

BAB III

ANALISA MASALAH DAN PERANCANGAN PROGRAM

III.1. Analisis Masalah Umum

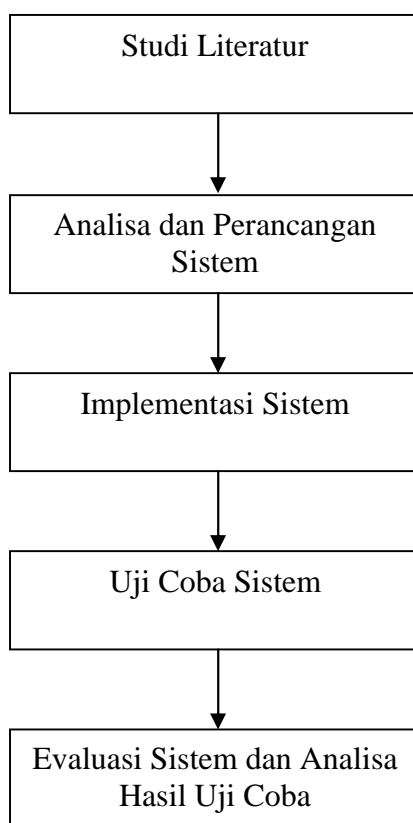
Pada bab ini akan dibahas tentang perancangan sistem deteksi plagiarisme dengan menggunakan algoritma Rabin-Karb. Algoritma yang digunakan adalah algoritma Rabin-Karp biasa dan algoritma Rabin-Karb yang telah dimodifikasi (Singh Kochar, 2008). Dalam perancangan system deteksi plagiarisme ini, adapun langkah-langkah yang dilakukan adalah:

1. Mempelajari konsep algoritma Rabin-Karp yang digunakan dalam mendeteksi plagiarisme. Konsep yang dipelajari meliputi algoritma Rabin-Karb asli (yang belum dimodifikasi) dan algoritma Rabin-karp yang telah dimodifikasi. Dan mempelajari metode-metode penunjang lainnya yang akan digunakan dalam penelitian ini, seperti yang telah dijelaskan pada bab sebelumnya.
2. Menganalisa dan merancang sistem untuk mendeteksi plagiarisme dengan menggabungkan beberapa metode yang telah dijelaskan sebelumnya.
3. Melakukan implementasi sistem berdasarkan analisa dan perancangan yang telah dilakukan sebelumnya.
4. Melakukan uji coba terhadap sistem yang telah dibuat dengan menganalisa hasil daripada sistem. Hasil yang dikeluarkan oleh sistem berupa waktu proses dan persentase kemiripan (*similarity*) antara dokumen teks yang diuji dengan dokumen teks asli dengan menggunakan algoritma Rabin-

Karp asli (yang belum dimodifikasi) dan menggunakan algoritma Rabin-Karp yang telah dimodifikasi.

5. Mengevaluasi hasil kedua algoritma tersebut apakah perbedaan yang dihasilkan dari algoritma Rabin-Karp yang belum dimodifikasi dengan algoritma Rabin-Karp yang telah dimodifikasi.

Langkah-langkah yang dilakukan dapat dilihat pada gambar III.1 berikut:



Gambar III.1 Diagram sistem

Berbeda dengan mendeteksi *plagiarisme* pada *source code* program yang memiliki aturan dan penulisan yang jelas dan teratur, mendeteksi *plagiarisme* antara dua dokumen teks lebih sulit dikarenakan bahasa manusia

lebih bersifat dinamis yaitu terus mengalami perubahan dan memiliki banyak factor pengubah yang juga ikut berubah (*fleksibel*). Oleh karena itu system yang dibangun bersifat hanya membantu semaksimal mungkin pengguna mempercepat dan mempertepat mengidentifikasi suatu dokumen teks yang terdeteksi *plagiat*.

