

PROPOSAL

IMPLEMENTASI ALGORITMA *RATCLIFF/OBERSHELP* DALAM PENGECEKAN KEMIRIPAN SKRIPSI MAHASISWA

(Studi Kasus: Mahasiswa Teknik Informatika, Universitas Halu Oleo)

Diajukan Untuk Memenuhi

Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



KALEBIANKA EFATA MEYLINA PAGILING

E1E1 18 066

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HALU OLEO

KENDARI

2022

LEMBAR PENGESAHAN

Seminar Proposal

**IMPLEMENTASI ALGORITMA RATCLIFF/OBERSHELP DALAM
PENGECEKAN KEMIRIPAN SKRIPSI MAHASISWA**
(Studi Kasus : Mahasiswa Teknik Informatika, Universitas Halu Oleo)

Adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Universitas Halu Oleo ataupun institusi pendidikan lainnya.

Kendari, Maret 2022

Kalebianka Efata Meylina Pagiling

NIM. E1E1 18 066

Pembimbing I

Bambang Pramono, S.Si., M.T.

NIP. 19710425 200801 1 010

Pembimbing II

Adha Mashur Sajjah, S.T., M.Eng.

NIP. 19910623 201803 1 001

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Teknik Universitas Halu Oleo



TEKUNY, S.Kom., MT

NIP. 19760222 201012 1 001

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	5
1.7 Tinjauan Pustaka.....	5
BAB II LANDASAN TEORI.....	9
2.1 Skripsi.....	9
2.2 Plagiarisme.....	9
2.3 <i>Website</i>	10
2.4 Information Retrieval.....	12
2.5 Algoritma Ratcliff/Obershelp.....	13
2.6 <i>Text Mining</i>	18
2.7 <i>Stop-Word</i>	19
2.8 PDF (<i>Portable Document Format</i>).....	19
2.9 <i>Hyper Text Markup Language</i> (HTML).....	20

2.10	PHP (<i>Hypertext Preprocessor</i>).....	21
2.11	MySQL.....	21
2.12	Database.....	21
2.13	XAMPP.....	22
2.14	Visual Studio Code.....	22
2.15	Unified Modeling Language (UML).....	23
2.15.1	Definisi Unified Modeling Language (UML).....	23
2.15.2	Diagram Unified Modeling Language (UML).....	23
2.16	Rational Unified Process (RUP).....	26
2.17	Metode Pengujian Sistem.....	27
2.17.1	Pengujian Perangkat Lunak.....	27
2.17.2	Black Box Testing.....	28
2.17.3	Confusion Matrix.....	28
BAB III METODOLOGI PENELITIAN.....		30
3.1	Waktu dan Tempat Penelitian.....	30
3.1.1	Waktu Penelitian.....	30
3.1.2	Tempat Penelitian.....	30
3.2	Metode Pengumpulan Data.....	30
3.3	Metode Pengembangan Sistem.....	31
3.4	Analisis Kebutuhan Sistem.....	31
3.4.1	Kebutuhan Fungsional.....	32
3.4.2	Kebutuhan Non-Fungsional.....	32
3.5	Analisis Perancangan Sistem.....	34
3.5.1	Unified Modeling Language (UML).....	34
3.5.2	Perancangan Antarmuka (<i>Interface</i>).....	46

3.6	Pengujian Perangkat Lunak.....	52
3.6.1	<i>Black Box Testing</i>	52
3.6.2	<i>Confusion Matrix</i>	52
DAFTAR PUSTAKA.....		54

DAFTAR GAMBAR

Gambar 2. 1 Contoh Tampilan <i>Website</i> (https://pddikti.kemdikbud.go.id/).....	11
Gambar 2. 2 <i>Information Retrieval</i>	12
Gambar 3. 1 Use Case Diagram Sistem Pengecekan Skripsi Mahasiswa	35
Gambar 3. 2 <i>Activity Diagram Login</i>	40
Gambar 3. 3 <i>Activity Diagram</i> Tambah Data Mahasiswa.....	40
Gambar 3. 4 <i>Activity Diagram</i> Edit Data Mahasiswa.....	41
Gambar 3. 5 <i>Activity Diagram</i> Hapus Data Mahasiswa.....	42
Gambar 3. 6 <i>Activity Diagram</i> Tambah Data Skripsi.....	42
Gambar 3. 7 <i>Activity Diagram</i> Edit Data Skripsi.....	43
Gambar 3. 8 <i>Activity Diagram</i> Hapus Data Skripsi.....	43
Gambar 3. 9 Tambah Data Kategori Skripsi.....	44
Gambar 3. 10 <i>Activity Diagram</i> Edit Data Kategori Skripsi.....	44
Gambar 3. 11 <i>Activity Diagram</i> Hapus Data Kategori Skripsi.....	45
Gambar 3. 12 <i>Activity Diagram</i> Cek Kesamaan Skripsi.....	45
Gambar 3. 13 <i>Class Diagram</i>	46
Gambar 3. 14 Tampilan Beranda Utama.....	47
Gambar 3. 15 Tampilan Halaman <i>Login</i>	47
Gambar 3. 16 Tampilan Halaman Beranda <i>Admin</i>	48
Gambar 3. 17 Tampilan Halaman Data Mahasiswa <i>Admin</i>	48
Gambar 3. 18 Tampilan Halaman Data Skripsi <i>Admin</i>	49
Gambar 3. 19 Tampilan Halaman Data Kategori Skripsi.....	49
Gambar 3. 20 Tampilan Halaman Beranda <i>User</i>	50
Gambar 3. 21 Tampilan Halaman Upload Dokumen Skripsi <i>User</i>	50
Gambar 3. 22 Tampilan Halaman Sesudah <i>User Upload</i> Dokumen.....	51
Gambar 3. 23 Tampilan Halaman Hasil Kesamaan Skripsi.....	51
Gambar 3. 24 Tampilan Halaman Data Skripsi <i>User</i>	52

DAFTAR TABEL

Tabel 2. 1 <i>Use Case Diagram</i> (Julianto & Setiawan, 2019).....	24
Tabel 2. 2 <i>Activity Diagram</i> (Julianto & Setiawan, 2019).....	25
Tabel 2. 3 <i>Class Diagram</i> (Julianto & Setiawan, 2019).....	26
Tabel 3. 1 <i>Gannt Chart</i> Waktu Penelitian	30
Tabel 3. 2 Spesifikasi Perangkat Keras.....	33
Tabel 3. 3 Spesifikasi Perangkat Lunak.....	33
Tabel 3. 4 Deskripsi <i>Case Login (Admin, User)</i>	35
Tabel 3. 5 Deskripsi <i>Case Data Mahasiswa Admin</i>	36
Tabel 3. 6 Deskripsi <i>Case Data Skripsi Admin</i>	36
Tabel 3. 7 Deskripsi <i>Case Kategori Skripsi Admin</i>	37
Tabel 3. 8 Deskripsi <i>Case Upload Skripsi User</i>	37
Tabel 3. 9 Deskripsi <i>Case Cek Kesamaan Skripsi</i>	38
Tabel 3. 10 Deskripsi <i>Case Lihat Data Skripsi</i>	39
Tabel 3. 11 <i>Confusion Matrix</i>	52

BAB I

PENDAHULUAN

1.1 Latar Belakang

Skripsi adalah penelitian yang disusun oleh mahasiswa tingkat akhir dengan menerapkan kaidah ilmiah sesuai bidang keilmuan program studi atau jurusan sebagai persyaratan wajib untuk memperoleh gelar sarjana. Penulisan skripsi ditempuh melalui pengajuan proposal penelitian kepada dosen pembimbing, mengikuti seminar proposal, mendapat Surat Keputusan (SK) tentang topik skripsi, melakukan penelitian dan penulisan hasil penelitian melalui proses bimbingan, serta pelaksanaan sidang skripsi. Hasil penulisan skripsi wajib disebarluaskan kepada khalayak luas minimal diunggah di dalam repositori digital *library* pendidikan tinggi (Darmalaksana, 2019). Program Studi Teknik Informatika Universitas Halu Oleo sebagai salah satu program studi Strata Satu (S1) tentunya juga mensyaratkan mahasiswanya membuat suatu karya tulis ilmiah yaitu skripsi untuk memperoleh gelar sarjana.

Perkembangan dunia teknologi yang semakin maju turut serta mempengaruhi dunia Pendidikan karena pemanfaatan teknologi tersebut. Teknologi di dunia pendidikan sangat memudahkan proses belajar-mengajar saat ini. Kemajuan teknologi di dunia Pendidikan khususnya perguruan tinggi yakni dengan pemanfaatan perangkat keras dan perangkat lunak seperti komputer, proyektor, *smartphone*, aplikasi pengolah kata, *spreadsheet*, *learning management system*, *video-conference* dan aplikasi lainnya.

Komputer dan aplikasi ini sangat membantu mahasiswa di perguruan tinggi. Alat-alat ini digunakan dalam pengolahan informasi, penyelesaian dan pengumpulan tugas kuliah mahasiswa. Informasi yang ada diproses melalui perangkat lunak komputer menghasilkan *softcopy* yang tersimpan dalam komputer. *File softcopy* ini sangat mudah untuk disebarkan dan disalin baik sebagian ataupun seluruh isinya. Akan tetapi kemudahan ini terkadang malah disalahgunakan oleh mahasiswa dimana terkadang mereka melakukan tindakan menjiplak tugas mahasiswa lainnya yang merupakan kategori perilaku plagiarisme.

Plagiarisme adalah perbuatan sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai (Menteri Pendidikan Nasional, 2010). Aksi plagiarisme merupakan tindakan yang sangat merugikan, tidak hanya merugikan orang lain yang karya tulisnya dijiplak tetapi merugikan juga orang yang melakukan tindakan plagiarisme itu sendiri. Dengan tindakan plagiarisme, secara tidak langsung dapat mematikan kreatifitas seseorang dan menimbulkan kecendrungan (Novantara, 2018).

Pengecekan kemiripan skripsi mahasiswa perlu dilakukan untuk mendeteksi kesamaan dokumen skripsi pada mahasiswa. Dengan melakukan pengecekan kemiripan skripsi dapat mencegah terjadinya plagiarisme dokumen skripsi pada mahasiswa karena mengingat skripsi merupakan suatu karya ilmiah yang dibuat oleh mahasiswa dan salah satu syarat kelulusan pada jenjang sarjana Strata Satu (S1) Perguruan Tinggi.

Skripsi dapat dianggap sebagai sebuah dokumen berupa kumpulan *string* yang sangat panjang. Oleh karena itu pengecekan kemiripan dokumen skripsi dapat memanfaatkan algoritma-algoritma perbandingan atau pengecekan kesamaan dua *string*.

Algoritma untuk mengukur kesamaan dua *string* yaitu Algoritma *Ratcliff Obershelp*, Algoritma *Rabin-Karp* dan juga Algoritma *Jaro-Winkler*. Berdasarkan penelitian yang dilakukan oleh Yusuf dkk (2019) Algoritma *Ratcliff/Obershelp* mendapatkan nilai *similarity* yang lebih baik dibandingkan dengan algoritma *Rabin-Karp*. Selain itu, algoritma *Ratcliff/Obershelp* juga lebih unggul dalam kecepatan dan konsistensinya daripada algoritma *Rabin-Karp*. Algoritma *Rabin-Karp* membutuhkan waktu 2 sampai dengan 7 *ms* untuk pengujian dengan Data-Uji DU4, DU3, DU2, DU1, dan DU sedangkan algoritma *Ratcliff/Obershelp* hanya memerlukan waktu 0,1 sampai dengan 2 *ms* untuk pengujian data yang sama. Hal ini dikarenakan algoritma *Rabin-Karp* cenderung membutuhkan waktu yang banyak dalam tahapan prosesnya.

Algoritma *Rabin-Karp* memiliki tingkat konsistensi yang rendah jika dibandingkan dengan algoritma *Ratcliff/Obershelp*. Algoritma *Ratcliff/Obershelp* hanya memiliki 2 nilai kemiripan (94 dan 95) untuk pengujian dengan menggunakan 10 data yang berbeda pada Data-Uji 90% (DU1) sedangkan Algoritma *Rabin-Karp* menghasilkan 5 nilai kemiripan yang berbeda (93, 94, 95, 96 dan 97) pada pengujian yang sama.

Berdasarkan penelitian yang dilakukan oleh Ilyankou (2014), Algoritma *Ratcliff/Obershelp* lebih efisien 4% jika dibandingkan dengan algoritma *Jaro-Winkler*. Algoritma *Ratcliff/Obershelp* memberikan skor kesamaan lebih tepat ketika sebuah kata mengandung satu kesalahan ketik, seperti huruf yang salah ketik dan beberapa huruf yang hilang atau urutan karakter yang salah. Algoritma *Ratcliff/Obershelp* bekerja sangat baik ketika karakter yang salah ketik (atau beberapa) adalah karakter pertama atau terakhir dalam *string*.

Dari pembahasan di atas, maka penelitian ini akan menggunakan Algoritma *Ratcliff/Obershelp* sebagai algoritma yang paling efektif dan efisien untuk menghitung kesamaan dokumen skripsi mahasiswa yang akan diterapkan ke dalam sistem pengecekan kesamaan skripsi mahasiswa dan menghasilkan *output* kesamaan dari dokumen skripsi yang tersimpan pada sistem. Dengan adanya sistem tersebut diharapkan dapat membantu mahasiswa serta pihak dosen maupun jurusan Teknik Informatika, Universitas Halu Oleo dalam mencegah serta mengurangi plagiarisme dokumen skripsi antar mahasiswa. Oleh karena itu, dilakukan penelitian dengan judul “**Implementasi Algoritma *Ratcliff/Obershelp* Dalam Pengecekan Kemiripan Skripsi Mahasiswa**”.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana rancang bangun sistem pengecekan kemiripan skripsi mahasiswa berbasis *website*?
2. Bagaimana menerapkan Algoritma *Ratcliff/Obershelp* pada sistem pengecekan kemiripan skripsi mahasiswa berbasis *website*?

3. Bagaimana mengembangkan sistem pengecekan skripsi mahasiswa agar dapat membantu dalam mendeteksi kesamaan dokumen skripsi dan mencegah plagiarisme?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya berfokus pada penerapan metode *Ratcliff/Obershelp* dalam pengecekan kemiripan skripsi mahasiswa Teknik Informatika.
2. Data yang digunakan yaitu data skripsi mahasiswa Teknik Informatika angkatan 2016-2017 yang sudah terdaftar pada sistem.
3. Sistem ini hanya digunakan dalam lingkup Jurusan Teknik Informatika, Universitas Halu Oleo.
4. Dokumen yang akan diinputkan oleh *user* untuk mengecek kemiripan skripsi merupakan *file* dengan format pdf (*Portable Document Format*).

1.4 Tujuan Penelitian

Adapun tujuan penelitian dari perancangan sistem Implementasi Metode *Ratcliff/Obershelp* Dalam Pengecekan Kemiripan Skripsi Mahasiswa pada Jurusan Teknik Informatika Universitas Halu Oleo adalah sebagai berikut:

1. Untuk membangun dan mengembangkan sebuah sistem pengecekan kemiripan mahasiswa Teknik Informatika Universitas Halu Oleo berbasis *website*.
2. Untuk menerapkan metode *Ratcliff/Obershelp* pada sistem pengecekan kemiripan Skripsi Mahasiswa Teknik Informatika Universitas Halu Oleo.

1.5 Manfaat Penelitian

Adapun manfaat penelitian dari perancangan sistem Implementasi Metode *Ratcliff/Obershelp* Dalam Pengecekan Kemiripan Skripsi Mahasiswa pada Jurusan Teknik Informatika Universitas Halu Oleo adalah dapat mempermudah

user dalam mengecek kesamaan skripsi dengan beberapa skripsi yang tersimpan dalam sistem.

1.6 Sistematika Penulisan

Adapun sistematika penulisan laporan proposal penelitian ini yaitu sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang deskripsi umum isi laporan yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

BAB II LANDASAN TEORI

Bab ini berisi teori-teori penunjang yang berhubungan dengan penelitian berupa teori dan metode yang di gunakan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tahapan tentang proses dan prosedur pengumpulan data, prosedur analisis metode, prosedur pengembangan sistem waktu dan tempat penelitian.

