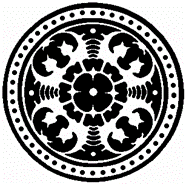
**Sistem Pengaduan Kerusakan Komputer dan rekomendasi solusi menggunakan algoritma cosine similarity**

**SKRIPSI**

****

**LUH GEDE AYU CANDRAWATI**

**NIM. 1408605018**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS UDAYANA**

**BUKIT JIMBARAN**

**2018**

**SURAT PERNYATAAN KEASLIAN KARYA ILMIAH**

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

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

Nama : Luh Gede Ayu Candrawati

NIM : 1408605018

Program Studi : Teknik Informatika

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

Nomor telp/HP: 082188686362

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

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

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

Denpasar, ……….. 2018

Yang membuat pernyataan

Luh Gede Ayu Candrawati

NIM. 1408605018

# LEMBAR PENGESAHAN TUGAS AKHIR

Judul :

Nama : Luh Gede Ayu Candrawati

NIM : 1408605018

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

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

# ABSTRAK

**Kata kunci:**

Title :

Name : Luh Gede Ayu Candrawati

Registration : 1408605018

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

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

# ABSTRACT

**Keywords:**

# KATA PENGANTAR

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

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

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

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

Bukit Jimbaran, 2018

Penulis

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# DAFTAR LAMPIRAN

# BAB I

**PENDAHULUAN**

## Latar Belakang

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

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

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

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

## Rumusan Masalah

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

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

## Tujuan Penelitian

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

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

## Batasan Masalah

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

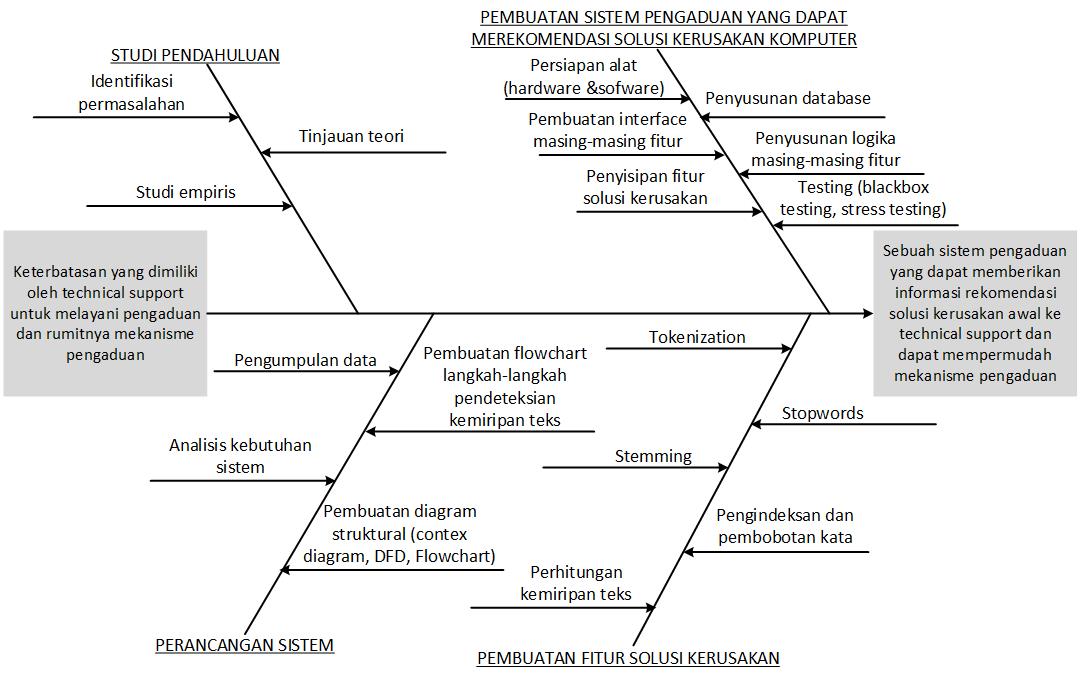
1. Rekomendasi solusi dari kerusakan komputer menggunakan metode *text similarity* dan algoritma *cosine similarity*
2. Sistem yang dibangun merupakan sistem berbasis web *responsive*.
3. Data solusi kerusakan yang digunakan pada sistem ini yaitu mencakup kerusakan *hardware, software,* dan jaringan.

## Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah membantu *technical support* yang ada di dinas kota Denpasar untuk mendapatkan solusi cepat dari kerusakan komputer. Selain itu juga membantu pegawai yang ada di dinas kota Denpasar untuk melakukan pengaduan agar pekerjaan pegawai menjadi lebih cepat dan efisien.

## Metodologi Penelitian

Pada bagian ini menjelaskan bagaimana langkah-langkah yang akan dilakukan dalam melakukan penelitian ini dalam sistem pengaduan kerusakan komputer dan rekomendasi solusi menggunakan algoritma *cosine similarity.* Langkah-langkah tersebut akan digambarkan pada diagram *fishbone* seperti pada Gambar x.x



Gambar x. x Diagram *Fish Bone* Penelitian

Gambar x.x merupakan *fish bone* penelitian. Pada *fish bone* ini terdapat empat bagian yaitu bagian studi pendahuluan, perancangan sistem, pembuatan sistem pengaduan yang dapat merekomendasi solusi kerusakan komputer dan pembuatan fitur solusi kerusakan.

Pada penelitian ini, peneliti mencoba untuk mengatasi permasalahan yang ada. Tahap pertama peneliti melakukan identifikasi permasalahan yang ada yaitu mengenai keterbatasan-keterbatasan *technical support*, dimana nantinya akan dibuatkan solusi dari permasalahan tersebut. Peneliti juga mempelajari jurnal penelitian terdahulu yang terkait dengan permasalahan yang diteliti dan mempelajari materi yang ada terkait pemasalahan tersebut.

