**IMPLEMENTASI ALGORITMA WINNOWING PADA APLIKASI PENDETEKSI KEMIRIPAN DOKUMEN**

**SKRIPSI**

Program Studi Teknik Informatika

Jurusan Teknik Informatika

Oleh:

**GLEN H.O MANGUNDAP**

NIM D1041141059



FAKULTAS TEKNIK

UNIVERSITAS TANJUNGPURA

PONTIANAK

2020

Halaman Pernyataan

Yang bertanda tangan di bawah ini:

Nama : Glen H.O Mangundap

NIM : D1041141059

menyatakan bahwa dalam skripsi yang berjudul “Implementasi Algoritma Winnowing Pada Aplikasi Pendeteksi Kemiripan Dokumen” tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi manapun. Sepanjang pengetahuan Saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenar-benarnya. Saya sanggup menerima konsekuensi akademis dan hukum di kemudian hari apabila pernyataan yang dibuat ini tidak benar.

Pontianak, 2020

Glen H.O Mangundap

NIM D1041141059

Halaman Pengesahan

IMPLEMENTASI ALGORITMA WINNOWING PADA APLIKASI PENDETEKSI KEMIRPAN DOKUMEN

Program Studi Sarjana Informatika

Jurusan Informatika

Oleh:

Glen H.O Mangundap

NIM D1041141059

Telah dipertahankan di depan Penguji Skripsi pada tanggal 2020  
dan diterima sebagai salah satu persyaratan untuk memperoleh gelar sarjana.

Susunan Penguji Skripsi:

Pembimbing 1, Penguji Utama,

Dr. Herry Sujaini, S.T., M.T. Xxxxxxx

NIP 19680629 199702 1001 NIP 19.....

Pembimbing 2, Penguji Pendamping,

Helen Sasty Pratiwi, S.T., M.Eng. Xxxxxxxx

NIP 19860117 201212 2004 NIP 19....

Pontianak, 2020

Dekan,

Xxxxx

NIP 19…..

Halaman Persembahan

Skripsi ini didedikasikan kepada kedua orang tuaku (Bapak X dan Ibu Y) dan saudara-saudaraku (Alfa, Beta, dan Gamma). Apa yang mereka telah berikan melebihi daripada apa yang pernah aku inginkan.

Kata Pengantar

Skripsi ini dibuat karna pengen lulus

Pontianak, 2020

Penulis,

Glen H.O Mangundap

# ABSTRAK

Entah apa .

Kata kunci: *undershoot*, *linear quadratic regulator*, kendali digital, *piecewise constant hold*

Daftar Isi

[Halaman Pernyataan ii](#_Toc58591292)

[Halaman Pengesahan iii](#_Toc58591293)

[Halaman Persembahan iv](#_Toc58591294)

[Kata Pengantar v](#_Toc58591295)

[ABSTRAK vi](#_Toc58591296)

[Daftar Isi vii](#_Toc58591297)

[Daftar Tabel ix](#_Toc58591298)

[Daftar Gambar x](#_Toc58591299)

[Daftar Lampiran xi](#_Toc58591300)

[Daftar Notasi xii](#_Toc58591301)

[Bab I Pendahuluan 1](#_Toc58591302)

[1.1 Latar Belakang 1](#_Toc58591303)

[1.2 Perumusan Masalah 2](#_Toc58591304)

[1.3 Tujuan Penelitian 2](#_Toc58591305)

[1.4 Pembatasan Masalah 2](#_Toc58591306)

[1.5 Sistematika Penulisan 3](#_Toc58591307)

[Bab II TINJAUAN PUSTAKA 4](#_Toc58591308)

[2.1 Plagiat dan Plagiarisme 4](#_Toc58591309)

[2.1.1 Tipe-Tipe Plagiarisme 4](#_Toc58591310)

[2.1.2 Bentuk Tindakan Plagiarisme 5](#_Toc58591311)

[2.2 Metode Mendeteksi Plagiarisme 6](#_Toc58591312)

[2.3 Information Retrieval 8](#_Toc58591313)

[*2.4* Metode Dokumen *Fingerprinting* 10](#_Toc58591314)

[*2.5* Algoritma *Winnowing* 10](#_Toc58591315)

[2.6 Dice Distance 12](#_Toc58591316)

[2.7 Chebyshev Distance 12](#_Toc58591317)

[2.8 Alat Bantu Perancangan 13](#_Toc58591318)

[2.8.1 Flowchart 13](#_Toc58591319)

[2.8.2 Data Flow Diagram 14](#_Toc58591320)

[2.9 Pengujian Perangkat Lunak 16](#_Toc58591321)

[2.9.1 Pengujian BlackBox 16](#_Toc58591322)

[2.9.2 Pengujian .. 18](#_Toc58591323)

[2.10 Kajian Terkait 18](#_Toc58591324)

[Bab III Metodologi Penelitian 20](#_Toc58591325)

[3.1 Perangkat dan Data Penelitian 20](#_Toc58591326)

[3.1.1 Perangkat Penelitian 20](#_Toc58591327)

[3.1.2 Data Penelitian 20](#_Toc58591328)

[3.2 Metode Penelitian 21](#_Toc58591329)

[3.3 Pengumpulan Data 22](#_Toc58591330)

[3.4 Perancangan Sistem 22](#_Toc58591331)

[3.4.1 Arsitektur Sistem 23](#_Toc58591332)

[3.4.2 Diagram Alir Sistem 23](#_Toc58591333)

[3.4.3 (DFD) Data Flow Diagram 24](#_Toc58591334)

[3.4.4 Spesifikasi Basis Data 29](#_Toc58591335)

[3.4.5 Hubungan Antar Tabel 31](#_Toc58591336)

[3.4.6 Perancangan Antarmuka Sistem 31](#_Toc58591337)

[3.4.7 Perancangan Pengujian Aplikasi 35](#_Toc58591338)

[3.5 Implementasi 36](#_Toc58591339)

[3.6 Pengujian Sistem 36](#_Toc58591340)

[3.7 Analisis Hasil Pengujian Sistem 37](#_Toc58591341)

[3.8 Penarikan Kesimpulan 37](#_Toc58591342)

[Bab IV Hasil dan Analisis 38](#_Toc58591343)

[4.1 Hasil Perancangan 38](#_Toc58591344)

[4.1.1 Antarmuka halaman login 38](#_Toc58591345)

[4.1.2 Antarmuka Dashboard Admin 38](#_Toc58591346)

[4.1.3 Antarmuka Dashboard Mahasiswa 40](#_Toc58591347)

[4.2 Analisis Hasil Pengujian 41](#_Toc58591348)

[Bab V Kesimpulan dan Saran 43](#_Toc58591349)

[5.1 Kesimpulan 43](#_Toc58591350)

[5.2 Saran 43](#_Toc58591351)

[Daftar Pustaka 44](#_Toc58591352)

Daftar Tabel

[**Tabel 2.1** Simbol flowchart 13](#_Toc53492108)

[**Tabel 2. 2** Simbol-simbol DFD 15](#_Toc53492109)

Daftar Gambar

[**Gambar 2.1** Alur Algoritma Winnowing 11](#_Toc53490952)

Table 4. 1

Daftar Lampiran

[Lampiran A Xxxxxxx xxxxxx A-1](#_Toc354695067)

[Lampiran B Xxxxx Xxxxxx B-1](#_Toc354695068)

[Lampiran C Xxxxx Xxxxxx C-1](#_Toc354695069)

[Lampiran D Xxxxx Xxxxx D-1](#_Toc354695070)

[Lampiran E Xxxxx Xxxxxx E-1](#_Toc354695071)

[Lampiran F Xxxxxx Xxxxxx F-1](#_Toc354695072)

[Lampiran G Xxxx Xxxxxx G-1](#_Toc354695073)

[Lampiran H Xxxx Xxxxxx H-1](#_Toc354695074)

[Lampiran I Xxxxx Xxxxxx I-1](#_Toc354695075)

[Lampiran J Xxxxx Xxxxxx J-1](#_Toc354695076)

[Lampiran K Xxxxxx Xxxxxx K-1](#_Toc354695077)

[Lampiran L Xxxxx Xxxxxx L-1](#_Toc354695078)

[Lampiran M Xxxxxx Xxxxxx M-1](#_Toc354695079)

[Lampiran N Xxxxxx Xxxxxx N-1](#_Toc354695080)

[Lampiran O Xxxxxx Xxxxxx O-1](#_Toc354695081)

[Lampiran P Xxxxxx Xxxxxx P-1](#_Toc354695082)

Daftar Notasi

Persamaan 1 12

Persamaan 2 12

# Pendahuluan

## Latar Belakang

Skripsi merupakan suatu bentuk karya ilmiah yang disusun oleh mahasiswa sebagai salah satu syarat dalam menyelesaikan studi program strata satu (S1). Dalam penyusunan skripsi diperlukan referensi sebagai acuan, dengan perkembangan teknologi yang pesat akhir-akhir ini membuat proses pembuatan karya tulis semakin mudah dan cepat. Hal tersebut membawa dampak positif dan negatif, salah satu dampak positifnya ialah pada kemudahan bertukar informasi, dengan kemudahan ini sering disalahgunakan sehingga menjadi dampak negatif, satu diantaranya yaitu plagiarisme. Menurut Soelistyo (2011), plagiarisme merupakan tindakan menjiplak ide, gagasan atau karya orang lain untuk diakui sebagai karya sendiri atau menggunakan karya orang lain tanpa menyebut sembernya sehingga menimbulkan asumsi yang salah atau keliru mengenai asal muasal dari suatu ide, gagasan atau karya.

Beberapa kasus penulisan skripsi mahasiswa memiliki topik judul yang sama, dengan samanya topik judul yang dibuat oleh mahasiswa membuat kemungkinan skripsi yang dibuat mirip. Kemiripan kalimat di dalam dokumen skripsi merupakan suatu hal yang lumrah dalam penulisannya. Kemiripan tersebut tidak bisa dibilang sebagai bentuk plagiarisme. Oleh karena itu dibutuhkan aplikasi yang dapat mendeteksi kemiripan antara dokumen skripsi dengan dokumen skripsi lainya agar dapat mengetahui apakah kalimat yang mirip di dalam dokumen skripsi mengandung plagiarisme atau tidak.

Untuk membuat aplikasi yang dapat mendeteksi kemiripan antar dokumen skripsi mahasiswa dibutuhkan sebuah algoritma agar aplikasi dapat berjalan dengan baik. Berdasarkan penelitian-penelitian sebelumnya ada banyak algoritma yang digunakan terkait dengan deteksi kemiripan dokumen, seperti algoritma Rabin-Karp, K-Nearest Neighbors, dan lain-lain.