1.7 Tinjauan Pustaka

Menurut penelitian yang dilakukan oleh Joane dkk (2017) dengan judul Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma *Ratcliff/Obershelp* bahwa perkembangan teknologi yang semakin berkembang ini yang digunakan dalam pengolahan informasi yang dapat membantu seseorang atau masyarakat terlebih dalam bidang akademik terutama oleh mahasiswa sebagai contoh dalam penyelesaian dan pengumpulan tugas kuliah sekarang ini, mahasiswa hanya perlu mengirimkan tugasnya dengan *file softcopy*.

Akan tetapi kemudahan ini tidak selalu berdampak positif ada juga dampak negatifnya yaitu dengan menyalahgunakannya dengan menjiplak *softcopy*

dokumen teks tugas mahasiswa lainnya dimana hal tersebut merupakan tindakan plagiat yang menghambat kreatifitas mahasiswa dalam mengerjakan tugas kuliah.

Oleh karena hal itu penelitian ini membangun sebuah aplikasi yang dikhususkan untuk Institusi Universitas Sam Ratulangi Manado untuk mendeteksi kemiripan dokumen teks dalam hal ini tugas mahasiswa yang diberi nama Aplikasi Pendeteksian Kemiripan Tugas Mahasiswa. Aplikasi Pendeteksian Aplikasi Kemiripan Tugas Mahasiswa dibangun untuk memudahkan dosen dalam mendeteksi plagiat dokumen tugas mahasiswa dengan menerapkan metode *Ratcliff/Obershelp* dengan konsep pencocokan dari algoritma ini dengan menemukan sub *string* terpanjang yang memiliki kesamaan dari *string* S1 dan S2 yang disebut *anchor*.

Ratcliff/Obershelp menggunakan proses yang sama untuk memutuskan seberapa mirip dua pola satu dimensi. Karena *string* teks merupakan satu dimensi, algoritma ini mengembalikan nilai yang dapat di gunakan sebagai faktor kepercayaan atau persentase, menunjukkan kesamaan dua *string* (Joane dkk., 2017).

Penelitian selanjutnya terkait algoritma *Ratcliff/Obershelp* Aritonang (2020) dengan judul Rancang Bangun Aplikasi Deteksi Kemiripan Dua Gambar Menggunakan Algoritma *Ratcliff/Obershelp* yaitu mendeteksi kemiripan dua gambar dengan menggunakan algoritma *Ratcliff/Obershelp*, dengan algoritma ini digunakan untuk mencari kemiripan dua buah gambar dengan mengambil nilai RGB dari masing-masing RGB dari pixel tiap gambar, Proses pencarian untuk mendapatkan gambar dengan kemiripan tertinggi. Format gambar yang bisa diinputkan kedalam sistem adalah citra bitmap.

Pemetaan bitmap gambar grafis komputer yang terdiri atas titik-titik yang membentuk baris dan kolom, citra yang terdiri atas titik dan pixel. Pada format bitmap, citra disimpan sebagai sebagai suatu matriks dimana masing-masing elemennya digunakan untuk menyimpan informasi warna untuk menyimpan informasi warna untuk setiap pixel. Penelitian ini menggunakan algoritma *Ratcliff/Obershelp* merupakan algoritma pencarian citra yang dilakukan dengan pencocokan dari nilai RGB yang sama pada citra. Proses pencarian kemiripan

gambar dapat membantu mengatasi masalah pada tingkat kemiripan gambar dan pencarian gambar lebih cepat dan akurat.

Penelitian selanjutnya terkait perbandingan algoritma *Ratcliff/Obershelp* Yusuf dkk (2019) dengan judul Analisis Perbandingan Algoritma *Rabin-Karp* dan *Ratcliff/Obershelp* Untuk Menghitung Kesamaan Teks Dalam Bahasa Indonesia. Dalam penelitian ini dilakukan analisis perbandingan antara algoritma *Rabin-Karp* dan algoritma *Ratcliff/Obershelp* untuk menghitung persentase nilai *similarity* dokumen teks. Perbandingan yang ingin diteliti adalah kinerja dan hasil akhir dari kedua algoritma tersebut.

Penelitian ini diharapkan dapat membantu peneliti untuk memilih algoritma yang lebih efisien untuk mendeteksi kemiripan kalimat pada suatu dokumen. Hasil pengujian didapatkan bahwa algoritma *Ratcliff/Obershelp* mendapatkan nilai *similarity* yang lebih baik sebesar $\pm 3\%$ dibandingkan dengan algoritma *Rabin-Karp*. Selain itu, algoritma *Ratcliff/Obershelp* juga lebih unggul dalam kecepatan dan konsistensinya dari algoritma *Rabin-Karp*. Hal ini dikarenakan algoritma *Rabin-Karp* cenderung membutuhkan waktu yang banyak dalam tahapan prosesnya.

Tinjauan selanjutnya terkait algoritma *Ratcliff/Obershelp* Jacob dkk (2019) dengan judul Rancang Bangun Aplikasi Kemiripan Dokumen Dengan Sumber-Sumber *Internet* dimana perkembangan teknologi yang semakin pesat dan segala bidang relatif bisa dipandang sangat relevan berhubungan dengan komputer sehingga dalam perkembangan teknologi tersebut dapat sangat mempengaruhi kehidupan masyarakat terlebih dalam dunia perguruan tinggi di Universitas Sam Ratulangi.

Universitas Sam Ratulangi telah menerapkan sistem *E-learning* (*Electronic Learning*) atau perkuliahan berbasis *online* dalam proses belajar mengajar jarak jauh dengan menggunakan teknologi *internet* dan komputer sebagai media penyampaian informasi, digunakan untuk saling berbagi informasi antara dosen dan mahasiswa atau diskusi antara dosen dan mahasiswa baik materi kuliah maupun tugas kuliah.

Hal tersebut tentu memberikan dampak positif dalam lingkungan akademis, tapi seiring dengan kemudahan yang diberikan terkadang disalahgunakan oleh beberapa orang dalam menyelesaikan pekerjaan atau tugas-tugas yang diberikan. Penyalahgunaan sering dilakukan oleh mahasiswa, contohnya dalam mengerjakan tugas kuliah, mahasiswa sering mengerjakan tugas kuliah dengan mengambil data dari sumber-sumber *internet* yang tidak jelas misalnya wikipedia, *wordpress* ataupun *blogspot*.

Hal tersebut tentu sangatlah berdampak buruk bagi mahasiswa selain itu dosen dan pengajar juga direpotkan dalam menganalisa satu per satu tugas mahasiswa, apalagi jika ingin membandingkan tugas dengan data dari sumber-sumber *internet* yang tidak jelas secara manual. Cara tersebut kurang efektif dan efisien mengingat jumlah mahasiswa yang tidak sedikit sehingga memerlukan waktu yang lebih lama untuk menganalisa tugas mahasiswa dengan data-data dari sumber *internet* yang tidak jelas.

Oleh karena itu perlu adanya aplikasi yang mampu mengukur tingkat kemiripan dokumen teks tugas mahasiswa yang dikumpulkan dengan data dari sumber-sumber *internet* yang tidak jelas. Dengan menerapkan metode *Ratcliff/Obershelp* pada sistem untuk mengukur kemiripan dokumen teks tugas mahasiswa, Aplikasi pendeteksi kemiripan telah berhasil di buat dan mampu mengukur tingkat kemiripan dokumen teks tugas mahasiswa dengan sumber-sumber *internet* (wikipedia), dengan menggunakan algoritma pengujian *Ratcliff/Obershelp*.

Pengujian berdasarkan 5 dokumen mahasiswa yang diuji kemiripannya dengan artikel dari wikipedia, dalam pengujian berkas tugas dengan artikel dimiliki hasil yaitu ada yang mendekati prediksi, ada yang di bawah prediksi, dan ada yang melewati prediksi. Jadi bisa dikatakan bahwa aplikasi berfungsi sebagaimana yang diharapkan. Waktu eksekusi yang diperlukan untuk mendeteksi kemiripan 80% - 100% yaitu 1 menit 32 detik. Banyaknya karakter dalam dokumen tugas mahasiswa mempengaruhi waktu eksekusi dari proses pengujian dengan artikel wikipedia.

BAB II

LANDASAN TEORI

2.1 Skripsi

Skripsi adalah karya tulis ilmiah yang menguraikan suatu pokok soal dengan cukup penting dalam suatu cabang ilmu sebagai cabang penelitian pustaka dan/atau lapangan yang dilakukan oleh seorang mahasiswa berdasarkan penugasan akademik dari perguruan tingginya untuk menjadi salah satu syarat kelulusannya sebagai sarjana (Azis, 2021).

Karya tulis ilmiah, merupakan gabungan dari tiga suku kata. menurut Kamus Besar Bahasa Indonesia, karya, dapat diartikan sebagai hasil sebuah usaha, upaya, perbuatan atau ciptaan, sedangkan tulis, atau menulis memiliki arti segala kegiatan yang terkait dengan huruf, angka, pena, atau media tulis yang lain. Yang ketiga adalah ilmiah, menurut Kamus Besar Bahasa Indonesia berarti bersifat ilmu, secara ilmu pengetahuan, atau memenuhi syarat (kaidah) ilmu pengetahuan. Jika diartikan secara menyeluruh, karya tulis ilmiah merupakan sebuah karya yang dihasilkan dari kegiatan menulis, dengan menggunakan penerapan kaidah ilmiah, mengutamakan aspek rasionalitas, mengusung permasalahan yang bersifat obyektif serta faktual (Ahmad, 2021).

Penulisan karya tulis ilmiah, menggunakan kata yang tidak ambigu, atau memiliki makna ganda, maka diperlukan penggunaan gaya bahasa yang lugas, eksplisit, menggunakan variasi istilah ilmiah yang sesuai dengan aturan pedoman umum ejaan bahasa Indonesia (Ahmad, 2021).

2.2 Plagiarisme

Plagiarisme atau plagiat adalah suatu perbuatan menjiplak ide, gagasan atau karya orang lain yang selanjutnya diakui sebagai karya sendiri atau menggunakan karya orang lain tanpa menyebutkan sumbernya sehingga menimbulkan asumsi yang salah atau keliru mengenai asal muasal dari suatu ide, gagasan atau karya (Riadi, 2019).

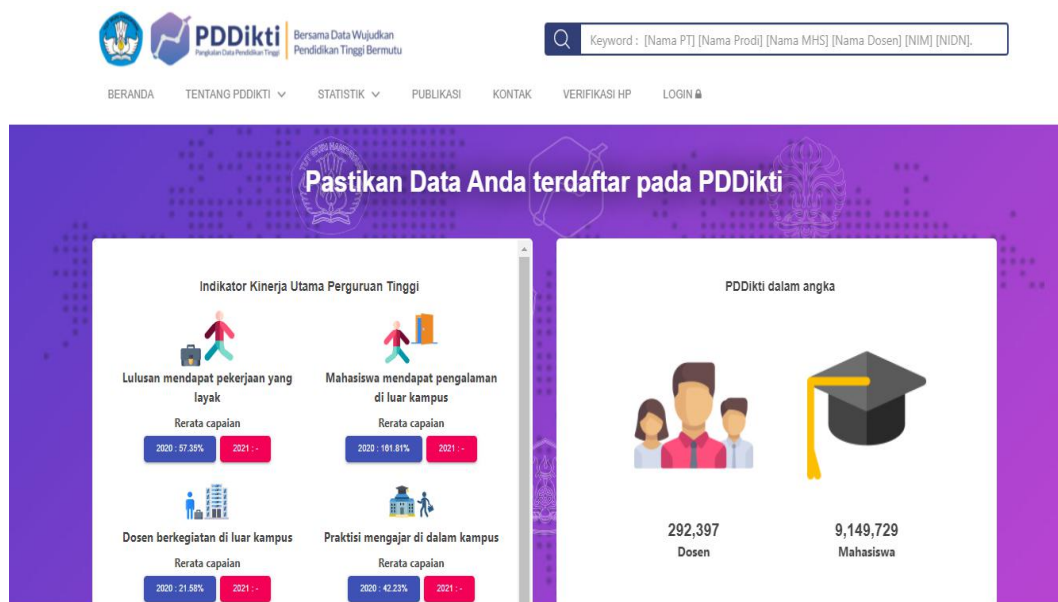
Plagiarisme atau sering disebut plagiat adalah kegiatan menjiplak, mengambil karangan, pendapat dan sebagainya dari orang lain untuk dijadikan seolah karangan dan pendapat sendiri tanpa meminta ijin dan menyertakan sumber yang dicatutnya. Masalah plagiarisme ini memang sudah menjadi masalah serius di dunia pendidikan yang selalu ada di setiap waktu dan dibelahan dunia manapun. Hal ini tentu memerlukan perhatian khusus bagi pemerhati dunia pendidikan untuk mencari solusi dan formula agar kegiatan plagiarisme dapat diminimalisir karena berdampak tidak sehat bagi dunia pendidikan. Plagiarisme di kalangan mahasiswa bisa saja terjadi karena sikap mahasiswa terhadap ketersediaan sumber bacaan yang relatif mudah diakses lewat *internet* sehingga mereka mengambil jalur pintas untuk membuat karya ilmiah tanpa bersusah payah berfikir. Penyebab lainnya mungkin dikarenakan masih terbatasnya referensi buku bacaan rujukan yang dianjurkan dosen yang masih terbatas atau bahkan belum bisa diketemukan di perpustakaan setempat (Hermawan, 2019).

2.3 *Website*

Website atau situs *web* adalah sekumpulan halaman *web* yang saling berhubungan dan dapat diakses secara *online* oleh perorangan, perusahaan, institusi pendidikan, pemerintahan, dan organisasi. Dengan adanya *website*, informasi yang diinginkan akan mudah dan cepat ditemukan bagi yang membutuhkan, kemudian mudah didesiminasikan oleh pemilik informasi. Oleh karena itu, *website* juga dapat digunakan sebagai sarana bisnis yang baik, di samping sebagai sarana hiburan (Gani & Achmad, 2019).

Website adalah salah satu teknologi yang terus dikembangkan di era global. Pada awalnya, *web* digunakan sebagai penyampaian informasi melalui perangkat PC (*Private Computer*). Perangkat untuk mengakses situs *web* kini tidak terbatas pada PC. Namun, dapat juga diakses melalui gawai (*gadget*) seperti *smartphone*. Semakin berkembangnya *web*, maka semakin banyak variasi media pembelajaran. Kemudahan dalam semua aspek, menjadikan *website* sangat potensial untuk diaplikasikan menjadi media pembelajaran yang interaktif dengan tawaran *web* yang beragam dan menarik. Revolusi teknologi yang kini mencapai era

digital 4.0 dimana segala sesuatunya tidak dapat dipisahkan dengan teknologi. Tentu hal ini berdampak pada pendidikan yang menuntut penggunaan teknologi dalam pembelajaran atau *E-learning* (Darmawan dkk., 2022).



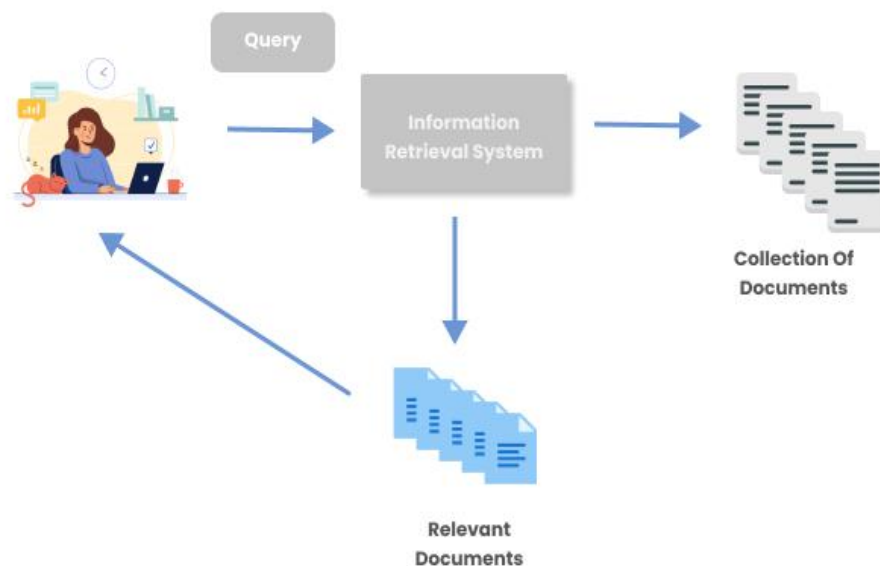
Gambar 2. 1 Contoh Tampilan *Website* (PDDikti, 2022)

Website atau situs adalah kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti media sosial (Facebook, Instagram, Twitter dll), portal berita (detik, cnnindonesia, tribunnews, dll), forum diskusi, dan *e-commerce* (bukalapak, tokopedia, dll). Dalam sisi pengembangannya, *website* statis hanya bisa *diupdate* oleh pemiliknya saja, sedangkan *website* dinamis bisa *diupdate* oleh pengguna maupun pemilik (Saputro, 2019).

