

**LAPORAN PENELITIAN
PENGEMBANGAN FAKULTAS DAN KEILMUAN
DANA BOPTN TAHUN ANGGARAN 2012**



**APLIKASI PENDETEKSIAN PLAGIAT PADA KARYA ILMIAH
MENGUNAKAN ALGORITMA RABIN-KARP**

Oleh :

**Dian Novian, S.Kom, MT (Peneliti Utama)
Tajuddin Abdillah, S.Kom, M.Cs (Anggota)
Moh.Syafri Tuloli, ST, MT (Anggota)
Rochmad M Thohir Yassin, S.Kom, M.Eng (Anggota)**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI GORONTALO**

2012

DAFTAR ISI

Abstrak	1
BAB I PENDAHULUAN	2
1.1 Latar Belakang	2
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
BAB II STUDI PUSTAKA	4
2.1 Deskripsi Teori	4
2.1.1 Teknik Plagiat	4
2.1.2 Algoritma Rabin-Karpp	4
2.2 Kerangka Berpikir.....	5
BAB III METODE PENELITIAN	7
3.1 Langkah-langkah penelitian.....	7
3.2 Metode Penelitian	9
3.2.1 Model rancangan eksperimen untuk menguji produk rancangan .	9
3.2.2 Populasi dan sampel.....	10
3.2.3 Teknik pengumpulan data.....	10
3.2.4 Instrumen penelitian	10
3.2.5 Teknik analisis data	10
BAB IV HASIL DAN PEMBAHASAN	11
4.1 Desain awal produk	11
4.1.1 Analisa kelemahan sistem.....	11
4.1.2 Analisa kebutuhan system	12
4.1.3 Perancangan Arsitektur Sistem.....	13
4.1.4 Perancangan Antarmuka	13
4.2 Hasil pengujian tahap pertama	16
4.2.1 Pengujian Penginputan dan Penghapusan Berkas di Database...	16
4.2.2 Pengujian Pendeteksian plagiat pada berkas	17
4.3 Revisi produk tahap pertama	18
4.4 Hasil pengujian tahap kedua.....	19
4.4.1 Pengujian metode pengkonversian representasi data.....	19
4.4.2 Sistem test menggunakan sample data real.....	20
4.5 Penyempurnaan produk	20
4.6 Pembahasan produk.....	21
4.6.1 Model Pendeteksian	21
4.6.2 Implementasi.....	23
4.6.2.1 Basis Data	23
4.6.2.2 Aplikasi Plagiat Detector	24
4.6.2.3.Sistem Pelaporan	35
BAB V KESIMPULAN DAN SARAN	36
5.1. Kesimpulan	36
5.2. Saran	36
DAFTAR PUSTAKA	37

DAFTAR GAMBAR

Gambar 3.1 Model <i>prototyping</i>	7
Gambar 3.2 <i>System prototype</i> Input	8
Gambar 3.3 <i>System prototype</i> Output	8
Gambar 4.1 Aplikasi pembantu pendeteksian secara visual.	12
Gambar 4.2 Arsitektur Sistem Deteksi Plagiat.....	13
Gambar 4.3 Disain antarmuka form utama	14
Gambar 4.4 Disain antarmuka form setting	14
Gambar 4.5 Disain antarmuka form database	14
Gambar 4.6 Disain antarmuka halaman utama pelaporan hasil pendeteksian	15
Gambar 4.7 Disain antarmuka pelaporan hasil pendeteksian	15
Gambar 4.8 Algoritma Utama Metode Pendeteksian.....	21
Gambar 4.9 Algoritma Rabin-Karp yang dimodifikasi.....	22
Gambar 4.10 perbandingan Algoritma Rabin-Karp.....	22
Gambar 4.11 Struktur Tabel Setting	23
Gambar 4.12 Struktur tabel Data.....	24
Gambar 4.13 Form awal bantuan instalasi	24
Gambar 4.14 Form penentuan folder tujuan instalasi	25
Gambar 4.15 Form penentuan grup pada Start Menu	25
Gambar 4.16 Form pembuatan Shortcut pada Desktop	26
Gambar 4.17 Form rangkuman settingan.....	26
Gambar 4.18 Form progress instalasi.....	27
Gambar 4.19 Form Microsoft Visual C++ Redistributable	28
Gambar 4.20 Form Progress Instalasi Microsoft Visual C++ Redistributable ...	28
Gambar 4.21 Form Instalasi Selesai.....	29
Gambar 4.22 Form Utama Aplikasi Pendeteksian Plagiat.....	30
Gambar 4.23 Form Database.....	30
Gambar 4.24 Form pilih berkas.....	31
Gambar 4.25 Form database yang telah berisi daftar berkas	31
Gambar 4.26 Memuat database pada form utama.....	32
Gambar 4.27 Berkas dari database ditampilkan pada form utama.....	33
Gambar 4.28 Memuat berkas yang akan diuji.....	33
Gambar 4.29 Daftar berkas yang akan diuji ditampilkan.....	34
Gambar 4.30 Form Setingan	34
Gambar 4.31 Hasil Pendeteksian ditampilkan	35
Gambar 4.32 Pelaporan Detail dari Proses Pendeteksian	35

DAFTAR TABEL

Tabel 3.1 Pengujian Sistem.....	9
Tabel 4.1 Tabel Penginputan dan penghapusan berkas di database.....	16
Tabel 4.2 Pengujian pendeteksian plagiat.....	17
Tabel 4.3 Pengujian metode pengkorvesian representasi data.....	19
Tabel 4.4 Testing menggunakan data tugas akhir.....	20

HALAMAN PENGESAHAN

1. Judul Penelitian : Aplikasi Pendeteksian Plagiat Pada Karya Ilmiah Menggunakan Algoritma Rabin-Karp
2. Ketua Peneliti :
 - a. Nama Lengkap : Dian Novian, S.Kom, MT
 - b. Jenis Kelamin : Laki-laki
 - c. NIP : 19751124 200112 1 001
 - d. Jabatan Struktural : -
 - e. Jabatan Fungsional : Lektor
 - f. Fakultas/Jurusan : Teknik/Teknik Informatika
 - g. Pusat Penelitian : Lembaga Penelitian Universitas Negeri Gorontalo
 - h. Alamat : Jl. Jenderal Sudirman No. 6 Kota Gorontalo
 - i. Telp/Faks : 0435-821183
 - j. Alamat Rumah : Jl. Taman Hiburan I, Perumahan Taman Indah Blok E No. 3, Kelurahan Wongkaditi Barat, Kecamatan Kota Utara, Kota Gorontalo.
 - k. Telp/Faks/Email : [081244097999/](tel:081244097999) aadian@ung.ac.id
3. Jangka Waktu Penelitian : 4 Bulan
4. Pembiayaan
Jmlh biaya diusulkan : Rp. 35.000.000,- (Tiga Puluh Lima Juta Rupiah)

Mengetahui,
Dekan Fakultas Teknik

Gorontalo, 20 November 2012

Ketua Peneliti

Ir. Rawiyah Husnan, MT
NIP. 19640427 199403 2 001

Dian Novian, S.Kom., MT
NIP. 19751124 200112 1 001

Menyetujui,
Ketua Lembaga Penelitian

DR. Fitryane Lihawa, M.Si
NIP. 19691209 199303 2001

Identitas Penelitian

1. Judul Usulan : Aplikasi Pendeteksian Plagiat Pada Karya Ilmiah Menggunakan Algoritma Rabin Karpp

2. Ketua Peneliti :

- a) Nama lengkap : Dian Novian, S.Kom, MT
- b) Bidang Keahlian : Teknik Komputer
- c) Jabatan Struktural : -
- d) Jabatan Fungsional : Lektor
- e) Unit Kerja : Fakultas Teknik
- f) Alamat Surat : Jl. Jenderal Sudirman No. 6 Kota Gorontalo
- g) Telpon/Faks : 0435-821183
- h) E-Mail : aadian@ung.ac.id

3. Anggota Peneliti :

No.	Nama dan Gelar Akademik	Bidang Keahlian	Instansi	Alokasi Waktu (jam/minggu)
1	Tajuddin Abdillah, S.Kom,M.Cs	Ilmu Komputer	Jurusan Teknik Informatika FT UNG	5
2	Moh. Syafri Tuloli, ST, MT	Teknik Informatika	Jurusan Teknik Informatika FT UNG	5
3	Rochmad M Thohir Yassin, S.Kom, M.Eng	Teknologi Informasi	Jurusan Teknik Informatika FT UNG	5

4. Objek Penelitian : Objek penelitian ini adalah Fakultas Teknik Universitas Negeri Gorontalo

5. Masa pelaksanaan penelitian :

- Mulai : Agustus 2012
- Berakhir : November 2012

6. Anggaran yang disusulkan : Rp. 35.000.000,-

7. Lokasi Penelitian : Fakultas Teknik Universitas Negeri Gorontalo

8. Hasil yang ditargetkan :

Hasil dari penelitian adalah terciptanya suatu aplikasi pendeteksian plagiat yang dapat diimplementasikan baik di lingkungan fakultas secara khusus maupun di lingkungan universitas secara umum.

ABSTRAK

Perkembangan internet semakin pesat hal ini berimbas kepada semakin banyaknya informasi yang tersedia. Hal ini memudahkan seseorang dalam melakukan penjiplakan suatu karya. Penjiplakan menurut Kamus Besar Bahasa Indonesia (KBBI) berarti menggambar atau menulis garis-garis gambaran atau tulisan yang telah tersedia (dengan menempelkan kertas kosong pada gambar atau tulisan yang akan ditiru), mencontoh atau meniru tulisan atau pekerjaan orang lain, mencuri karangan orang lain dan mengakui sebagai karangan sendiri, mengutip karangan orang lain tanpa seizin penulisnya. Penelitian ini bertujuan untuk mendesain aplikasi pendeteksian plagiat pada karya ilmiah menggunakan algoritma Rabin-Karpp.

Metode penelitian yang digunakan adalah penelitan dan pengembangan dan dalam pembangunan sistem menggunakan metode prototipe.

Aplikasi pendeteksian plagiat yang dihasilkan berbasis *desktop* sedangkan untuk sistem pelaporan berbasis web. Aplikasi yang dibangun berhasil mendeteksi kalimat yang sama antara berkas yang diuji dengan berkas yang ada pada *repository*. Kalimat yang sama antara berkas yang diuji dengan berkas yang ada pada *repository* ditandai dengan warna merah, sementara kalimat yang tidak identik ditandai dengan warna hitam.

Kata kunci : *Plagiat, algoritma Rabin-Karp, aplikasi berbasis dektop dan web*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pesatnya perkembangan internet menyebabkan semakin banyaknya informasi yang tersedia. Hal ini memudahkan seseorang dalam melakukan penjiplakan suatu karya. Penjiplakan menurut Kamus Besar Bahasa Indonesia (KBBI) berarti menggambar atau menulis garis-garis gambaran atau tulisan yang telah tersedia (dengan menempelkan kertas kosong pada gambar atau tulisan yang akan ditiru), mencontoh atau meniru tulisan atau pekerjaan orang lain, mencuri karangan orang lain dan mengakui sebagai karangan sendiri, mengutip karangan orang lain tanpa seizin penulisnya.

Maraknya kasus penjiplakan oleh golongan intelektual menjadi suatu tragedi dalam dunia pendidikan Indonesia seperti kasus profesor termuda bidang hubungan internasional yang diberhentikan secara tidak hormat di tahun 2010. Akibat banyaknya informasi tersedia secara *online* maka kebiasaan *copy-paste* tanpa menyebutkan referensi menjadi mudah dilakukan. Sehingga karya ilmiah yang dibuat menjadi hasil plagiat dari karya ilmiah lain. Namun dikarenakan sebagian besar karya ilmiah belum dilindungi Undang-Undang Hak atas Kekayaan Intelektual (HaKI) maka plagiarisme digolongkan sebagai kejahatan akademik yang termasuk sebagai pelanggaran etika dan sulit untuk dipidanakan. Sebagai langkah awal untuk mencegah kasus serupa diperlukan cara mendeteksi kemungkinan terjadinya penjiplakan seperti di lingkungan perguruan tinggi yaitu utamanya pada hasil tugas akhir calon sarjana S1 maupun tesis calon sarjana S2 dan disertasi calon sarjana S3 yang rawan penjiplakan.

Praktik plagiat tidaklah menjadi hal asing lagi, apalagi di kalangan mahasiswa yang hampir setiap hari mengerjakan tugas yang diberikan oleh dosen. Tak terkecuali pula, mahasiswa informatika sering mendapat tugas untuk membuat program aplikasi dengan menggunakan bahasa pemrograman tertentu. Dalam pengerjaan tugas tersebut, praktik plagiat tak terelakkan lagi untuk dilakukan mengingat waktu pengerjaan tugas yang terbatas dan tidak adanya motivasi untuk berusaha menyelesaikan tugas dengan kemampuan sendiri. Praktik plagiat dilakukan

dengan cara tukar-menukar kode program (*source code*) yang telah berhasil. Mahasiswa yang memplagiat dapat dengan mudah menyalin atau mengganti kode program yang telah didapatkan secara cepat dengan menggunakan fitur-fitur yang disediakan oleh komputer. Untuk mengatasi praktik plagiat, tidaklah cukup hanya mengingatkan kepada mahasiswa bahwa tindakan plagiat tidak baik dilakukan. Pendeteksian praktik plagiat merupakan solusi yang sebaiknya dilakukan sehingga tindakan curang tersebut dapat diminimalisasi.