Metode untuk mendeteksi plagiarisme memiliki beberapa cara, yaitu perbandingan teks lengkap, dokumen fingerprinting dan kesamaan kata kunci. Dalam penelitian ini menggunakan metode fingerprinting dan algoritma yang digunakan dalam metode ini adalah algoritma Winnowing. Algoritma Winnowing digunakan untuk meningkatkan efisiensi dari proses perbandingan dokumen dan algoritma Winnowing menghasilkan nilai fingerprinting. Nilai fingerprinting yang dibentuk dari algoritma winnowing digunakan untuk mengukur presentase kemiripan teks dengan pengukuran dice similarity distance dan chebyshev distance dimana hasil dari pengukuran distance tersebut digabungkan untuk memperoleh hasil kemiripan dokumen yang memeliki nilai tinggi dari dua pengukuran distance yang telah digabungkan.

Bedasarkan penjelesan diatas, maka akan dibuat suatu aplikasi yang dapat mendeteksi kemiripan dokumen skripsi mahasiswa dengan menggunakan algorima Winnowing. Dengan adanya aplikasi ini diharapkan nanti dapat berguna untuk melihat berapa persentase tingkat kemiripan skripsi antar mahasiswa Jurusan Informatika Universitas Tanjungpura.

## Perumusan Masalah

Mengacu pada latar belakang diatas, dapat disimpulkan permasalahannya yaitu bagaimana membangun aplikasi pendeteksi kemiripan dokumen dengan menggunakan Algoritma *Winnowing*.

## Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membangun sebuah apikasi yang dapat mendeteksi kemiripan dokumen skripsi dengan dokumen skripsi lainnya dengan memberi nilai persentase kemiripan

## Pembatasan Masalah

Pembatasan masalah dari penelitian yang dilakukan adalah:

1. Aplikasi yang dibangun berbasis web.
2. Data yang digunakan adalah dari data skripsi mahasiswa Jurusan Informatika Universitas Tanjungpura.

## Sistematika Penulisan

Sistematika dari penulisan tugas akhir ini disusun dalam 5(lima) bab yang terdiri dari Bab I Pendahuluan, Bab II Tinjauan Pustaka, Bab III Metodologi Penelitian dan Arsitektur Sistem, Bab IV Implementasi dan Hasil Pengujian serta Bab V penutup.

**Bab I Pendahuluan** adalah bab yang berisi latar belakang penelitian, perumusan masalah, tujuan penelitian, pembatasan masalah, dan sistematika penulisan.

**Bab II Tinjauan Pustaka** adalah bab yang berisi uraian sistematis tentang hasil-hasil penelitian yang didapat oleh peneliti terdahulu dan landasan teori yang berhubungan dengan penelitian yang akan dilakukan.

**Bab III Metodologi Penelitian dan Perancangan Sistem** adalah bab yang berisi tentang bahan penelitian, alat yang digunakan, metode penelitian, variable atau data, serta diagram alir penelitian.

**Bab IV Hasil dan Analisis** adalah bab yang berisi data hasil perancangan, pengujian dan sebagainya yang telah dirancang pada Bab III. Setiap hasil yang disajikan akan dilakukan analisis untuk mengarah keapada suatu kesimpulan.

**Bab V Penutup** adalah bab yang berisi kesimpulan dari penelitian yang telah dilakukan dan saran/rekomendasi untuk perbaikan, pengembangan atau kesempurnaan/kelengkapan penelitian yang telah dilakukan.

# TINJAUAN PUSTAKA

## Plagiat dan Plagiarisme

Kata plagiarisme berasal dari bahasa Latin “plagiare” yang berarti mencuri. Menurut Soelistyo (2011), plagiat adalah pengumuman sebuah karya pengetahuan atau seni oleh ilmuwan atau seniman kepada publik atas semua atau sebagian besar karya orang lain tanpa menyebut nama sang pengarang yang diambil karyanya.

Serupa dengan definisi dalam Kamus Besar Bahasa Indonesia disebutkan bahwa “Plagiat” adalah pengambilan karangan orang lain dan menjadikannya seolah-olah karangan sendiri. Sedangkan “Plagiarisme” adalah penjiplakan yang melanggar hak cipta.

Dari definisi diatas dapat disimpulkan plagiatrisme merupakan suatu tindakan penjiplakan karya orang lain dan membuat karya tersebut seolah-olah adalah hasil karya diri sendiri. Tindakan plagiarisme tersebut merupakan suatu bentuk pelanggaran hak cipta sehingga pelaku plagiarisme yang biasa disebut plagiator dapat dijatuhkan hukuman karena tindakannya yang secara tidak lansung mencuri karya orang lain.

### Tipe-Tipe Plagiarisme

Menurut Soelistyo (2011), plagiarisme atau plagiat terdapat beberapa tipe-tipe antara lain ialah :

1. Plagiarisme Ide (*Plagiarism of Ideas*)

Plagiarisme ide sulit dibuktikan karena ide atau gagasan itu bersifat abstrak dan kemungkinan memiliki kesamaan dengan ide orang lain. Oleh karena itu, perlu bahan bukti yang cukup kuat untuk memastikan adanya plagiarisme. Cara paling mudah untuk membutikan adanya plagiarisme adalah dengan mempertanyakan apakah ia mendapakan keuntungan dari pemikiran orang lain.

Plagiarisme atas ide ini sering sekalu terjadi dalam kehidupan berkesenian dan kebudayaan. Di Indonesia, karya film, tari maupun novel lazim diakui sebagia adaptasi, gubahan atau saduran. Semua itu merupakan plagiarisme ide, sejauh apabila tidak dinyatakan sumber yang menjadi rujukannya. Dalam UU Hak Cipta, karya-karya adaptasi, gubahan, dan saduran mendapatkan perlindungan tersendiri, demikian karya tafsir dan terjemahanya.

Plagiarisme seperti ini mendapat dukungan yang lemah dari undang-undang. Sebab secara konseptual UU Hak Cipta hanya komit untuk melindungi ekspresi bukan ide. Jadi, sepanjang masih berupa ide atau gagasan, UU Hak Cipta tidak menjamin perlindungan hukumnya.

1. Plagiarisme Kata Demi Kata (*Word for Word Plagiarism*)

Tipe plagiarisme ini merupakan tindakan mengutip karya orang lain secara kata demi kata tanpa menyebutkan sumbernya. Dianggap plagiarisme karena skala pengutipannya sangat substansial. Sehingga seluruh ide atau gagasan penulisanya benar-benar diambil. Banyak dilakukan pada karya tulis puisi.

1. Plagiarisme Atas Sumber (*Plagiarism of Source*)

Dikatakan plagiarisme karena tidak menyebutkan sumber secara lengkap selengkap-lengkapnya referensi yang dirujuk dalam kutipan.

1. Plagiarisme Kepengarangan (*Plagiarism of Authorship*)

Plagiarisme ini terjadi apabila seseorang mengaku sebagai pengarang dari karya tulis yang disusun oleh orang lain. Tindakan ini dilakukan dengan kesadaran dan motif kesengajaan untuk membohongi publik.

### Bentuk Tindakan Plagiarisme

Menurut Soelistyo (2011), berikut ini beberapa tindakan yang dianggap sebagai tindakan plagiarisme :

1. Mengakui tulisan orang lain sebagai tulisan sendiri;
2. Mengakui gagasan orang lain sebagai pemikiran sendiri;
3. Mengakui temuan orang lain sebagai kepunyaan sendiri;
4. Mengakui karya kelompok sebagai kepunyaan atau hasil sendiri;
5. Menyajikan karya tulisan yang sama dalam kesempatan yang berbeda tanpa menyebutkan asal-usulnya;
6. Meringkas dan memparafrasekan (mengutip tak lansung) tanpa menyebutkan sumbernya;
7. Meringkas dan memparafrasekan dengan menyebut sumbernya, tetapi rangkaian kalimat dan pilihan katanya masih terlalu sama dengan sumbernya.

Sementara itu, tindakan-tindakan yang tidak tergolong plagiarisme antara antara lain :

1. Menggunakan informasi yang berupa fakta umum.
2. Menuliskan kembali (dengan mengubah kalimat atau parafrase) opini orang lain dengan memberikan sumber jelas.
3. Mengutip secukupnya tulisan orang lain dengan memberikan tanda batas jelas begian kutipan dan menuliskan sumbernya.

## Metode Mendeteksi Plagiarisme

Dalam buku Kurniawati dan Simri (2008), untuk mendeteksian penjiplakan terdapat tiga metode yaitu:

1. Perbandingan Teks Lengkap

Metode ini diterapkan dengan membandingkan semua isi dokumen. Namun pendekatan ini membutuhkan waktu yang lama tetapi cukup efektif. Algoritma yang digunakan pada metode ini adalah algoritma *Brute Force*, algoritma *Edit Distance*, algoritma *Boyer Moore*. Pada algoritma *Edit Distance*, untuk mendeteksi plagiat pada dua teks sumber adalah dengan cara memasukkan isi tiap file sumber ke dalam string. Dalam proses memasukkan isi file, teks yang dimasukkan ke dalam string adalah teks program tanpa komentar. Jadi, ketika karakter yang dibaca dari teks sumber adalah penanda komentar maka teks dalam komentar tersebut tidak akan dimasukkan ke dalam string. Untuk mengecek keabsahan suatu teks digunakan persentase kemiripan (P) yang dapat dihitung dengan cara perbandingan hasil keluaran algoritma edit distance (D) dengan jumlah karakter terpanjang antara 2 masukan (T). Dari nilai persentase dapat diketahui besarnya tingkat kemiripan antara kedua teks masukan, apakah masih dalam batas toleransi (80%). Jika ternyata P lebih besar dari 80%, dapat diambil kesimpulan bahwa telah terjadi sebuah praktik plagiat. Namun, hal di atas tidaklah mutlak karena pengguna dapat menentukan sendiri nilai batas toleransi.

1. Dokumen *Fingerprint*

Dokumen *fingerprint* merupakan metode yang digunakan untuk mendeteksi keakuratan kesamaan antar dokumen. Prinsip kerja dari metode dokumen *fingerprint* ini dengan menggunakan teknik *hashing*. Teknik *hashing* adalah sebuah fungsi yang menkonversi setiap string menjadi bilangan. Algoritma yang digunakan pada metode ini seperti algoritma *Winnowing*, algoritma *Manber* dan algoritma *Rabin*-*Karp*.