2.4 *Information Retrieval*

Sistem Temu Kembali Informasi (*Information Retrieval*) digunakan untuk menemukan kembali informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Salah satu aplikasi umum dari sistem temu kembali informasi adalah *Search Engine* atau Mesin Pencarian. Saat ini Sistem Temu Kembali Informasi bukan hanya berbasis teks, tetapi juga berbasis isi visual dari suatu citra (Ode dkk., 2019).

Sistem Temu Kembali Informasi (*Information Retrieval*) digunakan untuk mengurangi apa yang disebut kelebihan informasi. Sistem temu kembali informasi adalah sistem perangkat lunak yang menyediakan akses ke buku, jurnal, surat, catatan, film, rekaman audio, surat kabar, buletin, poster seni, lagu, dan dokumen lainnya; menyimpan dan mengelola dokumen-dokumen tersebut (Ahmad dan Prasetyo 2021).



Gambar 2. 2 *Information Retrieval*

Information need merupakan topik dimana *user* ingin tahu lebih jauh, sedangkan *query* merupakan cara *user* berkomunikasi dengan komputer untuk memperoleh informasi yang diinginkan (Yossy, 2020).

Information Retrieval System berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Ekspresi kebutuhan pengguna yang disebut

query, juga tidak memiliki struktur. Hal ini yang membedakan *IR system* dengan sistem basis data. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut. Sebagai suatu sistem, *IR system* memiliki beberapa bagian yang membangun sistem secara keseluruhan. Bagian-bagian yang terdapat pada *IR system* digambarkan pada gambar diatas (Pratama, 2018).

Sistem temu kembali informasi adalah untuk memenuhi kebutuhan informasi pengguna dengan *retrieve* semua dokumen yang mungkin relevan, pada waktu yang sama *retrieve* sesedikit mungkin dokumen yang tidak relevan (Chaosakul, 2015).

2.5 Algoritma Ratcliff/Obershelp

Algoritma *Ratcliff/Obershelp* merupakan algoritma yang pencocokan *pattern* atau pola yang diperkenalkan oleh John W. Ratcliff dan John A. Obershelp pada tahun 1983. Algoritma ini berdampak pada industri *educational software*. Algoritma *Ratcliff/Obershelp* terkenal sebagai *Gestalt Pattern Matching* dimana algoritma ini menilai seberapa mirip pola pada dua *string* yang berbeda. Sebelumnya, *educational software* seringkali hanya menawarkan tes pilihan ganda, karena untuk jawaban yang diketik oleh *user* diperlukan algoritma untuk memproses dan memeriksa data yang dimasukkan apakah sudah sesuai dengan yang ada pada *database*. Algoritma *Ratcliff/Obershelp* membantu untuk menyelesaikan permasalahan diatas, Sama halnya dengan Algoritma *Jaro-Winkler Distance*, *Ratcliff/Obershelp* mengembalikan nilai dari 0 sampai 1, dimana 1 adalah kecocokan yang lengkap untuk dua *string* yang diberikan (Ilyankou, 2014).

Algoritma *Ratcliff/Obershelp* yang dapat menilai seberapa mirip pola dari dua *string* yang berbeda, dengan algoritma ini yang dapat mendeteksi kemiripan teks dengan cara membandingkan *string* yang berbeda sehingga spasi yang terdapat pada dokumen teks harus dihilangkan sehingga dokumen teks menjadi sebuah *string* (Joane dkk., 2017).

Algoritma *Ratcliff/Obershelp* dapat dinyatakan dengan rumus:

(2. 1)

$$D_{ro} = \frac{2 * K_m}{|S_1| + |S_2|}$$

Dimana:

K_m = Jumlah karakter yang sama

$|S_1|$ = Panjang dari *String* 1

$|S_2|$ = Panjang dari *String* 2

Contoh Kasus Algoritma *Ratcliff/Obershelp*

Mencari kesamaan kedua *string* UNIVERSITY dan UNIVERSITAS

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
S1	U	N	I	V	E	R	S	I	T	Y	
S2	U	N	I	V	E	R	S	I	T	A	S

1. Panjang dari *string* S_1 :

$$|S_1| = 10$$

Panjang dari *string* S_2 :

$$|S_2| = 11$$

2. *Substring* yang terpanjang dimiliki oleh kedua *string* adalah UNIVERSIT.

Maka UNIVERSIT merupakan sebuah *anchor*, dan $K_m = |\text{UNIVERSIT}| = 9$

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
S1	U	N	I	V	E	R	S	I	T	Y	
S2	U	N	I	V	E	R	S	I	T	A	S

3. Disebelah kanan dari *anchor* terdapat karakter Y dan A,S. Karena kedua karakter tersebut berbeda maka mereka tidak cocok. Maka nilai dari K_m tetap 9. Jadi kita memiliki semua data yang diperlukan untuk menghitung nilai dari *Ratcliff/Obershelp*.

Penilaian *Ratcliff/Obershelp* untuk *string* UNIVERSITY dan UNIVERSITAS adalah:

$$D_{ro} = \frac{2*9}{|10|+|11|} = \frac{18}{21} = 0,857$$

Jadi, dari kedua *string* UNIVERSITY dan UNIVERSITAS memiliki nilai kesamaan 0,857 yang bisa dikatakan sama.

Sebelum mengimplementasikan algoritma *Ratcliff/Obershelp* perlu dilakukan penganalisaan data dengan melakukan *text preprocessing* dan *processing*. Dimana dalam *text preprocessing* akan dilakukan pembersihan data teks dan pada *processing* dimana algoritma *Ratcliff/Obershelp* akan diimplementasikan.

1. *Text Preprocessing*

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) terhadap teks. Tujuan dari tahapan ini adalah untuk mempersiapkan teks menjadi data yang akan mengalami pengolahan lebih lanjut. Berikut adalah tahapan *text preprocessing* yang akan digunakan adalah:

a. *Data Input*

Berikut ini adalah contoh paragraf yang akan dibersihkan melalui tahapan *text preprocessing* sebelum kemudian diolah lebih lanjut.

Olahraga, secara teratur bisa membuat tubuh menjadi SEHAT dan BUGAR.

b. *Case Folding*

Case folding merupakan tahapan dimana semua huruf yang ada didalam dokumen diubah menjadi huruf kecil dan hanya huruf 'a' sampai dengan 'z' yang dapat diterima. Karakter selain huruf dihilangkan dan dianggap delimiter (pembatas).

Input: Olahraga, secara teratur bisa membuat tubuh menjadi SEHAT dan BUGAR ,

Output: olahraga, secara teratur bisa membuat tubuh menjadi sehat dan bugar.

c. *Cleansing*

Pada tahap ini dokumen akan dibersihkan dari tanda baca dan simbol-simbol yang ada di dalam dokumen.

Input: olahraga, secara teratur bisa membuat tubuh menjadi sehat dan bugar. *Output:* olahraga secara teratur bisa membuat tubuh menjadi sehat dan bugar

d. *Filtering*

Pada tahap *filtering* akan menggunakan algoritma *stoplist* dimana kata-kata yang kurang penting dan tidak memiliki pengaruh akan dibuang. Penghapusan *stopword* dilakukan dengan membandingkan kata-kata yang ada didalam dokumen dengan *stopword* yang sebelumnya telah disimpan. Jika ada kata yang sama, kata tersebut akan dihilangkan. Kata-kata yang dimaksudkan sebagai *stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopword* adalah ‘yang’, ‘dan’, ‘di’, ‘dari’ dan lain-lain. Berikut adalah contoh proses dari *filtering* dengan algoritma *stoplist*.

Input: olahraga secara teratur bisa membuat tubuh menjadi sehat dan bugar

Output: olahraga teratur membuat tubuh sehat bugar

e. Penghapusan Spasi

Setelah kalimat dibersihkan maka akan dilakukan penghapusan spasi agar menjadi satu *string*. Penghapusan spasi dilakukan karena algoritma yang digunakan merupakan pendeteksian *string*. Berikut adalah hasil dari kalimat yang telah menjadi *string*.

Input: olahraga teratur membuat tubuh sehat bugar

Output: olahragateraturmembuat tubuhsehatbugar

2. *Text Processing*

Pada proses ini adalah dimana algoritma *Ratcliff/Obershelp* akan di implementasikan. Cara kerja algoritma ini adalah untuk menemukan *subsequence* pencocokan terpanjang berdekatan dari dua data *string*. Proses yang sama kemudian diterapkan secara rekursif untuk potongan urutan ke kiri dan ke kanan *subsequence* yang cocok sampai tidak ditemukan lagi *subsequence* yang sama.

a. *Sequence Matching*

Tahap ini mencocokkan dua buah *string* untuk mendapatkan *subsequence* yang dimiliki oleh *String 1* dan *String 2*. Dibawah ini adalah teks dokumen yang telah melewati tahap *preprocessing*.

1) Menghitung Banyak Karakter

Pencarian nilai $|S1|$ dan $|S2|$ dimana $|S1|$ adalah banyak karakter yang dimiliki *String 1* sedangkan $|S2|$ adalah banyak karakter yang dimiliki oleh *String 2*.

String 1

berolahragamembuattubuhsehatdanbugar

$|S1| = 36$

String 2

olahragasecarateraturmembuattubuhsehatdanbugar

$|S2| = 46$

2) Pencarian *Subsequence*

Pencarian *subsequence* yang sama-sama dimiliki oleh *string1* dan *string2*, *subsequence* terpanjang dinamakan *anchor*.

String 1

Berolahraga|membuattubuhsehatdanbugar|

String 2

Olahragasecarateratur|membuattubuhsehatdanbugar|

Anchor :

membuattubuhsehatdanbugar

$Km = 25$

Setelah *Anchor* telah ditemukan, selanjutnya mencari *subsequence* lainnya yang terdapat pada *string1* dan *string2*.

String1 ber|olahraga||membuattubuhsehatdanbugar|

String 2 |olahraga|secarateratur|membuattubuhsehatdanbugar|

Substring baru : olahraga

Menggabung *substring-substring* yang telah ditemukan dari *String 1* dan *String 2*, maka total karakter dari *substring-substring* yang di tambahkan adalah seperti berikut:

$$Km = 25 + |\text{olahraga}| = 25 + 8 = 33$$

b. Menghitung Kesamaan

Tahap ini dilakukan setelah semua *substring* yang sama telah di temukan dari proses *sequence matching* dan dimana persamaan dari algoritma *Ratcliff/Obershelp* akan digunakan. Penilaian *Ratcliff/Obershelp* untuk *String 1* dan *String 2* adalah:

$$D_{ro} \frac{2 \times 33}{36 + 46} = \frac{66}{82} = 0.804$$

Jadi dari kedua *string* diatas memiliki nilai kesamaan 0.804 yang bisa dikatakan sama.

2.6 Text Mining

Text Mining merupakan salah satu bidang khusus dari *data mining*. *Text mining* lebih berfokus pada teknik dan metodologi dalam lingkup temu kembali sistem informasi. *Text mining* berfungsi untuk mendapatkan informasi dari sebuah dokumen teks (Joane dkk., 2017). Dalam penggalian informasi terdapat beberapa tahapan yang diantaranya adalah:

1. *Text preprocessing*

Text preprocessing merupakan tahap awal dari *Text mining* dimana data text akan dibersihkan sehingga text menjadi lebih terstruktur sebelum masuk ketahap berikutnya untuk diolah lebih lanjut (Joane dkk., 2017). Gabungan karakter yang berurutan (teks) wajib di pisah-pisah menjadi karakter yang lebih penting. Tahapan *text preprocessing* dalam penelitian ini meliputi:

1. *Case Folding*

Tahap *case folding* merupakan tahap dimana semua huruf dalam dokumen teks diubah menjadi huruf kecil dan hanya huruf a sampai z saja yang diterima.

2. *Filtering*

Pada tahap *filtering* dimana akan dilakukan penghapusan *stopwords* untuk mengeliminasi kata-kata yang tidak deskriptif dan yang sering muncul.

3. Penghapusan Spasi

Tahap ini merupakan tahap dimana semua spasi yang terdapat pada dokumen teks dihilangkan. Karena algoritma *Ratcliff/Obershelp* merupakan algoritma untuk mendeteksi kemiripan teks dengan cara membandingkan *string* yang berbeda sehingga spasi yang terdapat pada dokumen teks harus dihilangkan sehingga dokumen teks menjadi sebuah *string*.

2. *Pattern Discovery/Text preprocessing*

Pattern Discovery/Text preprocessing merupakan tahap dari terakhir dari *text mining* dimana dokumen teks yang telah dibersihkan dan sudah terstruktur akan diproses untuk mendapatkan pola dan informasi dari dokumen teks tersebut.

2.7 *Stop-Word*

Stop-word adalah istilah untuk kata yang tidak memiliki keterkaitan makna yang kuat terhadap hubungan kata-kata lainnya pada suatu kalimat atau paragraf (Joane dkk., 2017). *Stop-word* yaitu kata yang paling sering muncul dalam suatu kalimat

Seperti yang telah dijelaskan diatas, *stop-word* merupakan kata yang sering muncul dalam suatu kalimat, dalam bahasa Indonesia kata-kata yang sering muncul seperti ‘ada’, ‘dan’, ‘untuk’, ‘di’, ‘adalah’ dan lain sebagainya. Pengeleminasian *stop-word* mampu mengurangi kata dan karakter pada dokumen teks sehingga dapat mengurangi waktu eksekusi dalam pencarian kesamaan.

2.8 *PDF (Portable Document Format)*

PDF (*Portable Document Format*) adalah format *file* standar industri untuk berbagi dokumen. Dikembangkan oleh *Adobe Systems*, PDF adalah salinan

persis dari *file* yang ingin diduplikat, seperti foto, *file* teks, halaman *web*, desain grafis, halaman pengolah kata, tangkapan layar, dan banyak lagi. Mengonversi dokumen ke PDF meninggalkan jejak digital untuk pengeditan dan perubahan di masa mendatang. *File* PDF dapat dilindungi kata sandi dan kemampuan pengeditan atau pencetakannya dibatasi oleh pembuat dokumen (Deff, 2018).

2.9 *Hyper Text Markup Language (HTML)*

HTML merupakan singkatan dari *Hypertext Markup Language*, yaitu bahasa markup standar untuk membuat dan menyusun halaman dan aplikasi *web* (Ayunindya, 2022).

Menurut (Indonesia, 2021) HTML (*HyperText Markup Language*) adalah suatu bahasa yang menggunakan tanda-tanda tertentu (tag) untuk menyatakan kode-kode yang harus ditafsirkan oleh *browser* agar halaman tersebut dapat ditampilkan secara benar. Secara umum, fungsi HTML adalah untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat diakses dan ditampilkan di Internet melalui layanan *web*.

Fungsi HTML yang lebih spesifik yaitu :

1. Membuat halaman *web*.
2. Menampilkan berbagai informasi di dalam sebuah browser Internet.
3. Membuat link menuju halaman *web* lain dengan kode tertentu (*hypertext*).

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)*. HTML pada dasarnya merupakan dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu. HTML dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN dan dipopulerkan pertama kali oleh *browser* Mosaic. Selama awal tahun 1990 HTML mengalami perkembangan yang sangat pesat.

Setiap pengembangan HTML pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya. Kegunaan bahasa ini ialah untuk

memanipulasi *browser* sehingga dapat menampilkan informasi yang dapat dibaca oleh pengguna komputer (Taryana & Kom, 2021).

2.10 PHP (*Hypertext Preprocessor*)

PHP adalah bahasa *server-side scripting* yang bisa menyatu dengan tag-tag HTML. *Server-side scripting* adalah sintaks dan perintah-perintah yang dijalankan pada *server* dan disertakan pada dokumen HTML. Pada suatu halaman *web*, PHP berfungsi menjalankan suatu perintah tertentu, sedangkan HTML berfungsi sebagai struktur desain halaman *web* (Putra, 2018).