Tahap kedua peneliti merancang sistem yang dapat membantu permasalahan pada keterbatasan-keterbatasan *technical support*. Peneliti menganalisis kebutuhan sistem yang akan dibangun. Sistem yang dibangun berorientasi objek, maka dibuat perancangan sistem berupa diagram pemodelan sistem diantaranya adalah *context diagram*, *data flow diagram*, dan *flowchart*. Berdasarkan analisis kebutuhan dan perancangan sistem, peneliti mengumpulkan data yang akan menjadi dasar dalam pembuatan sistem ini. Sistem yang dirancang terdapat fitur solusi kerusakan komputer, oleh karena itu akan dirancang *flowchart* langkah-langkah pendeteksian kemiripan teks*.*

Tahap ketiga adalah pembuatan fitur solusi kerusakan komputer yang ada dalam sistem. Tahap ini diawali dengan proses *tokenization* untuk proses pemecahan kalimat menjadi kata. Kemudian proses penghapusan kata-kata yang tidak penting *(stopwords).* Setelah semua kata-kata yang tidak penting dihapus, proses selanjutnya adalah proses *stemming*. Poses *stemming* bertujuan merubah kata yang berisi awalan, imbuhan, dan akhiran menjadi kata dasar. Setelah semua kata menjadi kata dasar, masing-masing kata diindeks dan diberi bobot. Proses terakhir adalah perhitungan kemiripan teks menggunakan algoritma *cosine similarity.*

Tahap keempat adalah pembuatan sistem pengaduan yang dapat merekomendasi solusi kerusakan komputer. Porses pertama adalah persiapan alat yang dibutuhkan baik dari segi *hardware* maupun *software*. Proses selanjutnya yaitu penyusunan *database* menggunakan *entity relationship diagram*. Setelah itu, peneliti mendesain *interface* masing-masing fitur. Setelah semua fitur pada sistem berjalan, dilanjutkan dengan implementasi logika pada masing-masing fitur. Kemudian, menyisipkan fitur solusi kerusakan komputer. Proses terakhir yaitu melakukan pengujian pada sistem, dimana terdapat dua pengujian yang dilakukan yaitu *blackbox testing* dan *stress testing*.

### Metode Pengumpulan Data

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

### Metode Pengembangan Sistem

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

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



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

Menurut Pressman (2002), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Diberi nama waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Tahapan dalam Waterfall Model:

1. Communication (Project Initiation & Requirements Gathering)

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

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

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

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

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

1. Modeling (Analysis & Design)

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

1. Construction (Code & Test)

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

1. Deployment (Delivery, Support, Feedback)

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



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

### Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan dengan mencari informasi mengenai

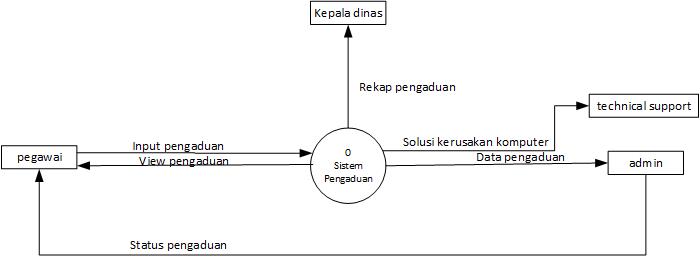
### Desain Penelitian

Penelitian ini berjudul ”Sistem Pengaduan Kerusakan Komputer Dan Rekomendasi Solusi Menggunakan Algoritma Cosine Similarity”. Sistem ini akan dibangun berbasis web *responsive*. Pengguna sistem ini yaitu pegawai dinas, *technical support,* kepala dinas, dan admin.

Kebutuhan sistem secara fungsional pada penelitian ini akan dijabarkan sebagai berikut :

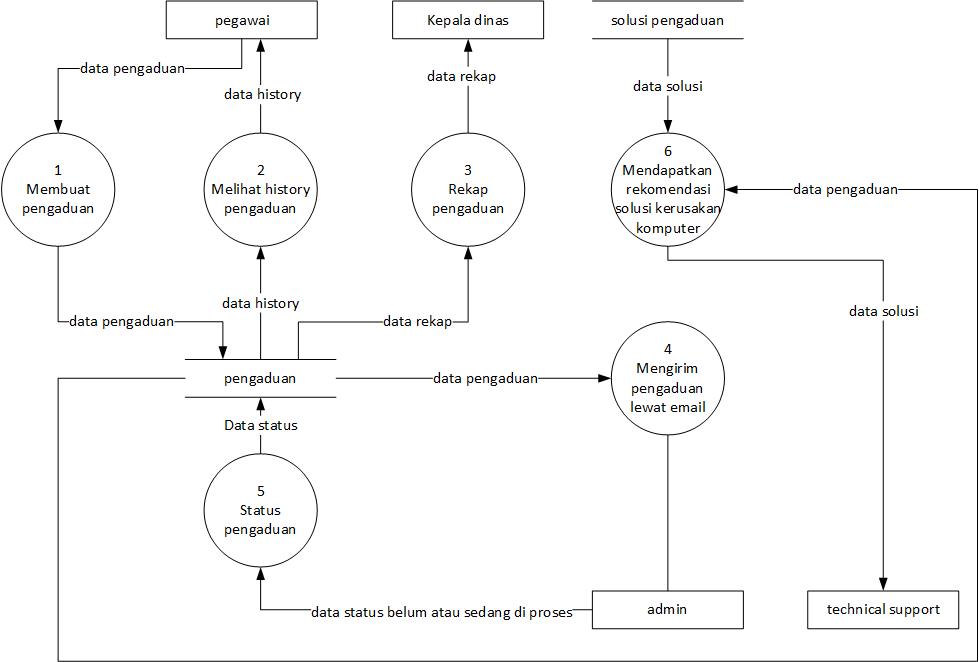
1. Pegawai mampu melakukan pengaduan secara *online*
2. Pegawai mendapatkan informasi mengenai status pengaduan, apakah pengaduan belum atau sedang diproses
3. Pegawai mendapatkan informasi mengenai *history* pengaduan yang pernah dilakukan
4. Kepala dinas mendapatkan laporan rekap pengaduan
5. Admin dapat melihat *history* pengaduan, melakukan pengaduan, edit pengaduan, hapus pengaduan, dan mengirim pengaduanberupa surat melalui *email*
6. *Technical support* mendapatkan solusi kerusakan komputer berdasarkan data pengaduan yang diinput oleh user

Berikut adalah *Contex Diagram* dan *DFD* *Level 0* dari penilitian yang terdapat pada Gambar x.x dan Gambar x.x di bawah ini :



Gambar x. x *Contex Diagram* Sistem

Gambar x.x merupakan *Context Diagram* Sistem yang berisi sistem pengaduan dan empat terminator (pegawai, kepala dinas, *technical support*, dan admin). Pada diagram ini pegawai dapat menginput pengaduan ke sistem pengaduan dan sistem pengaduan dapat memperlihatkan pengaduan kepada pegawai. Kepala dinas mendapat rekap pengaduan dari sistem pengaduan. *Technical support* mendapat solusi kerusakan komputer dari sistem pengaduan. Admin mendapat data pengaduan dari sistem pengaduan.



Gambar x. x *DFD Level 0* Sistem

Gambar x.x merupakan *DFD Level 0* Sistem. Pada diagram ini terdapat enam proses, empat terminator, dan dua tabel*.* Proses pertama yaitu dapat membuat data pengaduan yang kemudian disimpan ke dalam tabelpengaduan. Proses yang kedua yaitu dapat melihat *history* data pengaduan yang disimpan pada tabelpengaduan. Peranan pegawai pada sistem ini yaitu pegawai dapat melihat *history* pengaduan, CRUD perlu?. Proses ketiga yaitu sistem dapat merekap data pengaduan yang didapat dari tabelpengaduan dan dapat dilihat oleh kepala dinas. Proses keempat yaitu sistem dapat mengirim data pengaduan lewat email dari tabel pengaduan yang dikirim oleh admin. Data pengaduan yang dikirim adalah pengaduan yang tidak dapat ditangani oleh *technical support.* Proses kelima yaitu status pengaduan, dimana admin dapat mengubah status pegaduan di sistem apakah status pengaduan belum atau sedang diproses. Proses keenam yaitu mendapatkan rekomendasi solusi kerusakan berdasarkan tabel pengaduan dan tabel solusi pengaduan untuk *technical support*.