1. Kesamaan Kata Kunci

Prinsip dari metode kesamaan kata kunci adalah mencari kata kunci dari dokumen dan kemudian dibandingkan dengan kata kunci pada dokumen lain.

Sedangkan menurut Schleimer, Wilkerson, dan Aiken (2003) untuk melakukan pendeteksian plagiarisme dokumen teks sebuah algoritma harus memenuhi salah satu persyaratan berikut ini:

1. *Whitespace* *Insensitivity*, yang berarti dalam melakukan pencocokan terhadap dokumen teks seharusnya tidak terpengaruh oleh spasi, jenis huruf (kapital atau normal), tanda baca, simbol-simbol dan sebagainya.
2. *Noise* *Surpression*, yang berarti menghindari penemuan kecocokan dengan panjang kata yang terlalu kecil atau kurang relevan, misal: ‘the’. Panjang kata yang ditengarai merupakan penjiplakan harus cukup untuk membuktikan bahwa kata-kata tersebut telah dijiplak dan bukan merupakan kata yang umum digunakan.
3. *Position* *Independence*, yang berarti penemuan kecocokan atau kesamaan tidak harus bergantung pada posisi kata-kata. Meskipun berada pada posisi yang tidak sama, kecocokan atau kesamaan harus dapat ditemukan.

## Information Retrieval

Menurut Manning, Raghavan dan Schütze (2009) *Information* *Retrieval* adalah studi tentang sistem pengindeksan, pencarian, dan mengingat data, khususnya teks atau bentuk tidak terstruktur lainnya. *Information* *Retrieval* merupakan bagian dari computer *science* yang berhubungan dengan pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. *Information* *Retrieval* merupakan suatu pencarian informasi (biasanya berupa dokumen) yang didasarkan pada suatu *query* (masukkan *user*) yang diharapkan dapat memenuhi keinginan user dari kumpulan dokumen yang ada. Sedangkan, definisi *query* dalam *Information* *Retrieval* merupakan sebuah formula yang digunakan untuk mencari informasi yang dibutuhkan oleh *user* dan merupakan suatu *keywords* (kata kunci) dalam bentuk yang paling sederhana. Dalam *Information* *Retrieval*, ada beberapa tahapan yang dapat dilakukan dalam pengelolaan dokumen teks, diantaranya:

1. *Preprocessing*

Pada sebuah teks dokumen, informasi yang akan digali berisi informasi-informasi yang strukturnya sembarang. Oleh karena itu, diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhannya yang di kenal dengan *prepocessing*. *Preprocessing* merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan atau pembersihan teks yang dilakukan untuk mengubah data data berkulitas yaitu data yang telah memenuhi persyaratan untuk dieksekusi pada sebuah algoritma. Bentuk pembersihan teks ini seperti menghilangkan spasi, tanda baca, mengubah huruf kapital menjadi huruf kecil dan menghilangkan karakter-karakter yang tidak relevan lainnya.

1. Tokenisasi

Kok dan Salim (2010) menyebutkan proses tokenisasi yaitu proses pemisahan kata dari teks dokumen secara dengan menggunakan karakter spasi sebagai tanda pemisahnya. Dengan adanya pemisahan kata terlebih dahulu, string yang diinputkan akan terlihat lebih ringkas karena ditampilkan dalam bentuk tiap kata sesuai dengan spasi yang memisahkannya. Pemisahan kata atau *fingerprint* pada teks dokumen memiliki beberapa teknik yang sesuai:

1. *Character*-*based*

Teknik ini menggunakan urutan karakter untuk membentuk fingerprint pada semua dokumen. Misalnya, jika kita memiliki dokumen dengan panjang D= 5 yang memiliki satu kata "*touch*", maka kita dapat melihat bahwa "*touc*" dan "*ouch*" adalah semua kemungkinan substring panjang K = 4. Pada dasarnya, membandingkan dua dokumen menggunakan teknik ini adalah menghitung jumlah *substring* di kedua *fingerprint*.

1. *Phrase*-*based*.

Teknik untuk menghasilkan sidik jari menggunakan mekanisme frase untuk mengukur kemiripan antara dua dokumen ini pertama kali diperkenalkan oleh Lyon et al pada tahun 2001. Pada tahap awal, kita harus mengkonversi setiap dokumen untuk satu set *bigram* (dua kata) atau *trigram* (tiga kata). Misalnya kalimat “aplikasi pendeteksi penjiplakan teks dokumen” akan dikonversi ke bentuk *triword*, sehingga menghasilkan “aplikasi pendeteksi penjiplakan, pendeteksi penjiplakan teks, penjiplakan teks dokumen”. Kemudian set dari *triword* untuk setiap dokumen dibandingkan dengan dokumen lain.

1. *Statement*-*based*

Pro dan kontra dari character-based dan *frase*-*based* *fingerprint* telah membuat munculnya *fingerprint* untuk setiap pernyataan. Meskipun setiap nilai K dapat dipertimbangkan. Hal ini karena nilai-nilai yang lebih kecil dari K (yaitu, K = 1, 2, atau 3) tidak memberikan diskriminasi yang baik antara kalimat.

1. Metode *K*-*grams*

Metode *K-grams* merupakan salah satu metode yang terdapat dalam proses tokenisasi. Metode *K*-*grams* ini digunakan untuk membentuk *substring* sepanjang k karakter dari sebuah string. Biasanya yang dijadikan *substring* adalah kata. Semakin kecil nilai karakter K , maka pencarian nilai similaritas semakin efektif.

## Metode Dokumen *Fingerprinting*

Menurut Alamsyah (2017) metode *fingerprinting* adalah metode yang menelusuri karakter satu persatu pada deret karakter. Prinsip kerja dari metode dokumen *fingerprinting* ini adalah dengan menggunakan teknik *hashing*. Teknik *hashing* adalah sebuah fungsi yang mengkonversi setiap string menjadi bilangan. Bilangan-bilangan tersebut menghasilkan nilai-nilai *fingerprints* sebagai acuan dalam perhitungan kesamaan dokumen. Kelebihan dari metode *fingerprinting* adalah waktu proses lebih cepat dibandingkan dari metode perbandingan teks lengkap dan metode kesamaan kata kunci. Pengujian similaritas dokumen *fingerprinting* memiliki tahapan pencarian nilai *hashing* dari setiap kata, pengambilan nilai *fingerprints* dan pembobotan persentasi kemiripan. Beberapa algoritma yang termasuk dalam metode *fingerprinting* yaitu Algoritma *Winnowing* dan Algoritma *Rabin-Karp* .

## Algoritma *Winnowing*

Schleimer, Wilkerson dan Aiken (2003) menyimpulkan bahwa *Winnowing* adalah algoritma yang digunakan untuk melakukan proses *document* *fingerprinting*. *Document* *fingerprinting* merupakan metode yang digunakan untuk mendeteksi keakuratan salinan antar dokumen atau hanya sebagian teks saja. Prinsip kerja dari metode *document* *fingerprinting* ini adalah dengan menggunakan teknik *hashing*. Teknik *hashing* adalah sebuah fungsi yang mengkonversi setiap string menjadi bilangan. Algoritma *winnowing* melakukan penghitungan nilai-nilai *hash* dari setiap *k-gram*, untuk mencari nilai *hash* selanjutnya digunakan fungsi *rolling* *hash*. Kemudian dibentuk *window* dari nilai-nilai *hash* tersebut. Dalam setiap *window* dipilih nilai *hash* minimum. Jika ada lebih dari satu *hash* dengan nilai minimum, dipilih nilai *hash* yang paling kanan. Kemudian semua nilai *hash* terpilih disimpan untuk dijadikan *fingerprint* dari suatu dokumen. Input dari proses *document* *fingerprinting* adalah file teks. Kemudian outputnya akan berupa sekumpulan nilai *hash* yang disebut *fingerprint*. *Fingerprint* ini yang akan dijadikan dasar pembanding kesamaan antara teks yang telah dimasukkan. Syarat dari algoritma deteksi *plagiarism* yaitu;

1. *whitespace* *insensitivity*, yaitu pencocokan teks *file* seharusnya tidak terpengaruh oleh spasi, jenis huruf kapital, tanda baca dan sebagainya.
2. *noise* *surpression*, yaitu menghindari pencocokan teks *file* dengan panjang kata yang terlalu kecil atau kurang relevan dan bukan merupakan kata yang umum digunakan.
3. *position* *independence*, yaitu pencocokan teks *file* seharusnya tidak bergantung pada posisi kata-kata sehingga kata dengan urutan posisi berbeda masih dapat dikenali jika terjadi kesamaan.

*Winnowing* telah memenuhi syarat-syarat tersebut dengan cara membuang seluruh karakter-karakter yang tidak relevan misal: tanda baca, spasi dan juga karakter lain, sehingga nantinya hanya karakter-karakter yang berupa huruf atau angka yang akan diproses lebih lanjut. Menurut jurnal yang ditulis Jody (2015), alur algoritma *Winnowing* dapat dilihat pada gambar 2.1 dibawah.



**Gambar 2.1** Alur Algoritma Winnowing

## Dice Distance

Koefisien *Dice* yang dikenal juga dengan sebutan Sørensen–Dice index merupakan metode yang digunakan untuk membandingkan tingkat similaritas dari dua objek. Metode ini dipublikasikan oleh Sørensen dan Lee Raymond Dice pada 1948 dan 1945 secara berturut-turut. Rumus ini memiliki kesamaan dengan rumus Jaccard. Namun, perbedaannya terletak pada adanya pencocokan dua kali pada rumus *Dice coefficient*. Berikut adalah rumus *Dice coefficient*.

(2. 1 )

Persamaan 2.1 merupakan rumus *Dice distance.* Sebagai contoh apabila D dan Q adalah himpunan set string (bukan vektor), jumlah *unigram* dari “*Photography*” and “*Photographic*” masing-masing adalah |D| = 9 dan |Q| = 10 dimana |D n Q| = 8, sehingga dapat diperoleh *dice* (D,Q) = 0.842 atau persentase *similarity* dari kedua kata tersebut adalah 84.2%

## Chebyshev Distance

*Chebyshev* *Distance* merupakan suatu perhitungan jarak dari dua buah titik dalam sebuah *vector* *space*. *Chebyshev* *Distance* sering juga disebut *Chessboard* *Distance* (Jarak Papan Catur) karena dalam permainan catur dibutuhkan jumlah minimum untuk raja bergerak. Menurut Kardi (2015) *Chebyshev* *Distance* adalah mengukur besaran nilai mutlak dari koordinat pada sepasang objek.

