BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisa Kebutuhan Fungsional dan Non Fungsional

3.1.1 Analisa Kebutuhan Fungsional

Kebutuhan fungsional yang dibutuhkan oleh sistem ini adalah :

1. Membaca teks dokumen berupa *.txt*, *.doc*, dan *.pdf*.
2. Memproses kalimat menjadi kata dengan menggunakan *preprocessing*.
3. Melakukan *tokenization* kata.
4. Menyaring kata-kata yang umum, seperti saya, adalah, kamu, dia, dan lainnya.
5. *Hashing* dilakukan terhadap per kata.
6. Menghitung masing-masing *similarity* dari perkata dengan hasil akhir berupa persentase kesamaan.

3.1.2 Analisa Kebutuhan Non Fungsional

Spesifikasi perangkat keras yang di butuhkan oleh sistem ini adalah :

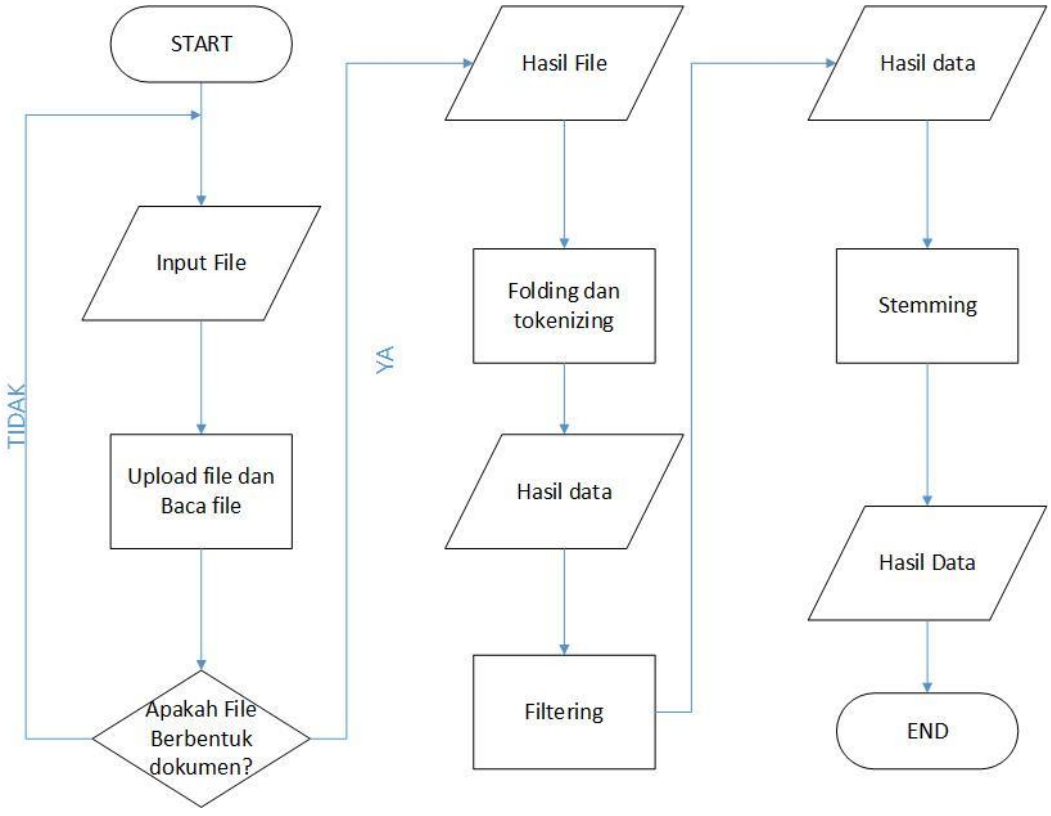
1. *Processor* Intel Core i5 6200U 2.3Ghz
2. *Memory* RAM 12 GB DDR3L
3. SSD 240GB

4. Perangkat *input* berupa *mouse* dan *keyboard*

5. Perangkat *output* berupa monitor *LCD* atau *LED*

3.2 Perancangan Proses Sistem

Adapun perancangan dari sistem yang sudah disediakan dalam bentuk *flowchart*, yang membantu dalam mengartikan sebuah alur dari program yang akan dibuat :

