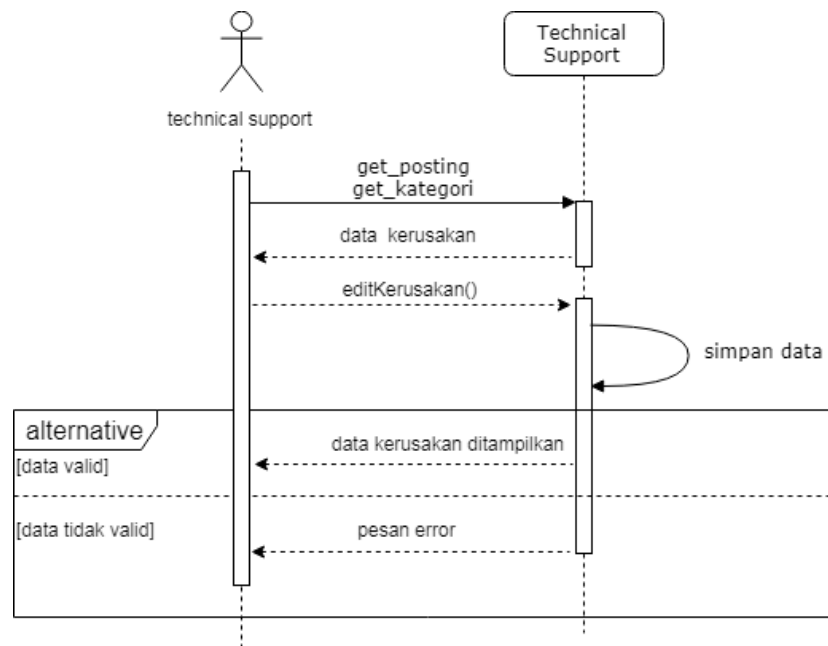
Sequence diagram pada *use case* lihat list solusi, *use case* lihat data pengaduan, *use case* lihat daftar user, dan *use case* lihat daftar instansi memiliki karakteristik yang sama dengan *sequence diagram* melihat daftar kerusakan. Sehingga hanya dibuat *sequence diagram* melihat daftar kerusakan 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 maka akan dikirim pesan error.

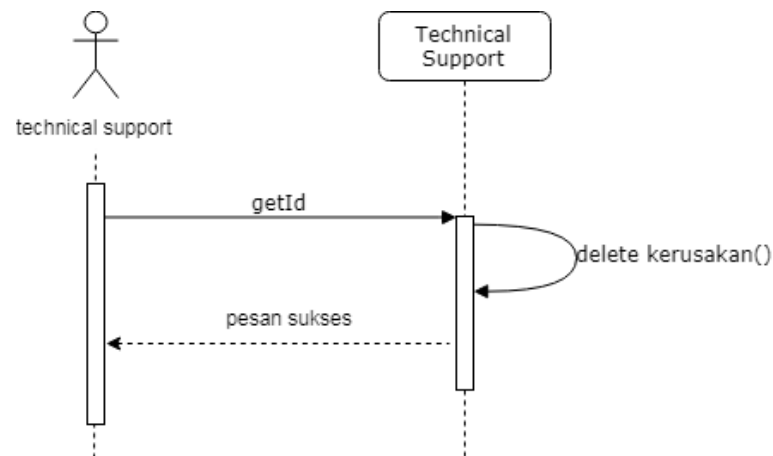
Sequence diagram pada *use case* tambah list solusi, *use case* tambah data pengaduan, *use case* tambah daftar user, dan *use case* tambah daftar instansi memiliki karakteristik yang sama dengan *sequence diagram* tambah daftar kerusakan. Sehingga hanya dibuat *sequence diagram* tambah daftar kerusakan 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 *query* “*get_posting*” dan “*get_kategori*” pada kelas “*TechnicalSupport()*” untuk mengambil data kerusakan dan kategori kerusakan berdasarkan parameter *idKerusakan* untuk ditampilkan pada form edit kerusakan.. Kemudian dilakukan pemanggilan fungsi “*editKerusakan()*” pada kelas “*TechnicalSupport()*” dengan parameter “*id*” kerusakan. Setelah data pada form telah diedit, kemudian *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.

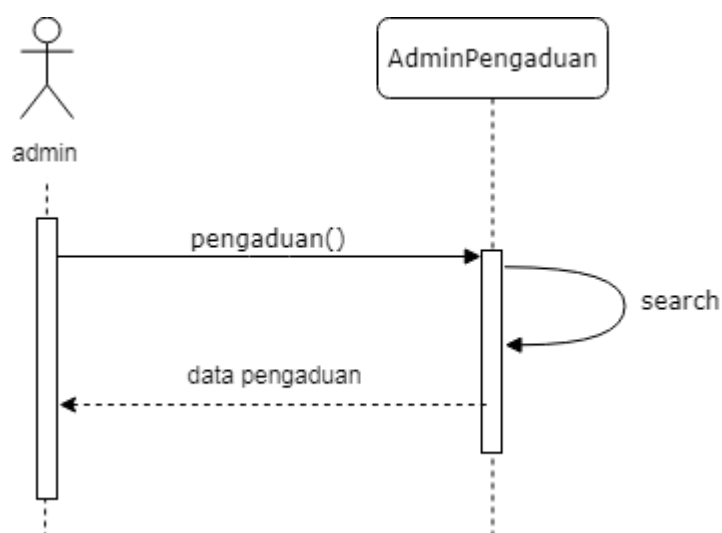
Sequence diagram pada *use case* edit list solusi, *use case* edit data pengaduan, *use case* edit daftar user, dan *use case* edit daftar instansi memiliki karakteristik yang sama dengan *sequence diagram* edit daftar kerusakan. Sehingga hanya dibuat *sequence diagram* edit daftar kerusakan saja.



Gambar 3. 29 Sequence Diagram Hapus Kerusakan

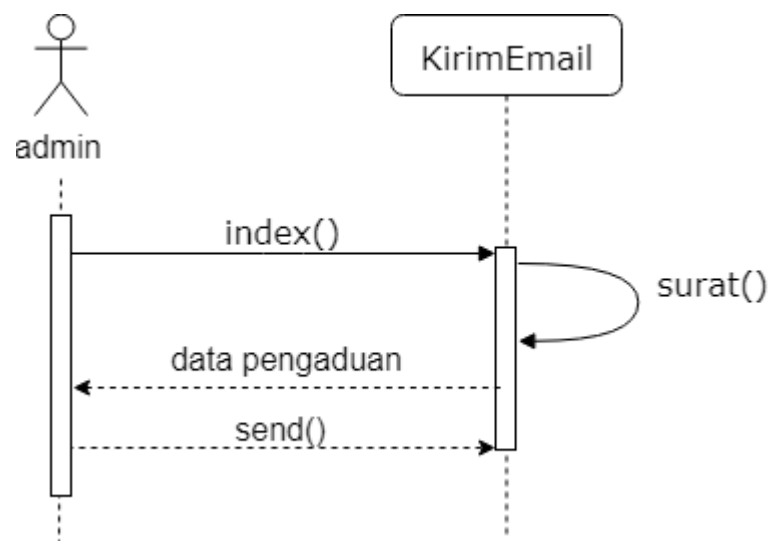
Sequence diagram pada Gambar 3.29 menjelaskan perilaku dan waktu yang digunakan class pada proses hapus kerusakan. Ketika *technical support* mengklik tombol “Hapus” pada tabel kerusakan maka akan dijalankan *query* “*getId*” pada kelas “*TechnicalSupport()*”. Kemudian sistem menjalankan fungsi “*deleteKerusakan()*”,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

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



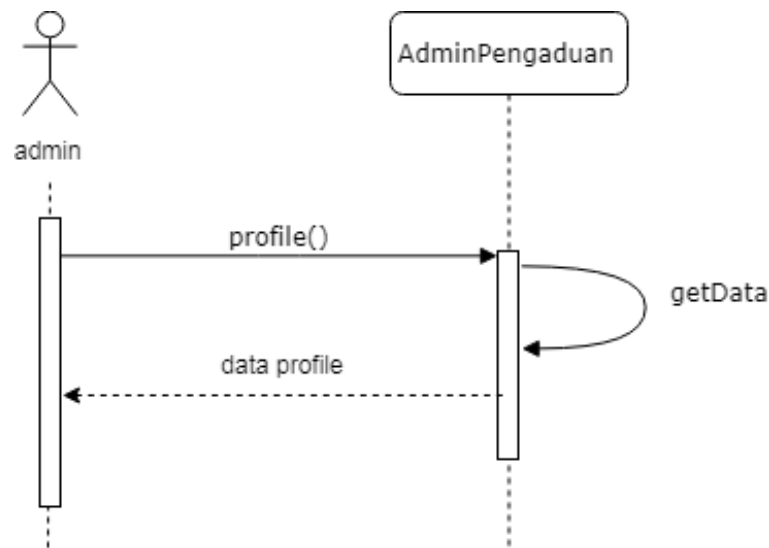
Gambar 3. 30 Sequence Diagram Search Pengaduan

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 *query* “*search*”, lalu data pengaduan yang dicari ditampilkan oleh sistem.



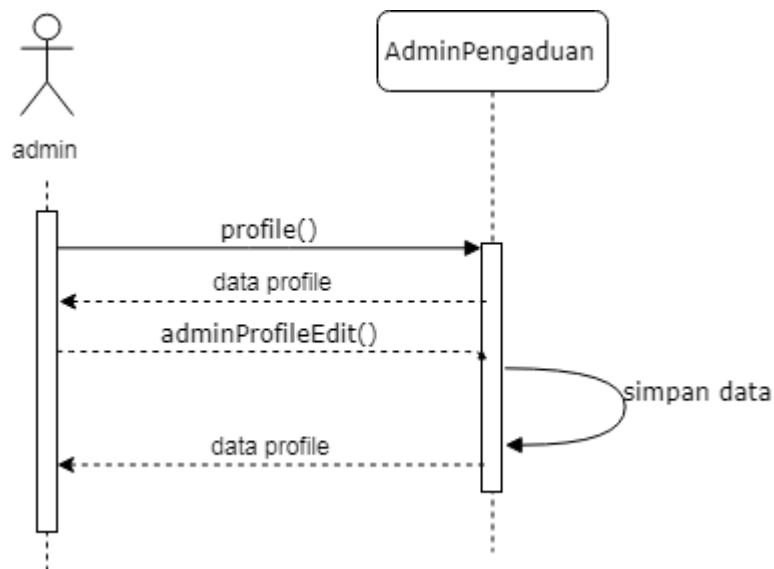
Gambar 3. 31 Sequence Diagram Kirim Pengaduan

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 ke alamat email yang terdaftar pada *database*. Pada proses kirim pengaduan hanya dapat dilakukan oleh admin. Hal pertama yang harus dilakukan untuk melakukan proses kirim pengaduan adalah dijalanannya fungsi “*index()*” pada kelas “*KirimEmail()*”. Kemudian sistem menjalankan fungsi “*surat()*”, lalu admin mengirim data pengaduan ke alamat email yang terdaftar pada *database*. Setelah itu dijalankan fungsi “*send()*”.



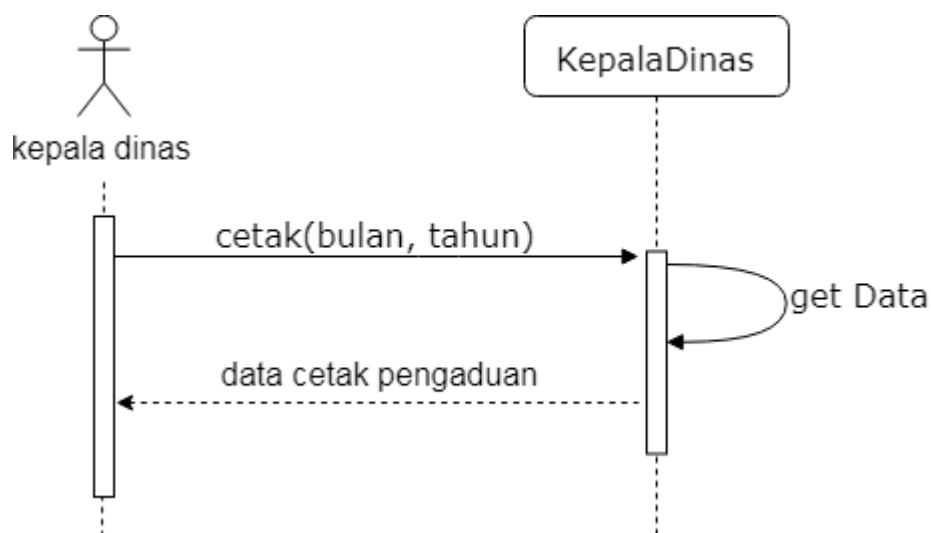
Gambar 3. 32 Sequence Diagram Melihat Profile Admin

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.



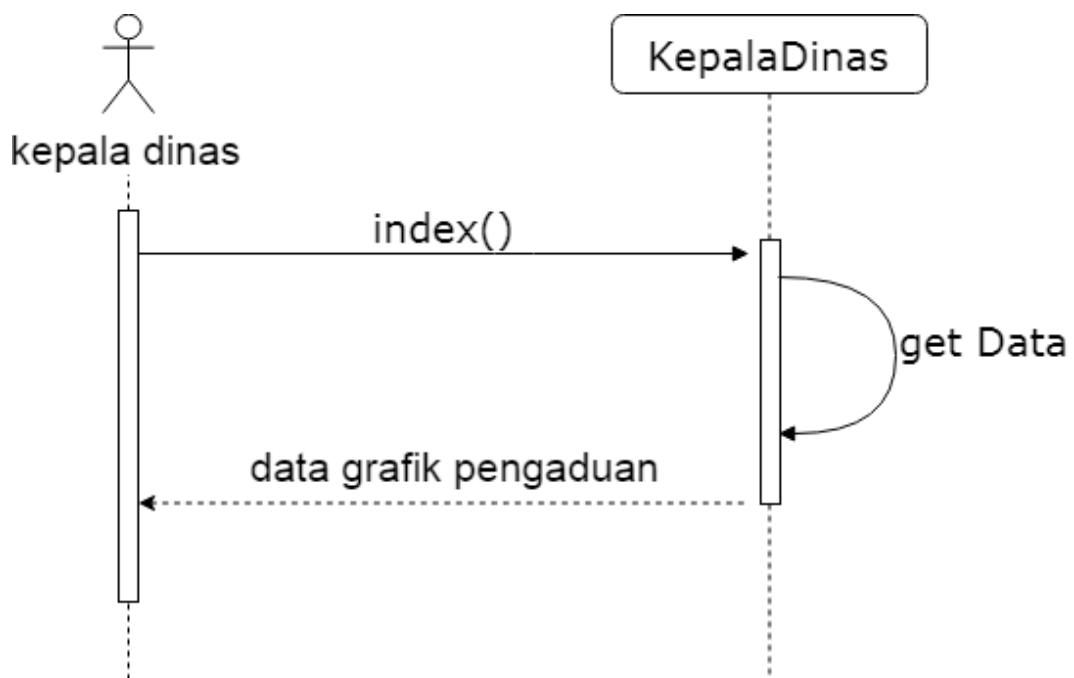
Gambar 3. 33 Sequence Diagram Mengedit Profile Admin

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.

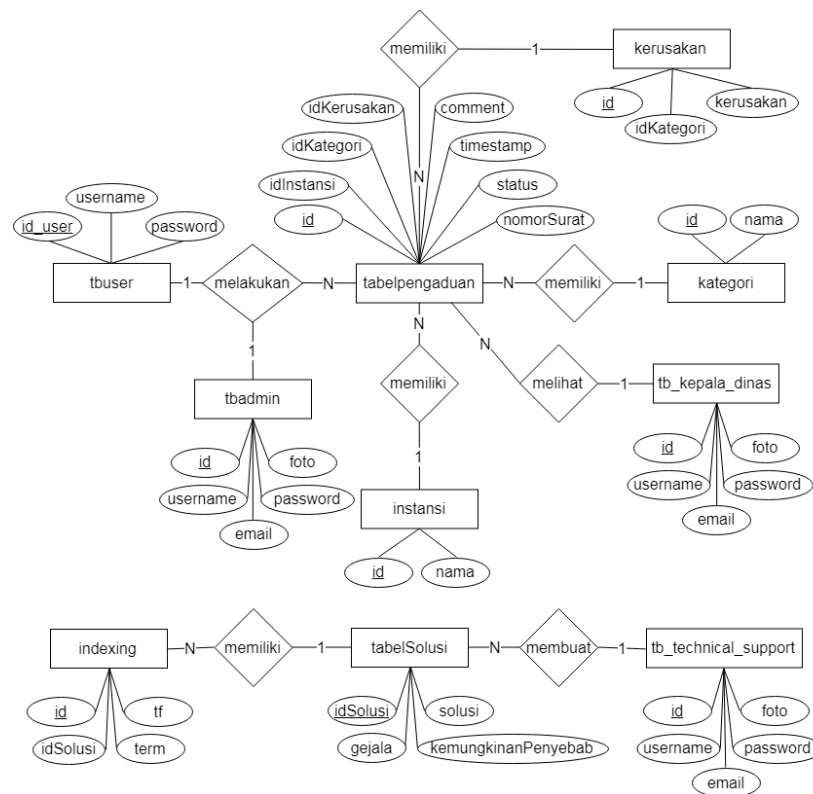


Gambar 3. 35 Sequence Diagram Melihat Grafik Pengaduan

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.

