

BAB IV

HASIL DAN UJI COBA

Dalam tahap implementasi sistem ada beberapa syarat yang harus disiapkan sebelumnya. Syarat-syarat tersebut meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*).

IV.1. Tampilan Hasil

Berikut ini adalah penjelasan tentang tampilan hasil dari perancangan pendeteksi *plagiarisme* dokumen teks dengan algoritma *Rabin – Karp*.

1. Tampilan JFrame Utama

Pada JFrame utama ini berfungsi sebagai tampilan utama dari program aplikasi pendeteksi *plagiarism*. Didalam JFrame utama ini menyediakan button seperti : Masuk dan Keluar. Gambar hasil dari JFrame utama seperti pada gambar IV.I.

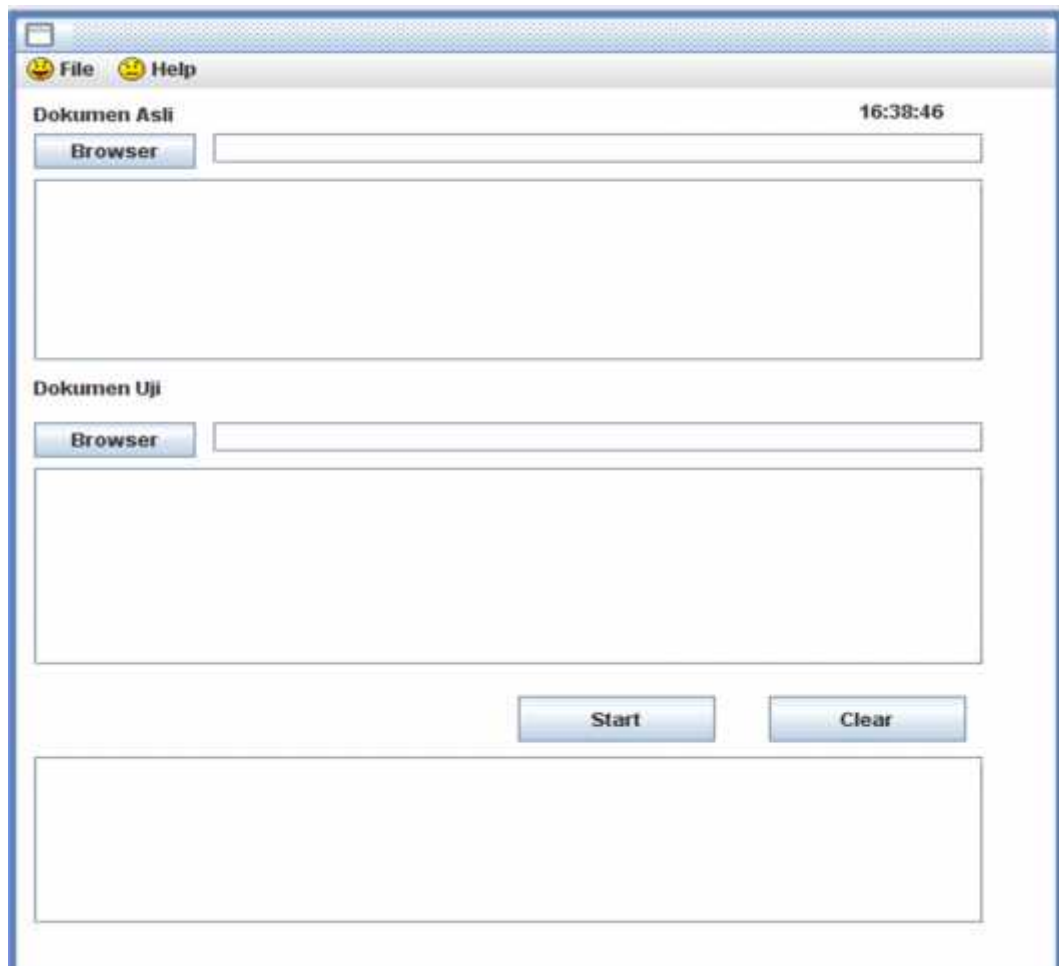


Gambar IV.1. Tampilan JFrame Utama

2. Tampilan JFrame Masuk

Pada JFrame Masuk ini berfungsi untuk masuk ke program utama.