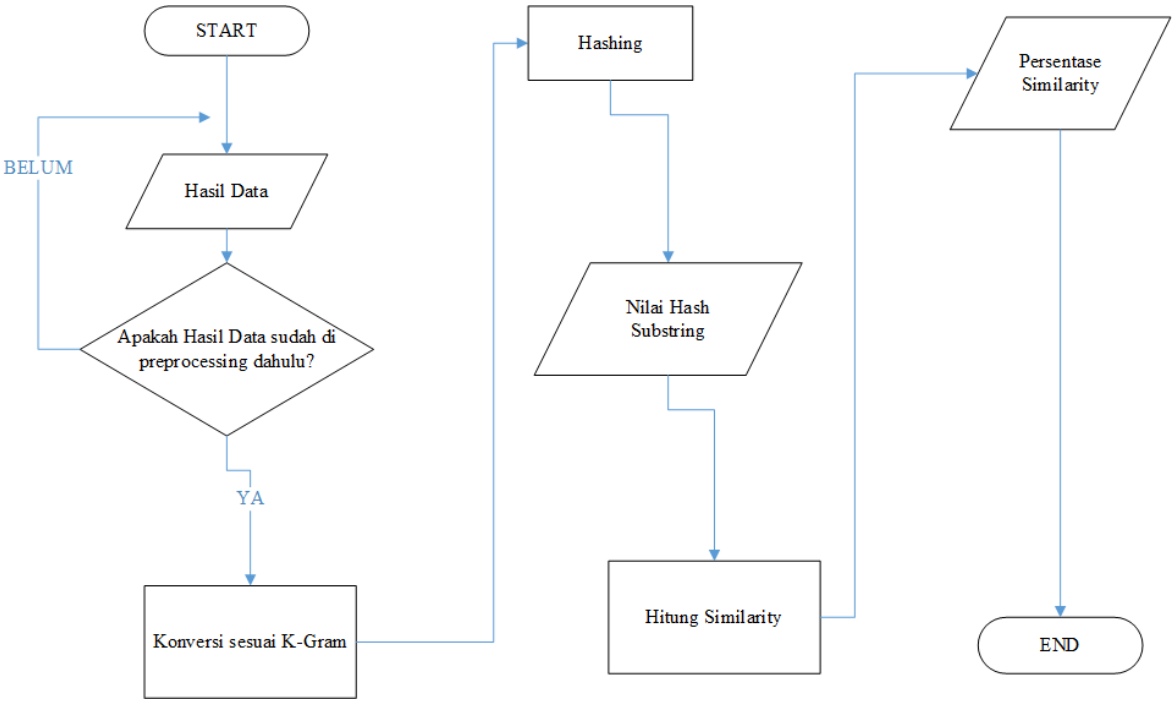


Gambar 3.1 Rancangan *Flowchart* untuk Aplikasi

Gambar 3.1 menjelaskan mengenai proses kerja sistem yang meliputi proses pengecekan dokumen dan hasil proses *text preprocessing*. Yang mana pada akhir

dari proses ini adalah berupa hasil data yang akan diolah selanjutnya oleh algoritma *Rabin Karp*.

Perancangan selanjutnya adalah proses ketika hasil data yang sudah di proses, akan dikerjakan oleh algoritma *Rabin Karp* yang mana *flowchart* algoritma *Rabin Karp* bisa di lihat di bawah ini :



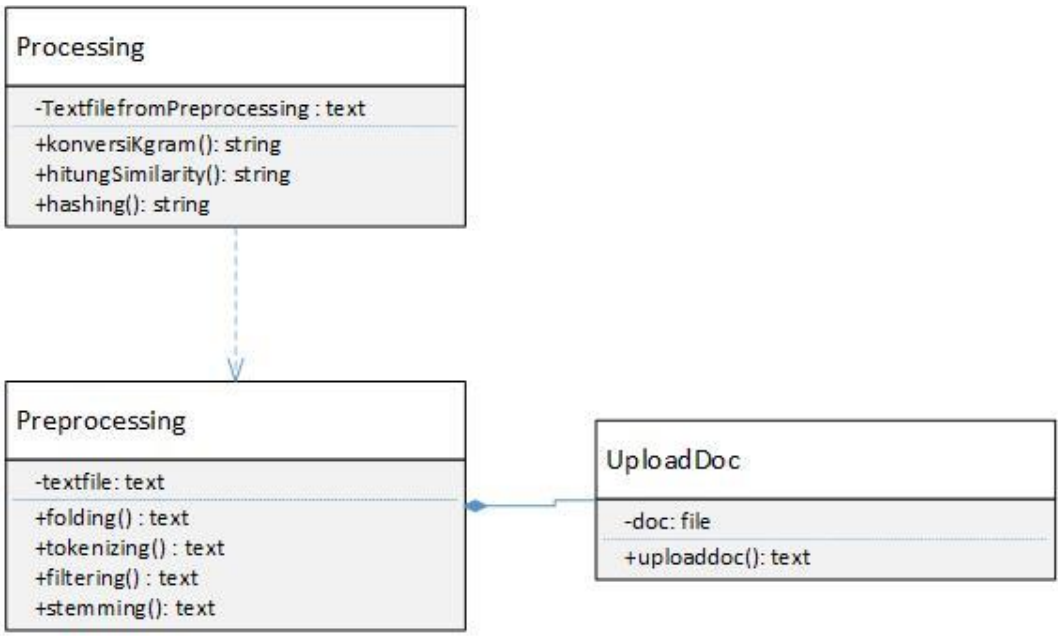
Gambar 3.2 Rancangan *Flowchart* Algoritma *Rabin Karp*