III.2. Analisa Kebutuhan Sistem

Pembuatan aplikasi pendeteksi plagiarisme ini membutuhkan serangkaian perangkat yang dapat mendukung kelancaran proses pembuatan dan pengujian aplikasi. Berikut ini aspek – aspek yang dibutuhkan dalam pembuatan aplikasi pendeteksian *plagiarisme*:

III.2.1 Perangkat Keras (*Hardware*)

Hardware merupakan komponen yang terlihat secara fisik, yang saling bekerjasama dalam pengolahan data. Perangkat keras (*hardware*) yang digunakan meliputi :

- a. Notebook
- b. Processor AMD Dual-Core C60
- c. Harddisk 320 GB
- d. Memory 2 GB DDR3
- e. Battery 6-cell Li-ion

III.2.2. Perangkat Lunak (*Software*)

Software adalah intruksi atau program – program computer yang dapat digunakan oleh computer dengan memberikan fungsi serta penampilan yang diinginkan. Dalam hal ini, perangkat lunak yang digunakan penulis untuk aplikasi perancangan pendeteksian *plagiarisme* adalah :

- a. Sistem Operasi Windows 7
- b. Netbeans 6.0.1
- c. Bahasa Pemrograman Java

III.2.3. Unsur Manusia (*Brainware*)

Brainware merupakan factor manusia yang menangani fasilitas computer yang ada. Factor manusia yang dimaksud adalah orang – orang yang memiliki bagian untuk menangani system dan merupakan unsure manusia yang meliputi :

- a. Analisa Sistem, yaitu orang membentuk dan membangun fasilitas rancangan system.
- b. *Programmer*, yaitu orang yang mengerti bahasa pemrograman yang digunakan dalam membuat dan membangun suatu program.
- c. *Operator* (Administrator), yaitu orang yang mengoperasikan system seperti memasukan data untuk dioperasikan oleh computer dalam menghasilkan informasi dan lain sebagainya.
- d. *Public* (Pengguna), yaitu orang yang memakai system yang telah dirancang untuk mendapatkan informasi yang dibutuhkan.

III.3. Analisis Perancangan

Dalam suatu pembangunan aplikasi, analisis perlu dilakukan sebelum tahap perancangan dilakukan. Perancangan aplikasi harus menganalisis kebutuhan apa saja yang diperlukan untuk membangun suatu perangkat lunak.

Bagian desain proyek berarti pengetahuan dan keterampilan mengenai computer, keahlian dalam perancangan program yang logis melalui informasi, semuanya difokuskan untuk membuat Sesuatu yang nyata. Mendesain berarti berfikir, memilih, membuat, dan mengerjakan.

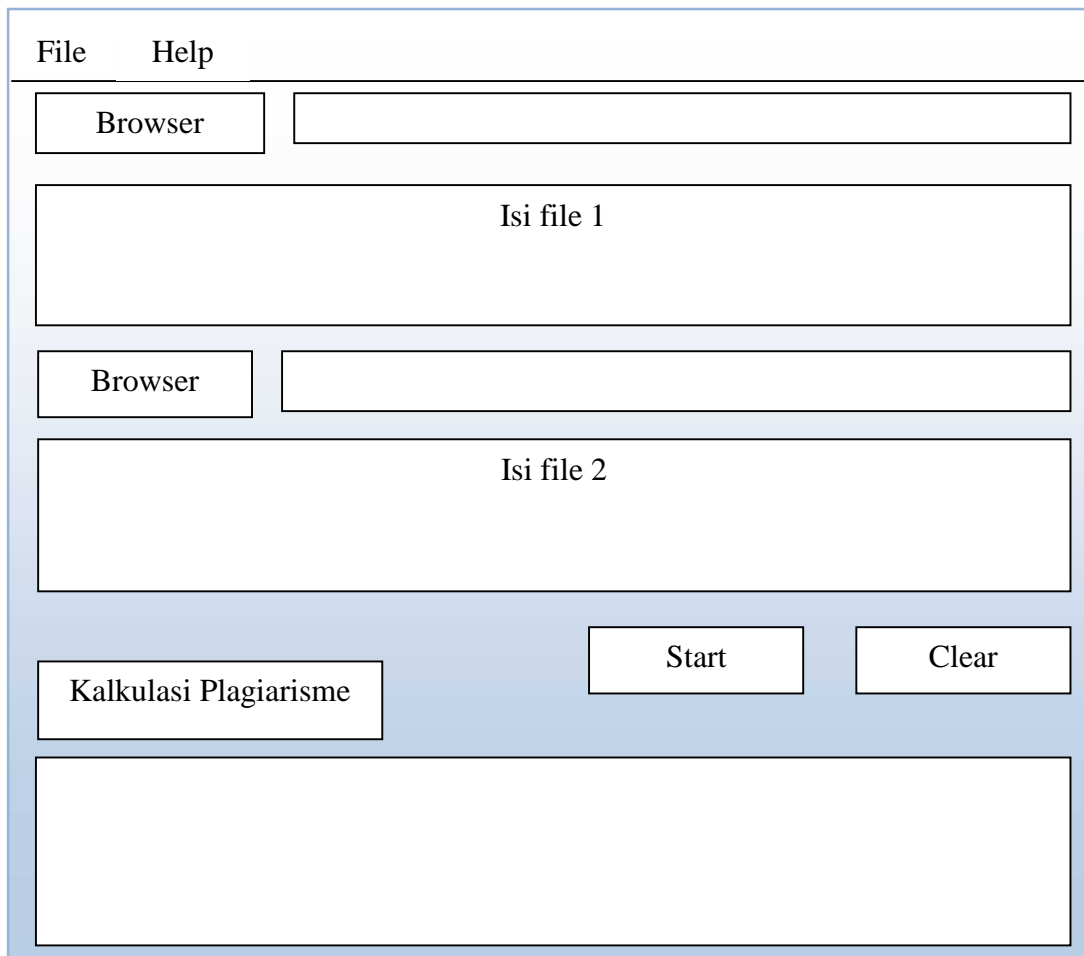
Kebutuhan utama dalam perancangan aplikasi pendeteksian *plagiarisme* perangkat lunak yang digunakan untuk membangun aplikasi yaitu : Netbeans 6.0.1 Semua kebutuhan itu harus dapat dituangkan ke dalam perancangan agar hasil aplikasi dapat sesuai dengan konsep pembangunan.