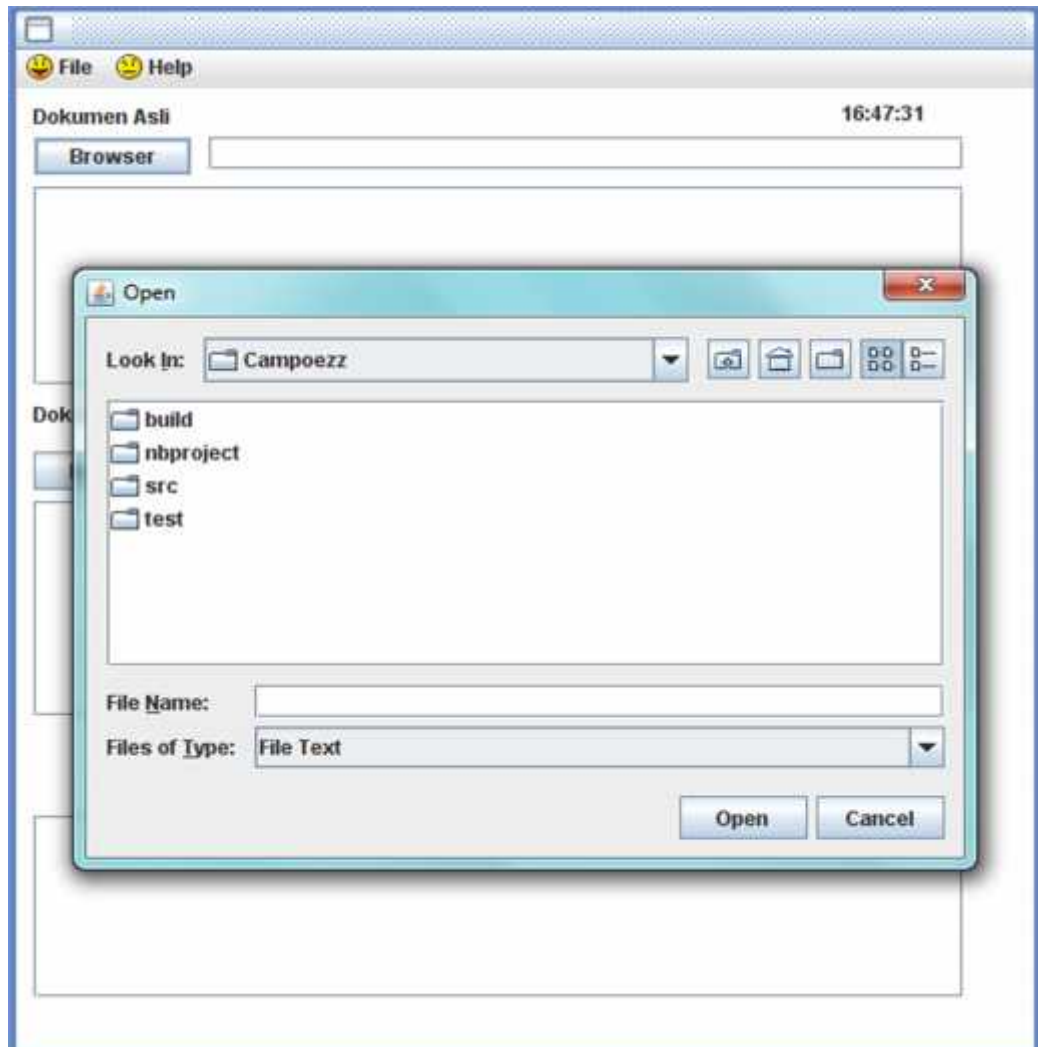
Gambar hasil dari JFrame Masuk seperti gambar IV.2.



