BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Penggunaan citra atau *image* sering dijadikan sebagai media untuk menyampaikan informasi. Banyak informasi disampaikan melalui teks pada gambar contohnya brosur, papan iklan, petunjuk jalan atau ketika ingin menyimpan informasi dari papan pengumuman secara instan, kita dapat meng-*capture* teks pada papan pengumuman tersebut menjadi citra dan menyimpannya.

Kekurangan dari citra atau *image* yakni, teks pada *image* tidak dapat diedit atau ditulis ulang tanpa mengetiknya secara manual satu persatu. Saat ini telah ada aplikasi yang menyediakan fasilitas konversi tulisan pada *image* ke dalam bentuk teks biasa, akan tetapi belum dirasa optimal baik dari segi kemudahan penggunaan ataupun hasil konversinya. Jika citra atau *image* yang ditangkap kurang jelas, buram, terlalu gelap, atau kurang maksimal maka dapat mengurangi keakuratan konversi citra ke teks tersebut, sehingga hasil konversi menjadi kurang optimal. Selain itu, teks pada citra atau *image* bisa berupa tulisan latin (alfabet) ataupun non-latin (karakter khusus seperti karakter kanji, mandarin, dan sebagainya), tetapi aplikasi yang ada saat ini belum bisa sepenuhnya mengenali berbagai karakter khusus tersebut sehingga dibutuhkan suatu *library* tambahan sebagai acuan bahasa untuk membedakan pengenalan karakternya.

Salah satu teknologi yang dapat dimanfaatkan untuk permasalahan ini adalah pengimplementasian *Leptonica* sebagai *image processing* dan penggunaan *Tesseract Library* serta *Optical Character Recognition (OCR)* yang diterapkan pada perangkat mobile berbasis Android. *Leptonica* merupakan *library* yang digunakan sebagai pengolah citra [4] dengan merubah atau mengolah citra ke dalam bentuk *binary image* ataupun mengatur tingkat kontras, kecerahan dan *perspective* citra sebelum dikonversi sehingga pengenalan karakter ketika citra dikonversi ke teks dapat lebih akurat. *Optical Character Recognition (OCR)* merupakan metode untuk mengubah teks pada *image* (citra) menjadi teks yang memungkinkan pengenalan karakter melalui mekanisme optik [1]. *Tesseract* merupakan salah satu *library* yang dapat digunakan pada OCR sebagai *library*

bahasa atau *language* [2] agar OCR dapat mengenali karakter dalam bahasa latin ataupun non-latin.

Android menjadi pilihan dalam penelitian ini karena saat ini sistem operasi Android merupakan salah satu sistem operasi yang saat ini tengah berkembang di masyarakat. Menurut hasil riset teknologi dan pasar dunia yakni IDC (idc.com) pada Agustus 2015 lalu [5], Android masih memimpin pangsa pasar OS dengan total 82,8 persen sedangkan iOS berada di urutan kedua dengan total 13,9 persen pada periode 2015 Q2.

Tabel 1.1 Hasil Riset IDC Pangsa Pasar OS 2015

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

Berdasarkan permasalahan diatas maka penulis mengangkat judul **IMPLEMENTASI IMAGE PROCESSING DENGAN LEPTONICA PADA PENGEMBANGAN APLIKASI IMAGE TO TEXT DENGAN OPTICAL CHARACTER RECOGNITION (OCR) DAN TESSERACT LIBRARY BERBASIS ANDROID.** Dengan adanya sistem yang dikembangkan pada penelitian ini diharapkan mampu memberikan kemudahan dan mengoptimalkan hasil dari konversi pada citra atau *image* ke dalam bentuk teks yang lebih baik.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat dirumuskan masalah yaitu:

1. Jika citra atau *image* yang ditangkap kurang jelas, buram, terlalu gelap, atau kurang maksimal maka akan dapat mengurangi keakuratan konversi citra ke teks tersebut.
2. Teks pada citra atau *image* bisa berupa tulisan latin (alfabet) ataupun non-latin (karakter khusus seperti karakter kanji, mandarin, dan sebagainya),

tetapi aplikasi yang ada saat ini belum bisa sepenuhnya mengenali berbagai karakter khusus tersebut.

1.3 Maksud dan Tujuan

Berdasarkan permasalahan yang ada, maksud dari penelitian ini adalah untuk mengimplementasikan *Leptonica Library* sebagai pengolah citra dan *Tesseract Library* sebagai *library* bahasa serta *Optical Character Recognition* pada pengembangan aplikasi *Image to Text* berbasis Android untuk mengoptimalkan hasil konversi.

Adapun tujuan dari penelitian ini, yaitu :

1. Mampu mengoptimalkan hasil dari konversi teks pada citra atau *image* ke dalam bentuk teks yang lebih baik dan memberikan kemudahan dengan tampilan yang mudah digunakan.
2. Dapat mengenali karakter bahasa berupa tulisan latin (alfabet) ataupun non-latin (karakter khusus seperti karakter kanji, mandarin, dan sebagainya) sesuai dengan bahasa yang dipilih.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Objek input berupa citra atau *image.*
2. OCR dilakukan hanya untuk mengkonversi ke dalam bentuk teks.
3. Di desain untuk mengenali *printed text*, karena keterbatasan dari Tesseract yang saat ini di desain hanya untuk *printed text* (belum mendukung *handwriting* dan karakter dengan banyak *style).* Dikutip dari pernyataan developer pada situs https://github.com/tesseract-ocr/tesseract/wiki/FAQ
4. Keluaran berupa teks hasil konversi yang dapat diedit dan disimpan dalam bentuk txt.
5. Sasaran pengguna yakni para pelajar atau umum yang membutuhkan kemudahan mengkonversi dari citra atau *image* ke dalam bentuk teks yang dapat diedit.
6. Aplikasi ini dapat digunakan secara offline dan hanya membutuhkan jaringan online untuk men-*download library language* tambahan.

