# BAB III METODOLOGI

Bab ini menjelaskan tahapan dan pendekatan yang digunakan dalam mengembangkan aplikasi pendeteksi plagiarisme project akhir menggunakan algoritma Rabin-Karp. Tahapan yang dijabarkan mencakup metode pengembangan, deskripsi solusi, desain sistem, serta jadwal pelaksanaan proyek.

1. **DESKRIPSI SOLUSI**

Solusi yang ditawarkan pada proyek ini adalah pengembangan Aplikasi Pendeteksi Plagiarisme pada dokumen project akhir mahasiswa dengan memanfaatkan Algoritma Rabin-Karp. Aplikasi ini dirancang untuk membantu perguruan tinggi untuk memeriksa keaslian karya ilmiyah mahasiswa secara efisien dan juga transparan.

Masalah utama yang diangkat adalah banyaknya kasus plagiarisme yang tidak terdeteksi karena pemeriksaan masih dilakukan secara manual, atau menggunakan system berbayar yang kurang sesuai dengan kebutuhan instansi. Oleh karena itu, aplikasi ini hadir sebagai alternatif yang bisa digunakan secara mandiri oleh institusi dan juga mahasiswa.

Aplikasi ini memiliki beberapa fitur utama yang dirancang sebagai Solusi dari permasalahan tersebut:

1. **Unggah Dokumen**

Pengguna (mahasiswa atau dosen) dapat mengunggah dokumen project akhir dalam format tertentu seperti .pdf, .docx, .txt. Dokumen ini akan dianalisis oleh system.

1. **Pencocokan Teks Otomatis**

Sistem akan membandingkan isi dokumen dengan referensi yang sudah tersimpan dalam database. Proses pencocokan dilakukan menggunakan algoritma Rabin-Karp, yang dikenal cepat dan efisien dalam mendeteksi kemiripan teks.

1. **Hasil Persentase Kesamaan**

Setelah dianalisis, system akan menampilkan persentase kemiripan teks antara dokumen yang diperiksa dengan dokumen lain di dalam database.

1. **Highlight Bagian Teks yang Terindikasi**

Sistem akan menandai bagian-bagian teks yang dianggap memiliki kemiripan tinggi. Ini memudahkan pengguna untuk mengecek langsung bagian mana saja yang perlu diperhatikan.

1. **Laporan Hasil Deteksi**

Pengguna bisa melihat atau mengunduh laporan hasil pemeriksaan, yang berisi informasi seperti persentase kemiripan, bagian teks yang ditandai, serta sumber referensi yang serupa.

1. **Dashboard Untuk Mahasiswa dan Dosens**

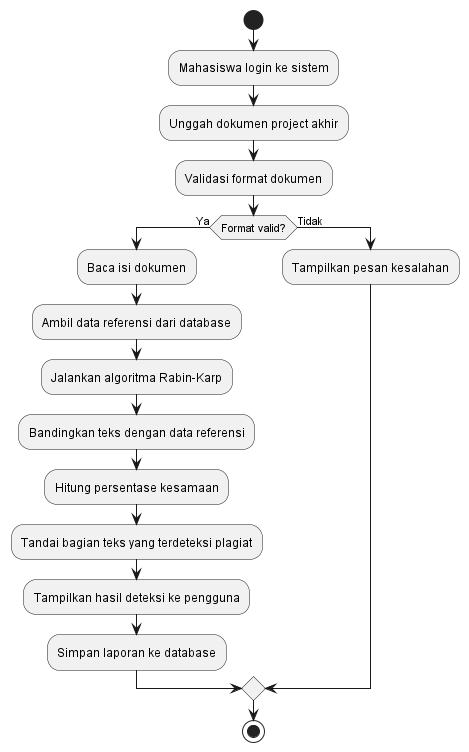
Mahasiswa bisa melihat Riwayat pengecekan dokumen mereka, sedangkan dosen atau admin bisa memantau aktivitas pengecekan dan mengakses laporan mahasiswa lainnya.