3.2.5 Entity Relational Diagram

Berdasarkan hasil dari analisis kebutuhan sistem dan rancangan *use case diagram* dapat dilihat pada Gambar 3.6 adalah Entity Relational Diagram (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:

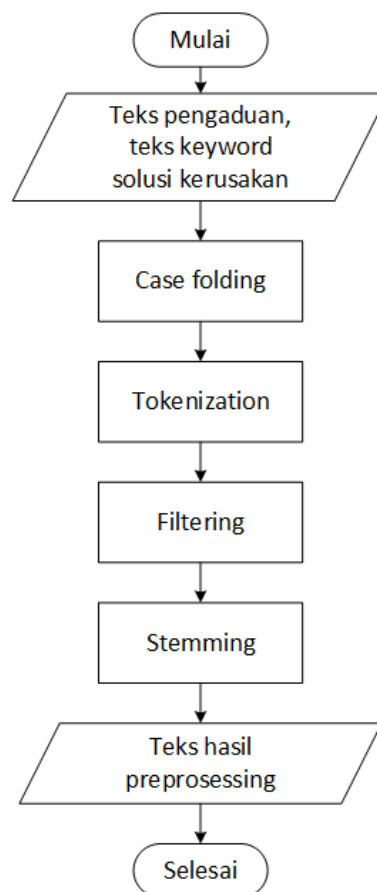
1. Relasi antara tbuser dengan tabelpengaduan, memiliki sifat *one to many* (1:N), dimana setiap baris data pada tbuser dapat berhubungan dengan banyak baris data pada tabelpengaduan. Karena setiap user dapat membuat banyak pengaduan.
2. Relasi antara tbadmin dengan tabelpengaduan, memiliki sifat *one 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 many* (1:N), dimana setiap baris data pada instansi dapat berhubungan dengan

banyak baris data pada tabelpengaduan. Karena setiap instansi dapat memiliki banyak pengaduan.

4. Relasi antara kerusakan dengan tabelpengaduan, memiliki sifat *one to many* (1:N), dimana setiap baris data pada kerusakan dapat berhubungan dengan banyak baris data pada tabelpengaduan. Karena setiap kerusakan dapat memiliki banyak pengaduan.
5. Relasi antara kategori dengan tabelpengaduan, memiliki sifat *one to many* (1:N), dimana setiap baris data pada kategori dapat berhubungan dengan banyak baris data pada tabelpengaduan. Karena setiap kategori dapat memiliki banyak pengaduan.
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.

3.2.6 *Preprocessing, TF-IDF, Cosine Similarity*

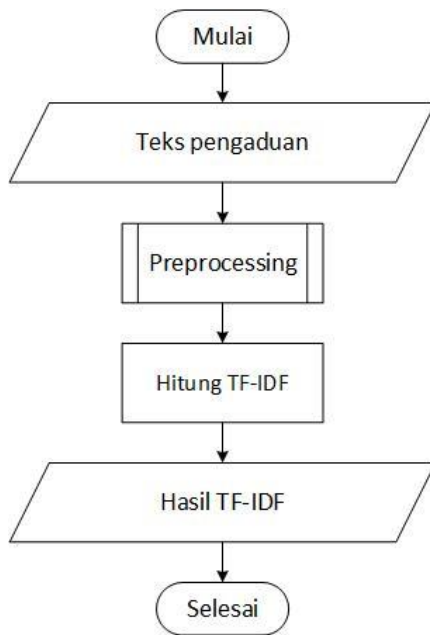
Dalam membangun sistem terdapat tahapan penting yang dilakukan pada penelitian ini untuk mendapatkan output berupa solusi kerusakan komputer yaitu tahap preprocessing data, TF-IDF, dan cosine similarity untuk mendeteksi kemiripan teks. Adapun *flowchart* preprocessing, TF-IDF, dan *cosine similarity*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



Gambar 3. 37 Flowchart Preprocessing

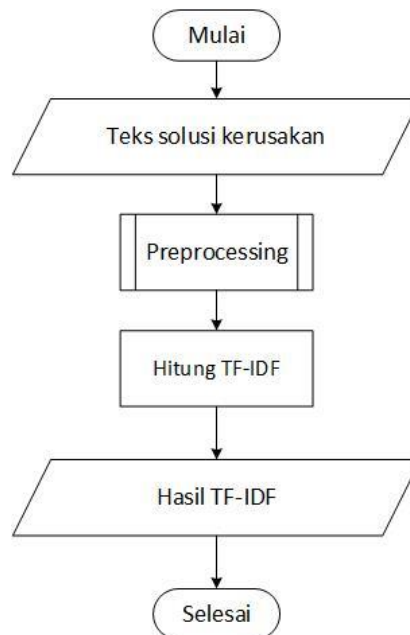
Flowchart preprocessing pada Gambar 3.37 adalah suatu proses yang bertujuan mengolah data mentah menjadi data yang berkualitas. Pada *flowchart preprocessing* inputan berupa teks pengaduan dan teks *keyword* solusi kerusakan. Dokumen-dokumen diproses dengan menghilangkan karakter tanda baca dan mengubah 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



(a)

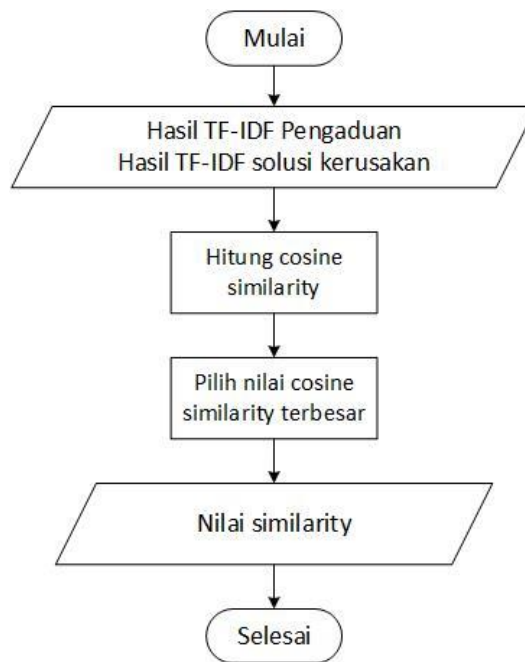
Flowchart TF-IDF Solusi



(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). *Term frequency* merupakan jumlah kemunculan suatu kata (*term*) pada dokumen. Rumus dari *invers document frequency* adalah $idf_j = \log \frac{D}{df_j}$. Dimana D adalah jumlah semua solusi yang ada pada *database* dan df_i adalah jumlah solusi pada *database* yang mengandung *term* t_j .



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 komputer menggunakan *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

$$Sim(d_i, q) = \frac{\sum_{i=1}^N w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^N w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^N w_{i,q}^2}}. \text{ Dimana } q \text{ adalah } query \text{ yang diinputkan oleh}$$

user dan d_j adalah bobot solusi untuk istilah j . $\sum_{i=1}^N w_{i,j}$ adalah jumlah bobot kata i pada solusi j . $\sum_{i=1}^N w_{i,q}$ adalah jumlah bobot kata i pada *query*. 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.

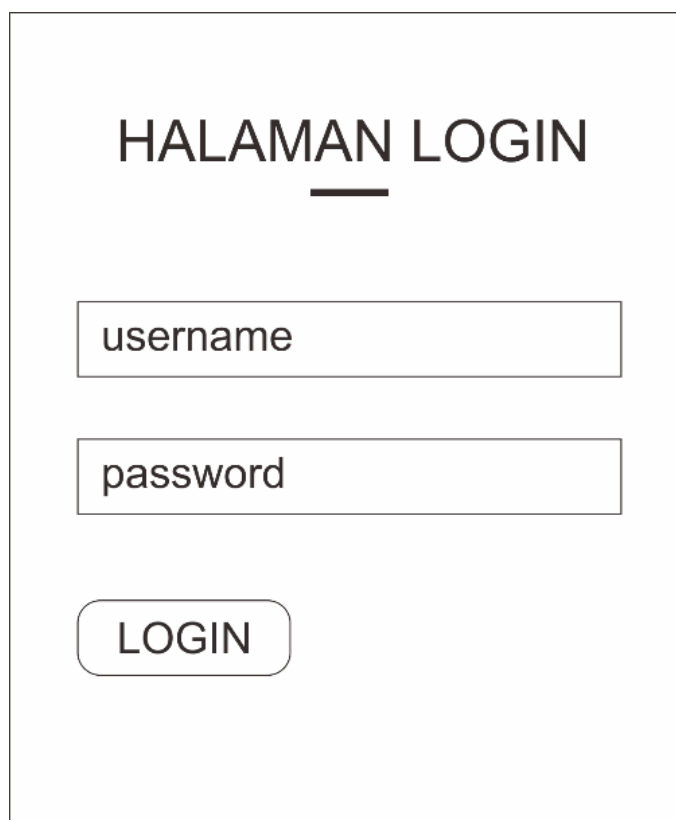
3.2.7 Perancangan Antar Muka

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

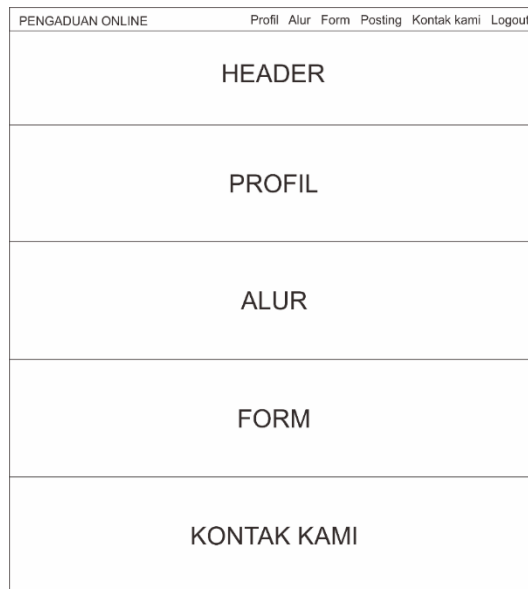
Rancangan tampilan login merupakan tampilan yang digunakan user untuk proses autentikasi. Tampilan ini diakses sebelum user ingin menuju halaman dashboard, dengan memasukkan *username* dan *password*. Desain tampilan login dapat dilihat seperti pada Gambar 3.40.



Gambar 3. 40 Desain Tampilan Login

2. Tampilan Halaman Awal Pegawai

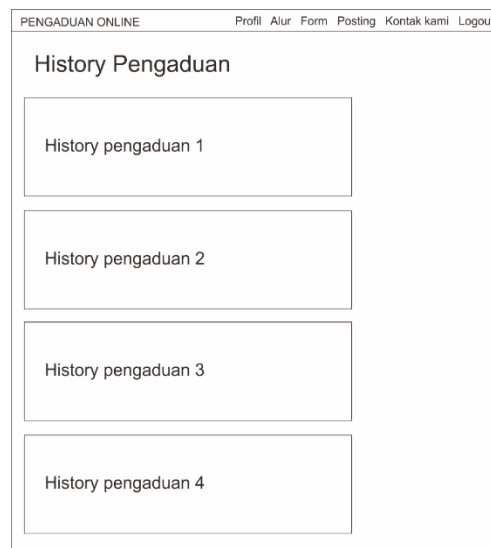
Rancangan tampilan halaman awal pegawai merupakan tampilan awal yang muncul setelah pegawai melalui proses login. Tampilan halaman awal pegawai terdapat menu navigasi (profil, alur, form, posting, kontak kami, dan logout), header, profil, alur, form, kontak kami. Desain tampilan halaman awal pegawai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



Gambar 3. 41 Desain Tampilan Halaman Awal Pegawai

3. 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

4. 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.

The wireframe shows an admin dashboard titled 'List Pengaduan'. On the left is a sidebar with a user profile section (Foto Admin) and a menu with items: Pengaduan, Daftar User, Profile, and Sign Out. The main content area contains a title 'List Pengaduan', a row of buttons for 'Tambah Pengaduan', 'Pilih tahun' (dropdown), 'Pilih bulan' (dropdown), and 'Cetak Pengaduan'. Below this is a placeholder for a 'Grafik' (chart). A search bar is located above a table. The table has columns: No, No Surat, Instansi, Kategori, Kerusakan, Status, and Action. The Action column contains buttons for 'edit', 'hapus', and 'kirim'.

No	No Surat	Instansi	Kategori	Kerusakan	Status	Action
						edit hapus kirim

Gambar 3. 43 Desain Tampilan Halaman Awal Admin

5. 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.

<div>Foto Admin</div> <div>Pengaduan</div> <div>Daftar User</div> <div>Profile</div> <div>Sign Out</div>	List User		
	Tambah User		
	No	Nama User	Action
			edit hapus

Gambar 3. 44 Desain Tampilan Halaman Daftar User

6. 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.

<div>Foto Admin</div> <div>Pengaduan</div> <div>Daftar User</div> <div>Profile</div> <div>Sign Out</div>	Profile Admin	
	Nama	
	<input type="text"/>	
	Email	
	<input type="text"/>	
	Foto <div>Foto</div> <div>Choose File</div> <div>Submit</div>	

Gambar 3. 45 Desain Tampilan Halaman Profile

7. 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.

<div> <div>Foto</div> <div>Technical Support</div> </div> <div> <div>List Pengaduan</div> <div>Daftar Kerusakan</div> <div>List Solusi</div> <div>Sign Out</div> </div>	List Pengaduan					
	No	Instansi	Kategori	Kerusakan	Status	Action
						<div>solusi</div>

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

8. 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.

<div>Foto</div> <div>Technical Support</div>	<h2>List Kerusakan</h2>																																																																								
List Pengaduan	<div>Tambah Kerusakan</div>																																																																								
Daftar Kerusakan																																																																									
List Solusi																																																																									
Sign Out																																																																									
	<table border="1"> <thead> <tr> <th>No</th> <th>Kerusakan</th> <th>Kategori</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td>edit hapus</td> </tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>	No	Kerusakan	Kategori	Action				edit hapus																																																																
No	Kerusakan	Kategori	Action																																																																						
			edit hapus																																																																						

Gambar 3. 47 Desain Tampilan Halaman Daftar Kerusakan

9. Halaman List Solusi

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

[illegible]

Gambar 3. 48 Desain Tampilan Halaman List Solusi

10. Halaman Kepala Dinas

Halaman kepala dinas merupakan halaman yang digunakan kepala dinas untuk melihat rekap pengaduan 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.

No	No Surat	Instansi	Kategori	Kerusakan	Status

Gambar 3. 49 Desain Tampilan Halaman Kepala Dinas

3.2.8 Skenario Pengujian Sistem

Bagian ini menjelaskan mengenai skenario pengujian sistem yang akan dilakukan pada sistem pengaduan. Pengujian sistem pengaduan dilakukan dengan menguji *Black Box Testing*, *Stress Testing*, dan pengujian terhadap respon pengguna.

1. *Black Box Testing*

Pada *black box testing* dilakukan dengan berfokus pada *output* yang dihasilkan dalam menerima *input* yang dipilih. *Input* yang dipilih berdasarkan kebutuhan fungsional sistem yang ada pada sub bab 3.1.2. *Black box* testing dilakukan dengan menentukan kesesuaian *input* dengan kebutuhan pengguna dalam kebenaran prosesnya. Skenario *black box testing* dapat dilihat pada tabel 3.10.

Tabel 3. 10 Skenario *Black Box Testing*

Kode Kebutuhan:			
Kasus:			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian

2. *Stress Testing*

Pada *stress testing* terbagi menjadi 2 tahap yaitu *performance test* dan *stress test*. Skenario pengujian pada *stress test* yaitu dengan menguji kekuatan sistem pengaduan apakah sudah sesuai dengan standar sistem pada umumnya dalam menangani beban user. Pada pengujian pada *stress test* menggunakan aplikasi WAPT. Skenario pengujian pada *performance test* yaitu dengan mencari rata-rata waktu yang dibutuhkan oleh sistem ketika menampilkan solusi kerusakan berdasarkan jumlah solusi yang ada pada *database* dan jumlah term pada *query* yang digunakan. Skenario pengujian pada *performance test* dapat dilihat pada tabel 3.11.