7. Aplikasi menggunakan platform Android
8. Perancangan menggunakan bahasa pemrograman Java untuk Android
9. *Tools* pembangun aplikasi ini menggunakan Android Studio, Android SDK dan NDK

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian tugas akhir ini adalah metode analisis deskriptif yaitu suatu metode yang ditujukan untuk mendapatkan gambaran yang lebih jelas mengenai hal-hal yang diperlukan dalam melakukan penelitian melalui beberapa tahap, yakni sebagai berikut:

1.5.1 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan adalah sebagai berikut :

a. Studi Literatur

Pengumpulan data dengan melakukan studi atau mempelajari hal-hal yang berkaitan dengan judul penelitian melalui berbagai macam literatur yang dibutuhkan baik melalui buku, karya ilmiah, jurnal dan juga data yang bersumber online yang berkaitan dengan aplikasi yang akan dibangun.

b. Pengamatan (Observasi)

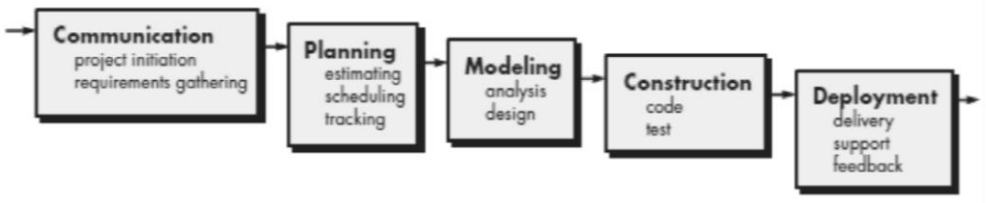
Pengamatan (Observasi) dilakukan pada aplikasi sejenis yang sudah ada untuk mengamati kelebihan dan kekurangan yang ada pada aplikasi tersebut.

1.5.2 Metode Pembangunan Perangkat Lunak

Metode pembangunan perangkat lunak menggunakan metode *waterfall*, Metode waterfall adalah metode yang digunakan dalam pendekatan yang sistematis dan *sequential* melalui tahapan-tahapan yang ada untuk membangun sebuah aplikasi.

Metode waterfall menekankan pada sebuah keterurutan dalam proses pengembangan sebuah aplikasi. Dalam metode waterfall, satu fase harus selesai terlebih dahulu sebelum fase berikutnya dimulai. Suatu fase dapat dinyatakan selesai apabila *output* dari fase tersebut sesuai dengan kebutuhan yang didefinisikan pada fase sebelumnya.

Menurut Roger S. Pressman (2010,p39), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Berikut adalah metode *waterfall* menurut Pressman yang ditunjukkan pada Gambar 1.2 :



Gambar 1.1 Model Waterfall

Berikut penjelasan mengenai fase-fase yang terdapat pada metode waterfall:

1. *Comunication*

Langkah ini merupakan analisis terhadap kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data dengan mengumpulkan data – data tambahan baik yang ada di jurnal, artikel maupun dari internet.

2. *Planning*

Proses *planning* merupakan lanjutan dari proses *communication (analiysis requirement).* Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang akan dilakukan

3. *Modelling*

Proses *modelling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum membuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software,* representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement.*

4. *Construction*

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh user.

5. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang

sudah jadi digunakan oleh user. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum mengenai penelitian yang dijalankan. Sistematika penulisan pada skripsi ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang permasalahan, merumuskan masalah yang dihadapi, menentukan maksud dan tujuan penelitian, pembuatan batasan masalah, metode yang digunakan dalam penelitian dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini membahas berbagai teori dan konsep dasar yang berkaitan dengan topik penelitian yang dilakukan untuk mendukung analisis dan perancangan sistem yang akan dibangun.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang analisis sistem meliputi, menganalisis aplikasi sejenis yang sudah ada, gambaran umum permasalahan yang dihadapi, usulan pemecahan masalah tersebut serta kebutuhan dan rancangan sistem yang dibangun mulai dari analisa kebutuhan non-fungsional dan analisa kebutuhan fungsional.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas implementasi dari tahapan analisis dan perancangan sistem ke dalam perangkat lunak (dalam bentuk bahasa pemrograman), beberapa implementasi yang akan dijelaskan adalah implementasi perangkat keras, implementasi perangkat lunak, dan implementasi antarmuka.

BAB 5 KESIMPULAN DAN SARAN

Bab ini terdiri dari kesimpulan dan saran yang berisikan hal-hal terpenting yang dibahas dan kemudian dijadikan kesimpulan. Bab ini juga berisi saran-saran yang dimungkinkan untuk pengembangan perangkat lunak selanjutnya