III.3.1. Perancangan Aplikasi

Perancangan tampilan menu atau rancangan *Jframe* dalam program ini sangat diperlukan dalam perograman visual karena *Jframe* ini merupakan bentuk tampilan saat program dijalankan. Pada aplikasi pendeteksian *plagiarisme* ini terdapat 1 form yang dirancang yaitu :

1. Tampilan *Jframe* Aplikasi Program

Pada tampilan *Jframe* program aplikasi pendeteksian *plagiarisme* ini berfungsi sebagai pengujian aplikasi program pendeteksi *plagiarisme*. *Jframe* aplikasi program ini dapat dilihat pada gambar III.2. di bawah ini:

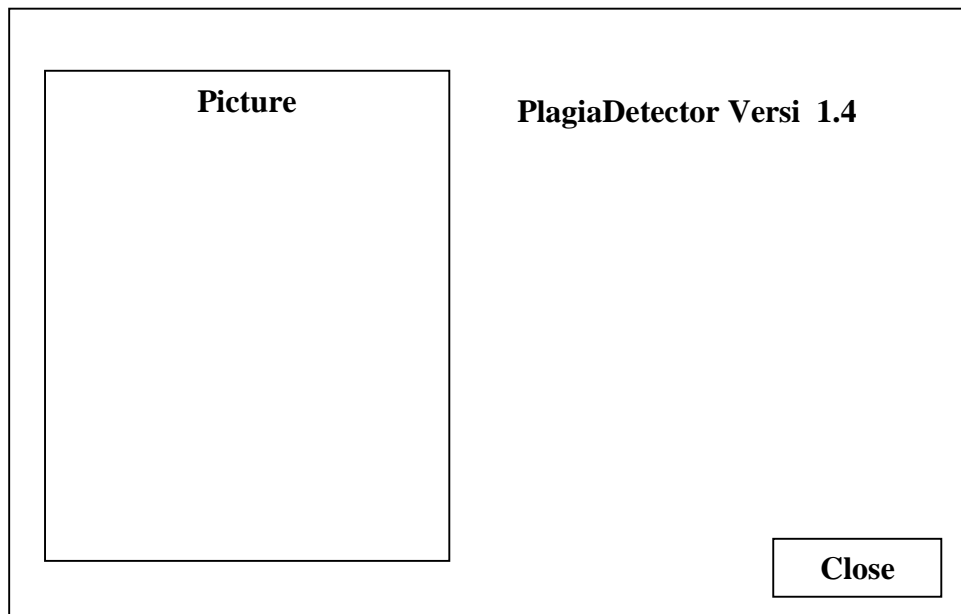


Gambar III.2. Tampilan *JFrame* Aplikasi Program

Pada gambar III.1. adalah tampilan *Jframe* aplikasi program yang akan pertama kali muncul saat program dijalankan. Fungsi *Jframe* ini sendiri untuk menguji program pendeteksi *plagiarisme* yaitu dengan membuka file .txt pertama dan membuka file .txt kedua. Lalu diklik kalkulasi *plagiarisme* dan akan muncul hasil persentasi tingkat kesamaan text pertama dan text kedua.