1.2. Perumusan Masalah

Berdasarkan uraian di atas, Penelitian ini di fokuskan untuk membuat aplikasi yang mampu mendeteksi plagiat, yang permasalahannya diperoleh sebagai berikut:

- (1) Bagaimana mendesain dan mengembangkan aplikasi pendeteksian plagiat?
- (2) Bagaimana menentukan sebuah teks, kalimat, karya ilmiah atau jurnal termasuk dalam plagiat atau bukan?
- (3) Bagaimana menerapkan algoritma Rabin-Karpp yang akan digunakan dalam pengembangan aplikasi pendeteksian plagiat?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah

- (1) Mendesain dan mengembangkan aplikasi sistem anti plagiat.
- (2) Menentukan susunan teks, kalimat, karya ilmiah atau jurnal termasuk plagiat atau bukan
- (3) Menerapkan algoritma Rabin-Karp dalam pengembangan aplikasi pendeteksian plagiat.

1.4. Manfaat Penelitian

Manfaat dari penelitian ini adalah :

- (1) Terciptanya sebuah sistem yang dapat mendeteksi plagiat.
- (2) Tersedianya *repository* karya ilmiah pada Fakultas Teknik Universitas Negeri Gorontalo.

BAB II STUDI PUSTAKA

2.1. Deskripsi Teori

2.1.1. Teknik Plagiat

Plagiat adalah teknik penyalinan atau meniru karya orang lain yang diklaim menjadi hasil karya sendiri. Tidak adanya motivasi ataupun kemudahan dalam proses penyalinan dengan harapan tidak diketahui orang lain menjadi alasan utama terjadinya praktik plagiat. Beberapa jenis plagiat yang dikenal selama ini, yaitu:

- a. *Word-for-word plagiarism* : menyalin setiap kata secara langsung tanpa diubah sedikitpun
- b. *Plagiarism of the form of a source* : menyalin dan atau menulis ulang kode-kode program tanpa mengubah struktur dan jalannya program
- c. *Plagiarism of authorship*: mengakui hasil karya orang lain sebagai hasil karya sendiri dengan cara mencantumkan nama sendiri menggantikan nama pengarang sebenarnya

Beberapa contoh praktik plagiat pada sebuah program, yaitu:

- a) Leksikal: perubahan pada kode (*source code*) program, misalnya:
 1. Komentar diubah (ditambah, dikurangi, atau diganti)
 2. Format penulisan diubah
 3. Nama variabel diubah
- b) Struktural : perubahan struktur program
 1. Perubahan urutan algoritma yang tidak mengubah jalannya program
 2. Prosedur diubah menjadi fungsi atau sebaliknya
 3. Pemanggilan prosedur diganti dengan isi prosedur itu sendiri.

2.1.2. Algoritma Rabin-Karpp

Sebuah algoritma pencarian *string* sederhana yaitu *brute-force algorithm* yang kemudian berkembang menjadi beberapa algoritma diantaranya adalah algoritma Knuth-Morris Pratt (KMP), algoritma Boyer-Moore (BM) dan Algoritma Rabin dan Karpp. Menurut (Abdeen, 2011) pada prinsipnya algoritma Rabin-Karp menghitung sebuah fungsi *hash* untuk mencari pola didalam sebuah teks yang

diberikan. Setiap karakter M *subsequence* dari pada teks akan dikomparasi, jika nilai *hash* tidak sama algoritma akan menghitung nilai *hash* untuk karakter M *subsequence* berikutnya. Dan jika nilai *hash* sama maka algoritma akan melakukan perbandingan secara *brute-force* antara pola dan karakter M *subsequence*, dengan cara ini hanya akan ada satu perbandingan per teks *subsequence* dan *brute-force* hanya dibutuhkan jika nilai *hash* cocok atau sama. (Jain, et., al, 2012)

Menurut Arliandinda dalam (Mutiara, et al, 2008) terdapat empat kategori proses perbandingan yaitu :

- a) Dari kanan ke kiri
- b) Dari kiri ke kanan-
- c) Dalam order spesifik
- d) Dalam order apapun

Berdasarkan keempat kategori tersebut algoritma Rabin-Karp termasuk dalam kategori dari kiri ke kanan. Algoritma Rabin-Karp menerapkan fungsi *hash* yang menyediakan metode sederhana untuk mencegah kompleksitas waktu $O(m^2)$. Fungsi *hash* setidaknya harus menyediakan empat properti yaitu :

1. Mampu melakukan komputasi secara efisien
2. Diskriminasi string yang tinggi
3. Fungsi *hash* ($y[j+1 \dots j+m]$) harus mudah mengkomputasi dari :
 - $Hash(y[j \dots j+m-1])$
 - $Hash(y[j+m])$

Algoritma Rabin-Karp menandai langkah-langkah berikut ini :

- a) Menerapkan fungsi *hash*
- b) Fase preproses dalam kompleksitas waktu $O(m)$ dan waktu yang konstan
- c) Fase pencarian dalam kompleksitas waktu $O(m)$
- d) $O(n+m)$ mengestimasi waktu aktif

2.2. Kerangka Berpikir

Plagiat adalah masalah yang sering terjadi pada lingkungan pendidikan, hal ini difasilitasi dengan semakin mudahnya proses pengiriman informasi melalui berbagai media. Beberapa pendekatan telah digunakan untuk mendeteksi plagiat seperti menggunakan fingerprinting dari dokumen (Schelimer, et al, 2003) atau yang

lebih lanjut lagi menggunakan conceptual footprint yang menggunakan thesaurus dari suatu ide untuk mendeteksi plagiat (Dreher, 2007), atau yang menggunakan database daftar sinonim sendiri (Anzelmi, et al, 2011), juga pendeteksian berbasis citasi (Gipp, et al, 2011).

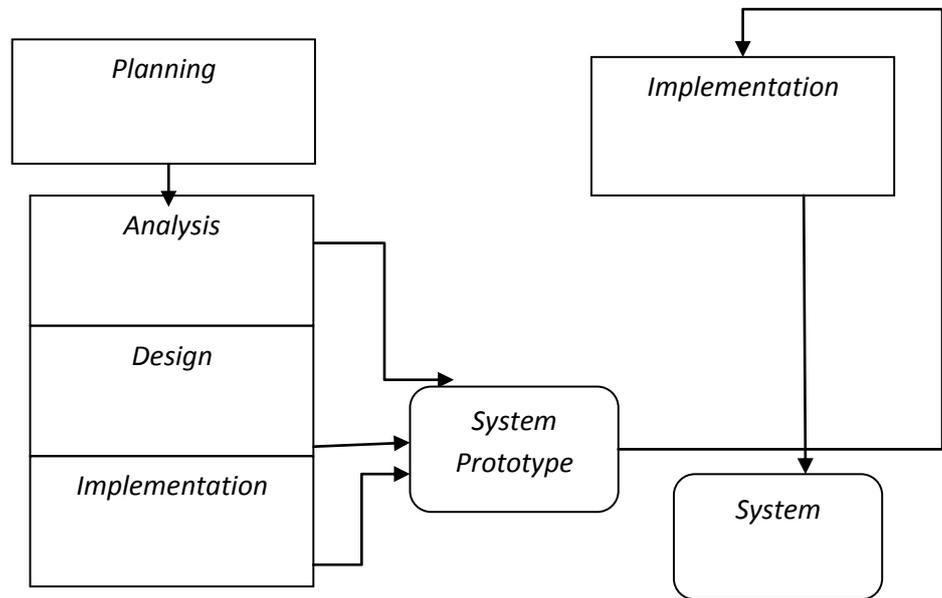
Beberapa sistem plagiat sudah berhasil dibuat, misalnya sistem untuk pendeteksian plagiarisme pada tugas pemrograman (Maharani, et al, 2011) aplikasi yang berbasis web seperti Moss dan JPlag (Lukashenko, 2007), juga aplikasi yang berbasis desktop seperti WCopyFind.

Sistem yang diusulkan akan berusaha mencoba memecahkan masalah plagiarisme pada lingkungan fakultas teknik dengan cara menggunakan suatu metode pendeteksian yang menggunakan algoritma pencocokan string (string matching) yaitu Rabin-Karpp, selain itu sistem yang akan dibangun menyediakan fungsi-fungsi lain yang dibutuhkan sesuai dengan hasil tahapan analisis kebutuhan.

BAB III METODE PENELITIAN

3.1. Langkah-langkah penelitian

Proses pembuatan perangkat lunak aplikasi pendeteksi plagiat menggunakan model *prototyping*, dimana tahapan-tahapan pada model *prototyping* yaitu *Planning*, *Analysis*, *Design*, *Implementation*, *System Prototype*, *Implementation* dan *System* seperti yang digambarkan pada Gambar 1. berikut ini :



Gambar 3.1 Model *prototyping*

Alasan digunakannya model *prototyping* karena dapat meminimalisir implementasi sistem yang belum sempurna atau belum stabil serta lebih cepat merespon kebutuhan pengguna. *System prototype* yang sudah dicoba dari sistem pendeteksi plagiat, dimana pada *system prototype* ini yang dibandingkan adalah sebuah dua buah kalimat. Semakin kecil nilai yang dikeluarkan oleh model matematis pada *system prototype* ini maka bisa dikatakan kalimat tersebut identik atau plagiat akan tetapi sebaliknya apabila angka yang dikeluarkan oleh *system prototype* tersebut besar maka kedua kalimat tersebut tidak identik atau bukan plagiat. Berikut adalah *system prototype* yang sangat sederhana tentang penggambaran aplikasi pendeteksi plagiat :

Aplikasi Pendekteksian Plagiat

Kalimat A	<input type="text" value="Sekarang Lagi bulan puasa"/>	Kalimat C	<input type="text" value="walaupun itali kalah tetap di hati"/>
Kalimat B	<input type="text" value="Sekarang lagi bulan puasa sebentar lagi lebaran"/>	Kalimat D	<input type="text" value="walaupun spanyol menang karena kebetulan"/>
<input type="button" value="Bandingkan"/> <input type="button" value="Clear"/>			

Perbandingan Kalimat A dengan Kalimat B
--dengan--
Hasil A dengan B : 0

Perbandingan Kalimat C dengan Kalimat D
--dengan--
Hasil C dengan D : 0

Gambar 3.2 *System prototype* Input

Pada Gambar 2 pengguna diminta memasukan empat buah kalimat, kalimat a akan dibandingkan kalimat b dan kalimat c akan dibandingkan dengan kalimat d, hasilnya seperti pada Gambar.3 berikut :

Aplikasi Pendekteksian Plagiat

Kalimat A	<input type="text"/>	Kalimat C	<input type="text"/>
Kalimat B	<input type="text"/>	Kalimat D	<input type="text"/>
<input type="button" value="Bandingkan"/> <input type="button" value="Clear"/>			

Perbandingan Kalimat A dengan Kalimat B
Sekarang Lagi bulan puasa --dengan-- Sekarang lagi bulan puasa sebentar lagi lebaran
Hasil A dengan B : 0.428571428571

Perbandingan Kalimat C dengan Kalimat D
walaupun itali kalah tetap di hati --dengan-- walaupun spanyol menang karena kebetulan
Hasil C dengan D : 0.586956521739

Gambar 3.3 *System prototype* Output

Sistem akan mengeluarkan angka dari perbandingan keduanya, apabila angka yang dihasilkan kecil atau mendekati nol maka bisa dikatakan kalimat tersebut terindikasi plagiat tapi tidak sebaliknya.

Berdasarkan *system prototype* tersebut, maka sistem pendeteksiian bukan hanya terbatas pada kalimat melainkan file, dimana file-file yang akan dibandingkan akan *dimuat* kedalam sistem dan sistem tersebut akan mendeteksi apakah file-file tersebut plagiat atau tidak. Nantinya di dalam penelitian ini akan dikembangkan lebih lanjut sistem yang lebih lengkap dan lebih banyak memiliki fungsi-fungsi untuk mendeteksi plagiat.

Adapun langkah-langkah penelitian yang akan dilakukan yaitu :

- 1) Studi kepustakaan, dimana untuk menemukan filosofis dan teori-teori baik mengenai algoritma Rabin-Karpp maupun penelitian-penelitian yang terkait dengan aplikasi pendeteksian plagiat, termasuk didalamnya mengidentifikasi masalah-masalah yang sering muncul dan cara penyelesaiannya.
- 2) Perancangan sistem dan arsitektur, langkah ini akan mencakup terkait dengan penerapan model *prototyping* melalui pendekatan *object oriented design*. Selain itu disertai dengan perancangan arsitektur sistem yang akan diterapkan
- 3) Pembuatan *system prototype* dan *system*, dimana untuk mengembangkan lebih lanjut *system prototype* awal yang telah dibuat selagi diimplementasikan secara parsial sehingga nantinya tercipta *system* yang utuh dan stabil.
- 4) Implementasi, perawatan dan sosialisasi sistem, pada tahapan akhir dilakukan implementasi yang sesungguhnya dalam arti diterapkan untuk menguji tugas, makalah, karya ilmiah, jurnal dan lainnya disertai dengan perawatan sistem serta sosialisasi penggunaan sistem tersebut di lingkungan Fakultas pada khususnya atau bahkan pada tingkatan Universitas.