Tabel 3. 11 Skenario Pengujian Pada *Performance Test*

Jumlah Solusi	Jumlah Term		
	1 Term	2 Term	3 Term
Rata-rata			

3. Pengujian Terhadap Respon Pengguna

Skenario pengujian terhadap respon pengguna dilakukan dengan membagikan kuisioner kepada pegawai dinas selaku *user* pada sistem pengaduan. Jumlah kuisioner yang dibagikan sebanyak 10 kuisioner. Daftar pertanyaan pada kuisioner dapat dilihat pada lampiran B.

BAB IV

HASIL DAN PEMBAHASAN

4.1 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.

4.2 Lingkungan Implementasi

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

1. Kebutuhan Perangkat Keras (*Hardware*)

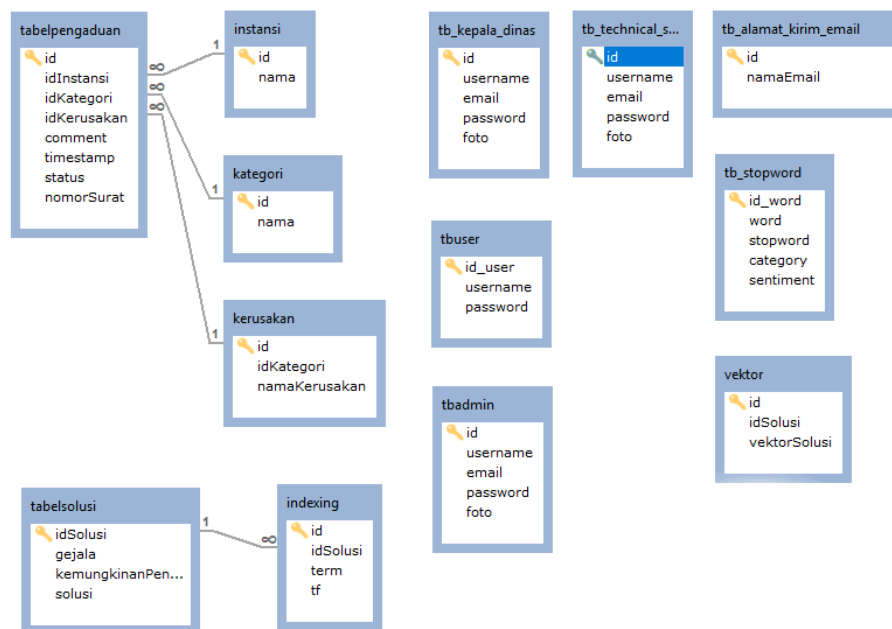
Spesifikasi minimal perangkat keras yang dibutuhkan untuk menunjang dan membantu dalam melakukan penelitian yaitu :

- a. Laptop atau *notebook* dengan spesifikasi minimal :
 - Processor intel(R) 2.0 GHz
 - Memory DDR2 2 GB
 - VGA 1 GB
 - Harddisk 80 GB

2. Kebutuhan Perangkat Lunak (*Software*)

Kebutuhan perangkat 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.

4.3 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 *tabelPengaduan* merupakan tabel yang dihasilkan dari entitas *tabelPengaduan* 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.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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*.

11. 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.

12. 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.

13. 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.

4.4 Implementasi Sistem

Tahap implementasi sistem 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.

4.4.1 Implementasi Proses *Preprocessing* Pada Sistem

Implementasi proses *preprocessing* terdiri dari *case folding*, *tokenizing*, *filtering*, dan *stemming*. *Preprocessing* 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	//menghilangkan tanda baca
2	\$namakerusakan = preg_replace("/[[:punct:]]+/", " ", \$query);
3	
4	//menghubah menjadi huruf kecil
5	\$namakerusakan = strtolower(\$namakerusakan);
6	
7	//menjadikan array
8	\$namakerusakan = explode(" ", \$namakerusakan);
9	
10	//pencocokan kata atau stopwords
11	\$fileStopword= base_url(). 'assets/preprocessing/stopword.txt';
12	\$stopwords = file(\$fileStopword, FILE_IGNORE_NEW_LINES);
13	\$stopword = array_values(array_diff(\$gejala, \$stopwords));
14	
15	//hitung tf
16	\$tf = array_values(array_count_values(\$stopword));
17	
18	//menghilangkan kata yg double
19	\$doubleTermRemove = array_values(array_unique(\$stopword));
20	
21	//Stemming
22	\$jumlahDoubleTermRemove = count(\$doubleTermRemove);
23	\$i=0;
24	while (\$i<\$jumlahDoubleTermRemove) {
25	\$this->load->library('stemming');
26	\$textAwal = \$doubleTermRemove[\$i];
27	\$stemming = \$this->stemming->stemming(\$textAwal);
28	if (\$stemming=='') {
29	term[\$i] = \$ doubleTermRemove[\$i];
30	}else{
31	term[\$i] = \$stemming;
32	}
33	\$i++;
34	}

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 dilakukan proses *tokenizing* dengan memecah teks kedalam bentuk kata (*term*) dan dijadikan kedalam bentuk *array*, seperti pada line 8. Kemudian dilakukan proses *filtering* yaitu dengan membandingkan daftar kata yang ada di *stopword* dengan *term* dari hasil *tokenizing* kemudian dijadikan kedalam bentuk *array*, seperti pada

line 13. Pada line 15 adalah proses menghitung jumlah *tf* dan dimasukkan kedalam bentuk *array*. Kemudian dilanjutkan dengan proses menghilangkan kata yang *double* (contoh: kata “aku aku” menjadi “aku”) dan di masukkan kedalam bentuk *array*, seperti pada line 18. Hasil dari proses penghilangan kata *double*, kemudian dihitung jumlah kata tersebut dan ditampung ke dalam *array*, seperti pada line 21. Kemudian dilakukan proses menghilangkan imbuhan, awalan, dan akhiran hingga menghasilkan kata dasar (*stemming*) seperti pada line 23-34.

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

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 ketersediaan *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*.

4.4.3 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 *Query*

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

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 *query*) dibagi dengan jumlah *tf*+1 (termasuk *query*) seperti pada line 12. Setelah itu menghitung nilai bobot *query* dengan carat *tf* dikali IDF.

Perhitungan bagian pengkuadratan pada penyebut (*qery*) yang ada pada rumus *cosine similarity* diselipkan gunanya untuk mengefisienkan kode. Proses perhitungannya adalah jumlah dari semua hasil kuadrat pada bobot *query* seperti pada line 15. Kemudian disimpan ke dalam bentuk *array* agar dapat dipanggil ketika perhitungan *cosine similarity*. Pada line 19 adalah proses untuk menampung bobot *term* dari *qery*. Pada line 23 adalah proses untuk menampung semua *term* yang mempunyai bobot pada *query*.

4.4.4 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	//mencari solusi kerusakan
2	\$id=\$this->ModelTechnicalSupport->select_id(); // mencari id
3	\$y=0;
4	
5	foreach (\$id->result() as \$id) {
6	\$idSolusi[\$y]=\$id->idSolusi; // id ubah ke array
7	
8	\$term=\$this->ModelTechnicalSupport->
9	selecttermIndex(\$idSolusi[\$y]); // mencari term berdasarkan id
10	dokumen
11	\$z=0;
12	\$bobotKuadrat=0;
13	\$bobotKuadratSolusi=0;
14	\$atas=0;
15	foreach (\$term->result() as \$term) {
16	if (in_array(\$term->term, \$termTerseleksi))
17	{ //menghitung pembilang, kata x yang ada query dan dokumen
18	\$df= \$this->ModelTechnicalSupport->select_df();
19	\$tf= \$this->ModelTechnicalSupport->select_tf(\$term->
20	term);
21	
22	\$idfSolusi=log10((\$df+1)/(\$tf+1));
23	\$bobot=\$term->tf* \$idfSolusi;
24	
25	\$bobotKuadrat=\$bobotKuadrat + pow(\$bobot, 2);
26	//bobot pembilang
27	

```

28         $atas= $atas+ $bobot* $bobotQ[$term->term];
29         $z++;
30     }
31     else{
32
33         //menghitung penyebut solusi
34         $df= $this->ModelTechnicalSupport->select_df();
35         $tf= $this->ModelTechnicalSupport->select_tf($term-
36         >term);
37         $idfSolusi=log10(($df+1)/($tf));
38         $bobot=$term->tf*$idfSolusi;
39         //rumus cosin penyebut
40         //bobotKuadratSolusi = perhitungan bobot solusi yg ada
41         di penyebut, dan hanya bobot kata x yg belum terhitung
42         dibagian pembilang. agar lebih efisien
43
44         $bobotKuadratSolusi=$bobotKuadratSolusi + pow($bobot,
45         2);
46     }
47 }
48 if ($z==0) { //perhitungan jika semua kata tidak ada yg sama di
49 query dan dokumen
50 $y++;
51 } else{
52     //penyebutSolusi
53     $kuadratSolusi=$bobotKuadrat+$bobotKuadratSolusi;
54     $jumDocKaliQuery[$idfSolusi[$y]]=$atas;
55     $jum_akarkuadrat[$idfSolusi[$y]]=sqrt($kuadratSolusi);
56     $jum_akarkuadratQuery=sqrt($bobotKuadratQuery);
57     //unset($QuerykaliTerm);
58     unset($bobotKuadrat);
59     unset($bobotKuadratSolusi);
60
61     $hasil[$idfSolusi[$y]]=$jumDocKaliQuery[$idfSolusi[$y]]/($jum_
62     akarkuadrat[$idfSolusi[$y]]*$jum_akarkuadratQuery);
63     $y++;
64
65     $max =max($hasil); //cari hasil cosine tertinggi
66     $key = array_keys($hasil, $max);
67     $keys =$key[0];
68 }

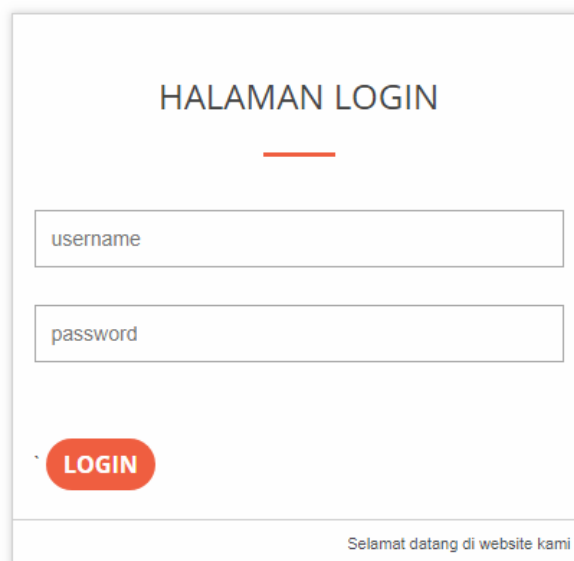
```

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, seperti pada line 2. 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*), seperti

pada line22. Kemudian menghitung bobot solusi dengan cara tf dikali IDF, seperti pada line 23. Setelah mendapatkan hasil dari proses hitung bobot solusi kemudian dikuadratkan, seperti pada line 25. Langkah selanjutnya untuk mencari bobot yang ada pada bagian pembilang rumus *cosine similarity* adalah menjumlahkan bobot kata yang sama pada *query* dengan bobot solusi, seperti pada line 27. Untuk menghitung bagian penyebut (solusi) yang ada dalam rumus *cosine similarity* adalah dengan menghitung bobot solusi kemudian dikuadratkan, seperti pada line 33-45. Pada line 48 adalah pengecekan *term* pada *query* dengan dokumen solusi. Jika tidak ada *term* yang sama maka lanjut ke dokumen solusi berikutnya, jika ada maka dijalankan proses perhitungan *cosine similarity*, seperti pada line 61. Kemudian dilanjutkan proses seleksi hasil perhitungan *cosine similarity* tertinggi, seperti pada line 64.

4.5 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 menggunakan HTML, CSS, dan Java Script.



The image shows a web form for logging in. At the top, the title 'HALAMAN LOGIN' is centered. Below it is a horizontal red line. There are two text input fields: the first is labeled 'username' and the second is labeled 'password'. Below these fields is a red button with the word 'LOGIN' in white capital letters. At the bottom right of the form, there is a small text message: 'Selamat datang di website kami'.

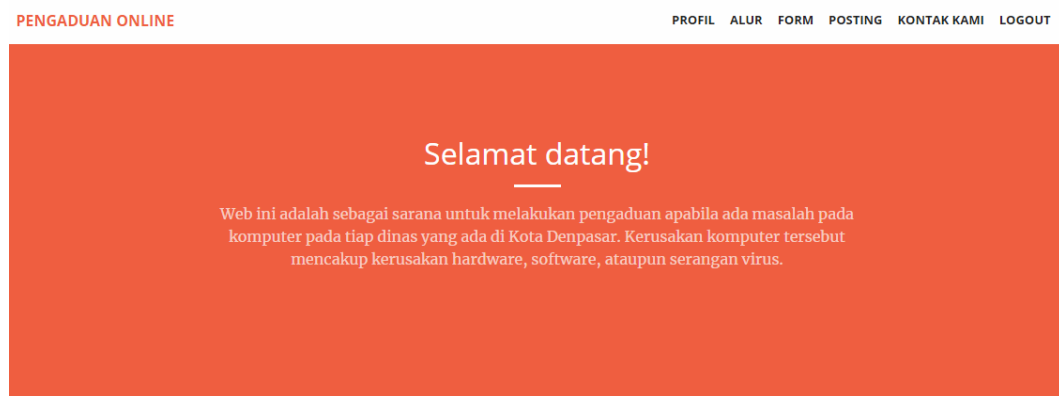
Gambar 4. 2 Tampilan Antar Muka Halaman Login

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.

PENGADUAN ONLINE PROFIL ALUR **FORM** POSTING KONTAK KAMI LOGOUT

FORM PENGADUAN

(*) Harap mengisi data yang benar

Instansi:

- Select Dinas -

Kategori:

- Select Kategori -

Kerusakan:

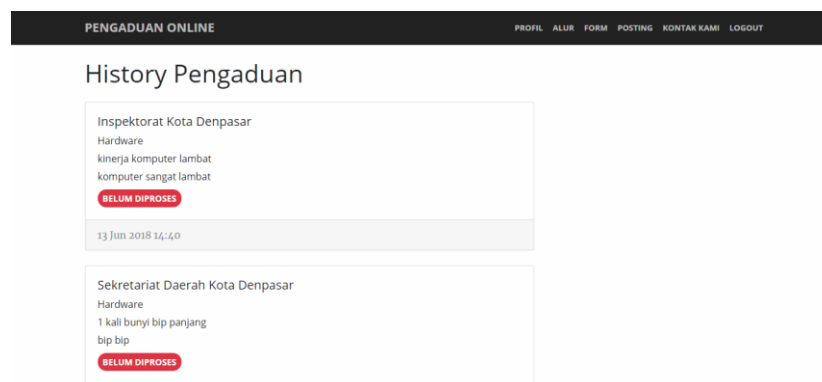
- Select Kerusakan -

Comment :

SUBMIT

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



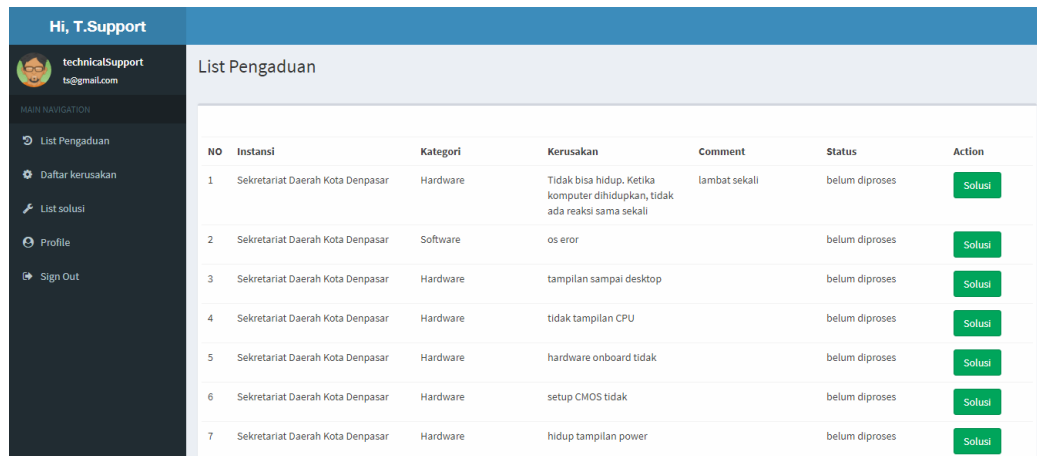
Gambar 4. 7 Tampilan Antar Muka Halaman Posting

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

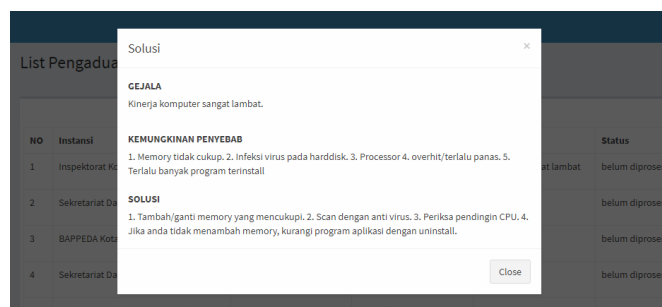
Gambar 4.8 merupakan tampilan dari halaman kontak kami pada sistem pengaduan untuk pegawai. Pada halaman kontak kami berisi alamat, nomor telepon dan email yang dapat dihubungi.