Gambar 3.2 menjelaskan tentang algoritma *Rabin Karp* dimana dari hasil data sebelum nya di jadikan sebagai masukan untuk di proses selanjutnya. Nilai *K-gram* sangat mempengaruhi cara algoritma mengambil kata yang kemudian di *hashing* menghasilkan nilai *substring*, kemudian antar dokumen satu dengan yang lain akan dibandingkan nilai *hash substring* yang di ketahui *similarity* dari hasil perbandingan. Yang menghasilkan output terakhir yang disebut sebagai persentase yang bertujuan untuk mengetahui tingkat plagiarisme yang terkandung.

3.3 Perancangan Sistem

Perancangan sistem yang digunakan adalah diagram UML berikut penjelasan :

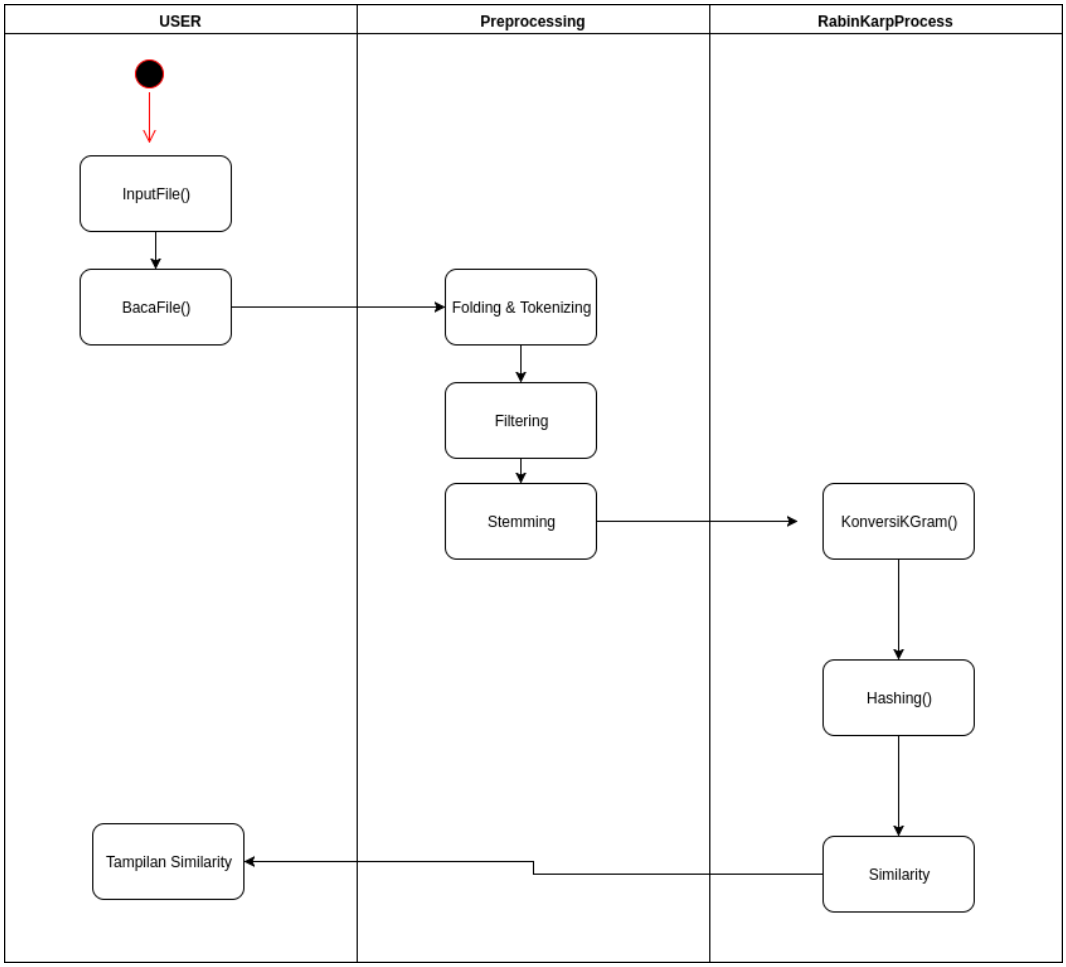
3.3.1 Perancangan Class Diagram



Gambar 3.3 Rancangan *Class* Diagram

Gambar 3.3 menjelaskan tentang *Class Diagram* dimana proses *Processing* di situ mempunyai sebuah *dependency* (ketergantungan) terhadap *class Preprocessing*, hal ini dikarenakan sebuah *class Processing* tidak bisa berjalan jika *Preprocessing* belum selesai dikerjakan. Dan *UploadDoc* mempunyai komposisi terhadap *class Preprocessing*, yang mana untuk menjalankan semua ini dibutuhkan input yang berasal dari *UploadDoc*. Ini berketerkaitan antara satu sama lain.