3.2. Metode Penelitian

3.2.1. Model rancangan eksperimen untuk menguji produk rancangan

Pengujian yang akan dilakukan terhadap sistem sebagian besar melalui pengujian dengan metode BlackBox, yaitu dengan membuat suatu input tertentu dan memperhatikan output yang dihasilkan, apabila output yang dihasilkan sesuai dengan output yang seharusnya (menurut spesifikasi requirement sistem) maka sistem berhasil lulus pada item pengujian yang diujikan. Rencana pengujian pengujian yang akan dilakukan :

Tabel 3.1 Pengujian Sistem

No	Item Pengujian	Jenis Pengujian
1	Pengujian Penginputan dan Penghapusan Berkas di Database	BlackBox
2	Pengujian pendeteksian plagiat pada berkas yang memiliki nilai kesamaan 0%, 25% 50%, 75% dan 100%.	BlackBox
3	Pengujian metode pengkonversian representasi data	BlackBox
4	System test menggunakan sample data real yaitu menggunakan data tugas akhir jurusan teknik informatika.	BlackBox

3.2.2. Populasi dan sampel

Sampel data yang akan digunakan adalah berupa data tugas akhir atau skripsi dari alumni jurusan teknik informatika.

3.2.3. Teknik pengumpulan data

Prosedur pengumpulan data adalah dengan meminta kerja sama dengan tenaga administrasi jurusan untuk menyediakan dan untuk mengumpulkan data tugas akhir atau skripsi alumni.

3.2.4. Instrumen penelitian

Instrumen penelitian yang digunakan adalah lembar observasi atau pengamatan.

3.2.5. Teknik analisis data

Analisa data dilakukan dengan melihat struktur isi data dan format berkas data yang dikumpulkan, hal ini dilakukan untuk menentukan kebutuhan kemampuan program dalam menangani berkas.

BAB IV

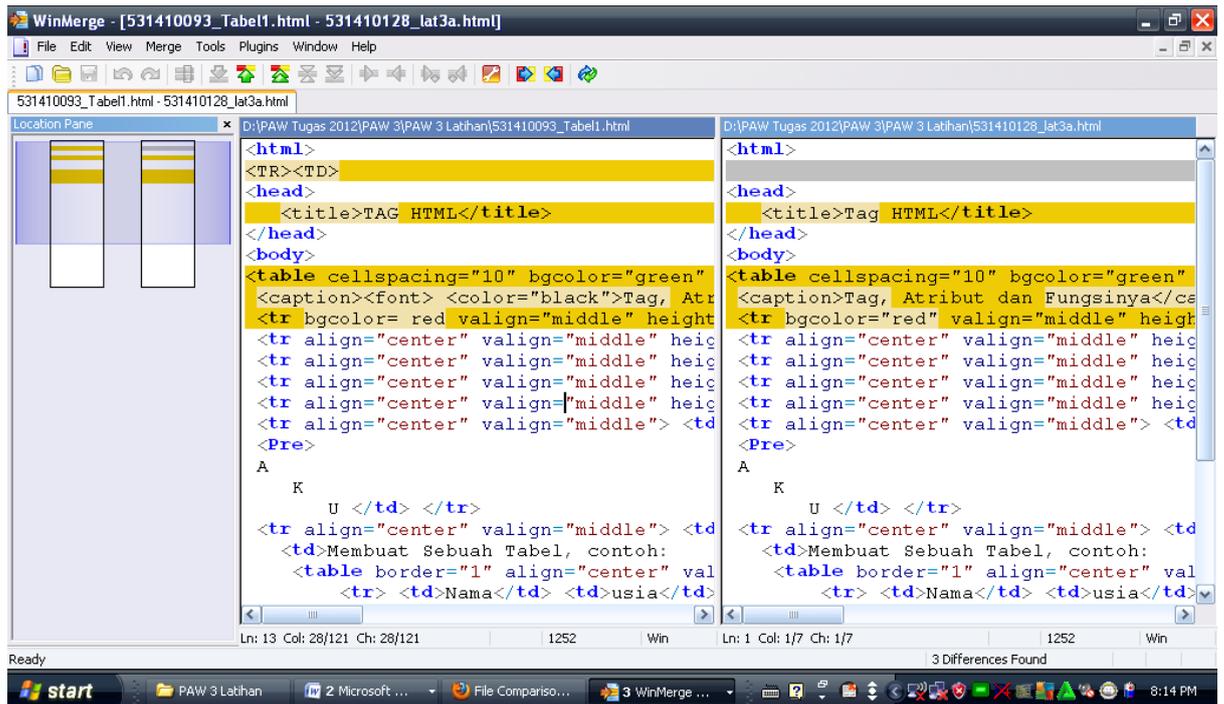
HASIL DAN PEMBAHASAN

4.1.Desain awal produk

4.1.1. Analisa kelemahan sistem

Dari hasil pengamatan terhadap penanganan plagiarism pada lingkungan penelitian diperoleh masalah yaitu:

- a. Seringnya terjadi plagiarism, baik dalam bentuk plagiarism secara semantic bahkan secara verbatim (kata-per-kata) dalam tugas perkuliahan, tugas akhir/skripsi yang lolos dari pendeteksian.
- b. Tidak tersedia suatu sistem repository tugas akhir/skripsi yang terpusat sehingga muncul kesulitan dalam melakukan melakukan pendeteksian plagiarism yaitu untuk membandingkan suatu tugas akhir/skripsi yang baru dengan yang sudah dibuat sebelumnya.
- c. Data yang tersimpan masih dalam format yang menggunakan memori yang besar, sehingga semakin lama bisa menyulitkan dalam pengumpulan, pengambilan, dan pembacaan data.
- d. Proses pendeteksian plagiarism masih mengandalkan tenaga manusia, sehingga sangat terbatas pada kemampuan memperhatikan dan mengingat untuk membandingkan dokumen. Beberapa alat bantu yang digunakan misalnya seperti aplikasi perbandingan secara visual (gambar dibawah). Pada alat tersebut dapat mendeteksi plagiat dengan membandingkan dua buah dokumen (terlihat pada jendela kiri dan kanan), baris berlatar putih termasuk plagiat, baris yang berwarna adalah baris yang tidak termasuk plagiat. Tool tersebut membantu dalam memperhatikan perbedaan dan persamaan pada suatu dokumen, tetapi belum cukup mengingat banyaknya tugas kuliah, tugas akhir/skripsi yang harus diperiksa.



Gambar 4.1 Aplikasi pembantu pendeteksian secara visual.

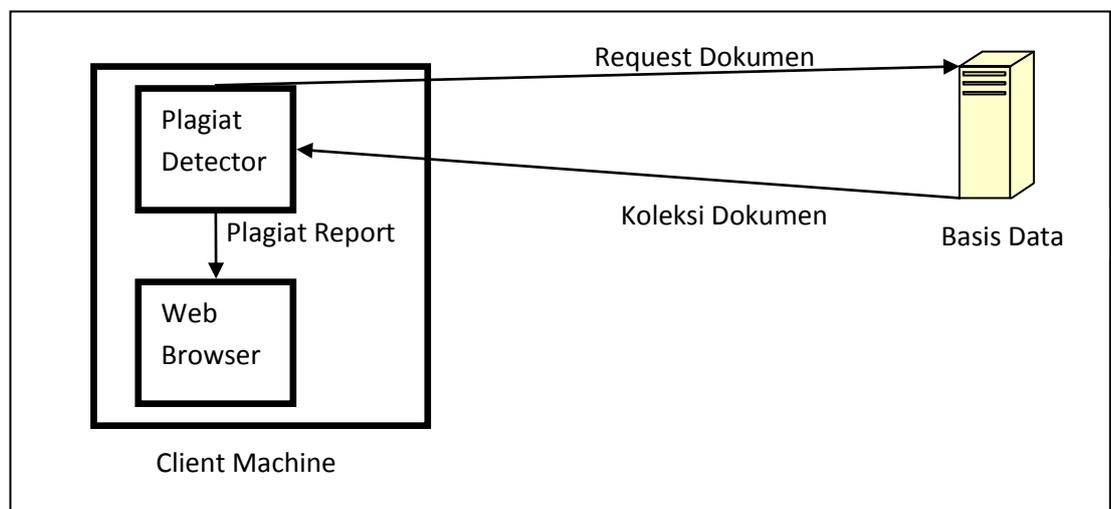
4.1.2. Analisa kebutuhan system

Dari hasil telaah terhadap kelemahan sistem yang ada, maka dapat disimpulkan bahwa dibutuhkan pembuatan sistem yang dapat mengatasi kekurangan-kekurangan sistem yang ada, yaitu sistem yang memiliki kemampuan:

- Sistem yang mampu mendeteksi terjadinya plagiarism terhadap tugas akhir/skripsi, mulai dari plagiarism secara verbatim, bahkan kalau dimungkinkan sampai pada plagiarism semantikal.
- Sistem memiliki fasilitas untuk melakukan penyimpanan terhadap data tugas akhir/skripsi supaya mempermudah dalam pengaksesan data-data tersebut.
- Sistem mengimplementasikan suatu metode representasi data yang dapat mengurangi penggunaan memori untuk penyimpanan data yang sangat banyak.
- Sistem melakukan tugas plagiarism tanpa memerlukan keterampilan khusus dari manusia (pengamatan, pengingatan atau penalaran), melainkan melakukan operasi pendeteksian secara otomatis menggunakan sumber daya pemrosesan komputer.

4.1.3. Perancangan Arsitektur Sistem

Arsitektur sistem yang digunakan adalah client-server yaitu sistem akan menggunakan server basis data yang akan menjadi repository dari dokumen-dokumen yang akan dijadikan bahan pembandingan untuk pendeteksian plagiarisme. Pada arsitektur ini, pada saat akan mendeteksi plagiat, aplikasi plagiat detector pada client akan mengirimkan permintaan dokumen pada server basis data. Setelah aplikasi mendapatkan koleksi dokumen, aplikasi akan melakukan proses pendeteksian dan membuat report plagiat. Report plagiat ini kemudian akan dikirimkan ke web browser untuk ditampilkan.



Gambar 4.2 Arsitektur Sistem Deteksi Plagiat

4.1.4. Perancangan Antarmuka

Dua jenis antarmuka dengan pengguna dari system adalah pada aplikasi plagiat detector dan pada pelaporan hasil pendeteksian :

- a. Antarmuka aplikasi plagiat detector

Antarmuka aplikasi plagiat detector menggunakan form dengan disain seperti dibawah ini:

Berkas Perbanding

Berkas Uji

Hasil Perbandingan

Tes

Setting

Database

Kosakata

Tutup

Gambar 4.3 Disain antarmuka form utama

Opsi-opsi pengaturan

<input checked="" type="checkbox"/>	Item pengaturan 1	<input checked="" type="checkbox"/>	Item pengaturan 4
<input checked="" type="checkbox"/>	Item pengaturan 2	<input checked="" type="checkbox"/>	Item pengaturan 5
<input checked="" type="checkbox"/>	Item pengaturan 3	<input checked="" type="checkbox"/>	Item pengaturan 6

Ok

Batal

Gambar 4.4 Disain antarmuka form setting

Berkas 1

Berkas 2

Berkas 3

Berkas 4

Jurusan

Informatika ▼

Jenis Berkas

Digital ▼

Tambah

Hapus

Tutup

Gambar 4.5 Disain antarmuka form database

b. Antarmuka pelaporan hasil pendeteksian

Antarmuka pelaporan menggunakan halaman web yang akan ditampilkan pada web browser.