NO	Instansi	Kategori	Kerusakan	Comment	Status	Action
1	Sekretariat Daerah Kota Denpasar	Hardware	Tidak bisa hidup. Ketika komputer dihidupkan, tidak ada reaksi sama sekali	lambat sekali	belum diproses	Solusi
2	Sekretariat Daerah Kota Denpasar	Software	os error		belum diproses	Solusi
3	Sekretariat Daerah Kota Denpasar	Hardware	tampilan sampai desktop		belum diproses	Solusi
4	Sekretariat Daerah Kota Denpasar	Hardware	tidak tampilan CPU		belum diproses	Solusi
5	Sekretariat Daerah Kota Denpasar	Hardware	hardware onboard tidak		belum diproses	Solusi
6	Sekretariat Daerah Kota Denpasar	Hardware	setup CMOS tidak		belum diproses	Solusi
7	Sekretariat Daerah Kota Denpasar	Hardware	hidup tampilan power		belum diproses	Solusi

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 berdasarkan peengaduan kerusakan yang telah diadu seperti pada Gambar 4.10



NO	Instansi	Status
1	Inspektorat Kota Denpasar	at lambat, belum diproses
2	Sekretariat Daerah Kota Denpasar	belum diproses
3	BAPPEDA Kota Denpasar	belum diproses
4	Sekretariat Daerah Kota Denpasar	belum diproses
5	Badan Penyelidikan Keusutan dan Ases	belum diproses

Solusi

GEJALA
Kinerja komputer sangat lambat.

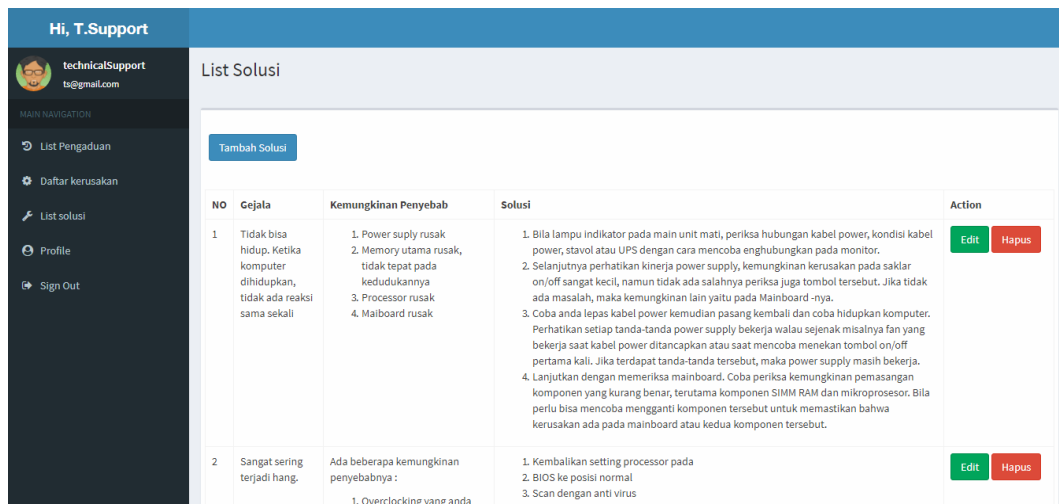
KEMUNGKINAN PENYEBAB
1. Memory tidak cukup. 2. Infeksi virus pada harddisk. 3. Processor 4. overhit/terlalu panas. 5. Terlalu banyak program terinstall

SOLUSI
1. Tambah/ganti memory yang mencukupi. 2. Scan dengan anti virus. 3. Periksa pendingin CPU. 4. Jika anda tidak menambah memory, kurangi program aplikasi dengan uninstall.

Close

Gambar 4. 10 Tampilan Antar Muka Halaman Solusi

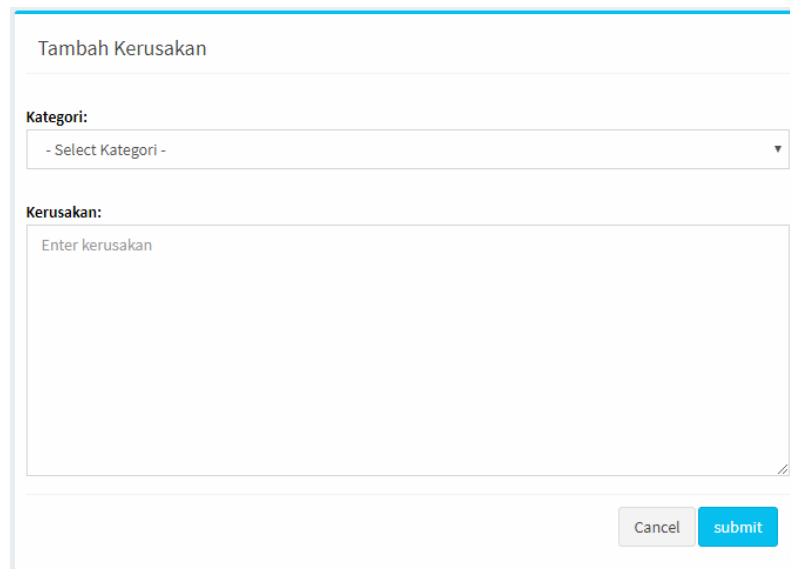
Gambar 4.10 merupakan tampilan dari halaman solusi pada sistem untuk *technical support*. Pada halaman solusi berisi informasi mengenai gejala, kemungkinan penyebab, dan solusi dari kerusakan komputer yang telah diadu oleh pegawai. Kemudian juga terdapat tombol *close* yang berguna ketika *technical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



NO	Gejala	Kemungkinan Penyebab	Solusi	Action
1	Tidak bisa hidup. Ketika komputer dihidupkan, tidak ada reaksi sama sekali	1. Power supply rusak 2. Memory utama rusak, tidak tepat pada kedudukannya 3. Processor rusak 4. Mainboard rusak	1. Bila lampu indikator pada main unit mati, periksa hubungan kabel power, kondisi kabel power, stavol atau UPS dengan cara mencoba menghubungkan pada monitor. 2. Selanjutnya perhatikan kinerja power supply, kemungkinan kerusakan pada saklar on/off sangat kecil, namun tidak ada salahnya periksa juga tombol tersebut. Jika tidak ada masalah, maka kemungkinan lain yaitu pada Mainboard -nya. 3. Coba anda lepas kabel power kemudian pasang kembali dan coba hidupkan komputer. Perhatikan setiap tanda-tanda power supply bekerja walau sejenak misalnya fan yang bekerja saat kabel power ditancapkan atau saat mencoba menekan tombol on/off pertama kali. Jika terdapat tanda-tanda tersebut, maka power supply masih bekerja. 4. Lanjutkan dengan memeriksa mainboard. Coba periksa kemungkinan pemasangan komponen yang kurang benar, terutama komponen SIMM RAM dan mikroprosesor. Bila perlu bisa mencoba mengganti komponen tersebut untuk memastikan bahwa kerusakan ada pada mainboard atau kedua komponen tersebut.	Edit Hapus
2	Sangat sering terjadi hang.	Ada beberapa kemungkinan penyebabnya : 1. Overclocking yang anda	1. Kembalikan setting processor pada 2. BIOS ke posisi normal 3. Scan dengan anti virus 4. ...	Edit Hapus

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.



Tambah Kerusakan

Kategori:

- Select Kategori -

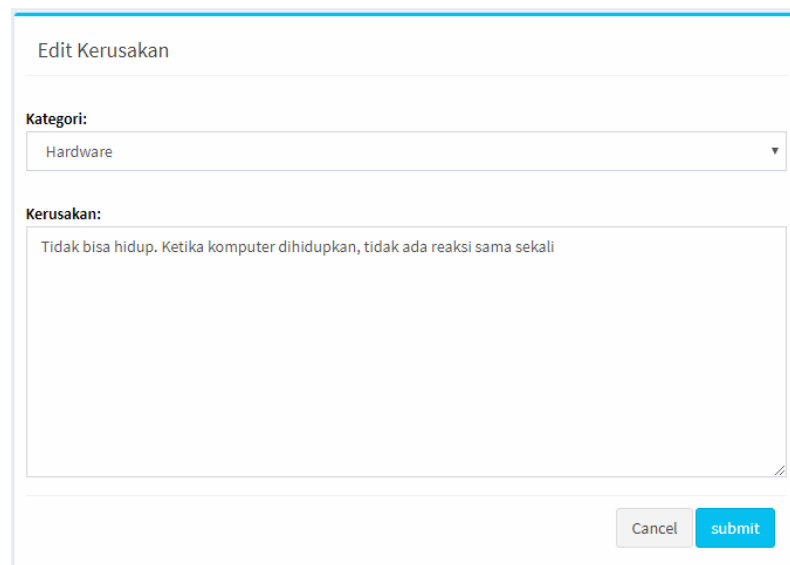
Kerusakan:

Enter kerusakan

Cancel submit

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.



Edit Kerusakan

Kategori:

Hardware

Kerusakan:

Tidak bisa hidup. Ketika komputer dihidupkan, tidak ada reaksi sama sekali

Cancel submit

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 penyebab, 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.

Tambah Solusi

Gejala:

Kemungkinan Penyebab:

Solusi:

Cancel submit

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.

Edit Solusi

Gejala:

Tidak bisa hidup. Ketika komputer dihidupkan, tidak ada reaksi sama sekali

Kemungkinan Penyebab:

1. Power suply rusak
2. Memory utama rusak, tidak tepat pada kedudukannya
3. Processor rusak
4. Malboard rusak

Solusi:

1. Bila lampu indikator pada main unit mati, periksa hubungan kabel power, kondisi kabel power, staval atau UPS dengan cara mencoba menghubungkan pada monitor.
2. Selanjutnya perhatikan kinerja power supply, kemungkinan kerusakan pada saklar on/off sangat kecil, namun tidak ada salahnya periksa juga tombol tersebut. Jika tidak ada masalah, maka kemungkinan lain yaitu pada Mainboard -nya.
3. Coba anda lepas kabel power kemudian pasang kembali dan coba hidupkan

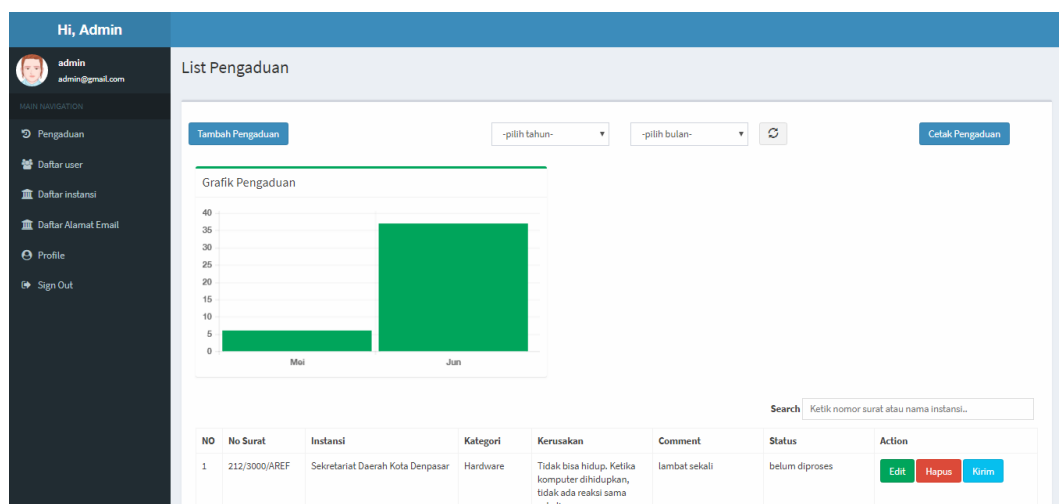
Cancel submit

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.

Gambar 4. 17 Tampilan Antar Muka Halaman Profile

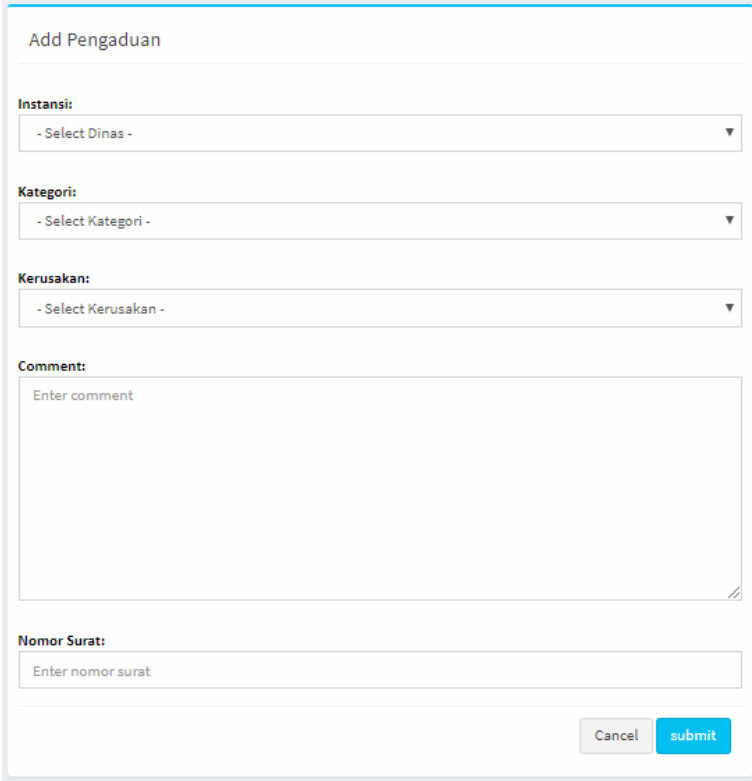
Gambar 4.17 merupakan tampilan dari halaman profile pada sistem pengaduan untuk *technical support*. Pada halaman profile *technical support* dapat mengubah nama, email, dan foto profile sesuai keinginan *technical support*.



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 admin ingin mengedit pengaduan maka admin dapat 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 admin dapat 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*.




The image shows a web form titled "Add Pengaduan". It contains the following fields:

- Instansi:** A dropdown menu with the option "- Select Dinas -".
- Kategori:** A dropdown menu with the option "- Select Kategori -".
- Kerusakan:** A dropdown menu with the option "- Select Kerusakan -".
- Comment:** A text area with the placeholder text "Enter comment".
- Nomor Surat:** A text input field with the placeholder text "Enter nomor surat".

At the bottom right of the form, there are two buttons: "Cancel" and "submit".