### Metode Pengembangan Sistem

### Implementasi dll… (seputar sistem yang akan dibuat dan masih berupa tulisan-tulisan dan teori-teori)

### Pengujian Sistem

# BAB II

**TINJAUAN PUSTAKA**

## Tinjauan Studi

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

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

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

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

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

(Andi Jumardi, Achmad Solichin, 2016)

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

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

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

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

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

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

(Ogie Nurdiana, Jumadi, dan Dian Nursantika 2016)

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

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

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

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

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

## Pengaduan

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

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

## Model Pengembangan Waterfall

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

1. Communication (Project Initiation & Requirements Gathering)

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

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

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

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

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

1. Modeling (Analysis & Design)

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

1. Construction (Code & Test)

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

1. Deployment (Delivery, Support, Feedback)

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



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

## Preprocessing Data, TF-IDF, dan Cosine Similarity

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



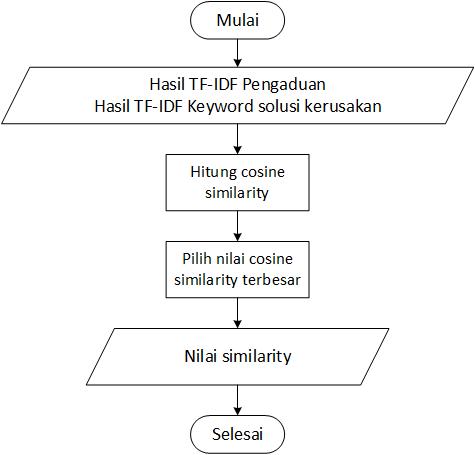
Gambar x.x *Flowchart* *Preprocessing*

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

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

Gambar 7. 6 *Flowchart* TF-IDF

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



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

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

## Pengujian Perangkat Lunak

### Pengujian *Black* *Box*

*Black box testing* adalah pengujian sesuai atau tidaknya *output* yang diharapkan oleh pengguna sistem. Pengujian ini tidak menguji proses yang terjadi pada sistem, melainkan berfokus pada keperluan fungsional dengan menemukan kesalahan dalam beberapa kategori seperti kesalahan fungsi-fungsi dan kesalahan *interface*. *Black box testing* tidak dilakukan oleh *programmer* yang membuat sistem, melainkan *user* sebagai tester sistem. Pada pengujian ini terdapat skenario pengujian dimana menilai kesesuaian antara pengaduan mengenai permasalahan kerusakan komputer yang diinput oleh user dengan solusi yang ditawarkan oleh sistem. Untuk memastikan kesesuaian antara permasalahan kerusakan komputer dengan solusi yang ditawarkan, *user* yang bertugas sebagai *tester. User* yang terlibat sebagai *tester* adalah pegawai, *technical support,* kepala dinas, dan admin. Pegawai, kepala dinas, dan *technical support* bertugas *testing* *front-end web* pada sistem, dan juga *technical support* bertugas *testing* kesesuaian antara pengaduan mengenai permasalahan kerusakan komputer yang diinput oleh user dengan solusi yang ditawarkan oleh sistem. Admin bertugas *testing* *back-end web* pada sistem. Berikut skenario pengujian *black box testing* :

Tabel 7. 1 Skenario pengujian *Black Box Testing*

|  |  |  |
| --- | --- | --- |
| Kelas Uji |  | |
| Nama Kelas Uji |  | |
| Tanggal Pengujian |  | |
| Skenario | | |
|  | | |
| Hasil Yang Diharapkan | Pengamatan | Kesimpulan |
|  |  |  |

### *Stress* *Testing*

*Stress testing* adalah jenis pengujian dengan memberikan beban lebih pada sistem yang bertujuan untuk mengetahui kemampuan maksimum dari sistem. Pengujian ini biasanya menguji banyak akses pada sistem dalam satu waktu. Contohnya adalah saat melakukan akses tunggal untuk sebuah *request* dalam 1 detik, hal tersebut tentunya tidak menganggu *server* dan tidak berpengaruh pada *server.* Namun ketika melakukan akses ganda dengan melakukan 100 *request* pada server dalam satu waktu, maka akan terjadi *delay* pada server yang tidak dapat diterima oleh *user*. Pada pengujian ini menggunakan bantuan aplikasi WAPT.