2.11 MySQL

MySQL atau dibaca *My Sequel* merupakan sebuah *Database Management System* atau sering disingkat DBMS yang dijalankan menggunakan perintah *SQL* (*Structured Query Language*) yang populer digunakan untuk pembuatan aplikasi berbasis *website*. Selain itu *SQL* dipuji karena kesederhanaan sintaks yang pendek dan mudah dipahami. Sehingga *SQL* cocok dipilih sebagai bahasa terbaik untuk memulai untuk belajar *data science* bagi pemula sebelum belajar bahasa pemrograman. MySQL juga termasuk ke dalam RDBMS atau *Relational Database Management System*, dimana di dalam struktur database-nya sehingga ketika proses pengambilan data menggunakan metode *relational database*. Yang juga menjadi penghubung antara perangkat lunak dan *database server* (Tineges, 2021).

2.12 Database

Database atau dalam bahasa Indonesia berarti Basis Data atau pangkalan data merupakan kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa/dipantau menggunakan suatu program komputer. Secara konsep, *Database* atau Basis Data yaitu kumpulan dari data-data yang membentuk suatu berkas atau *file* yang saling berhubungan dengan tatacara yang tertentu untuk membentuk data baru atau informasi. Atau bisa diartikan *Database* atau Basis Data merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lainnya yang diorganisasikan berdasarkan

skema atau struktur tertentu. *Database* atau Basis Data juga dapat diartikan sebagai koleksi terorganisir dari data. Data tersebut biasanya diselenggarakan untuk dapat model aspek realitas dengan cara yang mendukung proses yang sedang membutuhkan (Syafnidawaty, 2020).

2.13 XAMPP

XAMPP adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata *Apache*, *MySQL*/MariaDB, *PHP*, dan *Perl*. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. Program aplikasi XAMPP berfungsi sebagai *server* lokal untuk mengampu berbagai jenis data *website* yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam *website* kepada orang lain tanpa harus terkoneksi dengan internet, cukup akses melalui *Xampp control panel*, atau istilahnya *website offline*. XAMPP bekerja secara *offline* layaknya *web hosting* biasa namun tidak bisa diakses oleh banyak orang (Nugroho, 2019).

XAMPP adalah aplikasi *tools* untuk menyediakan paket lunak yang berisi konfigurasi *Web Server*, *Apache*, *PHP*, *MySQL* untuk membantu kita dalam proses pembuatan aplikasi *web* yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program *web* (Josi, 2017).

2.14 Visual Studio Code

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang

via *marketplace Visual Studio Code* (seperti C++, C#, Python, Go, Java, dst) (W, 2019).

Visual Studio Code merupakan sebuah aplikasi *editor code open source* yang dikembangkan oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOS. *Visual Code* memudahkan dalam penulisan *code* yang mendukung beberapa jenis pemrograman, seperti C++, C#, Java, Python, PHP, GO. *Visual Code* memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian *code* tersebut. *Visual Studio Code* juga telah terintegrasi ke Github. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di *Visual Studio Code* (Ramdhan, 2019).

2.15 Unified Modeling Language (UML)

2.15.1 Definisi Unified Modeling Language (UML)

Unified Modeling Language (UML) muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. *Unified Modeling Language* (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Putra, 2018).

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software* (Mubarak, 2019).

2.15.2 Diagram Unified Modeling Language (UML)








1. Use case Diagram

Use case diagram merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. *Use case diagram* bisa mendeskripsikan sebuah



interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case diagram* juga bisa digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan bisa juga mempresentasikan sebuah interaksi aktor dengan sistem (Hutauruk, 2019).

Use case diagram memodelkan perilaku sistem dan membantu menangkap persyaratan sistem. *Use case diagram* menggambarkan fungsi tingkat tinggi dan ruang lingkup suatu sistem. Diagram ini juga mengidentifikasi interaksi antara sistem dan aktornya (IBM, 2021).

Tabel 2. 1 Use Case Diagram (Julianto & Setiawan, 2019)

No	Gambar	Nama	Keterangan
1.		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2.		<i>Aktor/Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
3.		<i>Asosiasi /Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		<i>Include</i>	<i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.
5.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
6.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum–khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
7.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>dependent</i>) akan mempengaruhi elemen yang bergantung padanya elemen.







Tabel 2. 1 Use Case Diagram (Julianto & Setiawan, 2019) Lanjutan

No	Gambar	Nama	Keterangan
8.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemenya.

2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan aktor (Julianto & Setiawan, 2019).


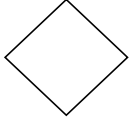


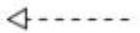
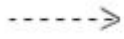
Tabel 2. 2 Activity Diagram (Julianto & Setiawan, 2019)

No	Gambar	Nama	Keterangan
1.		Aktivitas/ <i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
2.		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
3.		Percabangan/ <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Status Awal/ <i>Initial Node</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
5.		Status Akhir/ <i>Activity Final Node</i>	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.		Penggabungan/ <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

3. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Julianto & Setiawan, 2019).

Tabel 2. 3 Class Diagram (Julianto & Setiawan, 2019)

No	Gambar	Nama	Keterangan
1.		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Kelas pada struktur sistem.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

2.16 Rational Unified Process (RUP)

Metode RUP merupakan metode pembangunan perangkat lunak yang *iterative* dan *incremental* serta berfokus pada arsitektur. Metode RUP dapat menangani risiko yang berhubungan dengan pengembangan kebutuhan sistem berdasarkan perubahan yang diinginkan oleh klien. Untuk mengurangi risiko tersebut dilakukan dengan pengujian pada setiap akhir tahapan RUP, sehingga akan mudah melakukan perubahan sebelum mencapai tahap akhir. Metode RUP

juga mengutamakan kepuasan pengguna sehingga lebih sering melakukan interaksi dengan pengguna (Endang Anjarwani, 2020).

Dalam metode ini, terdapat empat tahap pengembangan perangkat lunak yaitu:

1. *Inception*

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan *user*, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*).

2. *Elaboration*

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilsan *prototype* versi *Betha* dari perangkat lunak.

3. *Construction*

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.

4. *Transition*

Instalasi *deployment* dan sosialisasi perangkat lunak dilakukan pada tahap ini. Pada tahapan ini dilakukan tahap evaluasi dari sistem yang telah direncanakan, dianalisa dan dibangun.

2.17 Metode Pengujian Sistem

2.17.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses atau rangkaian proses yang dirancang untuk memastikan bahwa program komputer menjalankan apa yang seharusnya dilakukan dan sebaliknya, memastikan program agar tidak melakukan hal yang tidak diharapkan. Sebuah perangkat lunak seharusnya dapat diprediksi dan konsisten. Tugas utama dari seorang penguji adalah untuk menemukan *bug* atau *error* sebanyak mungkin serta mengetahui bagaimana *error* atau bug itu dihasilkan. Sehingga dapat dikatakan bahwa pengujian adalah proses

mengeksekusi program dengan tujuan untuk menemukan *error*. Pengujian merupakan proses penting dalam siklus pengembangan *software* untuk memastikan kualitas dari sebuah *software* (Meiliana, 2018).

2.17.2 *Black Box Testing*

Blackbox-testing merupakan salah satu metode untuk menguji perangkat lunak yang telah dibangun, baik pengujian pada unit-unit kecil maupun hasil yang telah terintegrasi untuk menguji fungsional perangkat lunak. Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Syarif dkk., 2021).

2.17.3 *Confusion Matrix*

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Prasetyo, 2012).

Untuk melakukan pengujian sistem, maka dilakukan pencocokan antara hasil dari pengecekan kesamaan dokumen skripsi yang dilakukan oleh manusia (ahli) terhadap hasil yang dikeluarkan. Perhitungan nilai akurasi dapat dilihat pada persamaan (2.2), persamaan (2.3) merupakan persamaan untuk menghitung nilai *precision*, *precision* adalah perbandingan antara *True Positive* (TP) dengan banyaknya data yang diprediksi positif. *Recall* dihitung dengan menggunakan persamaan (2.4), *Recall* adalah perbandingan antara *True Positive* (TP) dengan banyaknya data yang sebenarnya positif. Persamaan (2.5) merupakan persamaan untuk menghitung nilai *f-measures*. *f-measures* merupakan salah satu perhitungan akurasi dalam *information retrieval* yang menggabungkan *recall* dan *precision*, nilai keakuratan pada *recall* dan *precision* dapat memiliki nilai bobot yang berbeda. Ukuran hubungan timbal balik antara *recall* dan *precision* adalah *f-measures*, yang merupakan bobot rata-rata yang diselaraskan, nilai aktual, dan presisi.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.2)$$

$$Presisi = \frac{TP}{TP + FP} \times 100\% \quad (2.3)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.4)$$

$$F - Measure = 2 \times \frac{Presisi \times recall}{Presisi + recall} \times 100\% \quad (2.5)$$

Dimana :

TP = True Positif, dalam artian hasil deteksi menyatakan plagiat, dan hasil aktual juga menyatakan plagiat

FP = False Positif, dalam artian hasil deteksi menyatakan plagiat, dan hasil aktual menyatakan tidak plagiat

FN = False Negatif, dalam artian hasil deteksi menyatakan tidak plagiat, namun hasil aktual menyatakan plagiat

TN = True Negatif, dalam artian hasil deteksi menyatakan tidak plagiat, dan hasil aktual juga menyatakan tidak plagiat

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Maret 2022 sampai dengan Juni 2022. Rincian kegiatan penelitian dapat dilihat pada Tabel 3.1 berikut:

Tabel 3. 1 *Gannt Chart* Waktu Penelitian

No	Uraian	Waktu															
		Maret				April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	<i>Inception</i>																
2.	<i>Elaboration</i>																
3.	<i>Construction</i>																
4.	<i>Transition</i>																

3.1.2 Tempat Penelitian

Adapun tempat penelitian tugas akhir akan dilakukan di Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo.

3.2 Metode Pengumpulan Data

Dalam penelitian ini metode pengumpulan data yang digunakan dalam perancangan sistem dengan melakukan Studi Literatur. Studi literatur dilakukan dengan cara mengumpulkan referensi yang berkaitan dengan objek penelitian yang dilakukan dengan mencari dan mempelajari buku-buku, jurnal maupun *e-book*, penelusuran internet yang berkaitan dengan algoritma *Ratcliff/Obershelp* sebagai penunjang yang dapat dijadikan sebagai landasan penelitian. Sehingga algoritma *Ratcliff/Obershelp* dapat diimplementasikan dalam sistem Pengecekan Kemiripan Skripsi Mahasiswa Teknik Informatika di Universitas Halu Oleo.

3.3 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *Rational Unified Process* (RUP). Dalam metode pengembangan sistem ini, terdiri dari 4 tahapan pengembangan perangkat lunak, yaitu:

3.3.1 Permulaan (*Inception*)

Pada tahap ini, memodelkan ruang lingkup proyek yang akan dibuat. Adapun ruang lingkup dari proyek ini adalah sebagai berikut:

1. Pengecekan kesamaan skripsi mahasiswa dengan data skripsi mahasiswa Teknik Informatika, Universitas Halu Oleo yang tersimpan pada sistem.
2. Melakukan proses analisis kebutuhan sistem yang akan dibuat serta melakukan kajian terhadap penelitian yang terkait dengan Algoritma *Ratcliff/Obershelp*.

3.3.2 Perluasaan/Perencanaan (*Elaboration*)

Pada tahap ini akan dilakukan analisis sistem dan perancangan sistem menggunakan UML (*Unified Modelling Language*) yang meliputi *use case diagram*, *activity diagram* dan *class diagram*.

3.3.3 Konstruksi (*Construction*)

Pada tahap ini proses yang dilakukan yaitu membangun sistem dengan perancangan yang telah dilakukan sebelumnya, mulai dari tampilan *user interface* sampai pada implementasi rancangan UML. Proses yang juga dilakukan pada tahap ini yaitu penerapan coding Algoritma *Ratcliff/Obershelp* pada sistem.

3.3.4 Transisi (*Transition*)

Pada tahap ini difokuskan untuk melakukan proses pengujian terhadap sistem dan memperbaiki masalah-masalah yang muncul pada saat dan setelah pengujian.

3.4 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem sangat diperlukan dalam mendukung kinerja sebuah sistem atau aplikasi, apakah sistem atau aplikasi yang dibuat telah sesuai dengan kebutuhan atau belum. Karena kebutuhan sistem akan mendukung tercapainya tujuan suatu aplikasi.

Analisis kebutuhan sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem dimana aplikasi dibangun, meliputi perangkat lunak, dan hasil analisis terhadap sistem serta elemen-elemen sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional.

3.4.1 Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun. Setelah melalui tahapan analisis, maka telah ditetapkan kebutuhan-kebutuhan untuk membangun sistem meliputi *input*, proses, dan *output*.

3.4.1.1 Analisis Kebutuhan *Input*

Input dari sistem yang akan dibangun berupa data-data skripsi mahasiswa Teknik Informatika, Universitas Halu Oleo.

3.4.1.2 Analisis Kebutuhan Proses

Analisis kebutuhan proses dalam sistem atau aplikasi ini adalah proses pembuatan sistem pengecekan skripsi mahasiswa Jurusan Teknik Informatika, Universitas Halu Oleo dengan menggunakan bahasa pemrograman PHP disisi *web service* dengan menggunakan Algoritma *Ratcliff/Obershelp*.

3.4.1.3 Analisis Kebutuhan *Output*

Output yang diperoleh dalam sistem ini adalah dapat mengetahui berapa persen kesamaan dalam pengecekan dokumen skripsi pada sistem Pengecekan Kesamaan Skripsi yang akan dibangun.

3.4.2 Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional merupakan sebuah langkah dimana pembangunan sebuah aplikasi melakukan analisis sumber daya kebutuhan yang digunakan untuk membangun aplikasi. Analisis kebutuhan non-fungsional dilakukan dengan dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak. kebutuhan perangkat keras yaitu kebutuhan perangkat atau komponen-komponen yang akan dibutuhkan pada system dan perangkat

lunak yaitu kebutuhan perangkat lunak terkait agar komponen perangkat keras dapat berfungsi dan dapat dijalankan pada sistem.

3.4.2.1 Kebutuhan Perangkat Keras

Untuk menerapkan rancangan yang telah di jelaskan sebelumnya, dibutuhkan beberapa perangkat keras sebagai sarana untuk mengimplementasikan aplikasi yang di bangun. Adapun beberapa spesifikasi perangkat keras yang di butuhkan adalah sebagai berikut:

Tabel 3. 2 Spesifikasi Perangkat Keras

No.	Nama Perangkat	Spesifikasi
1.	<i>PC</i>	<i>Acer Aspire A514-52G</i>
2.	<i>Processor</i>	<i>Intel Core i5</i>
3.	<i>RAM</i>	<i>4 GB</i>
4.	<i>Harddisk</i>	<i>128 GB SSD</i>
5.	Monitor	Monitor 14 Inch

3.4.2.2 Kebutuhan Perangkat Lunak

Untuk menerapkan rancangan yang telah di jelaskan sebelumnya,di butuhkan beberapa perangkat lunak sebagai sarana untuk mengimplementasikan aplikasi yang di bangun. Adapun beberapa spesifikasi perangkat lunak yang di butuhkan adalah sebagai berikut:

Tabel 3. 3 Spesifikasi Perangkat Lunak

No.	Nama Perangkat	Fungsi	Spesifikasi
1.	<i>Windows</i>	<i>Operating System</i>	<i>Windows 10 Home</i>
2.	<i>Xampp</i>	<i>Universal development environment</i>	<i>Xampp v3.2.2</i>
3.	<i>Apache</i>	<i>Web Server</i>	<i>Apache 2.4 win64-VC14</i>
4.	<i>Mysql</i>	<i>Database Management</i>	<i>Mysql 5.7</i>
5.	<i>PHP</i>	<i>Web development</i>	<i>PHP 7.3</i>
6.	<i>Chrome</i>	<i>Web browser</i>	<i>Chrome version 76.0</i>
7.	<i>Visual Studio Code</i>	<i>Code Editor</i>	<i>VS Code v1.38</i>

3.5 Analisis Perancangan Sistem

Perancangan sistem yang akan dibangun yaitu perancangan UML dan perancangan *user interface* sistem.