Laporan Pendeteksian Plagiat			
Pendeteksian dilakukan dengan parameter:			
Parameter 1 : nilai			
Parameter 2 : nilai			
Parameter 3 : nilai			
Tingkat Plagiat	Berkas uji	Berkas Pembanding	Visual
80%	Berkas A	Berkas Database 1	Link
75%	Berkas A	Berkas Database 2	Link
25%	Berkas A	Berkas Database 3	Link

Gambar 4.6 Disain antarmuka halaman utama pelaporan hasil pendeteksian

Dokumen Kiri	Dokumen Kanan
Isi dokumen Kiri... bagian plagiat	Isi dokumen Kanan.. bagian plagiat

Gambar 4.7 Disain antarmuka pelaporan hasil pendeteksian

4.2. Hasil pengujian tahap pertama

4.2.1. Pengujian Penginputan dan Penghapusan Berkas di Database

Tabel 4.1 Tabel Penginputan dan penghapusan berkas di database

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-1-01	Pengujian Penginputan Berkas ke Database	1. Siapkan berkas yang akan dimasukkan ke dalam aplikasi 2. Jalankan aplikasi, lakukan operasi penginputan berkas ke dalam database	Notifikasi berkas berhasil masuk ke database.	Seluruh isi berkas berhasil dimasukkan ke dalam database.	Berkas berhasil dimasukkan seluruhnya ke dalam database.	Diterima
U-1-02	Pengujian penghapusan berkas dari database	1. Siapkan suatu berkas. 2. Inputkan berkas ke dalam database. 3. Jalankan operasi penghapusan berkas dari database.	Notifikasi database berhasil dihapus dari database.	Data berkas berhasil dihapus seluruhnya dari database.	Isi berkas berhasil dihapus dari database.	Diterima

4.2.2. Pengujian Pendeteksian plagiat pada berkas

Tabel 4.2 Pengujian pendeteksian plagiat

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-2-01	Pengujian pendeteksian plagiat pada berkas yang memiliki nilai kesamaan 0%	<ol style="list-style-type: none"> 1. Siapkan dua buah berkas yang akan dimasukkan ke dalam aplikasi, kedua berkas ini memiliki tingkat kesamaan 0%. 2. Jalankan aplikasi, lakukan operasi perbandingan pada kedua dokumen tersebut. 3. Lihat pelaporan hasil perbandingan kedua dokumen. 	Laporan hasil pendeteksian plagiat.	Laporan menunjukkan tidak ditemukan plagiat.	Aplikasi tidak mendeteksi adanya plagiat pada kedua dokumen.	Belum Diterima
U-2-02	Pengujian pendeteksian plagiat pada berkas yang memiliki nilai kesamaan 25%	<ol style="list-style-type: none"> 1. Siapkan dua buah berkas yang akan dimasukkan ke dalam aplikasi, kedua berkas ini memiliki tingkat kesamaan 25%. 2. Jalankan aplikasi, lakukan operasi perbandingan pada kedua dokumen tersebut. 3. Lihat pelaporan hasil perbandingan kedua dokumen. 	Laporan hasil pendeteksian plagiat.	Laporan menunjukkan bahwa kedua dokumen memiliki tingkat plagiat 25%.	Aplikasi mendeteksi tingkat plagiat 25% pada kedua dokumen.	Belum Diterima
U-2-03	Pengujian pendeteksian plagiat pada berkas yang memiliki nilai kesamaan 50%	<ol style="list-style-type: none"> 1. Siapkan dua buah berkas yang akan dimasukkan ke dalam aplikasi, kedua berkas ini memiliki tingkat kesamaan 50%. 2. Jalankan aplikasi, lakukan operasi perbandingan pada kedua dokumen tersebut. 3. Lihat pelaporan hasil perbandingan kedua dokumen. 	Laporan hasil pendeteksian plagiat.	Laporan menunjukkan bahwa kedua dokumen memiliki tingkat plagiat 50%.	Aplikasi mendeteksi tingkat plagiat 50% pada kedua dokumen.	Belum Diterima
U-2-04	Pengujian	1. Siapkan dua buah berkas yang akan	Laporan hasil	Laporan menunjukkan	Aplikasi mendeteksi	Belum Diterima

	pendeteksi plagiat pada berkas yang memiliki nilai kesamaan 75%	dimasukkan ke dalam aplikasi, kedua berkas ini memiliki tingkat kesamaan 75%. 2. Jalankan aplikasi, lakukan operasi perbandingan pada kedua dokumen tersebut. 3. Lihat pelaporan hasil perbandingan kedua dokumen.	pendeteksi plagiat.	n bahwa kedua dokumen memiliki tingkat plagiat 75%.	ksi tingkat plagiat 75% pada kedua dokumen.	a
U-2-05	Pengujian pendeteksi plagiat pada berkas yang memiliki nilai kesamaan 100%	1. Siapkan dua buah berkas yang akan dimasukkan ke dalam aplikasi, kedua berkas ini memiliki tingkat kesamaan 100%. 2. Jalankan aplikasi, lakukan operasi perbandingan pada kedua dokumen tersebut. 3. Lihat pelaporan hasil perbandingan kedua dokumen.	Laporan hasil pendeteksi plagiat.	Laporan menunjukkan bahwa kedua dokumen memiliki tingkat plagiat 100%.	Aplikasi mendeteksi tingkat plagiat 100% pada kedua dokumen.	Diterima

4.3.Revisi produk tahap pertama

Pada pengujian pada tahap pertama aplikasi pendeteksi plagiat belum dapat melakukan kompresi pada berkas yang diuji sehingga dilakukan perbaikan pada aplikasi yang nantinya aplikasi dapat melakukan kompresi pada berkas. Kompresi pada berkas dilakukan agar dapat meminimalisir proses pencarian (*string matching*) pada teks sehingga waktu uji berkas lebih cepat.

4.4. Hasil pengujian tahap kedua

4.4.1. Pengujian metode pengkonversian representasi data.

Tabel 4.3 Pengujian metode pengkonversian representasi data

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-3-01	Pengujian pengonversian data dari dokumen dengan format pdf	<ol style="list-style-type: none">1. Siapkan berkas yang akan dimasukkan ke dalam aplikasi, berkas menggunakan format pdf dengan ukuran berkas tertentu2. Jalankan aplikasi, lakukan operasi penyimpanan ke database.3. Lihat representasi dokumen dalam database.	Representasi berkas dalam database.	Ukuran representasi berkas dalam dokumen, menggunakan memori dalam jumlah yang lebih kecil dari ukuran asli berkas.	Ukuran representasi berkas terbukti lebih kecil dari ukuran asli berkas.	Diterima

4.4.2. Sistem test menggunakan sample data real.

Tabel 4.4 Testing menggunakan data tugas akhir

Identifikasi	Deskripsi	Prosedur Pengujian	Keluaran yang diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
U-4-01	Pengujian sistem menggunakan data sample data tugas akhir jurusan teknik informatika	<ol style="list-style-type: none">1. Siapkan berkas data yang akan digunakan.2. Jalankan aplikasi, lakukan operasi penyimpanan ke database.3. Periksa bahwa seluruh berkas sudah berhasil masuk dalam database.4. Lakukan operasi pendeteksian plagiat.5. Periksa laporan hasil pendeteksian.	Notifikasi berkas dimasukkan kedalam database, laporan hasil pendeteksian.	Semua berkas berhasil dimasukkan dalam database, dan sistem berhasil mendeteksi plagiat dalam data berkas.	Semua berkas berhasil dimasukkan, dan sistem berhasil mendeteksi adanya atau tidak adanya plagiat dalam berkas.	Diterima

4.5. Penyempurnaan produk

Sebagai penyempurnaan produk pada aplikasi pendeteksian plagiat ini ditambahkan sebuah sistem sistem pelaporan berbasis web sehingga memudahkan pengguna untuk melacak berkas yang memiliki kesamaan atau terdapat aktivitas plagiat atau tidak.

4.6. Pembahasan produk

4.6.1. Model Pendeteksian

Model pendeteksian yang akan digunakan pada sistem ini adalah menggunakan algoritma pencocokan string Rabin-Karp yang dimodifikasi.

```
public RobinCarp(String namaTeks, String namaPembanding, Hashtable<Integer,ArrayList<String>> teks,
Hashtable<Integer,ArrayList<String>> pembanding, int panjangKalimat){
//kalau pembanding lebih besar dari teks maka ditukar saja
if(pembanding.size()>teks.size()){
    this.namaTeks = namaPembanding;
    this.namaPembanding = namaTeks;
    dkTeks = pembanding;
    dkPembanding = teks;

}
else{
    //kalau kasus normal
    this.namaTeks = namaTeks;
    this.namaPembanding = namaPembanding;
    dkTeks = teks;
    dkPembanding = pembanding;
}

this.pjngKmat = panjangKalimat;

daftarKataPlagiatDiTeks = new Hashtable<Integer, Hashtable<Integer,Boolean>>();
daftarKataPlagiatDiPembanding = new Hashtable<Integer, Hashtable<Integer,Boolean>>();
for(int i=1; i<=dkTeks.size(); i++)
    daftarKataPlagiatDiTeks.put(i, new Hashtable<Integer, Boolean>());
for(int i=1; i<=dkPembanding.size(); i++)
    daftarKataPlagiatDiPembanding.put(i, new Hashtable<Integer, Boolean>());

//System.out.println("Buat tabel hash");
buatTabelHash();

buatTabelHashSeluruhPotonganTeksDanPembanding();

for(int i=1; i<=dkTeks.size(); i++){
    //System.out.println("Halaman " + i);
    for(int j=1; j<=dkPembanding.size();j++){
        //System.out.println("Mulai buat halaman " + i+" dikompare deng halaman "+j);
        detect(i,j);
    }
}

laporan();
}
```

Gambar 4.8 Algoritma Utama Metode Pendeteksian

```

public int detect(int halamanTeks, int halamanPembanding){
int n = dkTeks.get(halamanTeks).size();

int dkPembandingGetHalamanPembandingSize = dkPembanding.get(halamanPembanding).size();
Hashtable <Integer,Double> tabelHashPotonganPembandingGetHalamanPembanding =
tabelHashPotonganPembanding.get(halamanPembanding);

//jendela bergeser
for(int j=0; j<dkPembandingGetHalamanPembandingSize-pjngKmat+1; j++){

    double teksHash;
    final double pembHash = tabelHashPotonganPembandingGetHalamanPembanding.get(j);

    for(int i = 0; i< n-pjngKmat+1; i++){
        teksHash = tabelHashPotonganTeks.get(halamanTeks).get(i);

        if(teksHash == pembHash){

            //set flag bagi kata-kata di Teks
            for(int k=0; k<pjngKmat; k++){
                daftarKataPlagiatDiTeks.get(halamanTeks).put(new Integer(i+k), true);
            }
//                                set flag bagi kata-kata di Pembanding
            for(int k=0; k<pjngKmat; k++){
                daftarKataPlagiatDiPembanding.get(halamanPembanding).put(new
Integer(j+k), true);
            }

            //break;
        }
    }
}
return -1;
}

```

Gambar 4.9 Algoritma Rabin-Karp yang dimodifikasi

Untuk perbandingan, Algoritma Rabin-Karp yang asli adalah seperti dibawah:

```

function rabin_karp($text, $pattern)
{
    $n = strlen($text);
    $m = strlen($pattern);
    $text_hash = hash_string(substr($text, 0, $m), $m);
    $pattern_hash = hash_string($pattern, $m);

    for ($i = 0; $i < $n-$m+1; $i++) {
        if ($text_hash == $pattern_hash) {
            return $i;
        }
        $text_hash = hash_string(substr($text, $i, $m), $m);
    }
    return -1;
}

```

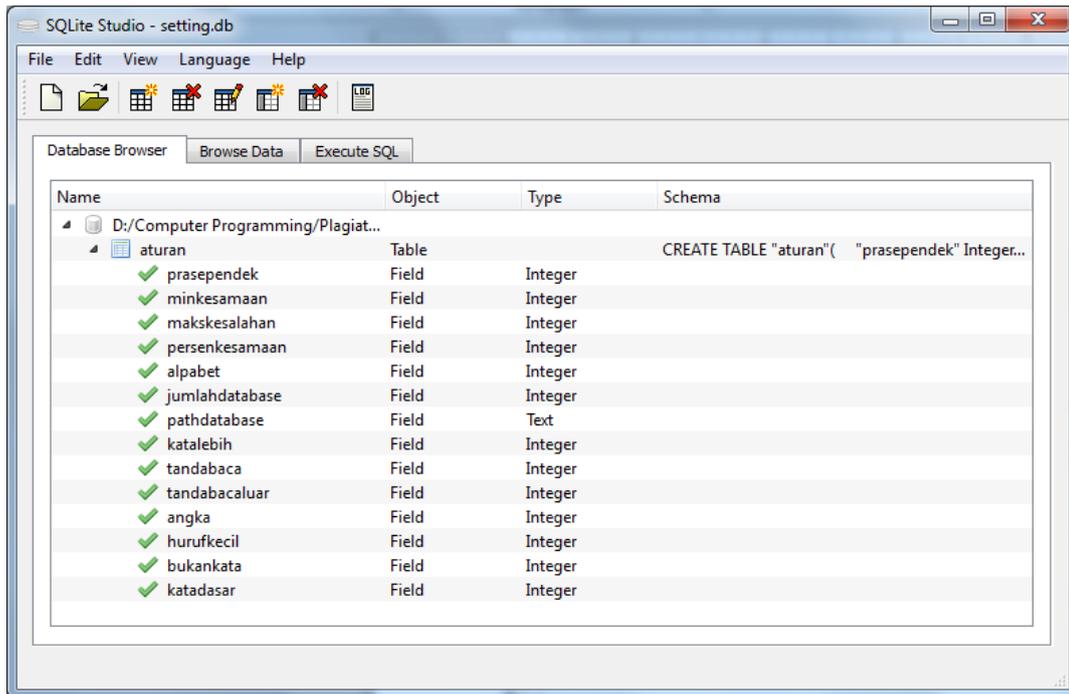
Gambar 4.10 perbandingan Algoritma Rabin-Karp