2. Tampilan *JInternalFrame* About

Pada tampilan *JInternalframe* program aplikasi About ini berisikan tentang pembuat skripsi dan menampilkan gambar . *JInternalframe* aplikasi program ini dapat dilihat pada gambar III.3. di bawah ini:

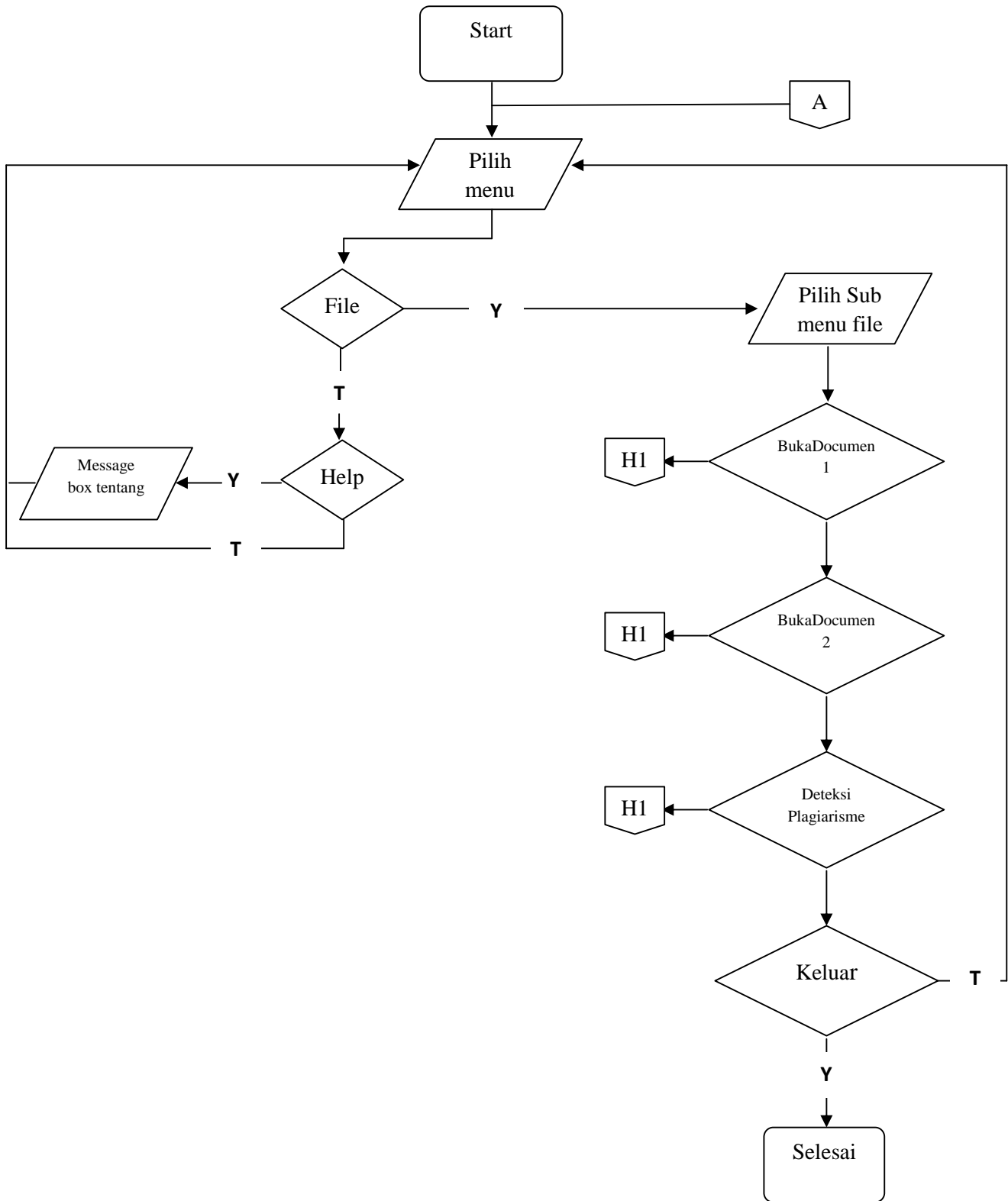


Gambar III.3. *JInternalFrame* About

III.3.2. Logika Program

Berikut ini adalah aliran (*Flowchart*) yang penulis gunakan dalam menggambarkan logika program, adalah sebagai berikut :