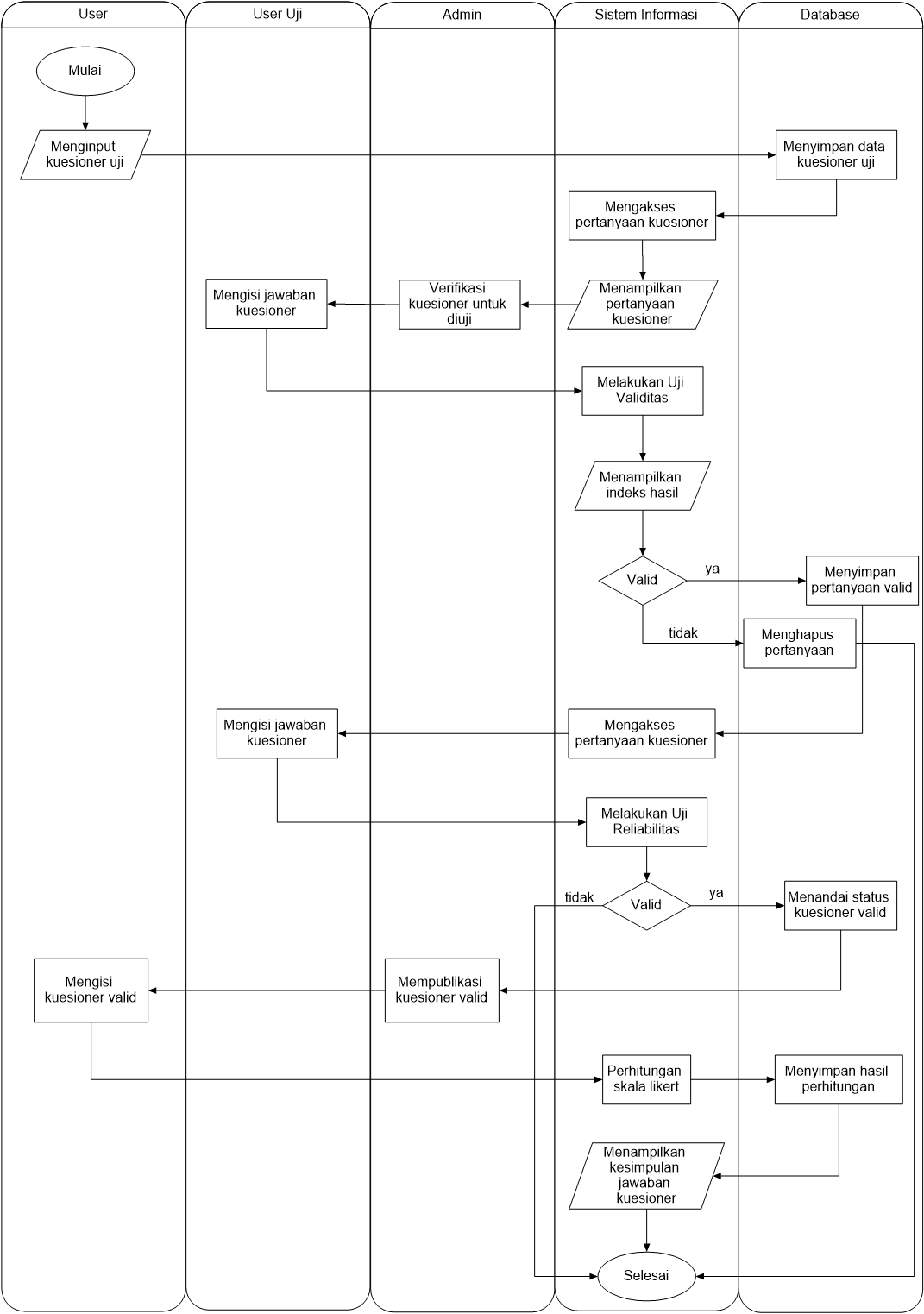
# BAB III

**ANALISIS DAN PERANCANGAN**

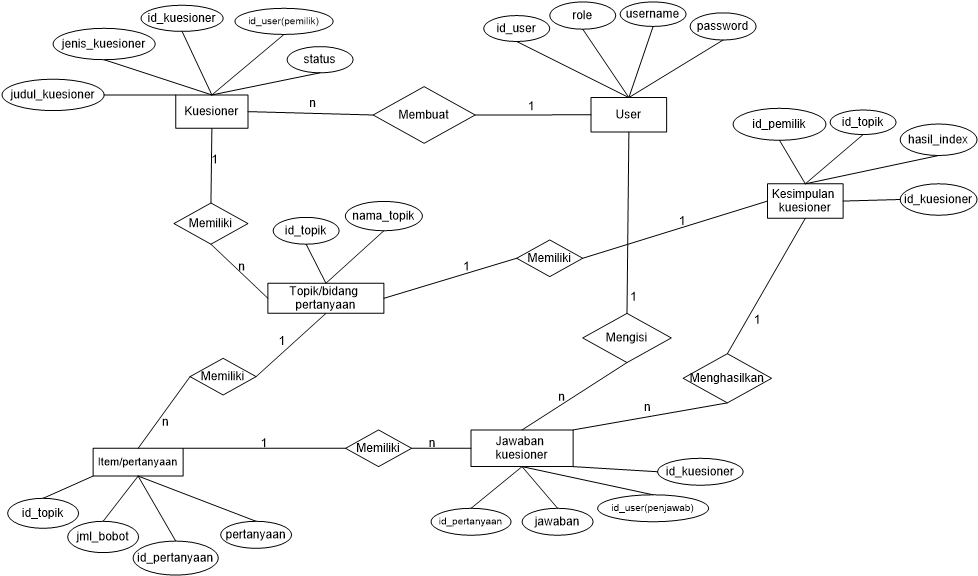
## Kebutuhan Fungsional dan Non-Fungsional