4.6.2. Implementasi

4.6.2.1. Basis Data

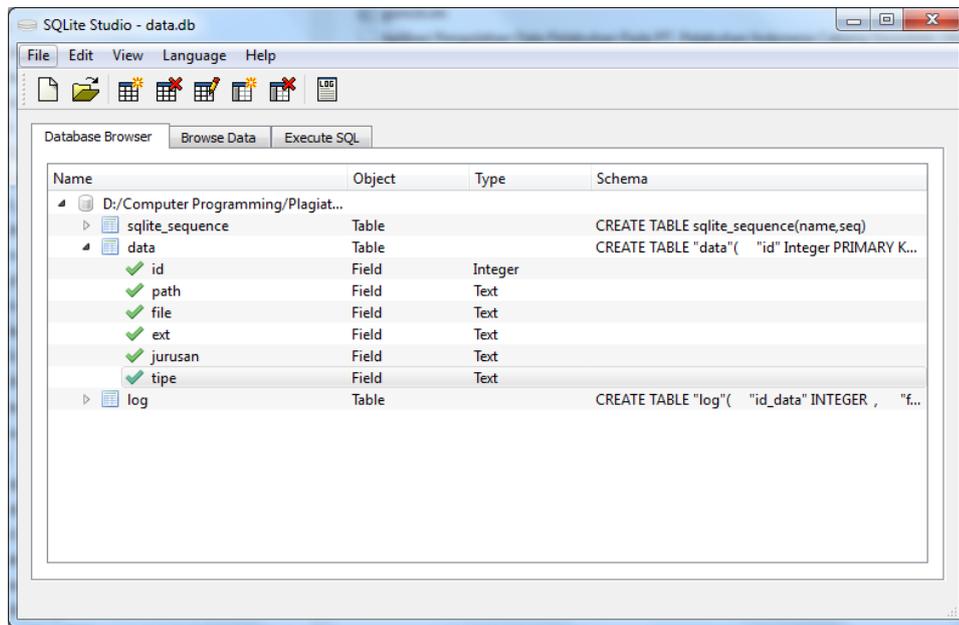
Server basis data menggunakan aplikasi database SQLite yaitu suatu database berbasis file. Adapun database yang digunakan terdiri dari dua buah tabel, yaitu:

1. Tabel Setting, yaitu tabel yang digunakan untuk menyimpan pengaturan terhadap proses pendeteksian.



Gambar 4.11 Struktur Tabel Setting

2. Tabel Data, yaitu tabel yang digunakan untuk menyimpan data dokumen-dokumen yang menggunakan format file yang berukuran kecil.



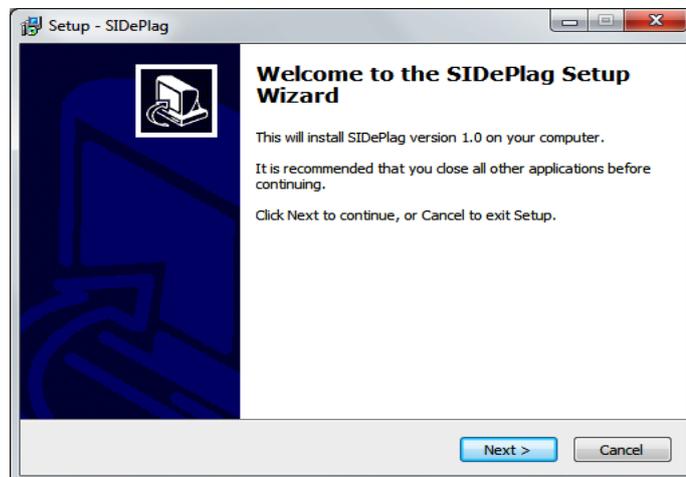
Gambar 4.12 Struktur tabel Data

4.6.2.2. Aplikasi Plagiat Detector

Aplikasi plagiat detector dikembangkan menggunakan tool pengembangan Microsoft Visual C++ yang menggunakan bahasa pemrograman C++. Prosedur penggunaan aplikasi:

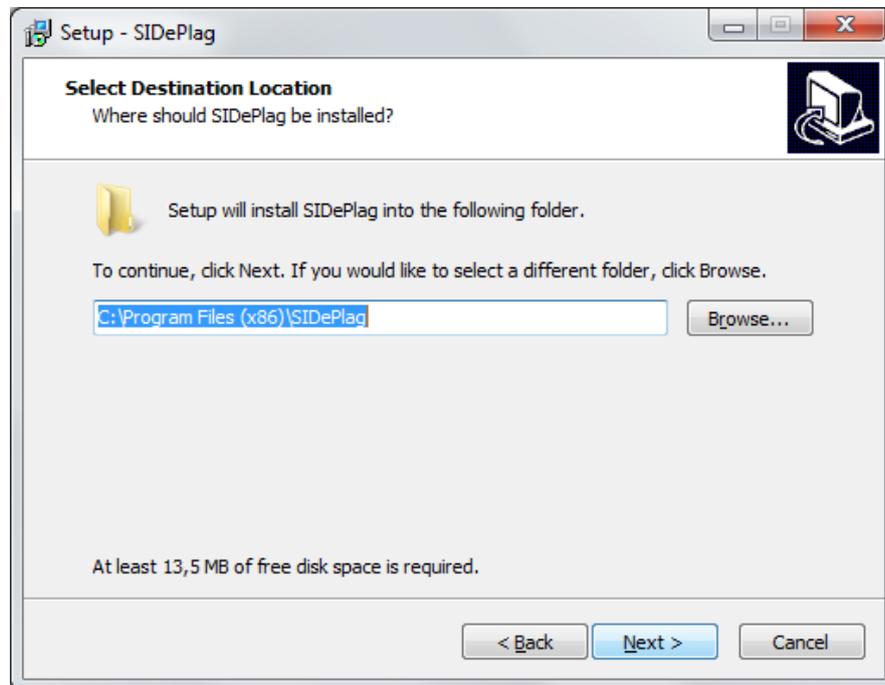
1. Instalasi

Pada saat program installer pertama kali dijalankan, akan menampilkan form awal dari bantuan instalasi (installation wizard)



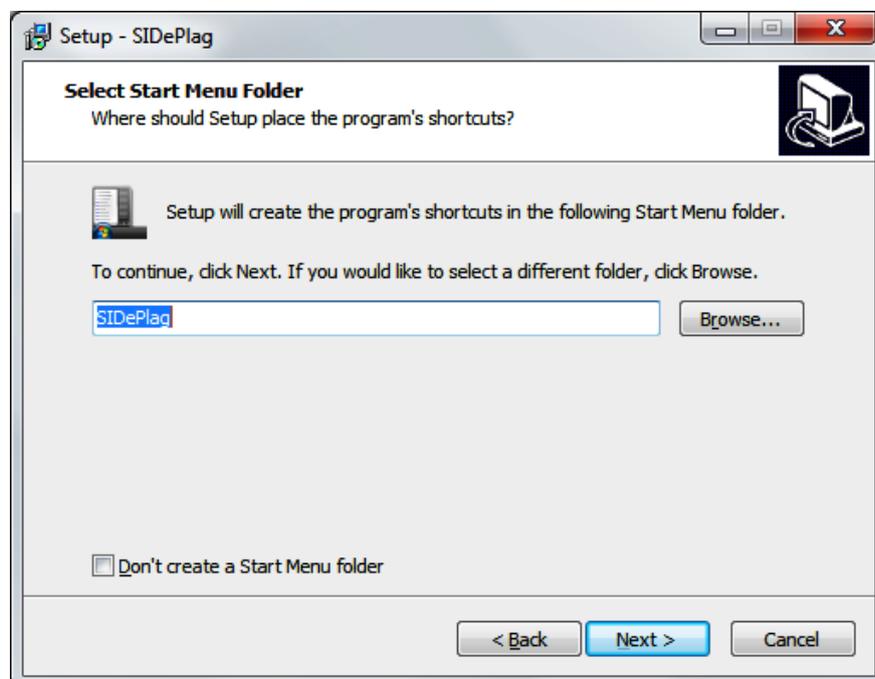
Gambar 4.13 Form awal bantuan instalasi

Selanjutnya program akan meminta dimasukkan lokasi tujuan instalasi, secara default program akan diinstall pada c:\program files\SIDePlag.



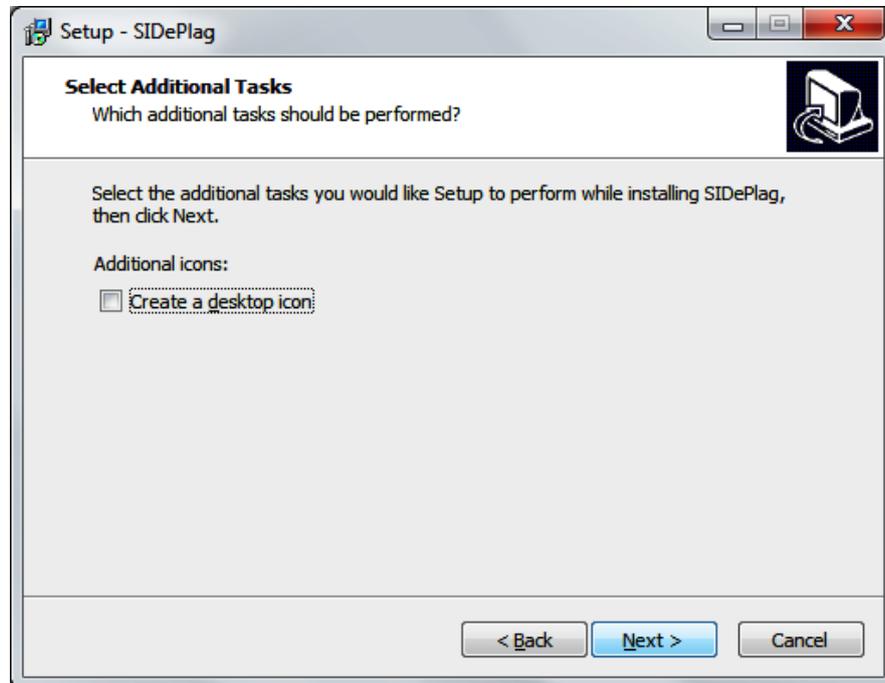
Gambar 4.14 Form penentuan folder tujuan instalasi

Selanjutnya program meminta nama group program yang akan diletakkan pada Start Menu, defaultnya program akan membuat group bernama SIDePlag.



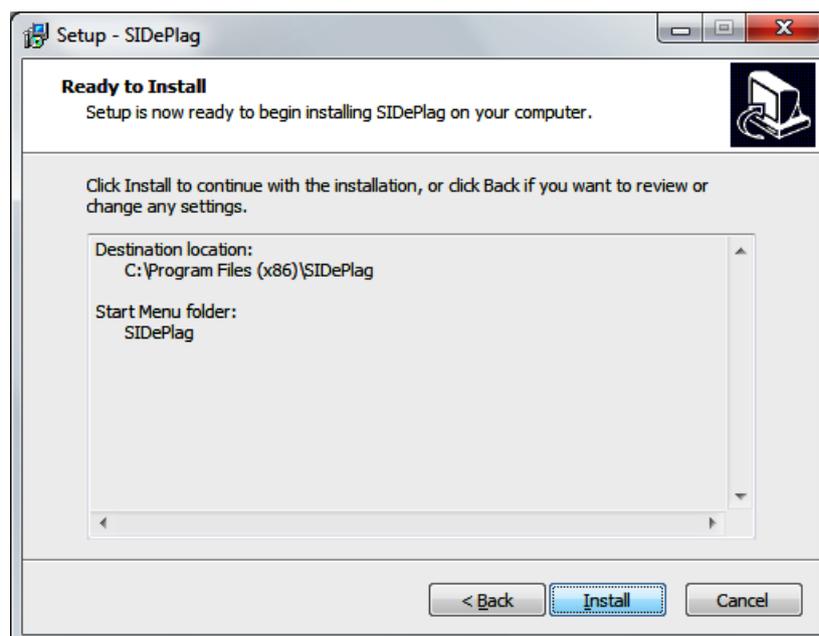
Gambar 4.15 Form penentuan grup pada Start Menu

Selanjutnya program akan meminta penentuan, pembuatan shortcut program pada Desktop, apabila pengguna menandai (centang) pilihan ini, maka aplikasi akan membuat shortcut pada desktop, secara default program pilihan ini tidak aktif.



