



PERNYATAAN

Saya yang bertanda tangan dibawah ini menyatakan bahwa skripsi yang saya buat ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah ditunjukkan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggung jawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 30 Agustus 2019



Ananda Dharmayana Rani

NPM 1411010073

HALAMAN PERSETUJUAN

Judul Laporan : Mengaplikasikan Metode Optical Character Recognition Untuk Ekstraksi Teks Pada Gambar

Nama Mahasiswa : Ananda Dharmayana Rani

Nomor Pokok Mahasiswa : 1411010073

Jurusan : S1 Teknik Informatika



Dosen Pembimbing

Ketua Jurusan

Teknik Informatika

Dr. Ir. Suhendro Yusuf Irianto,
M.Kom

NIK. 01600603

Yuni Arkhiansyah, S.Kom.,
M.Kom

NIK. 00480802

HALAMAN PENGESAHAN

Telah diuji dan dipertahankan di depan Tim Penguji Skripsi Jurusan
Teknik Informatika Institut Informatika & Bisnis Darmajaya Bandar
Lampung dan dinyatakan diterima untuk memenuhi syarat guna
memperoleh gelar Sarjana Komputer

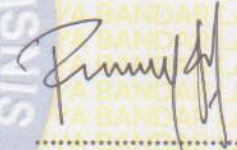
Mengesahkan

1. Tim Penguji :

Ketua : Septilia Arfida, S.Kom., M.T.I

Anggota : Rionaldi Ali, S.Kom., M.T.I

Tanda Tangan:



2. Dekan Fakultas Ilmu Komputer



Sriyanto, S.Kom., M.M

NIK. 00210800

ABSTRACT

APPLYING OPTICAL CHARACTER RECOGNITION METHOD FOR TEXT EXTRACTION IN PICTURES

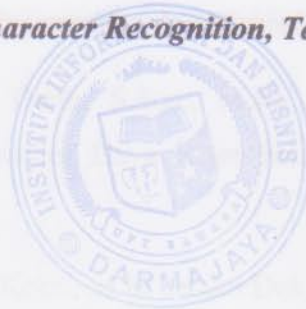
By:

Ananda Dharmayana Rani

1411010073

Today, writing-based information is something that cannot be separated from our daily lives. Various sorts of information can be found easily, ranging from print media, to electronic media. Easy information dissemination, especially through the internet media, allows us to get information from various parts of the country. However, not all information circulating can be used easily. Difficulty of taking information in the form of writing in a picture impedes many students who are conducting researches. The application is created using the *optical character recognition* method using Tesseract-OCR and developed in *Visual Studio Community 2017 IDE using C #* programming language. The result of this study produced the application that could facilitate the process of taking writing on the picture.

Keywords: *Optical Character Recognition, Tesseract-OCR, Computer Vision.*



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERNYATAAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERSEMBAHAN	v
RIWAYAT HIDUP	vi
MOTTO	vii
ABSTRAK	viii
ABSTRACT	ix
PRAKATA	x
DAFTAR ISI	xii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah.....	1
1.3 Rumusan Masalah.....	1
1.4 Ruang Lingkup	1
1.5 Tujuan Penelitian	2
1.6 Manfaat Penelitian	2
1.7 Sistematika Penulisan	2
BAB II LANDASAN TEORI	
2.1 Optical Character Recognition	5
2.2 Tesseract OCR	7
2.2.1 Arsitektur Tesseract	7
2.2.2 Pencarian Teks-Line dan Kata	8
2.2.3 Baseline Filtering	9

2.2.4 Perkiraan Ketinggian X Pada Teks	9
2.2.5 Character Chopping atau Pemotongan Karakter	10
2.2.6 Pemisahan Karakter Terhubung.....	10
2.3 Sistem.....	11
2.3.1 Klasifikasi Sistem	11
2.3.2 Karakteristik Sistem.....	12
2.3.3 Perancangan Sistem	14
2.4 Citra Digital	15
2.5 Metode Pengembangan Perangkat Lunak Model Prototype.....	15
2.6 Unified Modelling Language (UML)	16
2.7 Penelitian Terdahulu	19

BAB III METODE PENELITIAN

3.1 Metode Pengembangan Perangkat Lunak	21
3.1.1 Metode Pengumpulan Data	21
3.1.2 Metode yang Digunakan.....	22
3.1.2.1 Optical Character Recognition	22
3.1.2.2 Tesseract-OCR	22
3.1.2.3 Ekstraksi Teks	22
3.1.2.4 Adaptive Thresholding	22
3.1.2.5 Page Layout Analysis.....	22
3.1.2.6 Recognize Word	22
3.1.2.7 X-Height Fix.....	23
3.1.3 Model Pengembangan Perangkat Lunak	23
3.2 Pengumpulan Kebutuhan.....	23
3.2.1 Analisis Sistem yang Berjalan.....	24
3.2.2 Analisis Sistem yang Diajukan.....	24
3.3 Perancangan Prototype	24
3.3.1 Desain Use Case	25
3.3.2 Desain Activity Diagram	25
3.3.3 Desain Rancangan Aplikasi.....	26
3.4 Evaluasi Prototype	27
3.5 Proses Kerja Aplikasi.....	27

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian.....	29
4.1.1 Instalasi Perangkat Lunak.....	29
4.1.2 Penggunaan Perangkat Lunak	30
4.1.3 Pemilihan Gambar dan Bahasa.....	30
4.1.4 Tampilan Hasil Scan.....	31
4.1.5 Tampilan Simpan Tulisan.....	31
4.1.6 Hasil Simpan Tulisan	32
4.2 Pembahasan	33
4.2.1 Pengujian Sistem	33
4.2.2 Pengujian Program	33
4.2.2.1 Pengujian Instalasi Program	33
4.2.2.2 Pengujian Tampilan Utama Program	36
4.2.2.3 Pengujian Input Gambar dan Bahasa	38
4.2.2.4 Pengujian Ekstraksi Teks dari gambar	40
4.2.2.5 Pengujian Penyimpanan Hasil Ekstraksi Teks	42
4.2.2.6 Pengujian Hasil Penyimpanan Ekstraksi Teks	44
4.3 Kelebihan dan Kelemahan Aplikasi	47
4.3.1 Kelebihan Aplikasi	47
4.3.2 Kekurangan Aplikasi	47

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan.....	49
5.2 Saran	49

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Proses Pengenalan Oleh OCR Secara Umum.....	6
Gambar 2.2 Arsitektur Tesseract	7
Gambar 2.3 Contoh Tulisan Dengan <i>Baseline</i> Miring	9
Gambar 2.4 Contoh <i>Character Chopping</i>	10
Gambar 2.5 Ilustrasi Pemisahan Karakter Terhubung.....	10
Gambar 2.6 Bentuk Umum Sistem.....	14
Gambar 2.7 Model Prototype	15
Gambar 3.1 Use Case Diagram Untuk Aplikasi Yang Diajukan	25
Gambar 3.2 Activity Diagram Ekstraksi Teks.....	26
Gambar 3.3 Rancangan Tampilan Utama Aplikasi	27
Gambar 4.1 Instalasi Perangkat Lunak.....	29
Gambar 4.2 <i>Icon</i> Aplikasi Setelah Terinstal.....	29
Gambar 4.3 Tampilan Utama Aplikasi.....	30
Gambar 4.4 Tampilan Aplikasi Setelah Gambar dan Bahasa Dipilih	30
Gambar 4.5 Tampilan Aplikasi Dengan Hasil Ekstraksi Teks	31
Gambar 4.6 Dialog Simpan Tulisan	31
Gambar 4.7 Hasil Simpan Teks Dalam Aplikasi <i>Notepad</i>	32
Gambar 4.8 Uji Instalasi	34
Gambar 4.9 Uji Instalasi Lanjutan.....	35
Gambar 4.10 Uji Tampilan Utama	36
Gambar 4.11 Uji Tampilan Utama Lanjutan	37
Gambar 4.12 Uji Input Gambar dan Bahasa.....	38
Gambar 4.13 Uji Input Gambar dan Bahasa Lanjutan	39
Gambar 4.14 Uji Ekstraksi Teks.....	40
Gambar 4.15 Uji Ekstraksi Teks Lanjutan	41
Gambar 4.16 Uji Penyimpanan Hasil Ekstraksi Teks	42
Gambar 4.17 Uji Penyimpanan Hasil Ekstraksi Teks Lanjutan	43

Gambar 4.18 Uji Hasil Penyimpanan Ekstraksi Teks	45
Gambar 4.19 Uji Hasil Penyimpanan Ekstraksi Teks Lanjutan	46

DAFTAR TABEL

Tabel 2.1 Simbol Use Case Diagram.....	17
Tabel 2.2 Simbol Activity Diagram	18
Tabel 2.3 Tabel Penelitian Terdahulu.....	19
Tabel 4.1 Pengujian Instalasi Program.....	33
Tabel 4.2 Pengujian Tampilan Utama Program	36
Tabel 4.3 Pengujian Input Gambar dan Bahasa	38
Tabel 4.4 Pengujian Ekstraksi Teks dari Gambar	40
Tabel 4.5 Pengujian Penyimpanan Hasil Ekstraksi Teks	42
Tabel 4.6 Pengujian Hasil Penyimpanan Ekstraksi Teks	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di zaman sekarang ini, informasi berbasis tulisan merupakan sesuatu yang tidak dapat dipisahkan dari kehidupan sehari – hari kita. Berbagai informasi dapat ditemukan dengan mudah, mulai dari media cetak, hingga media elektronik.

Informasi yang tersedia dapat berasal dari mana saja. Mudahnya penyebaran informasi khususnya melalui media internet memperbolehkan kita untuk mendapatkan informasi dari berbagai penjuru negara. Walaupun begitu, tidak semua informasi yang beredar dapat digunakan dengan mudah.

Banyaknya informasi yang beredar dalam media citra *digital* dapat menyulitkan seseorang untuk mengambil tulisan yang ada di dalam sebuah citra *digital*. Selain itu, tidak seluruh dokumen beredar dalam bentuk yang mudah diolah.

Banyak informasi yang beredar di *internet* dalam bentuk sebuah *slide* presentasi. *Slide* tersebut biasanya tidak dapat diambil teksnya secara langsung. Kesulitan pengambilan informasi berbentuk tulisan dalam sebuah gambar menghambat banyak mahasiswa yang sedang melakukan penelitian. Banyaknya dokumen yang berbentuk gambar dapat menghambat proses pengambilan informasi yang ada di dalam gambar tersebut.

Maka dari itu, dalam kesempatan kali ini akan dibuat sebuah aplikasi yang dapat mengambil tulisan yang terdapat dalam sebuah gambar.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, masalah yang dihadapi adalah banyaknya waktu yang terbuang untuk mengetik kembali tulisan yang ada dalam sebuah gambar.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah diatas, maka penelitian ini akan merancang suatu aplikasi berbasis *Optical Character Recognition* guna membantu mengambil tulisan yang ada dalam sebuah gambar.

1.4 Ruang Lingkup

Agar penelitian dapat menjadi lebih terarah, maka ruang lingkup yang ditentukan adalah sebagai berikut:

1. Penelitian ini menggunakan *Tesseract* sebagai *library* yang digunakan untuk mendapatkan teks dari sebuah gambar.
2. Penelitian ini hanya membahas hasil teks yang diambil dari dalam bentuk gambar dengan ekstensi .jpg, .png.
3. Penelitian ini dilakukan untuk membuat aplikasi berbasis *Optical Character Recognition* untuk membantu pengambilan teks dalam sebuah citra digital.
4. Pengambilan data untuk penelitian ini akan dilakukan di DSC IIB Darmajaya.

1.5 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membuat aplikasi berbasis *Optical Character Recognition* untuk mengambil informasi tulisan yang ada dalam sebuah gambar.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk dapat mempermudah proses pengambilan tulisan dari sebuah gambar.

1.7 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan landasan teori yang sesuai dengan topik pembahasan skripsi.

BAB III METODE PENELITIAN

Bab ini menjelaskan tentang tata cara penelitian, baik metode yang digunakan dan tahapan – tahapan pembuatan aplikasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi uraian dan tata cara pemakaian dan pengujian aplikasi ekstraksi teks pada gambar berbasis *Optical Character Recognition*.

BAB V KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang kesimpulan dari permasalahan yang ada dan solusi baru yang ditemukan dalam penelitian ini dan saran terhadap objek penelitian untuk digunakan pada penelitian selanjutnya.

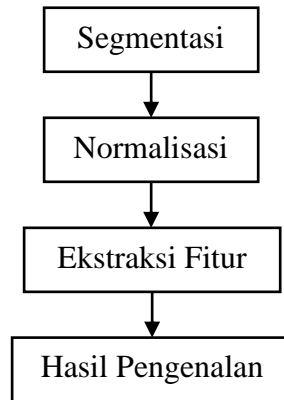
BAB II

LANDASAN TEORI

2.1. Optical Character Recognition

Pengenalan karakter dengan menggunakan alat optik (*optical character recognition*) adalah sebuah sistem komputer yang dapat membaca karakter termasuk (huruf), baik yang berasal dari sebuah alat pencetak (*printer* atau mesin ketik) maupun yang berasal dari tulisan tangan. Adanya sistem pengenalan huruf ini akan meningkatkan fleksibilitas ataupun kemampuan dan kecerdasan sistem komputer. Adanya sistem pengenalan huruf yang cerdas akan sangat membantu usaha besar - besaran yang saat ini dilakukan banyak pihak yakni usaha digitalisasi informasi dan pengetahuan, misalnya dalam pembuatan koleksi pustaka digital, koleksi sastra kuno digital, dll. Dengan adanya pengenalan huruf juga akan memudahkan penanganan pekerjaan yang memakai *input* tulisan seperti penyortiran surat di kantor pos, pemasukan data buku di perpustakaan, dll. Selain itu adanya sistem pengenalan huruf otomatis maka *user* dapat lebih leluasa memasukkan data karena *user* tidak harus memakai papan ketik tetapi bisa menggunakan pena elektronik untuk menulis sebagaimana *user* menulis di kertas. (Priyatma dkk, 2004) Dalam pengenalan pola otomatis, sistem pengenalan pola mencoba mengenali apakah citra masukan yang diterima cocok dengan salah satu citra yang telah ditentukan. Sistem ini misalnya dipakai untuk mendeteksi sidik jari, tanda tangan, bahkan wajah seseorang. OCR dapat dipandang sebagai bagian dari pengenalan otomatis yang lebih luas yakni pengenalan pola otomatis (*automatic pattern recognition*). Ada banyak pendekatan yang dapat dipakai untuk mengembangkan pembuatan pengenalan pola otomatis antara lain memakai pendekatan numerik, statistik, sintaktik, neural dan, aturan produksi (*rule - based*). Secara umum metode - metode tersebut dapat digolongkan menjadi dua kelompok metode yakni metode berbasis statistik dan metode berbasis struktur. Dalam metode yang berbasis statistik, setiap pola ditransformasi ke dalam vektor yang memakai ukuran dan karakteristik tertentu. Karakteristik ini seringkali lebih bersifat statistik misalnya distribusi *pixel* ataupun jarak *pixel*. Sedangkan dalam metode yang berbasis struktur, setiap pola yang diproses dinyatakan sebagai gabungan beberapa struktur elementer. Pengenalan selanjutnya dilakukan dengan

mencocokkan komposisi struktur elementer dengan struktur 8 yang sudah disimpan memakai aturan tertentu misalnya memakai pendekatan teori bahasa formal dan automata. (Sukmawan, 2008) Proses pengenalan karakter yang dilakukan oleh OCR disajikan pada gambar di bawah ini.