## Pengolahan-Pengolahan Data (Uji Validitas, Uji Reliabilitas, Skala Likert)\*\*

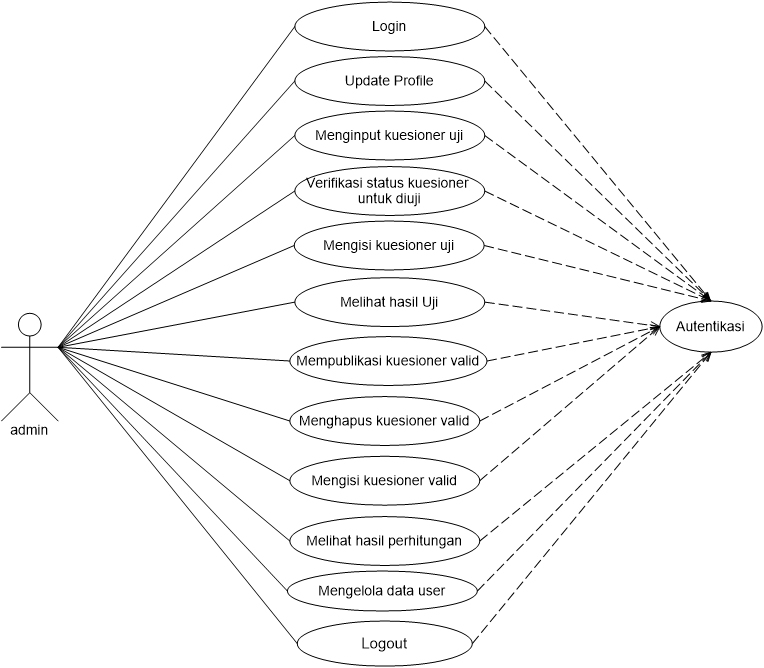
## Flowchart Proses Bisnis

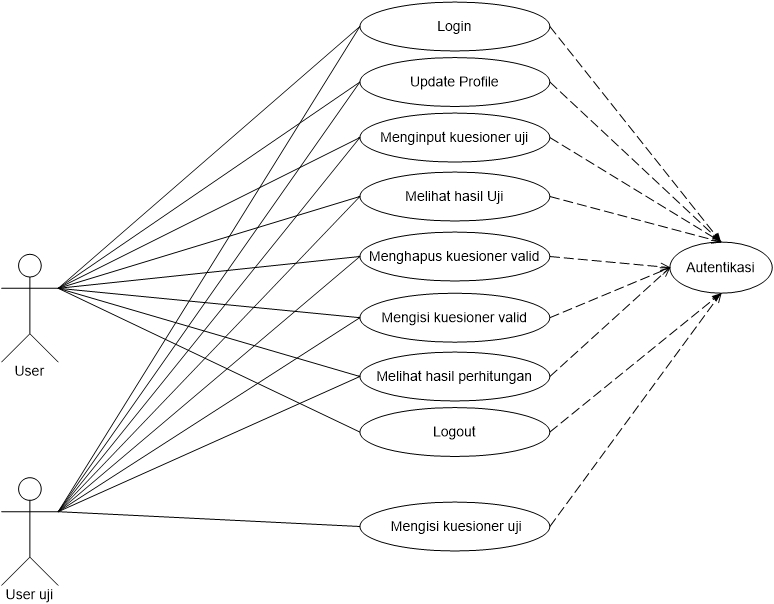


## ERD



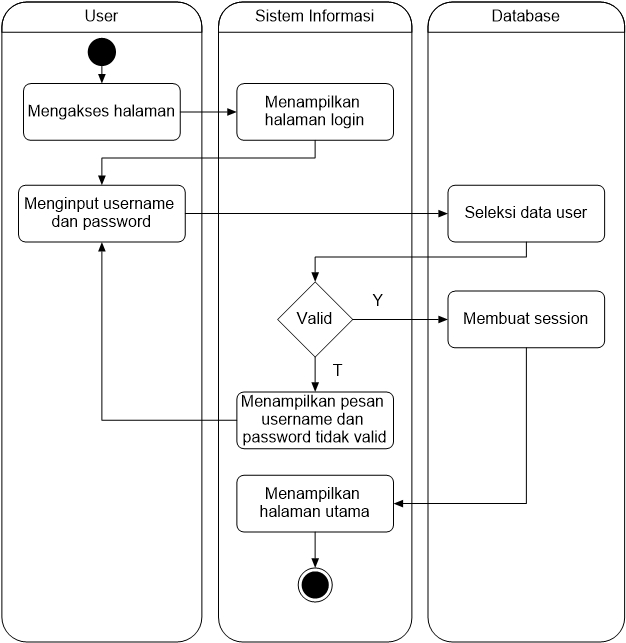
## Use Case



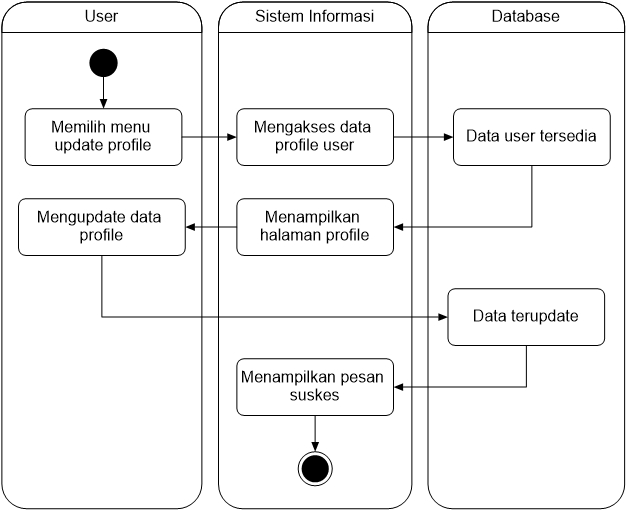


## Activity Diagram

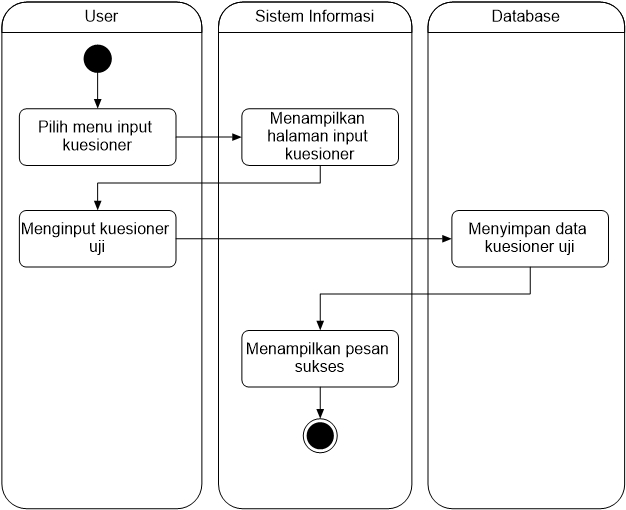
1. Login



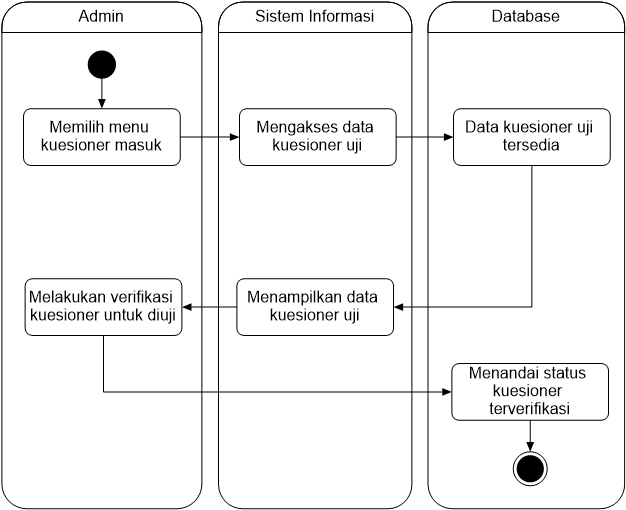