Gambar 4. 19 Tampilan Antar Muka Halaman Tambah Pengaduan

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

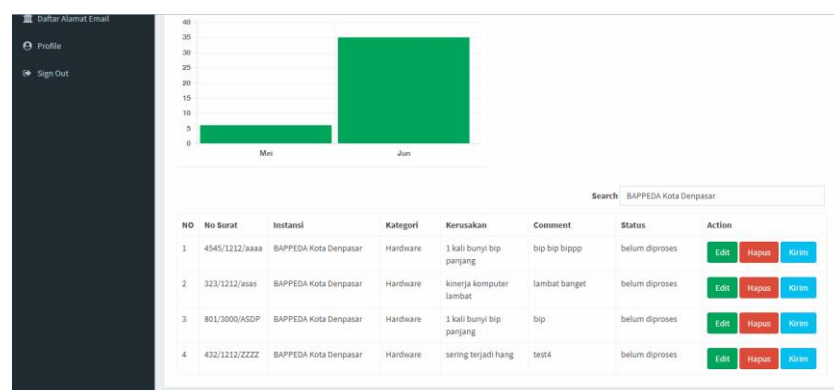


NO	INSTANSI	KATEGORI	KERUSAKAN	STATUS	NO SURAT
1	BAPPEDA Kota Denpasar	Hardware	sering terjadi hang	belum diproses	432/1212/ZZZZ
2	Inspektori Kota Denpasar	Hardware	1 kali bunyi bip panjang	belum diproses	800/3562/ZXCXV
3	Badan Pengolahan Keuangan dan Aset Daerah Kota Denpasar	Hardware	CMOS failure	sedang diproses	222/2222/qjhw
4	Sekretariat Daerah Kota Denpasar	Hardware	sering terjadi hang	belum diproses	111/1111/aaaa
5	BAPPEDA Kota Denpasar	Hardware	1 kali bunyi bip panjang	belum diproses	801/3000/ASDP
6	Sekretariat Daerah Kota Denpasar	Hardware	1 kali bunyi bip panjang	belum diproses	102/1091/DASV

KATEGORI KERUSAKAN	
HARDWARE	6
SOFTWARE	0
JARINGAN	0

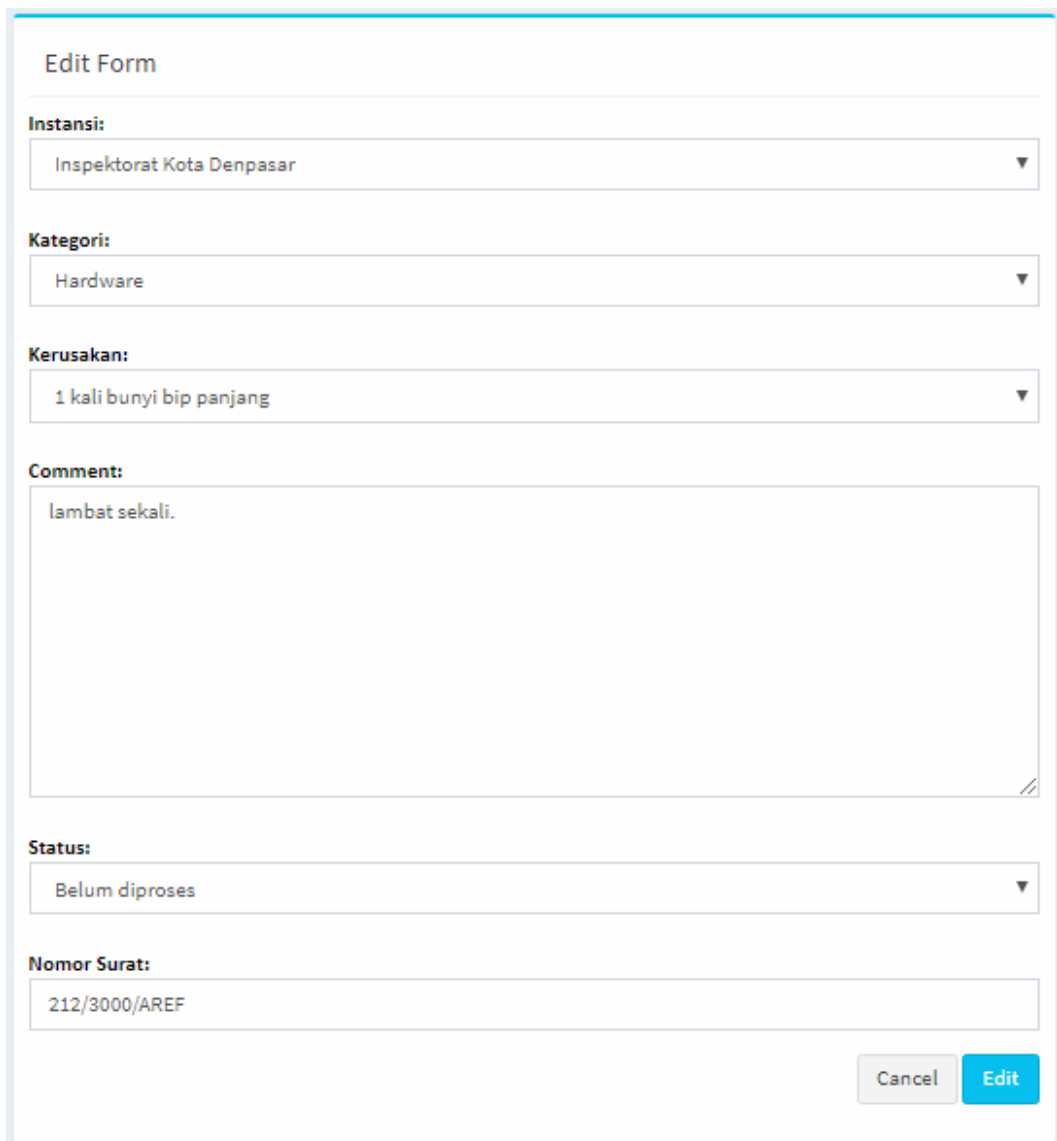
Gambar 4. 20 Tampilan Antar Muka Halaman Cetak Pengaduan

Tampilan antar muka halaman cetak pengaduan untuk kepala dinas sama dengan tampilan antar muka halaman cetak pengaduan untuk admin. Oleh karena itu gambar dan penjelasan yang dibuat hanya satu. Gambar 4.20 merupakan tampilan dari halaman cetak pengaduan pada sistem pengaduan untuk admin. Halaman cetak pengaduan berisi data rekap pengaduan berdasarkan tahun dan bulan yang diinginkan. Pada halaman ini berisi informasi mengenai instansi, kategori, kerusakan, status, nomor surat, dan jumlah kategori kerusakan.



Gambar 4. 21 Tampilan Antar Muka Search Pengaduan

Tampilan antar muka halaman *search* pengaduan untuk kepala dinas sama dengan tampilan antar muka halaman *search* pengaduan untuk admin. Oleh karena itu gambar dan penjelasan yang dibuat hanya satu. Gambar 4.21 merupakan tampilan dari *search* pengaduan pada sistem pengaduan untuk admin. *Search* pengaduan pada halaman admin dapat dilakukan berdasarkan nama instansi atau nomor surat.



Edit Form

Instansi:
Inspektorat Kota Denpasar ▼

Kategori:
Hardware ▼

Kerusakan:
1 kali bunyi bip panjang ▼

Comment:
lambat sekali.

Status:
Belum diproses ▼

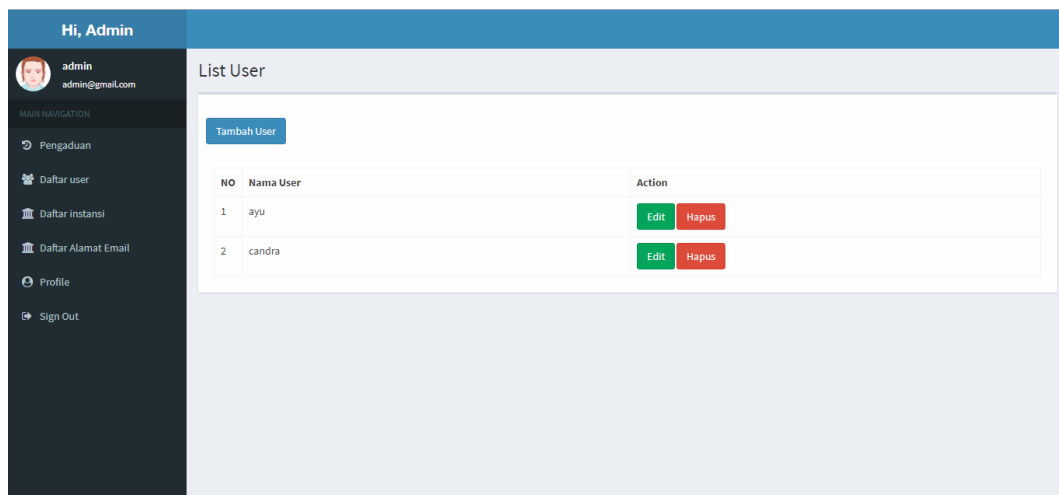
Nomor Surat:
212/3000/AREF

Cancel Edit

Gambar 4. 22 Tampilan Antar Muka Halaman Edit Pengaduan

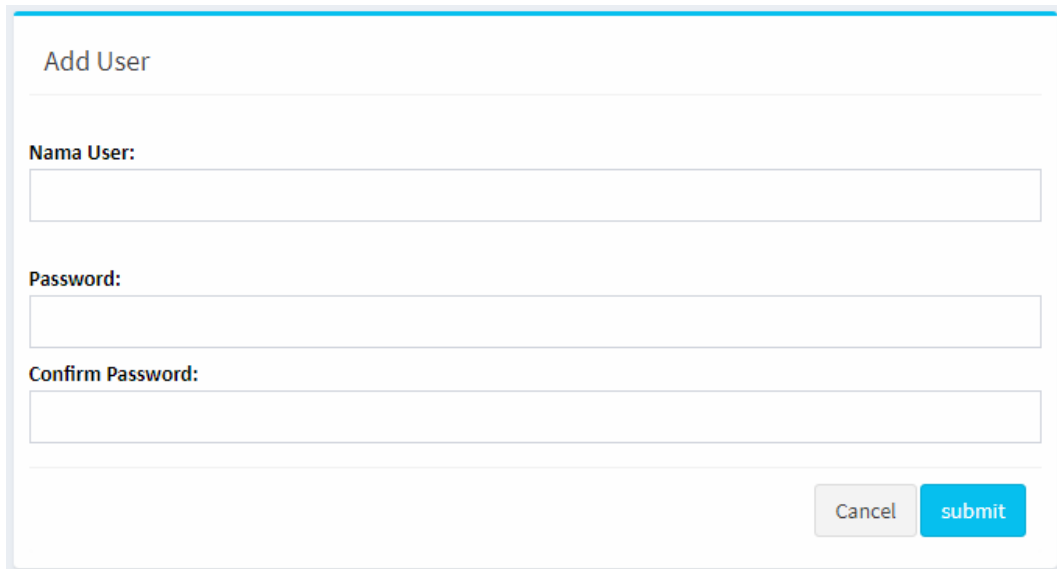
Gambar 4.22 merupakan tampilan dari halaman edit pengaduan pada sistem pengaduan untuk admin. Pada halaman edit pengaduan, admin dapat mengubah

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 admin ingin menambah list user maka admin dapat menekan tombol tambah user yang terdapat diatas tabel list user. Kemudian sistem akan menampilkan halaman tambah user seperti pada Gambar 4.24. Ketika admin ingin mengedit user maka admin dapat menekan tombol edit yang terdapat pada tabel list user. Kemudian sistem akan menampilkan halaman edit user seperti pada Gambar 4.25. Ketika admin ingin menghapus user maka admin dapat menekan tombol hapus yang terdapat pada tabel list user. Kemudian sistem akan menghapus data user yang ada pada sistem.



Add User

Nama User:

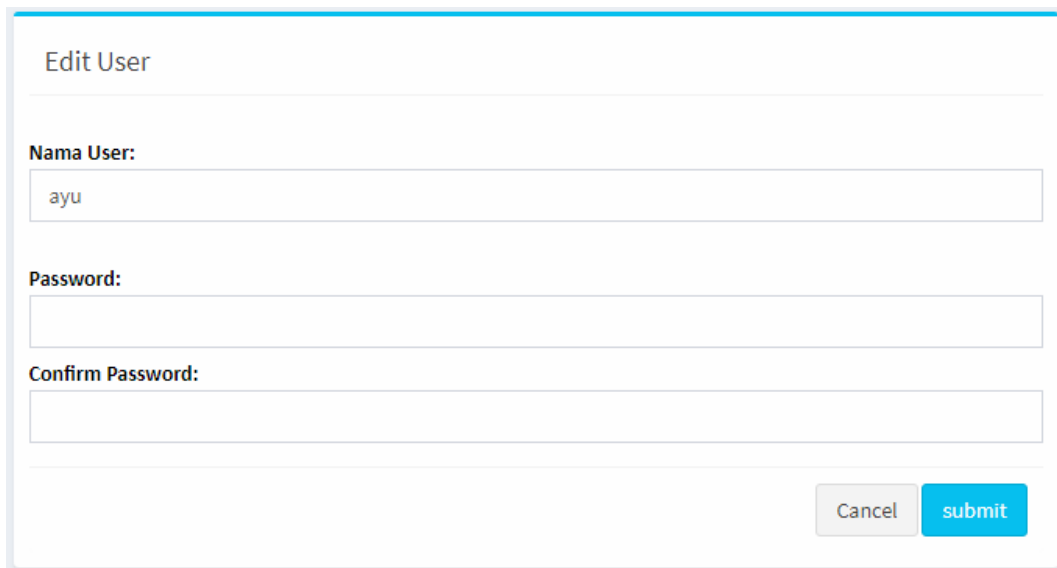
Password:

Confirm Password:

Cancel submit

Gambar 4. 24 Tampilan Antar Muka Halaman Tambah User

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



Edit User

Nama User:

ayu

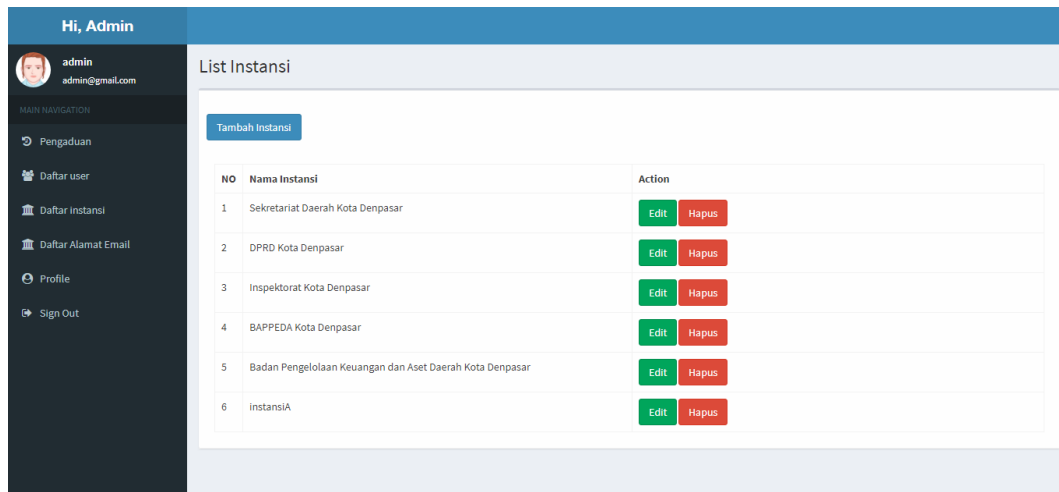
Password:

Confirm Password:

Cancel submit

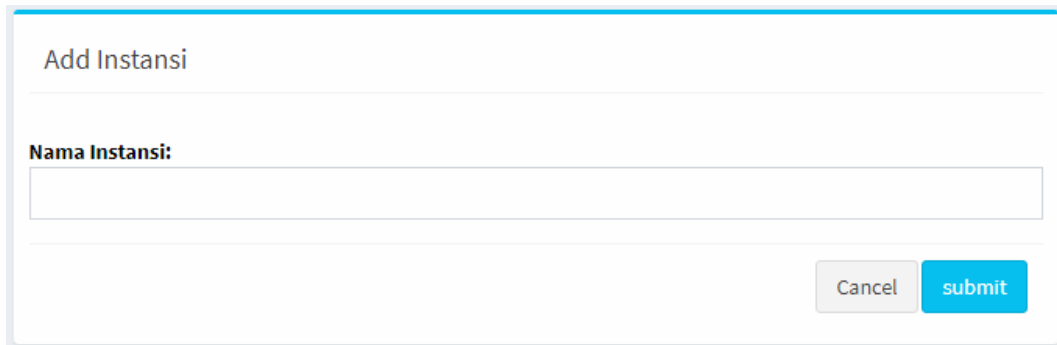
Gambar 4. 25 Tampilan Antar Muka Halaman Edit User

Gambar 4.25 merupakan tampilan dari halaman edit user pada sistem pengaduan untuk admin. Pada halaman edit user, admin dapat 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 admin ingin menambah instansi maka admin dapat menekan tombol tambah instansi yang terdapat diatas tabel list instansi. Kemudian sistem akan menampilkan halaman list instansi seperti pada Gambar 4.27. Ketika admin ingin mengedit instansi maka admin dapat menekan tombol edit yang terdapat pada tabel list instansi. Kemudian sistem akan menampilkan halaman edit instansi seperti pada Gambar 4.28. Ketika admin ingin menghapus instansi maka admin dapat menekan tombol hapus yang terdapat pada tabel list instansi. Kemudian sistem akan menghapus data instansi yang ada pada sistem.