𝑑𝑥𝑦 = 𝑚𝑎𝑥𝑘|𝑥𝑖𝑘 − 𝑦𝑗𝑘 | (2. 2)

Persamaan 2.2 merupakan rumus dari *Chebyshev distance,* dimana 𝑑𝑥𝑦 adalah jarak *Chebyshev* dari titik x ke titik y. Nilai 𝑚𝑎𝑥𝑘 adalah selisih maksimum dari objek 𝑥𝑖𝑘 dan 𝑦𝑗𝑘 . Semakin besar nilai 𝑑𝑥𝑦, maka semakin besar kesamaan antara kedua objek atau kasus tersebut.

## Alat Bantu Perancangan

Dalam melakukan perancangan sistem dibutuhkan alat bantu yang digunakan untuk membantu memodelkan sistem aplikasi yang akan dibuat. Ada beberapa alat yang dapat digunakan untuk membantu dalam perancangan sistem yaitu sebagai berikut:

### Flowchart

Suyanto (2004) menyimpulkan bahwa *flowchart* adalah gambaran dari tahapan proses suatu sistem, program *flowchart* meggambarkan urutan-urutan instruksi dari suatu program komputer, oleh karena itu *flowchart* yang dihasilkan dapat bervariasi antara suatu program dengan yang lainnya.

Pengertian *Flowchart* adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. Berikut adalah beberapa simbol yang digunakan dalam menggambar suatu *flowchart* dijelaskan pada tabel Pahlevy (2010). Menurut Jogiyanto (2005) “Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika”. Menurut Jogiyanto (2005) “Bagan alir program (program *flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem”.

Berdasarkan jurnal yang ditulis Pahlevy (2010), bagian-bagian dari *flowchart* dapat dilihat pada tabel 5.1 dibawah ini.

**Tabel 2.1** Simbol flowchart

| **Simbol** | **Nama** | **Fungsi** |
| --- | --- | --- |
|  | Terminator | Permulaan/ akhir program |
|  | Garis Alir (*Flow* *Line*) | Arah aliran program |
|  | Preparation | Proses inisialisasi/pemberian harwal awal |
|  | Proses | Proses perhitungan/proses pengolahan data |
|  | *Decision* | Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya |
|  | *On* *Page* *Connector* | Penghubung bagian-bagian *flowchart* yang berada pada satu halaman |
|  | *Off* *Page* *Connector* | Penghubung bagian-bagian *flowchart* yang berada pada halaman berbeda |
|  | Dokumen | Menunjukan dokumen input dan output baik untuk proses manual |
|  | Proses Manual | Melakukan proses manual dalam *flowchart* |

### Data Flow Diagram

Buku yang ditulis oleh Suyanto (2004) menyebutkan bahwa, d*ata* *flow* *diagram* (DFD) merupakan alat yang menggambarkan aliran data melalui sistem dan kerja atau pengolahan yang dilakukan oleh sistem tersebut.

Menurut Kristanto (2004), bahwa DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

Menurut Jogiyanto (2005), data flow diagram adalah diagram yang menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan.

Ada beberapa simbol digunakan pada DFD , yaitu :

1. Kesatuan Luar (*External* *Entity*)

Kesatuan luar (*external* *entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lain yang berada pada lingkungan luarnya yang memberikan input atau menerima output dari sistem.

1. Arus Data (*Data* *Flow*)

Arus Data (*data* *flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir di antara proses, simpan data dan kesatuan luar. Arus data ini menunjukan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

1. Proses (*Process*)

Proses (*process*) menunjukan pada bagian yang mengubah *input* menjadi *output*, yaitu menunjukan bagaimana satu atau lebih *input* diubah menjadi beberapa *output*.

1. Simpanan Data (*Data* *Store*)

*Data* *Store* merupakan simpanan dari data yang dapat berupa suatu *file* atau database pada sistem komputer.

Berdasarkan buku yang ditulis Jogiyanto (2005), simbol-simbol dari *DFD* dapat dilihat pada tabel 5.2 dibawah ini.

**Tabel 2.2** Simbol-simbol DFD

|  |  |
| --- | --- |
| **Simbol *DFD*** | **Nama** |
|  | Entitas / Kesatuanluar |
|  | Proses |
|  | Arus data / Aliran data |
|  | Data store / Simpanan data |

Buku yang ditulis oleh Ladjamudin (2005) menyebutkan ada beberapa tingkatan DFD, yaitu sebagai berikut:

1. Diagram Konteks (*Context* *Diagram*), adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan input ke sistem. Diagram ini hanya memiliki satu proses dan tidak ada *data* *store*.
2. Diagram Nol/*Zero* (*Overview* *Diagram*), menggambarkan proses dari DFD. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, dan *eksternal* *entity*.
3. Diagram Rinci (*Level* *Diagram*), adalah diagram yang menguraikan proses apa yang ada didalam diagram nol atau diagram level di atasnya.

## Pengujian Perangkat Lunak

Pengujian yang dilakukan dalam penelitian ini sebagai tahap implementasi untuk menguji tingkat minimal kesalahan dan keakuratan perangkat lunak yang dirancang. Adapun pengujian yang akan digunakan dalam penelitian yaitu pengujian sistem menggunakan pengujian *Black Box,* pengujian...

### Pengujian BlackBox

Pada metode *Black Box Testing* akan dilakukan pengujian yang berfokuskan kepada persyaratan fungsional sistem yang telah dibuat. Pengujian yang dilakukan dengan melakukan *input* data menggunakan data yang valid dan tidak valid. Selain itu juga dilakukan simulasi mengakses web secara bersamaan waktu untuk mengetahui kemampuan sistem apakah tetap dapat diakses dengan baik atau tidak.

Pada pengujian ini didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan proses yang diinginkan oleh pengguna. Pengujian ini tidak melihat dan tidak menguji *souce code* program.

Pengujian *Black Box* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Uji coba *Black Box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang.

2. Kesalahan antarmuka *interface.*

3. Kesalahan dalam struktur data atau akses database eksternal.

4. Kesalahan performa.

5. Kesalahan inisialisasi dan terminasi.

Uji coba *Black Box* diaplikasikan di beberapa tahapan berikutnya, karena ujicoba ini mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Uji coba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana validitas fungsionalnya diuji ?

2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik ?

3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?

4. Bagaimana batasan-batasan kelas data diisolasi?

5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?

6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Dengan mengaplikasikan ujicoba *Black Box*, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

1. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari ujikasus tambahan harus didesain untuk mencapai ujicoba yang cukup beralasan.

2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, daripada kesalahan yang terhubung hanya dengan suatu ujicoba yang spesifik.

### Pengujian ..

## Kajian Terkait

| No | Penulis | Judul | Keterangan |
| --- | --- | --- | --- |
| 1 | Nur Alamsyah | Perbandingan Algoritma *Winnowing* Dengan  Algoritma *Rabin* *Karp* Untuk Mendeteksi  Plagiarisme Pada Kemiripan Teks Judul Skripsi | * Penelitian ini dilakukan untuk membandingkan dua algoritma pendeteksi kemiripan dokumen yaitu; algoritma *winnowing* dan algoritma *rabin* *karp* * Dari hasil perbandingan yang telah dilakukan algoritma *winnowing* lebih unggul dari algoritma *rabin* *karp* untuk mendeteksi kemiripan dokumen. |
| 2 | Arif Nugroho Putro | Hubungan Fungsi Similaritas Dan Persepsi Penutur Dalam Menentukan Skor Similaritas Teks Bahasa Indonesia | * Pada penelitian yang berjudul Hubungan Fungsi Similaritas Dan Persepsi Penutur Dalam Menentukan Skor Similaritas Teks Bahasa Indonesia, dilakukan perbandingan fungsi similaritas antara *cosine similarity distance, dice similarity distance* dan *euclidean distance similarity.* * Bedasarakan perbandingan dari tiga fungsi similaritas, *dice distance similarity* yang paling unggul diantara dua lainnya. |
| 3 | Mario Anggara | Pemilihan *Distance* *Measure* Pada *K*-*Means* *Clustering* Untuk Pengelompokkan Member Di Alvaro Fitness | * Penelitan ini dilakukan untuk memilih *distance measure* pada *K-Means Clustering* untuk pengelompokan *member* di Alvaro *Fitness.* * Hasil dari pengujian *Silhouette Coefficient Clustering,* *distance measure* yang paling optimal untuk digunakan dalam kasus pengelompokkan *member* di Alvaro *fitness* adalah *Chebyshev Distance.* |

|  |  |  |  |
| --- | --- | --- | --- |
| No | Penulis | Judul | Keterangan |
| 1 | Glen Hizkia Oge Mangundap (2014), Universitas Tanjungpura Pontianak | Implementasi Algoritma *Winnowing* Pada Aplikasi Pendeteksi Kemiripan Dokumen. | * Penelitian ini membangun sebuah aplikasi yang dapat mendeteksi dokumen dengan menggunakan algoritma *Winnowing* dan menggunakan dua pengukuran *distance* yaitu *Dice Distance* dan *Chebyshev Distance.* |

# Metodologi Penelitian

## Perangkat dan Data Penelitian

### Perangkat Penelitian

Penelitian yang dilakukan menggunakan alat untuk menunjang penelitian terdiri dari perangkat keras dan perangkat lunak. Berikut beberapa perangakat keras dan perangkat lunak yang digunakan.

1. Perangkat Keras

Laptop Acer Swift 3 *Acer Day Edition* dengan spesifikasi sebagai berikut :

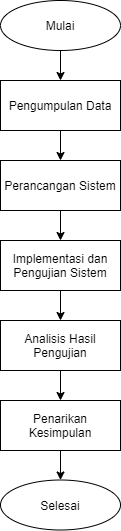
1. Processor inter core i5-8250U CPU @1.60Ghz
2. Ram 8 Gb
3. Storage HDD 1TB, SSD 250GB
4. Grafis Nvidia MX 150 2 GB
5. Perangkat Lunak
   1. OS Windows 10 64-bit
   2. Visual Studio Code untuk teks editor
   3. Xampp Data untuk data base

### Data Penelitian

Data penelitian yang digunakan berupa kumpulan dari seluruh skripsi mahasiswa-mahasiswi Teknik Informatika Universitas Tanjungpura.