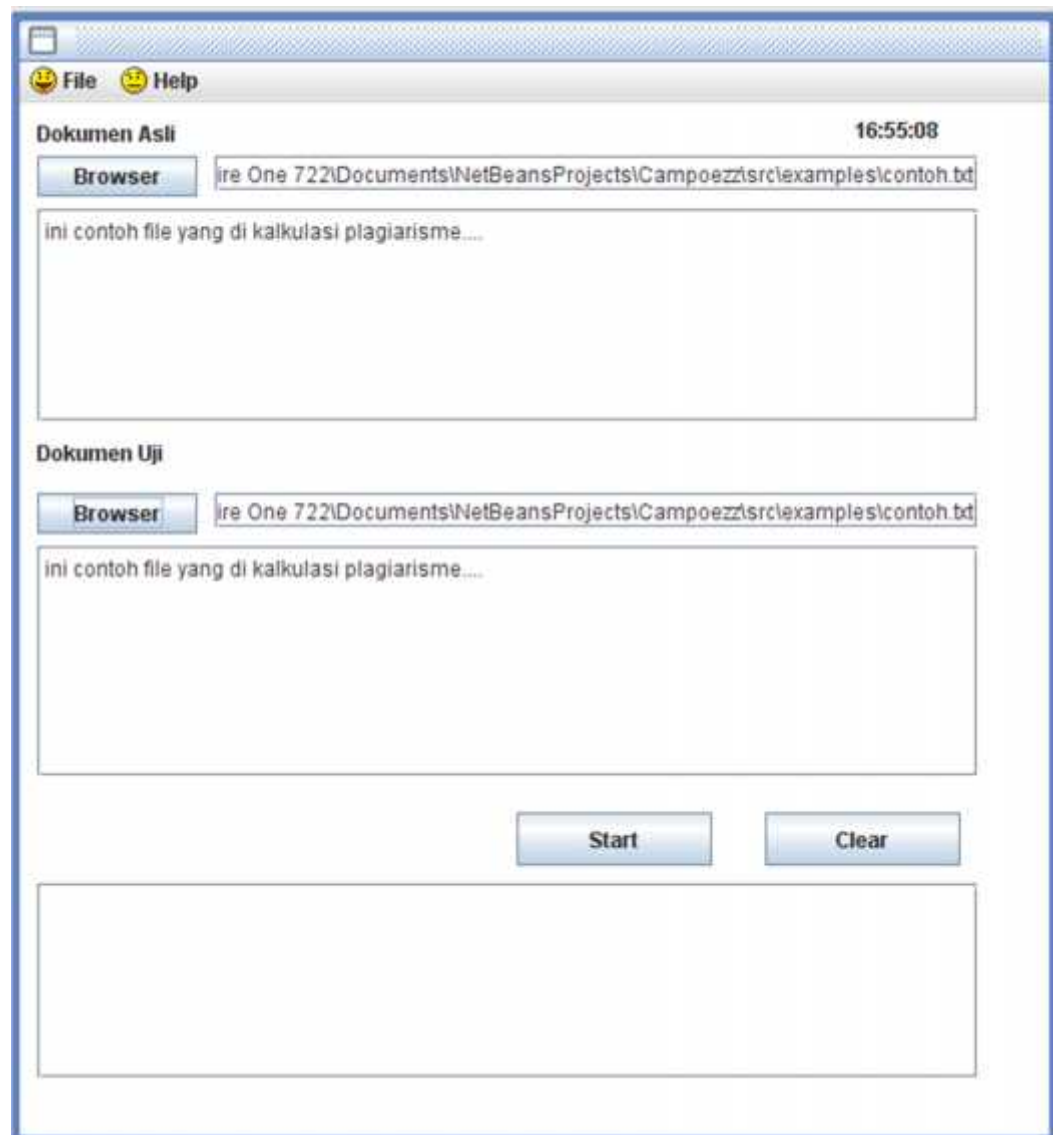
Gambar IV.2. Tampilan JFrame Masuk

3. Tampilan JFrame Browser

Pada JFrame Browser ini berfungsi untuk mengambil file yang berformat *.txt untuk pengujian kalkulasi *plagiarisme*. Gambar dari JFrame Browser seperti gambar IV.3 dan gambar IV.4.



