



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Berikut ini adalah metode penelitian yang digunakan pada penelitian Implementasi Skew Detection Berbasis Vertical Projection Dalam Mengenali Karakter Latin Pada Gambar Dokumen Teks.

1. Studi Literatur

Pada studi literatur, pembelajaran terkait teori-teori yang membantu dalam implementasi *Skew Detection* berbasis *Vertical Projection* dilakukan. Teori-teori antara lain adalah Optical Character Recognition yang mencakup definisi, preprocessing, dan teknik preprocessing, Metode *Skew Detection* berbasis *Vertical Projection*, cara mengukur akurasi dari sebuah metode *Skew Detection*, cara menghitung waktu komputasi sebuah metode, dan juga *library* Tesseract.

2. Perancangan Sistem

Pada tahap ini aplikasi dirancang. Rancangan aplikasi dibuat dalam bentuk DFD, *flowchart*, dan rancangan antarmuka aplikasi. *Flowchart* terdiri dari *flowchart* aplikasi, metode *Combined Vertical Projection*, modul DegreeToRad, dan modul RotateImage Aplikasi dirancang sesuai dengan kebutuhan penelitian.

3. Implementasi Metode

Pada tahap ini aplikasi akan dibangun menggunakan bahasa pemrograman C# pada *platform* Windows Form. Aplikasi dibangun sesuai dengan rancangan yang dibuat. Kemudian metode *Combined Vertical Projection* akan diimplementasikan pada aplikasi. Metode *Combined Vertical Projection* dipilih karena metode tersebut

tidak rentan terhadap *noise* dan cocok digunakan pada dokumen teks dengan karakter latin (Papandreou dan Gatos, 2011). Penelitian Khataneh dkk. (2015) juga menyatakan bahwa metode *Skew Detection* yang berbasis *Projection Profile Analysis* memberikan akurasi yang lebih baik dibandingkan dengan metode *Skew Detection* lain yang sering digunakan pada penelitian, yaitu *Hough Transform* dan *Nearest Neighbour*.

4. Evaluasi

Evaluasi akan dilakukan pada metode yang diimplementasikan, evaluasi mencakup perhitungan akurasi OCR sebagaimana dijelaskan pada telaah literatur, dan waktu komputasi yang dibutuhkan dari metode yang diimplementasi. Data tes akan diperoleh melalui dokumen teks cetak yang didigitalisasi menggunakan *scanner*.

5. Penulisan Naskah Penelitian

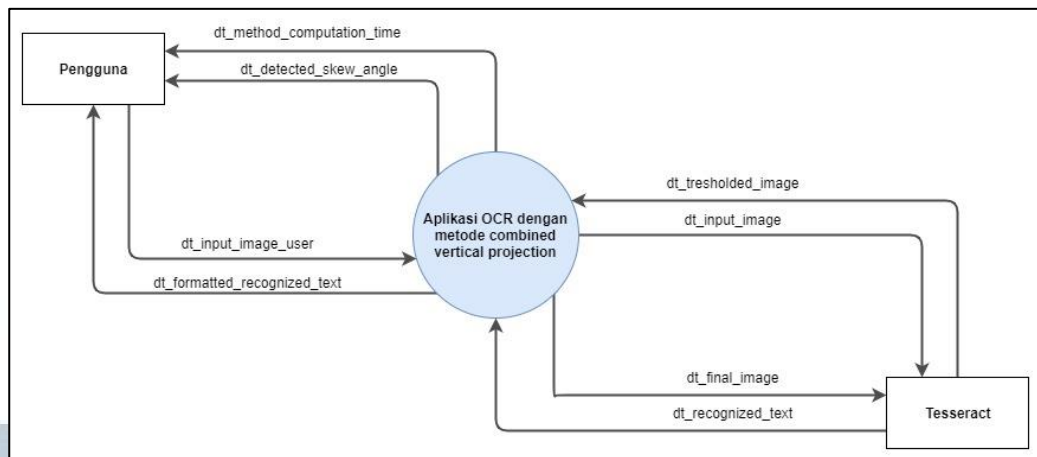
Berdasarkan penelitian yang dilakukan, penulisan naskah penelitian akan dilakukan disertai dengan konsultasi penulisan dengan dosen pembimbing.