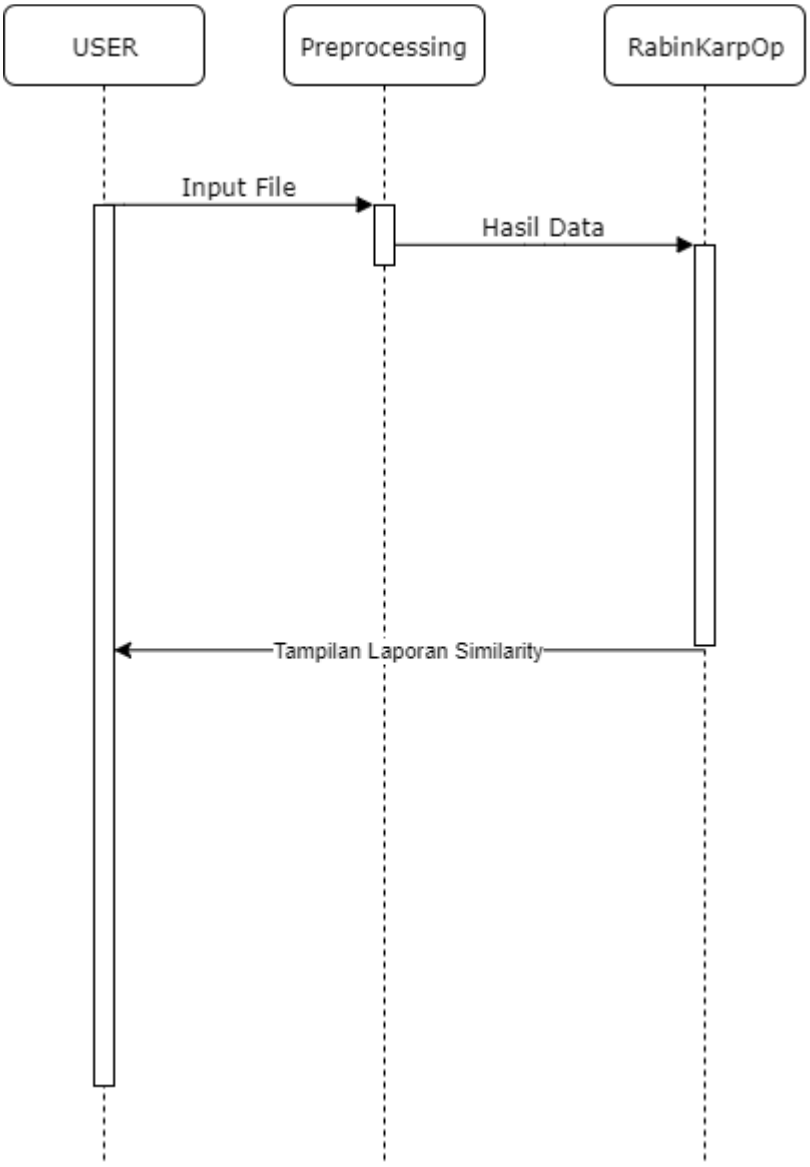
3.3.2 Perancangan Activity Diagram



Gambar 3.4 Rancangan Activity Diagram

Gambar 3.4 menjelaskan tentang diagram aktivitas yang terjadi dalam aplikasi yang akan dibuat. Alur aktivitas ini di awali oleh *user* (pengguna) yang melakukan aksi memasukkan *file* berupa dokumen, kemudian program akan membaca dokumen itu *valid* atau tidak. Kemudian akan dilanjutkan di sesi *preprocessing* dimana dalam tahap ini, kata-kata di dalam dokumen akan diolah menjadi hasil data. Kemudian proses akan dilanjutkan kepada fungsi *RabinKarpProcess* yang melakukan perhitungan *hashing* berdasarkan pada jumlah *K-gram* yang sudah di tentukan, hasil *output* akan keluar berupa persentase *similarity* yang akan disajikan kepada *user* dalam bentuk tampilan laporan.

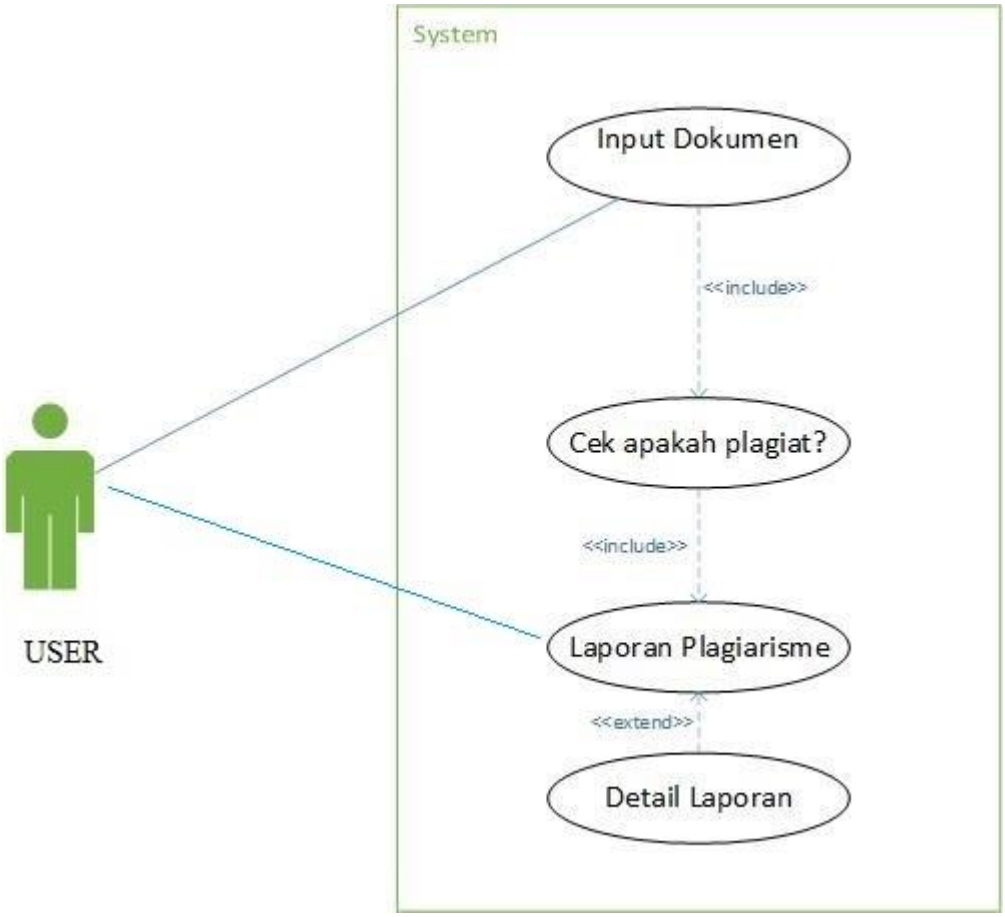
3.3.3 Perancangan *Sequence* Diagram



Gambar 3.5 Rancangan *Sequence* Diagram

Gambar 3.5 menjelaskan tentang *event* perjalanan yang dilewati dari aplikasi, yang dimulai dari user menginput file sampai dari hasil file menghasilkan hasil data yang di proses oleh algoritma dan kemudian hasil algoritma akan di proses menjadi sebuah tampilan laporan persentase yang menunjukkan hasil akhir.