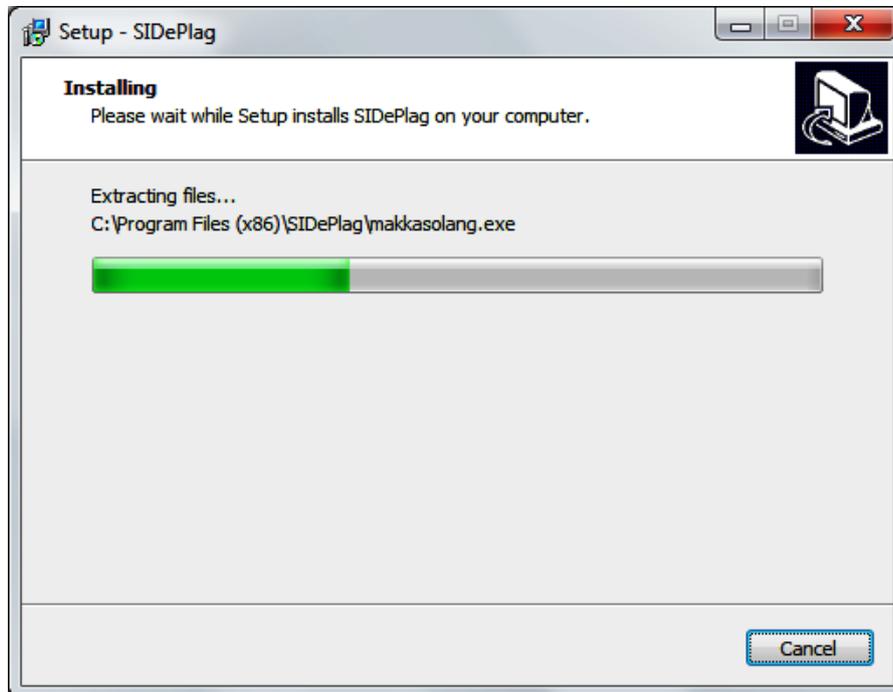
Gambar 4.16 Form pembuatan Shortcut pada Desktop

Selanjutnya program akan menampilkan rangkuman settingan yang akan digunakan untuk instalasi, apabila settingan tersebut telah sesuai pengguna dapat menekan tombol next sehingga program akan memulai melakukan instalasi pada folder tujuan.



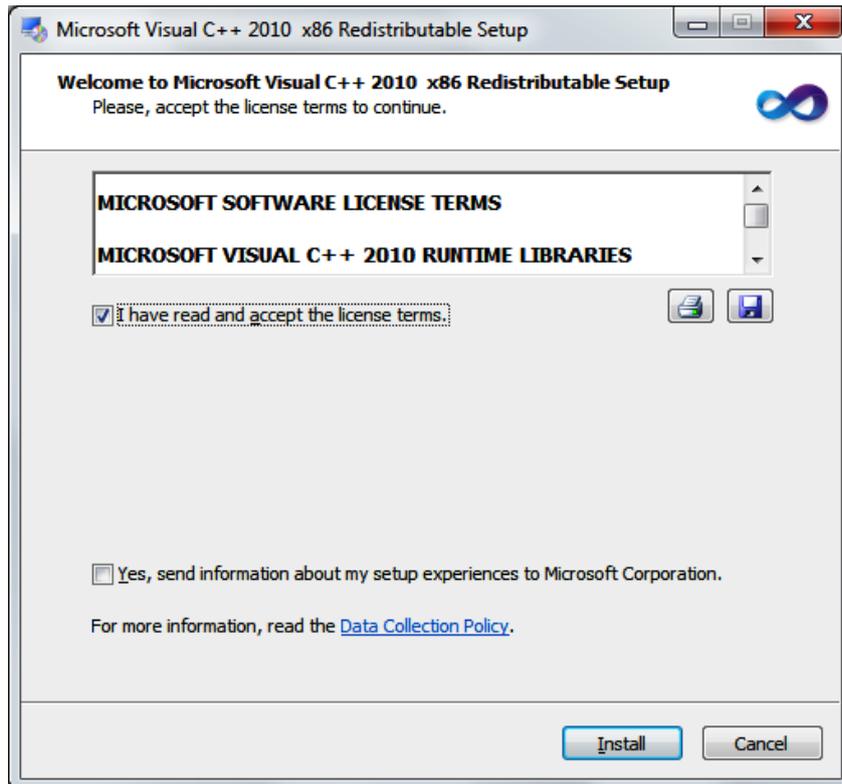
Gambar 4.17 Form rangkuman settingan

Pada form rangkuman setingan, apabila tombol Install ditekan, maka akan muncul form progress instalasi seperti dibawah. Pada form ini terlihat progress instalasi, apabila pengguna menekan tombol cancel, maka progress akan dihentikan.

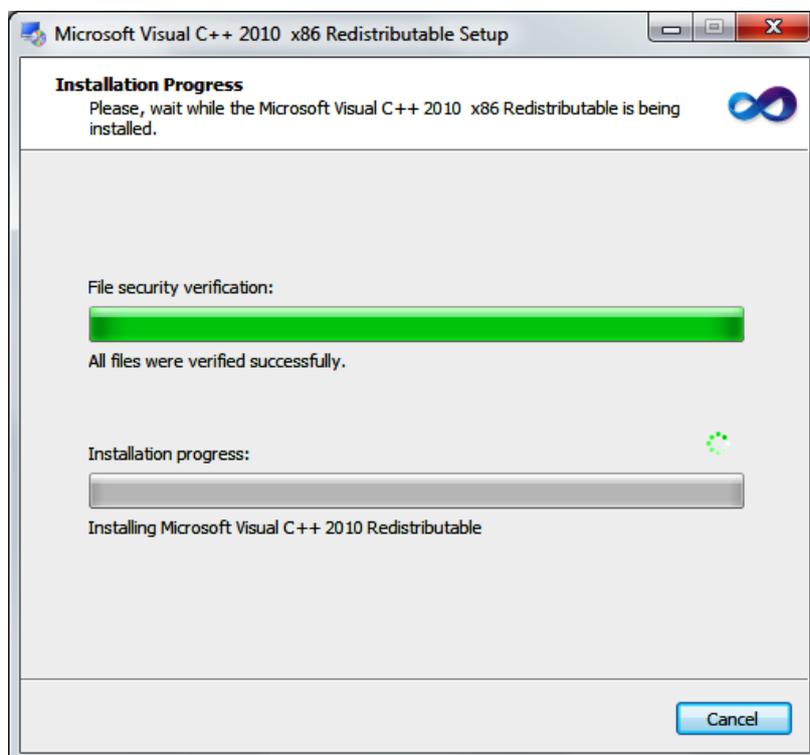


Gambar 4.18 Form progress instalasi

Program yang dikembangkan menggunakan Microsoft Visual C++ membutuhkan komponen library Visual C++ supaya bisa dijalankan, oleh karena itu komponen ini juga diinstallkan. Pada form dibawah, pengguna harus membaca lisensi dan manandai opsi **I have read and accept the license terms** untuk bisa melanjutkan.

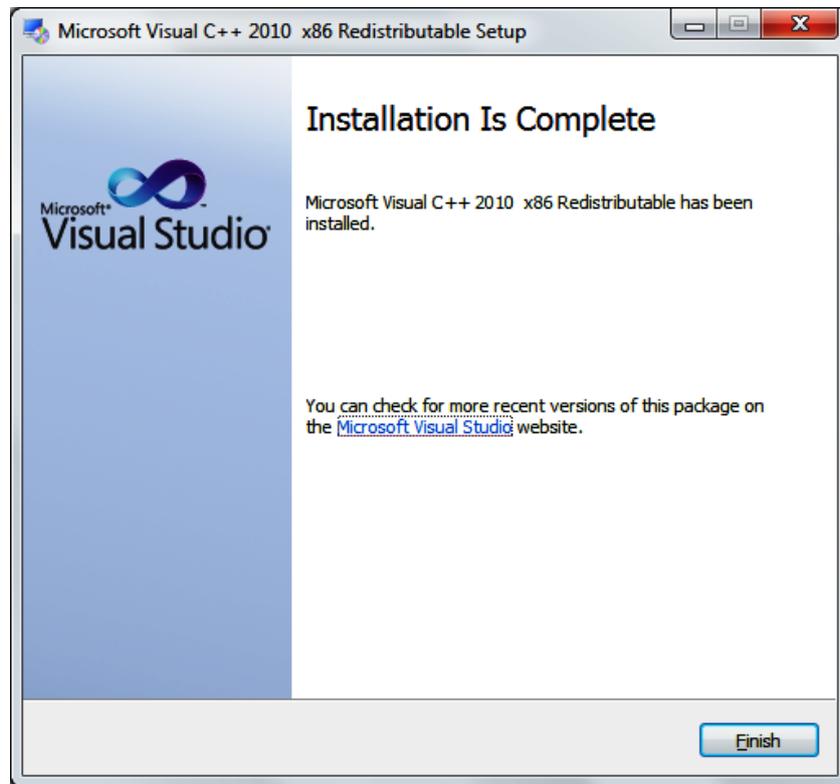


Gambar 4.19 Form Microsoft Visual C++ Redistributable
Setelah pengguna menekan tombol Install pada form sebelumnya, akan tampil form progress instalasi Microsoft Visual C++ Redistributable.



Gambar 4.20 Form Progress Instalasi Microsoft Visual C++ Redistributable

Apabila proses instalasi Microsoft Visual C++ Redistributable selesai, maka akan muncul form tanda instalasi telah selesai.

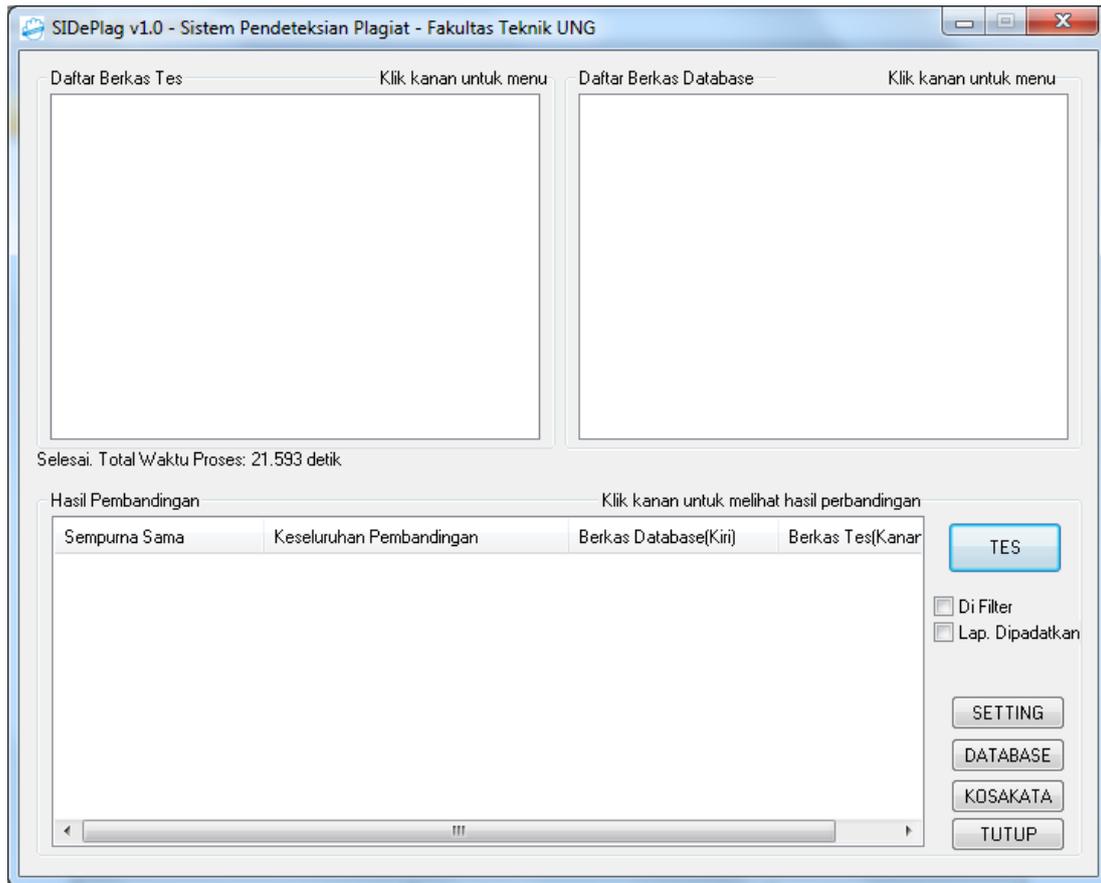


Gambar 4.21 Form Instalasi Selesai

Selanjutnya akan muncul form akhir dari bantuan instalasi (installation wizard), pada form ini apabila pengguna mengaktifkan pilihan **Launch SDePlag**, maka aplikasi pendeteksian akan otomatis dipanggil saat proses instalasi selesai.