## Metode Penelitian

Metode penelitian dilakukan secara bertahap mulai dari pengumpulan data sampai penarikan kesimpulan seperti gambar 3.1

****

**Gambar 3.1** Diagram Alir Penelitian

1. Pengumpulan Data

Pengumpulan data merupakan proses mengumpulkan data-data yang diperlukan untuk menjadi data base aplikasi yang akan dibangun.

1. Perancangan Sistem

Merancang aplikasi dari hasil analisis kebutuhan yang telah dilakukan

1. Implementasi dan Pengujian Sistem

Mengimplementasi setiap rancangan aplikasi dan membangun aplikasi berbasis web. Setelah proses implementasi selesai dilanjutkan dengan pengujian sistem dilakukan untuk melihat apakah aplikasi sudah berjalan dengan baik

1. Analisis Hasil Pengujian Sistem

Menganalisis hasil pengujian sistem apakah sudah sesuai dengan yang diinginkan

1. Penarikan Kesimpulan

Penarikan kesimpulan dilakukan dari hasil analisis pengujian

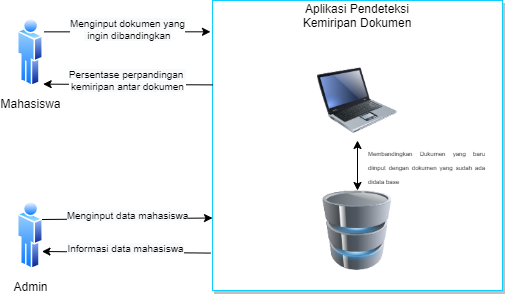
## Pengumpulan Data

Pengumpulan data dilakukan dengan cara mengumpulkan dokumen digital dari skripsi-skripsi mahasiswa Teknik Informatika Universitas Tanjungpura. Kemudian data-data tersebut diubah menjadi data base aplikasi. Kumpulan skripsi yang telah menjadi data base digunakan untuk menjadi perbandingan terhadap skripsi yang baru.

## Perancangan Sistem

Tahap ini merupakan tahap setelah pengumpulan data dilakukan, perancangan sistem ini dilakukan untuk memberi gambaran tentang sistem yang akan dibangun. Pada tahap perancangan sistem ini terdiri dari beberapa bagian sebagai berikut:

### Arsitektur Sistem

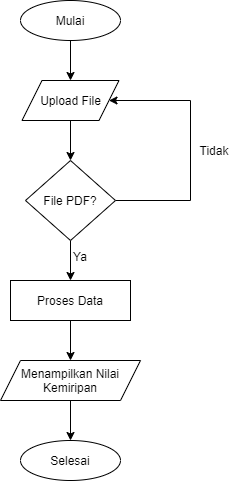
****

**Gambar 3.2** Arsitektur Sistem

Gambar 3.2 merupakan gambaran dari arsitektur sistem, dimana terdapat 2 user yaitu mahasiswa dan admin, admin menginput data mahasiswa agar mahasiswa dapat login dan mengguankan aplikasi. Sedangakan mahasiswa memasukan skripsi yang hendak dibandingkan dengan kumpulan skripsi yang telah diinputkan terlebih dahulu sebagai data base. Setelah dokumen skripsi diinput, dokumen kemudian diproses hingga muncul hasil berupa persentase kemiripan antara dokumen skripsi yang diinputkan dengan beberapa dokumen skripsi yang sudah ada di *database*, sehingga user dapat melihat kalimat mana saja yang mirip dengan dokumen skripsi lainnya agar user dapat merubah kalimat yang menggandung plagiarisme.

### Diagram Alir Sistem

Diagram alir sistem adalah diagram yang menggambarkan sistem yang akan dibangun. Adapun diagram alir sistem yang akan dibangun dapat dilihat pada gambar berikut

****

Gambar 3.3 Diagram Alir Sistem

Pada gambar 3.3 user melakukan upload file berupa dokumen skripsi berformat pdf, setelah file di upload file di proses kemudian nilai kemiripan dokumen yang di upload terhadap dokumen yang ada di database akan ditampilkan.

### (DFD) Data Flow Diagram

*Data Flow Diagram* adalah alat pembuatan *model* yang digunakan untuk menggambarkan aliran data yang ada pada sebuah sistem. Perancangan Data Flow Diagram dapat digambarkan sebagai berikut.

#### Diagram Konteks Sistem

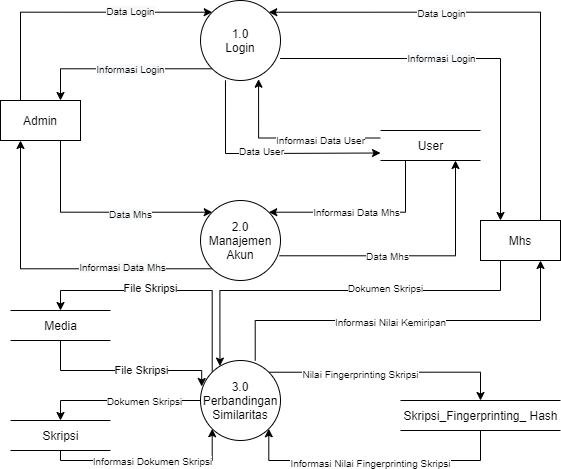
Diagram konteks adalah diagram yang memberikan gambaran umum terhadap kegiatan yang berlangsung dalam sistem. Diagram konteks dapat dilihat pada gambar berikut.



**Gambar 3.4** Diagram Konteks Sistem

#### Diagram Overview Sistem

Diagram *overview* sistem adalah diagram yang menjelaskan urutan-urutan proses dari diagram konteks. Perancangan diagram overview sistem dapat dilihat pada gambar berikut



**Gambar 3.5** Diagram *Overview* Sistem

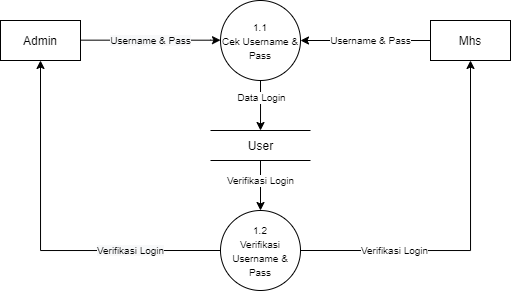
Berdasarkan diagram overview sistem diatas, terdapat 3 proses yang dapat dijelaskan sebagai berikut:

1. Proses 1.0 Login, pada proses ini user melakukan login pada aplikasi, login sebagai admin atau login sebagai mahasiswa.
2. Proses 2.0 Manajemen Akun, pada proses ini admin dapat melakukan CRUD akun mahasiswa untuk dapat mengakses aplikasi.
3. Proses 3.0 Perbandingan Similaritas pada proses ini mahasiswa yang telah melakukan login dapat mengupload file skripsi berformat pdf agar dapat dibandingkan tingkat kemiripannya dengan dokumen yang telah ada di database aplikasi.

#### Diagram Rinci Sistem

Diagram rinci digunakan untuk menguraikan lebih lanjut mengenai setiap proses yang terdapat pada diagram *overview*, yang memperlihatkan aliran data masuk dan data keluar. Berdasarkan diagram *overview* diatas dapat di gambarkan diagram rinci sebagai berikut:

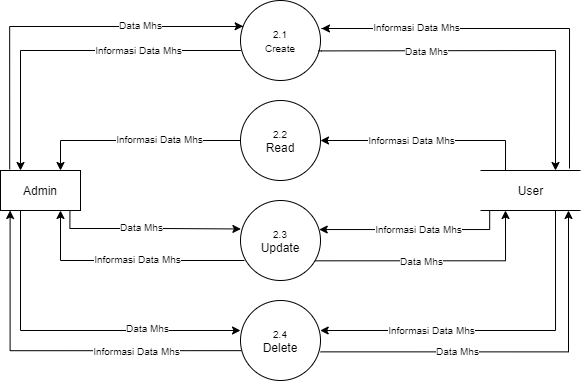
1. Proses 1.0 terdiri dari beberapa proses yaitu :



**Gambar 3.6** Diagram

* 1. Proses 1.1 Cek username & password, pada proses ini user melakukan login dengan menginputkan username dan password, sistem mengecek apakah user yang melakukan login sebagai admin atau mahasiswa dengan mengecek database user.
  2. Proses 1.2 Verifikasi username & password, pada proses ini setelah melakukan pengecekan pada database, jika username dan password terdaftar maka user dapan login, jika tidak terdaftar maka tidak dapat login dan ada pesan akun tidak terdaftar.

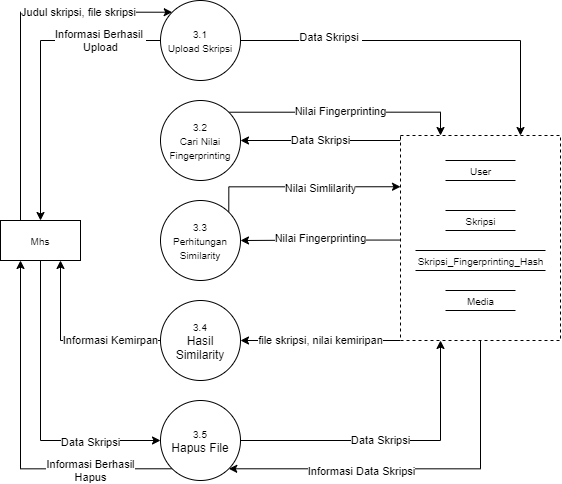
1. Proses 2.0



**Gambar 3.7** Diagram

* 1. Proses 2.1 Create, proses ini admin melakukan pembuatan akun mahasiswa pertama kali agar mahasiswa dapat mengakses aplikasi.
  2. Proses 2.2 Read, proses ini menampilkan daftar akun-akun mahasiswa yang telah dibuat sebelumnya.
  3. Proses 2.3 Update, proses ini admin melakukan update username dan password terhadap akun mahasiswa.
  4. Proses 2.4 Delete, proses ini akun mahasiswa dapat dihapus.

1. Proses 3.0



**Gambar 3.8** Diagram

* 1. Proses 3.1 Upload skripsi, proses ini mahasiswa mengisi form judul skripsi dan mengupload file skripsi yang hendak dibandingkan.
  2. Proses 3.2 Cari nilai fingerprinting, proses ini sistem mencari nilai fingerprinting dari file skripsi yang sudah diupload.
  3. Proses 3.3 Perhitungan similarity, proses ini sistem menghitung nilai fingerprinting yang telah dihitung pada proses sebelumnya dan nilai tersebut dibandingkan terhadap nilai2 fingerprinting yang sudah ada di database.
  4. Proses 3.4 Hasil similarity, proses ini mahasiswa dapat melihat nilai kemiripan file skripsi yang telah di upload dengan skripsi-skripsi yang sudah ada di database.
  5. Proses 3.5 Hapus file, proses ini mahasiswa dapat menghapus file skripsi yang telah di upload.