3.3.4 Perancangan *Use Case* Diagram

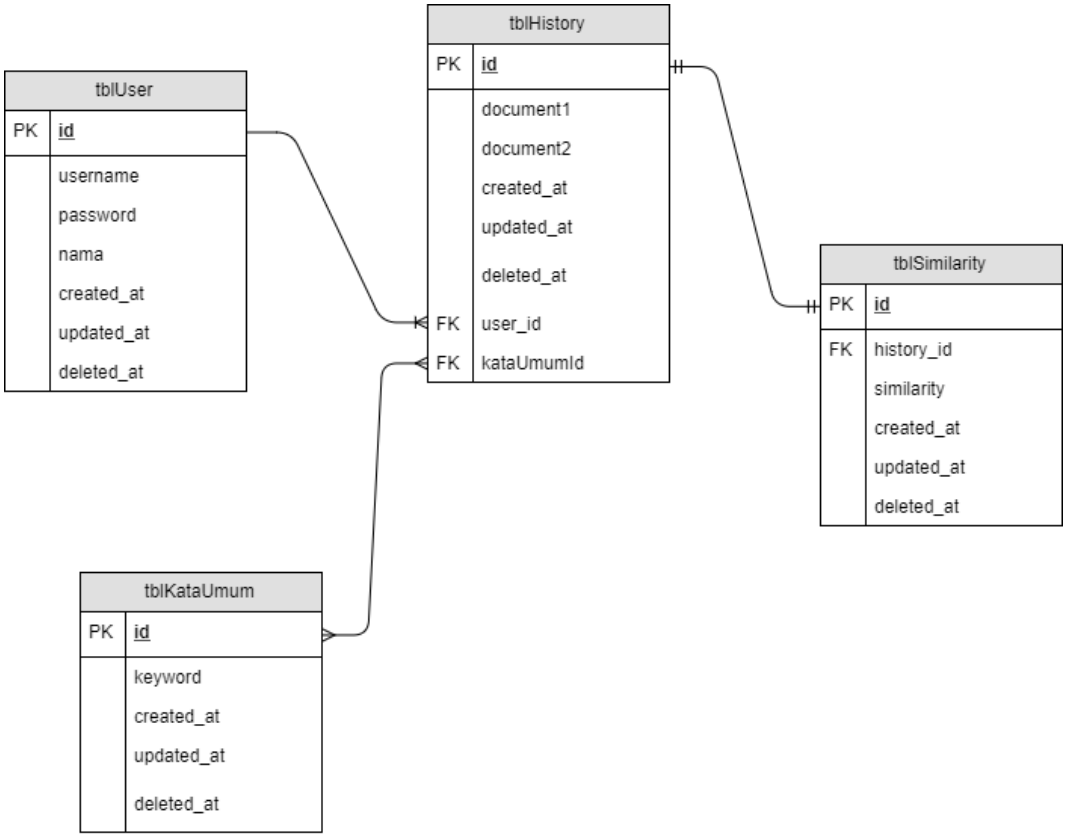


Gambar 3.6 Rancangan *Use Case* Diagram

Gambar 3.6 menjelaskan tentang *User* (pengguna) yang melakukan *action* berupa *input* file dokumen, yang kemudian akan di proses untuk di cek apakah plagiat atau tidak. Dengan hasil akhir berupa laporan plagiarisme. Dan mempunyai relasi antar satu sama lain, contohnya laporan plagiarisme membutuhkan data dari *input* dokumen supaya ada hasilnya.

3.3.5 Perancangan *Entity Relationship Diagram* (ERD)

Selain perancangan diagram UML di atas, adapun diagram database, yakni *Entity Relationship Diagram* (ERD) yang mana memakai *Crow's Foot* sebagai model *database*, yang bisa dilihat sebagai berikut :