1. Flowchart Sistem Menu Utama

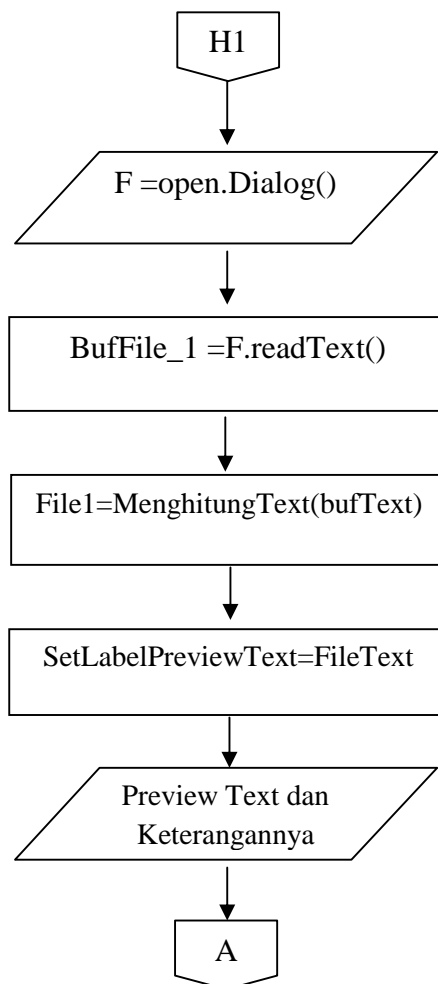


Gambar III.4. Flowchart Sistem Menu Utama Program

Algoritma Flowchart Utama :

- a. User memilih menu utama, menu file dan help
- b. Pilih sub menu file, buka dokumen 1, buka dokumen 2, deteksi *plagiarisme* dan keluar
- c. Keluar dari menu program.

2. Flowchart Buka Dokumen 1

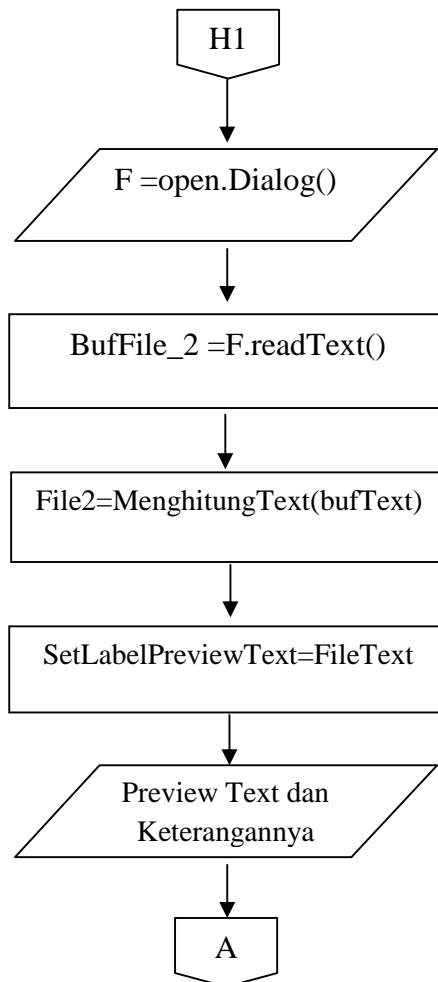


Gambar III.5. Flowchart Buka Dokumen I

Algoritma *flowchart* buka dokumen 1

- a. Pilih buka file dokumen 1
- b. Pilih dokumen text 1 yang berformat*.txt
- c. File text yang di ambil diset menjadi *preview* text
- d. Text dokumen 1 ditampilkan.