### Spesifikasi Basis Data

Aplikasi yang di bangun dengan menggunakan beberapa tabel *database* seperti *user,* skripsi, *skripsi\_fingerprinting\_hash.*

#### Tabel user

Nama *database* : Similaritas

Nama tabel : User

Keterangan : Tabel yang menjadi basis data admin dan user

**Tabel 3.1** Spesifikasi Tabel User

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Id | Int (10) | no | Prymary key |
| name | Varchar (255) | no |  |
| email | Varchar (255) | no |  |
| username | Varchar (255) |  |  |
| Email\_verifed\_at | timestamp |  |  |
| password | Varchar (255) |  |  |
| level | Varchar (255) |  |  |
| Remember\_token | Varchar (100) |  |  |
| Created\_at | Timestamp |  |  |
| Updated\_at | Timestamp |  |  |

#### Tabel Skripsi

Nama *database* : Similaritas

Name tabel : Skripsi

Keterangan : tabel skripsi yang diunggah mahasiswa

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Id | Int (10) | no | Prymary key |
| User\_id | Int (10) | no |  |
| judul | Varchar (255) | no |  |
| terverifikasi | Tinyint(1) |  |  |
| Created\_at | Timestamp |  |  |
| Updated\_at | Timestamp |  |  |

**Tabel 3.2** Spesifikasi tabel skripsi

#### Tabel Skripsi\_Fingerprinting\_hash

Nama database : Similaritas

Nama tabel : skripsi\_fingerprinting\_hash

Keterangan : Tabel nilai fingerprinting dari tiap skripsi

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Id | Int (10) | no | Prymary key |
| Skripsi\_id | Int (10) | no |  |
| position | Int (10) | no |  |
| hash | Varchar (255) |  |  |
| Created\_at | Timestamp |  |  |
| Updated\_at | Timestamp |  |  |

**Tabel 3.3** Spesifikasi tabel skripsi\_fingerprinting\_hash

#### Media

Nama database : Similaritas

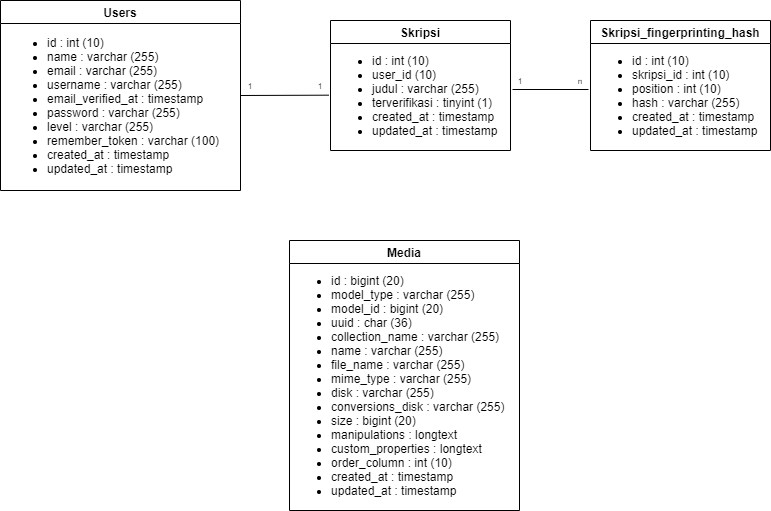
Nama tabel : media

Keterangan : Tabel file skripsi yang diupload

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Field | Tipe Data | Null | Key |
| Id | Bigint (20) | no | Prymary key |
| Model\_type | Varchar (255) | no |  |
| Model\_id | bigint (20) | no |  |
| Uuid | char (36) |  |  |
| Collection\_name | Varchar (255) |  |  |
| Name | Varchar (255) |  |  |
| File\_name | Varchar (255) |  |  |
| Mime\_type | Varchar (255) |  |  |
| Disk | Varchar (255) |  |  |
| Conversions\_disk | Varchar (255) |  |  |
| Size | Bigint (20) |  |  |
| Manipulations | Longtext |  |  |
| Custom\_properties | Longtext |  |  |
| Responsive\_images | Longtext |  |  |
| Order\_column | Int (10) |  |  |
| Created\_at | Timestamp |  |  |
| Update\_at | Timestamp |  |  |

**Tabel 3.4** Spesifikasi tabel media

### Hubungan Antar Tabel



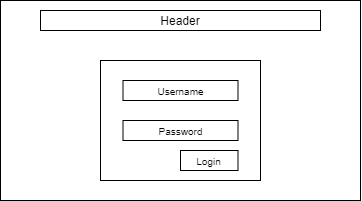
**Gambar 3.9** Hubungan antar tabel basis data

### Perancangan Antarmuka Sistem

Antarmuka sistem merupakan tampilan sistem yang akan dibuat, antarmuka sistem berfungsi sebagai media interaksi user terhadap sistem. Terdapat beberapa perancangan antarmuka sistem yang akan dibangun, sebagai berikut :

#### Halaman Login

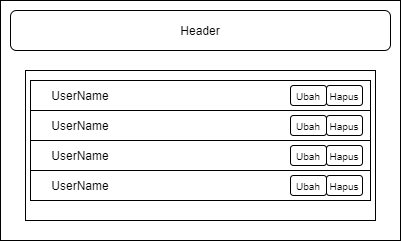
*Layout* halaman login merupakan halaman web yang pertama kali muncul saat aplikasi diakses. Pada halaman login ini user mengisi form username dan password untuk login kedalam aplikasi yang akan dibangun sebagai admin atau mahasiswa.

****

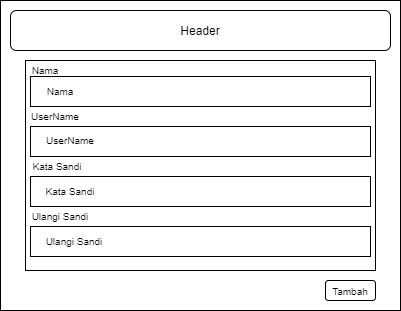
**Gambar 3.10** *layout* halaman login

#### Halaman Admin

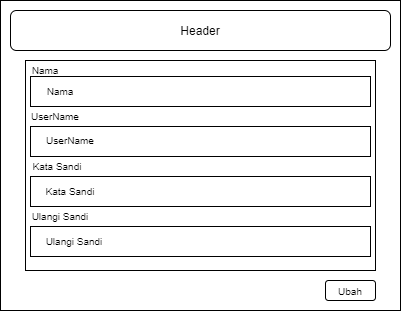
Halamanadmin merupakan halaman awal untuk admin yang tampil saat login sebagai admin. Pada halaman ini terdapat awal terdapat tabel akun mahasiswa yang telah dibuat. *Layout* *dashboard* admin seperti gambar 3.11. Admin dapat menambakah akun baru, *layout* halaman tambah akun seperti gambar 3.12. Admin juga dapat mengubah data akun mahasiswa, *layout* halaman ubah akun seperti gambar 3.13. Rancangan halaman awal dapat dilihat pada gambar 3.3 sebagai berikut:

****

**Gambar 3.11** *Layout* halaman admin.



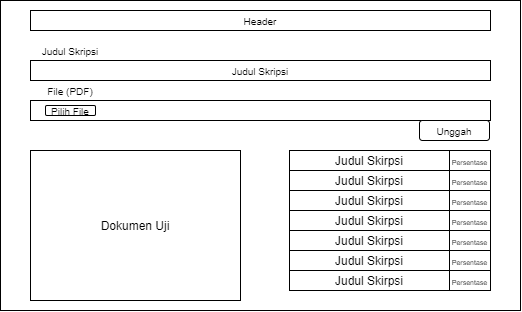
**Gambar 3.12** *layout*  halaman tambah akun

****

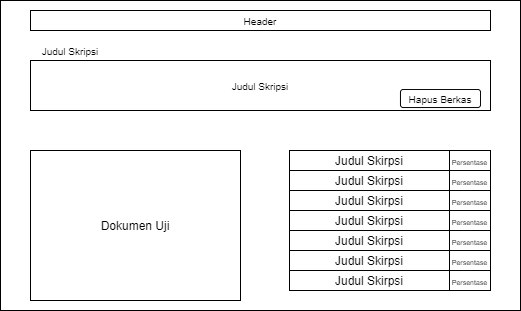
**Gambar 3.13** *Layout*  halaman edit akun.

#### Halaman Mahasiswa

Halaman mahasiswa merupakan halaman utama yang tampil jika login sebagai mahasiswa. *Layout dasboard* mahasiswa seperti gambar 3.14, mahasiswa mengisi form judul dan memilih file skripsi untuk diunggah. Sedangkan pada gambar 3.15 merupakan *layout dashboard* mahasiswa setelah mengupload file dan terdapat juga button hapus file yang telah diupload.

****

**Gambar 3.14** *Layout dashboard* mahasiswa sebelum unggah

****

**Gambar 3.15** *Layout dasbord* mahasiswa setelah unggah

### Perancangan Pengujian Aplikasi

Pada tahap ini akan dilakukan pengujian terhadap sistem yang dibangun. Pengujian yang dilakukan dalam penelitian ini untuk mengujian tingkat minimal kesalahan pada aplikasi yang di bangun. Metode pengujian yang dilakukan pada penelitian ini adalah pengujian Blackbox dan pengujian ....

#### Pengujian Black Box

Pengujian dengan menggunakan metode *Black Box* akan memerikasa apakah sistem dapat berjalan dengan benar sesuai dengan yang diharapkan. Data pengujian dipilih berdasarkan spesifikasi masalah tanpa memperhatikan detil internal dari sistem. Pemilihan cara pengujian dilakukan dengan menggunakan data yang kosong (*null*) dan data yang benar. Kemudian pengujian dilakukan juga pada hasil analisis yang dilakukan oleh sistem terhadap data yang telah dimasukkan oleh *user*. Berikut ini adalah rencana dan indikasi hasil pengujian sistem ini:

1. Saat Saat *user* melakukan input data, ada isian yang harus diisi dan ada isian yang boleh kosong (*null*). Pengujian akan menunjukkan bahwa jika *user* melakukan *input* data dengan keseluruhan data kosong atau terdapat data kosong pada isian yang harus diisi, sistem akan menampilkan pesan kesalahan. Pesan kesalahan berisi informasi data yang wajib diisi.
2. Saat *user* akan melakukan *input* data dengan data yang benar, sistem akan mengeksekusi *input* data tersebut. Sistem langsung menyimpan dalam basis data dan menampilkan pesan informasi bahwa data telah berhasil disimpan.
3. Pencarian data dengan kata kunci yang kosong tidak diproses oleh sistem. Sedangkan pencarian data dengan kata kunci yang benar akan menampilkan data yang akan mengandung kata kunci tersebut. Apabila tidak ada data yang memenuhi kriteria pencarian maka tidak ada data yang muncul pada hasil pencarian. Sistem tidak akan menampilkan pesan apapun kepada admin pada proses pencarian.

