BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Algoritma *template matching* merupakan metode yang sederhana dan banyak digunakan untuk mengenali pola. Algoritma ini bekerja dengan cara mengevaluasi pola citra yang akan dibandingkan dengan pola citra *template* pada basis data. Kelemahan algoritma ini adalah terbatasnya model yang akan dijadikan *template* sebagai pembanding pada basis data seperti bentuk, ukuran, dan orientasi[10].

Algoritma *feature extraction* menjawab masalah model *template* seperti bentuk, ukuran, dan orientasi yang ada pada algoritma *template matching* dengan cara memetekan ciri-ciri objek citra yang akan dikenali. Ciri-ciri yang dipetakan dari citra yang akan dikenali dan diklasifikasikan terhadap ciri-ciri citra *template* yang disimpan pada basis data. Algoritma *feature extraction* memiliki kelemahan dalam menentukan ciri-ciri khusus citra yang akan dikenali[10].

Optical character recognition digunakan untuk menerjemahkan karakter pada citra digital menjadi format teks. Penerapannya yang sederhana membuat algoritma *template matching* banyak digunakan dalam OCR[1]. Namun, penggunaan algoritma *template matcing* terhadap OCR masih memiliki tingkat akurasi pengenalan yang kecil. Hal ini terjadi karena terbatasnya *template* yang disimpan dalam basis data sedangkan model *template*, yaitu *font* terus bertambah.

Kelebihan algoritma *feature extraction* yang tidak terbatas dalam model *template* diharapkan bisa meningkatkan akurasi pengenalan terhadap OCR.

Untuk mengetahui apakah algoritma *feature extraction* lebih baik daripada algoritma *template matching* terhadap OCR, maka dibangunlah aplikasi yang bisa membandingkan kedua metode tersebut. Berdasarkan uraian di atas, penelitian skripsi ini berjudul "**PERBANDINGAN ALGORITMA *TEMPLATE MATCHING* DAN *FEATURE EXTRACTION* PADA *OPTICAL CHARACTER RECOGNITION***".

1.2 Rumusan Masalah

Rumusan masalah berdasarkan latar belakang yang telah diuraikan adalah bagaimana membangun aplikasi yang bisa membandingkan algoritma *template matching* dan *feature extraction* pada OCR.

1.3 Maksud dan Tujuan

Maksud dari penulisan skripsi ini adalah membangun apikasi yang bisa membandingkan algoritma *template matching* dan *feature extraction* pada *optical character recognition*.

Tujuan yang ingin dicapai dari pembangunan aplikasi perbandingan algoritma *template matching* dan *feature extraction* pada OCR adalah sebagai berikut:

1. Mengukur kedua algoritma berdasarkan akurasi, pengembangan, proses yang dibutuhkan, kompleksitas, dan waktu yang dibutuhkan dalam sistem OCR.

2. Mengetahui apakah alogritma *feature extraction* lebih baik daripada algoritma *template matching* terhadap OCR.

1.4 Batasan Masalah

Batasan masalah dalam membangun aplikasi perbandingan algoritma template matching dan feature extraction adalah sebagai berikut:

1. Citra atau gambar yang akan dikenali berformat .bmp dan .jpg.
2. Aplikasi diuji dengan dokumen citra karakter digital hasil *print screen* dan citra karakter digital yang dipindai serta tulisan tangan hasil pemindaian.
3. Agar karakter yang akan dikenali mudah dipisahkan dari latar belakangnya, maka latar belakang citra karakter yang digunakan berwarna putih. Latar belakang dengan warna lain dikhawatirkan dapat menjadi satu objek dengan karakter yang akan dikenali.
4. *Template* karakter yang digunakan adalah *font* Times News Roman dan Arial dengan *style* regular dan *font size* 12.
5. Keluaran dari aplikasi yang dibangun adalah hasil pengenalan dan waktu yang dibutuhkan masing-masing algoritma dalam mengenali citra karakter.
6. Pengukuran algoritma berupa kompleksitas, pengembangan, dan proses yang dibutuhkan dilakukan diluar aplikasi.
7. Resolusi maksimal dokumen yang bisa dikenali adalah 2500 x 2500 *pixel*.

8. Aplikasi ini akan dibangun dengan menggunakan pemodelan terstruktur dengan *tools Data Flow Diagram* (DFD) sebagai analisis kebutuhan fungsional dan *Entitiy Relationship Diagram* (ERD) sebagai analisis basis data.
9. Aplikasi dibangun dengan menggunakan bahasa pemrograman *Borland Delphi 7.*

1.5 Metodologi Penelitian

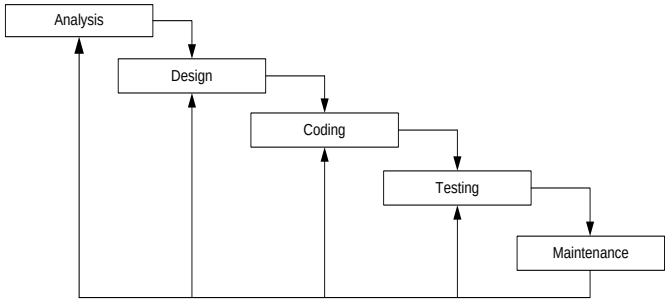
Metode penelitian dalam penyusunan tugas akhir ini dibagi menjadi dua tahap yaitu, tahap pengumpulan data dan tahap pembangunan perangkat lunak.

1.5.1 Tahap Pengumpulan Data

Teknik pengumpulan yang dilakukan adalah studi pustaka atau studi literatur. Pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

1.5.2 Tahap Pengembangan Perangkat Lunak

Dalam pengembangan aplikasi ini digunakan metode *The Classic Life Cycle* (Paradigma *Waterfall).* Tahap-tahap pembangunan perangkat lunak metode *waterfall* dapat dilihat pada gambar 1.1, dengan penjelasan sebagai berikut:



Gambar 1.1. Metode *The Classic Life Cycle / Waterfall*.

1. *Analysis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan atau pengembangan *software.*
2. *Design* adalah tahap menerjemahkan dari keperluan-keperluan yang dianalisis ke dalam bentuk yang lebih mudah dimengerti oleh pemakai.
3. *Coding* adalah tahap menerjemahkan data/pemecahan masalah *software* yang telah dirancang ke dalam bahasa pemograman yang telah ditentukan.
4. *Testing* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dilakukan untuk memastikan tidak ada kesalahan dan semua berjalan dengan baik dan *input* yang diberikan hasilnya sesuai dengan yang diinginkan.
5. *Maintenance* adalah perangkat lunak yang telah dibuat dapat mengalami perubahan sesuai permintaan pemakai atau perkembangan metode. Pada program yang dibangun ini tidak dilakukan tahap *maintenance*.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dilakukan. Dalam menyajikan laporan skripsi ini digunakan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan konsep dasar dan teori-teori dari tugas akhir yang digunakan.

BAB III ANALISIS DAN PERANCANGAN

Bab ini menjelaskan analisis dan perancangan sistem. Analisis yang dilakukan berupa analisis basis data, analisis kebutuhan non-fungsional, dan analisis kebutuhan fungsional. Perancangan yang dilakukan berupa perancangan data, perancangan anatar muka, jaringan semantik, dan perancangan prosedural.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari hasil analisis dan perancangan serta hasil pengujian perangkat lunak. Sistem diuji dengan menggunakan metode pengujian *white box*.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan berdasarkan tujuan yang ingin dicapai dalam penilitian skripsi ini dan saran-saran yang dapat diberikan untuk penggunaan dan pengembangan perangkat lunak lebih lanjut.