3. Flowchart Buka Dokumen 2

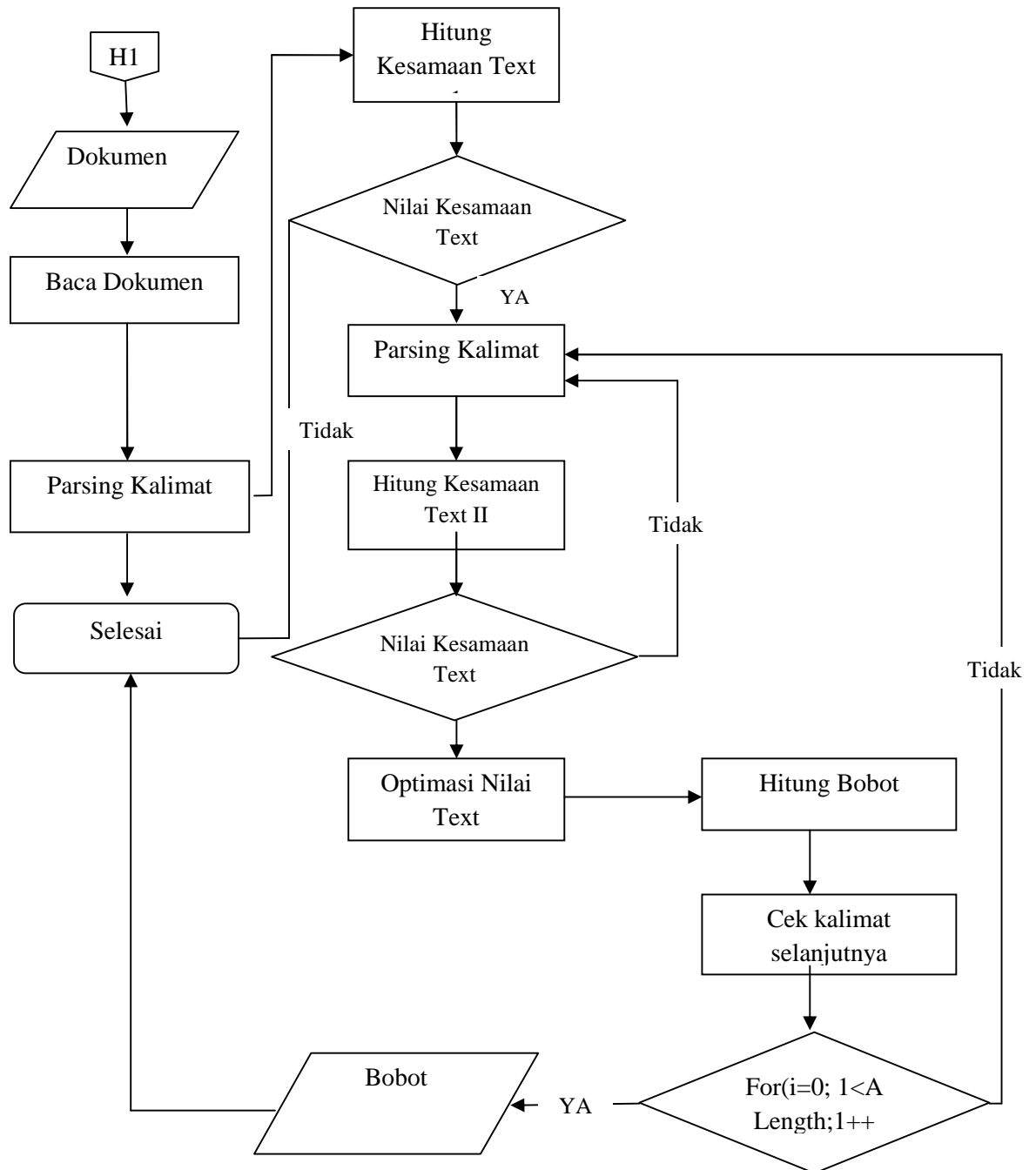


Gambar III.6. Flowchart Buka Dokumen II

Algoritma *flowchart* buka dokumen II

- e. Pilih buka file dokumen 2
- f. Pilih dokumen text 2 yang berformat*.txt
- g. File text yang di ambil diset menjadi *preview* text
- h. Text dokumen 2 ditampilkan.

4. Flowchart Deteksi Plagiarisme



Gambar III.7. Flowchart Deteksi Plagiarisme

Algoritma *Flowchart* deteksi *plagiarisme*

- a. Dokumen 1 dan dokumen 2
- b. Baca dokumen
- c. Jika error maka ke selesai, jika tidak maka akan menguraikan (parsing) dokumen
- d. Hitung kesamaan text dokumen 1
- e. Nilai kesamaan text
- f. Uraikan kalimat
- g. Hitung kesamaan text dokumen 2
- h. Nilai kesamaan text
- i. Mengoptimalkan nilai kesamaan text
- j. Hitung bobot
- k. Cek kalimat selanjutnya
- l. Maka mengincrementkan kalimat selanjutnya
- m. Maka menampilkan nilai bobot
- n. selesai