1. Update Profile



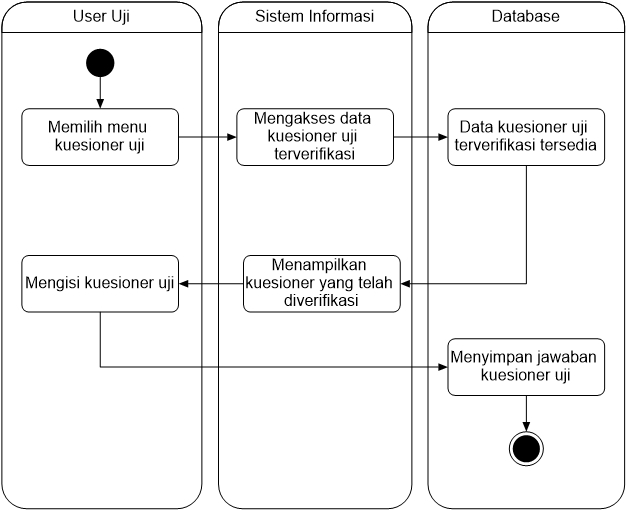
1. Input Kuesioner



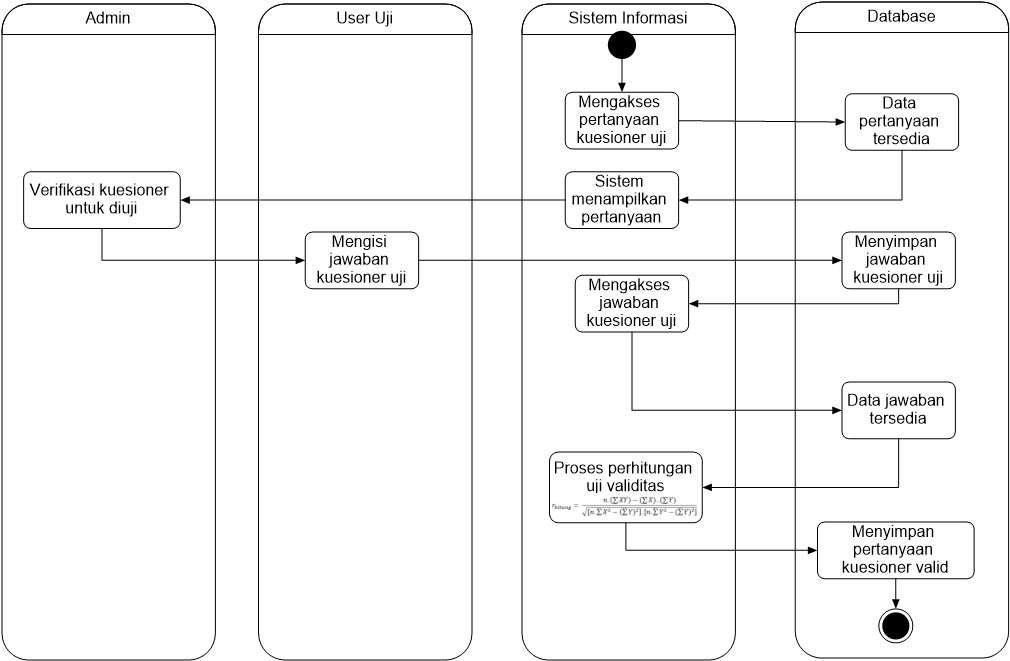
1. Verifikasi Kuesioner Uji



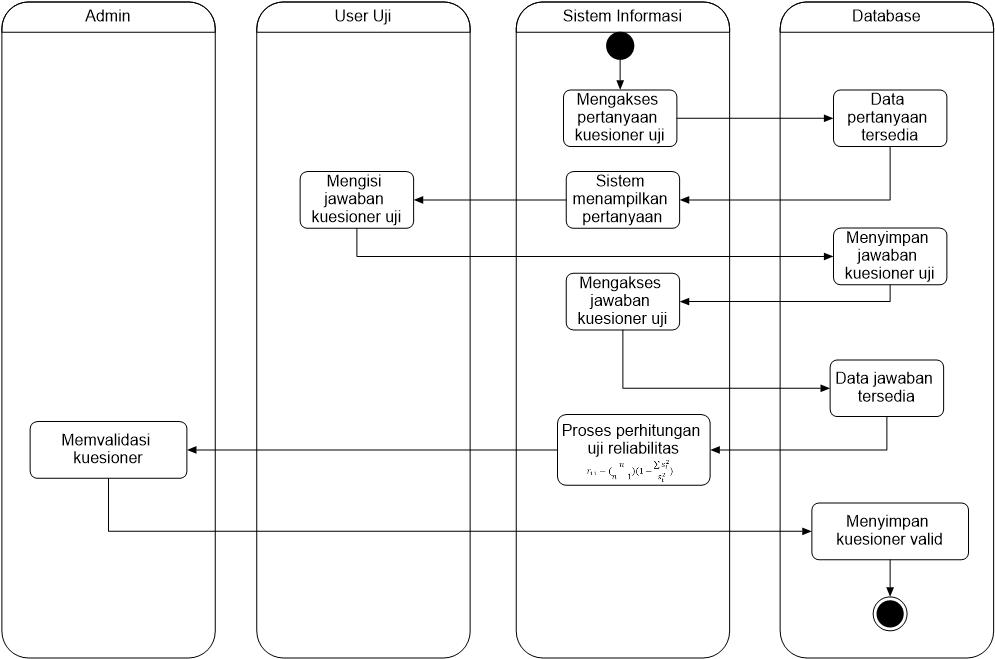
1. Mengisi Kuesioner Uji



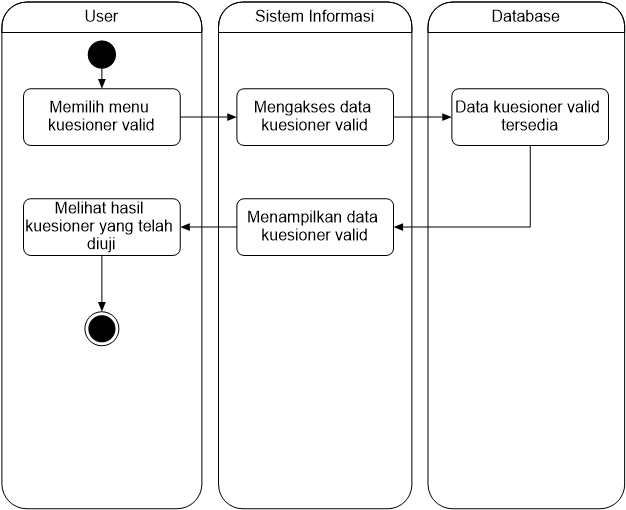
1. Uji Validitas



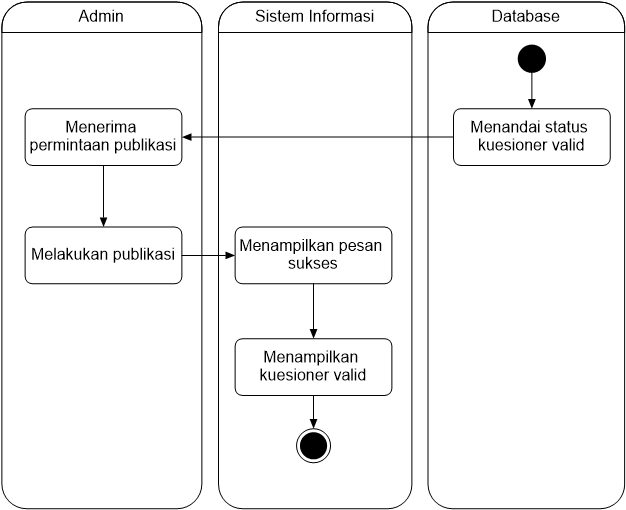
1. Uji Reliabilitas



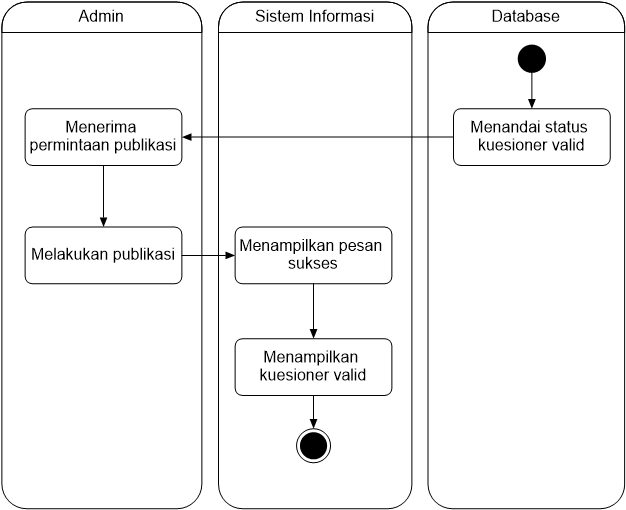