Gambar 2.1 Proses pengenalan oleh OCR secara umum

Proses segmentasi bertujuan untuk memisahkan wilayah (region) objek dengan wilayah latar belakang agar objek dalam citra mudah dianalisis dalam rangka mengenali objek. Dengan demikian citra yang besar terdiri dari objek karakter dapat disegmentasi menjadi masing-masing karakter. Proses selanjutnya adalah normalisasi, yang di dalamnya masih terdapat 2 proses lagi yaitu:

- *Scaling*, adalah fungsi untuk mengubah ukuran suatu gambar dimana *scaling* merupakan istilah yang cenderung untuk memperbesar gambar, dan *shrink* untuk memperkecil gambar.
- *Thinning*, adalah operasi morfologi yang digunakan untuk menghapus piksel *foreground* yang terpilih dari gambar biner, biasanya digunakan untuk proses mencari tulang dari sebuah objek.

Langkah selanjutnya adalah ekstraksi fitur (*feature extraction*). *Feature Extraction* adalah suatu proses analisis citra dalam mengidentifikasi sifat-sifat yang melekat dari tiap-tiap karakter atau disebut juga dengan fitur dari sebuah objek yang terdapat dalam citra. Karakteristik ini digunakan dalam mendeskripsikan sebuah objek atau atribut dari sebuah obyek, kemudian fitur yang dimiliki oleh karakter dapat digunakan sebagai proses *recognition*.

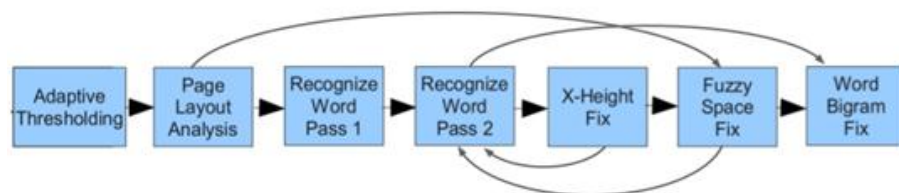
Setelah langkah-langkah di atas selesai dilakukan, maka OCR siap untuk melakukan tahap pengenalan dan akan memberikan output atau hasil pengenalan karakter angka maupun huruf.

2.2. Tesseract OCR

Tesseract adalah *engine open source* OCR yang awalnya dikembangkan HP (Hewlett-Packard) antara tahun 1984 dan 1994. Tesseract dimulai dari sebuah proyek penelitian PhD di HP Laboratorium, Bristol oleh Ray Smith. Setelah penelitian bersama antara HP Labs Bristol dan Scanner HP divisi di Colorado, tesseract secara signifikan memimpin dalam akurasi atas mesin komersial tetapi tidak menjadi produk. (Ray Smith, 2005, p1) Tahap perkembangan berikutnya kembali di HP Lab Bristol sebagai investigasi OCR untuk kompresi. Pada akhir proyek ini, pada tahun 1994, pengembangan berhenti sepenuhnya. Mesin ini dikirim ke UNLV (University Nevada Las Vegas) pada tahun 1995 menjalani tes akurasi tahunan OCR. (Ray Smith, 2005). Setelah tahun 2006, Tesseract dilepas oleh HP untuk dapat digunakan secara bebas. Sejak saat itu, Tesseract dikembangkan secara luas oleh Google dan dirilis di bawah lisensi *Apache 2.0* dan dapat diakses di <http://code.google.com/p/tesseract-ocr>. (Ray Smith, 2005)

2.2.1 Arsitektur Tesseract

Gambar berikut adalah ilustrasi arsitektur Tesseract.



Gambar 2.2 Arsitektur Tesseract

Tesseract OCR mengasumsikan input yang diterima berupa sebuah binary image. Pertama, analisis dilakukan pada komponen terhubung/*Connected Component (CC)* untuk menemukan di mana outline komponen disimpan. Pada tahap ini outlines dikumpulkan bersama menjadi *blob*. *Blob* disusun menjadi baris teks, sedangkan garis dan region dianalisis untuk pitch tetap dan teks proporsional. Baris teks dipecah menjadi kata-kata berbeda menurut jenis spasi karakter. Teks dengan pitch tetap dibagi menjadi sel-sel karakter. Teks proporsional dipecah menjadi kata-kata dengan menggunakan ruang pasti dan ruang *fuzzy*. Pengenalan kata pada image dilakukan pada dua tahap proses yang disebut *two-pass* (Smith, 2009).

Pada *pass* pertama dilakukan untuk mengenali masing-masing kata pada gilirannya. Kata-kata yang sukses pada *pass* pertama yaitu kata-kata yang terdapat di kamus dan tidak ambigu kemudian diteruskan ke adaptive classifier sebagai data pelatihan. Begitu adaptive classifier memiliki sampel yang cukup, adaptive classifier ini dapat memberikan hasil klasifikasi bahkan pada *pass* pertama. Proses *pass* kedua dilakukan untuk mengenali kata-kata yang mungkin saja kurang dikenali atau terlewatkan pada *pass* pertama, pada tahap ini adaptive classifier telah memperoleh informasi lebih dari *pass* pertama. Tahap terakhir menyelesaikan ruang *fuzzy* dan memeriksa hipotesis alternatif pada ketinggian- x untuk mencari teks dengan huruf kecil. Tesseract dirancang untuk mengenali teks putih di atas latar hitam dan teks hitam di atas latar putih. Hal ini menyebabkan rancangan mengarah pada analisis komponen terhubung/*connected component* (CC) dan operasi pada outline komponen. Langkah pertama setelah analisis CC ialah menemukan *blob* pada region teks. Sebuah *blob* merupakan unit putatif yang dapat diklasifikasikan, yang mana bisa satu atau lebih komponen-komponen yang saling tumpang tindih secara horizontal.

2.2.2 Pencarian Teks-Line dan Kata

Algoritma *line finding* dirancang supaya halaman yang miring dapat dikenali tanpa harus *de-skew* (proses untuk mengubah halaman yang miring menjadi tegak lurus) sehingga tidak menurunkan kualitas gambar. Kunci bagian proses ini adalah *blob filtering* dan *line construction*. (Smith, 2009, p1). Filtered *blob* lebih cenderung cocok dengan model non-overlapping, parallel, tetapi berupa garis-garis miring (*sloping line*). Pemrosesan *blob* oleh koordinat x memungkinkan untuk menetapkan *blob* ke sebuah baristeks yang unik. Sementara penelusuran kemiringan di seluruh halaman, dengan banyak mengurangi bahaya penugasan ke baris teks yang salah dengan adanya kemiringan (*skew*). Setelah *blob* tersaring ditetapkan ke garis, sebuah median terkecil dari kotak-kotak yang cocok

digunakan untuk memperkirakan baseline , dan *blob* yang sudah difilter dengan baik dipasang kembali ke garis yang sesuai. (Ray Smith, 2009, p2) Langkah terakhir dari proses pembuatan garis (line creation) adalah menggabungkan *blob* yang overlapping, menempatkan diacritical marks dengan dasar yang tepat, dan menghubungkan bagian-bagian dari beberapa karakter yang rusak secara benar. (Ray Smith, 2009, p2)

2.2.3 Baseline Fitting

Setelah baris teks telah ditemukan, garis pangkal (baseline) dicocokkan secara lebih tepat menggunakan *quadratic spline*. Hal ini merupakan salah satu kelebihan sistem OCR dan memungkinkan tesseract untuk menangani halaman dengan garis pangkal (baseline) yang miring. (Ray Smith, 2009, p2). Baseline dicocokkan oleh partisi *blob* menjadi beberapa kelompok dengan sebuah perpindahan kontinu yang cukup layak untuk garis pangkal lurus yang asli. *Quadratic spline* dicocokkan ke partisi yang paling padat (diasumsikan sebagai baseline) dengan kuadrat terkecil. *Quadratic spline* memiliki keuntungan bahwa perhitungan ini cukup stabil tetapi merugikan 15 jika muncul diskontinuitas ketika beberapa segmen spline diperlukan. Dalam hal ini, *cubic spline* bekerja lebih baik. Contoh tulisan dengan *baseline* miring tampak pada gambar dibawah ini. (Ray Smith, 2009, p2).



Gambar 2.3 Contoh tulisan dengan *Baseline* miring

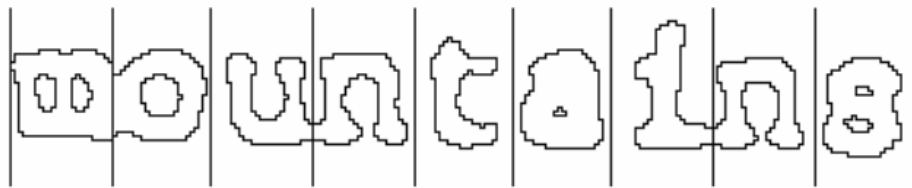
2.2.4 Perkiraan Ketinggian X Pada Teks

Setelah menemukan baris teks dan menyusun blok *blob* menjadi baris-baris, Tesseract mengestimasi ketinggian-x untuk setiap baris teks. Pertama, algoritma estimasi ketinggian-x menentukan batas-batas maksimum dan minimum dari ketinggian-x yang dapat diterima berdasarkan ukuran garis inisial yang dihitung untuk blok. Kemudian, setiap baris secara terpisah, ketinggian bounding box blob terjadi pada garis dikuantisasi dan dikumpulkan menjadi sebuah histogram. Dari

histogram ini, algoritma pencarian ketinggian-x mencari ketinggian dua mode yang paling sering terjadi yang cukup jauh terpisah untuk menjadi ketinggian-x dan ketinggian- ascender. Untuk mengantisipasi *noise*, algoritma memastikan mode ketinggian yang diambil menjadi ketinggian-x dan ketinggian - ascender memiliki jumlah yang cukup atau kejadian - kejadian relatif terhadap jumlah keseluruhan *blob* pada baris.

2.2.5 Character Chopping atau Pemotongan Karakter

Tesseract menguji baris teks (*text line*) untuk menentukan apakah mereka merupakan *fixed pitch*. Bila ditemukan *fixed pitch text*, tesseract memotong kata-kata menjadi karakter-karakter. Contoh kata yang telah dipotong menjadi karakter ada pada gambar 2.4 berikut ini. (Ray Smith, 2009, p2).

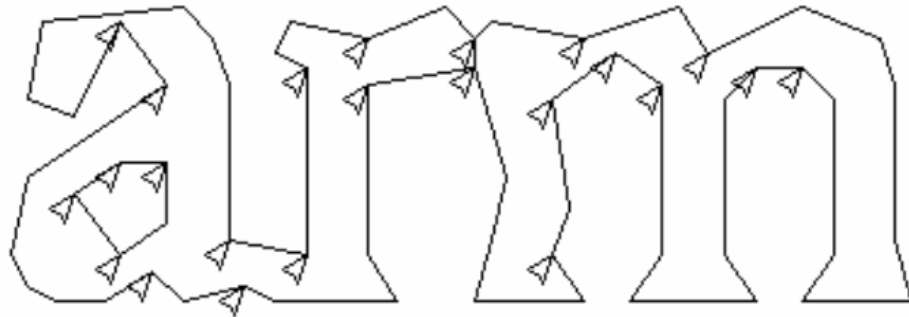


Gambar 2.4 Contoh *Character Chopping*

2.2.6 Pemisahan Karakter Terhubung

Apabila hasil dari pengenalan kata tidak memuaskan, tesseract berusaha untuk memperbaiki hasil dengan memisahkan *blob* dengan keyakinan terburuk dari pengklasifikasian (*classifier*) karakter. Kandidat untuk titik-titik pemisahan ditemukan dari simpul cekung dari pendekatan poligonal outline dan mungkin saja terdapat titik cekung berlawanan lainnya atau segmen garis. Ini akan menghabiskan sampai 3 pasang titik pemotongan

untuk memisahkan karakter yang terhubung dari set ASCII.



Gambar 2.5 Ilustrasi Pemisahan Karakter Terhubung

Gambar 2.5 di atas menunjukkan satu set calon titik potong (*chop points*) dengan tanda panah dan potongan terpilih sebagai sebuah garis melintasi kerangka dimana huruf 'r' bersentuhan dengan 'm'. (Ray Smith, 2005, p3).

2.3 Sistem

Karakteristik sebuah sistem adalah terdiri dari bagian-bagian yang saling berkaitan dan beroperasi untuk mencapai suatu tujuan. Sebuah sistem bukanlah seperangkat unsur yang tersusun secara tidak teratur, namun sistem terdiri dari unsur yang dapat dikenal untuk saling melengkapi karena memiliki maksud, tujuan dan sasaran tertentu.

Sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau sub sistem yang berinteraksi untuk mencapai suatu tujuan. (Jogiyanto, 2005)

2.3.1 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, di antaranya adalah sebagai berikut :

1. Sistem Abstrak (*abstract system*) dan Sistem Fisik (*physical system*)

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Sedangkan sistem fisik adalah sistem yang secara fisik dapat dilihat.

2. Sistem Deterministik (*deterministic system*) dan Sistem Probabilistik (*probabilistic system*)

Sistem *deterministic* adalah suatu sistem yang operasinya dapat diprediksi secara tepat. Sedangkan sistem probabilistik adalah sistem yang tidak dapat diramal dengan pasti karena mengandung unsur *probabilistic*.

3. Sistem Tertutup (*closed system*) dan Sistem Terbuka (*open system*)
Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Sedangkan sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

4. Sistem Alamiah (*natural system*) dan Sistem Buatan manusia (*human made system*)

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat oleh manusia). Sedangkan sistem buatan manusia adalah sistem yang dibuat oleh manusia.