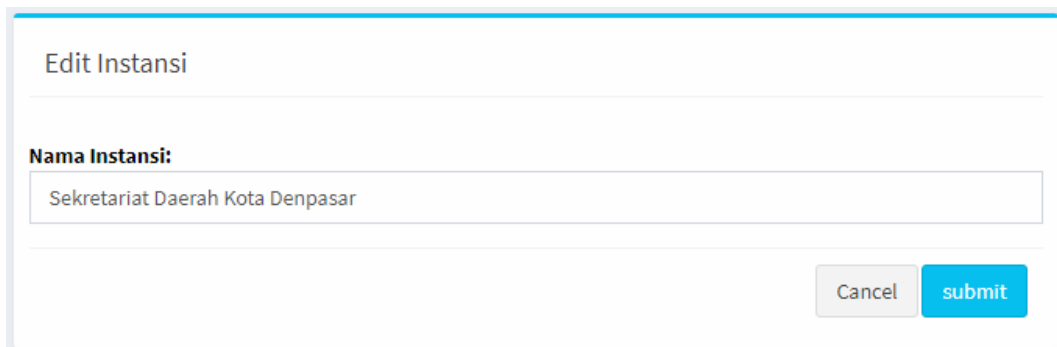
Add Instansi

Nama Instansi:

Cancel submit

Gambar 4. 27 Tampilan Antar Muka Halaman Tambah Instansi

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



Edit Instansi

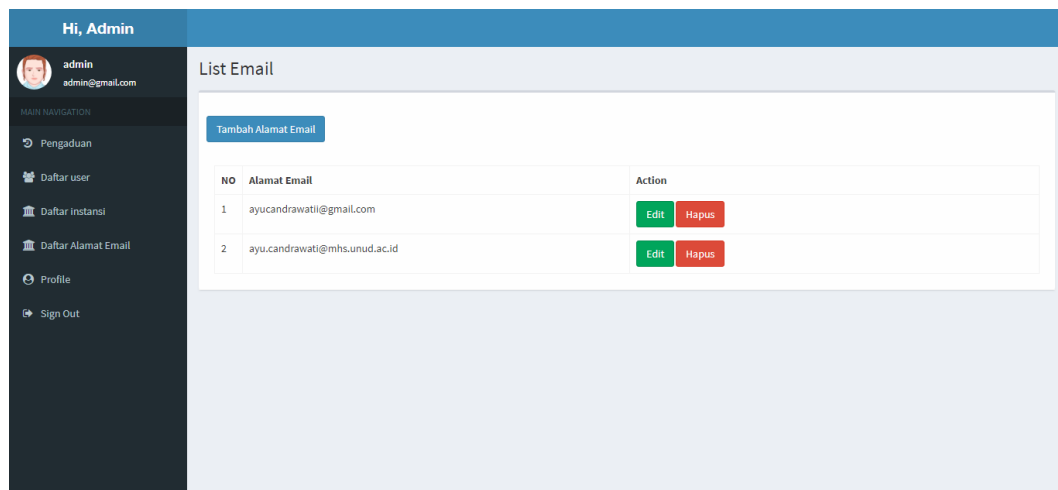
Nama Instansi:

Sekretariat Daerah Kota Denpasar

Cancel submit

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, admin dapat mengubah nama instansi pada *form* apabila terdapat kesalahan data.



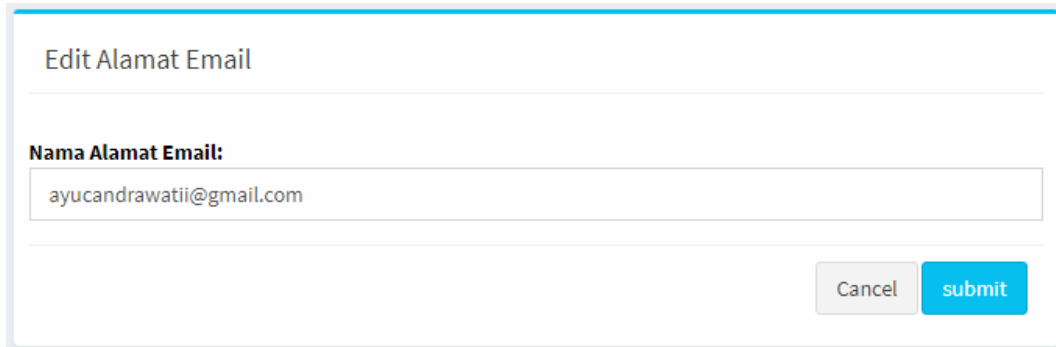
Gambar 4. 29 Tampilan Antar Muka Halaman List Email

Gambar 4.29 merupakan tampilan dari halaman list *email* pada sistem pengaduan untuk admin. Pada halaman list *email* terdapat navigasi disebelah kiri yang berisi menu pengaduan, daftar user, daftar instansi, daftar alamat *email*, profile, dan *signout*. Ketika admin ingin menambah alamat email yang bertujuan dalam proses mengirim pengaduan, maka admin dapat menekan tombol tambah alamat *email* yang terdapat diatas tabel list *email*. Kemudian sistem akan menampilkan halaman tambah list *email* seperti pada Gambar 4.30. Ketika admin ingin mengedit alamat *email* maka admin dapat menekan tombol edit yang terdapat pada tabel list alamat *email*. Kemudian sistem akan menampilkan halaman edit alamat *email* seperti pada Gambar 4.31. Ketika admin ingin menghapus alamat *email* maka admin dapat menekan tombol hapus yang terdapat pada tabel list alamat *email*. Kemudian sistem akan menghapus data alamat *email* yang ada pada sistem.

The screenshot shows a form titled 'Add Alamat Email'. It has a text input field for the email address. Below the input field, there are two buttons: 'Cancel' and 'submit'.

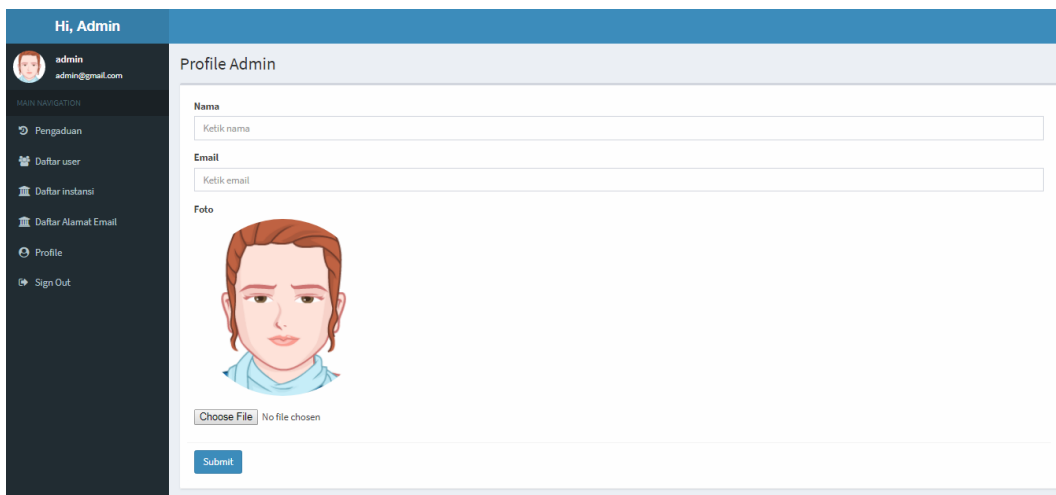
Gambar 4. 30 Tampilan Antar Muka Halaman Tambah List Email

Gambar 4.30 merupakan tampilan dari halaman tambah alamat *email* pada sistem pengaduan untuk admin. Ketika admin ingin menambah alamat *email* maka admin mengisi *form* dengan menginputkan nama *email*. Setelah admin selesai menginputkan data alamat *email* maka admin klik tombol submit untuk menyimpan data alamat *email*.



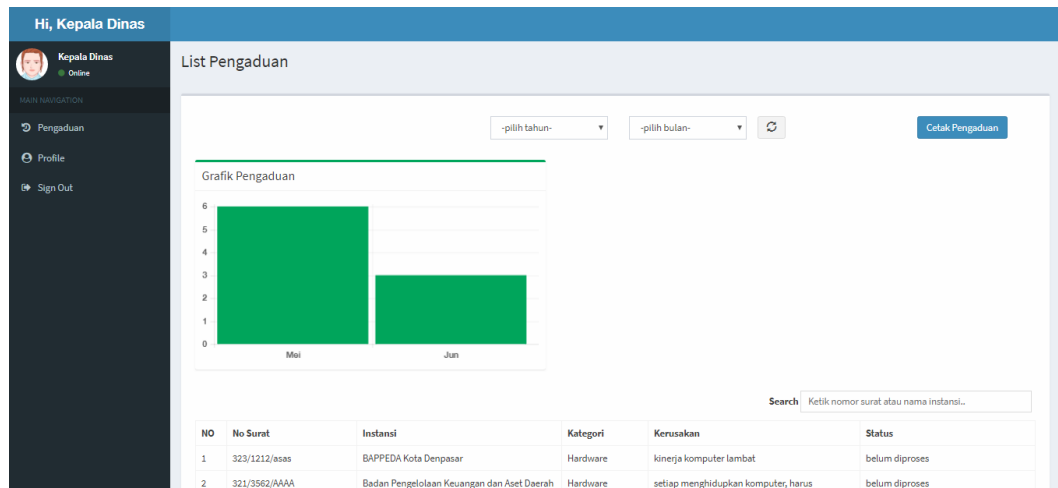
Gambar 4. 31 Tampilan Antar Muka Halaman Edit List *Email*

Gambar 4.31 merupakan tampilan dari halaman edit list *email* pada sistem pengaduan untuk admin. Pada halaman edit alamat *email*, admin dapat mengubah nama alamat *email* pada *form* apabila terdapat kesalahan data.



Gambar 4. 32 Tampilan Antar Muka Halaman Profile Admin

Gambar 4.32 merupakan tampilan dari halaman profile pada sistem pengaduan untuk admin. Pada halaman profile admin dapat mengubah nama, email, dan foto profile sesuai keinginan admin.



Gambar 4. 33 Tampilan Antar Muka Halaman Utama Kepala Dinas

Gambar 4.33 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. 34 Tampilan Antar Muka Halaman Profile Kepala Dinas

Gambar 4.34 merupakan tampilan dari halaman profile pada sistem pengaduan untuk kepala dinas. Pada halaman profile kepala dinas dapat mengubah nama, email, dan foto profile sesuai keinginan kepala dinas.

4.6 Pengujian Sistem

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

4.6.1 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, <i>technical support</i> , admin, kepala dinas	Terpenuhi
KF2	Pegawai	Terpenuhi
KF3	Pegawai	Terpenuhi
KF4	Pegawai	Terpenuhi
KF5	<i>Technical support</i>	Terpenuhi
KF6-A	<i>Technical support</i>	Terpenuhi
KF6-B	<i>Technical support</i>	Terpenuhi
KF6-C	<i>Technical support</i>	Terpenuhi
KF6-D	<i>Technical support</i>	Terpenuhi
KF7-A	<i>Technical support</i>	Terpenuhi
KF7-B	<i>Technical support</i>	Terpenuhi
KF7-C	<i>Technical support</i>	Terpenuhi
KF7-D	<i>Technical support</i>	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

Tabel 4.5 Checklist Pengujian Black Box (lanjutan)

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 4.5 *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* pengujian Tabel 4.5 akan dijelaskan lebih detail pada lampiran A.

4.6.2 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* dilakukan dengan mencatat waktu yang dibutuhkan sistem dalam memproses *query* sampai dengan tampilnya rekomendasi solusi yang didapat oleh sistem. Pengujian *stress test* dilakukan untuk mengetahui kemampuan perangkat lunak dalam memastikan bahwa aplikasi yang dibuat dapat menangani pengguna aplikasi sesuai jumlah yang ditargetkan.

1. Performance Test

Skenario dari *performance test* yang dilakukan pada sistem pengaduan yaitu melalui 3 tahap pengujian. Tahap pertama yaitu sistem diuji menggunakan *query* sejumlah 1 *term*. Tahap kedua yaitu sistem diuji menggunakan *query* sejumlah 2 *term*, dan tahap ketiga yaitu sistem diuji menggunakan *query* sejumlah 3 *term*. Setiap pengujian dilakukan 10 kali dengan menggunakan *term* yang dipilih secara

acak. Pada masing-masing tahap pengujian *query* dilakukan penambahan solusi pada database, dimulai dari 15 solusi hingga 75 solusi. Penambahan solusi bertujuan untuk mengetahui kinerja dari sistem apabila jumlah *term query* dan jumlah solusi ditambahkan.

Tabel 4. 6 Pengujian Menggunakan 1 *Term*

No	Total Solusi				
	15	30	45	60	75
1	0.25501	0.39102	0.66603	0.78804	1.02905
2	0.17125	0.35001	0.57603	0.77504	0.98105
3	0.26001	0.39402	0.66303	0.82504	1.03405
4	0.49602	0.80004	0.85904	1.17906	1.37107
5	0.57003	0.65703	1.12006	1.32307	1.37207
6	0.45402	0.57903	0.70104	1.00005	1.24207
7	0.39302	0.59503	0.80504	1.07406	1.21807
8	0.66403	0.68404	0.99405	1.18706	1.41007
9	0.47302	0.54803	0.91605	1.05806	1.17906
10	0.60903	0.72004	1.03905	1.24207	1.39832

Tabel 4. 7 Pengujian Menggunakan 2 *Term*

No	Total Solusi				
	15	30	45	60	75
1	0.21150	0.38502	0.60903	0.83104	0.99905
2	0.19500	0.35101	0.55203	0.85004	0.85004
3	0.21250	0.37702	0.60403	0.79404	1.01705
4	0.57402	0.74704	0.94105	1.21007	1.34707
5	0.63376	0.80604	0.96705	1.26607	1.36007
6	0.58601	0.76404	0.97305	1.18406	1.44208
7	0.43376	0.62103	0.86904	1.10006	1.24807
8	0.58975	0.86905	0.98105	1.26207	1.47008

Tabel 4.7 Pengujian Menggunakan 2 Term (lanjutan)

9	0.60750	0.79604	0.96205	1.26107	1.33307
10	0.56350	0.79804	0.98005	1.30207	1.38707

Tabel 4. 8 Pengujian Menggunakan 3 Term

No	Total Solusi				
	15	30	45	60	75
1	0.25401	0.38502	0.66203	0.92305	1.00202
2	0.19801	0.34601	0.58103	0.79604	0.96805
3	0.26701	0.44702	0.65103	0.89005	1.10506
4	0.56903	0.77004	0.93405	1.18406	1.47408
5	0.60203	0.80004	1.02305	1.22206	1.43008
6	0.67903	0.86804	1.00205	1.33807	1.49608
7	0.44902	0.71504	0.82904	1.06406	1.25307
8	0.67203	0.78504	1.02805	1.27907	1.4190
9	0.58003	0.79804	1.05006	1.30307	1.3620
10	0.53403	0.80204	0.96605	1.28107	1.5160

Pengujian yang dilakukan pada Tabel 4.6, Tabel 4.7, dan Tabel 4.8 akan didapatkan rata-rata dari waktu pengujian menggunakan 1 *term*, 2 *term*, dan 3 *term*. Berikut merupakan rata-rata hasil pengujian tersebut.