3.5.1 *Unified Modeling Language* (UML)

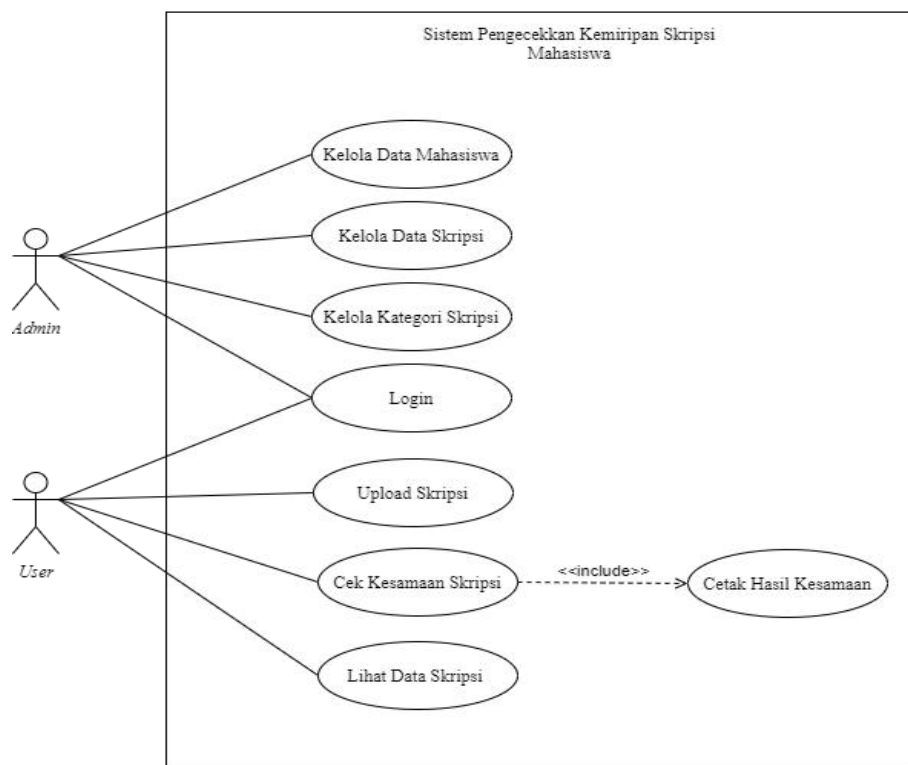
Sistem dalam penelitian ini dibangun menggunakan *Unified Modeling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.

1. *Use Case Diagram*

Use Case menggambarkan *external view* dari sistem yang akan dibuat modelnya, komponen pembentuk diagram *use case* adalah:

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use case*, aktivitas yang disiapkan oleh bisnis/sistem.
- c. Hubungan (*link*), aktor mana saja yang terlibat dalam use case ini.

Use case diagram akan memperlihatkan bagaimana peranan setiap aktor dalam interaksi dengan sistem. *Use case diagram* untuk sistem yang akan dikembangkan dapat dilihat seperti pada Gambar 1 di bawah ini:



Gambar 3. 1 Use Case Diagram Sistem Pengecekan Skripsi Mahasiswa

Tabel 3. 4 Deskripsi Case Login (Admin, User)

Nama	Login
Deskripsi Singkat:	Proses keamanan login untuk <i>Admin</i> dan <i>user</i>
Persyaratan:	<i>Admin</i> dan <i>user</i> harus memasukkan <i>username</i> dan <i>password</i>
Kondisi Akhir	Masuk untuk <i>manage</i> sistem
Situasi Kesalahan	i. Yang masuk bukan <i>Admin</i> dan <i>user</i> ii. Salah memasukkan <i>username</i> dan <i>password</i>
Status sistem dalam saat terjadi kesalahan	1. Menampilkan pesan <i>error</i> 2. Tidak dapat masuk ke dalam sistem
Aktor	<i>Admin</i> dan <i>user</i>
Trigger	Keamanan login untuk <i>Admin</i>

Tabel 3. 4 Deskripsi Case Login (Admin, User) Lanjutan

Nama	Login
Proses Standar	<ol style="list-style-type: none"> 1. <i>Admin</i> dan <i>user</i> masuk ke sistem 2. <i>Admin</i> dan <i>user</i> memasukkan <i>username</i> 3. <i>Admin</i> dan <i>user</i> memasukkan <i>password</i>
Proses Alternatif	Mengecek <i>username</i> dan <i>password</i>

Tabel 3. 5 Deskripsi Case Data Mahasiswa Admin

Nama	Data Mahasiswa
Deskripsi Singkat:	<i>Admin</i> ingin melihat, menambah, menghapus atau melihat data mahasiswa
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data mahasiswa di tampilkan
Situasi Kesalahan	-
Aktor	<i>Admin</i>
<i>Trigger</i>	-
Proses Standar	<ol style="list-style-type: none"> 1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>login</i> 2. <i>Admin</i> melihat data mahasiswa 3. <i>Admin</i> menambahkan data mahasiswa 4. <i>Admin</i> menghapus data mahasiswa 5. <i>Admin</i> mengedit data mahasiswa

Tabel 3. 6 Deskripsi Case Data Skripsi Admin

Nama	Data Skripsi
Deskripsi Singkat:	<i>Admin</i> ingin melihat, menambah, mengedit, menghapus atau melihat data skripsi
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data skripsi di tampilkan

Tabel 3. 6 Deskripsi Case Data Skripsi *Admin* Lanjutan

Nama	Data Skripsi
Situasi Kesalahan	-
Aktor	<i>Admin</i>
Proses Standar	<ol style="list-style-type: none"> 1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>login</i> 2. <i>Admin</i> melihat data skripsi 3. <i>Admin</i> menambahkan data skripsi 4. <i>Admin</i> menghapus data skripsi 5. <i>Admin</i> mengedit data skripsi

Tabel 3. 7 Deskripsi Case Kategori Skripsi *Admin*

Nama	Kategori Skripsi
Deskripsi Singkat	<i>Admin</i> ingin melihat, menambah, menghapus atau melihat kategori skripsi
Persyaratan	<i>Admin</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data kategori skripsi ditampilkan
Situasi kesalahan	-
Aktor	<i>Admin</i>
Proses Standar	<ol style="list-style-type: none"> 1. <i>Admin</i> masuk ke sistem dan telah melakukan proses <i>login</i> 2. <i>Admin</i> melihat kategori skripsi 3. <i>Admin</i> menambahkan kategori skripsi 4. <i>Admin</i> menghapus kategori skripsi 5. <i>Admin</i> mengedit kategori skripsi

Tabel 3. 8 Deskripsi Case Upload Skripsi *User*

Nama	Upload Skripsi
Deskripsi Singkat:	<i>User</i> ingin mengupload dokumen skripsi
Persyaratan	<i>User</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Dokumen skripsi berhasil dimasukkan
Situasi Kesalahan	Dokumen yang dimasukkan bukan dokumen dalam format pdf

Tabel 3. 8 Deskripsi Case Upload Skripsi User Lanjutan

Nama	Upload Skripsi
Status sistem dalam saat terjadi kesalahan	Mengupload kembali dokumen skripsi dengan format pdf
Aktor	<i>User</i>
Proses Standar	<ol style="list-style-type: none"> 1. <i>User</i> masuk ke sistem dan telah melakukan proses <i>login</i> 2. <i>User</i> mengupload dokumen skripsi

Tabel 3. 9 Deskripsi Case Cek Kesamaan Skripsi

Nama	Cek Kesamaan Skripsi
Deskripsi singkat	<i>User</i> ingin mengecek kesamaan skripsi
Persyaratan	<i>User</i> harus melakukan <i>login</i> terlebih dahulu dan melakukan <i>upload</i> dokumen skripsi
Kondisi Akhir	Hasil kesamaan skripsi ditampilkan
Aktor	<i>User</i>
Proses Standar	<ol style="list-style-type: none"> 1. <i>User</i> telah melakukan proses <i>login</i> dan melakukan <i>upload</i> dokumen skripsi 2. <i>User</i> mengecek kesamaan dan 3. Sistem menampilkan hasil kesamaan skripsi yang telah dimasukkan dengan skripsi yang tersimpan dalam sistem

Tabel 3. 10 Deskripsi Case Lihat Data Skripsi

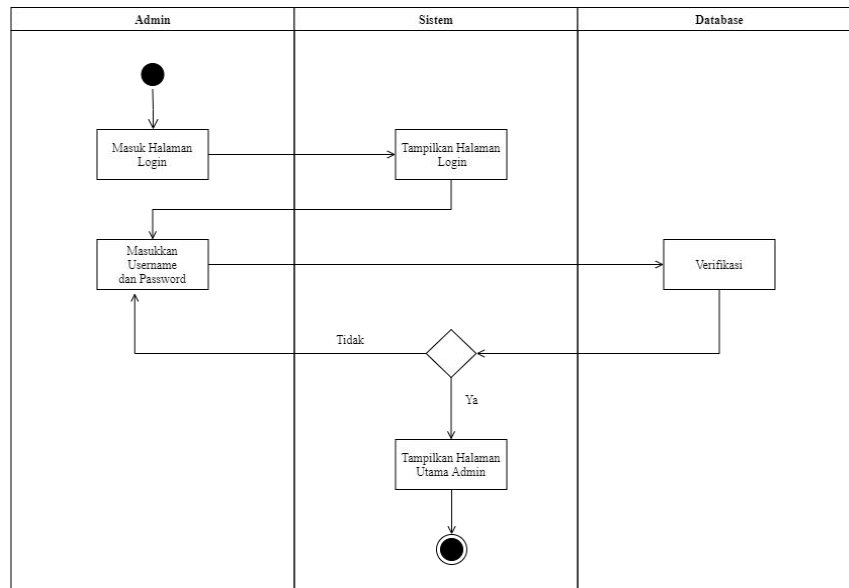
Nama	Lihat Data Skripsi
Deskripsi Singkat:	<i>User</i> ingin melihat data skripsi
Persyaratan	<i>User</i> harus melakukan <i>login</i> terlebih dahulu
Kondisi Akhir	Data skripsi ditampilkan
Aktor	<i>User</i>
Trigger	<i>User</i> ingin menambahkan dokumen skripsi
Proses Standar	<ol style="list-style-type: none"> 1. <i>User</i> masuk ke sistem dan telah melakukan proses <i>login</i> 2. <i>User</i> melihat data skripsi

2. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas pada sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana hal tersebut berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah *activity diagram* yang akan menggambarkan alir aktivitas sistem.

a. Activity Diagram Login

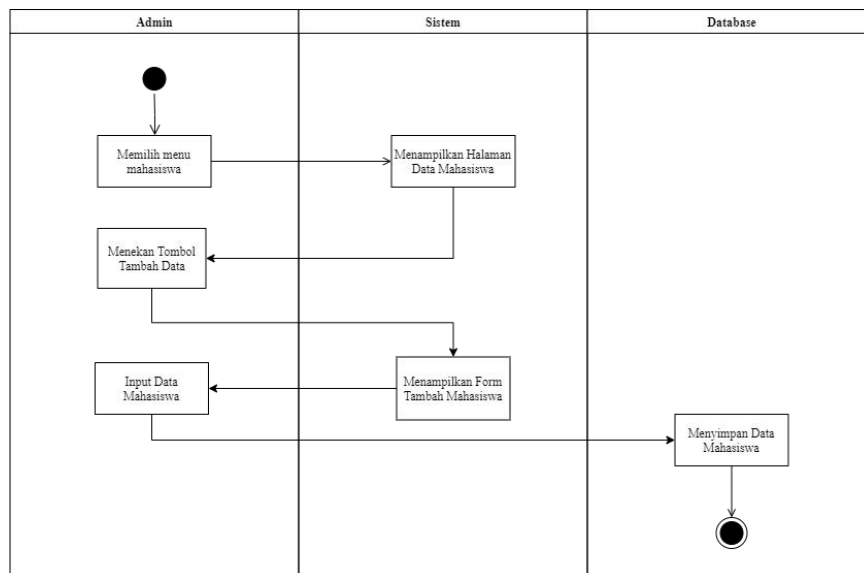
Pada gambar 3.2 merupakan diagram aktivitas yang menunjukkan aktivitas *user* ketika melakukan proses *login* dengan memasukkan *username* dan *password*, lalu sistem akan melakukan validasi pada *database*.



Gambar 3. 2 Activity Diagram Login

b. Activity Diagram Tambah Data Mahasiswa

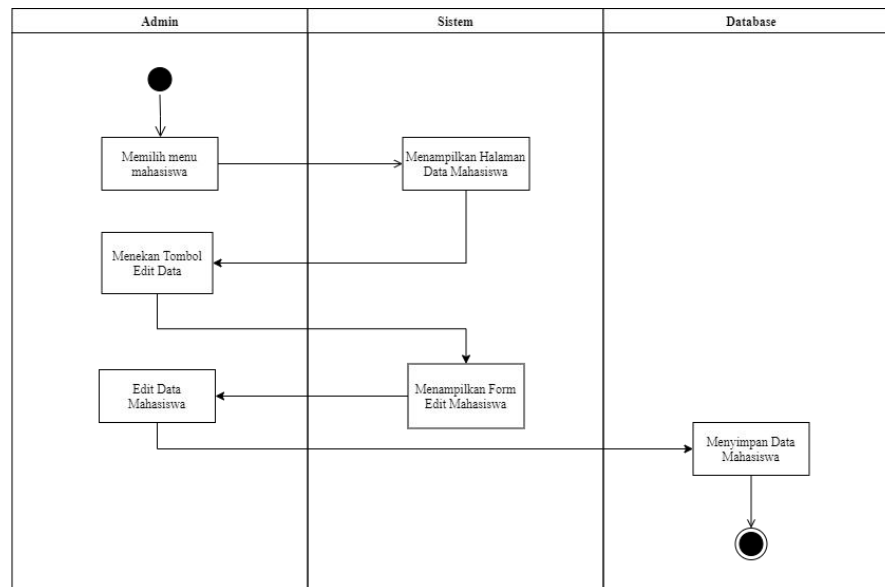
Pada gambar 3.3 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan halaman data mahasiswa.



Gambar 3. 3 Activity Diagram Tambah Data Mahasiswa

c. Activity Diagram Edit Data Mahasiswa

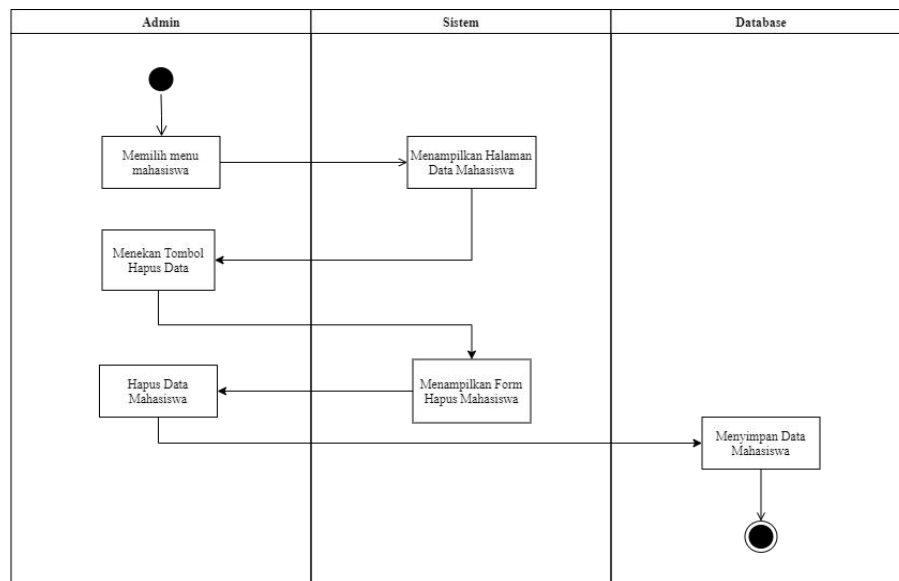
Pada gambar 3.4 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan halaman data mahasiswa.