5. Sistem Sederhana dan Sistem kompleks

Berdasarkan tingkat kerumitannya, sistem dibedakan menjadi sistem yang sederhana dan sistem yang kompleks.

2.3.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung dan sasaran, diantaranya:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sub sistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apa pun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Sub sistem

Bagian dari Sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarnya masing-masing.

5. Penghubung Sistem

Penghubung merupakan media penghubung antara satu sub sistem dengan sub sistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu sub sistem ke sub sistem yang lainnya. Keluaran dari satu sub sistem akan menjadi masukan untuk sub sistem yang lainnya dengan melalui penghubung.

6. Masukan Sistem

Masukan (input) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Masukan perawatan adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal adalah energi yang diproses untuk didapatkan keluaran.

7. Keluaran Sistem

Keluaran (output) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Misalkan untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

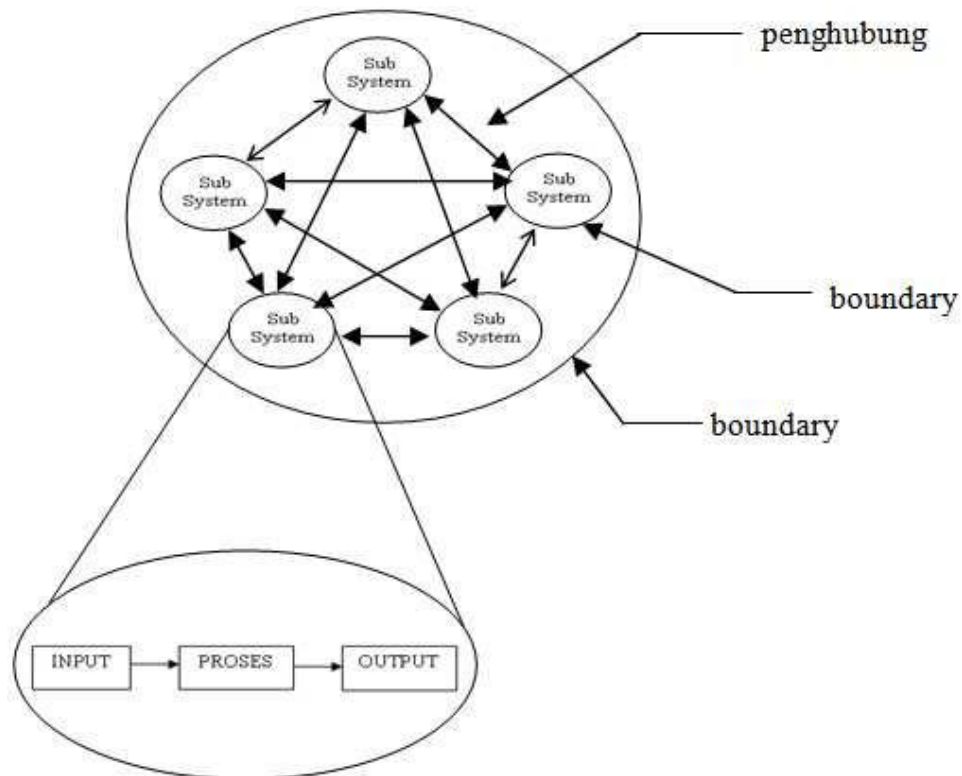
8. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

9. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan berguna. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

Gambar 2.6 berikut ini adalah ilustrasi dari karakteristik sistem.



Gambar 2.6 Bentuk Umum Sistem

Bentuk umum dari sistem terdiri dari atas masukan (input), proses, dan keluaran (output). Dalam bentuk umum sistem ini biasa melakukan satu atau lebih masukan yang akan diproses dan menghasilkan keluaran sesuai dengan yang direncanakan sebelumnya (Jogiyanto, 2005).

2.3.3 Perancangan Sistem

Desain Sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

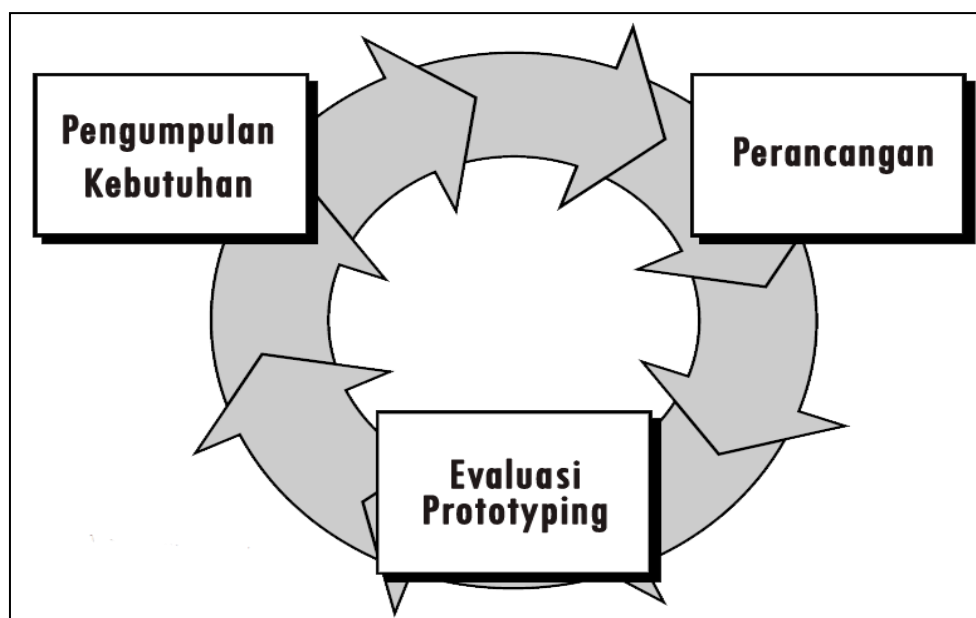
Dari penjelasan tersebut maka penulis dapat mengambil kesimpulan bahwa perancangan (Desain) adalah proses penerjemahan kebutuhan pemakai informasi dengan melakukan penggambaran, perencanaan, dan pembuatan sketsa ke dalam kesatuan yang berfungsi. (Jogiyanto, 2005)

2.4 Citra Digital

Citra digital adalah suatu matriks yang terdiri dari baris dan kolom dimana setiap pasang indeks baris dan kolom menyatakan suatu titik pada citra. Nilai dari setiap matriks menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra atau piksel. Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut (T Suyono dkk, 2009).

2.5 Metode Pengembangan Perangkat Lunak Model Prototype

Paradigma *Prototyping* dimulai dengan pengumpulan kebutuhan, pengembang bertemu dengan pengguna dan mengidentifikasi objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar di mana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat. (Pressman, 2012), Proses evaluasi prototyping dapat dilihat pada gambar 2.7 berikut :



Gambar 2.7 Model Prototype

Tahap–tahap rekayasa *software* dalam *Prototype model* pada gambar 2.7 di atas adalah sebagai berikut :

1) Pengumpulan kebutuhan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan. Selanjutnya peneliti akan melakukan analisis terhadap data apa saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis kebutuhan notifikasi *service* berkala.

2) Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan.

3) Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. *Software* yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan/ kerusakan yang tidak ditemukan pada langkah sebelumnya

Kelebihan dari *Prototype Model* adalah sebagai berikut :

- 1) End user dapat berpartisipasi aktif.
- 2) Penentuan kebutuhan lebih mudah diwujudkan.
- 3) Mempersingkat waktu pengembangan *software*.

Kekurangan dari *Prototype Model* adalah sebagai berikut:

- 1) Proses analisis dan perancangan terlalu singkat.
- 2) Mengesampingkan alternatif pemecahan masalah.
- 3) Biasanya kurang fleksibel dalam menghadapi perubahan.
- 4) *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
- 5) *Prototype* terlalu cepat selesai.

2.6 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin (Rosa dan Shalahuddin, 2016, p.133). Jenis-jenis diagram UML adalah sebagai berikut :


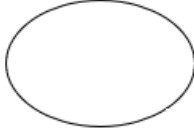

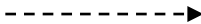
1. Use Case Diagram

Use case diagram adalah gambar dari beberapa atau seluruh aktor dan use case dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case mempresentasikan sebuah interaksi antara actor dan sistem.

Dalam use case diagram terdapat istilah seperti aktor, use case dan case relationship. Penjelasan simbol use case diagram ditunjukkan pada tabel 2.1.

Tabel 2.1. Simbol Use Case Diagram




Simbol	Keterangan
--------	------------

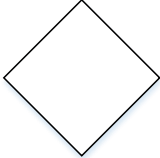
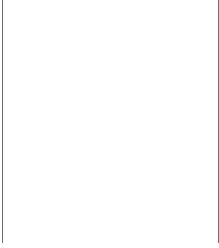
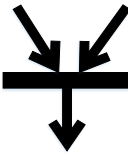
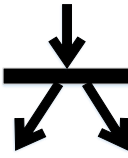
	Aktor: Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	Use case: perangkat tertinggi dari fungsionalitas yang dimiliki sistem.
	Association: adalah relasi antara actor dan use case.
	Generalisasi: untuk memperlihatkan struktur pewaris yang terjadi.

2. Activity Diagram

Activity Diagram menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi. Activity Diagram berupa flow chart yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam activity diagram ditunjukkan pada tabel 2.2 berikut ini.

Tabel 2.2. Simbol Activity Diagram.

Simbol	Keterangan
	Activity: Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	Initial Node: Bagaimana objek dibentuk atau diawali
	Activity Final Node: Bagaimana objek dibentuk dan diakhiri.

	<p>Decision: Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.</p>
<p>Nama Swimlane</p> 	<p>Swimlane: Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.</p>
	<p>Join: Digunakan untuk menunjukkan kegiatan yang digabungkan.</p>
	<p>Fork: Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel</p>

3. Sequence Diagram

Sequence Diagram menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Dalam sequence diagram terdapat 2 simbol yaitu: Actor, untuk menggambarkan pengguna sistem. Lifeline, untuk menggambarkan kelas dan objek.

4. Class Diagram

Class diagram menggambarkan struktur data dan deskripsi class, package, dan objek beserta hubungan satu sama lain. Class diagram berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Class memiliki 3 area pokok yaitu nama, atribut dan metode.

2.7 Penelitian Terdahulu

Penelitian terdahulu yang berkaitan dengan penelitian *Optical Character Recognition* dapat dilihat pada tabel 2.3.

Tabel 2.3. Tabel Penelitian Terdahulu

Nama	Judul	Terbit/Tahun	Keterangan
Muhammad Fajar	Perancangan Aplikasi Penelusuran Informasi Persuratan Berbasis Optical Character Recognition (OCR)	Universitas Islam Negeri Syarif Hidayatullah, 2015: Jakarta	Merancang sistem yang dapat membantu penelusuran informasi terhadap surat yang sudah diarsipkan berbasis OCR.
Fajar Aprilianto Susanto, Yos Richard Beeh, S.T., M.Cs.	Pemanfaatan Teknologi Optical Character Recognition (OCR) Untuk Mengenali Alfabet Yunani Berbasis Android	Universitas Kristen Satya Wacana, 2015: Salatiga	Menggunakan OCR sebagai alat bantu input Bahasa Yunani untuk diterjemahkan ke dalam Bahasa Indonesia.
R. Sandhika Galih A., Erik, Muhammad Lukman Hakim	Penerapan Teknik Ocr (Optical Character Recognition) Pada Aplikasi Terjemahan Kitab Fiqih Safinah An-Naja Menggunakan ReadIRIS	Universitas Pasundan Bandung, 2014: Bandung	Penggunaan OCR ReadIRIS untuk mendigitalisasi Kitab Fiqih <i>Safinah An-Naja</i>
Dwi Cahyo Nugroho, Mahmud Dwi Sulistiyo, Bedy Purnama	Optical Character Recognition Pada Smart Phone Menggunakan Contour Analysis Dan Feature	Universitas Telkom, 2014: Bandung	Penerapan OCR pada <i>smartphone</i> .

	Preselection		
Ossy D.E.W, T.M. Zaini, Frizka	Penerapan Optical Character Recognition Pada Penerjemah Inggris ↔ Indonesia Mobile Berbasis Augmented Reality	Jurusan Sistem Informasi, Universitas Lampung, Jurusan Sistem Informasi - Fakultas Ilmu Komputer Informatics & Business Institute Darmajaya, 2013: Bandar Lampung	Penggunaan OCR sebagai alat input aplikasi penerjemah Bahasa Inggris – Bahasa Indonesia berbasiskan Augmented Reality pada sebuah <i>smartphone</i> .

BAB III

METODE PENELITIAN

3.1 Pengumpulan Kebutuhan Perangkat Lunak

Penelitian ini menggunakan metode pengembangan perangkat lunak *prototype* untuk pembuatan Aplikasi Optical Character Recognition Untuk Ekstraksi Teks Pada Gambar yang dimulai dari tahap pengumpulan kebutuhan, perancangan, dan evaluasi *prototype*.

3.1.1 Metode Pengumpulan Data

Pengumpulan data merupakan tahap yang sangat penting dalam mendapatkan informasi untuk pembuatan aplikasi. Data yang objektif dan relevan dengan topik pembahasan merupakan hal yang penting dalam suatu penelitian. Pada kesempatan ini, peneliti menggunakan beberapa metode pengumpulan data, antara lainnya:

1. Studi Literatur

Metode ini dilakukan dengan mencari sumber – sumber dan data – data yang mendukung dan diperlukan dalam pengembangan.

Pengumpulan data yang dilakukan yaitu mempelajari buku-buku serta literatur-literatur terkait teori mengenai *Optical Character Recognition*, model pengembangan perangkat lunak, dan penelitian terkait dengan judul yang diangkat.

2. Observasi

Objek penelitian yang dipilih oleh peneliti adalah Mahasiswa Teknik Informatika IIB Darmajaya. Pengamatan akan dilakukan untuk mengetahui kebutuhan apa saja yang diperlukan oleh subjek penelitian untuk mempermudah proses pengambilan informasi dari sebuah gambar.

3. Wawancara

Metode pengumpulan data ini dilakukan dengan cara bertemu langsung dan melakukan tanya jawab atau wawancara dengan pihak yang berkaitan, pihak yang berkaitan pada penelitian kali ini adalah Mahasiswa Teknik Informatika IIB Darmajaya. Wawancara dilakukan

untuk mencari tahu apa saja yang diperlukan oleh subjek untuk mempermudah proses pengambilan informasi dari sebuah gambar.