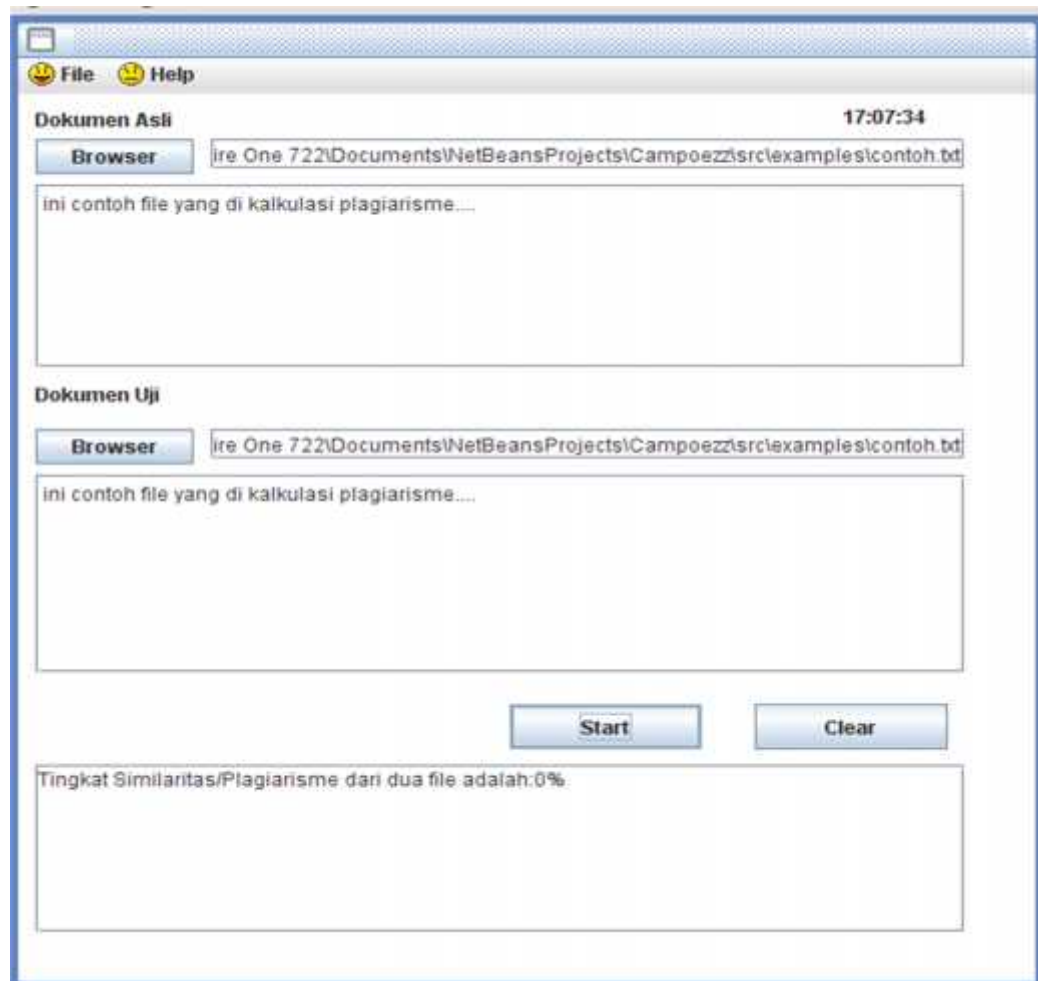
Gambar IV.3. Tampilan JFrame *Browser*



Gambar IV.4. Tampilan JFrame Hasil *Browser*

4. Tampilan JFrame *Start Plagiarisme*

Pada JFrame *Start Plagiarisme* ini berfungsi untuk menghitung persamaan teks 1 dan teks 2. Gambar hasil dari JFrame *Start Plagiarisme* seperti pada gambar IV.5.



Gambar IV.5. Tampilan *Start Plagiarisme*

5. Tampilan `JInternalFrame` About

Pada `JInternalFrame` about berfungsi untuk memberikan informasi tentang penulis. Gambar hasil dari `JInternalFrame` about seperti pada gambar IV.6.



Gambar IV.6. Tampilan JInternalFrame About

IV.2. Pembahasan

IV.2.1. Tampilan JFrame Utama

Tampilan JFrame utama adalah tampilan jframe yang terdiri dari berbagai button. Button dari JFrame utama adalah masuk, dan keluar. JFrame ini berfungsi sebagai halaman utama dari program aplikasi *Plagiarisme*.