1. Melihat Hasil Uji Kuesioner



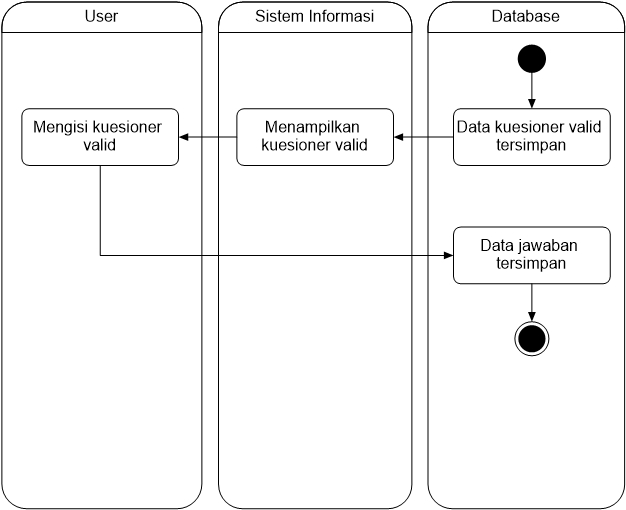
1. Melakukan Publikasi



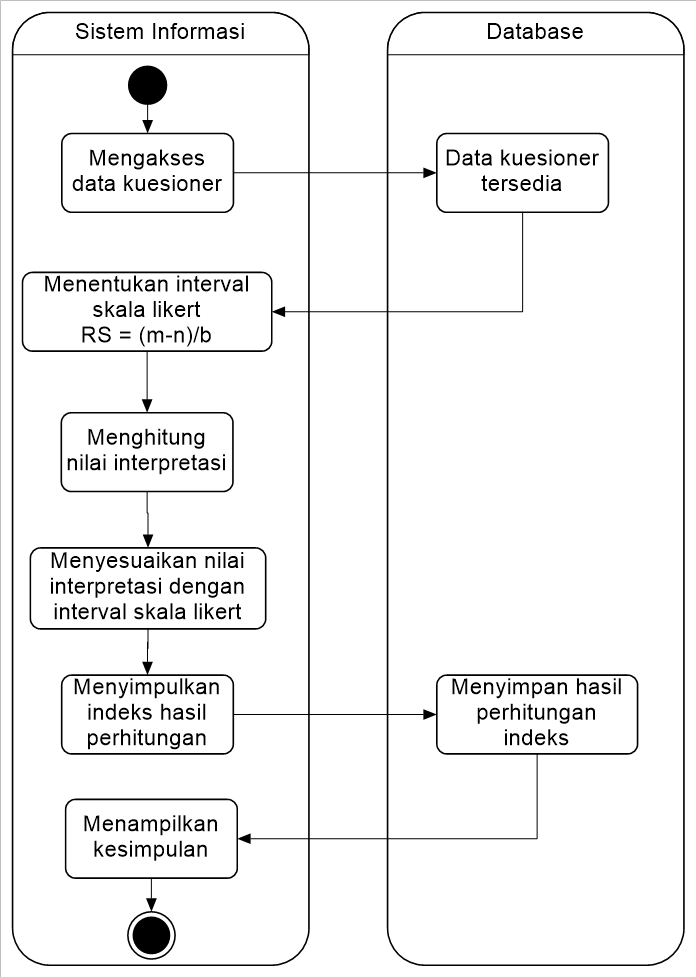
1. Menghapus Kuesioner



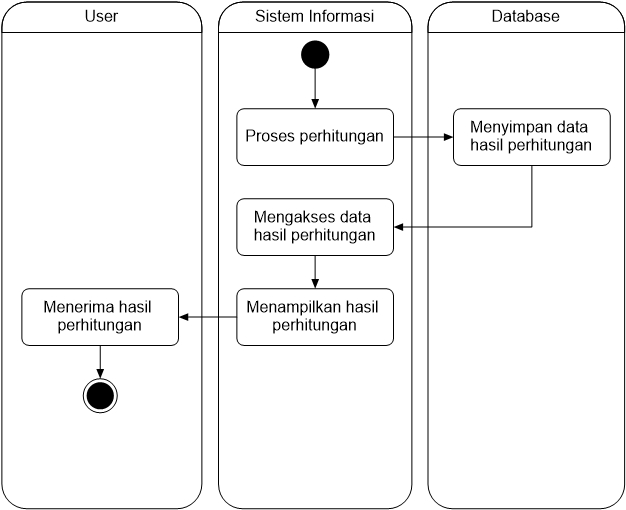
1. Mengisi Kuesioner Valid



1. Perhitungan Skala Likert



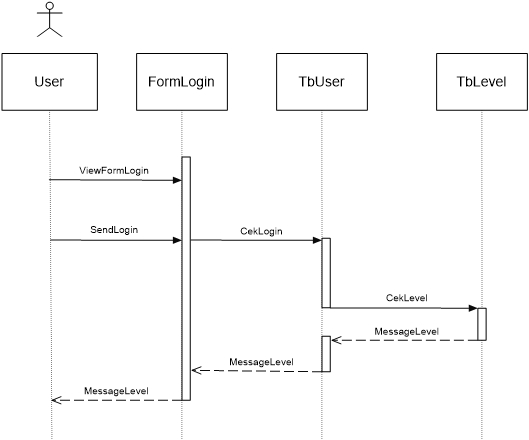
1. Melihat Hasil Perhitungan



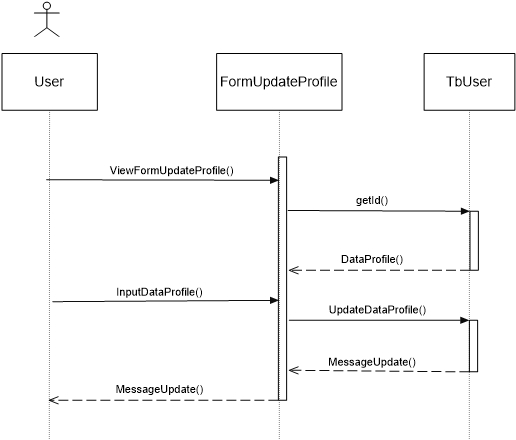
1. Perbarui Data User
2. Menghapus Data User

## Sequence Diagram

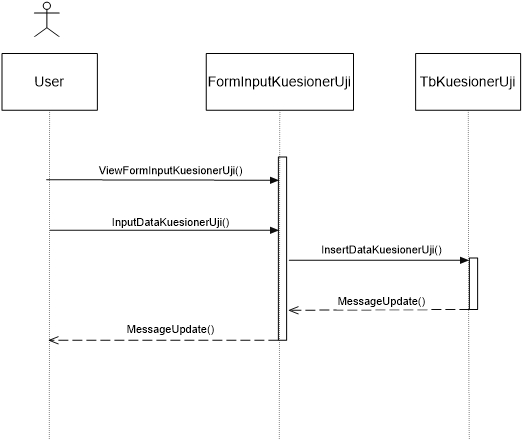
1. Login



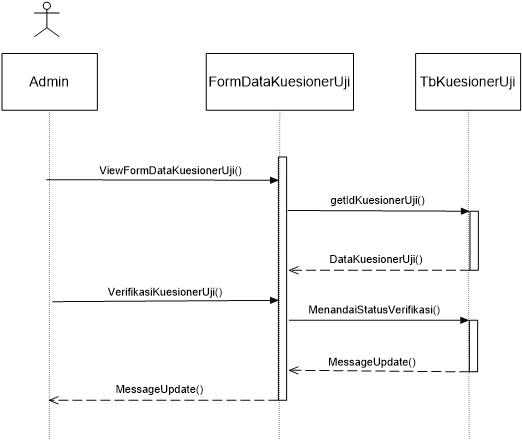
1. Update Profile



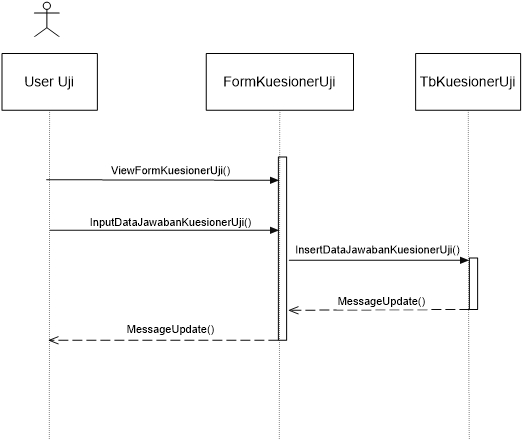
1. Input Kuesioner Uji



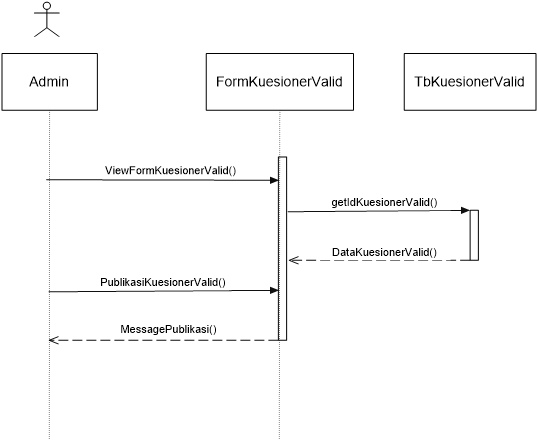
1. Verifikasi Kuesioner



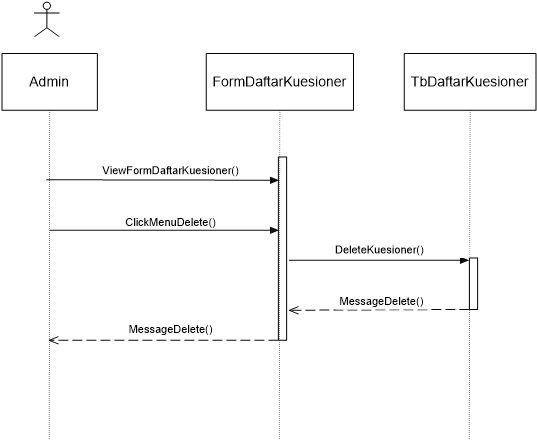
1. Mengisi Kuesioner Uji



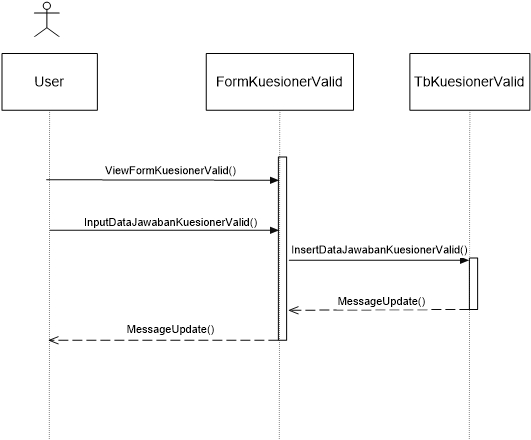
1. Publikasi Kuesioner Valid



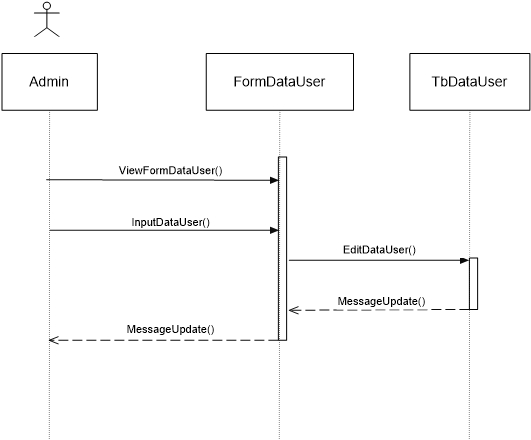
1. Menghapus Kuesioner



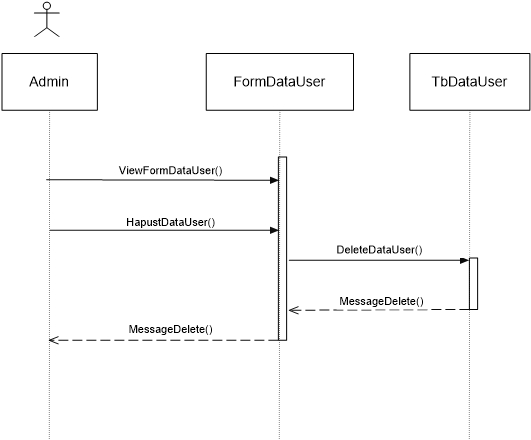
1. Mengisi Kuesioner Valid



1. Perbarui Data User



1. Menghapus Data User



## Class Diagram

## Perancangan Antarmuka

# BAB IV

**HASIL DAN PEMBAHASAN**

## Implementasi Sistem

## Implementasi Database

## Implementasi Antarmuka

### Antarmuka modul…..

## Implementasi Program

## Pengujian Perangkat Lunak

### Pengujian Black Box

### Pengujian White Box

### Stress Testing

# BAB V

**KESIMPULAN DAN SARAN**

## Kesimpulan

## Saran

# DAFTAR PUSTAKA