3.2 Perancangan Aplikasi

Perancangan aplikasi akan dibantu dengan pembuatan diagram-diagram pendukung yang terdiri dari *Data Flow Diagram* (DFD), *flowchart*, dan rancangan antarmuka.

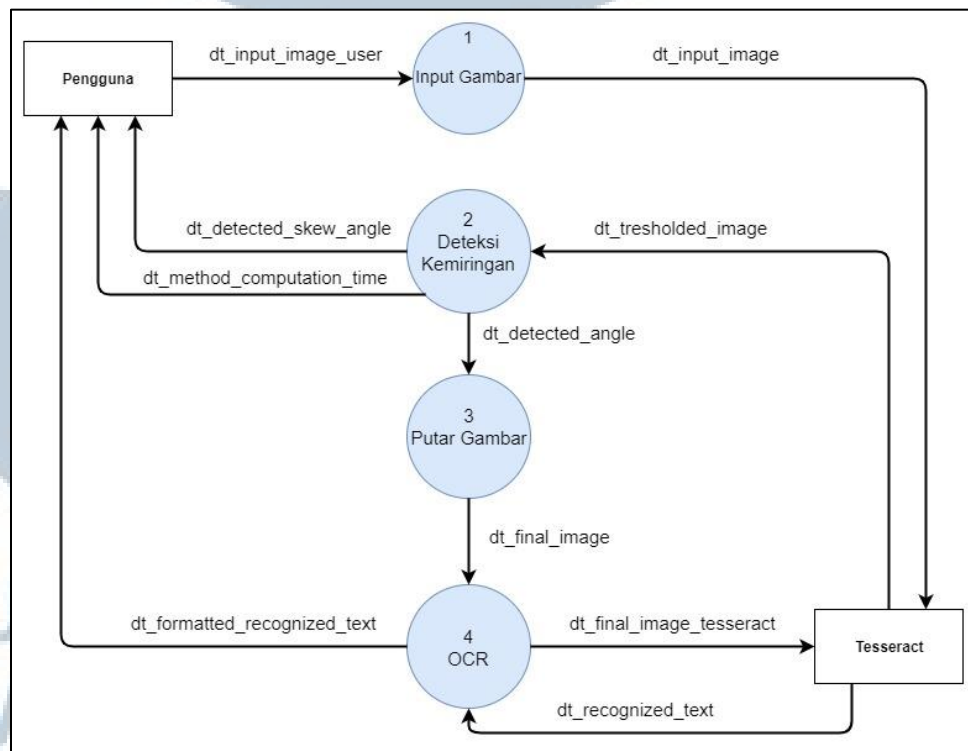
3.2.1 Data Flow Diagram

DFD dibuat dengan tujuan menggambarkan bagaimana alur data yang masuk dan keluar pada aplikasi dengan metode *Vertical Projection*.



Gambar 3.1 Context Diagram Aplikasi

Gambar 3.1 merupakan gambar *context diagram* aplikasi yang akan dibuat. Pada level ini hanya terdapat satu buah proses yaitu aplikasi dengan metode *Combined Vertical Projection*. Entitas yang berhubungan dengan aplikasi adalah pengguna aplikasi dan *library* Tesseract. Proses pada level ini dapat uraikan lebih jauh menjadi DFD level 1 yang digambarkan pada Gambar 3.2.



Gambar 3.2 DFD Level 1 Aplikasi

Gambar 3.2 merupakan gambar DFD level 1 aplikasi. Terdapat proses Input Gambar, Deteksi Kemiringan, Putar Gambar, dan OCR. Proses Input Gambar menerima *dt_input_image_user* yang berupa gambar yang akan digunakan pada Tesseract. Tesseract kemudian menghasilkan gambar yang sudah melalui tahap *preprocessing binarization* (tidak termasuk *Skew Detection*) yaitu *dt_tresholded_image*. Data *dt_tresholded_image* kemudian digunakan proses Deteksi Kemiringan untuk mendeteksi kemiringan gambar dokumen tersebut. Derajat kemiringan yang terdeteksi dan waktu komputasi metode *Combined Vertical Projection* akan ditampilkan kepada pengguna. Derajat kemiringan juga dikirimkan ke proses Putar Gambar untuk menghasilkan *dt_final_image* yang akan digunakan pada proses OCR. Tesseract menerima gambar akhir yang sudah melalui tahap *Skew Detection & correction* untuk melakukan pengenalan karakter. Tesseract akan mengirimkan teks yang berhasil dikenali, yaitu *dt_recognized_text*, ke proses OCR. Data teks yang berhasil dikenali akan ditampilkan kepada pengguna.

