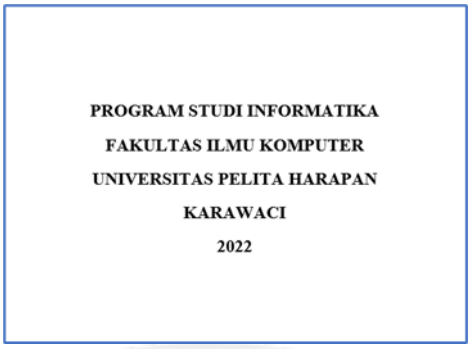
BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia ini, informasi bisa didapatkan di mana pun dan kapan pun. Sumber-sumber informasi dapat terlihat di sekitar kita, dan kebanyakan darinya dalam bentuk gambar atau teks, seperti buku atau kertas yang bersifat *hardcopy*. Karena hal tersebut, sulit untuk digitalisasi teks *hardcopy* menjadi teks, karena metode yang biasanya dipakai untuk menangkap teks *hardcopy*, seperti kamera ataupun *scanner*, hanya terbatas menyimpan *hardcopy* menjadi gambar. [1]

Ada kebutuhan mengubah teks *hardcopy* hasil *scanner* menjadi teks dalam bentuk *softcopy*. Hal ini tidak bisa dipisahkan dari perkembangan Teknologi Informasi dan Komunikasi (TIK). TIK yang berkembang, diiringi dengan menyebarnya perangkat seperti *smartphone*, dengan aplikasi-aplikasi baru yang dikembangkan, seperti kamera dan *scanner*, membuat masyarakat pada umumnya lebih cenderung menyukai media digital daripada media ketik. Juga dikarenakan memori yang dibutuhkan untuk penyimpanan media teks dalam perangkat tentu jauh lebih kecil dibanding dalam bentuk gambar. [2]



Gambar 1.1 Contoh Gambar Teks Berbentuk file .png dari Dokumen .docx

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PELITA HARAPAN
KARAWACI
2022

Gambar 1.2 Contoh Hasil Perubahan Teks dari Gambar 1.1 dalam file .txt
Sumber: https://www.imagetotext.info/

Gambar 1.1 dan 1.2 merupakan contoh dari perubahan teks dari gambar menjadi media teks melalui sebuah situs web yang menyediakan sarana tersebut. Dari perbandingan antara gambar 1.1 dan 1.2, dapat disimpulkan bahwa memori yang diperlukan untuk menyimpan media teks di atas memiliki perbedaan yang signifikan. Dilihat dari *Properties* dari kedua *file* di atas, gambar hasil teks yang terdapat di gambar 1.2 berukuran 89 *bytes*, jauh lebih sedikit daripada menyimpan gambar teks 1.1 yang berukuran 9,51 *kilobytes*, atau 9510 *bytes*. Meskipun beberapa detail berupa ketebalan *font*, spasi, dan *alignment* tidak dipertahankan dalam perubahan teks pada gambar 1.2 di atas, pada dasarnya, teks tersebut dapat dibaca dengan jelas. Selain ukuran, kejelasan teks juga penting.

Dalam zaman yang modern ini, *Machine Learning* berperan penting dalam pemrosesan data dan menggali informasi dari data tersebut. Salah satu *subset* dari

Machine Learning adalah *Deep Learning*, yang memiliki jaringan-jaringan yang dapat mempelajari data secara mandiri tanpa campur tangan manusia. *Deep Learning* terbukti efektif dalam pemrosesan data, terutama dalam jumlah yang besar (Big Data). Karena itu, *Machine Learning* sering sekali digunakan untuk pencarian dan penerapan informasi, dan juga melakukan prediksi dengan berbasis informasi. ("Why Deep Learning Is So Effective? Why It Is Important?" n.d.)