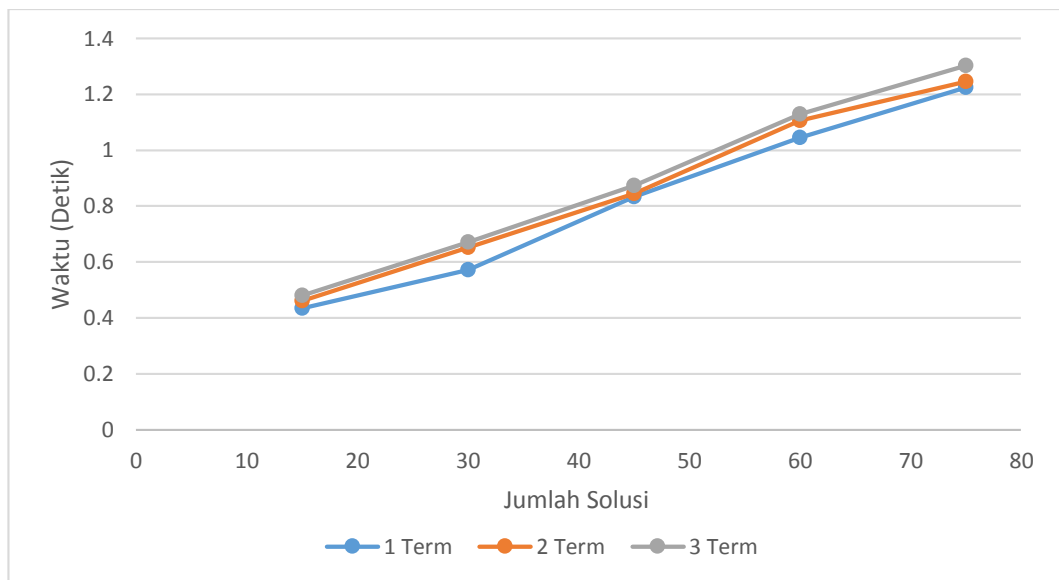
Tabel 4. 9 Rata-rata Waktu Hasil Pengujian

Jumlah Solusi	Jumlah Term		
	1 Term	2 Term	3 Term
15	0.43454	0.46073	0.48042
30	0.57182	0.65143	0.67163
45	0.83394	0.84384	0.87264
60	1.04515	1.10605	1.12806

Tabel 4.9 Rata-rata Waktu Hasil Pengujian

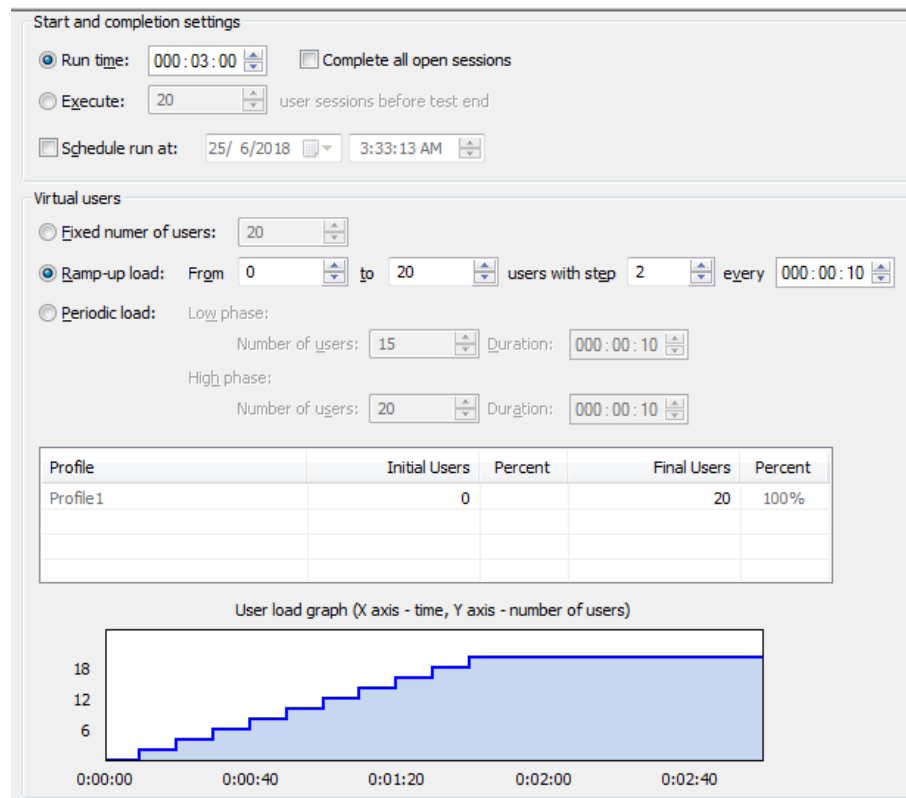
75	1.22348	1.24536	1.30254
Rata-rata	0.82179	0.86148	0.89106

Dari tabel 4.9 dapat dilihat bahwa semakin banyak jumlah term pada *qery* yang digunakan saat melakukan proses pencarian solusi, maka semakin meningkat waktu yang diperlukan sistem untuk menampilkan solusi. Selain itu juga semakin banyak jumlah solusi yang tersimpan pada *database*, maka semakin meningkat waktu yang diperlukan sistem untuk menampilkan solusi. Peningkatan waktu proses pencarian solusi dapat dilihat pada gambar 4.35.

**Gambar 4. 35 Grafik Peningkatan Waktu Proses Pencarian Solusi**

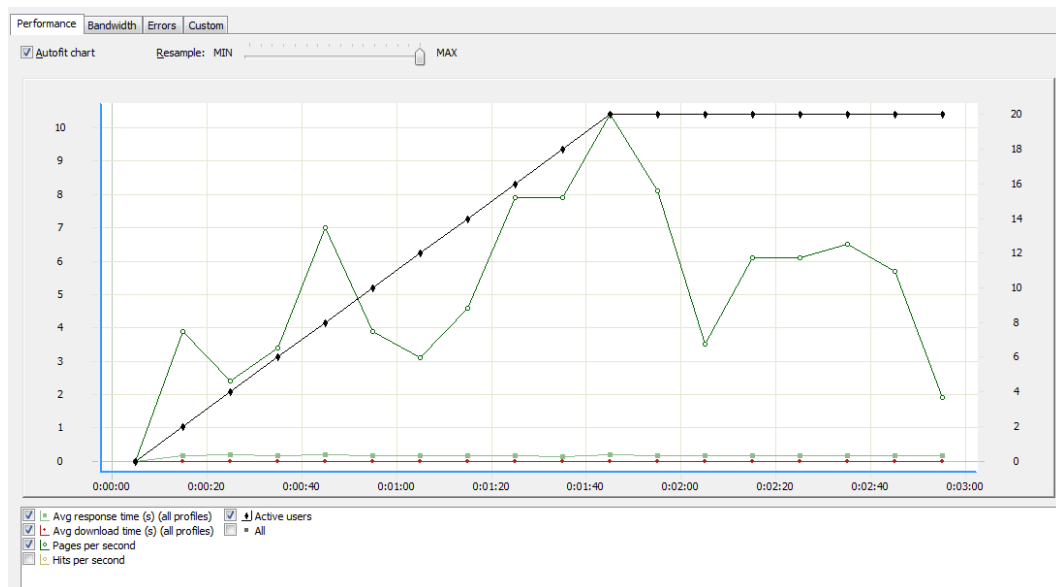
2. Stress Test

Stress test mengacu pada test yang menekankan pada pengujian yang menekankan pada kekuatan, ketersediaan, dan kesalahan penanganan dibawah beban berat. Tool yang digunakan pada *stress test* sistem pengaduan yaitu dengan menggunakan aplikasi WAPT. Skenario *stress test* dapat dilihat pada gambar 4.36.



Gambar 4.36 Skenario *Stress Test* Sistem

Pada gambar 4.36 merupakan scenario pengujian pada sistem pengaduan. Skenario *stress test* yaitu sistem akan diberikan beban berupa 20 *user* yang akan mengakses *dashboard* sistem selama 3 menit secara bersamaan. Setiap *user* mengakses sistem bertambah sejumlah 2 *user* setiap 10 detik. Namun jumlah *user* pada sistem pengaduan dapat meningkat sesuai dengan jumlah pegawai yang melakukan pengaduan. Maka diperlukan peningkatan kualitas kecepatan *server* untuk menampung jumlah *user* yang lebih besar lagi. Hasil statistik dari *stress test* terlampir pada lampiran C. Hasil data statistik tersebut dalam bentuk grafik yang dapat dilihat pada gambar 4.37.



Gambar 4. 37 Grafik *Stress Test* Sistem

Pada gambar 4.37 garis berwarna hitam merupakan jumlah *user* yang aktif pada sistem. Garis berwarna oranye merupakan jumlah *user* download pada sistem. Garis berwarna hijau tua merupakan jumlah halaman yang diakses *user* setiap detik pada sistem. Garis berwarna hijau muda merupakan waktu merespon permintaan pada setiap aktivitas *user*. Berdasarkan gambar 4.37, dalam waktu 3 menit data *average response* time cukup rendah, maka dari itu dapat disimpulkan bahwa sistem masih dapat menangani dengan baik.

4.6.3 Pengujian Terhadap Respon Pengguna

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

Tabel 4. 10 Bobot Nilai Kuisisioner

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

Tabel 4. 11 Bobot Nilai Kuisisioner

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 kuisisioner oleh 10 pengguna dan setiap kuisisioner 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

5.1 Kesimpulan

Berdasarkan pada penelitian yang telah dilakukan, maka diambil kesimpulan meliputi

1. Hasil survey respon pengguna melalui kuisioner menyatakan bahwa 77% pengguna merasa sangat setuju dan 23% pengguna merasa setuju dapat disimpulkan bahwa sistem ini mampu memberikan kemudahan dalam melakukan pengaduan secara *online* dan mampu mempermudah *technical support* dalam mencari solusi dari kerusakan komputer.
2. Pengujian performansi sistem memberikan hasil dimana semakin banyak jumlah *term* pada *query* yang digunakan, maka semakin meningkat waktu yang dibutuhkan sistem dalam menampilkan solusi kerusakan dan semakin banyak jumlah solusi yang ada pada *database*, maka semakin meningkat waktu yang dibutuhkan sistem dalam menampilkan solusi kerusakan. *Stress testing* memberikan hasil bahwa dalam waktu 3 menit data *average response time* cukup rendah, maka dari itu sistem masih dapat menangani dengan baik.

5.2 Saran

Berdasarkan pada penelitian yang telah dilakukan, maka saran yang dapat disampaikan untuk pengembangan sistem lebih lanjut yaitu :

1. Mencoba mengembangkan teknik *stemming* pada tahap *preprocessing* untuk meningkatkan waktu *indexing* pada solusi
2. Sistem dapat melakukan pencarian solusi dengan memperhitungkan sinonim kata yang terdapat pada *query* saat melakukan proses pencarian

DAFTAR PUSTAKA

- Agusta. L. 2009. *Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia*. Konferensi Nasional Sistem dan Informatika 2009; Bali, November 14, 2009.
- Firdaus. A., Ernawati., Vatesia. A. 2014. *Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani Dan Metode Cosine Similarity*. Jurnal Teknologi Informasi, Volume 10 Nomor 1, ISSN 1414-9999
- Fitri. M. 2013. *Kombinasi Tf-Idf, Perancangan Sistem Temu Balik Informasi Dengan Metode Pembobotan Kombinasi Tf-Idf Untuk Pencarian Dokumen Berbahasa Indonesia*. Tanjungpura, 2013, pp. 1-6
- Jumardi. A., Solichin, Achmad. 2016. *Prototype Aplikasi Layanan Pengaduan Masyarakat Berbasis Android Dan Web Service*. Jurnal TELEMATIKA MKOM Vol.8 No.1, Maret 2016
- Kurniawan. A., Solihin. F., dan Hastarita. F. 2014. *Perancangan dan Pembuatan Aplikasi Pencarian Informasi Beasiswa dengan Menggunakan Cosine Similarity*. Jurnal SimanteC Volume 4, Nomor 2.
- Mandala, R., dan Setiawan, H. 2002. *Peningkatan Performansi Sistem Temu Kembali Informasi dengan Perluasan Query secara Otomatis*. Bandung, Indonesia: Institut Teknologi Bandung.
- Masya. F., Elvina., Simanjuntak. F. M. 2012. *Sistem Pengaduan Masyarakat Pada Divisi Humas Polri Berbasis Web*. Seminar Nasional Aplikasi Teknologi Informasi. ISSN: 1907-5022
- Niswara. B., Putri. R., Hidayat. N. 2018. *Rekomendasi Pemilihan Paket Personal Computer Menggunakan Metode AHP-TOPSIS*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 5, Mei 2018, hlm. 1998-2007
- Nurdiana. O., Jumadi, Nursantika. D. 2016. *Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian*

Terjemah Al-Qur'an Dalam Bahasa Indonesia. JOIN, Volume I Nomor 1, ISSN 2527-9165

- Nurjanah. M., Hamdani, dan Astuti. I., Fitri. 2013. *Penerapan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) untuk Text Mining*. Jurnal Informatika Volume 8, Nomor 3.
- Pradnyana. G. A. dan Sanjaya. N. A. 2012. *Cosine Similarity, Perancangan Dan Implementasi Automated Document Integration Dengan Menggunakan Algoritma Complete Linkage Agglomerative Hierarchical Clustering*. Vol. 5, (2), pp. 1-10, September 2012.
- Pressman, Roger. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. Yogyakarta: Penerbit Andi.
- Pusat Bahasa Departement Pendidikan Nasional. 2008. *KBBI*. Jakarta: Departemen Pendidikan Nasional
- Robertson, S.E. 2004. *Understanding inverse document frequency: on theoretical arguments for IDF*. Journal of Documentation, 60, 5.
- Shabir. F., Rachman. M. 2016. *Rekomendasi Pembelian Personal Komputer Dengan Metode Ranked Clustering*. Jurnal Ilmiah ILKOM Volume 8 Nomor 2 (Agustus 2016)
- Wahyuni. R.T., Prastiyanto. D., Suprpto. E. 2017. *Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi*. Jurnal Teknik Elektro Vol. 9 No. 1

LAMPIRAN A

BLACK BOX TESTING

Tabel Lampiran A. 1 Pengujian Validasi Input Pada Proses Login

Kode Kebutuhan: KF1			
Kasus: <i>Melakukan pengujian validasi input pada proses login</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Pengguna memasukkan input <i>string username</i> dan <i>password</i> yang benar pada <i>form</i> dan menekan tombol <i>login</i>	Sistem mencocokkan data yang dimasukkan dan mengarahkan pengguna ke halaman utama	Sesuai Harapan
2	Pengguna memasukkan input <i>string username</i> dan <i>password</i> yang salah pada <i>form</i> dan menekan tombol <i>login</i>	Sistem mencocokkan data yang dimasukkan dan kembali ke halaman <i>login</i>	Sesuai Harapan

Tabel Lampiran A. 2 Pengujian Hak Akses Proses Login

Kode Kebutuhan: KF1			
Kasus: <i>Melakukan pengujian hak akses proses login</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Pengguna melakukan <i>login</i> dengan memilih hak akses sebagai pegawai	Sistem memberi hak akses pengguna sebagai pegawai dan menuju halaman utama pegawai	Sesuai Harapan
2	Pengguna melakukan <i>login</i> dengan memilih hak akses sebagai <i>technical support</i>	Sistem memberi hak akses pengguna sebagai pegawai dan menuju halaman utama <i>technical support</i>	Sesuai Harapan
3	Pengguna melakukan <i>login</i> dengan memilih hak akses sebagai admin	Sistem memberi hak akses pengguna sebagai pegawai dan menuju halaman utama admin	Sesuai Harapan

Tabel Lampiran A. 2 Pengujian Hak Akses Proses Login (lanjutan)

4	Pengguna melakukan <i>login</i> 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: <i>Melakukan pengujian proses sign out</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Pengguna sebagai pegawai menekan tombol <i>sign out</i> pada menu navigasi	Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman <i>login</i>	Sesuai Harapan
2	Pengguna sebagai <i>technical support</i> menekan tombol <i>sign out</i> pada menu navigasi	Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman <i>login</i>	Sesuai Harapan
3	Pengguna sebagai admin menekan tombol <i>sign out</i> pada menu navigasi	Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman <i>login</i>	Sesuai Harapan
4	Pengguna sebagai kepala dinas menekan tombol <i>sign out</i> pada menu navigasi	Pengguna akan keluar dari sistem dan pengguna diarahkan ke halaman <i>login</i>	Sesuai Harapan

Tabel Lampiran A. 4 Pengujian Proses Pengaduan

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

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

Kode Kebutuhan: KF3			
Kasus: <i>Melakukan pengujian proses melihat history pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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 Status Pengaduan

Kode Kebutuhan: KF4			
Kasus: <i>Melakukan pengujian proses melihat status pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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: <i>Melakukan pengujian proses mendapatkan solusi</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> 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 Kerusakan

Kode Kebutuhan: KF6-A			
Kasus: <i>Melakukan pengujian proses menambah kerusakan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> menekan tombol tambah kerusakan pada halaman daftar kerusakan	Sistem menampilkan <i>form</i> tambah kerusakan	Sesuai Harapan

Tabel Lampiran A. 8 Pengujian Proses Menambah Kerusakan (lanjutan)

2	<i>Technical support</i> mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah kerusakan dan menekan tombol submit	Sistem menyimpan inputan ke <i>database</i> dan menampilkan halaman daftar kerusakan	Sesuai Harapan
3	<i>Technical support</i> tidak mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah kerusakan dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> harus diisi inputan	Sesuai Harapan
4	<i>Technical support</i> menekan tombol <i>cancel</i> pada <i>form</i> daftar kerusakan	Sistem menampilkan halaman daftar kerusakan	Sesuai Harapan

Tabel Lampiran A. 9 Pengujian Proses Melihat Daftar Kerusakan

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

Tabel Lampiran A. 10 Pengujian Proses Mengedit Kerusakan

Kode Kebutuhan: KF6-C			
Kasus: <i>Melakukan pengujian proses mengedit kerusakan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> menekan tombol edit pada halaman daftar kerusakan	Sistem menampilkan <i>form</i> edit kerusakan	Sesuai Harapan

Tabel Lampiran A. 10 Pengujian Proses Mengedit Kerusakan (lanjutan)