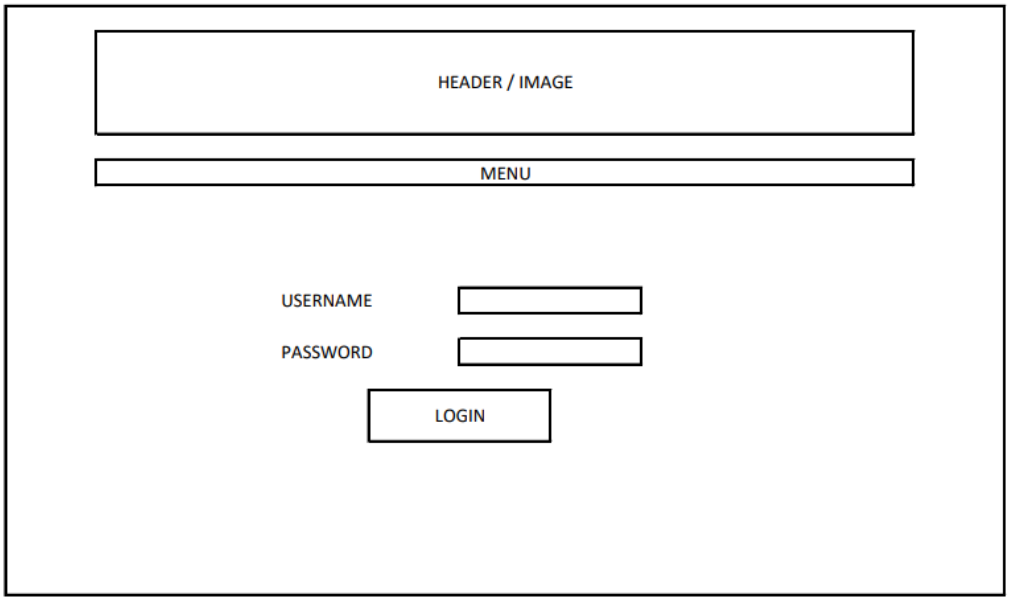
Gambar 3.7 Rancangan *Crow's Foot* Model

Gambar 3.7 menjelaskan tentang keterkaitan tabel yang di buat dalam aplikasi ini. Tabel yang dipakai terdiri dari tabel user, tabel history, table kata umum, dan tabel similarity. Untuk tabel user ke tabel history menggunakan rule one to many. Sedangkan untuk tabel kata umum dengan tabel history menggunakan rule many to many, dan yang terakhir tabel history dengan tabel similarity menggunakan rule one to one.

3.5 Perancangan Tampilan

Fitur-fitur yang tersedia dalam web ini adalah :

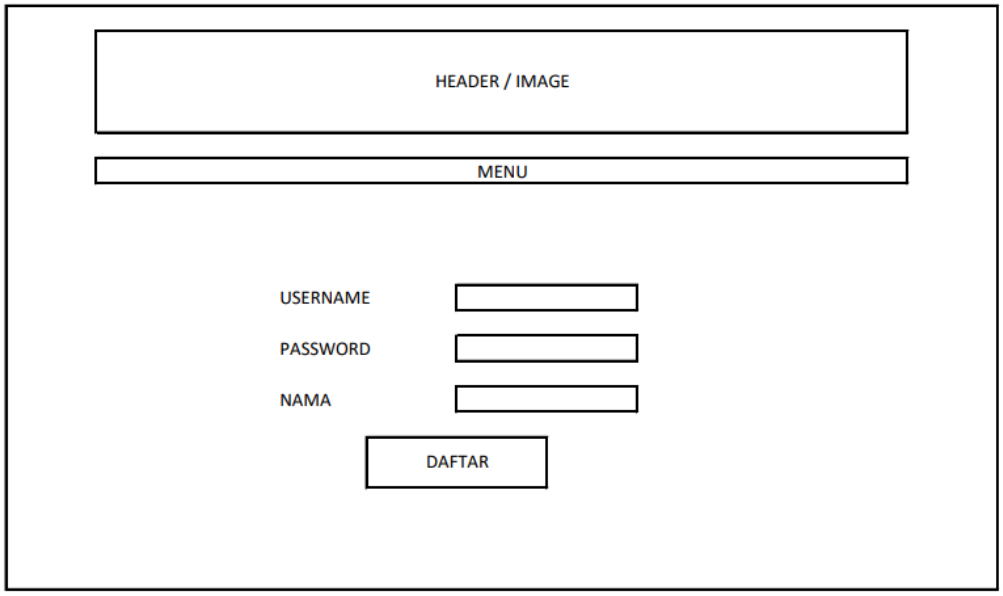
1. *Login*



Gambar 3.8 Rancangan Tampilan *Login*

Gambar 3.8 menjelaskan rancangan tampilan *login* dimana pengguna diwajibkan memasukkan *username* dan *password* sebagai tanda pengenal. *Username* dan *password* ini akan di validasi bilamana salah satu kolom isian kosong, maka akan terjadi sebuah peringatan atau instruksi yang mengharuskan kita untuk mengisi kolom yang kosong. Setelah halaman ini berhasil *login*, maka pengguna akan di *redirect* kepada halaman *home* dengan informasi pengguna yang disesuaikan dengan informasi data yang sudah dimasukkan ketika pengguna mendaftar.