3.2.2 Flowchart

Flowchart ditujukan untuk menggambarkan alur aplikasi dan proses deteksi kemiringan dengan metode *Vertical Projection*.

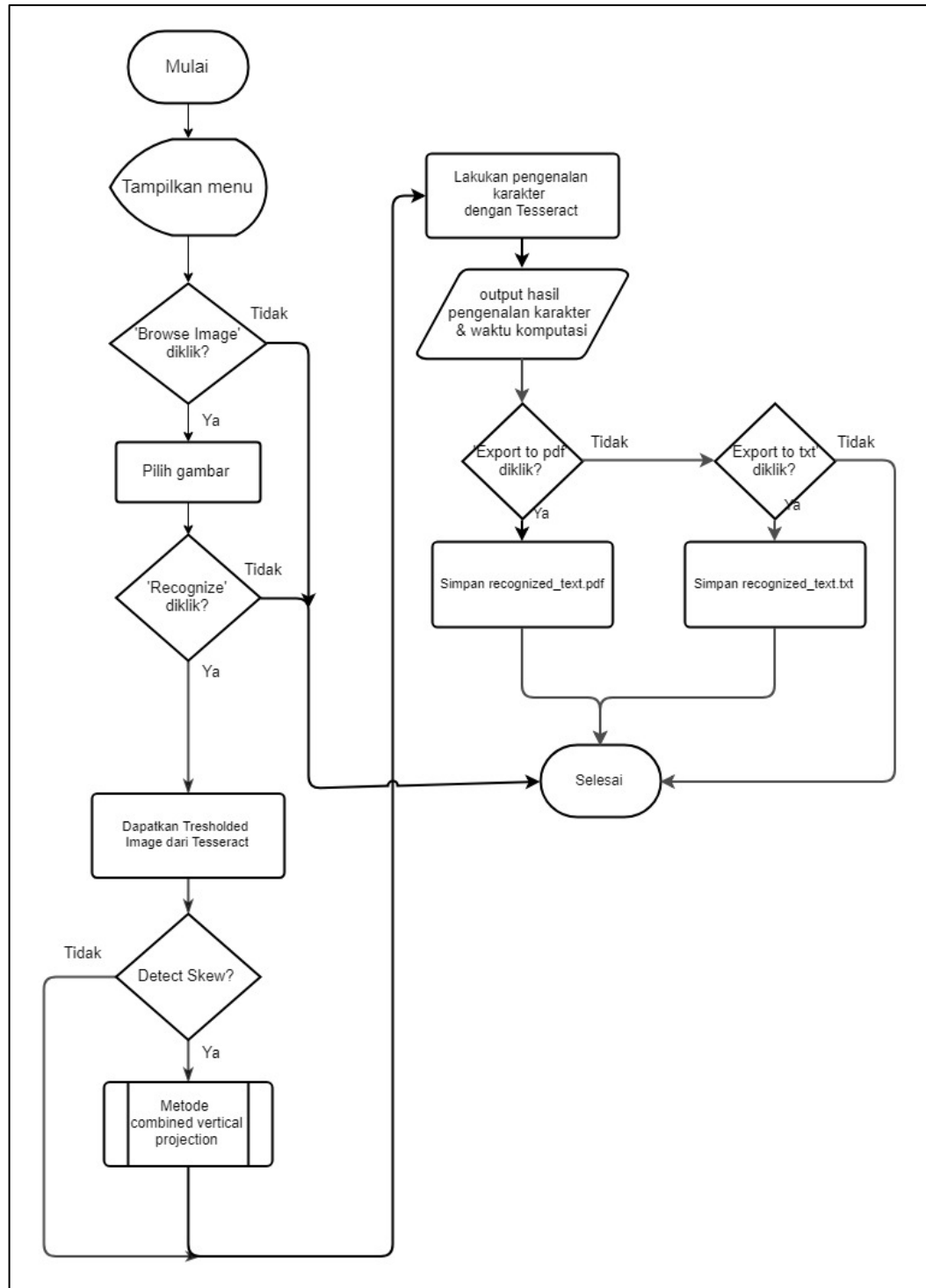
A. Flowchart Aplikasi

Gambar 3.3 menunjukkan *flowchart* aplikasi pengenalan karakter pada gambar dokumen teks. Aplikasi dimulai dari menampilkan menu yang dilanjutkan dengan pemilihan gambar yang akan digunakan oleh pengguna. Gambar yang dipilih oleh pengguna harus berupa *file* gambar dengan ekstensi .jpg, .jpeg, .png, ataupun .tif.

Aplikasi akan memperoleh gambar yang sudah melalui tahapan *preprocessing* (*thresholed image*) dari Tesseract.

Metode *Combined Vertical Projection* akan melakukan proses *Skew Detection* untuk mendeteksi derajat kemiringan dan menghasilkan gambar akhir, yaitu gambar yang sudah melalui tahap *Skew Detection & correction*. Sebelum gambar akhir digunakan pada proses pengenalan karakter Tesseract, waktu komputasi proses *Skew Detection* dan sudut yang terdeteksi akan ditampilkan untuk kebutuhan mengukur waktu komputasi dan akurasi metode *Combined Vertical Projection*. Aplikasi akan menampilkan teks hasil pengenalan karakter yang diperoleh dari Tesseract. Pengguna dapat meng-ekspor teks ke dalam dokumen dengan format .pdf dan .txt. Proses *Skew Detection* dengan *Combined Vertical Projection* akan dijelaskan lebih lanjut dengan *flowchart* pada Gambar 3.4.