2	<i>Technical support</i> mengedit semua <i>input field</i> yang ada pada <i>form</i> edit kerusakan dan menekan tombol submit	Sistem menyimpan inputan yang telah diubah ke <i>database</i> dan menampilkan halaman daftar kerusakan	Sesuai Harapan
3	<i>Technical support</i> mengosongkan semua <i>input field</i> yang ada pada <i>form</i> edit kerusakan dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> harus diisi inputan	Sesuai Harapan
4	<i>Technical support</i> tidak mengubah data kerusakan dan menekan tombol submit	Sistem akan menyimpan data yang tidak dirubah dan menampilkan halaman daftar kerusakan	Sesuai Harapan
5	<i>Technical support</i> menekan tombol <i>cancel</i> pada <i>form</i> edit kerusakan	Sistem menampilkan halaman daftar kerusakan	Sesuai Harapan

Tabel Lampiran A. 11 Pengujian Proses Menghapus Kerusakan

Kode Kebutuhan: KF6-D			
Kasus: <i>Melakukan pengujian proses menghapus kerusakan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> menekan tombol hapus pada halaman daftar kerusakan	Sistem menampilkan pesan untuk konfirmasi hapus data kerusakan	Sesuai Harapan
2	<i>Technical support</i> menekan tombol OK	Sistem menghapus data kerusakan dan menampilkan halaman daftar kerusakan	Sesuai Harapan
3	<i>Technical support</i> 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: <i>Melakukan pengujian proses menambah solusi</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> menekan tombol tambah solusi pada halaman list solusi	Sistem menampilkan <i>form</i> tambah solusi	Sesuai Harapan
2	<i>Technical support</i> mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah solusi dan menekan tombol submit	Sistem menyimpan inputan ke <i>database</i> dan menampilkan halaman list solusi	Sesuai Harapan
3	<i>Technical support</i> tidak mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah solusi dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> harus diisi inputan	Sesuai Harapan
4	<i>Technical support</i> menekan tombol <i>cancel</i> pada <i>form</i> tambah solusi	Sistem menampilkan halaman list solusi	Sesuai Harapan

Tabel Lampiran A. 13 Pengujian Proses Melihat List Solusi

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

Tabel Lampiran A. 14 Pengujian Proses Menghapus Solusi

Kode Kebutuhan: KF7-D			
Kasus: <i>Melakukan pengujian proses menghapus solusi</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> menekan tombol hapus pada halaman list solusi	Sistem menampilkan pesan untuk konfirmasi hapus data solusi	Sesuai Harapan

Tabel Lampiran A. 14 Pengujian Proses Menghapus Solusi (lanjutan)

2	<i>Technical support</i> menekan tombol OK	Sistem menghapus data solusi dan menampilkan halaman list solusi	Sesuai Harapan
3	<i>Technical support</i> 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: <i>Melakukan pengujian proses menambah pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Admin menekan tombol tambah pengaduan pada halaman pengaduan admin	Sistem menampilkan <i>form</i> tambah pengaduan	Sesuai Harapan
2	Admin mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah pengaduan dan menekan tombol submit	Sistem menyimpan inputan ke <i>database</i> dan menampilkan halaman pengaduan admin	Sesuai Harapan
3	Admin tidak mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah pengaduan dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> harus diisi inputan	Sesuai Harapan
4	Admin menekan tombol <i>cancel</i> pada <i>form</i> tambah pengaduan	Sistem menampilkan halaman pengaduan admin	Sesuai Harapan

Tabel Lampiran A. 16 Pengujian Proses Melihat Pengaduan

Kode Kebutuhan: KF8-B			
Kasus: <i>Melakukan pengujian proses melihat daftar pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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: <i>Melakukan pengujian proses mengedit pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Admin menekan tombol edit pada halaman pengaduan	Sistem menampilkan <i>form</i> edit pengaduan	Sesuai Harapan
2	Admin mengedit semua <i>input field</i> yang ada pada <i>form</i> edit pengaduan dan menekan tombol submit	Sistem menyimpan inputan yang telah diubah ke <i>database</i> dan menampilkan halaman pengaduan admin	Sesuai Harapan
3	Admin mengosongkan semua <i>input field</i> yang ada pada <i>form</i> edit pengaduan dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> 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 <i>cancel</i> pada <i>form</i> edit pengaduan	Sistem menampilkan halaman pengaduan admin	Sesuai Harapan

Tabel Lampiran A. 18 Pengujian Proses Menghapus Pengaduan

Kode Kebutuhan: KF8-D			
Kasus: <i>Melakukan pengujian proses menghapus pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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: <i>Melakukan pengujian proses mengirim pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Admin menekan tombol kirim pada halaman pengaduan admin	Sistem mengirim pengaduan ke alamat <i>email</i> yang sudah ditentukan pada sistem	Sesuai Harapan

Tabel Lampiran A. 20 Pengujian Proses Mencetak Pengaduan

Kode Kebutuhan: KF8-F			
Kasus: <i>Melakukan pengujian proses mencetak pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Admin memilih tahun dan bulan pada menu pilih tahun dan pilih bulan, lalu klik tombol <i>refresh</i> 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: <i>Melakukan pengujian proses mencari pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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: <i>Melakukan pengujian proses melihat grafik pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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: <i>Melakukan pengujian proses menambah daftar user</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Admin menekan tombol tambah user pada halaman daftar user	Sistem menampilkan <i>form</i> tambah user	Sesuai Harapan
2	Admin mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah user dan menekan tombol submit	Sistem menyimpan inputan ke <i>database</i> dan menampilkan halaman daftar user	Sesuai Harapan
3	Admin tidak mengisi semua <i>input field</i> yang ada pada <i>form</i> tambah user dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> harus diisi inputan	Sesuai Harapan
4	Admin menekan tombol <i>cancel</i> pada <i>form</i> tambah user	Sistem menampilkan halaman daftar user	Sesuai Harapan

Tabel Lampiran A. 24 Pengujian Proses Melihat Daftar User

Kode Kebutuhan: KF9-B			
Kasus: <i>Melakukan pengujian proses melihat daftar user</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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: <i>Melakukan pengujian proses mengedit daftar user</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Admin menekan tombol edit pada halaman daftar user	Sistem menampilkan <i>form</i> edit user	Sesuai Harapan
2	Admin mengedit semua <i>input field</i> yang ada pada <i>form</i> edit user dan menekan tombol submit	Sistem menyimpan inputan yang telah diubah ke <i>database</i> dan menampilkan halaman daftar user	Sesuai Harapan
3	Admin mengosongkan semua <i>input field</i> yang ada pada <i>form</i> edit user dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> 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 <i>cancel</i> pada <i>form</i> edit user	Sistem menampilkan halaman daftar user	Sesuai Harapan

Tabel Lampiran A. 26 Pengujian Proses Menghapus Daftar User

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

Tabel Lampiran A. 28 Pengujian Proses Melihat Daftar Instansi

Kode Kebutuhan: KF10-B			
Kasus: <i>Melakukan pengujian proses melihat daftar instansi</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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: <i>Melakukan pengujian proses mengedit daftar instansi</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Admin menekan tombol edit pada halaman daftar instansi	Sistem menampilkan <i>form</i> edit instansi	Sesuai Harapan
2	Admin mengedit semua <i>input field</i> yang ada pada <i>form</i> edit instansi dan menekan tombol submit	Sistem menyimpan inputan yang telah diubah ke <i>database</i> dan menampilkan halaman daftar instansi	Sesuai Harapan
3	Admin mengosongkan semua <i>input field</i> yang ada pada <i>form</i> edit instansi dan menekan tombol submit	Sistem menampilkan pesan pada <i>form input field</i> bahwa <i>form input field</i> 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 <i>cancel</i> pada <i>form</i> edit instansi	Sistem menampilkan halaman daftar instansi	Sesuai Harapan

Tabel Lampiran A. 30 Pengujian Proses Menghapus Daftar Instansi

Kode Kebutuhan: KF10-D			
Kasus: <i>Melakukan pengujian proses menghapus daftar instansi</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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. 31 Pengujian Proses Mendapatkan Informasi Dan Cetak Pengaduan

Kode Kebutuhan: KF11-A			
Kasus: <i>Melakukan pengujian proses mendapatkan informasi dan cetak pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Kepala dinas memilih tahun dan bulan pada menu pilih tahun dan pilih bulan, lalu klik tombol <i>refresh</i> 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. 32 Pengujian Proses Melihat Grafik Pengaduan

Kode Kebutuhan: KF11-B			
Kasus: <i>Melakukan pengujian proses melihat grafik pengaduan</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	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. 33 Pengujian Proses Mengubah Nama Profile

Kode Kebutuhan: KF12-A			
Kasus: <i>Melakukan pengujian proses mengubah nama profile</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> , kepala dinas, dan admin klik menu profile pada navigasi	Sistem menampilkan halaman profile masing-masing user	Sesuai Harapan
2	<i>Technical support</i> , 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. 34 Pengujian Proses Mengubah Alamat *Email* Profile

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

Tabel Lampiran A. 35 Pengujian Proses Mengubah Foto Profile

Kode Kebutuhan: KF12-C			
Kasus: <i>Melakukan pengujian proses mengubah foto profile</i>			
No.	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Technical support</i> , 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 kuisisioner 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 kuisisioner 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 <i>online</i>				
2	Fitur solusi dari kerusakan komputer memudahkan <i>technical support</i> dalam mencari solusi kerusakan komputer				
3	Penjelasan yang keluar ketika fitur solusi diklik sudah sesuai dengan masalah kerusakan komputer yang diadu				
4	<i>Technical support</i> diberi kemudahan dalam mengelola daftar kerusakan				
5	<i>Technical support</i> diberi kemudahan dalam mengelola list solusi				
6	Admin diberi kemudahan dalam mengelola daftar pengaduan				
7	Admin diberi kemudahan dalam mengelola daftar user				
8	Admin diberi kemudahan dalam mengelola daftar instansi				
9	Kepala dinas diberi kemudahan dalam rekap pengaduan				
10	<i>Technical support</i> , admin, dan kepala dinas diberi kemudahan dalam mengelola profile masing-masing user				

Catatan:

LAMPIRAN C

HASIL STATISTIK *STRESS TEST*

Test execution parameters:

Test status: finished

Test started at: 25/6/2018 3:21:21 AM

Test finished at: 25/6/2018 3:24:21 AM

Scenario name:

Test run comment:

Test executed by: Toshiba

Test executed on: TOSHIBA-PC

Test duration: 0:03:00

Virtual users: 0 - 20

Summary

Profile	Sessions performed	Sessions with errors	Pages performed	Pages with errors	Hits performed	Hits with errors	Total KBytes sent	Total KBytes received
Profile 1	905	448	1,372	448	12,887	2,322	4,957	255,521
Total	905	448	1,372	448	12,887	2,322	4,957	255,521

Number of active users

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00
Profile1	2	6	8	12	16	20	20	20	20	20
Total	2	6	8	12	16	20	20	20	20	20

Sessions

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	19	27	36	33	60	92	59	29	65	485	905
Total	19	27	36	33	60	92	59	29	65	485	905

Pages

Profile	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	39	58	70	70	125	183	116	61	126	524	1,372
Total	39	58	70	70	125	183	116	61	126	524	1,372

Hits

Profile	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	523	769	952	931	1,670	2,474	1,573	813	1,715	1,467	12,887
Total	523	769	952	931	1,670	2,474	1,573	813	1,715	1,467	12,887

Sessions per second

Profile	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	0.95	1.35	3.6	1.65	3	4.6	2.95	2.9	3.25	24.3	5.03
Total	0.95	1.35	3.6	1.65	3	4.6	2.95	2.9	3.25	24.3	5.03

Pages per second

Profile	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	1.95	2.9	7	3.5	6.25	9.15	5.8	6.1	6.3	3.8	5.13
Total	1.95	2.9	7	3.5	6.25	9.15	5.8	6.1	6.3	3.8	5.13

Hits per second

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	22.3	32.6	81.2	39.5	70.8	105	67.0	68.5	73.0	43.1	58.7
Total	22.3	32.6	81.2	39.5	70.8	105	67.0	68.5	73.0	43.1	58.7

Pages with errors

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	0	0	0	0	0	0	0	0	0	448	448
Total	0	0	0	0	0	0	0	0	0	448	448

Hits with errors

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	78	116	140	141	253	371	234	128	256	605	2,322
Total	78	116	140	141	253	371	234	128	256	605	2,322

KBytes sent

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	203	298	370	361	647	960	611	314	666	527	4,957
Total	203	298	370	361	647	960	611	314	666	527	4,957

KBytes received

Profile	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	10,712	15,684	19,595	19,010	34,153	50,811	32,343	16,607	35,308	21,298	255,521
Total	10,712	15,684	19,595	19,010	34,153	50,811	32,343	16,607	35,308	21,298	255,521

Response time, sec

Name	Time	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00
Profile1.page_1: http://localhost/skripsi/	Min	0.11	0.11	0.1	0.1	0.1	0.1	0.07	0.09	0.11	0.08
	Avg	0.16	0.19	0.17	0.15	0.15	0.18	0.16	0.18	0.15	0.15
	Avg90	0.22	0.39	0.36	0.21	0.19	0.39	0.23	0.25	0.21	0.2
	Max	0.28	0.99	0.98	0.3	0.33	1.46	0.32	0.33	0.32	0.29
Profile1.page_2: http://localhost/skripsi/Pengaduan/index	Min	0.13	0.12	0.1	0.12	0.1	0.08	0.09	0.12	0.1	0.11
	Avg	0.17	0.17	0.19	0.18	0.17	0.17	0.16	0.17	0.17	0.18
	Avg90	0.22	0.22	0.39	0.26	0.23	0.24	0.23	0.23	0.22	0.25
	Max	0.26	0.28	1.04	0.41	0.45	0.39	0.39	0.29	0.31	0.36

HTTP reply codes

Code	Request	Number
Profile1	-	12,887
200 OK	Profile1.All	10,565
200 OK	Profile1.page_1: http://localhost/skripsi/ (1)	3,724
	Profile1.page_2: http://localhost/skripsi/Pengaduan/index (2)	6,841
403 Forbidden	Profile1.page_1: http://localhost/skripsi/ (1)	934
	Profile1.page_2: http://localhost/skripsi/Pengaduan/index (2)	914
500 Internal server error	Profile1.page_1: http://localhost/skripsi/ (1)	439
	Profile1.page_2: http://localhost/skripsi/Pengaduan/index (2)	9
Network error	Profile1.page_1: http://localhost/skripsi/ (1)	12
	Profile1.page_2: http://localhost/skripsi/Pengaduan/index (2)	14

HTTP errors %

Profile	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	14.9	15.1	14.7	15.0	15.0	14.8	14.7	15.0	14.7	40.9	17.8
Total	14.9	15.1	14.7	15.0	15.0	14.8	14.7	15.0	14.7	40.9	17.8

Socket errors %

Profile	0:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	0	0	0	0.11	0.18	0.2	0.13	0.74	0.23	0.34	0.2
Total	0	0	0	0.11	0.18	0.2	0.13	0.74	0.23	0.34	0.2

Timeouts %

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	0	0

Receiving speed (kbit/s)

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	4,285	6,274	15,676	7,604	13,661	20,325	12,937	13,286	14,123	8,519	11,357
Total	4,285	6,274	15,676	7,604	13,661	20,325	12,937	13,286	14,123	8,519	11,357

Sending speed (kbit/s)

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	81.2	119	296	144	259	384	245	251	266	211	220
Total	81.2	119	296	144	259	384	245	251	266	211	220

Receiving per user speed (kbit/s)

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	2,142	1,267	1,960	705	899	1,066	647	664	706	426	1,019
Total	2,142	1,267	1,960	705	899	1,066	647	664	706	426	1,019

Sending per user speed (kbit/s)

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	40.6	24.1	37.0	13.4	17.0	20.1	12.2	12.5	13.3	10.5	19.6
Total	40.6	24.1	37.0	13.4	17.0	20.1	12.2	12.5	13.3	10.5	19.6

Total errors %

Profile	0:00:00-0:00:18	0:00:18-0:00:36	0:00:36-0:00:54	0:00:54-0:01:12	0:01:12-0:01:30	0:01:30-0:01:48	0:01:48-0:02:06	0:02:06-0:02:24	0:02:24-0:02:42	0:02:42-0:03:00	Total
Profile1	14.9	15.1	14.7	15.1	15.1	15.0	14.9	15.7	14.9	41.2	18.0
Total	14.9	15.1	14.7	15.1	15.1	15.0	14.9	15.7	14.9	41.2	18.0

URLs**Profile "Profile1"**

Name	Server/Port	Page	User thinking time
Profile1.page_1: http://localhost/skripsi/	http://localhost:80	/skripsi/	0-0
Profile1.page_2: http://localhost/skripsi/Pengaduan/index	http://localhost:80	/skripsi/Pengaduan/index	0-0