IV.2.2. Implementasi

Proses implementasi yang dilakukan bertujuan untuk merancang sebuah aplikasi pendeteksi *plagiarisme* menggunakan algoritma Rabin – Karp dengan proses bantuan *pre – processing*.

Aplikasi yang dihasilkan dari proses implementasi pada pembuatan skripsi ini merupakan sebuah aplikasi system. Dikatakan aplikasi system karena aplikasi yang dibangun merupakan jaringan kerja yang terdiri dari prosedur – prosedur yang saling berhubungan satu sama lain untuk melakukan suatu usaha atau sasaran tertentu, yaitu mendeteksi *plagiarisme*. Pada umumnya di dalam sebuah system terdapat fungsi –fungsi teknologi seperti *input, process, output, storage* dan *communication*. Selain *hardware, software*, informasi dan prosedur – prosedur dalam bentuk program, factor *brainware* sebagai salah satu esensi system juga dibutuhkan. Sistem pedeteksi *plagiarisme* yang dirancang dalam penulisan skripsi ini mengadopsi penjelasan tersebut dengan mengikut – sertakan pengguna sebagai *user, operator* dan *decision maker* terhadap bobot/nilai hasil keluaran system dalam menentukan pasangan dokumen apakah termasuk hasil tindakan plagiat atau tidak. Hal ini menjelaskan bahwa bobot yang dihasilkan oleh system pendeteksi *plagiarisme* ini merupakan nilai relatif.

IV.2.3. Lingkungan Implementasi

Lingkungan implementasi meliputi lingkungan perangkat keras dan perangkat lunak.

IV.2.3.1. Lingkungan Perangkat Keras

Dalam perancangan dan pengembangan sistem deteksi antiplagiarisme ini menggunakan komputer (PC) dengan spesifikasi:

1. Prosesor AMD Dual-Core Processor C60 to 1.333 GH
2. Memory 2 GB
3. Harddisk 320 GB
4. Monitor 11”

IV.2.3.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan system deteksi anti-plagiarisme ini adalah:

1. Operating Sistem Microsoft Windows 7
2. JDK 6, mencakup Java versi 1.6.0 dan Java™ SE Runtime Environment
3. Java Editor Netbeans 6.0.1
4. Text Editor Notepad

IV.2.4. Batasan Implementasi

Batasan implementasi yang didefinisikan untuk implementasi system pendeteksi ini adalah sebagai berikut:

1. Mengasumsikan bahwa pengguna atau user hanya memasukan data berupa dokumen berbahasa Indonesia.
2. Hanya dapat membandingkan dua dokumen dalam sekali proses pengidentifikasi *plagiarisme*.

3. Hanya memproses file dokumen dengan format *.txt(tekstual), sehingga tidak membandingkan gambar, tabel atau objek semacamnya.
4. Mengansumsikan bahwa dokumen merupakan kalimat, tugas dan makalah, maka system menganggap penulisan dokumen telah sesuai dengan aturan – aturan penulisan dokumen berbahasa Indonesia secara resmi.

IV.2.5. Analisa Hasil dan Uji Coba

Adapun analisa hasil dan uji coba aplikasi program pendeteksi *plagiarisme* ada sebagai berikut :

IV.2.5.1. Menjalankan Program

Cara menjalankan program aplikasi pendeteksi *plagiarisme* adalah:

1. Buka aplikasi netbeans 6.0
2. Setelah itu buka program aplikasi pendeteksi *plagiarisme*.
3. Kemudian klik kanan pada nama file pilih run, untuk menjalankan program aplikasi pendeteksi *plagiarisme*.

IV.2.5.2. Interaksi Program dengan User

Interaksi program dengan user dapat dilakukan dengan cara sebagai berikut:

1. *User* terlebih dahulu menjalankan aplikasi program agar *user* dapat menggunakan program aplikasi pendeteksi *plagiarisme*.
2. Setelah program dijalankan, *user* harus mengklik tombol masuk.
3. Setelah masuk, untuk mengetes pendeteksian *plagiarisme file* teks. *User* harus membuka browser dan browser.