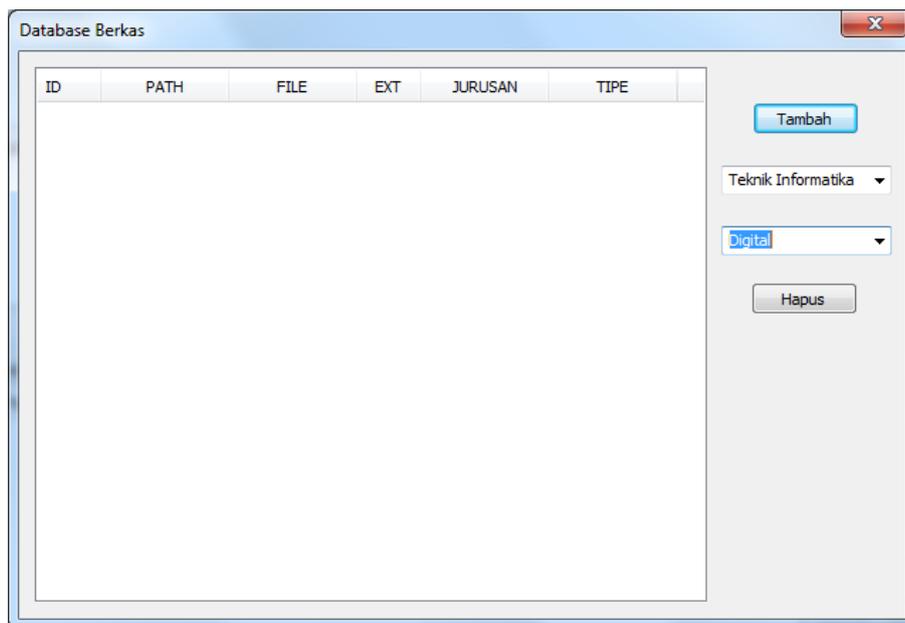
2. Pengoperasian

Saat aplikasi pendeteksian plagiat dijalankan maka aplikasi akan menampilkan form utama.



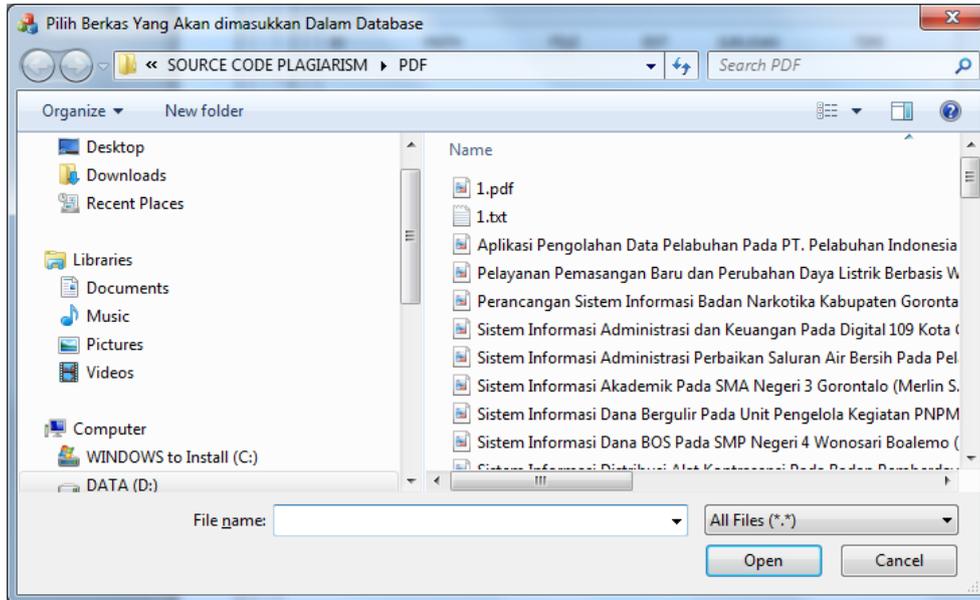
Gambar 4.22 Form Utama Aplikasi Pendeteksian Plagiat

Pada awal aplikasi belum ada database yang dimuat, untuk memuat database pengguna harus masuk pada form database dengan menekan tombol **database**.



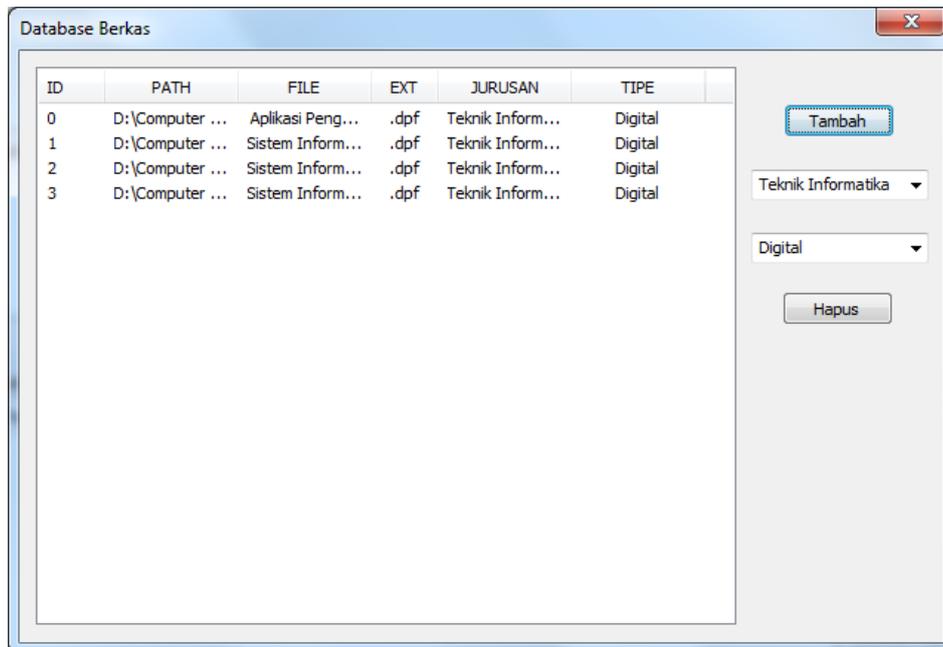
Gambar 4.23 Form Database

Untuk memasukkan berkas ke dalam database, pengguna dapat menekan tombol **tambah**, sehingga akan muncul form pilih berkas. Pada form ini pengguna dapat memilih satu atau lebih berkas yang akan dimuat ke dalam database, untuk memilih lebih dari satu berkas, tahan tombol **ctrl**.



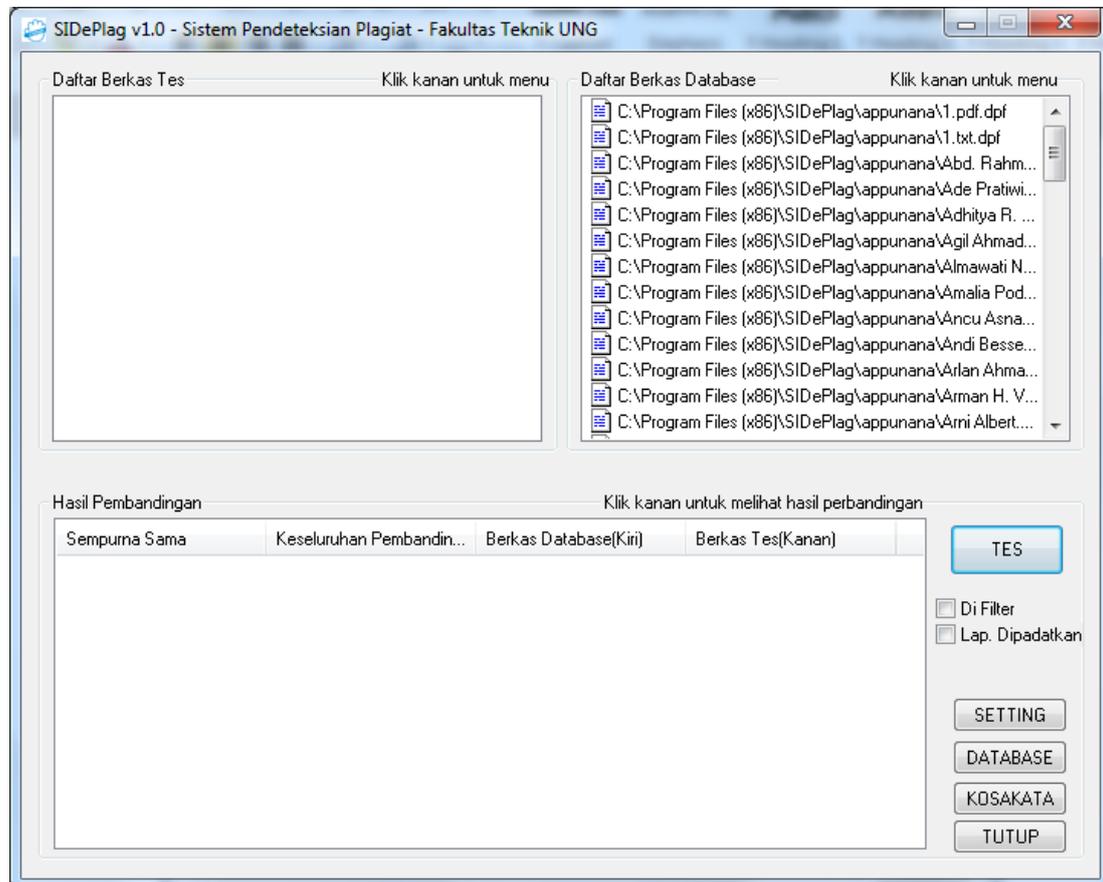
Gambar 4.24 Form pilih berkas

Setelah dimuat maka berkas akan ditampilkan pada form database seperti dibawah:



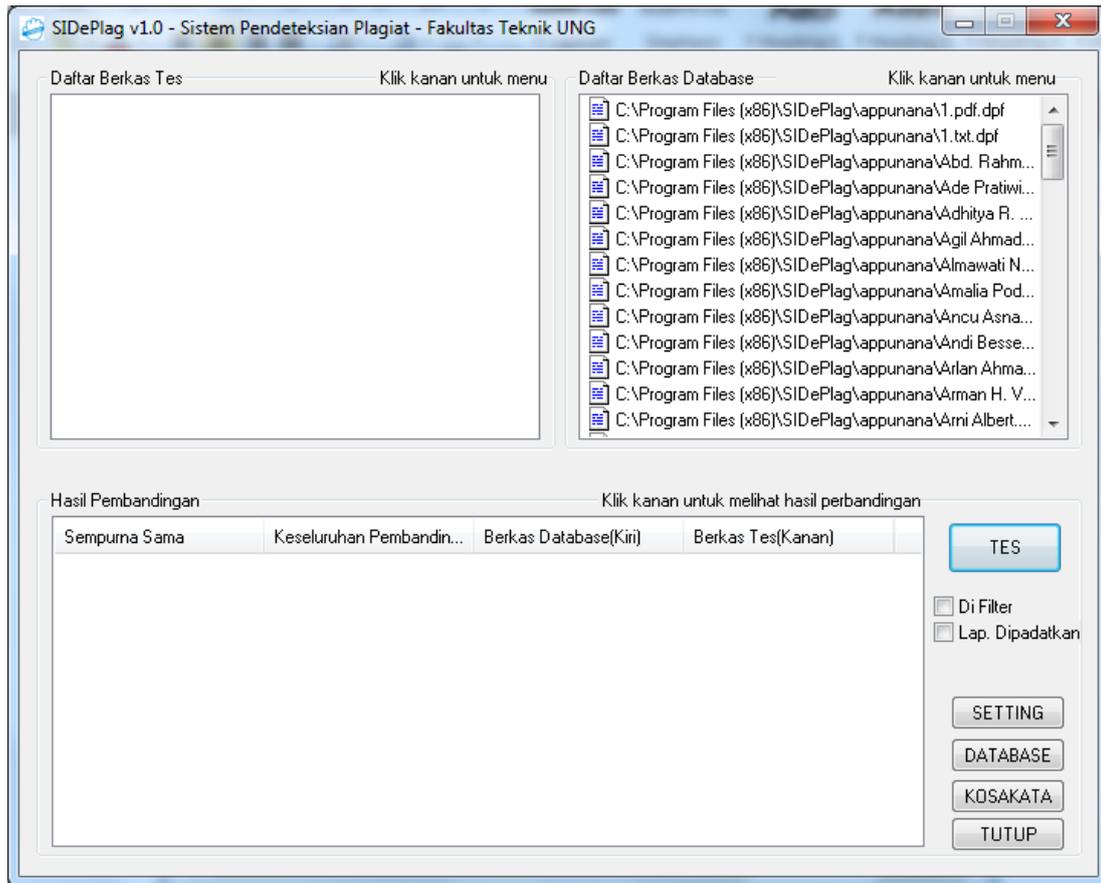
Gambar 4.25 Form database yang telah berisi daftar berkas

Setelah database dibuat, pengguna dapat kembali ke form utama, dan dapat memuat berkas dari database yang telah dibuat sebelumnya dengan melakukan klik kanan mouse pada bagian “Daftar Berkas Database” kemudian memilih **Load dari Database**.