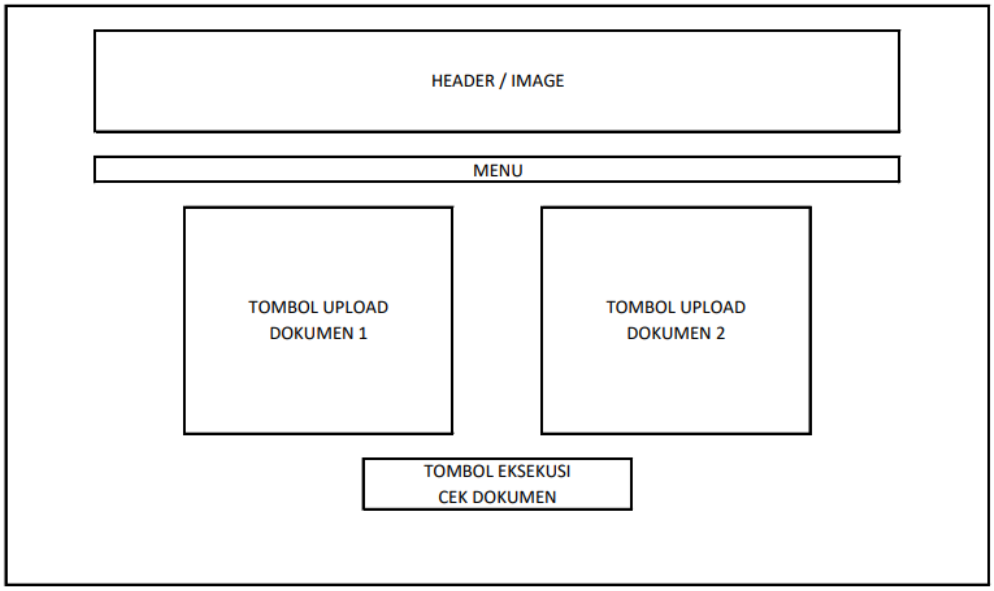
2. Daftar



Gambar 3.9 Rancangan Tampilan Daftar

Gambar 3.9 menjelaskan tampilan daftar pengguna, dimana pengguna ini bisa mendaftar dengan memasukkan *username*, *password*, dan nama. Tombol daftar berfungsi untuk memproses semua hasil informasi yang sudah dimasukkan ke dalam *database*. Ada validasi dalam kolom isian ini, bilamana salah satu tidak diisi, maka akan muncul sebuah pesan yang memberi tahu bahwa pengguna tidak memasukkan atau mengisi pada kolom yang sudah di sediakan. Setelah pengguna berhasil mendaftar, maka hasil dari data yang sudah di validasi dan sudah dinyatakan valid, maka data akan berhasil masuk ke dalam database, yang kemudian bisa dipakai untuk di olah atau di gunakan. Hasil dari pengguna yang mendaftar ada disediakan *timestamp* yang bisa memantau kapan pengguna mendaftar pada halaman kita. Hasil dari daftar ini akan masuk ke dalam tabel *User* sesuai dengan kolom dan tempat yang sudah di sediakan oleh *database*.

3. Home



Gambar 3.10 Rancangan Tampilan *Home*

Gambar 3.10 menjelaskan rancangan tampilan halaman depan. Dimana pengguna disajikan 2 tombol untuk *upload* dokumen, ada dua tombol yang ada disini, dokumen 1 dan dokumen 2. Kemudian setelah pengguna selesai menggunakan tombol *upload*. Tombol eksekusi cek dokumen berfungsi sebagai *submit* yang bertujuan untuk memproses semua hasil dokumen, dengan hasilnya berupa laporan hasil kesamaan antar dokumen 1 dan dokumen 2.

3.6 Gantt Chart

NO	TASK		AGUSTUS			September				Oktober				November				Desember			
		WEEK	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Initial Investigation																				
2	System Analyst																				
3	System Design																				
4	Implementation																				

Gambar 3.11 *Gantt Chart* Pengerjaan Laporan Penelitian

Penjelasan dari gambar 3.11 adalah :

1. *Initial Investigation*

   Rencana melakukan pengumpulan data dari Agustus minggu kedua sampai minggu keempat.

2. *System Analyst*

   Rencana untuk melakukan analisis sistem dari Agustus minggu keempat sampai Oktober minggu ketiga.

3. *System Design*

   Rencana untuk merancang pada sistem mulai Oktober minggu keempat sampai Desember minggu pertama.

4. *Implementation*

   Rencana untuk menerapkan percobaan algoritma *Rabin Karp* pada sistem mulai November minggu kedua sampai Desember minggu keempat.