4. Setelah file teks tampil, untuk menjalankan hasil tes pendeteksian *plagiarisme*, user dapat mengklik tombol *start*.
5. Setelah mengklik tombol *start*, maka akan tampil tingkat similaritas dari dua *file*.

IV.2.5.3. Analisa Hasil dan Uji Coba Sistem

IV.2.5.3.1 Hasil Uji Coba Nilai *Similarity* dan Waktu Proses

Pada uji coba ini terdapat 4 dokumen uji yang masing-masing mempunyai 8 dokumen latih menggunakan *kgram* 1 sampai 5 dengan algoritma Rabin-Karp dan Rabin-Karp yang telah dimodifikasi serta menggunakan *stemming* dan *non-stemming*. Tabel IV.1. adalah contoh hasil uji coba. Untuk data hasil percobaan selengkapnya terdapat pada lampiran.

No	Kode Dokumen	Dokumen latih	Size (bytes)	Rabin - Karp	
				T(s)	S(%)
1	A	100% sama	2147	0.56	100.00
2		80% sama	2067	0.45	78.09
3		50% sama	765	0.53	53.00
4	B	70% sama	1834	0.45	72.00
5		60% sama	1232	0.43	59.00
6		70 % sama	1843	0.24	74.98
7		40 % sama	567	0.45	39.00

Tabel IV.1. Hasil Pengujian Akurasi Similaritas atau Tingkat Kesamaan dengan Algoritma Rabin - Karp

IV.2.5.4 Analisa Perhitungan kalkulasi Plagiarisme

Contoh Kasus :

A : Saya pergi kepasar semalam | string = 23 Karakter

B : Saya pergi kepasar tadi | string = 20 Karakter

Kalimat A = di split (dipisah) dengan delimiter (penanda) {spasi} Kalimat A Char

Array

A[]

A[0] = Saya

B[0] = Saya

A[1] = pergi

B[1] = pergi

A[2] = kepasar

B[2] = kepasar

A[3] = semalam

B[3] = tadi

“For(i=0;1<A.lenght;1++)”

Jumlah karakter kalimat 1 = 26 karakter dan beda dengan karakter kalimat 2 = 3

$$26 - 3 = 23$$

$$23/26 * 100\% = 88.46\% \text{ Kalkulasi Plagiarismenya.}$$

IV.3. Kelebihan dan Kekurangan Sistem Yang Dibuat

IV.3.1. Kelebihan Sistem

- a. Aplikasi pendeteksi *plagiarisme* ini tidak memiliki keterbatasan karakter huruf. Karena pendeteksi *plagiarisme* ini dilengkapi scroll.
- b. Aplikasi pendeteksi *plagiarisme* ini desain tampilannya simple karena aplikasi ini menggunakan 2 tampilan JFrame dan JInternalframe.

- c. Netbeans yang digunakan untuk membuat program aplikasi pendeteksian *plagiarisme* yaitu menggunakan Netbeans 6.0 yang dapat memudahkan *user* untuk merancang dan mendesain tampilan sesuai dengan yang diinginkan.
- d. Aplikasi pendeteksi *plagiarisme* ini apabila *file* yang mau dideteksi kosong, maka aplikasi akan memberikan pesan, kalau salah satu masing kosong.
- e. System yang lebih *user friendly* dengan menerapkan bentuk GUI pada antarmuka system.

IV.3.2. Kekurangan Sistem

- a. File yang digunakan untuk mengujian pendeteksi *plagiarisme* hanya *file* *.txt.
- b. Tingkat keakurasiannya aplikasi pendeteksi *plagiarisme* ini belum sepenuhnya akurat, karena masih perlu adanya pengembangan pada aplikasi pendeteksi *plagiarisme* yang penulis rancang
- c. Untuk menjalankan aplikasi pendeteksian *plagiarisme* ini membutuhkan memory yang cukup, karena aplikasi ini menggunakan IDE Netbeans.
- d. Aplikasi pendeteksian *plagiarisme* ini untuk penilaian kalkulasi berdasarkan dari pencocokan teks (*matching text*)