Aplikasi ini dikembangkan menggunakan Express.js, PostgreSQL, dan Prisma untuk bagian Backend, serta Next.js dan Tailwind CSS untuk tampilan frontend. Sistem juga dirancang agar berjalan secara real-time dan mudah digunakan oleh pengguna dari berbagai latar belakang.

Dengan fitur-fitur tersebut, diharapkan aplikasi ini bisa menjadi alat bantu yang efektif untuk menekan angka plagiarisme dan meningkatkan kesadaran mahasiswa akan pentingnya orisinalitas dalam penulisan ilmiah.

1. **DESAIN SYSTEM**
2. **Flowchart Aplikasi**

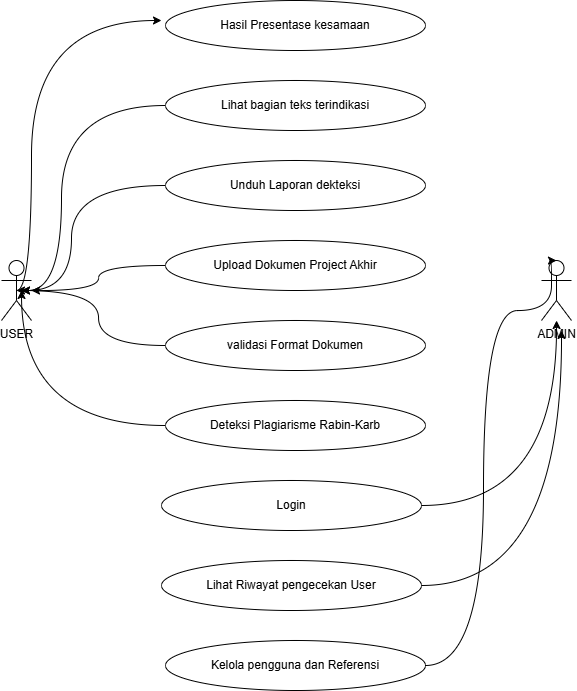
Flowchart ini menggambarkan alur kerja utama dari Aplikasi Pendeteksi Plagiarisme Project Akhir yang dikembangkan menggunakan algoritma Rabin-Karp. Proses dimulai dari saat pengguna masuk ke sistem hingga sistem menampilkan hasil deteksi plagiarisme.



Berikut penjelasan dari Langkah-langkah pada diagram di atas:

1. **Login ke Sistem**  
   Mahasiswa dan admin harus login terlebih dahulu untuk mengakses fitur aplikasi.
2. **Validasi Login**  
   Sistem memverifikasi akun. Jika tidak valid, pengguna akan mendapatkan pesan “Login gagal”.
3. **Unggah Dokumen**  
   Setelah login berhasil, mahasiswa dapat mengunggah dokumen project akhir untuk diperiksa.
4. **Validasi Format File**  
   Sistem mengecek apakah format file yang diunggah sesuai (.pdf, .docx, .txt). Jika tidak sesuai, akan muncul pesan kesalahan.
5. **Baca Isi Dokumen**  
   Jika format valid, sistem membaca teks dalam dokumen untuk dianalisis.
6. **Ambil Referensi dari Database**  
   Sistem memuat dokumen referensi sebagai bahan perbandingan.
7. **Deteksi dengan Algoritma Rabin-Karp**  
   Sistem mencocokkan teks dokumen dengan referensi menggunakan algoritma Rabin-Karp berbasis hashing.
8. **Hitung Persentase & Tandai Kemiripan**  
   Sistem menandai bagian teks yang mirip dan menghitung total persentase kemiripan.
9. **Tampilkan Hasil Deteksi**  
   Hasil deteksi ditampilkan kepada pengguna, termasuk bagian yang terindikasi plagiat.
10. **Simpan Laporan ke Database**  
    Hasil pemeriksaan disimpan dalam database agar bisa diakses kembali oleh mahasiswa atau admin.
11. **Use Case Diagram**

Use Case Diagram berikut menjelaskan fungsi-fungsi utama yang tersedia dalam aplikasi, serta bagaimana pengguna berinteraksi dengan sistem. Diagram ini membantu memberikan gambaran umum mengenai peran pengguna dan proses utama seperti mengelola dokumen dan melakukan pengecekan plagiarisme.