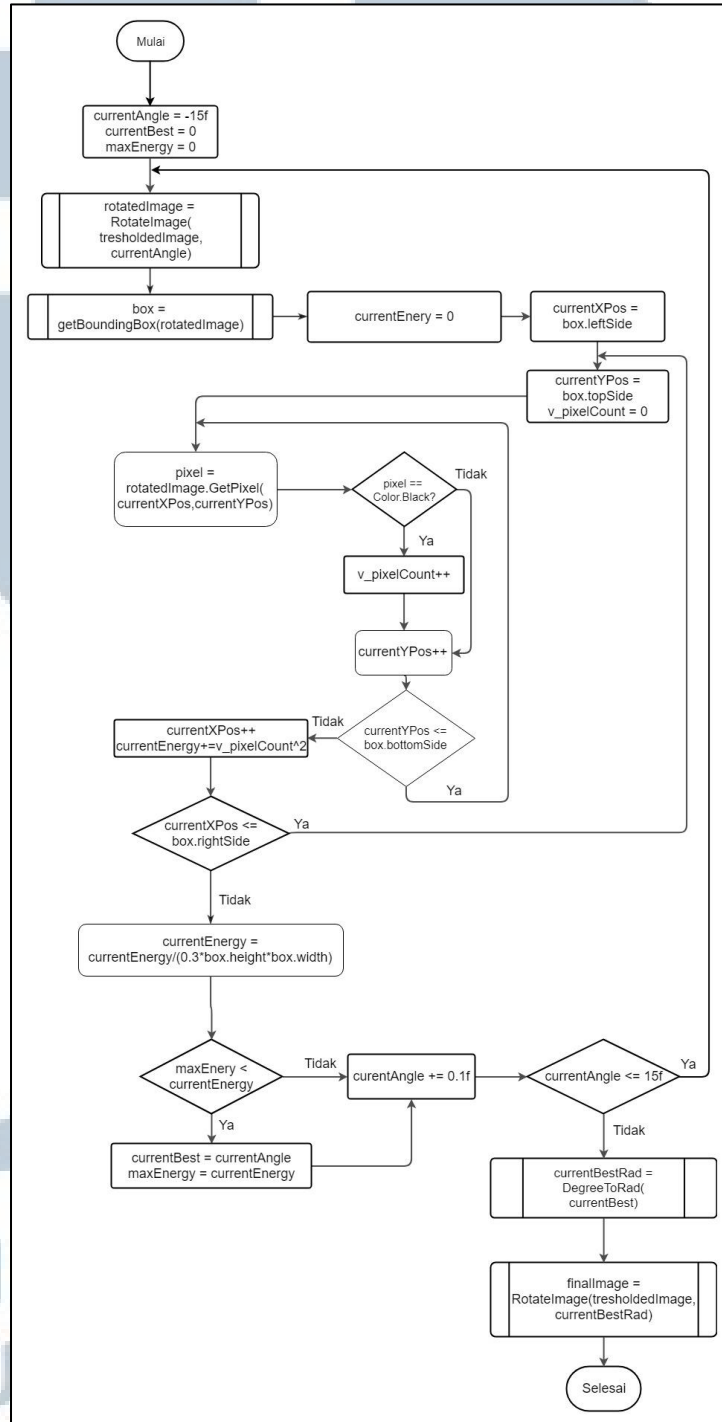




Gambar 3.3 *Flowchart* Aplikasi

B. Flowchart Metode Combined Vertical Projection

Gambar 3.4 menunjukkan *flowchart* pada proses *Skew Detection* dengan *Combined Vertical Projection*.