Gambar 3. 4 Activity Diagram Edit Data Mahasiswa

d. Activity Diagram Hapus Data Mahasiswa

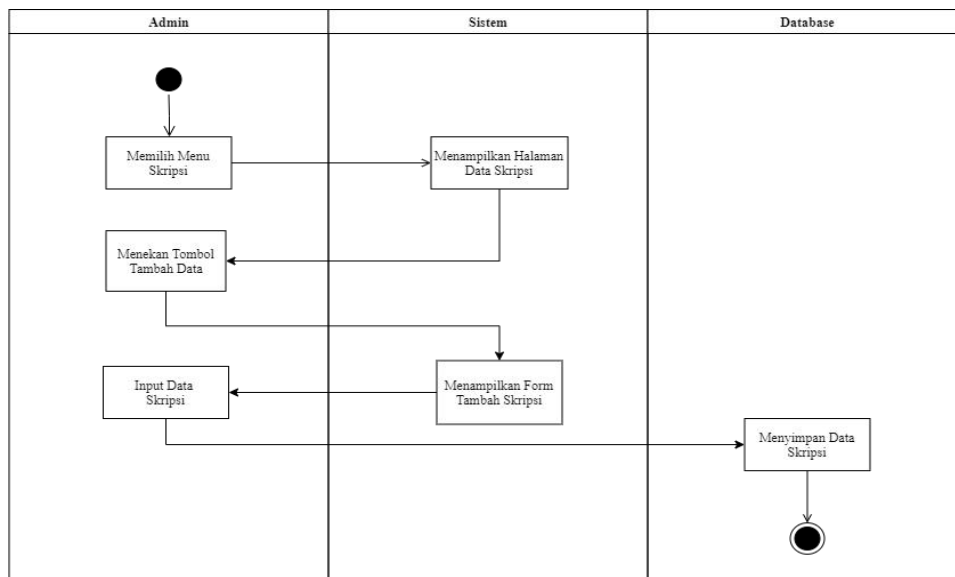
Pada gambar 3.5 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu mahasiswa lalu sistem menampilkan halaman data mahasiswa.



Gambar 3. 5 Activity Diagram Hapus Data Mahasiswa

e. Activity Diagram Tambah Data Skripsi

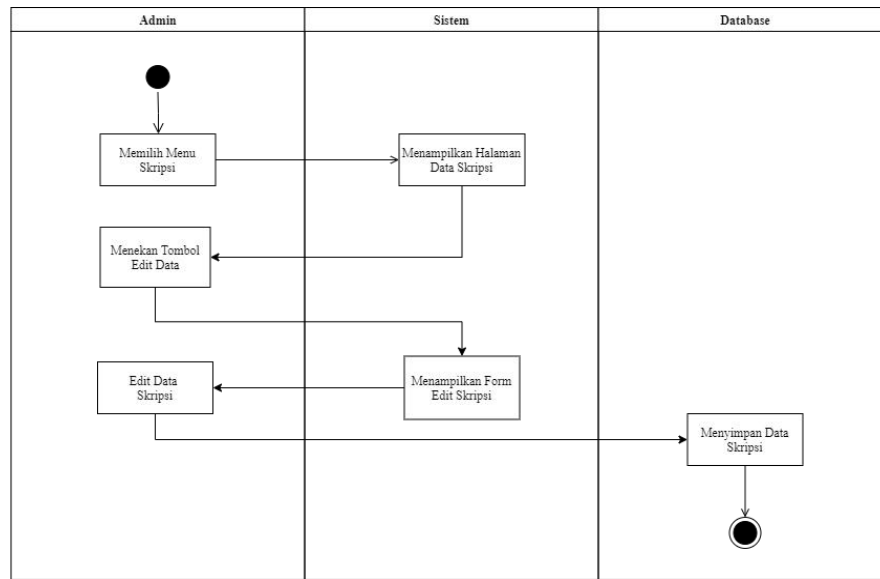
Pada gambar 3.6 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu skripsi lalu sistem menampilkan halaman data skripsi.



Gambar 3. 6 Activity Diagram Tambah Data Skripsi

f. Activity Diagram Edit Data Skripsi

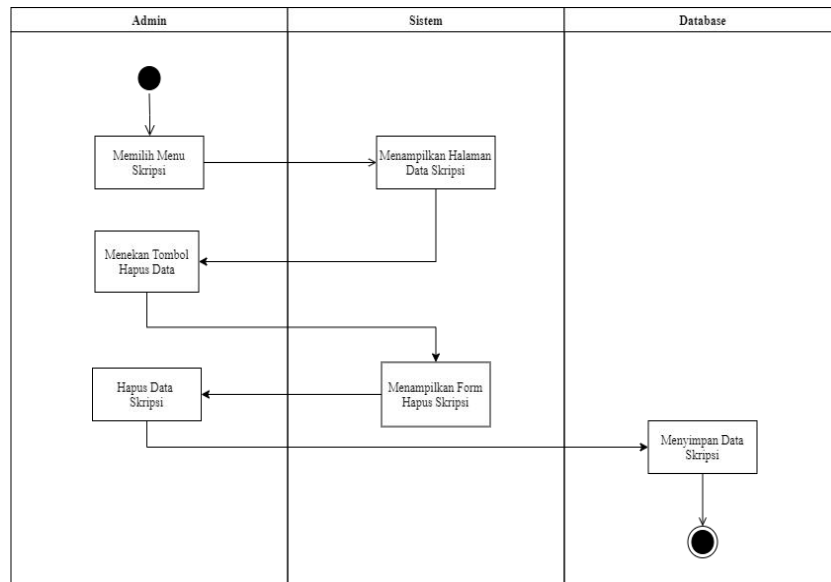
Pada gambar 3.7 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu skripsi lalu sistem menampilkan halaman data skripsi.



Gambar 3. 7 Activity Diagram Edit Data Skripsi

g. Activity Diagram Hapus Data Skripsi

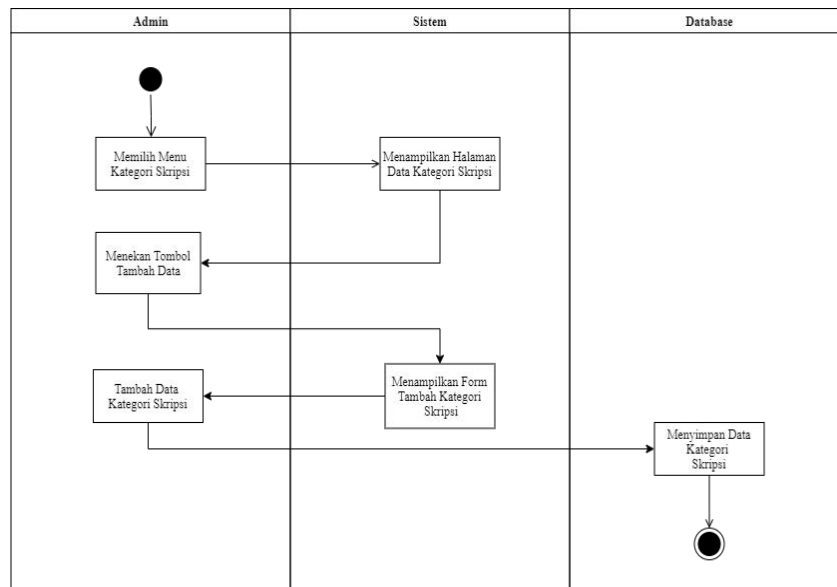
Pada gambar 3.8 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu skripsi lalu sistem menampilkan halaman data skripsi.



Gambar 3. 8 Activity Diagram Hapus Data Skripsi

h. *Activity Diagram* Tambah Data Kategori Skripsi

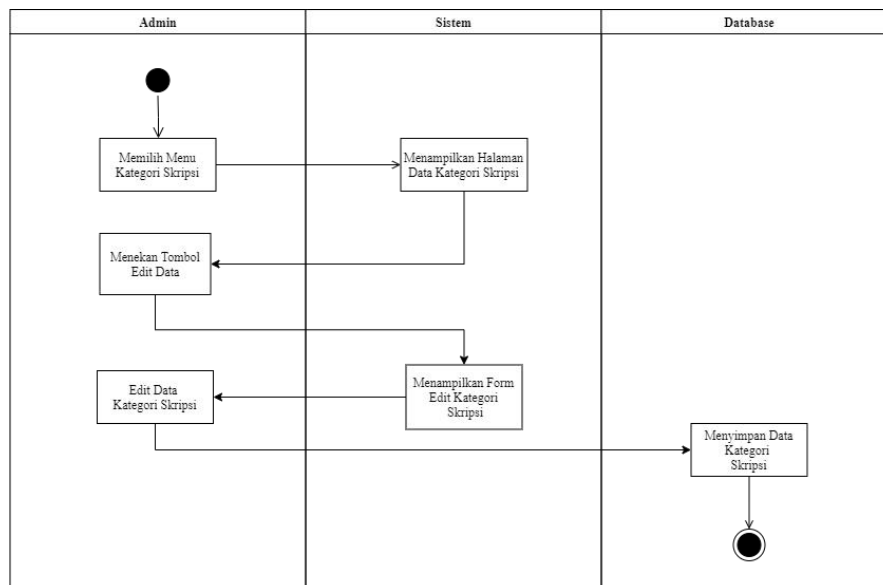
Pada gambar 3.9 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu kategori skripsi lalu sistem menampilkan halaman data kategori skripsi.



Gambar 3. 9 Tambah Data Kategori Skripsi

i. *Activity Diagram* Edit Data Skripsi

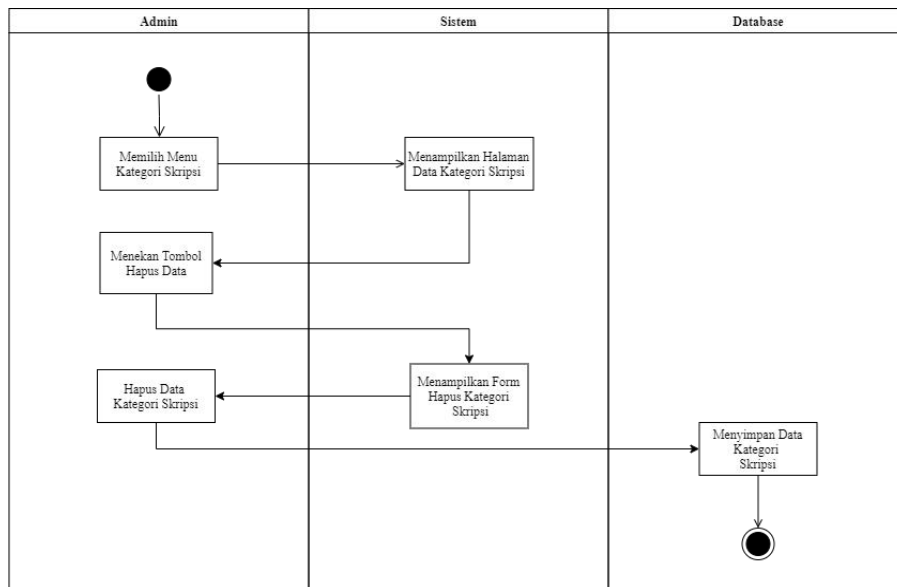
Pada gambar 3.10 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu kategori skripsi lalu sistem menampilkan halaman data kategori skripsi.



Gambar 3. 10 Activity Diagram Edit Data Kategori Skripsi

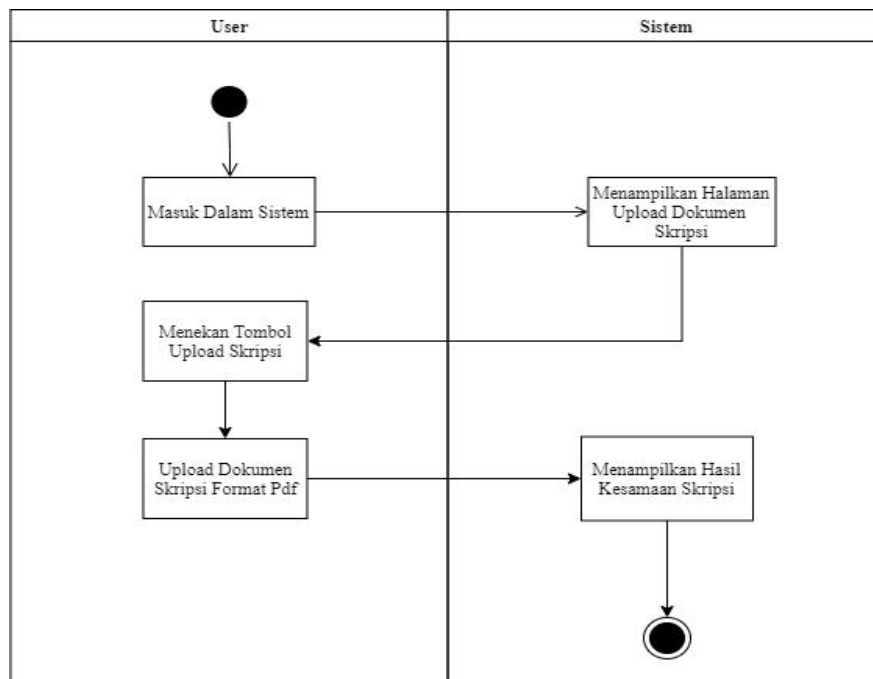
j. Activity Diagram

Pada gambar 3.11 merupakan diagram aktivitas yang menunjukkan aktivitas *admin* memilih menu kategori skripsi lalu sistem menampilkan halaman data kategori skripsi.



Gambar 3. 11 Activity Diagram Hapus Data Kategori Skripsi

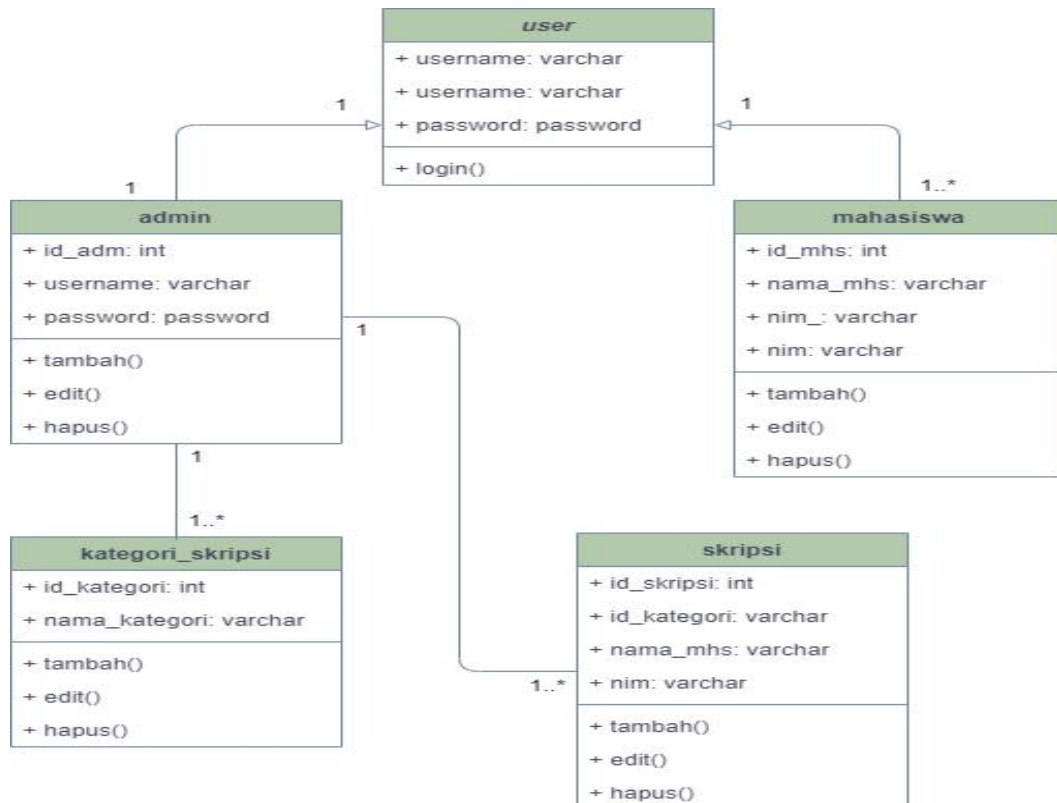
k. Activity Diagram Cek Kesamaan Skripsi



Gambar 3. 12 Activity Diagram Cek Kesamaan Skripsi

3. Class Diagram

Class diagram merupakan diagram untuk menjelaskan pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berinteraksi untuk mencapai suatu tujuan. Berikut ini adalah *class diagram system*.