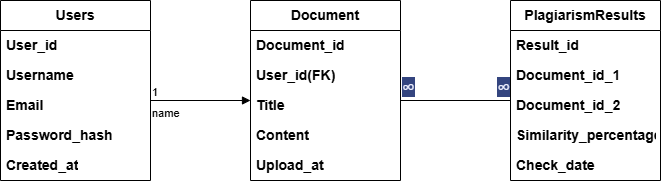
Penjelasan Langkah-langkah pada diagram:

* 1. Pada aplikasi ini ada 2 pengguna utama yakni USER dan ADMIN
  2. USER menggunakan aplikasi untuk mengecek dokumen tugas akhir mereka, apakah mengandung plagiarisme atau tidak.
  3. ADMIN bertugas untuk mengelola sistem dan memantau aktivitas pengecekan yang dilakukan oleh mahasiswa.

Fitur-fitur utama yang ada pada aplikasi:

1. **Login**  
   Digunakan oleh semua pengguna untuk masuk ke dalam sistem.
2. **Upload Dokumen Project Akhir**  
   Mahasiswa mengunggah dokumen tugas akhir yang ingin dicek.
3. **Validasi Format Dokumen**  
   Sistem akan memeriksa apakah file yang diunggah sesuai dengan format yang diperbolehkan (misalnya PDF, DOCX, atau TXT).
4. **Deteksi Plagiarisme (Rabin-Karp)**  
   Sistem menjalankan proses pengecekan isi dokumen menggunakan algoritma Rabin-Karp.
5. **Lihat Hasil Persentase Kesamaan**  
   Mahasiswa bisa melihat berapa persen kemiripan isi dokumen mereka.
6. **Lihat Bagian Teks Terindikasi**  
   Sistem menampilkan bagian teks mana saja yang dianggap mirip dengan referensi.
7. **Unduh Laporan Deteksi**  
   Mahasiswa dapat mengunduh laporan hasil pengecekan sebagai bukti.
8. **Kelola Pengguna & Dokumen Referensi**  
   Admin dapat mengatur data pengguna dan dokumen referensi yang dijadikan pembanding.
9. **Lihat Riwayat Pengecekan Mahasiswa**  
   Admin bisa melihat histori pengecekan dokumen dari mahasiswa yang pernah menggunakan sistem.
10. **Entity Relationship Diagram (ERD)**

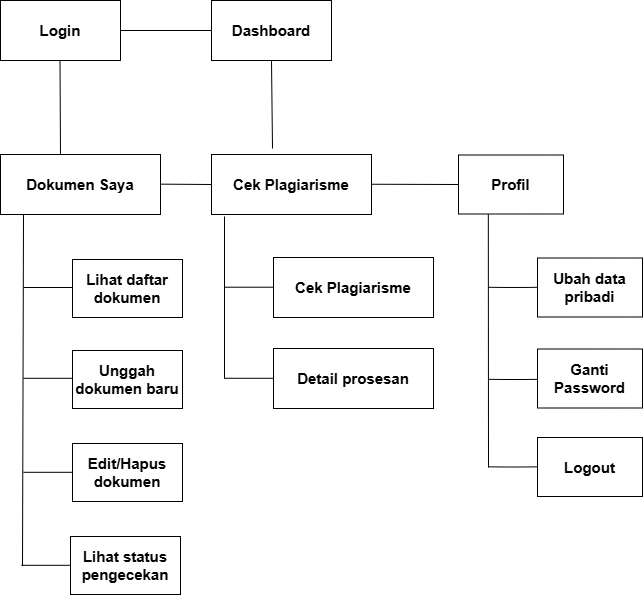
Entity Relationship Diagram (ERD) berikut menggambarkan hubungan antar entitas yang terdapat dalam sistem Aplikasi Pendeteksi Plagiarisme Project Akhir. Diagram ini dirancang untuk memodelkan struktur basis data, termasuk entitas utama seperti pengguna, dokumen, dan hasil pengecekan plagiarisme beserta relasinya.