3.1.2 Metode yang Digunakan

Aplikasi yang dibuat menggunakan metode *optical character recognition* menggunakan Tesseract-OCR dan dikembangkan dalam *IDE Visual Studio Community 2017* menggunakan bahasa pemrograman C#.

3.1.2.1 Optical Character Recognition

Optical Character Recognition merupakan sebuah teknologi yang memungkinkan sebuah komputer mengenali karakter melalui mekanisme optik.

3.1.2.2 Tesseract-OCR

Tesseract-OCR adalah *engine open source optical character recognition* yang dikembangkan oleh Ray Smith. Tesseract-OCR dipilih untuk digunakan sebagai *engine optical character recognition* pada kali ini karena akurasi dan kecepatan *scanning* pada gambar yang baik. Aplikasi ini akan dikembangkan menggunakan Tesseract-OCR versi 3.0.2.

3.1.2.3 Ekstraksi Teks

Proses ekstraksi teks yang dilakukan oleh Tesseract-OCR dapat dilihat pada Bab II pada bagian 2.1.1 Arsitektur Tesseract.

3.1.2.4 Adaptive Thresholding

Tesseract-OCR melakukan *adaptive thresholding* pada gambar input. *Adaptive thresholding* dilakukan agar *background* dan *foreground* yaitu tulisan yang ingin diekstrak dapat teridentifikasi dengan baik.

3.1.2.5 Page Layout Analysis

Tesseract-OCR menganalisa tata letak halaman (*Page layout analysis*) untuk mengidentifikasi dimana letak tulisan dalam sebuah halaman.

3.1.2.6 Recognize Word

Recognize Word adalah tahapan dimana Tesseract-OCR mendeteksi karakter yang ada dalam gambar. *Recognize Word*

dicapai Tesseract-OCR dengan melakukan *line finding*, *baseline fitting*, dan *character chopping*.

3.1.2.7 X-Height Fix

Tesseract-OCR akan memperkirakan ketinggian-x (*X-Height Fix*) untuk memisahkan antara huruf kapital dan huruf kecil.

3.1.3 Model Pengembangan Perangkat Lunak

Aplikasi yang dibuat menggunakan model pengembangan perangkat lunak *prototyping* yang dimulai dari tahap pengumpulan kebutuhan, perancangan, dan evaluasi prototype.

3.2 Pengumpulan Kebutuhan

Data dan kebutuhan perangkat lunak maupun keras yang telah diperoleh pada tahap sebelumnya kemudian dianalisa dan menghasilkan sebuah *user requirement*. Adapun analisa kebutuhan perangkat lunak dan perangkat keras yang diperoleh adalah sebuah kebutuhan yang akan digunakan untuk membangun Aplikasi *Optical Character Recognition* untuk Ekstraksi Teks pada Gambar adalah sebagai berikut:

1. Kebutuhan Perangkat Lunak:

Spesifikasi minimal perangkat lunak yang digunakan untuk mendukung pembuatan program ini adalah sebagai berikut:

- a. OS Windows 10
- b. IDE Visual Studio Community 2017
- c. Tesseract-OCR

Perangkat lunak ini akan dikembangkan menggunakan Visual Studio Community 2017 dan menggunakan tambahan tesseract-ocr untuk fungsi OCR.

2. Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk mendukung pembuatan program adalah sebagai berikut:

a. Spesifikasi komputer yang digunakan dalam pembuatan aplikasi adalah:

1. Prosesor *Intel(R) Core(TM) i5-2500K*
2. RAM 12GB
3. *Harddisk 320GB*
4. Kartu Grafis *Radeon RX550 2GB*

Komponen diatas adalah komponen yang digunakan penulis untuk merancang dan membangun aplikasi *Optical Character Recognition* untuk Ekstraksi Teks pada Gambar.

3.2.1 Analisis Sistem yang Berjalan

Setelah melakukan kegiatan observasi dan wawancara pada mahasiswa IIB Darmajaya, ditemukan kurangnya penggunaan aplikasi berbasis *optical character recognition* untuk membantu mempermudah pengambilan tulisan dari sebuah gambar digital. Hasil observasi banyak menunjukkan mahasiswa masih mengambil data secara manual, yaitu dengan mengetikkan kembali tulisan yang ada dalam sebuah gambar. Hal ini memakan banyak waktu karena mahasiswa perlu mengetik ulang dan mengecek kembali hasil ketikan yang sudah selesai.

3.2.2 Analisis Sistem yang Diajukan

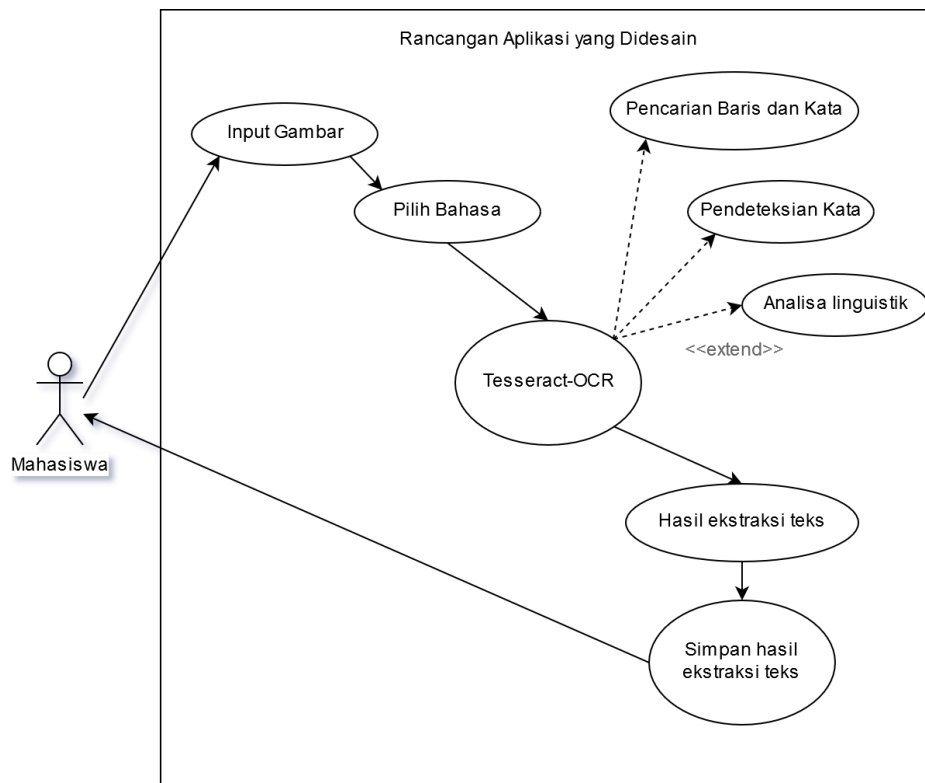
Dari hasil analisa sistem yang berjalan maka diajukanlah aplikasi berbasis *Optical Character Recognition* untuk Ekstraksi Teks pada Gambar. Berdasarkan hasil analisa, dibutuhkan perangkat lunak yang dapat mempercepat proses pengambilan tulisan pada gambar dan dengan akurasi yang cukup tinggi. Dengan dibangunnya aplikasi ini diharapkan dapat membantu mahasiswa IIB Darmajaya untuk mempercepat proses pengambilan tulisan yang ada dalam sebuah gambar.

3.3 Perancangan Prototype

Pada tahap ini, peneliti akan merancang desain aplikasi berbasis *Optical Character Recognition* untuk Ekstraksi Teks pada Gambar yang akan dibuat.

3.3.1 Desain Use Case

Berikut ini adalah rancangan Use Case Diagram aplikasi yang diajukan.

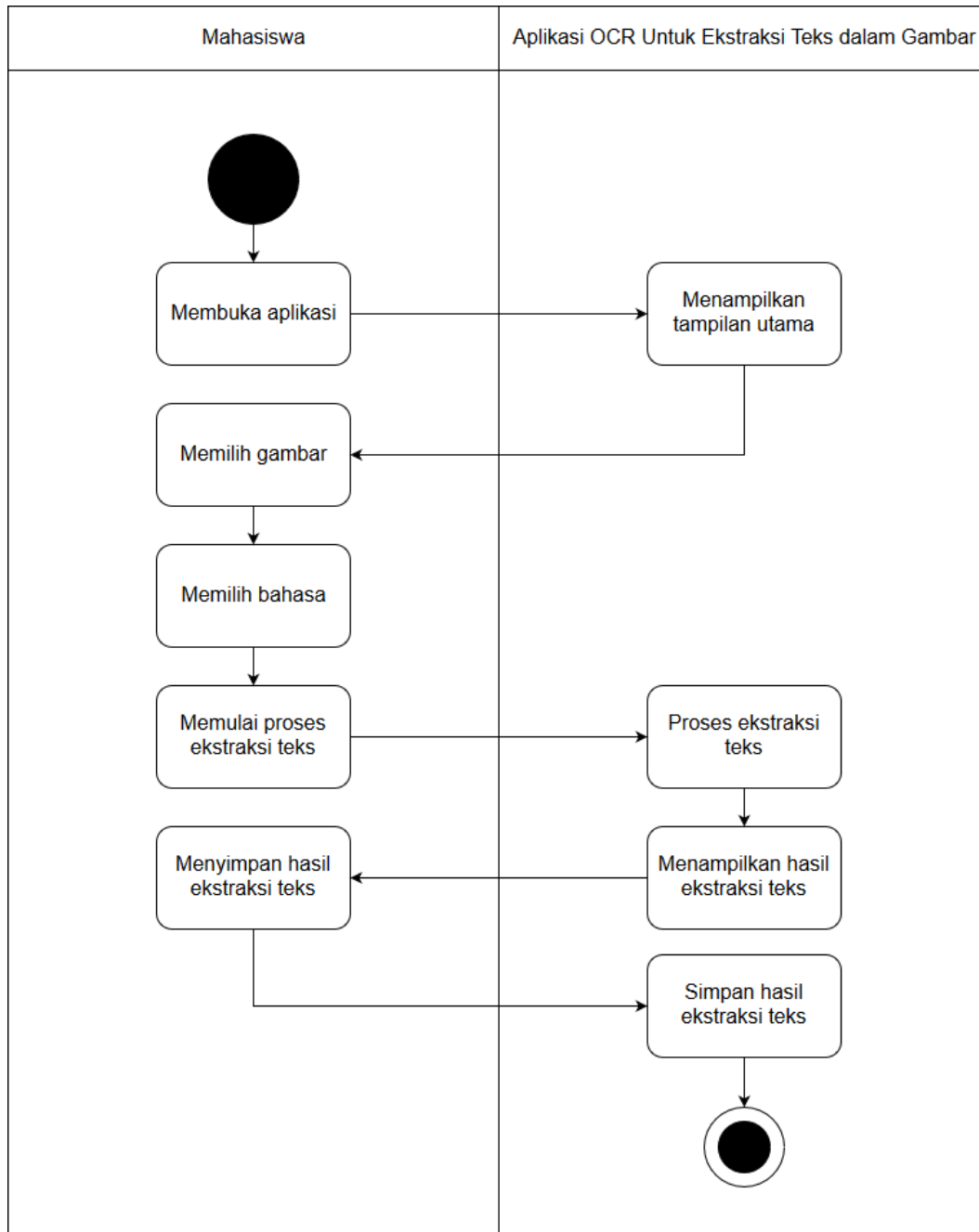


Gambar 3.1 Use Case Diagram untuk aplikasi yang diajukan

Fungsi – fungsi pada gambar 3.1 dapat dijelaskan secara singkat sebagai berikut. Mahasiswa menginputkan gambar kedalam aplikasi, setelah itu Tesseract-OCR akan mengambil teks yang ada dalam sebuah gambar. Setelah itu, tulisan dapat disimpan oleh mahasiswa.

3.3.2 Desain Activity Diagram

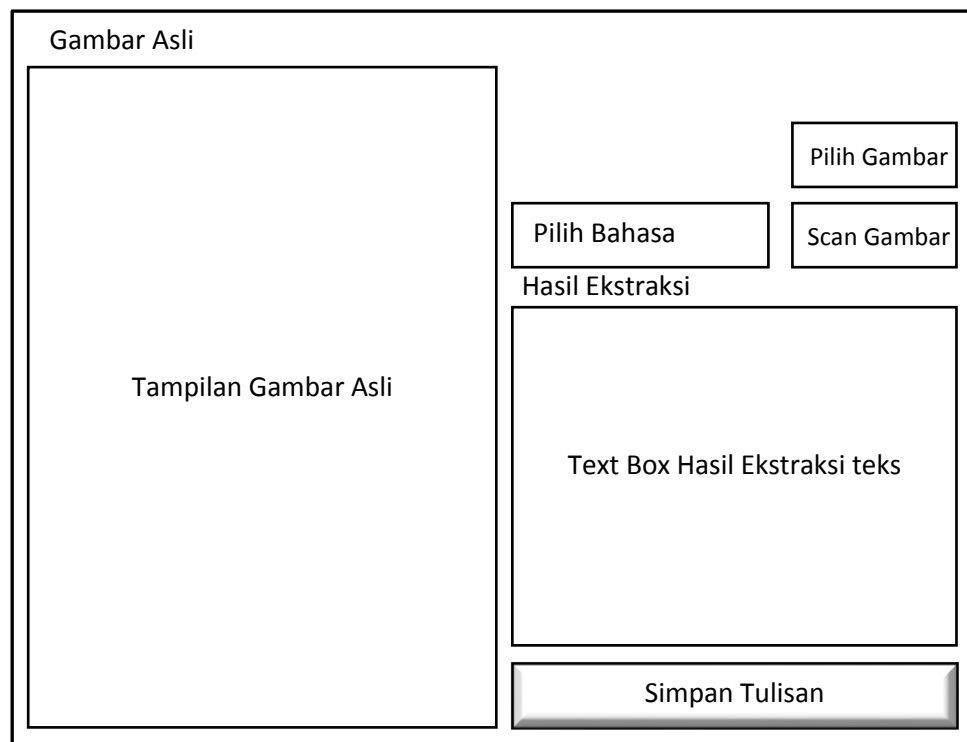
Activity Diagram berguna untuk memberikan visualisasi alur tindakan dalam sistem, percabangan yang mungkin dapat terjadi, dan bagaimana alur sistem dari awal mula hingga akhir.



Gambar 3.2 Activity Diagram Ekstraksi Teks

3.3.3 Desain Rancangan Aplikasi

Rancangan untuk tampilan halaman utama aplikasi dapat dilihat pada gambar 3.3.



Gambar 3.3 Rancangan Tampilan Utama Aplikasi

3.4 Evaluasi Prototype

Merupakan tahap dimana rancangan aplikasi yang telah dibangun di evaluasi oleh pengguna. Tahapan pengevaluasian ini dilakukan untuk memperjelas spesifikasi kebutuhan pengguna terhadap aplikasi sehingga sesuai dengan keinginan pengguna.

3.5 Proses Kerja Aplikasi

Aplikasi *Optical Character Recognition* untuk Ekstraksi Teks pada Gambar merupakan aplikasi berbasis *Windows Desktop* yang dapat mengambil teks dalam sebuah gambar. Berikut adalah proses mengambil teks :

1. Instal dan jalankan aplikasi pada perangkat berbasis *Windows*.
2. Masukkan gambar yang ingin diproses ke dalam aplikasi.
3. Pilih bahasa teks yang ingin diekstraksi.
4. Klik scan gambar, maka hasil ekstraksi teks akan tampil.
5. Klik simpan tulisan untuk menyimpan hasil ekstraksi teks.

BAB IV

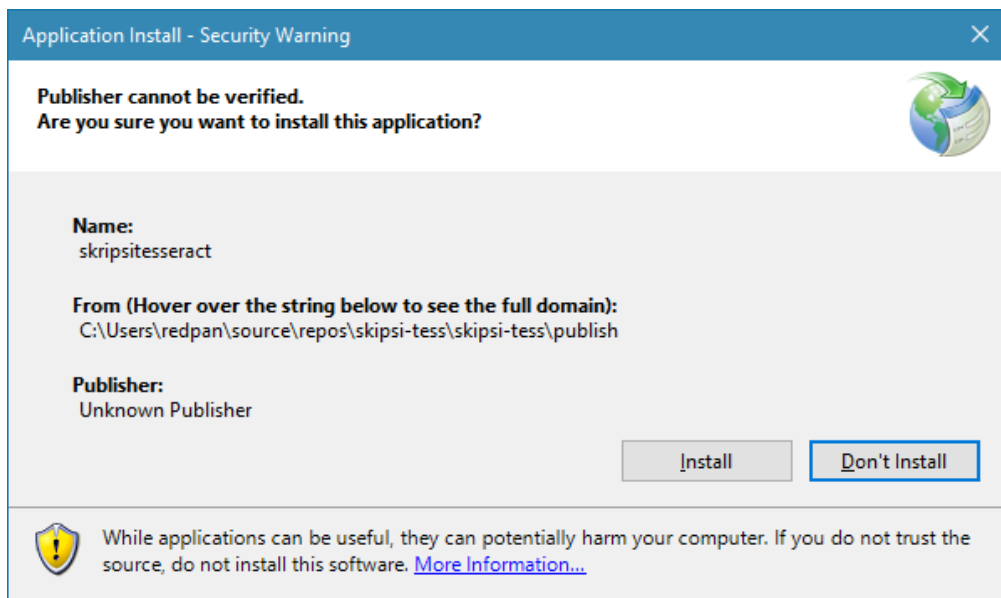
HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Setelah melalui analisa kebutuhan program dan perancangan maka tahap selanjutnya adalah pembuatan kode program. Tampilan, *output* dan cara penggunaan aplikasi ini adalah sebagai berikut.

4.1.1 Instalasi Perangkat Lunak

Aplikasi *Optical Character Recognition* untuk Ekstraksi Teks pada Gambar akan diinstal pada sistem operasi Windows 10 akan tampak seperti gambar 4.1.



Gambar 4.1 Instalasi Perangkat Lunak

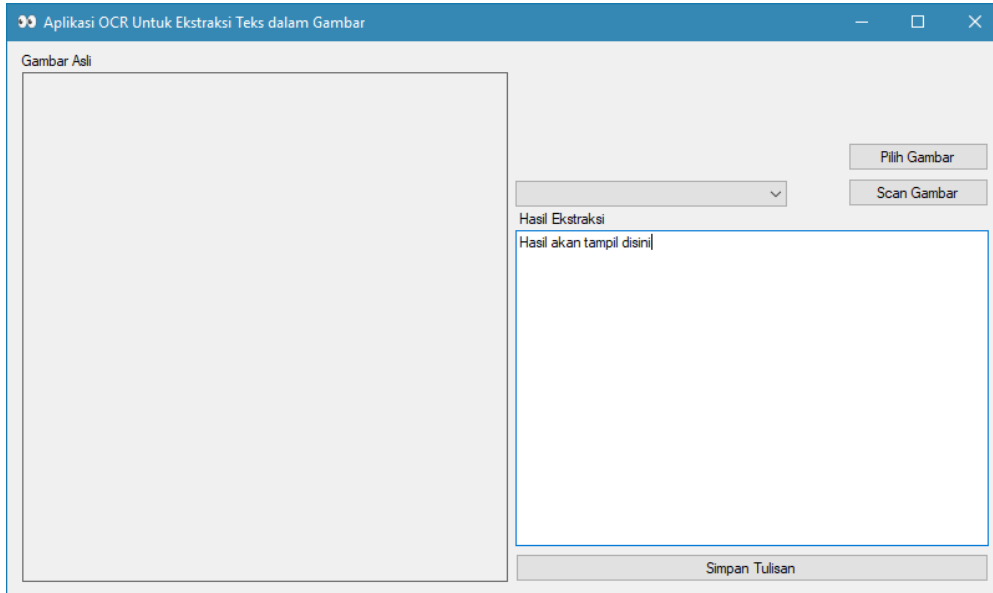
Aplikasi dapat segera digunakan setelah selesai melewati proses instalasi. Aplikasi ini memiliki tampilan *icon* sebagai berikut:



Gambar 4.2 *Icon* aplikasi setelah terinstal

4.1.2 Penggunaan Perangkat Lunak

Pada saat aplikasi pertama kali dibuka, aplikasi akan menampilkan layar utama seperti tampak pada gambar 4.3 berikut:



Gambar 4.3 Tampilan utama aplikasi

4.1.3 Pemilihan Gambar dan Bahasa

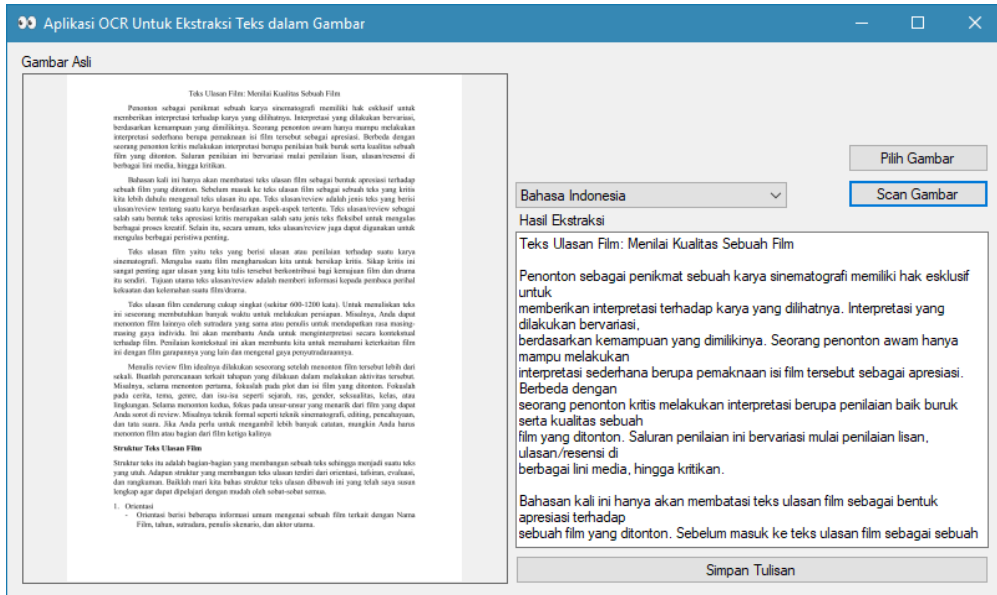
Berikut adalah tampilan aplikasi setelah gambar yang ingin di ekstrak teksnya telah dipilih, dan bahasa teks yang ada di dalam gambar telah dipilih:



Gambar 4.4 Tampilan aplikasi setelah gambar dan bahasa dipilih

4.1.4 Tampilan Hasil Scan

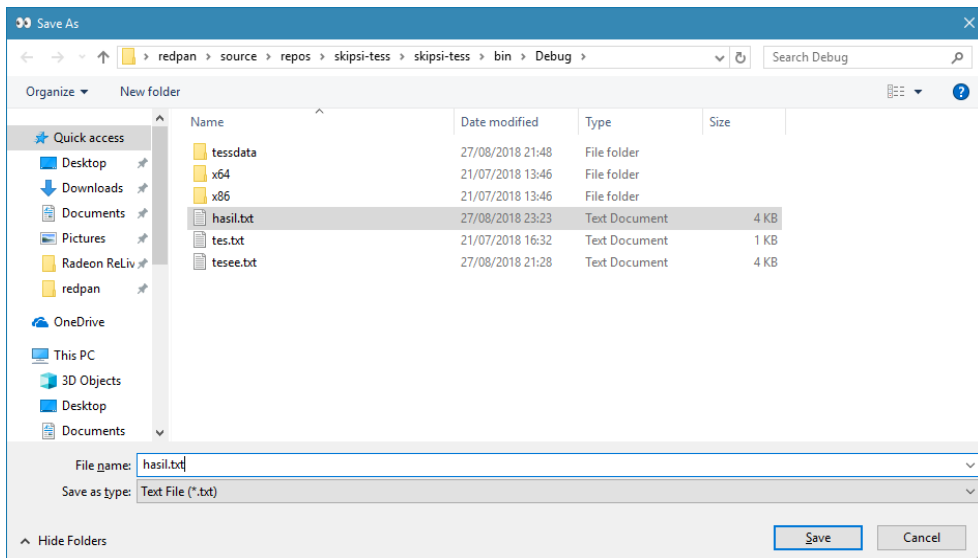
Pada saat tombol Scan Gambar ditekan, aplikasi akan melakukan ekstraksi teks pada gambar. Setelah selesai, maka hasil ekstraksi teks akan muncul pada kotak Hasil Ekstraksi seperti gambar 4.5 berikut:



Gambar 4.5 Tampilan aplikasi dengan hasil ekstraksi teks

4.1.5 Tampilan Simpan Tulisan

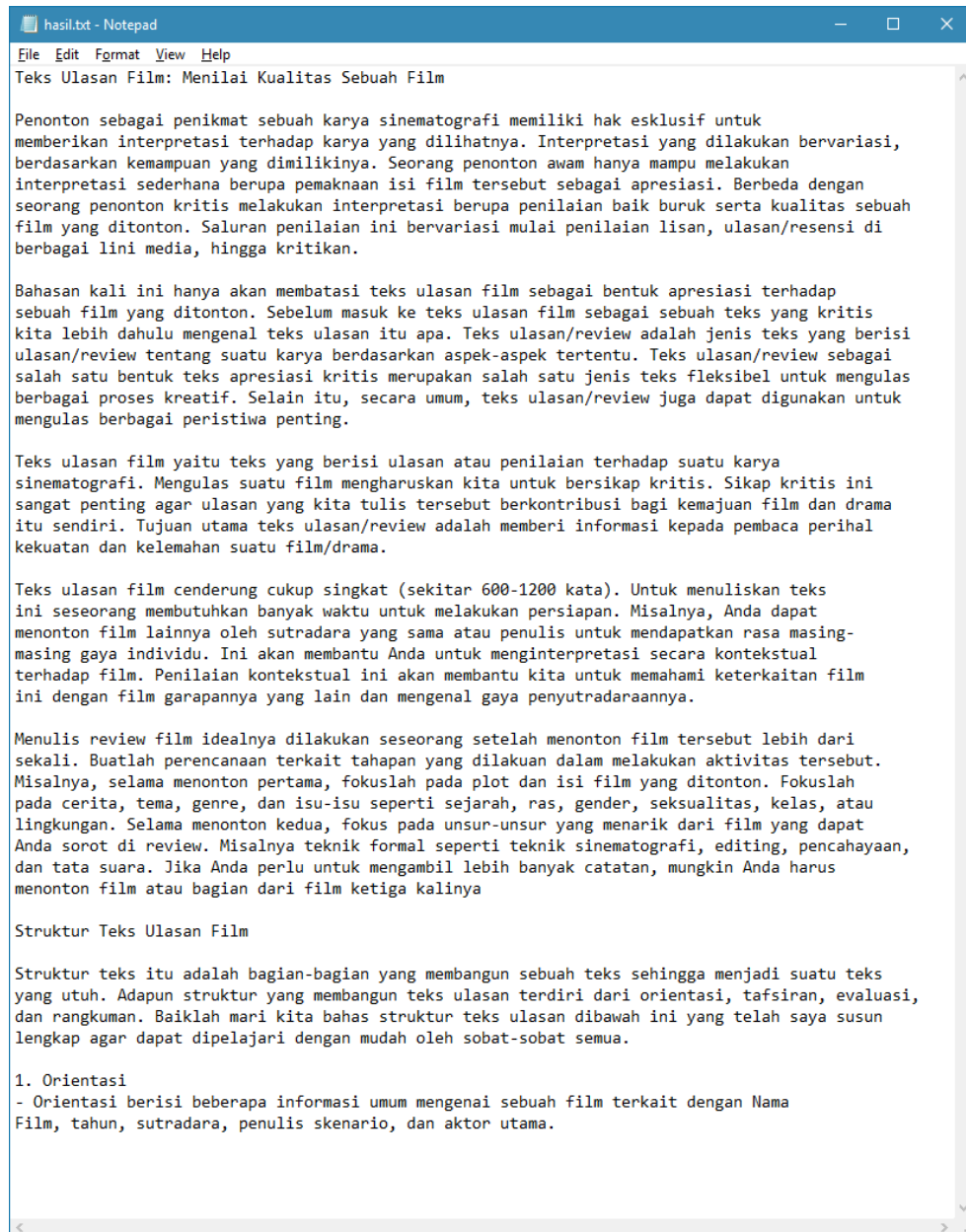
Aplikasi akan menampilkan *dialog box* untuk memilih lokasi penyimpanan *file* hasil ekstraksi dan nama *file* yang diinginkan seperti pada gambar berikut:



Gambar 4.6 Dialog simpan tulisan

4.1.6 Hasil Simpan Tulisan

Aplikasi akan menyimpan tulisan hasil ekstraksi kedalam sebuah *file* berekstensi .txt. Hasil simpan tulisan pada aplikasi *Notepad* tampak seperti gambar 4.7 berikut:



Gambar 4.7 Hasil simpan teks dalam aplikasi *notepad*.

4.2 Pembahasan

4.2.1 Pengujian Sistem

Proses pengujian sistem akan dilaksanakan dengan metode *black box testing*. Pengujian akan dilakukan pada komputer *desktop* berspesifikasi *CPU Intel Core i5-2500K, 12 GB RAM, HDD 320GB, OS Windows 10*, dan komputer *laptop* berspesifikasi *CPU AMD A4-3330MX, 4 GB RAM, HDD 120GB, OS Windows 7*. Kedua komputer dapat menjalankan *program* dengan baik, seluruh fungsi dapat digunakan. Aplikasi dapat dijalankan dalam sistem operasi minimal *Windows 7* atau lebih baru.

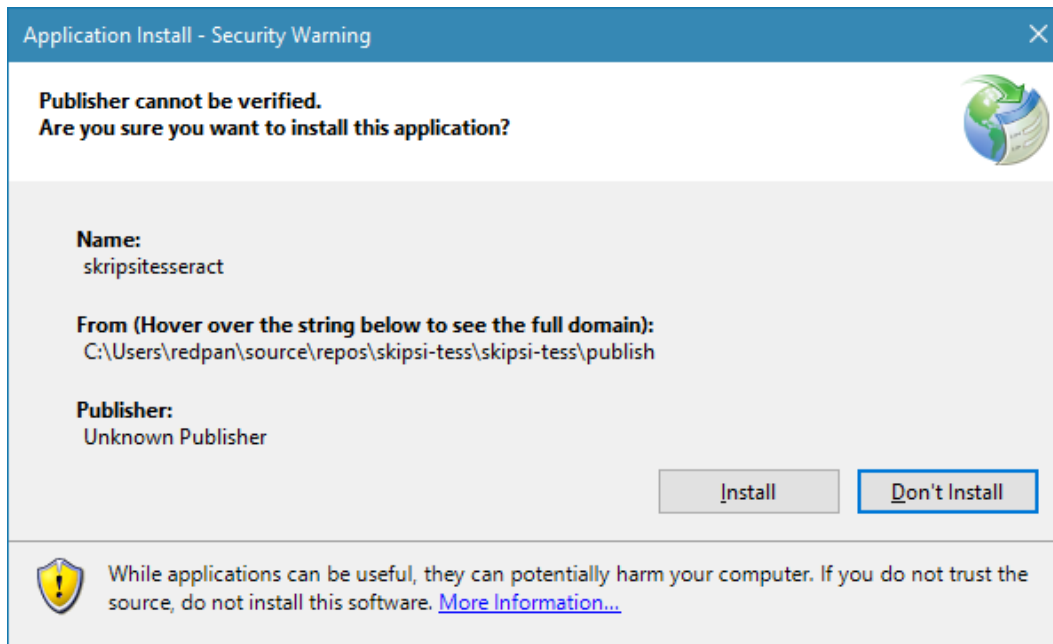
4.2.2 Pengujian Program

4.2.2.1 Pengujian Instalasi Program

Hasil uji aplikasi pada tahap instalasi dilakukan terhadap beberapa jenis komputer menggunakan sistem operasi berbasis *Windows* akan dijelaskan pada tabel 4.1 sebagai berikut:

Tabel 4.1 Pengujian Instalasi Program

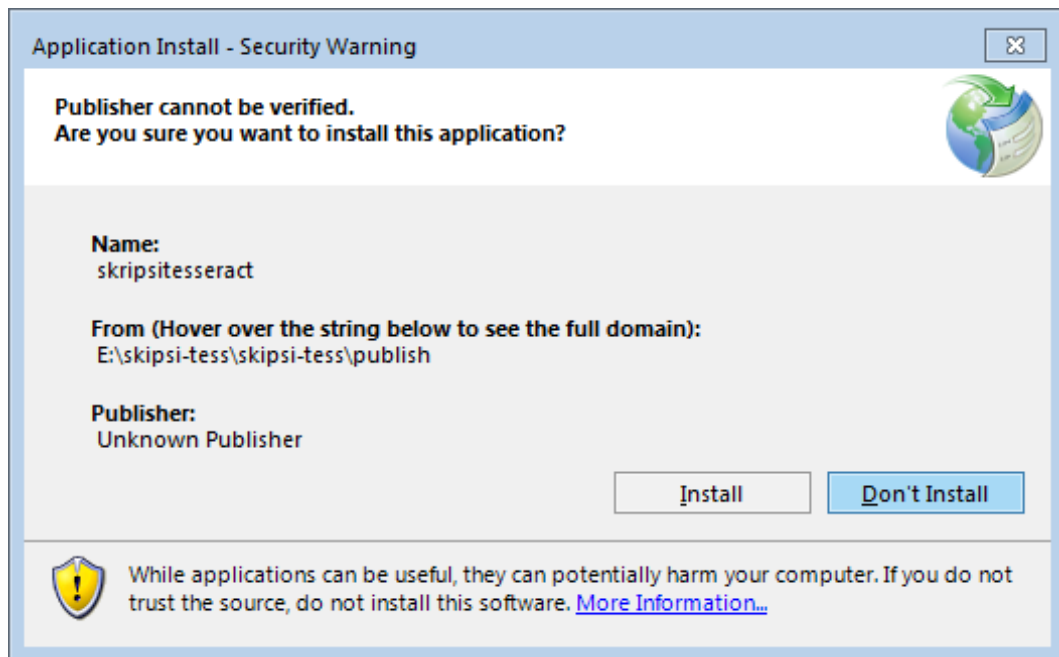
No	Item Uji	Tipe Item	Gambar	Keterangan
1	<i>Processor</i>	<i>Intel Core i5-2500K</i>	Lihat Gambar 4.8	Aplikasi dapat terinstall dengan baik.
	<i>RAM</i>	12 GB		
	<i>HDD</i>	320 GB		
	<i>Versi Windows</i>	<i>Windows 10</i>		



Gambar 4.8 Uji Instalasi

Tabel 4.1 (Lanjutan) Pengujian Instalasi Program

No	Item Uji	Tipe Item	Gambar	Keterangan
2	Processor	AMD A4-3330MX	Lihat Gambar 4.9	Aplikasi dapat terinstall dengan baik.
	RAM	4 GB		
	HDD	120 GB		
	Versi Windows	Windows 7		



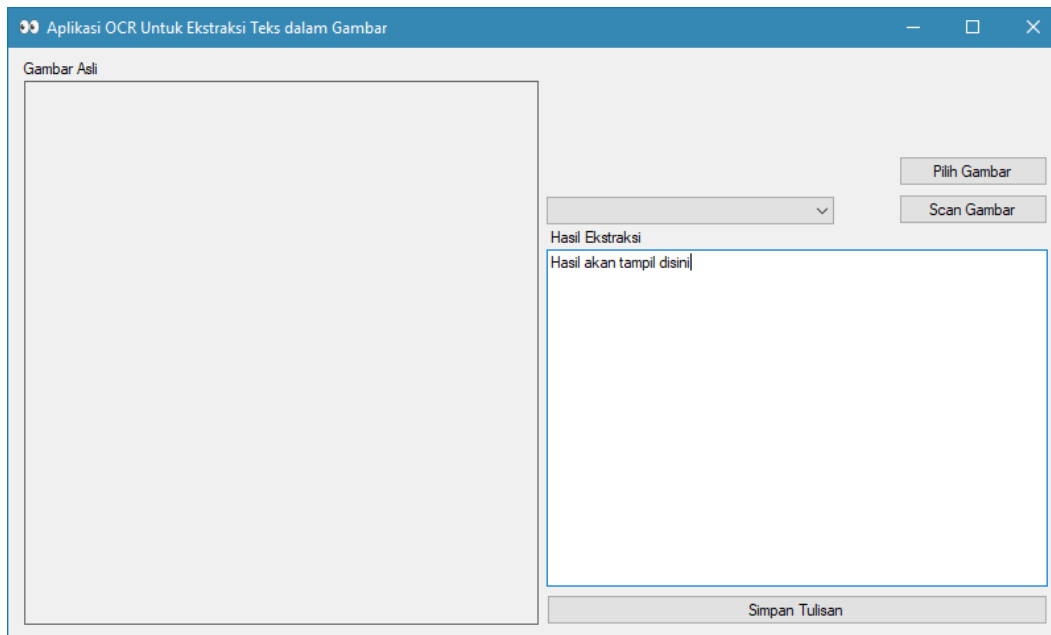
Gambar 4.9 Uji Instalasi Lanjutan

4.2.2.2 Pengujian Tampilan Utama Program

Hasil uji aplikasi pada tampilan utama dilakukan terhadap beberapa jenis komputer menggunakan sistem operasi *Windows* akan dijelaskan pada tabel 4.2 berikut:

Tabel 4.2 Pengujian Tampilan Utama Program

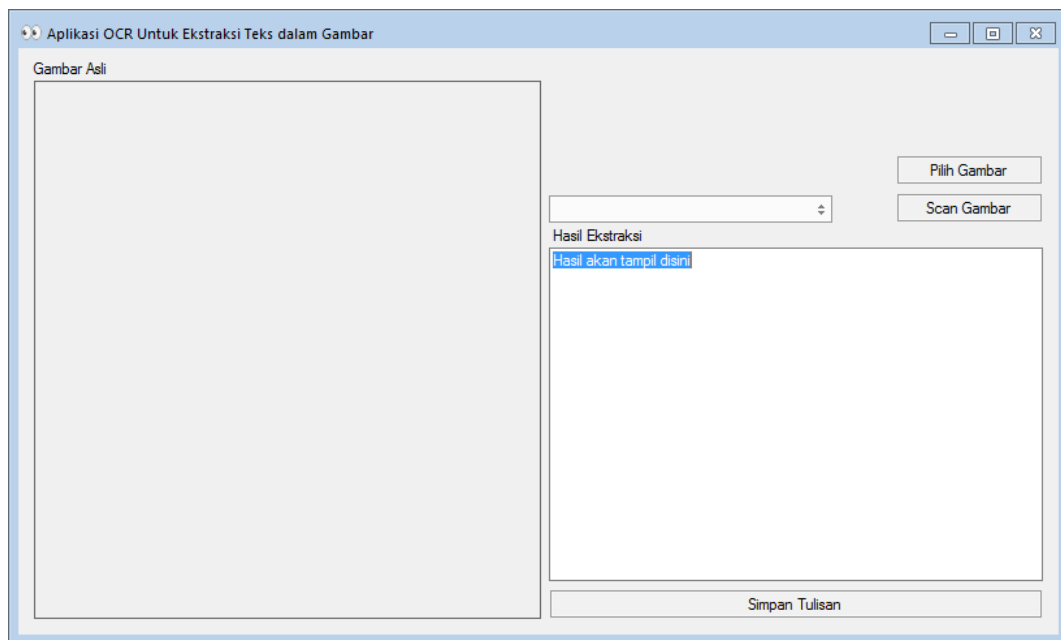
No	Item Uji	Tipe Item	Gambar	Keterangan
1	<i>Processor</i>	<i>Intel Core i5-2500K</i>	Lihat gambar 4.10	Aplikasi dapat dibuka dengan lancar dan berfungsi dengan baik.
	<i>RAM</i>	12 GB		
	<i>HDD</i>	320 GB		
	<i>Versi Windows</i>	<i>Windows 10</i>		



Gambar 4.10 Uji Tampilan Utama

Tabel 4.2 (Lanjutan) Pengujian Tampilan Utama Program

No	Item Uji	Tipe Item	Gambar	Keterangan
2	<i>Processor</i>	<i>AMD A4-3330MX</i>	Lihat gambar 4.11	Aplikasi dapat dibuka dengan lancar dan berfungsi dengan baik.
	<i>RAM</i>	4 GB		
	<i>HDD</i>	120 GB		
	<i>Versi Windows</i>	<i>Windows 7</i>		



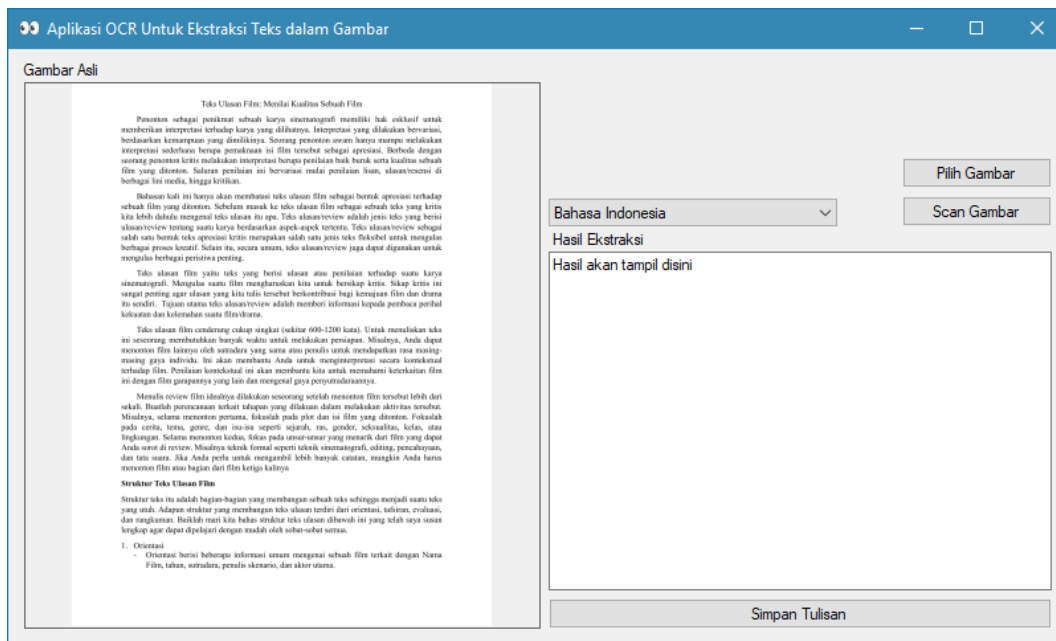
Gambar 4.11 Uji Tampilan Utama Lanjutan

4.2.2.3 Pengujian Input Gambar dan Bahasa

Hasil uji aplikasi untuk menginputkan gambar dan bahasa dilakukan terhadap beberapa jenis komputer menggunakan sistem operasi *Windows* akan dijelaskan pada tabel 4.3 berikut:

Tabel 4.3 Pengujian Input Gambar dan Bahasa

No	Item Uji	Type Item	Gambar	Keterangan
1	Processor	Intel Core i5-2500K	Lihat gambar 4.12	Aplikasi dapat mengakses dan menampilkan gambar dan bahasa dengan benar.
	RAM	12 GB		
	HDD	320 GB		
	Versi Windows	Windows 10		



Gambar 4.12 Uji Input Gambar dan Bahasa

Tabel 4.3 (Lanjutan) Pengujian Input Gambar dan Bahasa

No	Item Uji	Tipe Item	Gambar	Keterangan
2	<i>Processor</i>	<i>AMD A4-3330MX</i>	Lihat gambar 4.13	Aplikasi dapat mengakses dan menampilkan gambar dengan baik dan benar.
	<i>RAM</i>	<i>4 GB</i>		
	<i>HDD</i>	<i>120 GB</i>		
	<i>Versi Windows</i>	<i>Windows 7</i>		



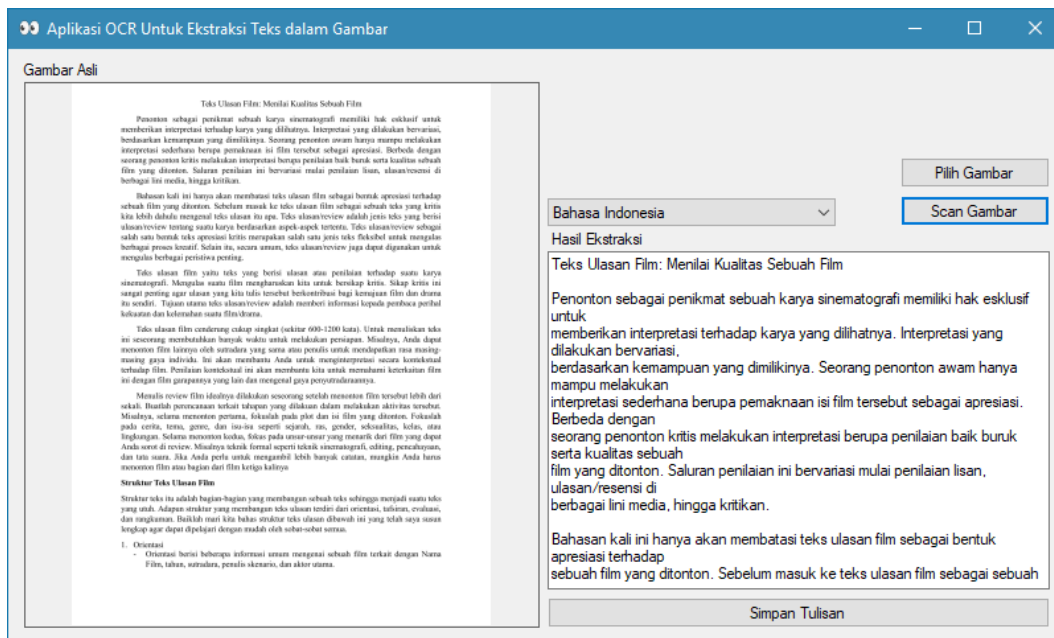
Gambar 4.13 Uji Input Gambar dan Bahasa Lanjutan

4.2.2.4 Pengujian Ekstraksi Teks dari Gambar

Hasil uji aplikasi untuk mengekstraksi teks dari gambar dilakukan terhadap beberapa jenis komputer menggunakan sistem operasi *Windows* akan dijelaskan pada tabel 4.4 berikut:

Tabel 4.4 Pengujian Ekstraksi Teks dari Gambar

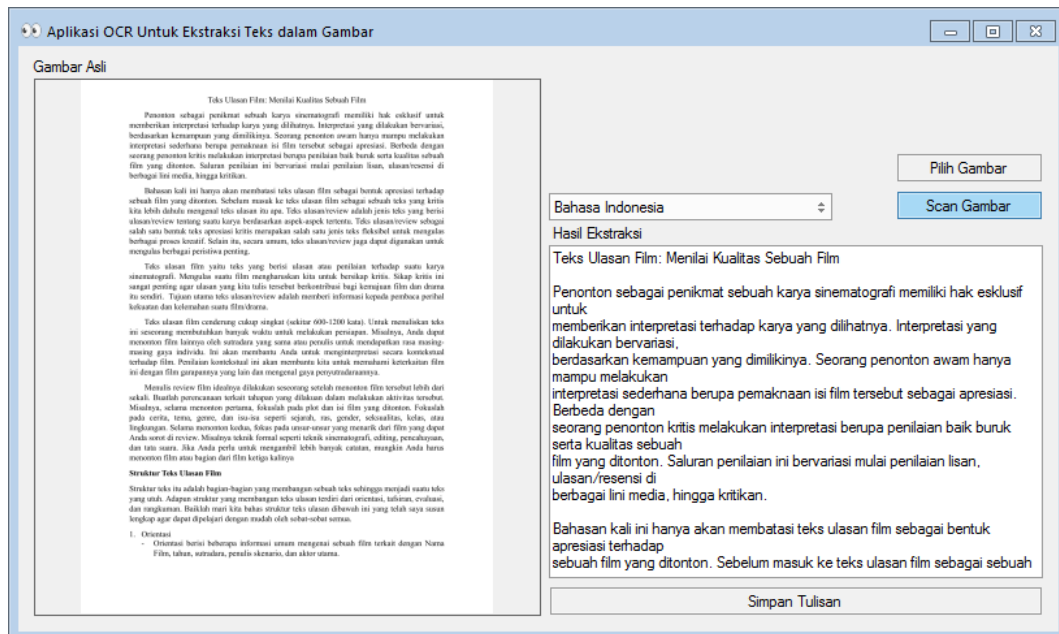
No	Item Uji	Tipe Item	Gambar	Keterangan
1	<i>Processor</i>	<i>Intel Core i5-2500K</i>	Lihat gambar 4.14	Aplikasi dapat melakukan ekstraksi teks dengan baik.
	<i>RAM</i>	12 GB		
	<i>HDD</i>	320 GB		
	<i>Versi Windows</i>	<i>Windows 10</i>		



Gambar 4.14 Uji Ekstraksi Teks

Tabel 4.4 (Lanjutan) Pengujian Ekstraksi Teks dari Gambar

No	Item Uji	Tipe Item	Gambar	Keterangan
2	Processor	AMD A4-3330MX	Lihat Gambar 4.15	Aplikasi dapat melakukan ekstraksi teks dengan baik.
	RAM	4 GB		
	HDD	120 GB		
	Versi Windows	Windows 7		



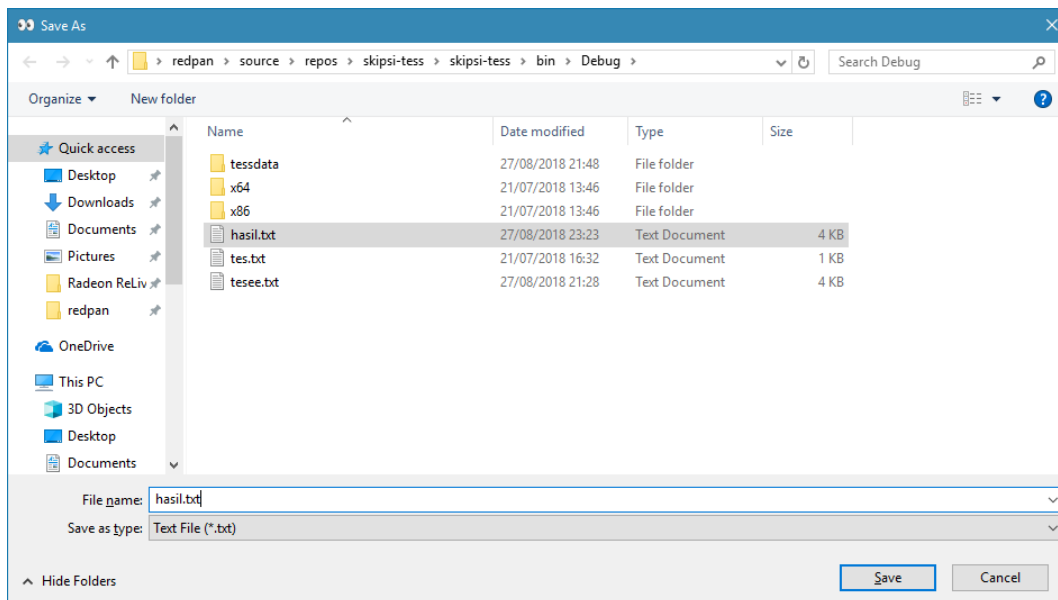
Gambar 4.15 Uji Ekstraksi Teks Lanjutan

4.2.2.5 Pengujian Penyimpanan Hasil Ekstraksi Teks

Hasil uji aplikasi untuk menyimpan hasil ekstraksi teks ke dalam sebuah *file* dilakukan terhadap beberapa jenis komputer menggunakan sistem operasi *Windows* akan dijelaskan pada tabel 4.5 berikut:

Tabel 4.5 Pengujian Penyimpanan Hasil Ekstraksi Teks

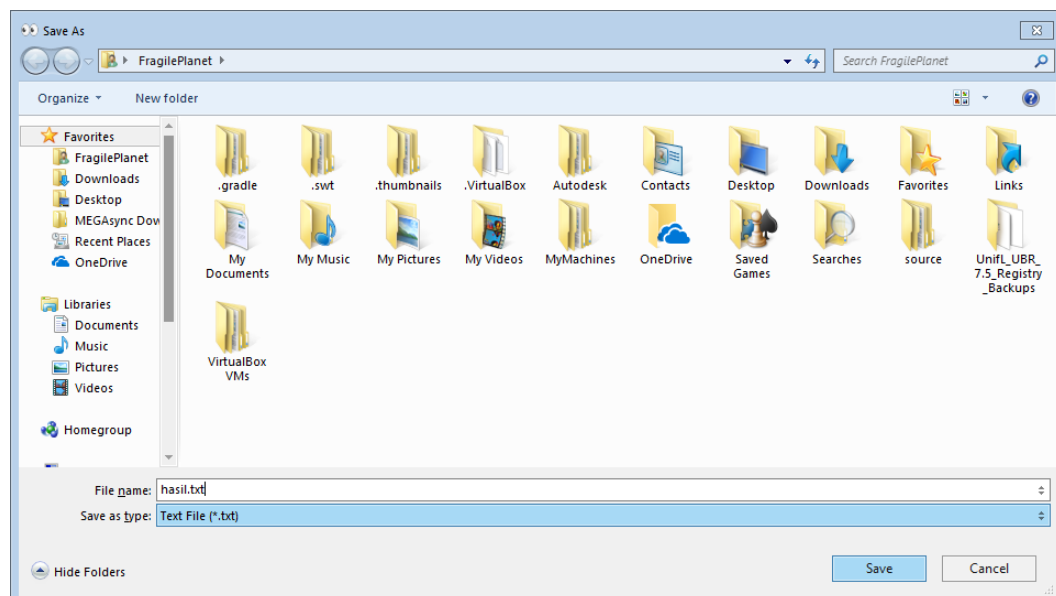
No	Item Uji	Type Item	Gambar	Keterangan
1	Processor	Intel Core i5-2500K	Lihat Gambar 4.16	Aplikasi dapat menyimpan hasil ekstraksi teks dengan baik dan benar.
	RAM	12 GB		
	HDD	320 GB		
	Versi Windows	Windows 10		



Gambar 4.16 Uji Penyimpanan Hasil Ekstraksi Teks

Tabel 4.5 (Lanjutan) Pengujian Penyimpanan Hasil Ekstraksi Teks

No	Item Uji	Tipe Item	Gambar	Keterangan
2	<i>Processor</i>	<i>AMD A4-3330MX</i>	Lihat gambar 4.17	Aplikasi dapat menyimpan hasil ekstraksi teks dengan baik dan benar.
	<i>RAM</i>	4 GB		
	<i>HDD</i>	120 GB		
	<i>Versi Windows</i>	<i>Windows 7</i>		



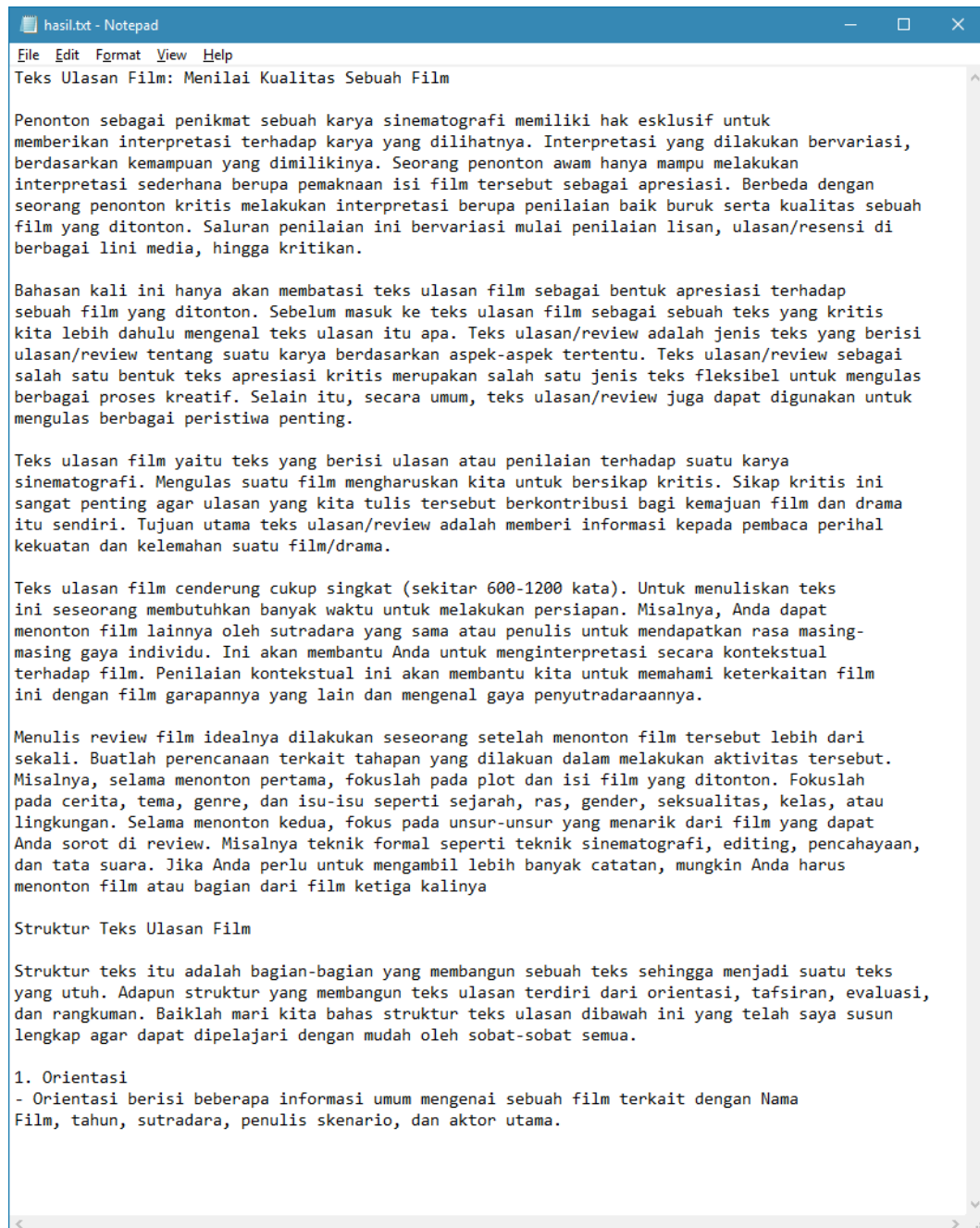
Gambar 4.17 Uji Penyimpanan Hasil Ekstraksi Teks Lanjutan

4.2.2.6 Pengujian Hasil Penyimpanan Ekstraksi Teks

Hasil uji aplikasi untuk hasil penyimpanan ekstraksi teks dilakukan terhadap beberapa jenis komputer menggunakan sistem operasi *Windows* akan dijelaskan pada tabel 4.6 berikut:

Tabel 4.6 Pengujian Hasil Penyimpanan Ekstraksi Teks

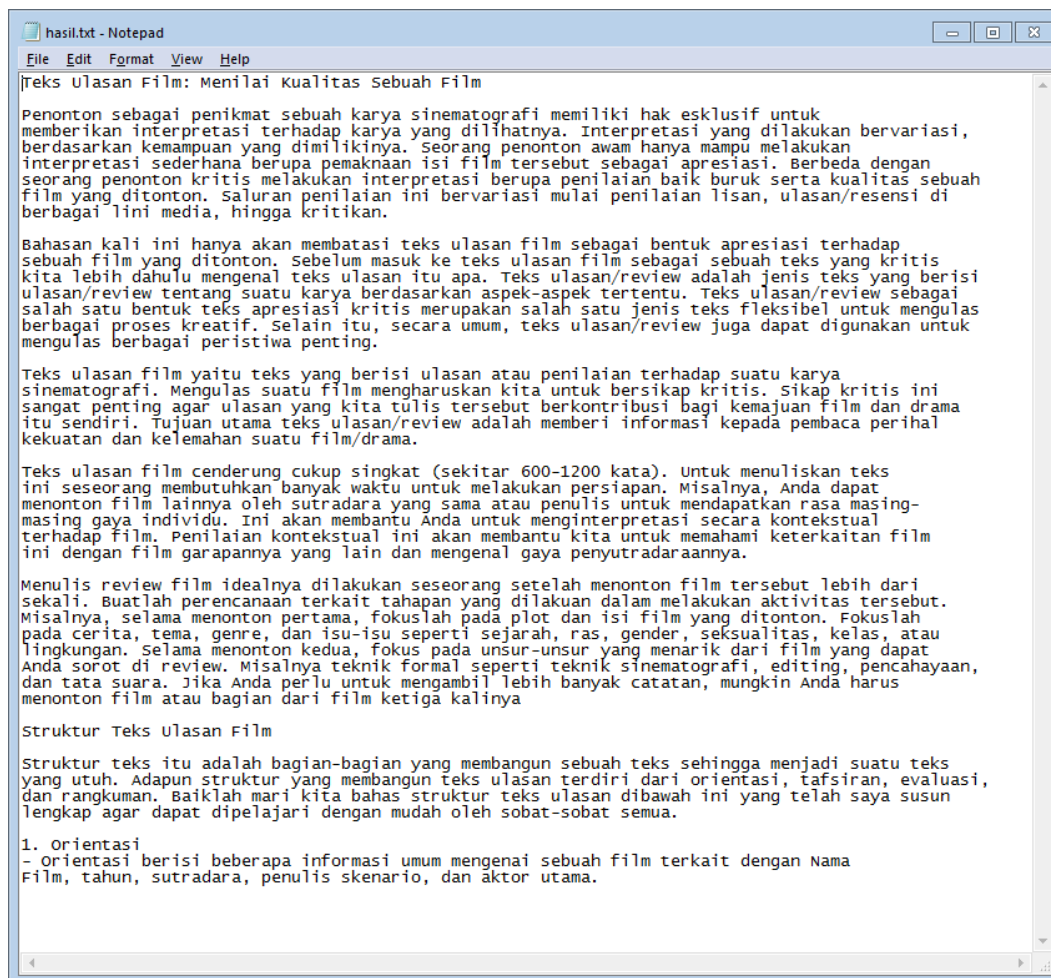
No	Item Uji	Type Item	Gambar	Keterangan
1	<i>Processor</i>	<i>Intel Core i5-2500K</i>	Lihat gambar 4.18	Hasil ekstraksi teks dari aplikasi dapat ditampilkan dan digunakan dengan baik.
	<i>RAM</i>	12 GB		
	<i>HDD</i>	320 GB		
	<i>Versi Windows</i>	<i>Windows 10</i>		



Gambar 4.18 Uji Hasil Penyimpanan Ekstraksi Teks

Tabel 4.6 (Lanjutan) Pengujian Hasil Penyimpanan Ekstraksi Teks

No	Item Uji	Tipe Item	Gambar	Keterangan
2	Processor	AMD A4-3330MX	Lihat gambar 4.19	Hasil ekstraksi teks dari aplikasi dapat ditampilkan dan digunakan dengan baik.
	RAM	4 GB		
	HDD	120 GB		
	Versi Windows	Windows 7		



Gambar 4.19 Uji Hasil Penyimpanan Ekstraksi Teks Lanjutan

4.3 Kelebihan dan Kelemahan Aplikasi

Di dalam aplikasi *Optical Character Recognition* untuk ekstraksi teks pada gambar terdapat beberapa kelebihan dan kekurangan sebagai berikut:

4.3.1 Kelebihan Aplikasi

Terdapat beberapa kelebihan dari aplikasi *Optical Character Recognition* untuk ekstraksi teks pada gambar, yaitu:

- a. Aplikasi sudah menggunakan Tesseract-OCR sebagai engine ekstraksi teks berakurasi dan berkecepatan tinggi.
- b. Kecepatan aplikasi untuk mengekstraksi teks jauh lebih cepat daripada mengetikkan kembali tulisan yang ada dalam gambar.
- c. Aplikasi mendukung beragam bahasa, yaitu Bahasa Indonesia, Inggris, Jepang, Korea, dan Mandarin.

4.3.2 Kekurangan Aplikasi

Terdapat beberapa kekurangan dari aplikasi *Optical Character Recognition* untuk ekstraksi teks pada gambar yang perlu dikembangkan kedepannya, yaitu:

- a. Tampilan aplikasi masih sangat sederhana.
- b. Aplikasi tidak dapat mengenali gambar yang ada dalam sebuah gambar, sehingga tidak dapat mengambil gambar yang ada.
- c. Akurasi pendeteksian bergantung dengan ukuran gambar yang disediakan oleh pengguna.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Dengan adanya aplikasi *Optical Character Recognition untuk ekstraksi teks pada gambar* memudahkan para mahasiswa IIB Darmajaya untuk mengambil tulisan yang ada dalam sebuah gambar.
2. Aplikasi *Optical Character Recognition untuk ekstraksi teks pada gambar* dapat membantu mahasiswa IIB Darmajaya untuk mempercepat proses ekstraksi teks dalam gambar.

5.2 Saran

Aplikasi yang dibangun masih memiliki beberapa kelemahan, untuk itu masih diperlukan adanya perbaikan maupun pengembangan untuk penelitian selanjutnya.

Saran yang diperlukan untuk pengembangan aplikasi ini lebih lanjut adalah:

1. Meningkatkan tingkat akurasi dan pendeteksian teks dalam aplikasi sehingga tulisan yang diambil sesuai dengan tulisan apa yang ada dalam sebuah gambar.
2. Agar aplikasi dapat mengenali gambar yang ada dalam sebuah gambar sehingga gambar tersebut dapat diambil secara otomatis juga.
3. Menambahkan fitur lainnya ke dalam aplikasi ini, misalnya yaitu pengaturan ukuran tulisan saat ditampilkan, dukungan bahasa yang lebih luas, dan dapat memperbesar sendiri gambar yang digunakan sehingga dapat meningkatkan akurasi pendeteksian teks.

DAFTAR PUSTAKA

- Jogiyanto, HM. 2005. Analisis Dan Desain Sistem Informasi : Pendekatan Terstruktur Teori Dan Praktik Aplikasi Bisnis. Yogyakarta: Andi.
- Mustakini, Jogiyanto Hartono, 2009. Sistem Informasi Teknologi. Yogyakarta: Andi Offset.
- Putra, Darma. 2010. Pengolahan Citra Digital. Yogyakarta: Andi.
- Shalahuddin, M., & Rosa, A.S. 2016. Rekayasa Perangkat Lunak. Penerbit Informatika, Bandung.
- Smith R. 2007. *An Overview of the Tesseract OCR Engine. ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition II; 2007 Sept 23-26; Curitiba, Brasil.* Washington DC (US): IEEE Computer Society.

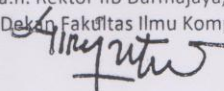
LAMPIRAN



SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK.0180/DMJ/DFIK/BAAK/III-18
Tentang
Dosen Pembimbing Skripsi
Program Studi S1 Teknik Informatika
REKTOR IIB DARMAJAYA

- Memperhatikan :** 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam Skripsi.
2. Laporan dan usulan Ketua Program Studi S1 Teknik Informatika.
- Menimbang :** 1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan Dosen Pembimbing Skripsi.
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.
- Mengingat :** 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
4. STATUTA IBI Darmajaya
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.
- Menetapkan**
- Pertama :** Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Teknik Informatika.
- Kedua :** Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.
- Ketiga :** Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.
- Keempat :** Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

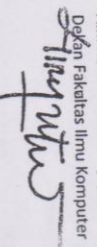
Ditetapkan di : Bandar Lampung
Pada tanggal : 12 Maret 2018
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer


Sriyanto, S.Kom., M.M.
NIK. 00210800

1. Ketua Jurusan S1 Teknik Informatika
2. Yang bersangkutan
3. Arsip

Lampiran : Surat Keputusan Rektor IIB Darmajaya
Nomor : SK. 0180/DMM/DFIK/BAAK/III-18
Tanggal : 12 Maret 2018
Perihal : Pembimbing Penulisan Skripsi
Program Studi Strata Satu (S1) Teknik Informatika

No	NAMA	NPM	JUDUL	PEMBIMBING
1	Ananda Dharmayana Rani	1411010073	JUDUL SKRIPSI DAN DOSEN PEMBIMBING PROGRAM STUDI STRATA SATU (S1) TEKNIK INFORMATIKA Mengaplikasikan Metode Optical Character Recognition Untuk Ekstraksi Teks Pada Gambar	Dr. Suhendro Yusuf I

A.n. Rektor IIB Darmajaya
Dekan Fakultas Ilmu Komputer

Sriyanto, S.Kom., M.M. 18
NIK. 00210800



Institut Informatika & Bisnis

DARMAJAYA

Yayasan Alfian Husin

Jl. Zainal Abidin Pager Alam No. 93 Bandar Lampung 35142 Telp 787214 Fax. 700261 http://darmajaya.ac.id

FORMULIR

BIRO ADMINISTRASI AKADEMIK KEMAHASISWAAN (BAAK)

FORM KONSULTASI/BIMBINGAN SKRIPSI/TUGAS AKHIR *)

NAMA : Ananda Dharmayana Rani
 NPM : 1411010073
 PEMBIMBING I : DR. IR. Suhendyo Yusuf Lianto, M.kom.
 PEMBIMBING II :
 JUDUL LAPORAN : Mengaplikasikan Metode Optical Character Recognition untuk
 Restorasi Teks pada Gambar.
 TANGGAL SK : s.d (6+2 bulan)

No	HARI/TANGGAL	HASIL KONSULTASI	PARAF
1	3/8 2018	Latar belakang	
2	8/8 2018	Tujuan, Rumusan masalah, Identifikasi masalah	
3	14/8 2018	Tinjauan Pustaka	
4	20/8 2018	Metodologi Penelitian	
5	24/8 2018	Rancangan Gangguan Program, f UJI Program	
6	29/8 2018	Penyusunan Hasil UJI Program	
7	30/8 2018	Acc Sidang	
8			
9			
10			

*) Coret yang tidak perlu

Bandar Lampung, 30 Agustus 2018
Ketua Jurusan

Yuni Auliansyah, S.kom, M.kom
NIK. 00480802

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Tesseract;

namespace skipsi_tess
{
    public partial class Form1 : Form
    {
        String bahasa;
        String[] bahasa3h = { "eng", "ind", "kor", "jpn", "chi_sim", "chi_tra" };
        int pbahasa;
        Bitmap gambar;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void BtScan_click(object sender, EventArgs e)
        {
            try {
                var ocr = new TesseractEngine("./tessdata", bahasa,
EngineMode.TesseractOnly);

                var halaman = ocr.Process(gambar);
                string hasil= halaman.GetText();
                textBoxHasil.Text = hasil.Replace("\n", Environment.NewLine);
            }
            catch (System.ArgumentException) {
                MessageBox.Show("Mohon periksa bahasa dan gambar!",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void CboxBahasa_SelectedIndexChanged(object sender, EventArgs e)
        {
            pbahasa = CboxBahasa.SelectedIndex;
            bahasa = Convert.ToString(bahasa3h[pbahasa]);
        }

        public void BtPGambar_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFile = new OpenFileDialog();
            if (openFile.ShowDialog() == DialogResult.OK)
            {
                if (openFile.FileName.EndsWith(".pdf"))

```

```

        {
        }
        else
        {
            gambar = new Bitmap(openFile.FileName);
            pictureBox1.Image = gambar;
        }
    }
}

private void BtSimpan_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFile = new SaveFileDialog();
    saveFile.Filter = "Text File (*.txt)|*.txt";
    saveFile.DefaultExt = ".txt";
    saveFile.AddExtension = true;
    DialogResult result = saveFile.ShowDialog();
    if (result == DialogResult.OK)
    {
        string name = saveFile.FileName;
        File.WriteAllText(name, textBoxHasil.Text);
    }
}

private void label1_Click(object sender, EventArgs e)
{
}
}
}

```