Indikasi hasil pengujian akan dirangkum dalam sebuah tabel seperti pada Tabel 3.5

**Tabel 3.5** Tabel pengujian blackbox

|  |  |  |  |
| --- | --- | --- | --- |
| **Fungsi** | **Hasil Yang Diharapkan** | **Hasil Eksekusi** | **Keterangan** |
|  |  |  |  |
|  |  |  |  |

Tabel pengujian terdiri dari 4 kolom, yaitu:

1. Fungsi : Jenis masukan yang digunakan sebagai bahan pengujian seperti *input* data kosong, *input* data salah, dan *input* data benar.
2. Hasil yang diharapakan : merupakan hasil yang diinginkan sesuai kebutuhan.
3. Hasil eksekusi: merupakan hasil dari pengujian yang telah dilakukan dengan masukan data pada kolom contoh data. Berupa status berhasil atau tidaknya proses masukan data tersebut.
4. Keterangan: merupakan penjelasan dari hasil eksekusi pengujian, berupa pesan kesalahan atau respon balik dari aplikasi.

#### Pengujian ..

## Implementasi

Implementasi sistem dilakukan berdasarkan dari hasil perancangan sistem yang telah dilakukan pada tahap perancangan sistem. Pada tahap implementasi ini sistem akan dibangun berbasis web dengan menggunakan PHP.

## Pengujian Sistem

Pengujian sistem merupakan tahap yang digunakan untuk mengevaluasi atau memberi penilaian terhadap sistem yang di bangun. Terdapat beberapa pengujian yang dilakukan terhadap sistem yang dibangun ini, pengujian yang dilakukan adalah pengujian blackbox dan ....

## Analisis Hasil Pengujian Sistem

Pada tahap ini, hasil dari pengujian sistem yang telah dilakukan dianalisis semuanya, hal ini dilakukan agar pada tahap penarikan kesimpulan dapat dilakukan dengan mudah.

## Penarikan Kesimpulan

Pada tahap panarikan kesimpulan dilakukan berdasarkan tahapan-tahapan yang telah dilakukan sebelumnya, apakah sistem yang dibangun dapat berjalan dengan baik sesuai dengan yang diinginkan.

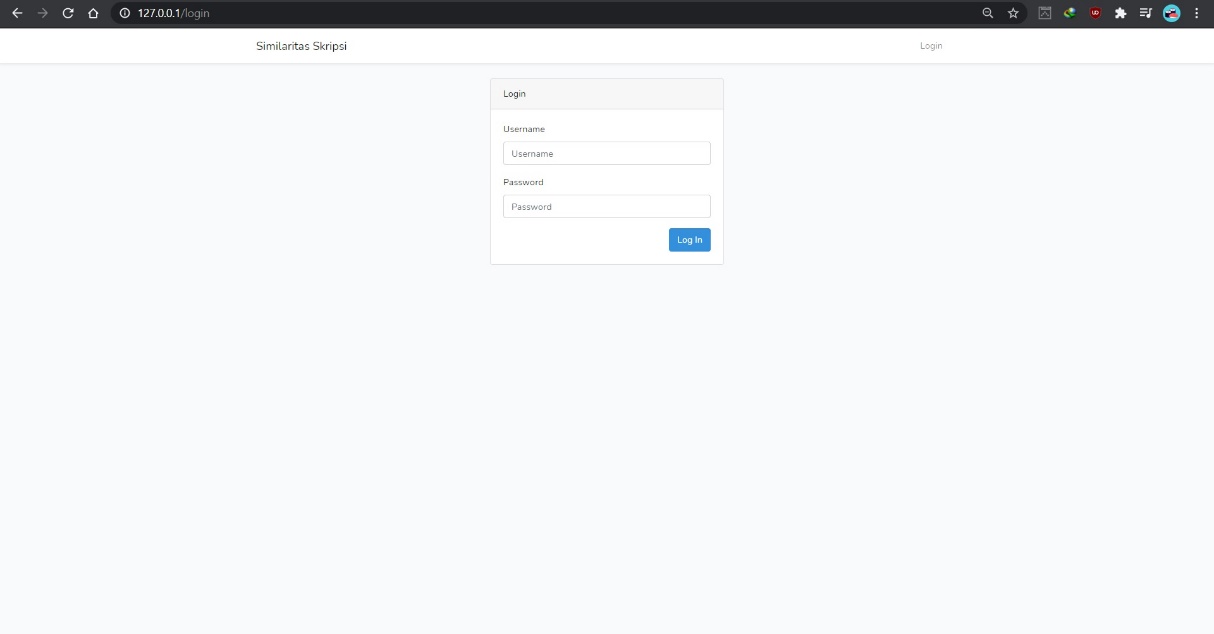
# Hasil dan Analisis

## Hasil Perancangan

Hasil dari perancangan yang telah dilakukan dari arsitektur sistem, diagram alir sistem, data flow diagram (DFD), dan basis data menghasilkan sebuah model aplikasi pendeteksi kemiripan dokumen.

### Antarmuka halaman login

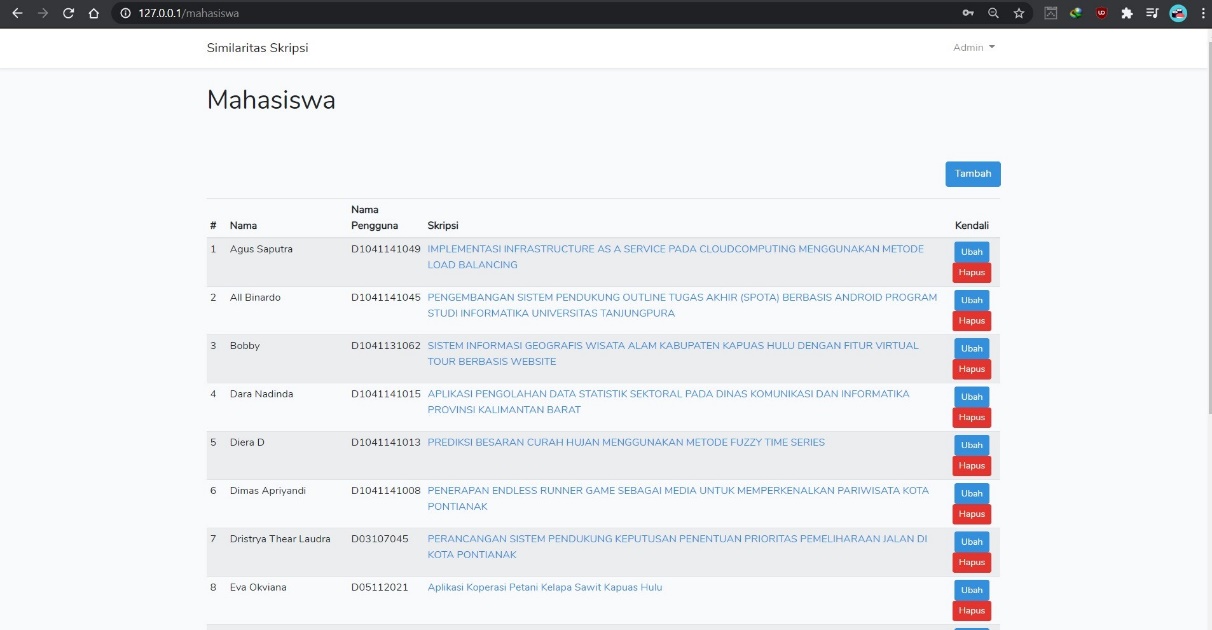
Antarmuka halaman login merupakan tampilan yang pertama kali muncul saat user mengakses halaman web aplikasi. Halaman login digunakan untuk proses autentikasi antara mahasiswa dan admin sebelum mengakses aplikasi. Pada halaman login user menginput username dan password. Tampilan halaman login dapat dilihat pada gambar 4.1 seperti dibawah ini.



**Gambar 4.1** Halaman Login

### Antarmuka Dashboard Admin

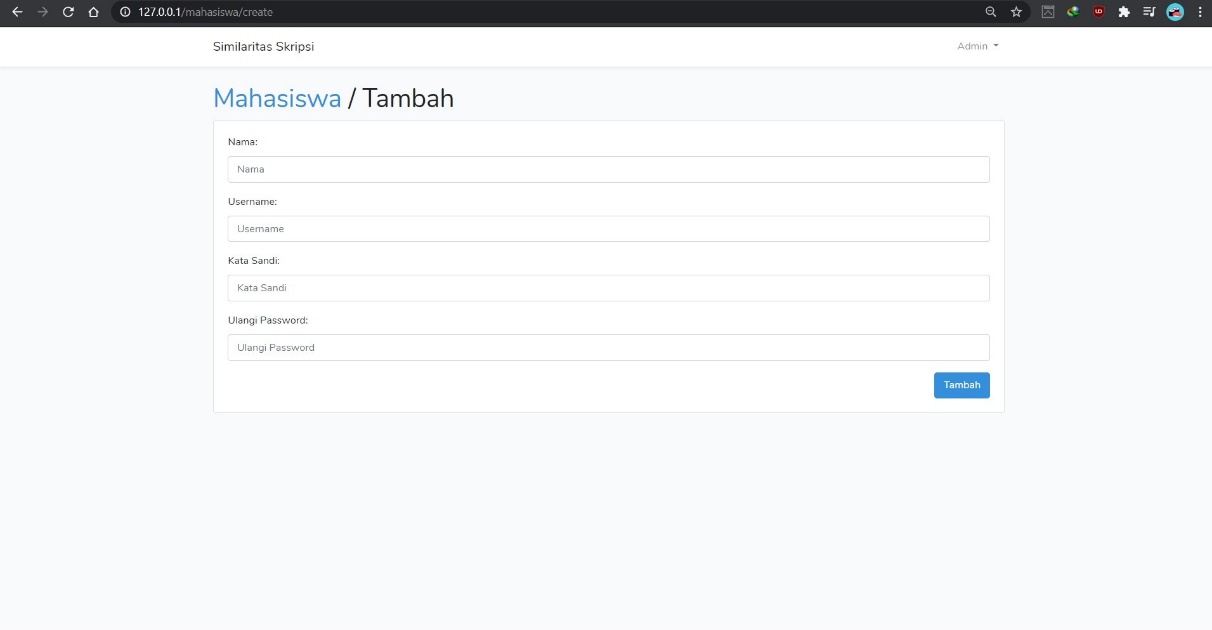
Halaman dashboard admin merupakan halaman setelah admin melakukan login, halaman ini menampilkan seluruh akun mahasiswa yang telah terdaftar di database. Antarmuka dashboard admin dapat dilihat pada gambar 4.2 dibawah ini.