Gambar 3. 13 Class Diagram

3.5.2 Perancangan Antarmuka (Interface)

Rancangan Antarmuka atau Desain *User Interface* (UI) adalah proses yang digunakan desainer untuk membuat tampilan dalam perangkat lunak atau perangkat terkomputerisasi, dengan fokus pada tampilan atau gaya. Tujuan dari desainer UI adalah untuk membuat desain antarmuka yang membuat pengguna mudah untuk digunakan dan menyenangkan (Binus University, 2020).

1. Halaman Beranda Utama

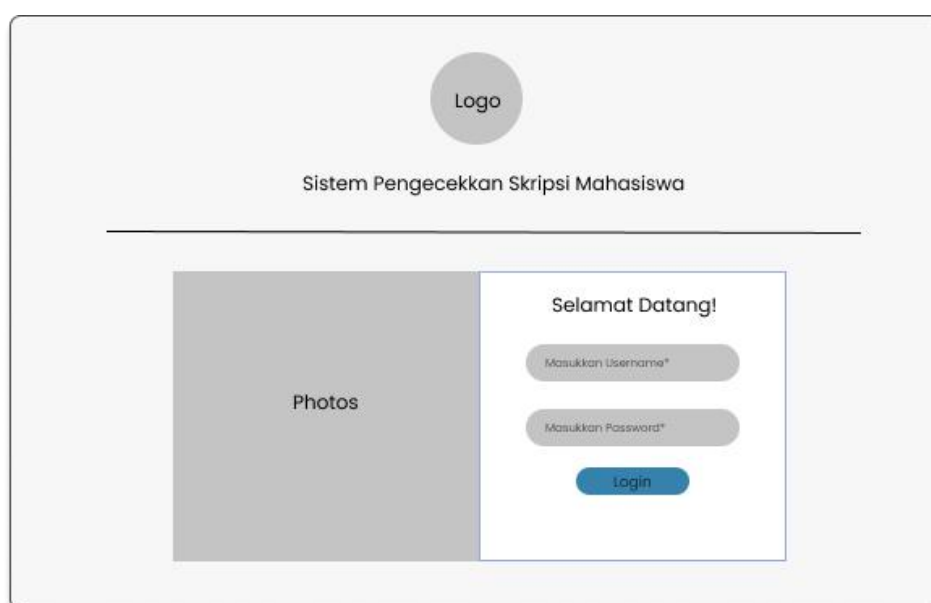
Halaman beranda merupakan halaman pertama yang dijumpai ketika masuk ke dalam sistem pengecekan kesamaan skripsi mahasiswa.



Gambar 3. 14 Tampilan Beranda Utama

2. Halaman *Login*

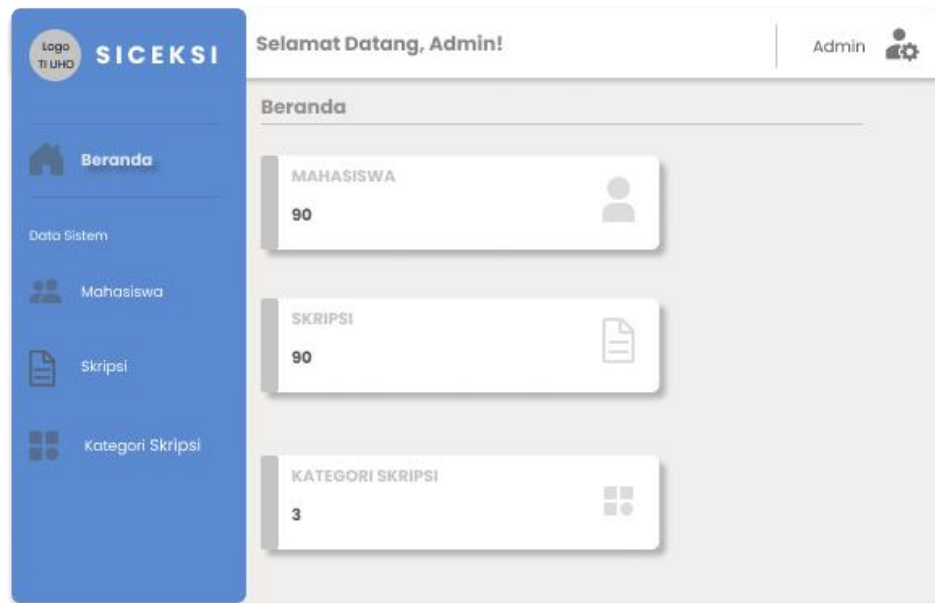
Halaman login merupakan halaman bagi *user* dan *admin* untuk melakukan proses masuk pada sistem.



Gambar 3. 15 Tampilan Halaman *Login*

3. Halaman Beranda *Admin*

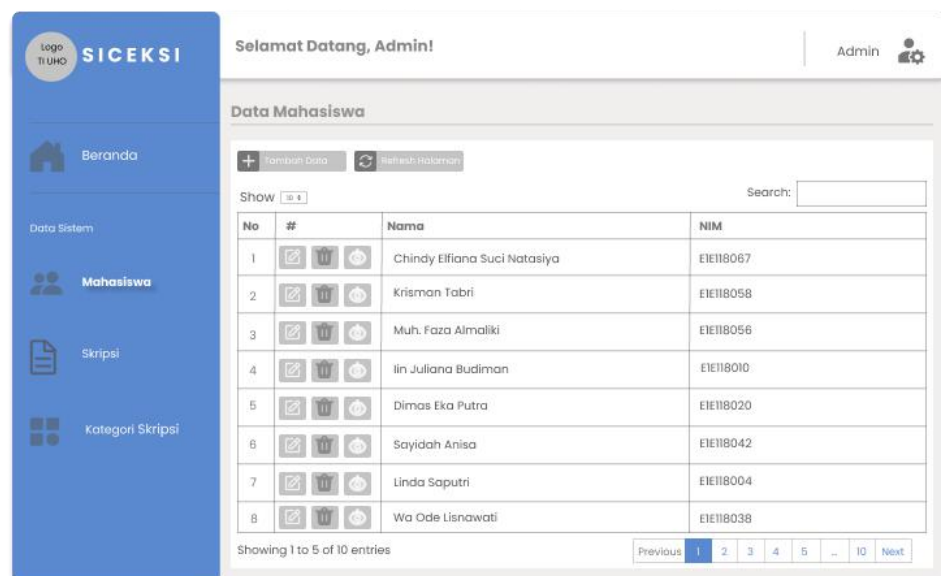
Halaman beranda merupakan halaman pertama yang dijumpai ketika berhasil *login* kedalam sistem pengecekan kesamaan skripsi mahasiswa.



Gambar 3. 16 Tampilan Halaman Beranda *Admin*

4. Halaman Data Mahasiswa *Admin*

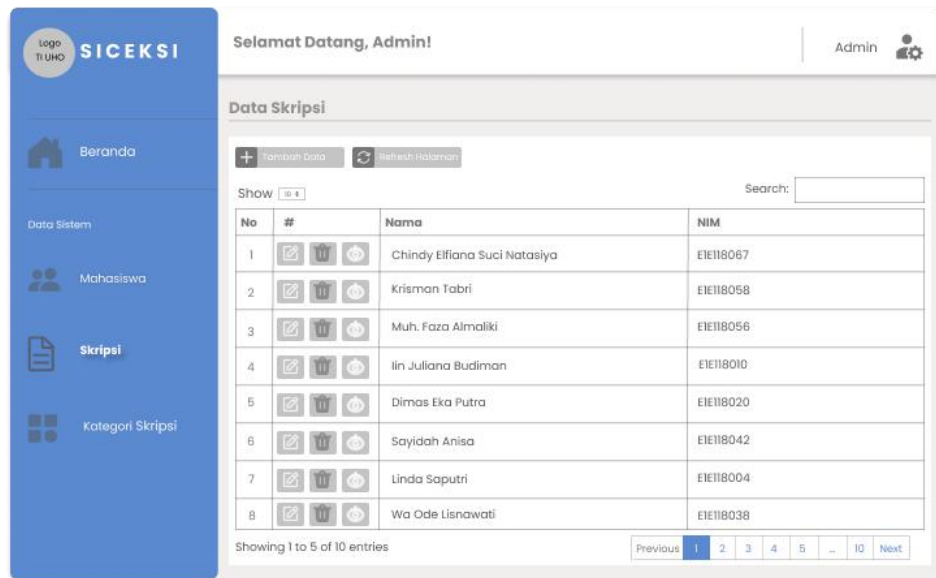
Halaman yang menampilkan data mahasiswa dimana *admin* dapat mengelola data mahasiswa.



Gambar 3. 17 Tampilan Halaman Data Mahasiswa *Admin*

5. Halaman Data Skripsi *Admin*

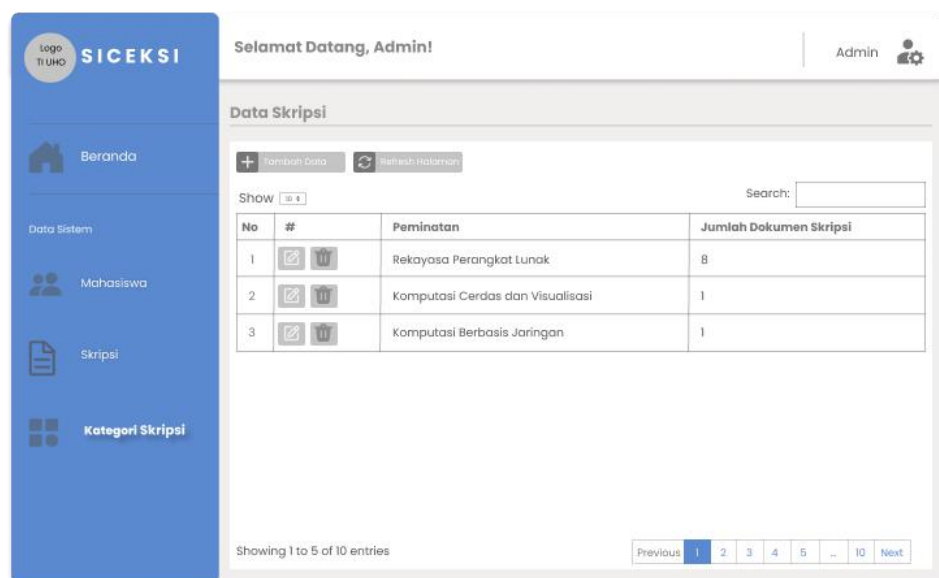
Halaman yang menampilkan data skripsi dimana *admin* dapat mengelola data skripsi.



Gambar 3. 18 Tampilan Halaman Data Skripsi *Admin*

6. Halaman Data Kategori Skripsi *Admin*

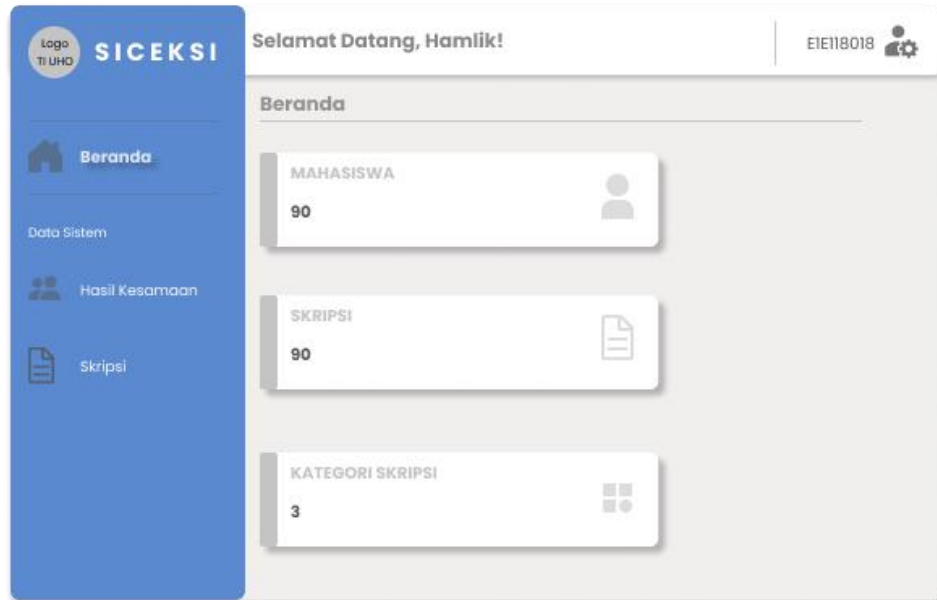
Halaman yang menampilkan data skripsi dimana *admin* dapat mengelola data kategori skripsi.



Gambar 3. 19 Tampilan Halaman Data Kategori Skripsi

7. Halaman Beranda *User*

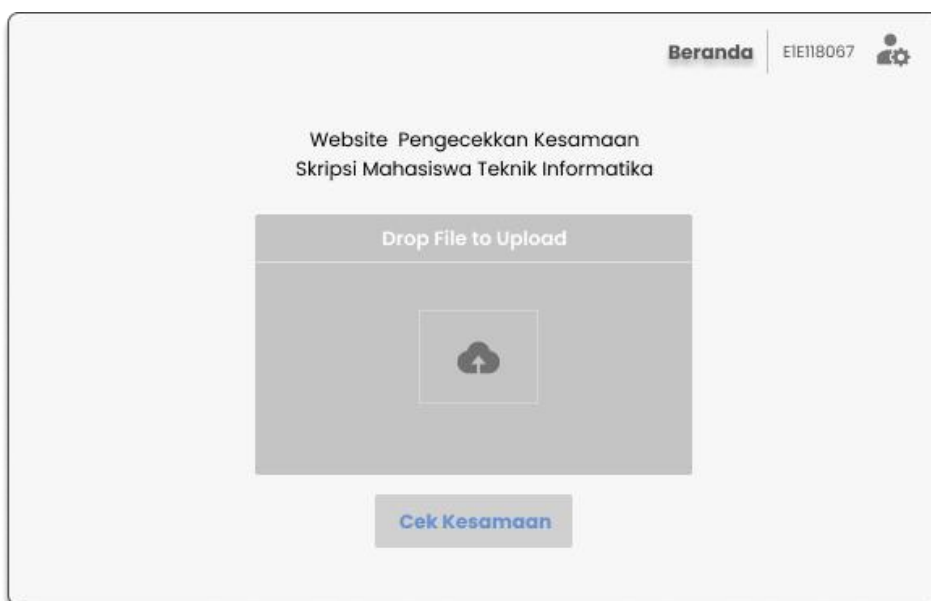
Halaman beranda merupakan ketika *user* berhasil *login* kedalam sistem pengecekan kesamaan skripsi mahasiswa.



Gambar 3. 20 Tampilan Halaman Beranda *User*

8. Halaman Upload Dokumen Skripsi *User*

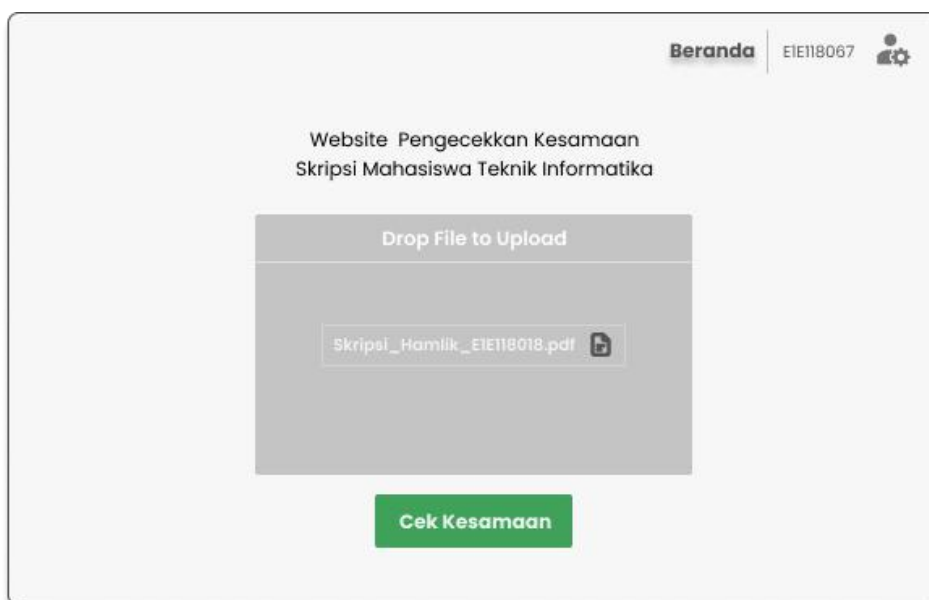
Halaman dimana *user* akan mengupload dokumen skripsi.