Perbedaan terbesar antara *Machine Learning* dan *Deep Learning* adalah bahwa *Machine Learning* menerapkan algoritma-algoritma sederhana, sementara *Deep Learning* menerapkan algoritma-algoritma yang jauh lebih kompleks. Algoritma kompleks tersebut merupakan bagian dari jaringan saraf buatan yang mendalam dan kompleks, terdiri dari lapisan-lapisan JST yang tersembunyi. *Deep Learning* juga terbukti lebih sulit untuk dimengerti daripada *Machine Learning*. Meskipun begitu, tidak dapat dipungkiri bahwa hasil *Deep Learning* lebih akurat daripada *Machine Learning*, dan *Machine Learning* tetap lebih unggul daripada *Deep Learning* dalam bidang tertentu. Saat ini, penerapan *Deep Learning* mencakupi *Natural Language Processing* (NLP), *Computer Vision, Pattern Recognition*, dan lain sebagainya. [4]

Salah satu yang disebut di atas adalah *Computer Vision*, yang termasuk pemrosesan gambar dan klasifikasi gambar. Dalam hal tersebut, penggunaan CNN, atau *Deep Learning*, lebih unggul daripada menggunakan algoritma *Machine Learning* tradisional, seperti *Linear Regression* dan *Nearest Neighbor*. [5]. Salah satu aspek terpenting dalam *Computer Vision*, yang sering dipakai dalam kehidupan sehari-hari adalah *Optical Character Recognition* (OCR). OCR memiliki dua tahap,

yaitu pendeteksi teks dan pengenalan teks, baik teks yang berbentuk tertulis ataupun diketik. [6]

Dengan begitu, penulis bertujuan untuk melakukan penerapan salah satu metode *Deep Learning* untuk mendeteksi dan mengenali teks dalam gambar, dan, dengan begitu, berhasil melakukan OCR dengan tingkat keakuratan yang tinggi dan *error rate* yang rendah. Seperti yang dijelaskan di atas, diperlukan sebuah metode OCR yang efektif, yang dapat mendeteksi teks secara akurat.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas adalah sebagai berikut:

1) Bagaimana menerapkan *Tesseract*, yang menerapkan Jaringan Saraf Buatan LSTM, untuk mendeteksi dan mengenali teks dalam gambar melalui OCR?
2) Seberapa akurat Tesseract OCR dengan LSTM dalam mendeteksi dan mengenali teks? Bagaimana perhitungan *Accuracy* dan *Error Rate* dari teks hasil OCR tersebut?
3) Bagaimana dampak *pre-processing*, kualitas gambar, dan isi halaman terhadap hasil OCR?

1.3 Batasan Masalah

Dalam penelitian ini, diperlukan beberapa batasan yang digunakan sebagai acuan dalam pelaksanaan penelitian untuk memberikan arah yang jelas dalam pengembangannya. Batasan-batasan yang terdapat dalam penelitian adalah sebagai berikut:

1) Penelitian dan perancangan sistem dilakukan dengan menggunakan bahasa pemrograman *Python*, dengan menggunakan *library Tesseract*,

OpenCV, dan *Levenshtein*, dilakukan dalam aplikasi *Visual Studio Code* dalam bentuk *file Jupyter Notebook*.

2) Metode yang dipilih untuk *pre-processing* gambar adalah *OpenCV*, dengan menerapkan metode *Grayscale*, *Noise Removal* dengan *Median Filter*, *Gaussian Filter*, *Bilateral Filter*, *Global Thresholding*, *Otsu Thresholding*, *Otsu Thresholding* dengan *Gaussian Filter*, dan *Canny Edge Detection*.
3) Metode yang dipilih untuk OCR adalah *Tesseract*, dengan konfigurasi yang menerapkan JST LSTM, menerapkan OSD (Orientation and Script Detection) untuk *segmentation*, dan menerapkan data *trained* dan model-model bahasa *tessdata-best*.
4) Jenis teks yang akan dideteksi melalui gambar teks berbahasa Indonesia yang bersifat diketik, diambil dari buku berjudul *Komunikasi Nirkabel dan Aplikasinya di Bidang Telekomunikasi dan Informatika*.
5) Gambar yang diambil merupakan hasil *scan* dengan menggunakan aplikasi *Adobe Scan* yang terdapat dalam *smartphone*.
6) *Levenshtein* akan digunakan untuk menghitung *Character Error Rate* dan *Accuracy*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang dan menerapkan sebuah contoh model *Deep Learning* untuk mendeteksi teks dalam sebuah contoh gambar yang diambil dari buku berjudul *Komunikasi Nirkabel dan Aplikasinya di Bidang Telekomunikasi dan Informatika*, yang di-*scan* melalui aplikasi *Android Adobe*

Scan yang tersedia di *smartphone*. Metode yang dipilih adalah *Tesseract* yang menerapkan JST *Deep Learning LSTM*. Sistem yang dirancang dalam *Jupyter Notebook* akan diuji untuk melihat hasil dari sistem tersebut. Sebelum itu, gambar akan diproses melalui *pre-processing* dengan *OpenCV*. Hasil yang diinginkan adalah teks yang berhasil diambil dan dijadikan *output* dalam *Jupyter Notebook* tersebut, dengan *accuracy* yang tinggi dan *error rate* yang rendah.

1.5 Metodologi

Untuk menyelesaikan masalah pada penelitian ini, metode-metode yang digunakan adalah:

1. Melakukan studi pustaka untuk mendapatkan berbagai informasi yang berhubungan dengan penerapan metode *Deep Learning* dalam bahasa pemrograman *Python*.
2. Merancang sebuah sistem dalam *Jupyter Notebook* dengan *Visual Studio Code* yang menerapkan apa yang dipelajari dari studi pustaka yang disebutkan di atas.
3. Melakukan OCR dengan menggunakan *Tesseract* dengan konfigurasi LSTM terhadap gambar yang diambil dari *scan* buku *Komunikasi Nirkabel dan Aplikasinya di Bidang Telekomunikasi dan Informatika*
4. Melakukan *pre-processing* gambar dengan metode *OpenCV* dengan menerapkan metode *Grayscale*, *Noise Removal* dengan *Median Filter*, *Gaussian Filter*, *Bilateral Filter*, *Global Thresholding*, *Otsu Thresholding*, *Otsu Thresholding* dengan *Gaussian Filter*, dan *Canny Edge Detection*.

5. *Accuracy* dan *Error Rate* dari hasil OCR akan dihitung dengan *Levenshtein* dengan mengacu pada file .txt yang berisikan teks yang seharusnya dibaca.
6. Hasil dari gambar dan perhitungan *Accuracy* dan *Error Rate* akan dianalisis untuk mengetahui dampak *OpenCV* terhadap hasil OCR dan keakuratan *Tesseract* dalam melakukan OCR.

1.6 Sistematika Penulisan

Tugas akhir ini ditulis dengan sistematika penulisan yang dapat dilihat di bawah ini:

BAB I PENDAHULUAN

Bab ini dimulai dengan latar belakang untuk masalah yang menjadi topik penulisan tugas akhir ini, termasuk yang berkaitan dengan perihal *Deep Learning* dan *Optical Character Recognition*, atau OCR. Diikuti dengan rumusan masalah dan batasan masalah. Tujuan penelitian dan metodologi penelitian juga dijelaskan pada bab tersebut.

BAB II LANDASAN TEORI

Bab ini berisikan landasan teori yang dijadikan acuan untuk merancang dan mengembangkan topik tugas akhir ini. Teori yang digunakan meliputi OCR beserta langkah-langkahnya, *Deep Learning* beserta *subset-subset* yang bersangkutan, bahasa pemrograman *Python*, *OpenCV*, *Tesseract*, *Levenshtein*, *Jupyter Notebook*, dan *Visual Studio Code*.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisikan analisis dan perancangan sistem untuk proses OCR, dengan proses-proses meliputi proses perancangan sistem yang menggunakan bahasa pemrograman *Python*, file *Jupyter Notebook*, piranti lunak *Visual Studio Code*, *library Tesseract* dan *OpenCV*. Langkah-langkah diawali dengan cara instalasi dan pengaturan sistem. Diikuti dengan langkah-langkah OCR dengan menerapkan bahasa pemrograman *Python* dalam Jupyter Notebook, meng-*import library OpenCV* dan *Tesseract* untuk pemrosesan gambar dan pengambilan teks. Pada akhir bab, terdapat implementasi *library Levenshtein* untuk menghitung *accuracy* dan *error rate* dari hasil OCR *Tesseract*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini berisikan implementasi dan pengujian sistem yang sudah dirancang dalam *file Jupyter Notebook* melalui *Visual Studio Code*, disertai dengan langkah-langkah OCR yang dilakukan dalam setiap *cell Jupyter Notebook* tersebut. Bab juga berisikan empat gambar yang menjadi subjek untuk pengujian sistem OCR, dengan mengikuti langkah-langkah OCR. Setiap hasil pengujian terhadap gambar yang dipilih akan dianalisis untuk membuat kesimpulan dan saran.

BAB V PENUTUP

Bab ini berisikan kesimpulan yang diambil dari hasil pengujian sistem OCR yang menerapkan *Tesseract* tersebut. Hasil juga akan diberi saran untuk pengembangan lebih lanjut.