**Gambar 4.2** Dashboard Admin

#### Antarmuka Halaman Tambah Akun

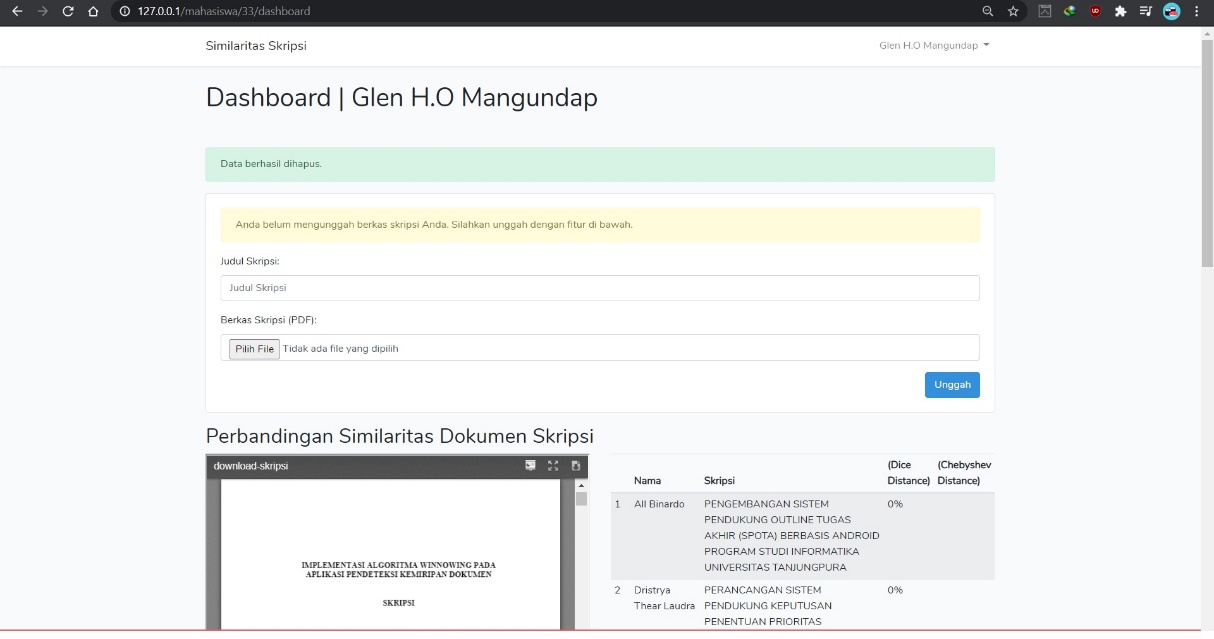
Halaman tambah akun merupakan halaman yang terdapat didashboard admin. Pada halaman ini admin dapat menambahkan data akun mahasiswa yang terdapat form nama, username, kata sandi. Tampilan halaman tambah akun dapat dilihat pada gambar 4.3 dibawah ini.



**Gambar 4.3** Halaman Tambah Akun

### Antarmuka Dashboard Mahasiswa

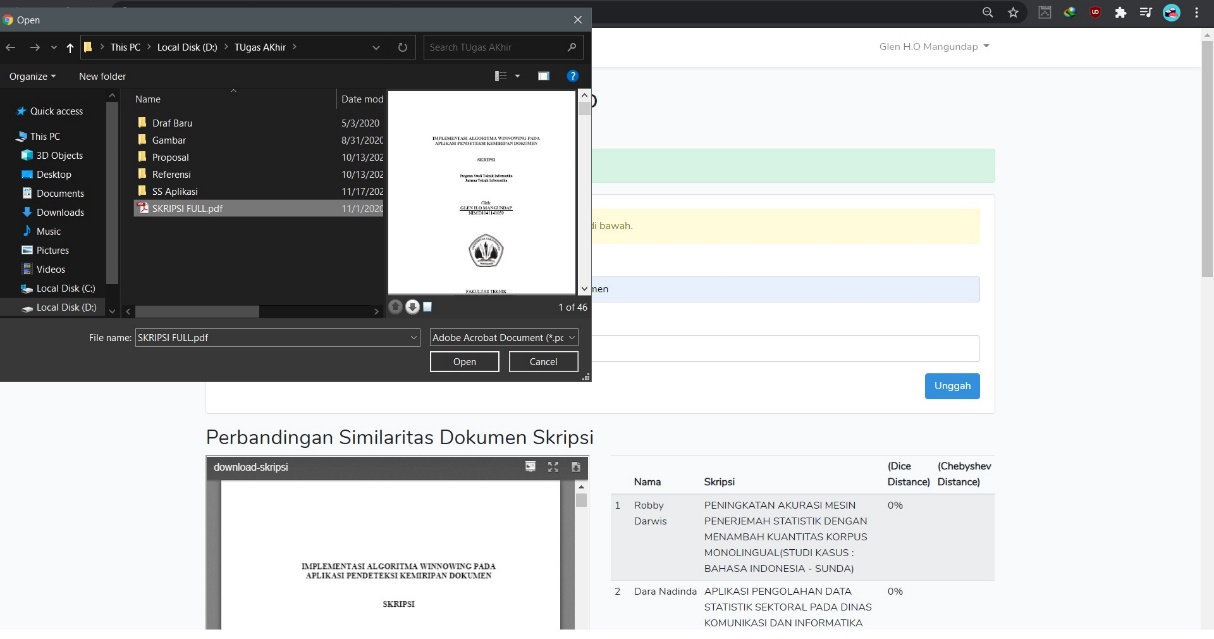
Dashboard mahasiswa merupakan halaman web aplikasi yang pertama kali ditampilkan setelah mahasiswa melakukan login. Pada halaman ini berisi form judul skripsi dan button pilih file skripsi. Tampilan dashboard mahasiswa dapat dilihat seperti gambar 4.4 dibawah ini.



**Gambar 4.4** Dashboard Mahasiswa

#### Antarmuka Upload File

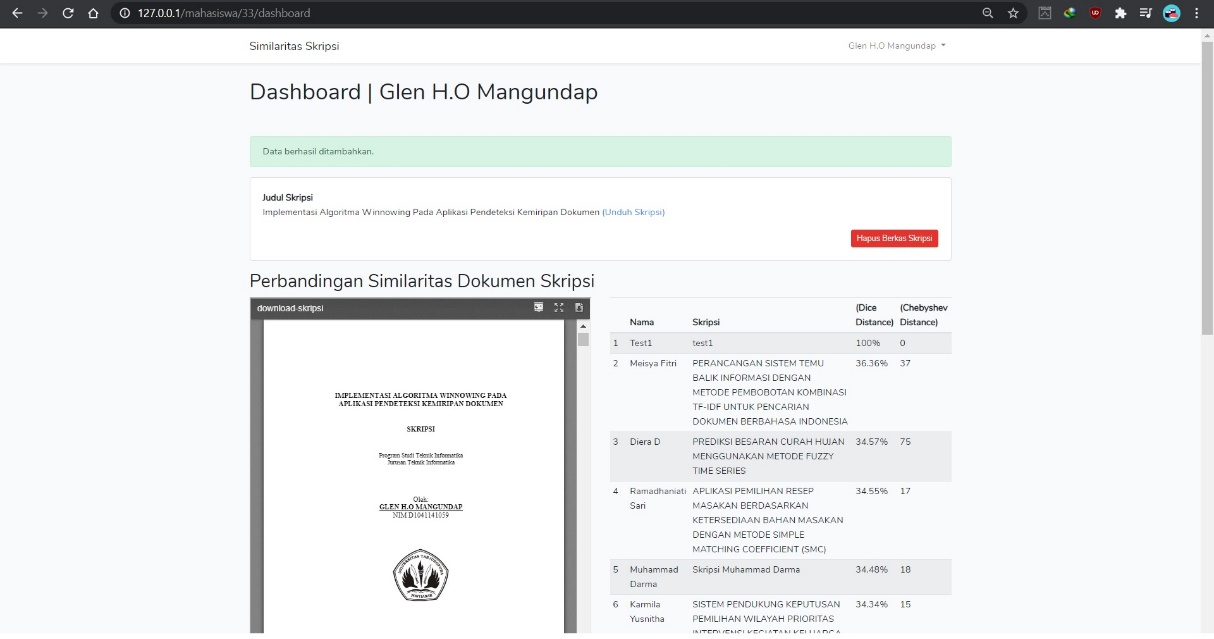
Tampilan upload file merupakan tampilan aplikasi saat mahasiswa akan memilih file skripsi yang akan diupload pada aplikasi. Tampilan upload file dapat dilihat seperti gambar 4.5 dibawah ini.



**Gambar 4.5** Tampilan Upload File

#### Antarmuka Halaman Hasil Perbandingan

Halaman hasil perbandingan merupakan halaman setelah mahasiswa mengupload file skripsi, halaman ini berisi tabel tingkat kemiripan dari skripsi yang diupload terhadap skripsi-skripsi yang sudah ada didata base aplikasi. Tampilan halaman hasil perbandingan dapat dilihat seperti gambar 4.6 dibawah ini.



**Gambar 4.6** Halaman Hasil Perbandingan

## Analisis Hasil Pengujian

Hidup sudah sering diuji janganlah aplikasi diuji juga

# Kesimpulan dan Saran

## Kesimpulan

Xxxxx xxxx xxxx xxxxx xxxx xxx xxx xxx xxx xxx xxx xxx xxxxxx xxxx xxxx.

## Saran

Xxxxx xxxx xxxx xxxxx xxxx xxx xxx xxx xxx xxx xxx xxx xxxxxx xxxx xxxx.

Daftar Pustaka

1. Xxxxxxx xxxxxx
2. Xxxxx Xxxxxx
3. Xxxxx Xxxxxx