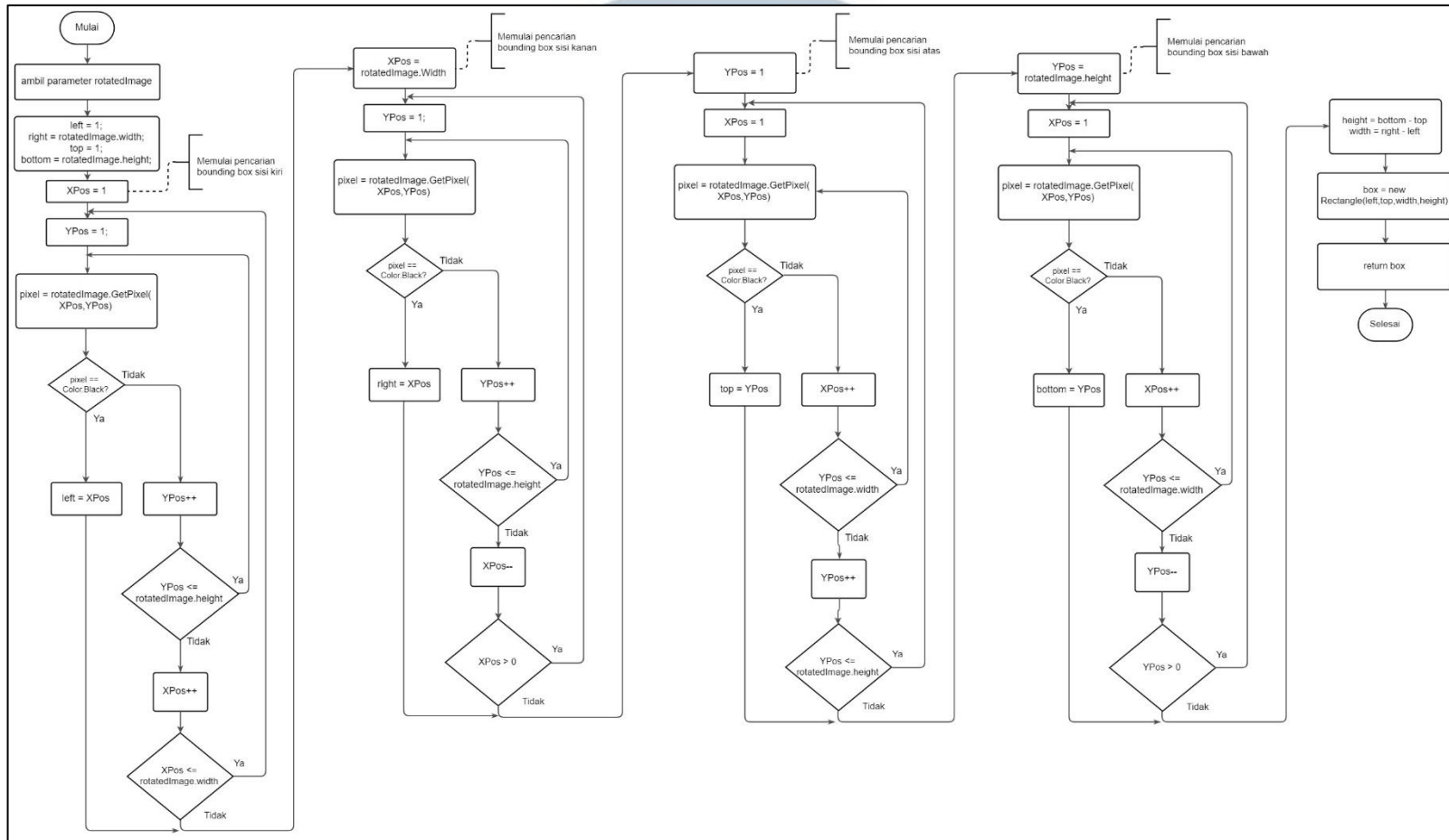


Gambar 3.4 Flowchart Metode Combined Vertical Projection

Gambar akan diputar dari sudut -10 derajat hingga 10 derajat dengan penambahan 0,1 derajat di setiap iterasinya. Setiap iterasi akan menghitung *energy* yang diperoleh dari jumlah piksel hitam yang terdapat pada setiap kolom gambar dan dipangkatkan dua. Pada setiap iterasi juga akan menghitung luas dari *bounding box* untuk membagi *energy* dan memperoleh nilai akhir *energy* pada setiap iterasi. Sudut dengan *energy* tertinggi akan digunakan untuk memutar *tresholddedImage* untuk menghasilkan *finalImage* yang akan digunakan pada proses pengenalan karakter pada Tesseract.

C. Flowchart Modul GetBoundingBox

Gambar 3.5 adalah *flowchart* modul *getBoundingBox*. Modul ini digunakan untuk mencari *bounding box* pada *rotatedImage* yang diterima sebagai parameter. Pencarian *bounding box* dimulai dengan penentuan *extreme points* pada *rotatedImage* dengan menggunakan proyeksi vertikal untuk sisi kiri dan kanan, dan proyeksi horizontal untuk sisi atas dan bawah. Piksel hitam pertama pada sisi kiri, kanan, atas, dan bawah masing-masing akan mencari *extreme point* pada sisi tersebut. Tinggi *bounding box* diperoleh dari jarak *extreme point* sisi atas dan bawah, sedangkan lebar *bounding box* diperoleh dari jarak *extreme point* sisi kiri dan kanan.