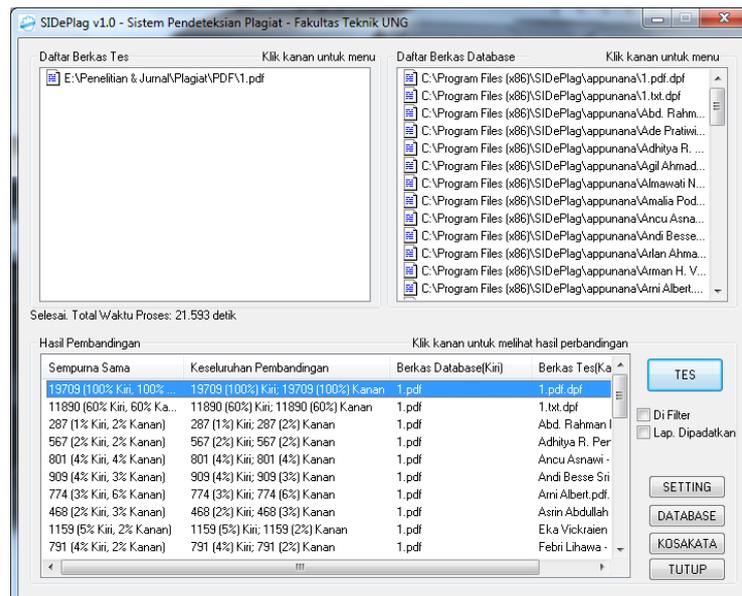
Gambar 4.26 Memuat database pada form utama

Apabila database telah dimuat, maka berkas yang ada dalam database tersebut akan ditampilkan.



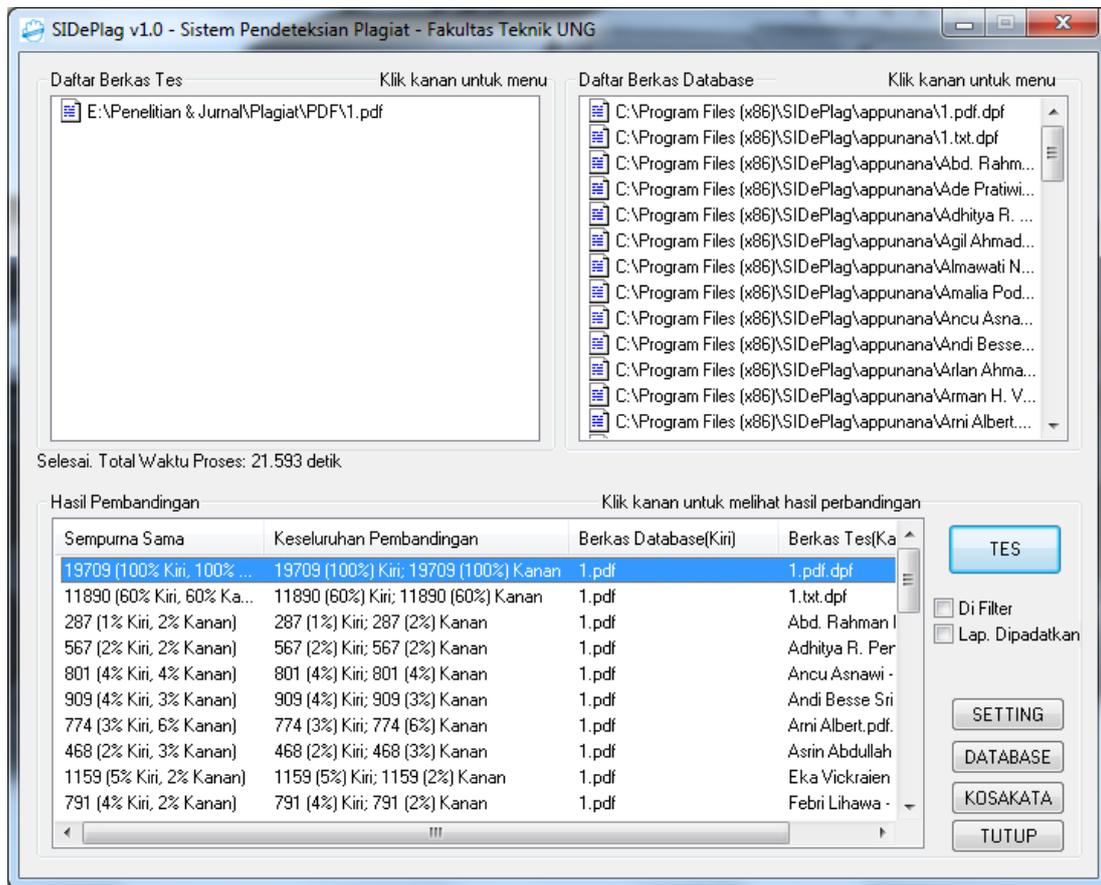
Gambar 4.27 Berkas dari database ditampilkan pada form utama

Setelah database selesai dimuat, selanjutnya adalah memuat berkas yang akan diuji, caranya adalah dengan melakukan click kanan pada bagian “Daftar Berkas Tes” dan memilih **cari berkas**.



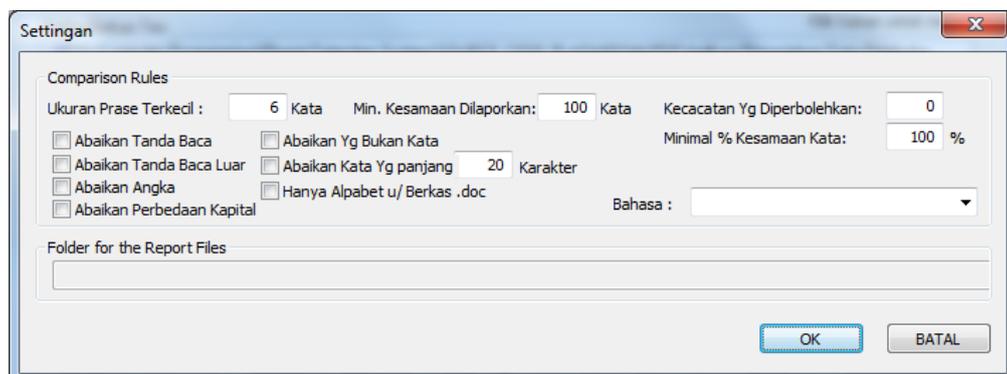
Gambar 4.28 Memuat berkas yang akan diuji

Apabila berkas yang akan diuji selesai dimuat maka berkas akan ditampilkan pada bagian “Daftar Berkas Tes”.



Gambar 4.29 Daftar berkas yang akan diuji ditampilkan

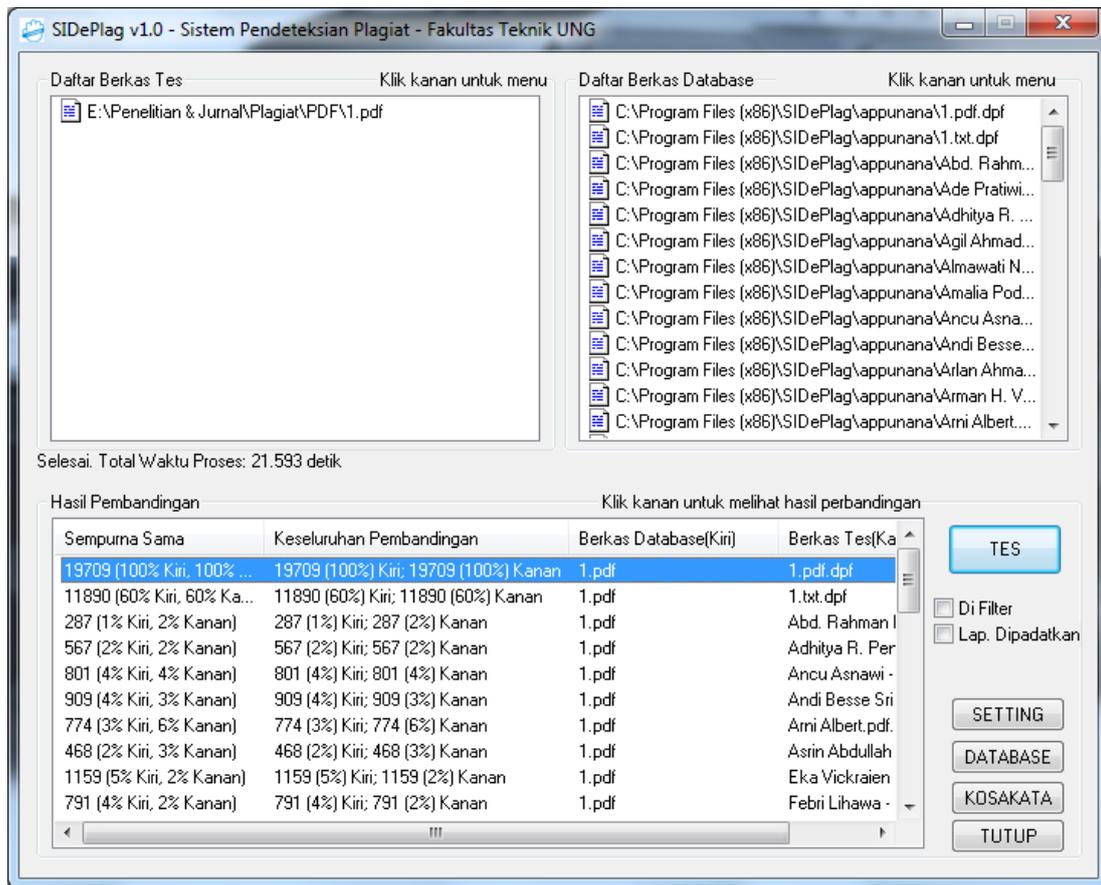
Sebelum melakukan pendeteksian, pengguna dapat melakukan pengaturan proses pendeteksian, yaitu dengan menekan tombol **Setting**, sehingga akan muncul form Settingan seperti dibawah.



Gambar 4.30 Form Settingan

Setelah selesai melakukan pengaturan, pengguna dapat menekan tombol **OK** untuk kembali ke form utama. Untuk memulai proses pendeteksian pengguna dapat

menekan tombol **TES**, sehingga aplikasi akan memulai pendeteksian dan menampilkan hasilnya pada bagian “Hasil Perbandingan”.



Gambar 4.31 Hasil Pendeteksian ditampilkan

4.6.2.3. Sistem Pelaporan

Saat aplikasi selesai melakukan pendeteksian, aplikasi juga akan menampilkan secara otomatis, pelaporan secara detil dari hasil berkas-berkas yang terdeteksi pada halaman web yang dibuka pada web browser default yang terinstall.



Gambar 4.32 Pelaporan Detail dari Proses Pendeteksian

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. Aplikasi pendeteksian plagiat didesain berbasis *desktop* sedangkan untuk sistem pelaporan berbasis web. Pada proses pengembangannya menggunakan model prototipe.
2. Aplikasi yang dibangun berhasil mendeteksi kalimat yang sama antara berkas yang diuji dengan berkas yang ada pada *repository*. Kalimat yang sama antara berkas yang diuji dengan berkas yang ada pada *repository* ditandai dengan warna merah, sementara kalimat yang tidak identik ditandai dengan warna hitam.
3. Peneliti melakukan modifikasi pada algoritma rabin-karpp khususnya pada teknik *hashing*, hal ini dilakukan agar proses pencarian kalimat menjadi lebih cepat.

5.2. Saran

1. Sistem dapat dikembangkan menjadi berbasis web sehingga dapat digunakan melalui internet.
2. Penelitian dilanjutkan dengan mencoba penggunaan metode pencarian yang berbeda untuk menemukan metode yang lebih baik untuk diterapkan.

DAFTAR PUSTAKA

- Abdeen, Ali.,Rawan, 2011, *An Algorithm for String Searching Based on Brute-Force Algorithm*, Intenational Journal of Computer Science and Network Security, Vol.11 No.7.
- Anzelmi, Daniele., et. Al, 2011, Plagiarism Detection Based SCAM Algorithm, Proceedings of the International MultiConference of Engineers and Computer Scientist 2011, Vol.1.
- Dreher,. Heinz, 2007, Automatic Conceptual Analysis for Plagiarism Detection, Issues in Informing Science and Information Technology Volume 4,
- Firdaus, Bagus, 2008, *Deteksi Plagiat Dokumen Menggunakan Algoritma Rabin-Karp*, Makalah If2251 Strategi Algoritmik Tahun 2008
- Gipp, Bela, et. Al, 2011, Citation Pattern Matching Algorithms for Citation-based Plagiarism Detection: Greedy Citation Tiling, Citation Chunking and Longest Common Citation Sequence, scholar.google.com, diakses 14 November 2012.
- Jain, Shivani., Rao, Nersimha, A.L., Agarwal, Pankaj, *A Relative Study of Pattern Matching Algorithms*, Journal of Computing Technologies, Vol.2 Issue 1.
- Lukashenko, Romans., et. al., 2007, Computer-Based Plagiarism Detection Methods and Tools: An Overview, International Conference on Computer Systems and Technologies - CompSysTech'07.
- Mutiara, Benny. A, Agustina., Sinta, 2008, *Anti Plagiarism Application With Algorithm Karp-Rabin At Thesis in Gunadharma University*, Gunadharma University, Jakarta.
- Maharani., Puanta Della, et. al., 2012, Penilai Otomatis Praktikum Pemrograman dengan Pendeteksi Plagiarisme untuk Praktikum Pengenalan Teknologi Informasi (PTI) B, Jurnal Sarjana Institut Teknologi Bandung Bidang Teknik Elektro dan Informatika Volume 1, Number 2, Juli 2012.
- Schelimer., Saul, et. Al, 2003 WInnowing: Local Algorithms for Document Fingerprinting, <http://dl.acm.org/citation.cfm?id=872770>, diakses 13 November 2012