Gambar 3. 21 Tampilan Halaman Upload Dokumen Skripsi *User*

9. Halaman Sesudah *User Upload* Dokumen

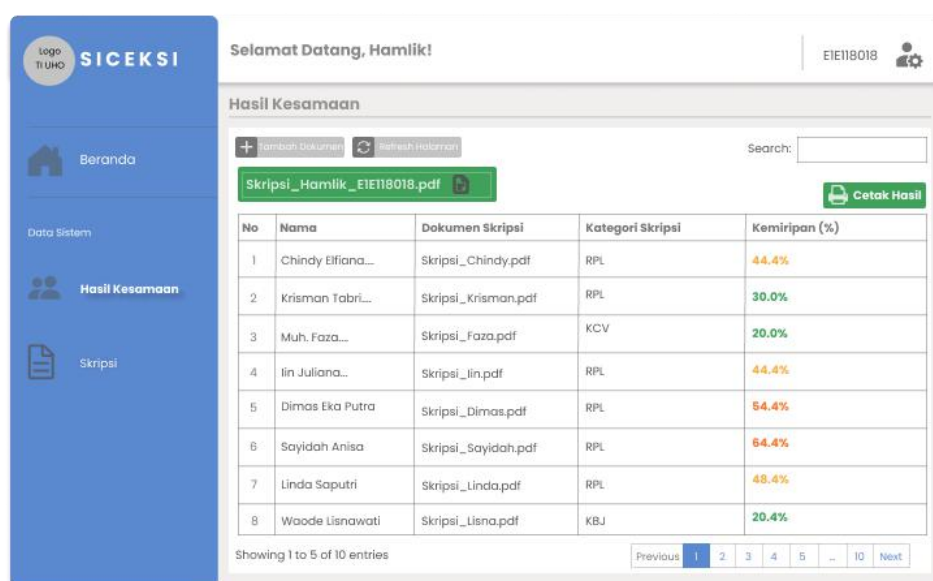
Halamaan dimana *user* sesudah upload dokumen skripsi dengan format pdf.



Gambar 3. 22 Tampilan Halaman Sesudah *User Upload* Dokumen

10. Halaman Hasil Kesamaan Skripsi

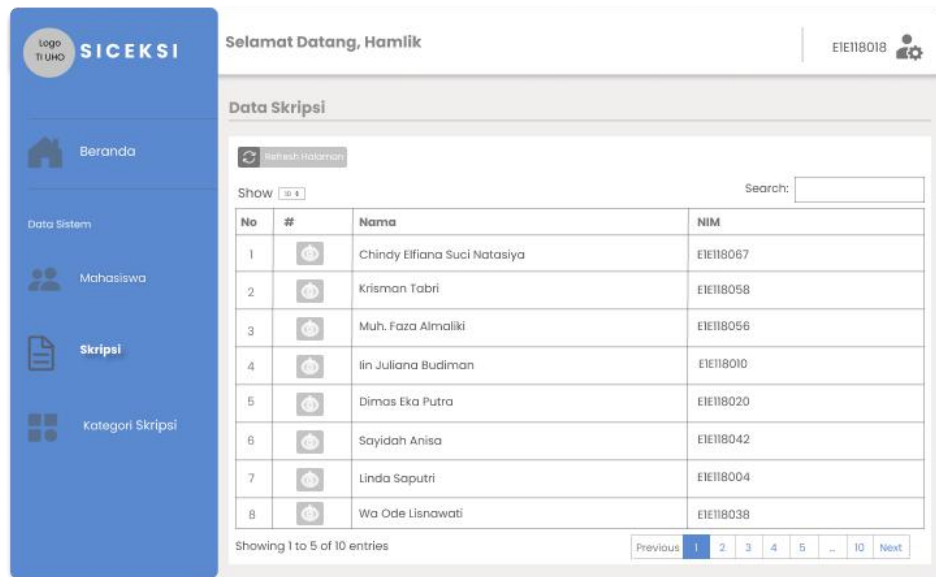
Halaman dimana ketika *user* berhasil melakukan dan menekan tombol cek kesamaan dan sistem akan menampilkan hasil kesamaan dokumen skripsi.



Gambar 3. 23 Tampilan Halaman Hasil Kesamaan Skripsi

11. Halaman Data Skripsi User

Halaman dimana *user* dapat melihat data skripsi yang terdapat pada sistem.



Gambar 3. 24 Tampilan Halaman Data Skripsi User

3.6 Pengujian Perangkat Lunak

3.6.1 Black Box Testing

Pengujian *black box* dalam sistem ini dilakukan dengan mengecek semua fungsi dari tampilan interface pada sistem dan memastikan semua fungsi berjalan sesuai harapan dan bekerja dengan baik.

3.6.2 Confusion Matrix

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual.

Tabel 3. 11 Confusion Matrix

n = 4	Aktual Positif (1)	Aktual Negatif (0)
Prediksi Positif (1)	TP : a	FP : c
Prediksi Negatif (0)	FN : b	TN : d
	a + b	c + d

1. Akurasi (*Accuracy*)

$$Akurasi = \frac{a + d}{a + d + c + b} \times 100\% \quad (3.1)$$

2. Presisi (*Precision*)

$$Presisi = \frac{a}{a + c} \times 100\% \quad (3.2)$$

3. Recall (Sensitivity)

$$Presisi = \frac{a}{a + c} \times 100\% \quad (3.3)$$

4. *F-Measures (F-Score)*

$$F - Measure = 2 \times \frac{Presisi \times recall}{Presisi + recall} \times 100\% \quad (3.4)$$

DAFTAR PUSTAKA

- Ahmad. (2021). *Karya Tulis Ilmiah: Pengertian, Fungsi, Struktur dan Contoh KTI*. Gramedia. <https://www.gramedia.com/literasi/karya-tulis-ilmiah/>
- Aritonang, L. W. (2020). Rancang Bangun Aplikasi Deteksi Kemiripan Dua Gambar Menggunakan Algoritma Ratcliff/Obershelp. *Journal of Computer System and Informatics ...*, 11(3), 191–198. <https://ejurnal.seminar-id.com/index.php/josyc/article/view/169>
- Ayunindya, F. (2022). *Apa Itu HTML? Fungsi dan Cara Kerja HTML*. Hostinger. <https://www.hostinger.co.id/tutorial/apa-itu-html>
- Azis, Y. A. (2021). *Apa itu Skripsi? Pengertian, Jenis, Struktur dan Karakteristik*. <https://penerbitbukudeepublish.com/apa-itu-skripsi/>
- Binus University. (2020). *Apa itu User Interface Design*. <https://binus.ac.id/knowledge/2020/01/apa-itu-user-interface-design/>
- Darmalaksana, W. (2019). Panduan Penulisan. *TESLA: Jurnal Teknik Elektro*, 21(1), 1. <https://doi.org/10.24912/tesla.v21i1.3257>
- Darmawan, I. A., Fatkhurrokhman, M., Darmawan, I. A., Fatkhurrokhman, M., Sultan, U., & Tirtayasa, A. (2022). *Penerapan Media Pembelajaran Pada Mata Pelajaran Sistem Elektropneumatik Berbasis website*. 6(1), 97–103.
- Deff, D. (2018). *Apa itu File PDF dan Bagaimana Cara Membuat File PDF*. <https://www.dafideff.com/2018/09/apa-itu-file-pdf-dan-bagaimana-cara-membuat-file-pdf.html>
- E. Prasetyo. (2012). *Data Mining: Konsep dan Aplikasi menggunakan Matlab*.
- Emny Harna Yossy, S.Kom., M. T. I. (2020). *Metode-Metode Information Retrieval*. <https://onlinelearning.binus.ac.id/computer-science/post/metode-metode-information-retrieval>
- Endang Anjarwani, S. (2020). *PENERAPAN METODE RATIONAL UNIFIED PROCESS (RUP) DALAM PENGEMBANGAN SISTEM INFORMASI MEDICAL CHECK UP PADA CITRA MEDICAL CENTRE (The Application of RationalUnifiedProcess (RUP) in Development of a Medical CheckUpInformation System at Citra Medical Centre)*. 2(1), 76–88. <http://jtika.if.unram.ac.id/index.php/JTIKA/>
- Gani, L., & Achmad, A. (2019). Website dan HTML Daftar Isi. *Website dan HTML*, 58. <https://www.pustaka.ut.ac.id/lib/wp-content/uploads/pdfmk/MSIM4309-M1.pdf>

- Hendra W Saputro. (2019). *Pengertian Website, Web Hosting Dan Domain Name*. BOC Indonesia. <https://www.boc.web.id/pengertian-website-webhosting-domainname/>
- Hermawan, A. (2019). Kebijakan Dosen Mengurangi Plagiarisme pada Karya Ilmiah Mahasiswa. *IJIP : Indonesian Journal of Islamic Psychology*, 1(2), 264–284. <https://doi.org/10.18326/ijip.v1i2.264-284>
- IBM. (2021). *Use-case diagrams*. <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>
- Ilyankou, I. (2014). Comparison of Jaro-Winkler and Ratcliff / Obershelp algorithms in spell check. *Ieee*, May, 1–35.
- Indonesia, L. S. K. (2021). *Fungsi HTML (HyperText Markup Language)*. <https://lmsspada.kemdikbud.go.id/mod/page/view.php?id=57743>
- Jacob, V. E., Lumenta, A. S. M., Jacobus, A., Elektro, T., Sam, U., & Manado, J. K. B. (2019). Rancang Bangun Aplikasi Kemiripan Dokumen Dengan Sumber – Sumber Internet. *Rancang Bangun Aplikasi Kemiripan Dokumen Dengan Sumber – Sumber Internet*, 14(2), 159–164.
- Joane, Y. L., Sinsuw, A., & Jacobus, A. (2017). Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Ratcliff/Obershelp. *Jurnal Teknik Informatika*, 11(1). <https://doi.org/10.35793/jti.11.1.2017.17654>
- Josi, A. (2017). Penerapan Metode Prototyping Dalam Membangun Website Desa (Studi Kasus Desa Sugihan Kecamatan Rambang). *Jti*, 9(1), 50–57.
- Julianto, S., & Setiawan, S. (2019). Perancangan Sistem Informasi Pemesanan Tiket Bus Pada Po. Handoyo Berbasis Online. *Simatupang, Julianto Sianturi, Setiawan*, 3(2), 11–25. <https://journal.amikmahaputra.ac.id/index.php/JIT/article/view/56/48>
- MEILIANA, S.KOM., M. S. (2018). *Siklus Pengujian Perangkat Lunak*. Binus University. <https://socs.binus.ac.id/2018/12/06/siklus-pengujian-perangkat-lunak/>
- Menteri Pendidikan Nasional. (2010). Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 17 Tahun 2010 Tentang Pencegahan Dan Penanggulangan Plagiat Di Perguruan Tinggi. *International Institute for Environment and Development*, 07/80(2), 125. https://arxiv.org/pdf/1707.06526.pdf%0Ahttps://www.yrpri.org%0Ahttp://weekly.cnbnews.com/news/article.html?no=124000%0Ahttps://www.fordfoundation.org/%0Ahttp://bibliotecavirtual.clacso.org.ar/Republica_Dominicana/ccp/20120731051903/prep%0Ahttp://webpc.cia
- Michael Kharisma Hutauruk. (2019). *UML Diagram : Use Case Diagram* (hal. 1). <https://socs.binus.ac.id/2019/11/26/uml-diagram-use-case-diagram/>

- Mubarak, A. (2019). Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek. *JIKO (Jurnal Informatika dan Komputer)*, 2(1), 19–25. <https://doi.org/10.33387/jiko.v2i1.1052>
- Novantara, P. (2018). Implementasi Algoritma Jaro-Winkler Distance Untuk Sistem Pendeteksi Plagiarisme Pada Dokumen Skripsi. *Buffer Informatika*, 3(1). <https://doi.org/10.25134/buffer.v3i2.960>
- Nugroho, A. (2019). *Pengertian XAMPP*. qwords. <https://qwords.com/blog/pengertian-xampp/>
- Nur Ariesanto Ramdhan, D. A. N. (2019). RANCANG BANGUN DAN IMPLEMENTASI SISTEM INFORMASI SKRIPSI ONLINE BERBASIS WEB. *Jurnal Ilmiah Intech*, 1(November 2019), 1–17.
- Nuraini Ahmad, Arienda Addis Prasetyo, A. M. (2021). *PENERAPAN INFORMATION RETRIEVAL PADA SEARCH ENGINE*. 1(1), 1–9.
- Ode, W., Karyima, R., Sarita, M. I., & Ningrum, I. P. (2019). *Wavelet dalam sistem temu kembali informasi kemiripan motif kain tenun sulawesi tenggara*. 5(1), 9–16.
- PDDikti. (2022). *Pastikan Data Anda terdaftar pada PDDikti*. <https://pddikti.kemdikbud.go.id/>
- Pratama, E. E. (2018). Information Retrieval pada Proses Penyimpanan dan Pencarian Dokumen Digital Menggunakan Metode Text Mining Enda Eisyudha Pratama. *Fakultas Teknik Jurusan Informatika*, 736–742.
- Putra, H. N. (2018). Implementasi Diagram UML (Unified Modelling Language) dalam Perancangan Aplikasi Data Pasien Rawat Inap pada Puskesmas Lubuk Buaya. *Sinkron: jurnal dan penelitian teknik informatika*, 2(2), 67–77. <https://jurnal.polgan.ac.id/index.php/sinkron/article/view/130>
- Riadi, M. (2019). *Pengertian, Jenis dan Identifikasi Plagiarisme*. <https://www.kajianpustaka.com/2019/02/Plagiarisme.html>
- Sawaluke Chaosakul. (2015). SISTEM INFORMASI PENGELOLAAN ARSIP STATIS PADA BADAN ARSIP DAN PERPUSTAKAAN PROVINSI JAWA TENGAH MENGGUNAKAN VECTOR SPACE MODEL. *Journal of Education Graduate Studies Research*, 5, 191–196. <http://www.tci-thaijo.org/index.php/EDGKKUJ/article/viewFile/51570/42715>
- Syafnidawaty. (2020). *Database*. Universitas Raharja. <https://raharja.ac.id/2020/04/25/database/>
- Syarif, M., Pratama, E. B., Bina, U., Informatika, S., & Barat, K. (2021). Testing dan Pemodelan Diagram Uml Pada Aplikasi Veterinary Services Yang Dikembangkan Dengan Model Waterfakk. *Jurnal Teknik Informatika Kaputama (JTik)*, 5(2), 253–258.

- Taryana, O. :, & Kom, S. M. (2021). Pengenalan HTML, Browser dan Text Editor. *Pengenalan HTML, Browser dan Text Editor*, 10. <https://us04web.zoom.us/j/3629329963?pwd=ZUhWcXl6RHp3dTNKZmlxWkUrV2ZCQT09>
- Tineges, R. (2021). *Fungsi MySQL, Kelebihan Serta Kekurangannya* (hal. 1). DQ LAB Indonesia. <https://dqlab.id/fungsi-mysql-kelebihan-serta-kekurangannya>
- W, R. Y. (2019). *Ekstensi dan Tema Visual Studio Code yang Saya Gunakan* (hal. 1). Medium. <https://medium.com/kode-dan-kodean/ekstensi-dan-tema-visual-studio-code-yang-saya-gunakan-6c3555762816>
- Yusuf, B., Vivianie, S., Malem Marsya, J., Sofyan, Z., Studi Teknologi Informasi, P., Sains dan Teknologi, F., Studi Pendidikan Teknologi Informasi, P., Tarbiyah dan Keguruan, F., & Ar-Raniry Banda Aceh, U. (2019). Analisis Perbandingan Algoritma Rabin-Karp dan Ratcliff/Obershelp untuk Menghitung Kesamaan Teks dalam Bahasa Indonesia. *Seminar Nasional APTIKOM*, 2019. <https://www.tempo.co/>