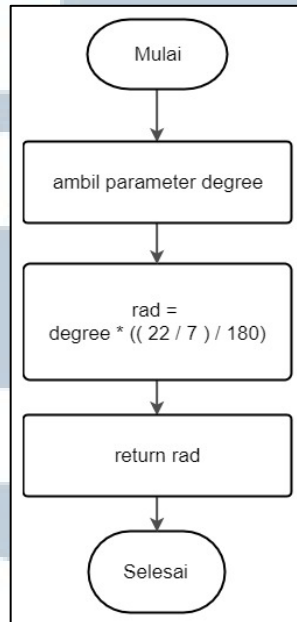


Gambar 3.5 Flowchart Modul GetBoundingBox

UNIVERSITAS
MULTIMEDIA
NUSANTARA

D. Flowchart Modul DegreeToRad

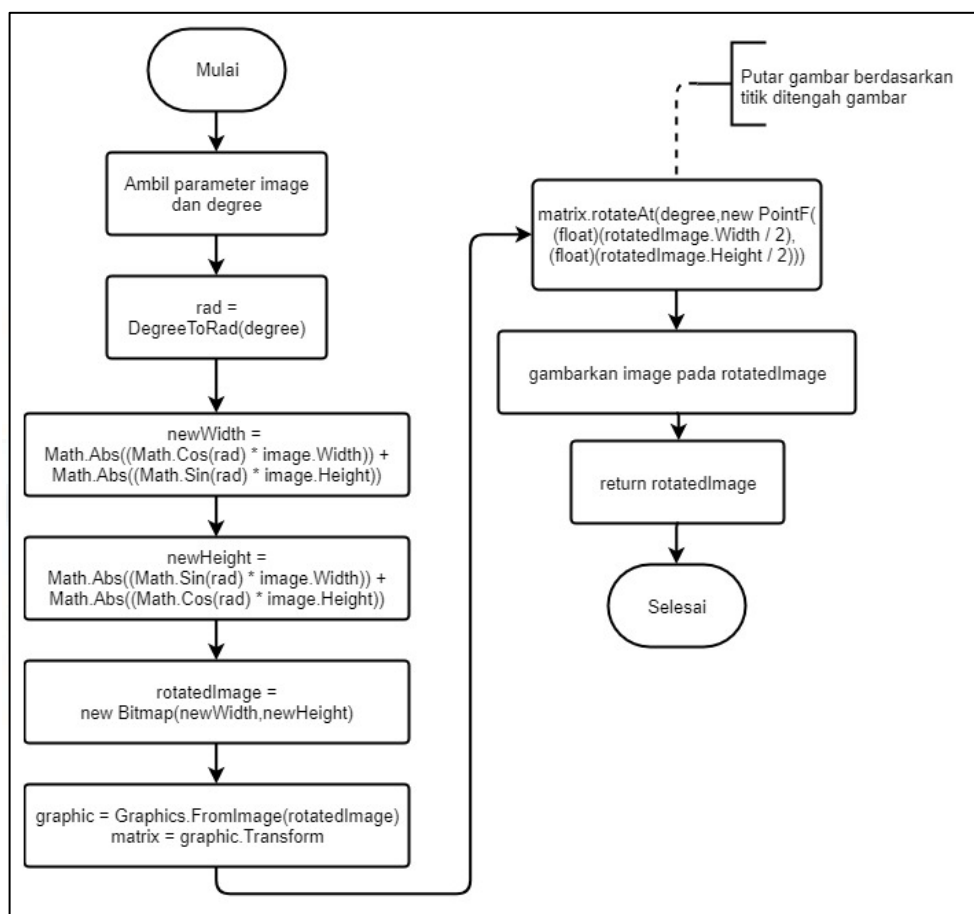
Gambar 3.6 menunjukkan *flowchart* modul DegreeToRad. Modul ini digunakan untuk mengkonversikan satuan derajat sudut ke dalam satuan radian. Modul ini menerima satu buah parameter, yaitu degree dalam tipe data *float* dan mengembalikan rad, yaitu nilai derajat sudut yang dikonversikan menjadi radian.



Gambar 3.6 *Flowchart* Modul DegreeToRad

E. Flowchart Modul RotateImage

Gambar 3.7 adalah *flowchart* modul RotateImage. Modul ini digunakan untuk memutar gambar. Modul ini menerima dua buah parameter, yaitu degree yang merupakan derajat sudut pemutaran yang akan dilakukan dan image yang berupa gambar yang akan diputar. Modul ini digunakan untuk memutar gambar berdasarkan satuan derajat dan mengubah dimensi gambar sehingga sesuai dengan gambar yang telah diputar.



Gambar 3.7 Flowchart modul RotateImage

3.2.3 Rancangan Antarmuka Aplikasi

Aplikasi akan terdiri dari 1 *window*/halaman dengan rancangan antarmuka sesuai pada Gambar 3.8. Rancangan antarmuka aplikasi akan disesuaikan dengan kebutuhan penelitian. Tombol Browse Image digunakan untuk memilih gambar dokumen yang akan digunakan untuk pengenalan karakter. Tombol Recognize Text digunakan untuk memulai proses pengenalan karakter setelah pemilihan gambar. Setelah proses pengenalan karakter selesai, teks yang berhasil dikenali akan di tampilkan pada *textbox*. Pengguna dapat meng-ekspor teks hasil pengenalan karakter ke dalam *file* .pdf maupun .txt dengan menggunakan tombol Export to pdf dan Export to txt. Derajat kemiringan dokumen yang terdeteksi dan lama waktu komputasi juga akan ditampilkan pada aplikasi.

Gambar dokumen	Textbox hasil pengenalan karakter
Browse Image	detected skew: X degree computation time: X seconds
Recognize Text	Export to pdf Export to txt

Gambar 3.8 Rancangan Antarmuka Aplikasi

UMN

UNIVERSITAS

MULTIMEDIA

NUSANTARA