****

Penjelasan dari ERD:

1. **User**
2. Entitas pengguna sistem.
3. Menyimpan data login dan identitas pengguna.
4. Satu user dapat mengunggah banyak dokumen.
5. **Document**
6. Entitas dokumen yang diunggah oleh user.
7. Berisi judul, isi, dan waktu unggah.
8. Digunakan untuk proses pengecekan plagiarisme.
9. **PlagiarismResult**
10. Entitas hasil pengecekan plagiarisme.
11. Menyimpan dua dokumen yang dibandingkan.
12. Berisi persentase kemiripan dan tanggal pengecekan.
13. **Struktur Menu**

Struktur menu berikut dirancang untuk mempermudah navigasi pengguna dalam menggunakan Aplikasi Pendeteksi Plagiarisme Project Akhir. Menu-menu ini disusun berdasarkan fitur utama yang tersedia dalam sistem, mulai dari proses login, pengelolaan dokumen, pengecekan plagiarisme, hingga pengaturan profil pengguna.

`

Berikut ini penjelasan dari Struktur Menu:

* + 1. **Login**

Fungsi: Mengautentikasi pengguna sebelum masuk ke sistem.

Deskripsi: Pengguna harus memasukkan email dan password untuk dapat mengakses fitur aplikasi.

* + 1. **Dashboard**

Fungsi: Memberikan gambaran umum aktivitas pengguna dalam aplikasi.

Deskripsi: Menampilkan statistik jumlah dokumen, jumlah pengecekan plagiarisme, dan status terakhir dari dokumen yang telah dicek.

* + 1. **Dokumen Saya**

Fungsi: Mengelola dokumen milik pengguna.

Deskripsi: Pengguna dapat mengunggah dokumen baru, melihat daftar dokumen, mengedit judul atau isi dokumen, serta menghapus dokumen yang tidak dibutuhkan.

* + 1. **Cek Plagiarisme**

Fungsi: Melakukan proses pengecekan plagiarisme antar dokumen.

Deskripsi: Pengguna memilih dokumen yang ingin dicek, lalu sistem membandingkan dokumen tersebut dengan dokumen lain menggunakan algoritma Rabin-Karp dan menampilkan hasilnya.

* + 1. **Hasil Pengecekan**

Fungsi: Menyimpan dan menampilkan histori pengecekan plagiarisme.

Deskripsi: Menampilkan daftar hasil pengecekan, termasuk persentase kemiripan, nama dokumen yang dibandingkan, dan tanggal pengecekan.

* + 1. **Profil**

Fungsi: Mengelola informasi akun pengguna.

Deskripsi: Pengguna dapat mengubah data profil seperti nama, email, dan password.

* + 1. **Logout**

Fungsi: Keluar dari sesi pengguna di sistem.

Deskripsi: Mengakhiri sesi login dan kembali ke halaman login.

## JADWAL PELAKSANAAN

Berikut adalah jadwal perkiraan yang akan menjadi target dalam merancang Aplikasi Pendeteksi Plagiarisme Project Akhir Menggunakan Algoritma Rabin-karp:

| Kegiatan | Bulan Ke- | | | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | | | | 2 | | | | 3 | | | | 4 | | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Analisis Kebutuhan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desain Dan Perancangan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pembuatan Kode Program |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |