

**IMPLEMENTASI *TEMPLATE MATCHING* DENGAN KORELASI
PEARSON UNTUK MENDETEKSI HUKUM TAJWID CITRA
TULISAN AL-QURAN PADA *PLATFORM* ANDROID**

SKRIPSI

**Diajukan Kepada :
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Menempuh Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer**

**oleh :
M. NUR FUAD
NIM. 10650015 / S-1**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2014**

**IMPLEMENTASI *TEMPLATE MATCHING* DENGAN KORELASI
PEARSON UNTUK MENDETEKSI HUKUM TAJWID CITRA
TULISAN AL-QURAN PADA *PLATFORM* ANDROID**

SKRIPSI

oleh :

**M. NUR FUAD
NIM. 10650015**

Telah Diperiksa dan Disetujui untuk Diuji :
Tanggal 20 November 2014

Pembimbing I,

Pembimbing II,

**Irwan Budi Santoso, M. Kom
NIP. 19770103 201101 1 004**

**Umaivatus Svarifah, M.A
NIP. 19820925 200901 2 005**

Mengetahui,
Ketua Jurusan Teknik Informatika

**Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008**

**IMPLEMENTASI *TEMPLATE MATCHING* DENGAN KORELASI
PEARSON UNTUK MENDETEKSI HUKUM TAJWID CITRA
TULISAN AL-QURAN PADA *PLATFORM* ANDROID**

SKRIPSI

oleh :

**M. NUR FUAD
NIM. 10650015**

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S. Kom)
Tanggal 27 November 2014

Susunan Dewan Penguji :

Tanda Tangan

- | | | |
|--------------------|---|-----|
| 1. Penguji Utama | : <u>Dr. Cahyo Crys dian</u>
NIP. 19740424 200901 1 008 | () |
| 2. Ketua Penguji | : <u>Dr. M. Amin Hariyadi, M.T</u>
NIP. 19670118200501 1 001 | () |
| 3. Sekretaris | : <u>Irwan Budi Santoso, M. Kom</u>
NIP. 19770103 201101 1 004 | () |
| 4. Anggota Penguji | : <u>Umaiyatus Syarifah, M.A</u>
NIP. 19820925 200901 2 005 | () |

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Cahyo Crys dian
NIP. 19740424 200901 1 008

HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ
اللهم صل وسلم وبارك على سيدنا ورسولنا وحبیبنا محمد

Karya ini kupersembahkan untuk :

Ayah dan Ibu

M. Musa dan Siti Mujiati

terima kasih atas semua doa, dukungan, dorongan, bantuan, kasih sayang dan semua yang beliau berdua berikan padaku.

Untuk dua adikku **Irsyad Adifa Muhammad** & **Muhammad Sholeh Al Fadhil**, sepupu-sepupuku Naila, Dela, Syaunqi, Aliza, Nabila, Kana, Aqul, Nabhan, Husein, Mas Aghis, dan Mas ulin

juga seluruh keluarga besarku Nenek, Paman, Bibi, Kakak & Adik terima kasih atas dukungan, doa, senyum, tawa, dan keberadan kalian yang membuatku berani berjuang sampai sejauh ini.

Untuk sahabat-sahabatku, Adhib, Arip, Ghulam, Bagus, Na', Aziz, Pudín, Fahim, Wahyu, Dian, Tria, Agus, Wahyu, Muiz, Jun serta Almarhum Muslih, semoga segala amal dan ibadahnya diterima Ilahi, terima kasih atas semua semangat, pengalaman dan bantuan perjuangan kalian selama ini, kalian semua memang yang terbaik. Semoga Allah SWT melindungi dan menjaga mereka semua.

Aamiin...

MOTTO

العلم نور

“Belajarlah, sesungguhnya setiap manusia dihargai karena ilmunya”

“Cahaya ilmu akan menerangi kegelapan ketidaktahuanmu”



PERNYATAAN KEASLIAN TULISAN

Yang bertanda tangan di bawah ini :

Nama : M. Nur Fuad
NIM : 10650015
Jurusan : Teknik Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa tugas akhir /skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pemikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan tugas akhir/skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 November 2014
Yang membuat pernyataan,

M. Nur Fuad
NIM. 10650015

KATA PENGANTAR



Segala puja dan puji syukur penulis panjatkan kepada Allah SWT atas berkat, rahmat, taufik dan hidayah-Nya sehingga penyusunan dan penulisan skripsi yang berjudul “Implementasi *Template Matching* Dengan Korelasi Pearson Untuk Mendeteksi Hukum Tajwid Citra Tulisan Al-Quran Pada *Platform Android*” dapat terselesaikan dengan baik.

Shalawat serta salam semoga tercurah terhadap junjungan Nabi Besar Muhammad saw. Beliau lah yang membimbing umat Islam dari gelapnya kekufuran menuju cahaya Islam yang terang benderang.

Dengan segala kerendahan hati, penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, dalam kesempatan ini penulis ingin mengucapkan terima kasih tak terhingga kepada :

1. Irwan Budi Santoso, M. Kom selaku Dosen Pembimbing I yang senantiasa membimbing, mengarahkan, memberi saran, kemudahan serta memberikan kepercayaan kepada penulis dalam pengerjaan skripsi.
2. Ibu Umairatus Syarifah, M.A selaku Dosen Pembimbing II, yang telah memberi masukan, bimbingan, kemudahan serta melancarkan proses penyelesaian skripsi.

3. Bapak-Ibu Dosen dan seluruh civitas akademik Fakultas Sains dan Teknologi yang telah memberikan ilmu dan kemudahan selama penulis berada di Fakultas Sains dan Teknologi Universitas Islam Negeri Malang. Pak Fatchur, Pak Cahyo, Pak Amin, Pak Syahid, Pak Yaqin, Pak Faisal, Pak Fresy, Pak Fatchul, Bu Ririen, Pak Totok, Pak Syahid, Bu Roro, Bu Hani, Pak Sauqi, Pak Gun, Pak Soffin, Pak Agung, Pak Aziz. Semoga ilmu yang telah didapat bermanfaat dalam kehidupan ini.
4. Teman-teman Infinity seperjuangan, baik yang sama-sama sedang berjuang maupun yang akan berjuang.
5. Dan seluruh pihak-pihak yang telah membantu dalam penyelesaian skripsi ini.

Hanya ucapan terima kasih dan doa tulus penulis berikan atas apa yang telah mereka berikan semoga apa yang telah mereka lakukan dapat memantulkan kebaikan kembali kepada mereka. *Aamiin yaa robbal 'aalamiin.*

Malang, 20 November 2014

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	ii
HALAMAN PERSEMBAHAN	iv
MOTTO	v
PERNYATAAN KEASLIAN TULISAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
ABSTRAK	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Metode Penelitian.....	5
1.7 Sistematika Penelitian	7
BAB II TINJAUAN PUSTAKA.....	8
2.1 <i>Template Matching</i>	8
2.2 Korelasi Pearson.....	11
2.3 Ilmu Tajwid.....	12
2.4 Transliterasi.....	16
2.5 <i>Platform</i> Android	18
BAB III ANALISA DAN PERANCANGAN.....	20
3.1 Perancangan dan Pengumpulan Data	20
3.2 Desain Sistem Aplikasi	20
3.3 Perancangan <i>Interface</i>	40
BAB IV HASIL DAN PEMBAHASAN	43
4.1 Implementasi Sistem	43
4.2 Uji Coba Sistem	54

4.3 Integrasi Aplikasi Dengan Pembelajaran Dan Tata Cara Membaca Al-Quran Yang Baik Dan Benar	66
BAB V PENUTUP	69
5.1 Kesimpulan	69
5.2 Saran.....	70
Daftar Pustaka.....	71
Lampiran	73



DAFTAR TABEL

Tabel 2.1 Transliterasi.....	16
Tabel 4.1 Pencocokan Huruf	55
Tabel 4.2 Pencocokan Harakat	59
Tabel 4.3 Pengecekan Tajwid	61
Tabel 4.4 Transliterasi	63
Tabel 4.5 Total Akurasi	65



DAFTAR GAMBAR

Gambar 1.1	Prosedur Penelitian.....	5
Gambar 2.1	Metode <i>Template Matching</i>	10
Gambar 3.1	<i>Image 20x20</i>	20
Gambar 3.2	Desain Sistem Aplikasi	22
Gambar 3.3	<i>Flowchart capture</i> kamera	23
Gambar 3.4	<i>Flowchart scaling image</i> (250 x 200)	25
Gambar 3.5	<i>Flowchart cropping</i> berdasarkan bingkai <i>cropping</i>	26
Gambar 3.6	<i>Flowchart</i> binerisasi	28
Gambar 3.7	<i>Flowchart cropping</i> berdasarkan nilai biner terluar.....	29
Gambar 3.8	<i>Flowchart scaling image</i> kedua (20 x 20).....	31
Gambar 3.9	<i>Flowchart grayscale</i>	32
Gambar 3.10	<i>Flowchart</i> template matching by korelasi.....	33
Gambar 3.11	<i>Flowchart</i> korelasi Pearson.....	34
Gambar 3.12	<i>Flowchart</i> cek tajwid.....	36
Gambar 3.13	<i>Flowchart</i> transliterasi	38
Gambar 3.14	Halaman Bantuan	40
Gambar 3.15	Halaman Kamera.....	41
Gambar 3.16	Prosedur Utama	42
Gambar 3.17	Halaman Bantuan	42
Gambar 4.1	<i>Source code</i> menampilkan <i>view</i> kamera	43
Gambar 4.2	Halaman Kamera.....	44
Gambar 4.3	<i>Source code</i> simpan <i>bitmap</i>	44
Gambar 4.4	<i>Source code</i> <i>bitmap scaling</i>	45
Gambar 4.5	<i>Source code</i> menampilkan <i>imScale.png</i>	45
Gambar 4.6	Halaman Utama.....	45
Gambar 4.7	Halaman Utama setelah bingkai <i>cropping</i> digeser.....	46
Gambar 4.8	<i>Source code</i> <i>cropping</i> berdasarkan bingkai <i>cropping</i>	46
Gambar 4.9	<i>Source code</i> binerisasi	47
Gambar 4.10	<i>Source code</i> <i>cropping</i> berdasarkan nilai biner terluar.....	48
Gambar 4.11	Hasil penyimpanan <i>imCrop.png</i>	48
Gambar 4.12	<i>Source code</i> menampilkan semua <i>imCrop</i>	49
Gambar 4.13	<i>Source code</i> <i>grayscale</i>	49
Gambar 4.14	<i>Source code</i> korelasi pearson	50
Gambar 4.15	<i>Source code</i> cek tajwid.....	51
Gambar 4.16	<i>Source code</i> transliterasi a.....	52
Gambar 4.17	<i>Source code</i> transliterasi b.....	53
Gambar 4.18	Halaman Hasil	54

ABSTRAK

Fuad, M. Nur. 2014. Skripsi. **Implementasi *Template Matching* Dengan Korelasi Pearson Untuk Mendeteksi Hukum Tajwid Citra Tulisan Al-Quran Pada Platform Android.** Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Irwan Budi Santoso, M. Kom (II) Umaiyatus Syarifah M.A.

Kata Kunci : *Template Matching*, Korelasi Pearson, Ilmu Tajwid

Hukum mengamalkan ilmu tajwid dalam membaca al-Quran adalah wajib Ain bagi setiap muslim. Berdasarkan hal tersebut, dilakukan penelitian tentang pengenalan tajwid melalui suatu aplikasi *mobile* yang mampu digunakan dan dibawa ke berbagai tempat. Aplikasi yang dibangun berupa aplikasi kamera dalam *platform* Android, yang mampu mengenali huruf dan harakat dalam al-Quran sehingga mampu digunakan untuk membantu mengenali tajwid dan transliterasinya. Aplikasi ini memanfaatkan metode *template matching* dengan korelasi pearson untuk mengenali *image* huruf dan harakat. Pengenalan huruf dan harakat dilakukan dengan mencocokkan *input* kamera dengan *template* yang terdapat di dalam aplikasi, selanjutnya dilakukan pengecekan tajwid dari hasil pencocokan tersebut. Untuk memperingan aplikasi yang ditanamkan pada *device mobile*, *array grayscale* dibuat sebagai *template* yang akan dicocokkan dengan *input*. Akurasi aplikasi ini mencapai 89 % untuk pencocokan huruf, 95 % untuk pencocokan harakat, dan 100 % untuk pengecekan tajwid dan transliterasi.

ABSTRACT

Fuad, M. Nur. 2014. Thesis. **Implementation of Template Matching by Pearson Correlation for Tajweed detection of Al-Quran Image on Android Platform.** Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Islamic State University of Malang. Adviser : (I) Irwan Budi Santoso, M. Kom (II) Umaiatus Syarifah M.A.

Keyword : *Template Matching*, Pearson Correlation, Tajweed

Tajweed practice law in reading the Koran is obligatory for every Muslim. Based on this fact, the research has been done about the identified of tajweed through a mobile application that is able to be used and taken to various places. The applications are made in the form of a camera application in Android platform, that is able to recognize letters and the vowels in the Koran so that it can be used to help identify the tajweed and transliteration. This application utilizes template matching method with pearson correlation to recognize letters and the vowel image. The identified of letters and the vowels can be done with matching camera input with the template that contained in the application, then checking tajweed of the matching results. To lighten the applications deployed on mobile devices, grayscale array has made as template to be matched with the input. The application accuracy is 89% for letters matching, 95% for vowel matching, and 100% for checking of tajweed and transliteration.

مستخلص البحث

محمد نور فؤاد، 2014، بحث جامعي، تطبيق *Template Matching* مع ارتباط Pearson لكشف أحكام التجويد من صورة كتابة القرآن الكريم على خطة أندرويد، قسم المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم بمالانج.

المشرف: (1) إيوان بودي سانطاسا الماجستير، (2) أمية الشريفة الماجستير

الكلمات المفتاحية: *Template Matching*، ارتباط بيرسون، علم التجويد

إن تطبيق علم التجويد في قراءة القرآن هو فرض عين على كل مسلم. بناء على ذلك أجري البحث على تعريف علم التجويد من خلال برمجية الجوال التي يمكن استخدامها و حملها إلى أماكن مختلفة. بنيت البرمجية في شكل برمجية كاميرا في خطة أندرويد التي تعرف الأحرف وأشكالها في القرآن، لذلك يمكن استخدامها لمساعدة في تعرف أحكام التجويد و النقحرة. هذه البرمجية تستخدم طريقة *Template Matching* مع ارتباط Pearson لتعرف صورة الأحرف وأشكالها. ويتم التعرف عن طريق مطابقة مدخلات كاميرا مع قوالب موجودة في البرمجية ثم أقيم تحقق التجويد من نتيجة المطابقة. ولتخفيف البرمجية المدججة على جهاز الجوال، استخدم *Array Grayscale* كقالب يكون مطابقا مع المدخلات. الدقة من هذه البرمجية تصل 89% في مطابقة الأحرف و 95% في مطابقة الأشكال و 100% في تحقق أحكام التجويد و النقحرة.

BAB I PENDAHULUAN

1.1 Latar Belakang

Hukum mempelajari ilmu tajwid adalah wajib kifayah bagi setiap muslim namun hukum mengamalkan ilmu tajwid dalam membaca al-Quran adalah wajib ain bagi setiap muslim (Muthohhar et al, 2012). Al-Quran mengajarkan umat Islam untuk membaca al-Quran dengan menggunakan bacaan yang benar, sebagaimana firman Allah SWT dalam Al Baqarah (2) : 121, serta diperkuat dengan surat Al Muzammil (73) : 4 yang memerintahkan membaca al-Quran dengan *tartil*.

الَّذِينَ آتَيْنَاهُمُ الْكِتَابَ يَتْلُونَهُ حَقَّ تِلَاوَتِهِ أُولَئِكَ يُؤْمِنُونَ بِهِ وَمَنْ يَكْفُرْ بِهِ فَأُولَئِكَ هُمُ

الْحَاسِرُونَ ﴿١٢١﴾

Artinya : “Orang-orang yang telah Kami berikan Al Kitab kepadanya, mereka membacanya dengan bacaan yang sebenarnya, mereka itu beriman kepadanya. Dan barangsiapa yang ingkar kepadanya, maka mereka itulah orang-orang yang rugi”.(QS. Al Baqarah (2) : 121)

أَوْ زِدْ عَلَيْهِ وَرَتِّلِ الْقُرْآنَ تَرْتِيلاً ﴿٤﴾

Artinya : “Atau lebih dari seperdua itu. Dan bacalah al-Quran itu dengan perlahan-lahan”. (QS. Al Muzammil (73) : 4)

Tartil berasal dari kata *ratala* yang memiliki arti sama dengan *hasan* atau *tahsin* yaitu memperbaiki bacaan. Dalam al-Quran kata *tartil* terdapat dalam dua tempat, yaitu surat Al Furqan (25) : 32 dan Al Muzammil (73) : 4, dan diartikan sebagai bacaan yang teratur dan benar (Ridwan, 2007).

Untuk mampu membaca al-Quran secara *tartil*, diperlukan pemahaman terhadap ilmu tajwid. Secara bahasa tajwid memiliki arti *at-tahsiinu* yang bermakna membaguskan. Secara terminologi menurut para ulama ilmu tajwid memiliki arti sebagai berikut: “Ilmu yang dengannya dapat mengetahui cara memberikan hak dan mustahak kepada setiap huruf yang terdiri atas sifat-sifat huruf, hukum mad dan lain sebagainya” (Ridwan, 2007).

Dalam ilmu tajwid terdapat beberapa pembagian hukum, diantaranya yakni hukum nun mati dan tanwin yang dibagi menjadi 4 hukum (*Idzhar, Idgham, Iqlab* dan *Ikhfa'*), hukum mim mati yang dibagi menjadi 3 hukum (*Idzhar Syafawi, Idgam Mutamassilain* dan *Ikfa' Syafawi*), hukum *al-ta'rif* yang dibagi menjadi 2 hukum (*Idzhar Qomariah* dan *Idgham Syamsiyah*) dan lain-lain (Muthohhar et al, 2012).

Mempelajari ilmu tajwid tidak hanya melalui buku dan kitab saja, namun juga dapat melalui komputer atau smartphone yang saat ini telah menjadi media yang semakin penting kegunaannya dalam kehidupan sehari-hari. *Smartphone* sendiri dapat digunakan dalam banyak hal, mulai dari sms dan telepon sampai koneksi internet dan perhitungan taktis melalui media pengolah angka seperti excel yang telah tertanam di dalamnya.

Dengan teknologi *smartphone* yang semakin maju, *smartphone* juga dapat dimanfaatkan sebagai media untuk mempelajari tajwid, seperti mengenali hukum tajwid melalui citra dari al-Quran. Citra tersebut kemudian akan melalui tahap preprosesing citra meliputi *grayscale*, segmentasi, *resize* dan *binerisasi* dengan memanfaatkan *resource* dari *processor* dan *Random Acces Memory smartphone* untuk mengambil data yang dibutuhkan. Selanjutnya *template maching* dapat

dilakukan dengan pendekatan korelasi Pearson untuk menentukan jenis huruf-huruf yang akan diperiksa tajwidnya berdasarkan kitab *Tuhfah At-Tullab* karya Ahmad bin Muthohhar, Abdur Rahman Al-Murofi, dan H. Adnin. Digunakan kitab *Tuhfah At-Tullab* karena kitab ini sudah mencakup hukum-hukum dasar tajwid beserta contoh dan penjelasannya.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian, maka dapat diidentifikasi beberapa masalah sebagai berikut:

- a. Bagaimana membangun Aplikasi Implementasi *Template Matching* dengan Korelasi Pearson untuk Mendeteksi Hukum Tajwid Citra Tulisan al-Quran pada *Platform* Android?
- b. Seberapa baik akurasi metode *Template Matching* dengan Korelasi dalam mendeteksi hukum Tajwid Citra Tulisan al-Quran?

1.3 Batasan Masalah

Untuk memfokuskan penelitian ini, maka terdapat batasan masalah sebagai berikut :

- a. Data yang digunakan adalah *image* dari 3 huruf dan 3 harakat yang diambil melalui kamera. Diambil 3 huruf dan 3 harakat karena hukum tajwid sudah dapat diambil dari susunan tersebut dan juga untuk memudahkan transliterasi.
- b. Al-Quran yang digunakan adalah al-Quran yang diterbitkan oleh penerbit J-Art.
- c. Korelasi Pearson digunakan untuk metode *template matching*.
- d. Aturan ilmu tajwid mengikuti kitab *Tuhfah At-Tullab*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah sebagai berikut:

- a. Membangun aplikasi Implementasi *Template Matching* dengan Korelasi Pearson untuk Mendeteksi Hukum Tajwid Citra Tulisan al-Quran pada *Platform* Android.
- b. Akurasi metode *Template Matching* dengan Korelasi dalam mendeteksi hukum Tajwid Citra Tulisan al-Quran.

1.5 Manfaat Penelitian

Penelitian ini diharapkan mempunyai 2 manfaat, yaitu manfaat teoritis dan manfaat praktis.

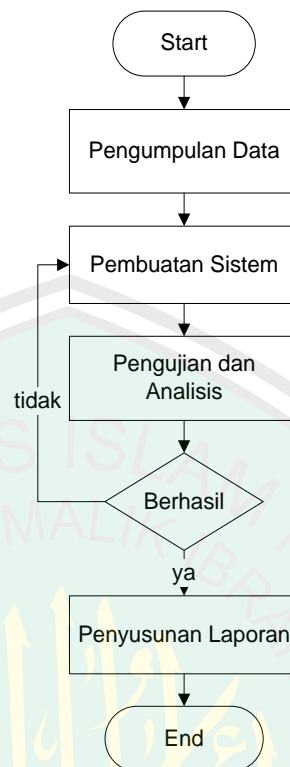
- a. Manfaat teoritis

Secara teoritis, penelitian ini diharapkan dapat memperkaya wawasan dalam pengetahuan ilmu Teknik Informatika khususnya bidang Pengolahan Citra Digital dalam bidang *template matching* dengan korelasi.

- b. Manfaat Praktis

1. Membantu mengetahui hukum tajwid pada tulisan al-Quran bagi masyarakat yang masih awam dan belum faham tentang ilmu tajwid.
2. Diharapkan hasil penelitian ini dapat dipakai sebagai informasi tambahan bagi penelitian berikutnya yang berhubungan dengan *template matching* dengan Korelasi.

1.6 Metode Penelitian



Gambar 1.1. Prosedur Penelitian

Pada penelitian ini terdapat beberapa tahap penelitian yang mengacu pada prosedur penelitian pada Gambar 1.1 yang meliputi :

a. Pengumpulan Data

Pengumpulan data dilakukan dengan cara mengumpulkan data tentang pembuatan aplikasi kamera pada Android. Data ini didapat dengan cara mempelajari tutorial pembuatan aplikasi Android serta mempelajari buku-buku Android. Pengumpulan data selanjutnya yakni pengumpulan referensi yang berkaitan dengan metode *Template Matching* dengan korelasi dimana metode ini akan digunakan untuk mencocokkan *source image* dengan *template image* yang sudah terdapat dalam sistem. Referensi ini didapat dari jurnal-jurnal ilmiah dan buku-buku *image processing*.

Selanjutnya dilakukan pengumpulan data *template image* huruf hijaiyah sebagai database yang akan menjadi acuan pencocokan dengan gambar sumber. Pengumpulan data *template image* ini dilakukan dengan cara memfoto huruf-huruf hijaiyah yang terdapat dalam al-Quran terbitan penerbit J-Art. Kemudian dilakukan pengumpulan informasi tentang ilmu tajwid dengan cara mempelajari kitab *Tuhfatul Tullab* karya Ahmad bin Muthohhar, Abdur Rahman Al-Murofi, dan H. Adnin.

b. Pembuatan Sistem

Pembuatan sistem ini meliputi penerapan metode *template matching* dengan korelasi sebagai metode untuk mencocokkan *image source* dengan *template image*. Mulai dari desain *interface* sistem sampai pengkodean dilakukan dengan menggunakan bahasa pemrograman *java*.

c. Pengujian dan Analisis

Pengujian dan analisis dilakukan setelah sistem selesai dikerjakan. Dalam hal ini akan dibahas tentang pengujian berdasarkan perencanaan dari sistem yang dibuat serta melakukan analisis pada sistem untuk mengetahui apakah metode *template matching* dengan korelasi sesuai dengan objek penelitian atau tidak. Dan jika terjadi kesalahan maka kembali pada prosedur pembuatan sistem.

d. Penyusunan Laporan

Kesimpulan serta laporan didapat saat pengujian dan analisis berjalan dengan baik dan metode yang digunakan dalam penelitian ini dapat diterapkan pada objek penelitian.

1.7 Sistematika Penelitian

Penulisan skripsi ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Bab ini menerangkan tentang latar belakang masalah, identifikasi masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penelitian.

BAB II Landasan Teori

Bab landasan teori berisi beberapa teori yang mendasari penyusunan tugas akhir ini. Dalam bab ini membahas tentang dasar teori yang berkaitan dengan pembahasan tentang tajwid dan *Platform* Android.

BAB III Analisa dan Perancangan

Bab ini menjelaskan analisa kebutuhan sistem untuk membuat aplikasi meliputi desain sistem dan langkah-langkah pembuatan aplikasi Implementasi *Template Matching* dengan Korelasi Pearson untuk Mendeteksi Hukum Tajwid Citra Tulisan al-Quran pada *Platform* Android.

BAB IV Hasil dan Pembahasan

Bab ini menjelaskan hasil pengujian aplikasi Implementasi *Template Matching* dengan Korelasi Pearson untuk Mendeteksi Hukum Tajwid Citra Tulisan al-Quran pada *Platform* Android yang telah diterapkan.

BAB V Penutup

Bab ini berisi kesimpulan dan saran.

BAB II TINJAUAN PUSTAKA

2.1 *Template Matching*

Template matching adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan). Teknik ini banyak digunakan dalam bidang industri sebagai bagian dari *quality control* (Wardhana, 2008). Metode *Template matching* merupakan salah satu metode yang digunakan untuk menjelaskan bagaimana otak mengenali kembali bentuk-bentuk atau pola-pola (Leksono, 2010).

Pada dasarnya *template matching* adalah sebuah tehnik dalam pengolahan citra digital untuk menemukan bagian-bagian kecil dari gambar yang cocok dengan *template* gambar. Energi cahaya yang terpancar dari suatu bentuk mengenai retina mata dan diubah menjadi energi neural yang kemudian dikirim ke otak. Selanjutnya terjadi pencarian diantara *template-template* yang ada. Jika sebuah *template* yang ditemukan sesuai dengan pola tadi, maka subjek dapat mengenal bentuk tersebut. Setelah kecocokan antara objek dan *template* terjadi, proses lebih lanjut dan intepretasi terhadap objek dapat terjadi (Pramana et al, 2010).

Template matching merupakan proses pengenalan karakter dengan cara membandingkan karakter yang ada pada dokumen dengan karakter pada *template*, untuk mencari karakter yang memiliki nilai kemiripan terbesar dengan *template*. *Template matching* sering digunakan untuk pengenalan karakter hasil *print* dari

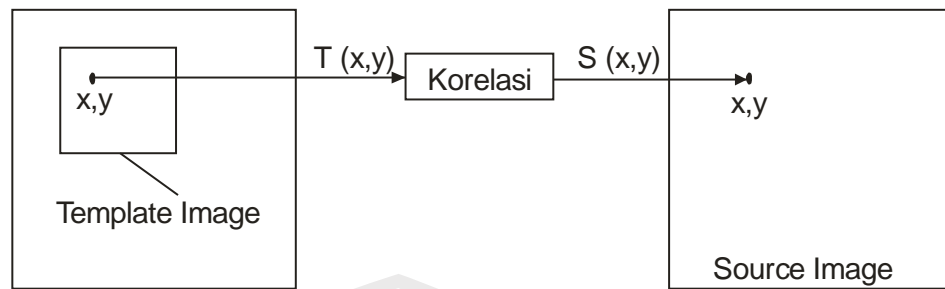
beberapa objek sederhana. Setiap piksel dari karakter memiliki nilai 0 sedangkan *background* dari karakter tersebut memiliki nilai 255. Proses *template matching* dilakukan dengan menggerakkan *template* keseluruhan posisi yang mungkin pada citra (Hastiana, 2010).

Prinsip metode ini adalah membandingkan antara *image* objek yang akan dikenali dengan *template* yang ada. *Image* objek yang akan dikenali mempunyai tingkat kemiripan tersendiri terhadap masing-masing *template*. Pengenalan dilakukan dengan melihat nilai tingkat kemiripan tertinggi dan nilai batas ambang pengenalan dari *image* objek tersebut. Bila nilai tingkat kemiripan berada di bawah nilai batas ambang maka *image* objek tersebut dikategorikan sebagai objek tidak dikenal (Zainudin, 2011).

Pada prinsipnya metode *template matching* memiliki karakteristik antara lain :

1. Relatif mudah untuk diaplikasikan dalam teknik pengolahan citra digital.
2. Hasilnya relatif sangat akurat karena mendeteksi kesalahan hingga ukuran pixel.
3. Metode ini cukup rentan terhadap perbedaan orientasi antara citra sampel (*template image*) dengan citra yang akan diidentifikasi (*source image*), yang meliputi : ukuran, posisi dan kualitas citra.
4. Untuk mendapatkan hasil yang maksimal maka metode ini sangat tergantung pada teknik pengolahan citra digital yang lain seperti *enhancement*, *color filtering* dan lain-lain. (Andriessen 2011).

Gambaran umum tentang pencocokan dengan menggunakan metode *template matching* terdapat pada Gambar 2.1.



Gambar 2.1. Metode *Template Matching*

Sumber : *Template Matching Persentation Based on a project by Roland Miezianko.*

Proses *matching* dengan korelasi menggerakkan *template image* pada semua posisi yang memungkinkan dalam *source image* dan menghitung *index* numerik yang akan mengindikasikan seberapa baik kecocokan *template* pada posisi tersebut. Pencocokan dilakukan hingga ukuran antara satu *pixel* dengan *pixel* lainnya (Latecki, 2002).

Perbandingan antara *template* dengan keseluruhan objek pada citra dapat dilakukan dengan menghitung selisih jaraknya seperti berikut :

$$D(m, n) = \sum_j \sum_k [f(i, k) - T(j - m, k - m)] \quad (2.1)$$

Dengan $f(j, k)$ menyatakan citra tempat objek yang akan dibandingkan dengan *template* $T(j, k)$. Sedangkan $D(m, n)$ menyatakan jarak antar *template* dengan objek pada citra. Pada umumnya ukuran *template* jauh lebih kecil dari ukuran citra.

Secara ideal, *template* dikatakan cocok dengan objek pada citra jika $D(m, n) = 0$, namun kondisi seperti ini sulit dipenuhi apalagi bila *template* merupakan suatu citra *grayscale*. Oleh karena itu, aturan yang digunakan untuk menyatakan *template* cocok dengan objek adalah bila $D(m, n) < LD(m, n)$, dengan $LD(m, n)$ merupakan nilai *threshold* (Putra, 2010).

2.2 Korelasi Pearson

Korelasi adalah suatu metode statistik untuk menentukan letak hubungan antar variabel. Sedangkan korelasi Pearson adalah suatu bentuk rumus yang digunakan untuk mencari hubungan (*relationship*) antara dua variabel, yaitu variabel bebas (*independent variable*) dan variabel terikat (*dependent variable*). *Independent variable* adalah variabel bebas yang dapat dikontrol dan dimanipulasi sedangkan *dependent variable* adalah variabel terikat yang tidak dapat dikontrol atau dimanipulasi (Bluman, 2004).

Korelasi Pearson digunakan untuk mengetahui tingkat atau keeratan hubungan secara linier antara dua variabel atau dua fitur objek (Irwan, 2013). Berdasarkan jumlah variabel yang dipelajari, *relationship* dibagi menjadi 2 yakni *simple relationship* dan *multiple relationship*. Pada *simple relationship* hanya ada 2 variabel, sedangkan pada *multiple relationship* memiliki banyak variabel. *Relationship* sendiri juga dibagi 2 berdasarkan sifat variabelnya, yakni *positive relationship* jika semua variabel bertambah dan berkurang pada waktu yang sama dan *negative relationship* jika satu variabel bertambah maka variabel yang lain berkurang.

Selanjutnya koefisien korelasi adalah standarisasi angular separation dengan pengurangan nilai koordinat dengan nilai mean. Nilainya antara -1 dan +1. Serupa dengan *angular separation*, koefisien korelasi menghitung nilai kesamaan dibanding ketidaksamaan, jadi semakin tinggi nilainya, menunjukkan 2 vektor semakin mirip (Putra, 2010).

Berikut rumus koefisien korelasi pearson:

$$r_{ij} = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i) * (x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (x_{ik} - \bar{x}_i)^2 * \sum_{k=1}^n (x_{jk} - \bar{x}_j)^2}} \quad (2.2)$$

Dimana $\bar{x}_i = \frac{1}{n} \sum_{k=1}^n x_{ik}$ dan $\bar{x}_j = \frac{1}{n} \sum_{k=1}^n x_{jk}$.

Ket : r_{ij} = koefisien korelasi

x_{ik} = *template image*

x_{jk} = *source image2*

\bar{x}_i = nilai rata-rata *template image*

\bar{x}_j = nilai rata-rata *source image*

n = jumlah pixel pada citra

2.3 Ilmu Tajwid

Tajwid menurut Kamus Besar Bahasa Indonesia adalah cara membaca Al-Quran dengan lafal atau bacaan yang benar. Tajwid menurut bahasa merupakan bentuk masdar dari kata “*jawwada, yujawwidu, tajwidan*” yang memiliki makna “*at-tahsin*” (memperbaiki). Sedangkan menurut istilah, tajwid adalah ilmu yang mempelajari cara mengucapkan huruf-huruf al-Quran yang meliputi sifat, makhroj, dan ahkamul huruf (Muthohhar et al, 2012).

Ilmu tajwid ialah ilmu yang digunakan untuk mengetahui bagaimana sebenarnya membunyikan huruf-huruf dengan benar, baik huruf yang berdiri sendiri maupun huruf dalam rangkaian dengan sebaik-baiknya sesuai dengan tuntunan dan ajaran yang dicontohkan Nabi Muhammad SAW (Rochmah, 2010).

Teori tajwid secara umum dibagi menjadi hukum nun mati dan tanwin, hukum mim mati, hukum *al-ta'rif*. Dalam hukum *nun* mati dan tanwin, dibagi lagi menjadi 4 yakni *Idzhar, Idgham, Iqlab, dan Ikhfa'*. Sedangkan hukum *mim* mati dibagi menjadi 3 yakni *Idzhar Syafawi, Idgham Mutamassilain, dan Ikhfa'*

Syafawi. Hukum *alif lam* dibagi menjadi 2 bagian menjadi *Alif Lam Syamsiah* dan *Alif lam Qomariah*.

Apabila ada *nun* mati atau tanwin bertemu salah satu huruf hijaiyah, maka ada 4 hukum bacaan. Pertama *Idzhar*, yaitu apabila ada *nun* mati atau tanwin bertemu dengan salah satu dari 6 huruf *halqi* (huruf yang keluar dari tenggorokan). Huruf *halqi* tersebut adalah sebagai berikut :

1. huruf *hamzah* (ء)
2. huruf *Ha* (هـ)
3. huruf *ain* (ع)
4. huruf *ha* (ح)
5. huruf *gain* (غ)
6. huruf *kha* (خ)

Cara membaca *Idzhar* yaitu membaca dengan terang atau mengeluarkan huruf dari *makhrojnya* dengan tanpa bercampur *ghunnah* (mendengung) dan *syaddah*. Contoh : مَنْ أَمِنَ، أَنْعَمْتَ.

Hukum *nun* mati dan tanwin yang kedua adalah *Idgham*, yaitu apabila ada *nun* mati atau tanwin bertemu dengan salah satu huruf yang terkumpul dalam *lafadz* يرملون (ن , و , ل , م , ر , ي), akan tetapi 6 huruf tersebut dibagi menjadi 2 hukum yaitu:

1. *Idgham Bighunnah*, yaitu apabila ada *nun* mati atau tanwin bertemu dengan salah satu huruf yang terkumpul dalam *lafadz* ينمو (و , م , ن , ي) .

Cara membaca Idgham Bighunnah yakni dengan memasukkan huruf yang satu pada huruf yang lain disertai dengan dengung. Contoh : مَنْ يَقُولُ ,

يَوْمَئِذٍ نَاعِمَةٌ . Hukum ini tidak berlaku jika *nun* mati atau tanwin bertemu

dengan salah satu huruf tersebut dalam satu kalimat, jika hal tersebut terjadi maka wajib dibaca *Idzhar* agar tidak serupa dengan *lafadz mudha'af*. Contoh : دُنْيَا, صِنْوَانٍ .

2. *Idgham Bighairi Ghunnah (Bilaa Ghunnah)*, yaitu apabila ada *nun* mati atau tanwin bertemu dengan huruf *lam* (ل) dan *ro* (ر). Cara membaca *Idgham Bilaa Ghunnah* yakni dengan memasukkan huruf yang satu pada huruf yang lain tanpa disertai dengan dengung kemudian huruf *lam* (ل) dan *ro* (ر) tersebut bunyinya wajib diulang-ulangi (digetarkan). Contoh :

مِنْ رَبِّهِمْ, وَلَكِنْ لَا .

Hukum *nun* mati dan tanwin yang ketiga adalah *Iqlab*, yaitu apabila ada *nun* mati atau tanwin bertemu dengan huruf *ba* (ب), maka wajib menukar bunyi *nun* mati dengan suara *mim* disertai dengung dan samar yang berlangsung dalam satu kalimat atau dua kalimat. Contoh : مِنْ بَعْدِ , أَنْبِئُهُمْ .

Hukum *nun* mati dan tanwin yang keempat adalah *Ikhfa'*, yaitu apabila ada *nun* mati atau tanwin bertemu dengan salah satu dari 15 huruf yang terangkum

dalam awal tiap kata pada bait berikut : صِفْ ذَا ثَنَا كَمْ جَا دَ شَخْصٌ قَدْ سَمَّاؤُكُمْ طَيِّبًا

زِدْنِي تُقِي ضَعَّ ظًا لِمَا . *Ikhfa'* artinya samar sehingga ketika membaca *Ikhfa'* maka

dibaca dengan samar antara *Idgham* dan *Idzhar* disertai dengan dengung (*ghunnah*). Atau melafalkan *nun* mati atau tanwin akan tetapi *makhrojnya* sudah

masuk pada huruf selanjutnya. Contoh : مَنْ كَنْ , مُنْدِرٌ .

Kemudian apabila ada *mim* mati bertemu dengan salah satu huruf hijaiyah, maka ada 3 hukum bacaan yakni *Ikfa' Syafawi*, *Idgham Mitslain* dan *Idzhar Syafawi*. Hukum *mim* mati yang pertama adalah *Ikfa' Syafawi*, yaitu apabila ada *mim* mati bertemu dengan *ba* (ب), maka akan dibaca dengan menyamakan

bacaan huruf tadi pada bibir disertai dengan dengung. Contoh : أَمْ بِهِ جِنَّةٌ .

Hukum *mim* mati yang kedua adalah *Idgham Mitslain*, yaitu apabila ada *mim* mati bertemu dengan *mim* (م). Cara membacanya seperti membaca *mim* (م) bertasydid (memasukkan *mim* (م) yang awal pada *Mim* (م) yang kedua) disertai

dengung. Contoh : أَمْ مَنْ .

Hukum *mim* mati yang ketiga adalah *Idzhar Syafawi*, yaitu apabila ada *mim* mati bertemu dengan salah satu huruf hijaiyah selain *ba* (ب) dan *mim* (م).

Cara membaca *Idzhar Syafawi* mirip dengan *Idzhar* yaitu membaca dengan terang

atau mengeluarkan huruf dari *makhroj* tanpa bercampur dengan *ghunnah* (mendengung) dan *tasydid*. Contoh : *عَلَيْهِمْ غَيْرٌ , أَمْ عِنْدَهُمْ*.

Kemudian apabila ada *al-ta'rif* (*alif lam* yang dirangkai dengan kalimat *isim*) bertemu huruf hijaiyah, maka ada 2 hukum bacaan yakni *Idzhar Qomariah* dan *Idgham Syamsiyah*. Hukum *al-ta'rif* yang pertama adalah *Idzhar Qomariah* yaitu apabila ada *al-ta'rif* bertemu dengan salah satu dari huruf yang terangkai dalam *lafadz*: *أَنْبَغِ حَجَّكَ وَخَفِ عَقِيمَةَ*. Disebut *al-Qomariah*, sebab seperti lamnya *lafadz* *أَلْقَمَرُ* dalam wajibnya dibaca *Idzhar* (jelas). Hukum *al-ta'rif* yang kedua adalah *Idgham Syamsiyah*, yaitu apabila *al-ta'rif* bertemu 14 huruf yang terkumpul dalam awal kata bait

طَبُّ ثُمَّ صِلَ رَحْمًا تَفْرُضِيفَ ذَا نِعَمٍ دَعِ سُوءَ ظَنِّ زُرِّ شَرِيفٍ لِّلْكَرِيمِ

Disebut *al-Syamsiyah*, sebab seperti lamnya *lafadz* *الشَّمْسُ* wajibnya dibaca *Idgham* (*al* yang masuk/melebur pada huruf sesudahnya).

2.4 Transliterasi

Pada aplikasi ini, pedoman transliterasi yang digunakan adalah Sistem Transliterasi Arab-Latin Berdasarkan SKB Menteri Agama dan Menteri P&K RI No. 158/1987 dan No. 0543 b/U/1987 tertanggal 22 Januari 1988. Transliterasi huruf arab terdapat pada Tabel 2.1.

Tabel 2.1. Transliterasi

Huruf Arab	Nama	Huruf Latin	Keterangan
ا	Alif	-	tidak dilambangkan
ب	bā	b	-

ت	tā	T	-
ث	śā	s	s (dengan titik di atasnya)
ج	Jīm	j	-
ح	hā	h	(dengan titik di bawahnya)
خ	khā	kh	-
د	Dal	d	-
ذ	Ẓal	z	z (dengan titik di atasnya)
ر	rā	r	-
ز	Zai	z	-
س	Sīn	s	-
ش	Syīn	sy	-
ص	Ṣād	ṣ	s (dengan titik di bawahnya)
ض	Dād	d	d (dengan titik di bawahnya)
ط	ṭā	t	t (dengan titik di bawahnya)
ظ	zā	z	z (dengan titik di bawahnya)
ع	‘ain	‘	koma terbalik (di atas)
غ	Gain	g	-
ف	fā	f	-
ق	Qāf	q	-
ك	Kāf	k	-
ل	lām	l	-
م	mīm	m	-
ن	nūn	n	-
و	wāwu	w	-
هـ	Ha’	H	-
ء	hamzah	’	apostrof, tetapi lambang ini tidak dipergunakan untuk hamzah di awal kata

ي	Ya'	y	-
---	-----	---	---

2.5 Platform Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia (Pratama, 2011).

Android sendiri memiliki beberapa versi yang selalu berkembang sesuai dengan fitur-fitur baru yang ditambahkan pada telepon selular, mulai dari Android versi 1.0, Android versi 1.1, Android versi 1.5 (*Cupcake*), Android versi 1.6 (*Donut*), Android versi 2.0/2.1 (*Eclair*), Android versi 2.2 (*Froyo*), Android versi 2.3/2.3.7 (*Gingerbread*), Android versi 4.0/4.0.4 (*Ice Cream Sandwich*), Android versi 4.1 (*Jelly Bean*). dan yang paling baru saat ini adalah Android versi 4.3 (*Kit Kat*).

Platform Android diciptakan dibawah lisensi *open source*, dimana para pengembang bebas untuk mengembangkan aplikasi pada *platform* ini. Android menggunakan Linux kernel 2.6. Android adalah *platform mobile* yang tidak memiliki batasan dalam mengembangkan aplikasinya. Tidak ada lisensi dalam mengembangkan aplikasi Android. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun (Pratama, 2011).

Android juga memiliki tampilan yang menarik ditambah fitur notifikasi yang memudahkan pengguna jika ada pemberitahuan masuk, baik itu *sms*, panggilan, *email* dan sebagainya. Android juga didukung dengan *device* yang semakin canggih mengikuti perkembangan zaman saat ini. Disamping itu Android juga memungkinkan pemrosesan *multitasking* yang membuat Android mampu menjalankan beberapa aplikasi sekaligus dalam satu waktu.

Hanya saja Android memiliki beberapa kelemahan dalam hal resource. Android didesain untuk *device mobile* yang mampu dibawa kesana kemari dengan pengoperasian yang cepat, namun dengan ukuran *device* yang kecil maka kemampuan penyimpanan dan pemrosesan Android menjadi terbatas. *Device* Android memiliki kemampuan penyimpanan data yang terbatas pada memori internal dan eksternal. Rata-rata Android memiliki Ram sebesar 250 MB sampai 1 GB dengan kecepatan prosesor mencapai 1 GHz.

BAB III ANALISA DAN PERANCANGAN

3.1 Perancangan dan Pengumpulan Data

Perancangan dan pengumpulan data dalam penelitian ini merupakan pengambilan data sampel yang nantinya akan dijadikan data *template*. Data yang dikumpulkan berupa *image* berukuran 20x20 yang berisi *image* huruf maupun harakat. *Image* ini didapatkan dari *capture* kamera *device mobile* terhadap al-Quran terbitan j-Art, kemudian dilakukan *cropping* perhuruf maupun perharakat.

Selanjutnya dilakukan lagi *cropping* berdasarkan nilai piksel biner hitam terluar untuk menghilangkan ruang kosong dipinggir *image*. Kemudian dilakukan *scaling* menjadi 20x20 piksel untuk memeperingan dan mempermudah pencocokan.

Image berukuran 20x20 piksel tersebut selanjutnya dijadikan *array grayscale* dengan cara mengambil nilai rata-rata dari nilai merah, hijau dan biru dari tiap pikselnya. kemudian *array grayscale* tersebut disimpan sebagai *array template* di dalam aplikasi. Contoh *image* 20x20 terdapat pada Gambar 3.1.



Gambar 3.1. *Image* 20x20

Jumlah dari semua *image* 20x20 ada 268 buah, sehingga terdapat 268 *array* yang disimpan dalam *array* 3 dimensi. Diambil sampel sebanyak ini karena keanekaragaman dari tiap jenis huruf dan harakat dalam al-Quran.

3.2 Desain Sistem Aplikasi

Penelitian ini dilakukan menggunakan sistem operasi Android pada perangkat *mobile*. Digunakan sistem operasi Android karena

memiliki banyak pengguna dan mudah digunakan. Selanjutnya penelitian diimplementasikan pada perangkat *mobile* untuk memudahkan penggunaan aplikasi oleh pengguna. Ketika pengguna ingin memeriksa hukum tajwid dalam al-Quran, maka pengguna tinggal membuka aplikasi pada perangkat *mobile*.

Kemudian, pengguna melakukan *capture* susunan huruf dalam al-Quran yang akan diperiksa hukum tajwidnya. Aplikasi akan memberikan opsi pemotongan gambar dan kemudian menampilkan hasil berupa nama huruf, hukum tajwid sampai cara baca dari susunan huruf dan harakat tersebut.

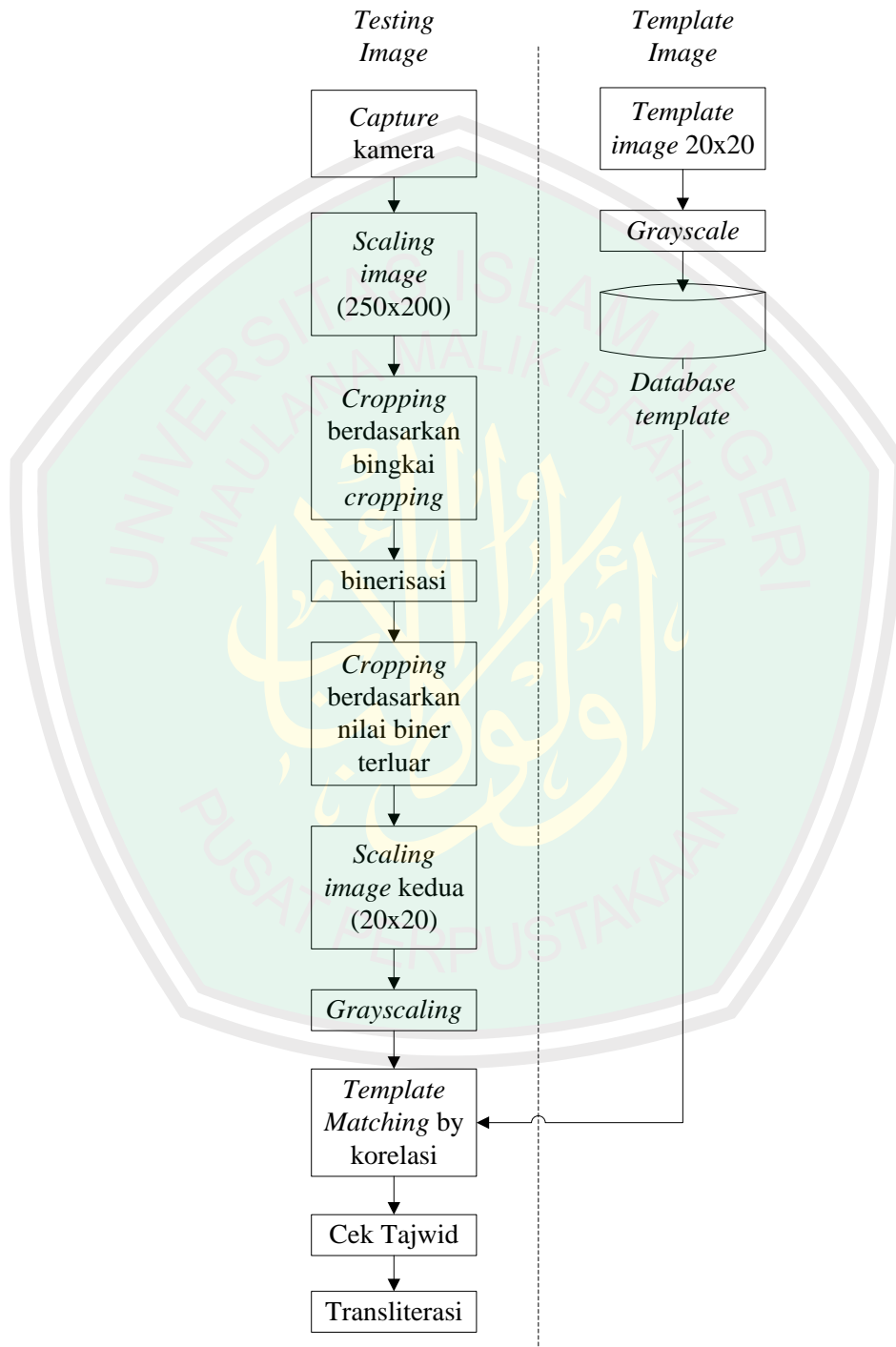
Adapun desain aplikasi tersebut dimulai dari *input* yang didapat dari *capture view* kamera perangkat *mobile*. Hasil dari *capture* ini kemudian disesuaikan ukurannya menggunakan *scaling* dan disimpan dalam format *portable network graphics* (PNG).

Selanjutnya dilakukan *cropping* pada *image* PNG dengan memanfaatkan bingkai *crop* pada aplikasi. *Cropping* dilakukan untuk memisah tiap huruf dan harakat, sehingga didapat individu huruf dan harakat. Setelah diperoleh masing-masing huruf dan harakat, maka akan dilakukan *scaling* lagi untuk menyesuaikan dengan ukuran *template*.

Selanjutnya masing-masing huruf dan harakat hasil dari *cropping* dicocokkan dengan tiap *template* huruf dan harakat dalam aplikasi. Pencocokan dilakukan dengan menggunakan metode *template matching* dengan korelasi pearson. Masing-masing huruf akan dicocokkan dengan tiap *template* huruf sedangkan masing-masing harakat akan dicocokkan dengan tiap *template* harakat.

Setelah masing-masing huruf dan harakat dicocokkan dengan tiap *template* huruf dan harakat, kemudian dilakukan pengecekan hukum tajwid dan cara baca.

Adapun tahap terakhir adalah aplikasi menampilkan *output* berupa nama dari masing-masing huruf dan harakat beserta hukum tajwid dan cara bacanya. Desain sistem aplikasi terdapat pada Gambar 3.2.

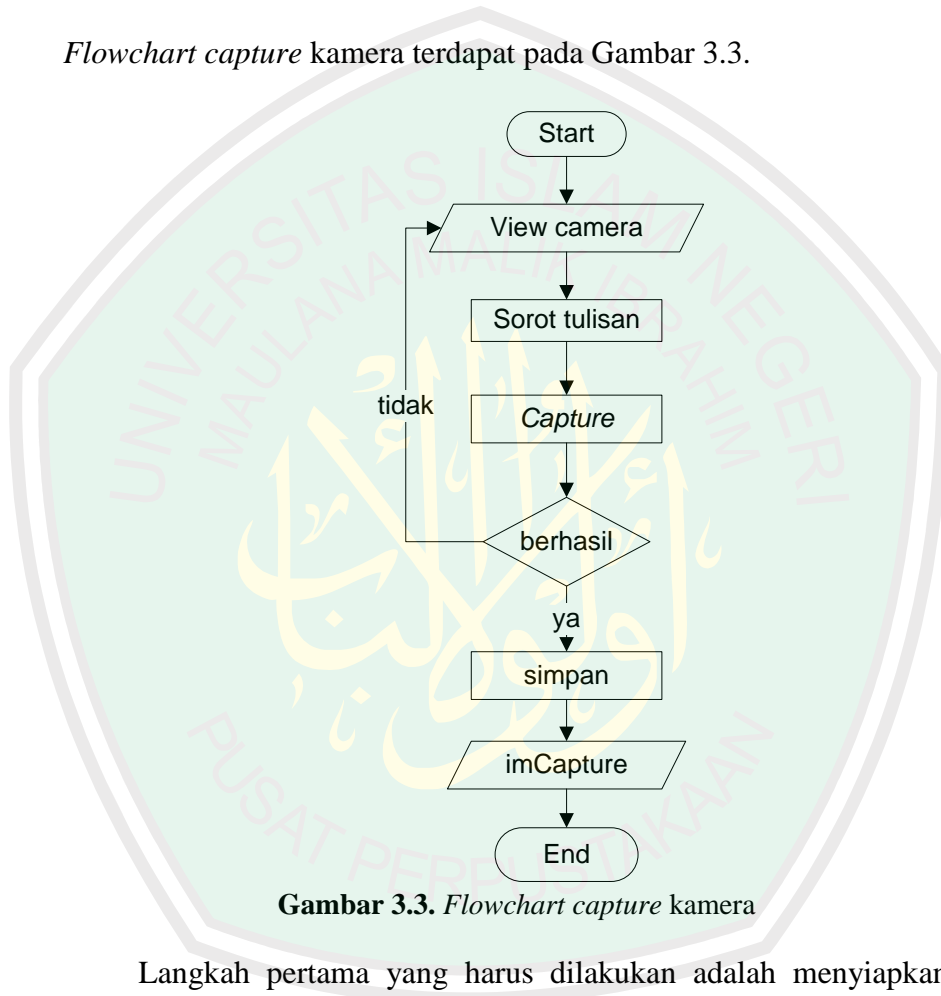


Gambar 3.2. Desain Sistem Aplikasi

Berikut penjelasan dari tiap proses pada desain sistem :

3.3.1 Capture Kamera

Input utama dari aplikasi ini adalah hasil *capture view* dari tampilan *view* kamera yang terdapat pada *layout* Android. Sehingga aplikasi ini mutlak harus mampu menampilkan *view* kamera pada *layout* Android. *Flowchart capture* kamera terdapat pada Gambar 3.3.



Gambar 3.3. *Flowchart capture* kamera

Langkah pertama yang harus dilakukan adalah menyiapkan *layout* sebagai tempat untuk menampilkan *view* dari kamera. Selanjutnya mengakses dan membuka kamera *device mobile*. Diperlukan *permission* pada *AndroidManifest* untuk dapat mengakses kamera pada *device mobile*.

Setelah kamera diakses, selanjutnya kamera dijalankan sehingga kamera mampu mendapatkan *input* berupa tampilan yang disorot. Hasil

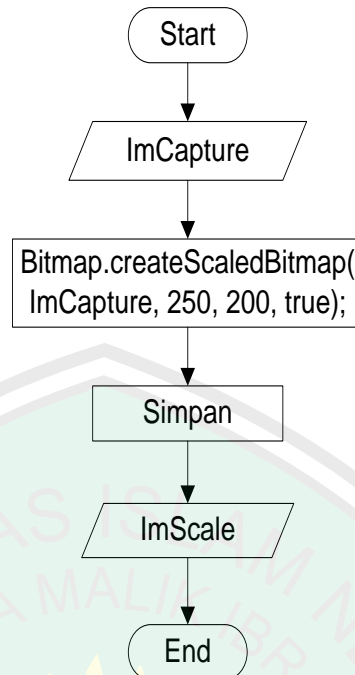
sorotan dari kamera tersebut selanjutnya ditampilkan pada *layout* yang sebelumnya telah disiapkan.

Setelah *view* kamera berhasil ditampilkan pada *layout Android*, selanjutnya *view* kamera tersebut *dicapture* dengan menekan tombol “Ambil” untuk selanjutnya disimpan dalam *image* berformat PNG. *Capture* hanya dilakukan pada tampilan yang muncul pada *layout*, yakni hasil dari sorotan kamera, bukan keseluruhan tampilan pada layar Android.

Setelah tombol “Ambil” ditekan, maka *capture* dilakukan, selanjutnya dilakukan pengecekan sebelum penyimpanan. Pengecekan ini dilakukan untuk memastikan keberhasilan *capture* yang dilakukan, jika berhasil maka hasilnya akan disimpan, sedangkan jika *capture* gagal maka penyimpanan tidak dapat dilakukan. Ketika *capture* berhasil, maka *image* akan disimpan dalam format PNG dan disimpan dalam folder yang disediakan aplikasi.

3.3.2 *Scaling Image (250x200)*

Selanjutnya dilakukan *scaling* untuk menyeragamkan ukuran *image* sehingga dapat dilakukan *cropping* pada proses selanjutnya. Jika tidak dilakukan *scaling*, maka pada kamera dengan resolusi berbeda akan mengalami kesulitan untuk melakukan *cropping*. Hal ini membuat *cropping* yang dilakukan mengacu pada ukuran *imageView* pada tampilan Android, bukan mengacu pada ukuran asli dari *image*. Oleh karena itu ukuran *image* dirubah menyesuaikan ukuran *imageView* pada tampilan Android. *Flowchart scaling image* terdapat pada Gambar 3.4.

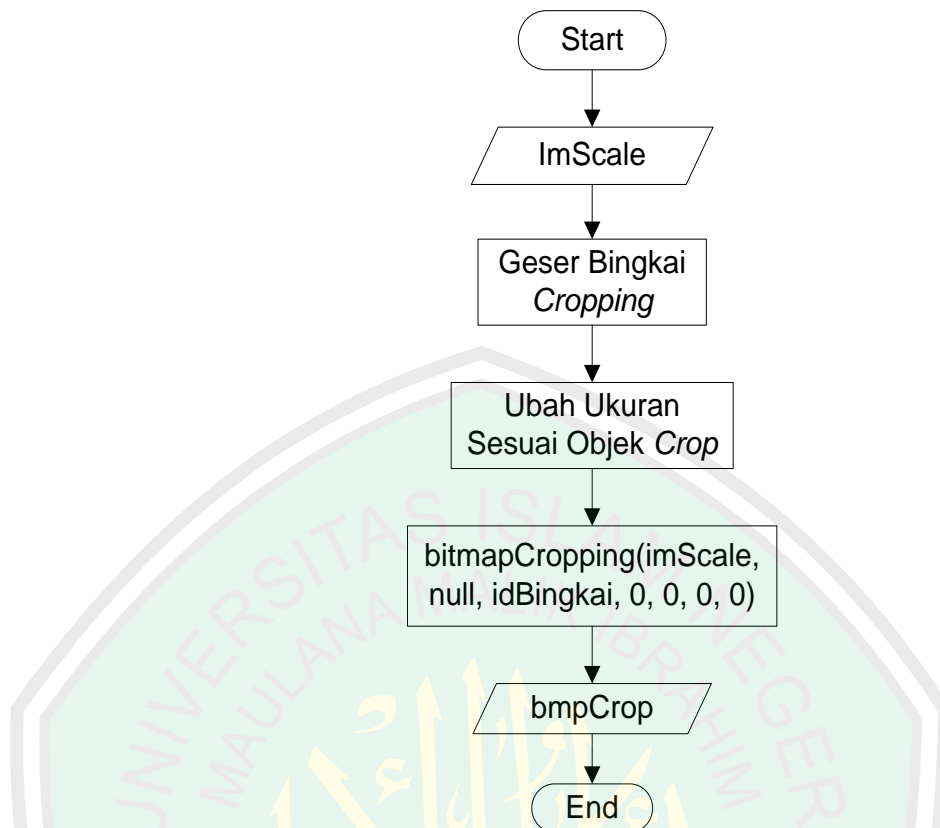


Gambar 3.4. Flowchart scaling image (250 x 200)

Untuk melakukan *scaling*, *image* yang telah *dicapture* dipanggil ke dalam *bitmap*. Selanjutnya dilakukan *scaling* pada *bitmap* tersebut berdasarkan ukuran *layout* tampilan Android, yakni 250 x 200 piksel. Kemudian hasil dari *scaling* disimpan dalam *image* berformat PNG dalam folder yang disediakan aplikasi.

3.3.3 Cropping Berdasarkan Bingkai Cropping

Selanjutnya *image* yang telah melalui *scaling* dan disimpan, ditampilkan pada *imageView* untuk dilakukan *cropping* menggunakan bingkai *cropping* pada *image* tersebut. Flowchart *cropping* berdasarkan bingkai *cropping* dapat dilihat pada Gambar 3.5.



Gambar 3.5. Flowchart cropping berdasarkan bingkai cropping

Untuk menampilkan *image scaling* yang disimpan, sebelumnya *image scaling* dipanggil ke dalam *bitmap* terlebih dahulu. Selanjutnya *bitmap* dimasukkan ke dalam *imageView* dengan menggunakan *method setImageBitmap()*. Hasilnya *image scaling* akan tampil pada *imageView*.

Image scaling yang ditampilkan harus memiliki ukuran yang sama dengan ukuran *imageView*. Hal ini dilakukan supaya *cropping* dapat dilakukan dengan menggunakan bingkai *cropping* yang berada di atas *imageView*. Bingkai *cropping* tersebut digunakan sebagai alat bantu untuk melakukan *cropping* pada *image scaling*.

Bingkai *cropping* akan digerakkan menuju posisi bidang yang akan *dicropping* serta menyesuaikan ukuran bidang tersebut. Hasil dari *cropping* ini adalah *crop* dari *image scaling* yang bersesuaian dengan bidang yang

dilingkupi bingkai *cropping*. Begitu juga ukuran dari hasil *cropping* juga sesuai ukuran bingkai *cropping*.

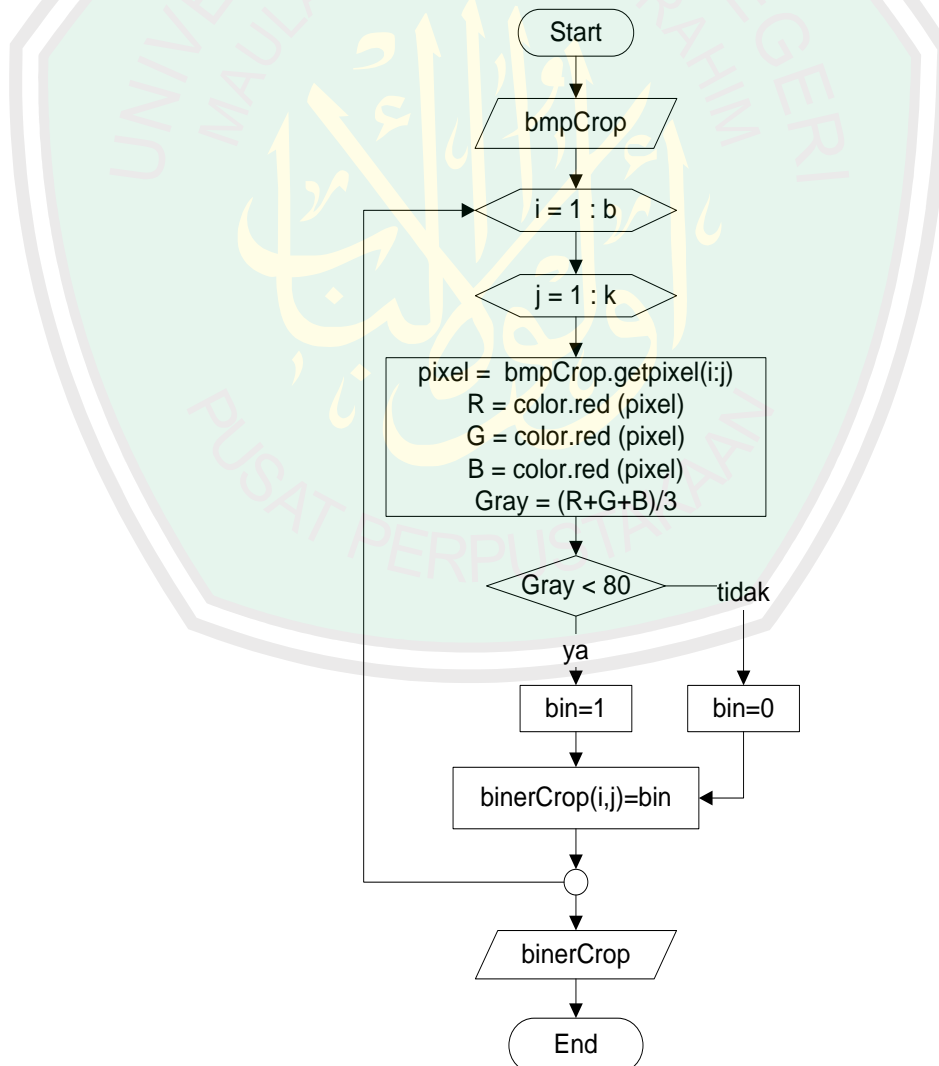
Bingkai *cropping* memiliki tipe *button* dengan jumlah 6 buah. Bingkai *cropping* memiliki bentuk persegi dengan warna transparan dan hanya memiliki *border* berwarna putih. Bingkai *cropping* dapat digeser dengan memanfaatkan *method onDragListener()*. Sedangkan untuk merubah ukuran bingkai *cropping*, digunakan tombol navigasi yang berisi *method setLayoutParams()*, baik untuk memperbesar atau memperkecil ukuran sisi datar maupun sisi samping dari bingkai *cropping*. Bingkai *cropping* berjumlah 6 buah karena 3 buah digunakan untuk melakukan *crop* terhadap harakat sedangkan 3 buah lainnya digunakan untuk melakukan *crop* terhadap huruf.

Bingkai *cropping* dibuat dapat digeser maupun dirubah ukurannya agar dapat melakukan *cropping* sesuai kebutuhan pada *image scaling* yang berada dalam *imageView*. *Cropping* dilakukan berdasarkan posisi x, y dari bingkai *cropping* terhadap *imageView*, serta panjang dan lebar dari bingkai *cropping* tersebut.

Posisi x dan y digunakan sebagai acuan untuk posisi awal dari *cropping*. Sedangkan panjang dan lebar bingkai *cropping* digunakan sebagai acuan panjang dan lebar dari bidang yang akan *dicropping*. Hasil dari *cropping* ini adalah *bitmap* berisi *image* yang merupakan *cropping* dari *image scaling* sesuai bingkai *cropping*. Hasil *cropping* ini disebut hasil *cropping* pertama.

3.3.4 Binerisasi

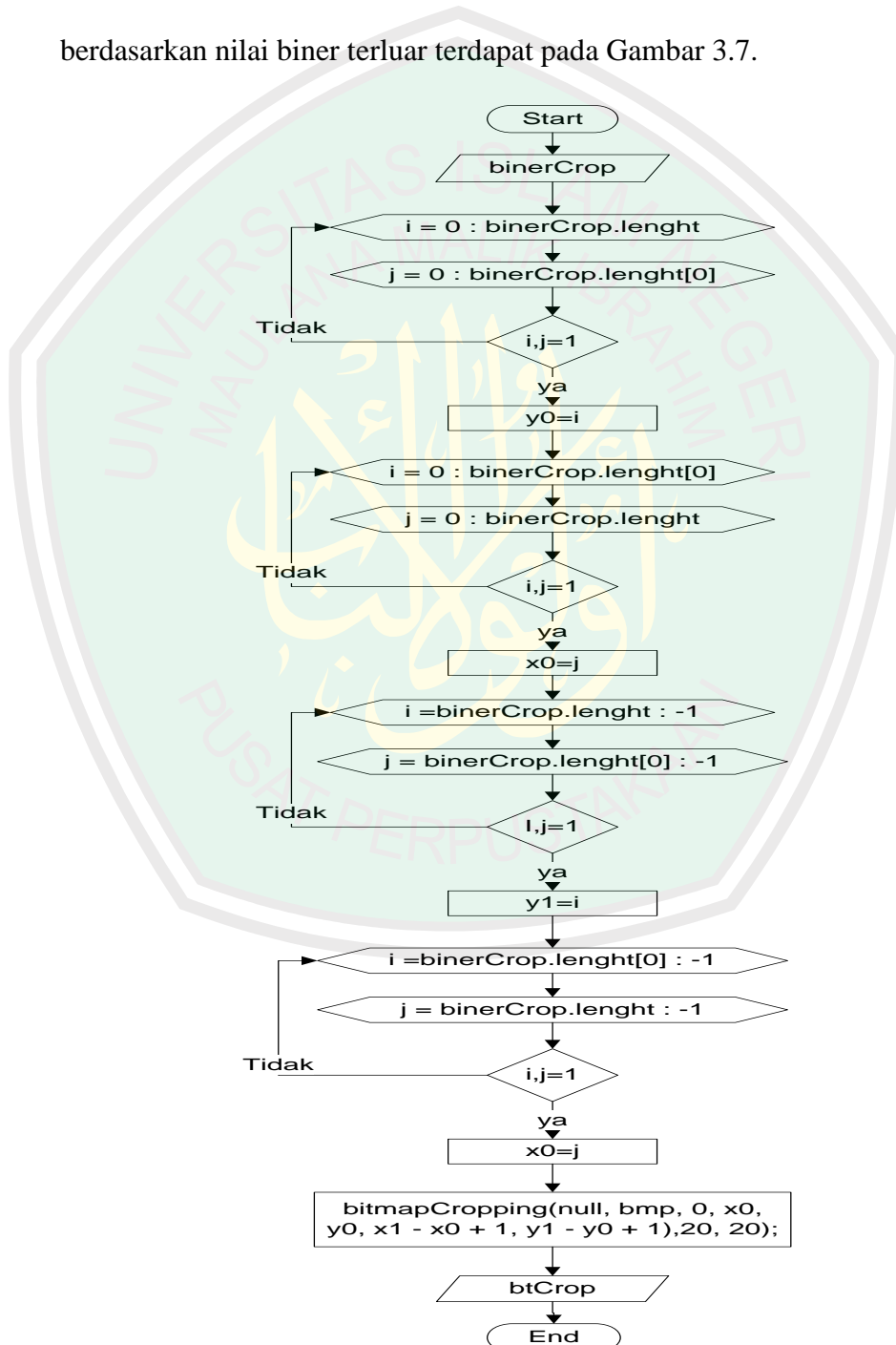
Selanjutnya dilakukan binerisasi pada *bitmap* hasil *cropping* pertama untuk mendapatkan *array* biner. Pada *array* biner ini, nilai biner 1 adalah nilai untuk biner hitam sedangkan nilai biner 0 adalah nilai untuk biner putih. Nilai biner tiap piksel diambil dari rata-rata nilai merah, hijau dan biru (nilai *grayscale*) dari tiap piksel dengan *threshold* bernilai 80. Selanjutnya setelah didapatkan *array* biner, maka dilakukan *cropping* berdasarkan nilai piksel biner hitam yang terluar. *Flowchart* binerisasi terdapat pada Gambar 3.6.



Gambar 3.6. Flowchart binerisasi

3.3.5 Cropping Berdasarkan Nilai Biner Terluar

Selanjutnya dilakukan *cropping* lagi pada *bitmap* hasil *cropping* pertama untuk mendapatkan nilai piksel biner hitam terluar, sehingga tidak ada ruang kosong pada pinggiran *image cropping*. Nilai piksel biner hitam terluar didapatkan dari *array* hasil binerisasi. *Flowchart* dari *cropping* berdasarkan nilai biner terluar terdapat pada Gambar 3.7.



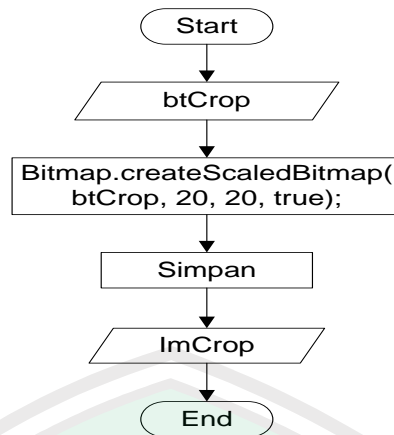
Gambar 3.7. Flowchart *cropping* berdasarkan nilai biner terluar

Pengecekan itu dilakukan dengan cara memeriksa nilai piksel biner dengan nilai 1 pada posisi paling atas (y_0), paling bawah (y_1), paling kiri (x_0) dan paling kanan (x_1). Setelah mendapatkan semua nilai piksel biner yang dibutuhkan, selanjutnya melakukan *cropping* lagi dengan *bitmap* berdasarkan nilai piksel biner tersebut.

Nilai biner pixel x_0 dan y_0 menjadi acuan awal dari posisi *cropping*. Sedangkan nilai x_1 dikurangi x_0 ditambah 1 ($(x_1 - x_0) + 1$) dan nilai y_1 dikurangi y_0 ditambah 1 ($(y_1 - y_0) + 1$) menjadi acuan lebar dan panjang dari *cropping*. Masing-masing $(x_1 - x_0) + 1$ dan $(y_1 - y_0) + 1$ memiliki penambahan 1 karena nilai panjang dan lebar dimulai dari nilai 1 sedangkan x dan y dimulai dari nilai 0 (karena merupakan nilai indeks). Hasil dari *cropping* ini adalah *bitmap* berisi *image* yang merupakan *cropping* dari hasil *cropping* pertama sesuai nilai piksel biner terluar. Hasil *cropping* ini disebut hasil *cropping* kedua.

3.3.6 *Scaling Image* Kedua (20x20)

Kemudian dilakukan lagi *scaling* yang kedua untuk menyesuaikan ukuran hasil *cropping* kedua dengan *template*. Di sini dilakukan *scaling* lagi menjadi ukuran 20x20 piksel untuk memudahkan dan meringankan pencocokan namun tetap mempertahankan akurasi. *Flowchart Scaling image* kedua terdapat pada Gambar 3.8.



Gambar 3.8. Flowchart scaling image kedua (20 x 20)

Dalam *scaling* kedua ini, hasil *cropping* kedua dipanggil ke dalam *bitmap*. Selanjutnya dilakukan *scaling* pada *bitmap* tersebut menjadi 20x20 piksel. Kemudian hasil dari *scaling* disimpan dalam *image* berformat PNG dalam folder yang disediakan aplikasi.

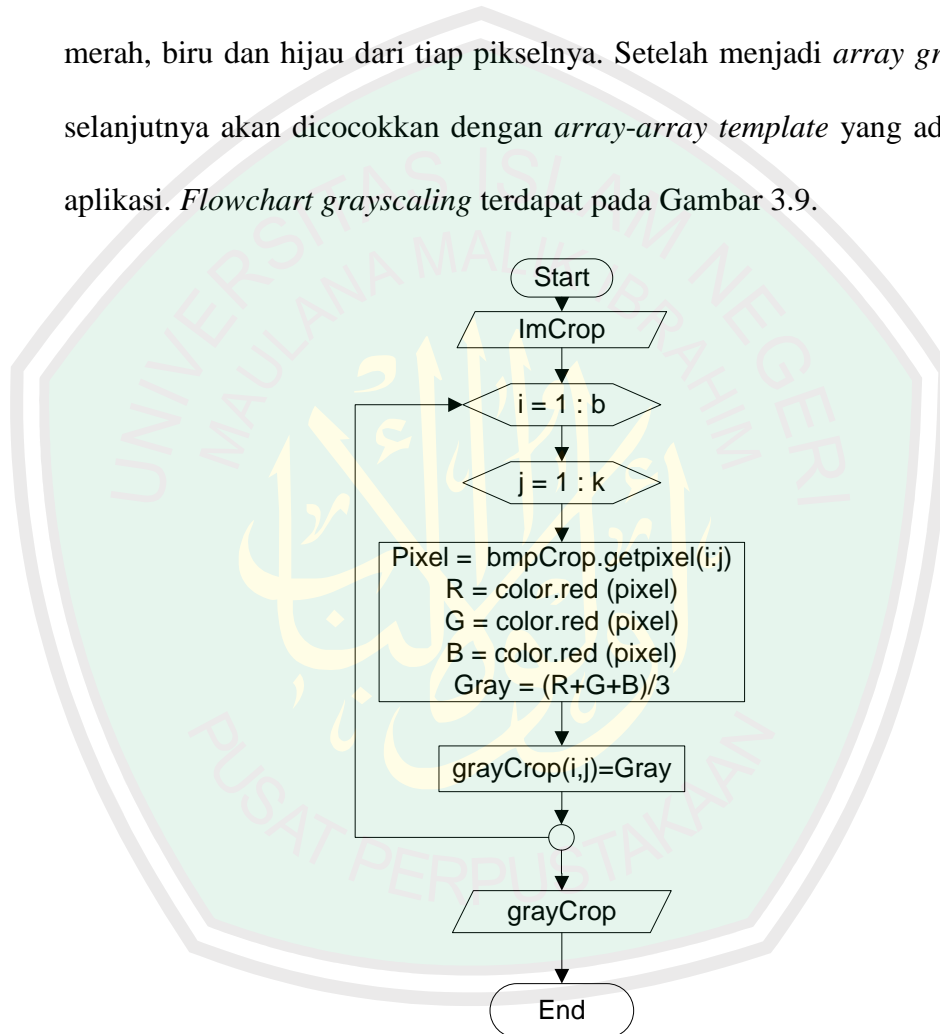
Perlu diketahui bahwa dalam bagian c sampai f ini terdapat 6 *image* hasil *cropping* yang menjalani proses berupa *cropping* berdasarkan bingkai *cropping*, binerisasi, *cropping* berdasarkan nilai piksel biner hitam terluar dan *scaling* menjadi 20x20 piksel. Hal ini membuat *output* pada proses ini menjadi 6 *image* hasil *scaling* kedua yang berukuran 20x20 piksel.

Selanjutnya 6 *image* hasil *scaling* kedua yang berukuran 20x20 piksel ditampilkan pada 6 *imageView*, 3 *imageView* untuk 3 harakat dan 3 *imageView* untuk 3 huruf. Sebagaimana cara *image scaling* yang disimpan pada tahap 2 bagian a, masing-masing 6 *image* hasil *scaling* kedua dipanggil ke dalam *bitmap* terlebih dahulu.

Kemudian masing-masing *bitmap* tersebut dimasukkan ke dalam *imageView* yang sesuai (harakat pada bagian harakat dan huruf pada bagian huruf) dengan menggunakan *method setImageBitmap()*. Hasilnya masing-masing *image* hasil *scaling* kedua akan tampil sesuai dengan *imageView*nya.

3.3.7 Grayscale

Selanjutnya dilakukan *grayscale* karena *template* dalam aplikasi sudah berupa *array grayscale* 3 dimensi. Untuk untuk menyesuaikan *template* tersebut maka semua *image* hasil *scaling* kedua dirubah terlebih dahulu menjadi *array grayscale* dengan cara mengambil rata-rata warna merah, biru dan hijau dari tiap pikselnya. Setelah menjadi *array grayscale*, selanjutnya akan dicocokkan dengan *array-array template* yang ada dalam aplikasi. *Flowchart grayscale* terdapat pada Gambar 3.9.

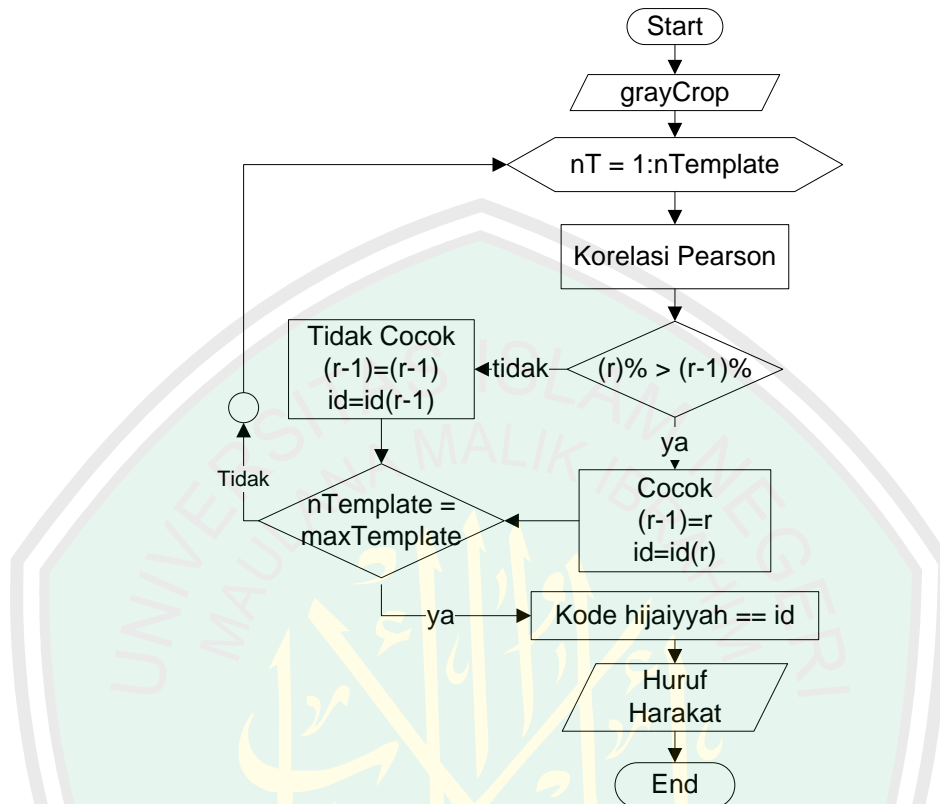


Gambar 3.9. Flowchart grayscale

3.3.8 Template Matching By Korelasi

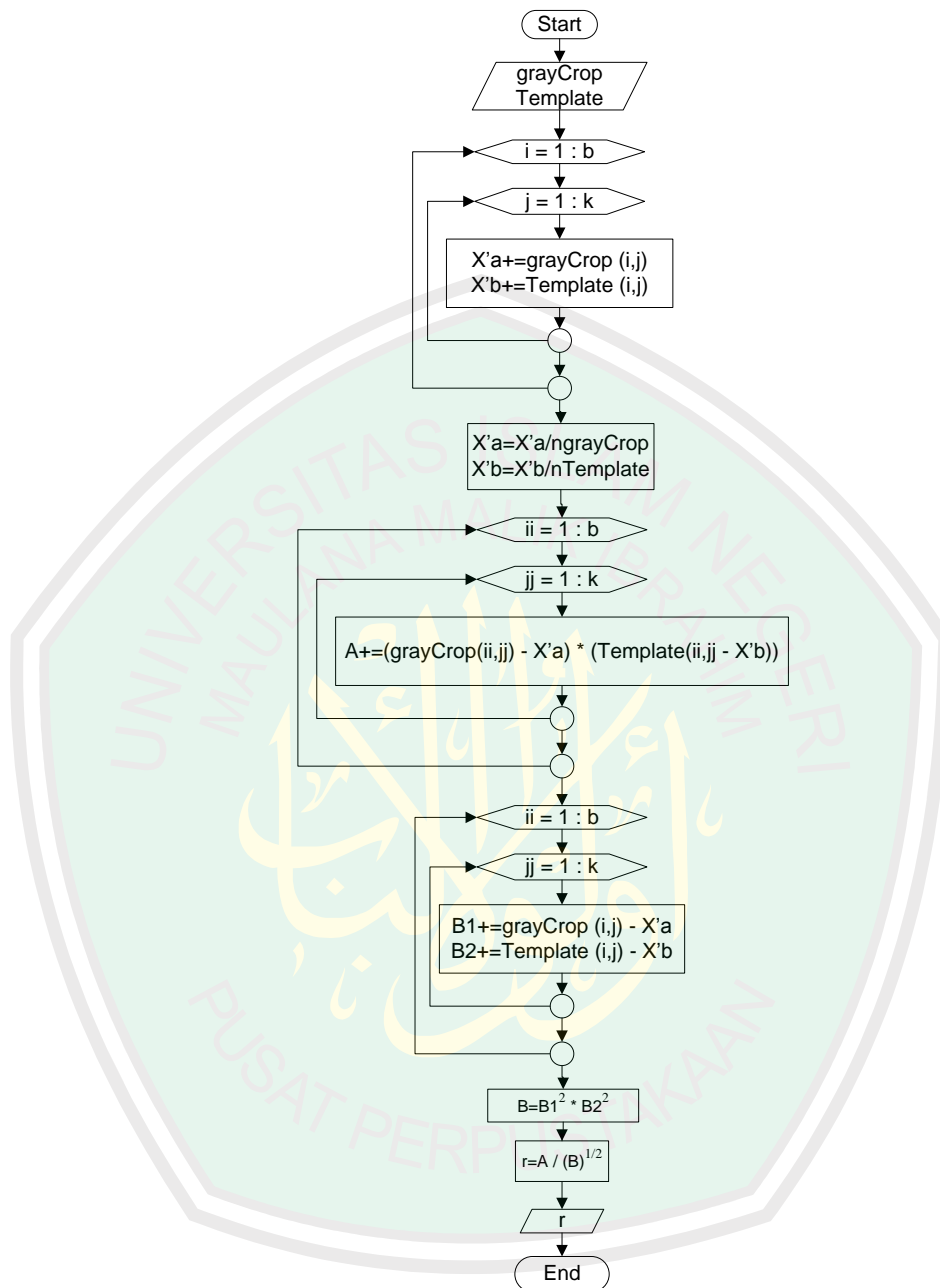
Selanjutnya dilakukan pencocokan antara *array grayscale* dari *image* hasil *scaling* kedua dengan *template* dalam aplikasi menggunakan metode *template matching* dengan korelasi. *Template* yang digunakan aplikasi ini sudah berupa *array grayscale* 3 dimensi bertipe objek sedangkan

korelasinya menggunakan korelasi pearson. *Flowchart template matching by korelasi* terdapat pada Gambar 3.10.



Gambar 3.10. *Flowchart template matching by korelasi*

Pada aplikasi ini terdapat 3 jenis *array template* yakni *array template* huruf, *array template* harakat dan *array template* harakat kasrah. *Array grayscale* dari *image* huruf akan dicocokkan dengan *array template* huruf, sedangkan *array grayscale* dari *image* harakat akan dicocokkan dengan *array template* harakat atau *array template* harakat kasrah. *Array grayscale* dari *image* harakat akan dicocokkan dengan *array template* harakat ketika posisi y dari harakat tersebut lebih kecil daripada posisi y huruf ditambah $\frac{1}{2}$ dari tinggi huruf, sedangkan jika lebih besar maka akan dicocokkan dengan *array template* harakat kasrah. Pencocokan tersebut dapat dilihat pada *flowchart* korelasi Pearson pada Gambar 3.11.



Gambar 3.11. Flowchart korelasi Pearson

Selanjutnya dalam melakukan pencocokan, kedua *array* (*array grayscale* hasil *scaling* kedua dan *array template*) diambil nilai rata-rata masing-masing. Kemudian nilai rata-rata masing-masing digunakan untuk mengurangi nilai tiap piksel masing-masing *array*. Selanjutnya masing-

masing piksel dari *array grayscale* hasil *scaling* kedua dikalikan dengan masing-masing piksel *array template* yang posisinya sama contoh piksel posisi 0,0 *array grayscale* hasil *scaling* kedua dikali dengan posisi 0,0 *array template*, posisi 0,1 *array grayscale* hasil *scaling* kedua dikali dengan posisi 0,1 *array template* dan seterusnya sehingga dihasilkan *array* baru hasil perkalian kedua *array* tersebut. Selanjutnya, diambil nilai penjumlahan dari semua piksel *array* baru tersebut dan nilai ini disimpan dahulu.

Kemudian masing-masing *array* yang tadi telah dikurangi nilai rata-ratanya, dikuadratkan tiap pikselnya dan diambil nilai penjumlahan dari tiap pikselnya tersebut. Kemudian hasil penjumlahan dari kedua *array* yang dikuadratkan tersebut dikalikan satu sama lain dan hasilnya diakar.

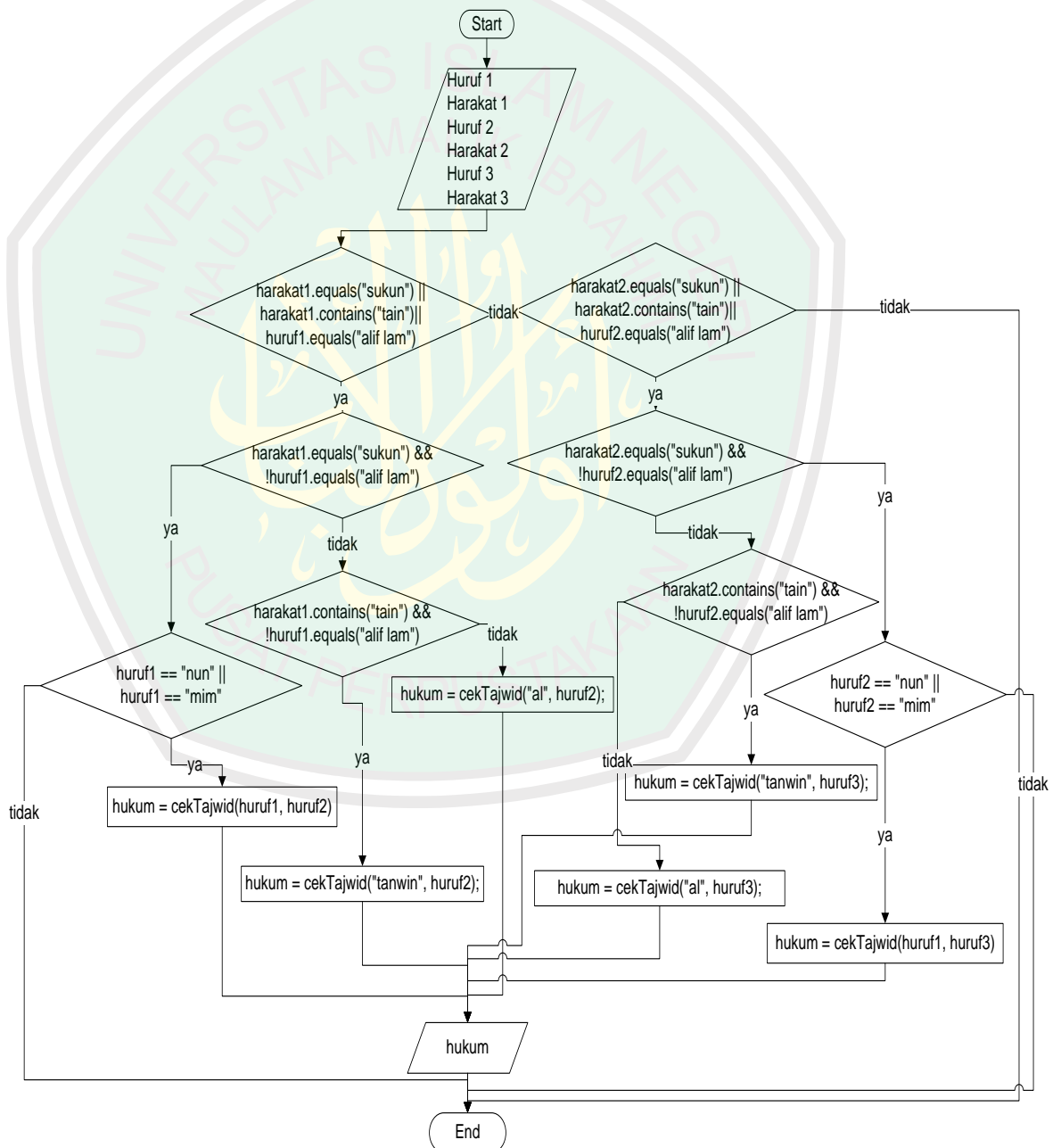
Setelah itu, untuk mendapatkan nilai korelasi dari kedua *array* maka nilai penjumlahan yang tadi telah disimpan, dibagi dengan akar dari *array* yang dikuadratkan. Hasilnya akan berupa desimal, sedangkan untuk menjadikannya satuan persen maka dikali 100%.

Huruf maupun harakat dianggap cocok ketika persentasenya lebih dari 90% atau persentasenya merupakan persentase tertinggi dibandingkan dengan pencocokan pada *array template* lainnya. Hasil dari pencocokan ini adalah nama dari 3 huruf dan 3 harakat yang dicocokkan.

Setelah semua nama huruf dan harakat ditemukan, selanjutnya nama huruf dan harakat tersebut ditampilkan pada *textView* masing-masing. Caranya dengan menggunakan *method setText()* pada masing-masing *textView*. Hasilnya berupa nama huruf maupun harakat ditampilkan pada *textView* yang dapat dilihat pada halaman hasil pada aplikasi.

3.3.9 Cek Tajwid

Setelah setiap nama huruf dan harakat ditemukan, selanjutnya dilakukan pengecekan tajwid. Pada aplikasi ini pengecekan tajwid menggunakan metode *rule base* tajwid berdasarkan kitab *Tuhfah at-Tullab*. Tajwid yang dapat dicek berupa hukum *nun* mati dan tanwin, hukum *mim* mati dan hukum *al-ta'rif*. Flowchart cek tajwid terdapat pada Gambar 3.12.



Gambar 3.12. Flowchart cek tajwid

Pengecekan tajwid dilakukan dengan cara memasukkan semua huruf dan harakat yang ditemukan ke dalam *method prosesTajwid()*. Dalam *method* ini, langkah pertama yakni melakukan pengecekan harakat. Dimulai dari harakat pertama, dan dilanjutkan harakat kedua.

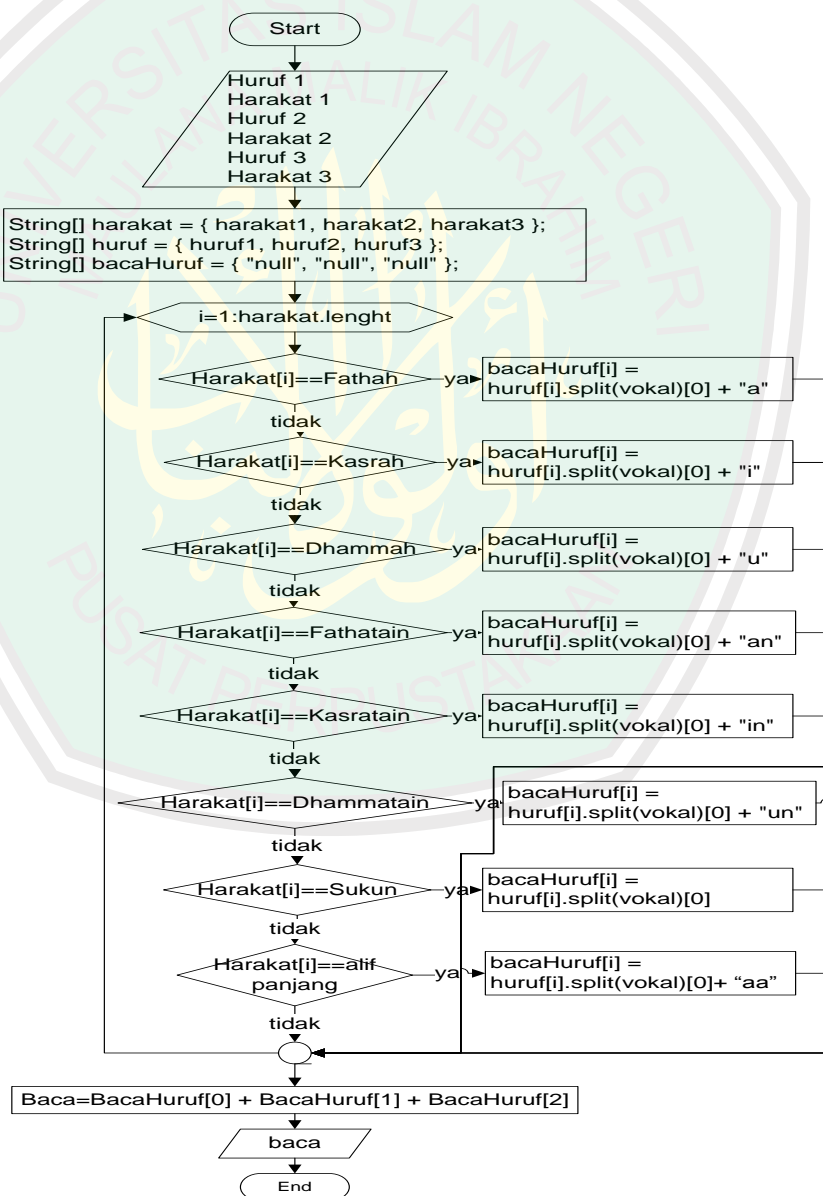
Pengecekan yang dilakukan terhadap harakat pertama yakni apakah berisi (*equals*) kata “*sukun*” atau mengandung (*contains*) kata “*tain*”. Jika harakat pertama *equals* “*sukun*”, maka dicek lagi apakah huruf pertama adalah mim atau nun. Jika termasuk salah satu huruf tersebut maka yang dicek tajwidnya adalah antara huruf pertama dan huruf kedua. Sedangkan jika harakat pertama *contains* “*tain*”, maka harakat pertama dirubah menjadi “*tanwin*” dan yang dicek tajwidnya adalah harakat pertama dan huruf kedua.

Jika harakat pertama tidak *equals* kata “*sukun*” atau *contains* kata “*tain*”, maka dilakukan pengecekan terhadap harakat kedua dengan cara yang sama dengan harakat pertama. Jika harakat kedua *equals* “*sukun*”, maka dicek lagi apakah huruf kedua adalah *mim* atau *nun*. Jika termasuk salah satu huruf tersebut maka yang dicek tajwidnya adalah antara huruf kedua dan huruf ketiga. Sedangkan jika harakat kedua *contains* “*tain*”, maka harakat kedua dirubah menjadi “*tanwin*” dan yang dicek tajwidnya adalah harakat kedua dan huruf ketiga.

Setelah hukum tajwid ditemukan, selanjutnya hukum tajwid tersebut ditampilkan pada *textView*. Caranya dengan menggunakan *method setText()* pada *textView*. Hasilnya berupa hukum tajwid ditampilkan pada *textView* yang dapat dilihat pada halaman hasil pada aplikasi.

3.3.10 Transliterasi

Selanjutnya dilakukan transliterasi dari kombinasi huruf dan harakat berdasarkan *rule base* transliterasi. Transliterasi dilakukan dengan menghubungkan antara harakat dengan huruf yang bersesuaian. Mula-mula harakat dihubungkan dengan huruf pada urutan yang sama, harakat pertama dengan huruf pertama, harakat kedua dengan huruf kedua dan harakat ketiga dengan huruf ketiga. *Flowchart* cek tajwid terdapat pada Gambar 3.13.



Gambar 3.13. *Flowchart* transliterasi

Harakat yang diperiksa berupa *fathah*, *kasrah*, *dhammah*, *fathatain*, *kasratain*, *dhammatain*, *sukun*, *alif panjang* dan *syaddah*. Setelah harakat selesai diperiksa, selanjutnya diperiksa hurufnya. Setiap huruf tadi kemudian diambil cara bacanya seperti *alif* yang *difathah* akan dibaca *a*, *ba* yang *difathah* akan dibaca *ba* dan seterusnya. Namun cara baca tersebut tidak berlaku khusus untuk huruf *alif lam*, pada huruf ini jika merupakan huruf pertama akan menjadi *al*, sedangkan jika bukan huruf pertama akan menjadi suara lam disukun yakni *l*.

Setelah semua harakat dan huruf diambil cara bacanya, maka ketiga huruf dan harakat tersebut dikombinasikan menjadi satu. Contohnya jika *ya fathah* (*ya*), *nun sukun* (*n*) dan *mim dhammah* (*mu*) digabung maka akan menjadi *yanmu*.

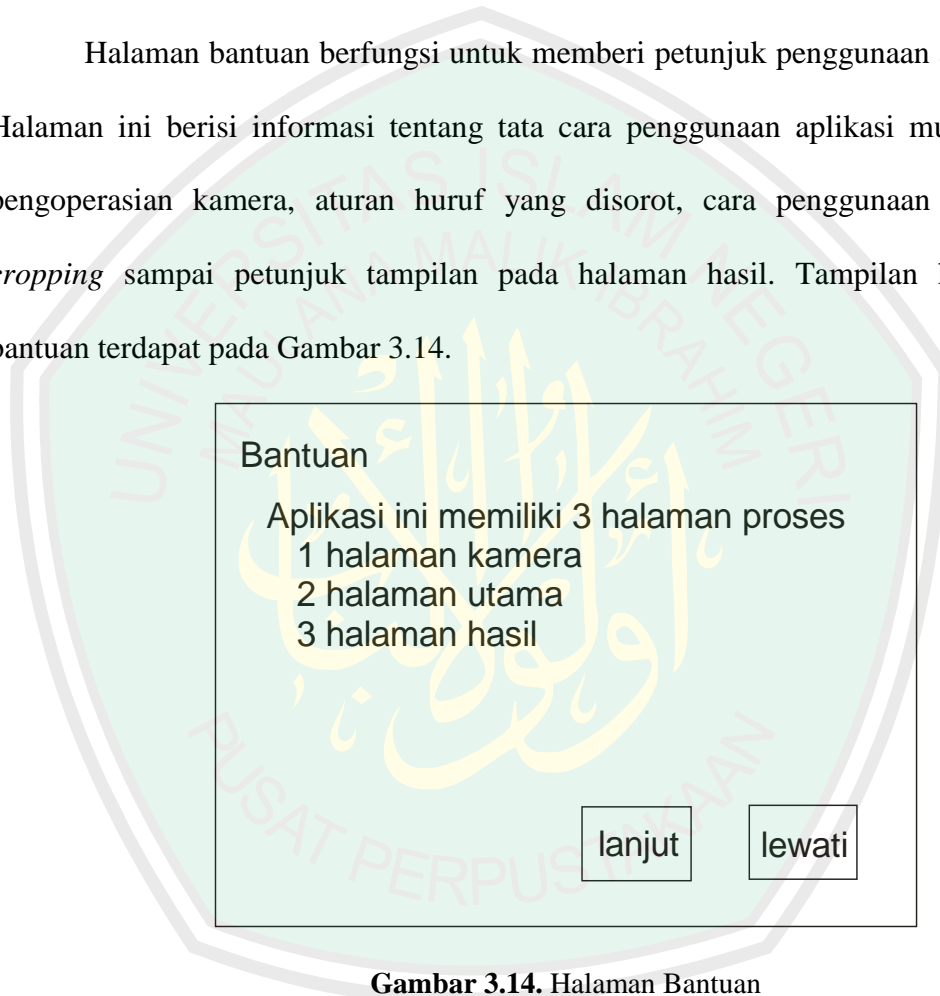
Kemudian jika terdapat hukum tajwid *idgham* maupun *iqlab* maka cara baca dari kombinasi tersebut juga ikut disesuaikan. Jika terdapat hukum *idgham bighunnah* maupun *idgham bilaa ghunnah* maka huruf nun melebur menjadi huruf sesudahnya, contoh *yanmu* menjadi *yammu*, *kunla* menjadi *kulla*. Sedangkan jika terdapat hukum *iqlab* maka nun sukun berubah menjadi mim sukun, contoh *minba* menjadi *mimba*. Dan jika terdapat hukum *idgham Syamsiah* maka bacaan alif lam akan berubah menjadi huruf sesudahnya, contoh *alnur* menjadi *annur*.

Setelah transliterisasi ditemukan, selanjutnya transliterisasi tersebut ditampilkan pada *textView*. Caranya dengan menggunakan *method setText()* pada *textView*. Hasilnya berupa transliterisasi ditampilkan pada *textView* yang dapat dilihat pada halaman hasil pada aplikasi.

3.3 Perancangan *Interface*

Untuk memudahkan *user* dalam penggunaan aplikasi ini, dibutuhkan *interface* yang bagus, *simple* dan mudah difahami. Pada aplikasi ini terdapat 4 *interface* utama yakni halaman bantuan, halaman kamera, halaman utama dan halaman hasil.

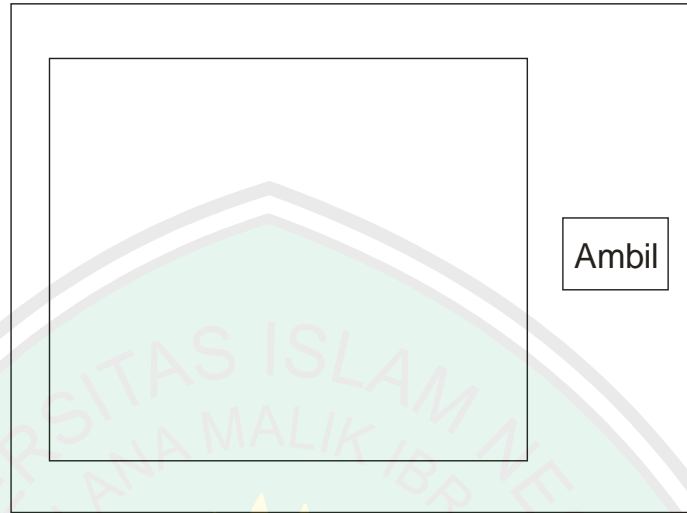
Halaman bantuan berfungsi untuk memberi petunjuk penggunaan aplikasi. Halaman ini berisi informasi tentang tata cara penggunaan aplikasi mulai dari pengoperasian kamera, aturan huruf yang disorot, cara penggunaan bingkai *cropping* sampai petunjuk tampilan pada halaman hasil. Tampilan halaman bantuan terdapat pada Gambar 3.14.



Gambar 3.14. Halaman Bantuan

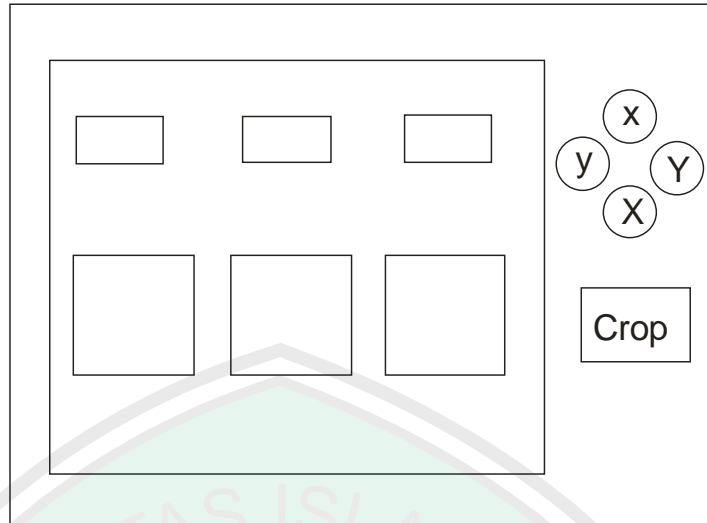
Halaman kamera berisi tampilan ketika kamera menyorot objek, di sini tampilan yang ada merupakan hasil sorotan kamera dengan *zoom* terbesar yang dimiliki oleh *device mobile*. Hal ini untuk memudahkan pengambilan *image* pada halaman al-Quran sehingga *user* tidak perlu untuk mendekatkan kamera terlalu dekat dengan al-Quran yang mengakibatkan *image* menjadi buram. Pada halaman ini selain dapat menampilkan hasil yang disorot juga terdapat tombol *capture*

untuk mengambil *image* yang disorot. Tampilan halaman kamera terdapat pada Gambar 3.15.



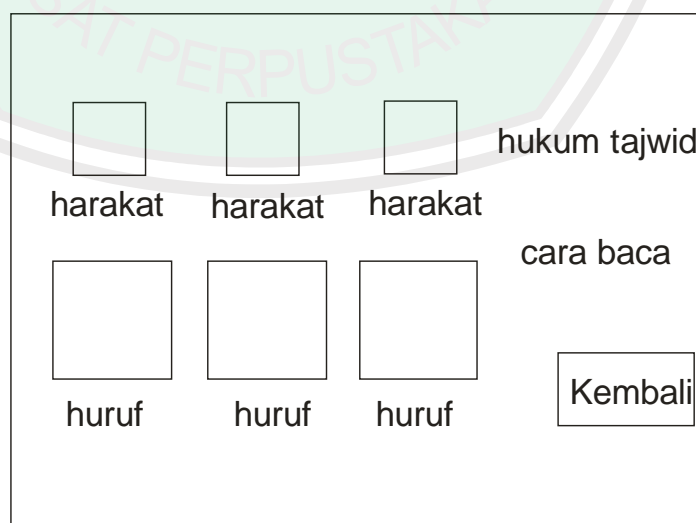
Gambar 3.15. Halaman Kamera

Halaman utama berisi hasil *capture* dari halaman kamera beserta fitur *cropping* yang dimiliki aplikasi. Fitur *cropping* ini berupa bingkai *cropping* yang berbentuk persegi yang mampu dipindah posisinya. Untuk membantu mengubah ukuran bingkai *cropping*, diberikan 4 tombol tambahan yang berfungsi untuk memperkecil panjang bingkai *cropping*, memperbesar panjang bingkai *cropping*, memperkecil lebar bingkai *cropping* dan memperbesar lebar bingkai *cropping*. Selain itu untuk mengeksekusi hasil *cropping*, terdapat tombol *cropping* yang akan mengeksekusi hasil *cropping* dan membuka halaman hasil. Tampilan halaman utama terdapat pada Gambar 3.16.



Gambar 3.16. Halaman Utama

Halaman hasil berfungsi untuk menampilkan hasil dari aplikasi berupa *image* hasil *cropping*, nama huruf dan harakat, hukum tajwid dan transliterasi. Halaman ini berisi 6 *imageView* dan 8 *textView*, *imageView* tersebut untuk menampilkan *image* hasil *cropping*, dan 8 *textView* untuk menampilkan 3 nama huruf dan 3 nama harakat, hukum tajwid dan transliterasi. Selanjutnya terdapat tombol kembali untuk menjalankan ulang aplikasi mulai dari awal lagi. Tampilan halaman hasil terdapat pada Gambar 3.17.



Gambar 3.17. Halaman Hasil

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Aplikasi ini diimplementasikan ke dalam sistem operasi *andorid*. Proses *capture* kamera dan *scaling image* (250 x 200) terdapat pada halaman kamera. Selanjutnya proses *cropping* berdasarkan bingkai *cropping*, binerisasi, *cropping* berdasarkan nilai biner terluar, dan *scaling image* kedua (20 x 20) terdapat pada halaman utama. Sedangkan proses *grayscaleing*, *template matching by* korelasi, cek tajwid, dan transliterasi terdapat pada halaman hasil.

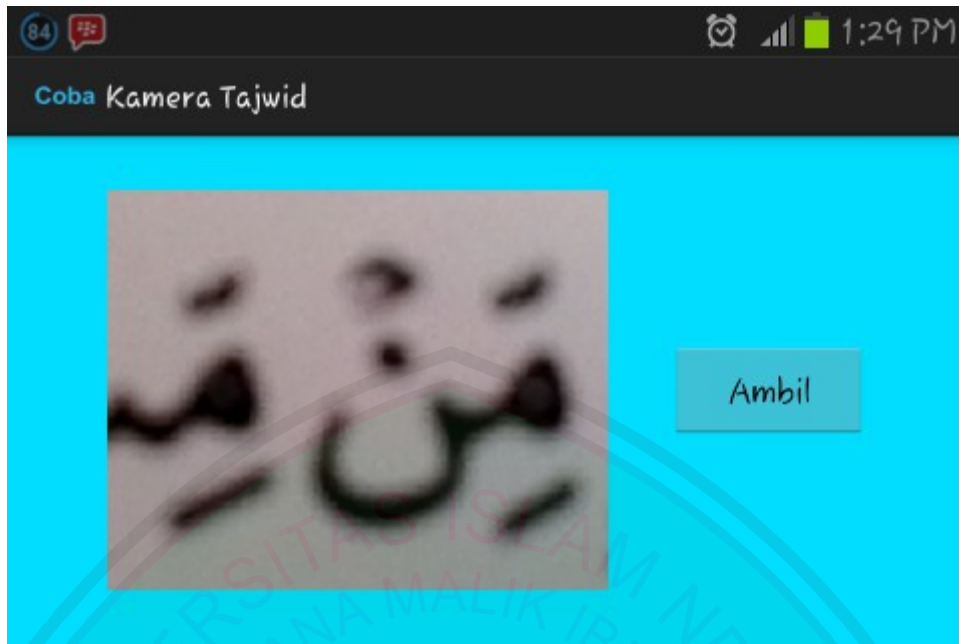
4.1.1. Halaman Kamera

Pada halaman kamera, terdapat fungsi untuk menampilkan hasil *view* kamera *mobile* pada *layout* yang terdapat pada halaman kamera. Fungsi tersebut terdapat dalam *source code* pada Gambar 4.1.

```
camera = Camera.open();  
cameraPreview = new Camera_Preview(this, camera);  
FrameLayout  
preview=(FrameLayout) findViewById(R.id.frame_layout);  
preview.addView(cameraPreview);
```

Gambar 4.1. *Source code* menampilkan *view* kamera

Pada *source code* menampilkan *view* kamera terdapat kelas *cameraPreview* yang berfungsi untuk menampilkan hasil *view* oleh kamera. Selanjutnya hasil *view* ini dimasukkan ke dalam *layout* yang bernama *preview*. Hasilnya *view* dari kamera akan muncul pada *layout* pada halaman kamera seperti terdapat pada Gambar 4.2.



Gambar 4.2. Halaman Kamera

Setelah didapat *view* yang cocok, maka selanjutnya dilakukan *capture* pada *view* tersebut menggunakan *method camera.takePicture()* yang merupakan *default method* dari Android. Setelah *capture* dilakukan, *image* hasil *capture* disimpan sebagai *imCapture.png* menggunakan *source code* pada Gambar 4.3.

```
FileOutputStream out = new FileOutputStream(file);
if (bitmap != null) {
    bitmap.compress(Bitmap.CompressFormat.PNG, 90, out);
    out.flush();
}
else if (data != null) {
    out.write(data);
}
else {
    toast("bitmap dan data kosong");
}
out.close();
```

Gambar 4.3. *Source code* simpan *bitmap*

Selanjutnya dilakukan proses *scaling* pertama. Dalam melakukan *scaling* diperlukan bantuan dari *bitmap* yang berfungsi untuk melakukan *scaling* terhadap *image* yang diproses. *Bitmap scaling* menggunakan *source code* pada Gambar 4.4.

```
bitmap = Bitmap.createScaledBitmap(BitmapFactory.decodeStream(new FileInputStream(file)), width, height, true);
```

Gambar 4.4. Source code bitmap scaling

Kemudian, *bitmap* tersebut disimpan sebagai *image* dengan nama *imScale.png* pada *device mobile*. Selanjutnya halaman utama dipanggil untuk melakukan proses *cropping*.

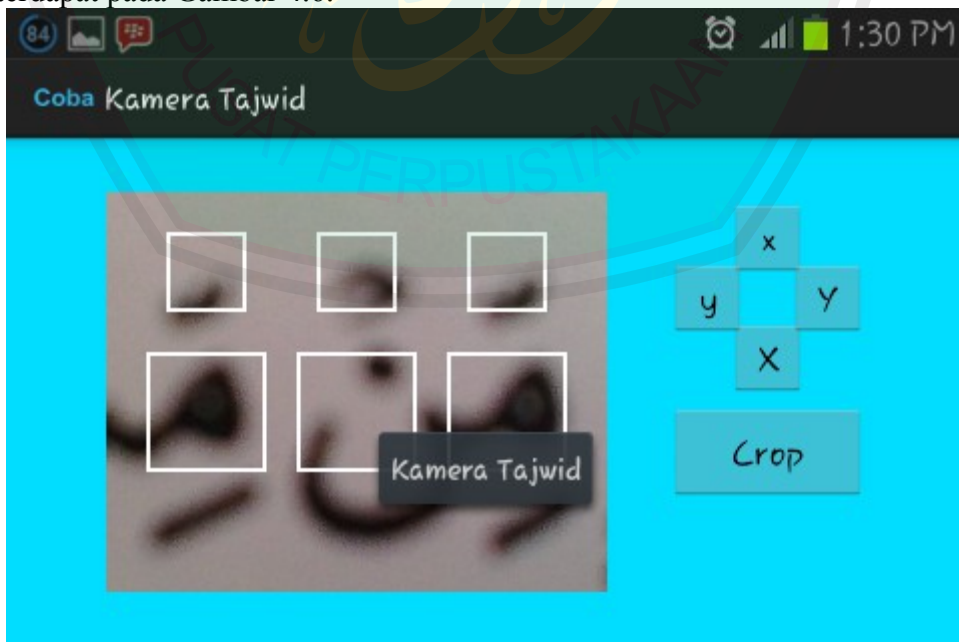
4.1.2. Halaman Utama

Pada halaman utama, tahap pertama yang dilakukan adalah menampilkan *imScale.png* ke dalam *imageView*. Untuk melakukan hal tersebut, digunakan *source code* pada Gambar 4.5.

```
ImageView iv = (ImageView) findViewById(id.image_view);  
Bitmap bitmap = BitmapFactory.decodeFile(Dir + "/imScale.png");  
iv.setImageBitmap(bitmap);
```

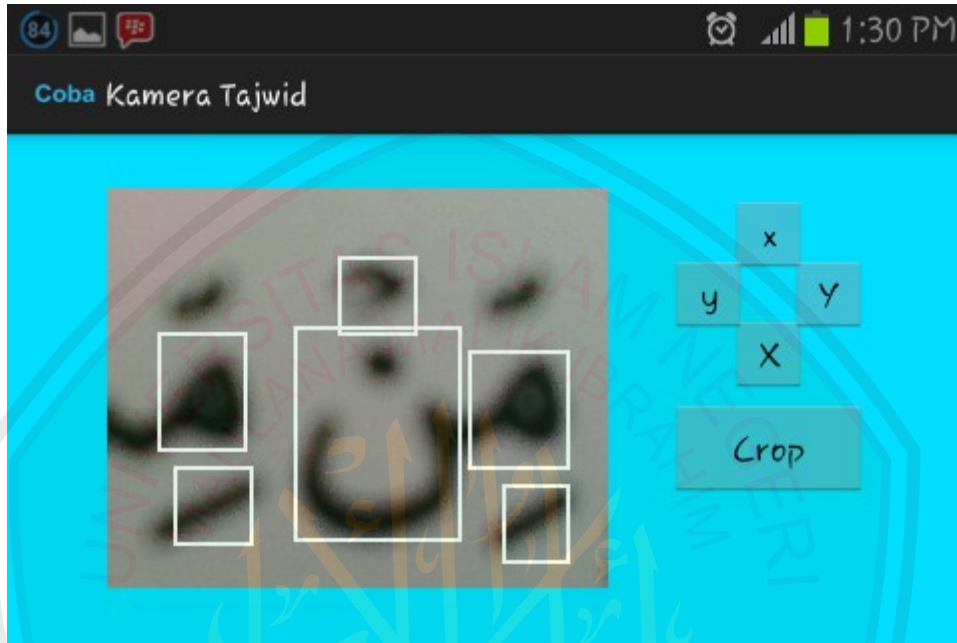
Gambar 4.5. Source code menampilkan *imScale.png*

Tampilan halaman utama setelah *imScale.png* berada dalam *imageView* terdapat pada Gambar 4.6.



Gambar 4.6. Halaman Utama

Selanjutnya dalam melakukan *cropping* berdasarkan bingkai *cropping*, diberikan 6 bingkai *cropping* bertipe *button*. Untuk mendapatkan hasil yang akurat, maka bingkai *cropping* digeser dan diubah ukurannya seperti pada Gambar 4.7.



Gambar 4.7. Halaman Utama setelah bingkai *cropping* digeser

Dalam melakukan *cropping* ini digunakan *source code* pada Gambar 4.8.

```
private Bitmap bitmapCropping(File file, Bitmap bmp, int idBox,
int x,int y, int width, int height) {
    Bitmap bitmap = null;
    int boxx = (int) findViewById(idBox).getX();
    int boxy = (int) findViewById(idBox).getY();
    int boxheight = (int) findViewById(idBox).getHeight();
    int boxwidth = (int) findViewById(idBox).getWidth();
    if (boxx >= 0 && boxy >= 0
        && (boxheight + boxy) <= 200
        && (boxwidth + boxx) <= 250) {
        bitmap = bitmap.createBitmap(BitmapFactory
            .decodeStream(new FileInputStream(file)), boxx,
            boxy, boxwidth, boxheight);
    }
    return bitmap;}

```

Gambar 4.8. *Source code cropping* berdasarkan bingkai *cropping*

Pada *source code cropping* berdasarkan bingkai *cropping*, *cropping* berdasarkan bingkai *cropping* dilakukan pada masing-masing bingkai

berdasarkan id dari bingkai yang dimaksud. Hasil dari *cropping* ini masih disimpan dalam *bitmap* untuk dilakukan *cropping* lagi berdasarkan nilai biner terluar.

Sebelum melakukan *cropping* kedua berdasarkan nilai biner terluar, dilakukan binerisasi terhadap *bitmap* tersebut terlebih dahulu. *Threshold* yang digunakan adalah sebesar 80 dengan biner 1 untuk warna hitam dan biner 0 untuk warna putih. Binerisasi menggunakan *source code* pada Gambar 4.9.

```
double[][] biner(Bitmap bmp) {
    double[][] Biner = new double[bmp.getHeight()][bmp.getWidth()];
    for (int i = 0; i < bmp.getHeight(); i++) {
        for (int j = 0; j < bmp.getWidth(); j++) {
            int pixel = bmp.getPixel(j, i);
            int red = Color.red(pixel);
            int green = Color.green(pixel);
            int blue = Color.blue(pixel);
            int graycl=(int)(red + green + blue)/3;
            if (graycl < 80) {
                graycl = 1;
            } else {
                graycl = 0;
            }
            Biner[i][j] = graycl;
        }
    }
    return Biner;
}
```

Gambar 4.9. *Source code* binerisasi

Hasil dari *bitmap* biner ini akan *dicropping* nilai biner terluarnya. Kemudian dilakukan *cropping* berdasarkan biner 1 yang terletak pada posisi paling atas, paling bawah, paling kanan dan paling kiri. Masing-masing posisi biner terluar tersebut dijadikan sebagai acuan untuk melakukan *cropping* lagi terhadap *bitmap* hasil *cropping* berdasarkan bingkai *cropping*. *Cropping* berdasarkan nilai biner terluar menggunakan *source code* pada Gambar 4.10.

```

Bitmap cropPinggir(Bitmap bmp) {
    double bin[][] = biner(bmp);
    int x0 = 0, y0 = 0, x1 = 0, y1 = 0, counter = 0;
    String X0 = "", Y0 = "", X1 = "", Y1 = "";
    for (int i = 0; i < bin.length; i++) {
        for (int j = 0; j < bin[0].length; j++) {
            if (bin[i][j] == 1 && counter == 0) {
                y0 = i;
                counter = 1;
            }
        }
    }
    counter = 0;
    for (int j = 0; j < bin[0].length; j++) {
        for (int i = 0; i < bin.length; i++) {
            if (bin[i][j] == 1 && counter == 0) {
                x0 = j;
                counter = 1;
            }
        }
    }
    counter = 0;
    for (int i = bin.length - 1; i > -1; i--) {
        for (int j = bin[0].length - 1; j > -1; j--) {
            if (bin[i][j] == 1 && counter == 0) {
                y1 = i;
                counter = 1;
            }
        }
    }
    counter = 0;
    for (int j = bin[0].length - 1; j > -1; j--) {
        for (int i = bin.length - 1; i > -1; i--) {
            if (bin[i][j] == 1 && counter == 0) {
                x1 = j;
                counter = 1;
            }
        }
    }
    return bitmapResize(bitmapCropping(null, bmp, 0, x0, y0, x1 -
x0 + 1, y1 - y0 + 1), 20, 20);
}

```

Gambar 4.10. Source code *cropping* berdasarkan nilai biner terluar

Hasil dari *cropping* berdasar nilai biner terluar ini sekaligus menjadi hasil

scaling kedua. Setelah proses ini selesai, selanjutnya hasilnya disimpan sebagai

imCrop1.png sampai imCrop6.png. Hasil penyimpanan imCrop.png terdapat

pada Gambar 4.11.



Gambar 4.11. Hasil penyimpanan imCrop.png

Kemudian dilanjutkan pada halaman hasil untuk melakukan pengecekan huruf dan harakat menggunakan implementasi *template matching* dengan korelasi pearson dan pengecekan tajwid beserta transliterasi.

4.1.3. Halaman Hasil

Pada halaman ini, semua *imCrop* mulai *imCrop1.png* sampai *imCrop6.png* ditampilkan pada masing-masing *imageView*. Fungsi untuk menampilkan tersebut terdapat dalam *source code* pada Gambar 4.12.

```
ImageView tampilBox1 = (ImageView) findViewById(id.tampilbox1);
tampilBox1.setImageBitmap(ambilFileBitmap("im"rop1.png))"
tampilBox1 = (ImageView) findViewById(id.tampilbox2);
tampilBox1.setImageBitmap(ambilFileBitmap("im"rop2.png))"
tampilBox1 = (ImageView) findViewById(id.tampilbox3);
tampilBox1.setImageBitmap(ambilFileBitmap("im"rop3.png))"
tampilBox1 = (ImageView) findViewById(id.tampilbox4);
tampilBox1.setImageBitmap(ambilFileBitmap("im"rop4.png))"
tampilBox1 = (ImageView) findViewById(id.tampilbox5);
tampilBox1.setImageBitmap(ambilFileBitmap("im"rop5.png))"
tampilBox1 = (ImageView) findViewById(id.tampilbox6);
tampilBox1.setImageBitmap(ambilFileBitmap("im"rop6.png))"
```

Gambar 4.12. *Source code* menampilkan semua *imCrop*

Selanjutnya semua *Imcrop* dirubah akan dirubah menjadi *array grayscale* untuk kemudian dicocokkan dengan *template* pada aplikasi. Fungsi untuk mengubah *imCrop* menjadi *array grayscale* terdapat dalam *source code* pada Gambar 4.13.

```
double[][] grayscale(Bitmap bmp) {
    double[][] Grayscale = new
double[bmp.getHeight()][bmp.getWidth()];
    for (int i = 0; i < bmp.getHeight(); i++) {
        for (int j = 0; j < bmp.getWidth(); j++) {
            int pixel = bmp.getPixel(j, i);
            int red = Color.red(pixel);
            int green = Color.green(pixel);
            int blue = Color.blue(pixel);
            int graycl = (int)(red + green + blue)/3;
            Grayscale[i][j] = graycl;
        }
    }
    return Grayscale;
}
```

Gambar 4.13. *Source code grayscale*

Selanjutnya dilakukan pencocokan *array* grayscale dengan template pada aplikasi dengan menggunakan metode *template matching* dengan *korelasi pearson*. *Source code* metode korelasi Pearson terdapat pada Gambar 4.14.

```
double korelasiPearson = 0, AveGray = 0, Avetemp = 0;
double[][] gray = InputGray;
double[][] temp = template;
int nBiner = gray.length * gray[0].length;
int nTemp = temp.length * temp[0].length;
for (int i = 0; i < temp.length; i++) {
    for (int j = 0; j < temp[0].length; j++) {
        AveGray += gray[i][j];
        Avetemp += temp[i][j];
    }
}
AveGray = AveGray / nBiner;
Avetemp = Avetemp / nTemp;
for (int i = 0; i < temp.length; i++) {
    for (int j = 0; j < temp[0].length; j++) {
        gray[i][j] = gray[i][j] - AveGray;
        temp[i][j] = temp[i][j] - Avetemp;
    }
}
for (int i = 0; i < temp.length; i++) {
    for (int j = 0; j < temp[0].length; j++) {
        korelasiPearson += gray[i][j] * temp[i][j];
    }
}
AveGray = 0;
Avetemp = 0;
for (int i = 0; i < temp.length; i++) {
    for (int j = 0; j < temp[0].length; j++) {
        AveGray += gray[i][j] * gray[i][j];
        Avetemp += temp[i][j] * temp[i][j];
    }
}
int tes = 0;
AveGray = AveGray * Avetemp;
AveGray = (int) Math.sqrt(AveGray);
korelasiPearson = (korelasiPearson * 100) / AveGray;
AveGray = 0;
Avetemp = 0;
```

Gambar 4.14. *Source code* korelasi Pearson

Hasil dari pencocokan pencocokan *array* grayscale dengan template pada aplikasi adalah nama huruf dan harakat. Berdasarkan jumlah *imCrop*, maka akan diperoleh 3 huruf dan 3 harakat. Kombinasi dari 3 huruf dan harakat inilah yang akan dicek tajwid beserta transliterasinya.

Setelah nama huruf dan harakat ditemukan, selanjutnya dilakukan pengecekan tajwid. Dalam pengecekan tajwid digunakan kombinasi 3 huruf dan 3 harakat. Dengan kombinasi 3 huruf dan 3 harakat tersebut kemudian dicek penyebab hukum tajwid. Pengecekan pertama dilakukan pada harakat pertama, selanjutnya dilakukan pengecekan pada harakat kedua. Fungsi untuk mengecek tajwid terdapat dalam *source code* pada Gambar 4.15.

```
String prosesTajwid() {
String hukum = "hu`um tajwid tidak cocok";
"if (harakat1.equals("su`un") || harakat1.contains("ta`n")
"    || huruf1.equals("al`f lam"))"{
"    if (harakat1.equals("su`un") "& !huruf1.equals("al`f
lam"))"{
"        {
"            if (huruf1.equals("nu`") || huruf1.equals("mi`"))"{
"                hukum = cekTajwid(huruf1, huruf2);
"            }
"        }
"        else if (harakat1.contains("ta`n") "& !huruf1.equals("al`f
lam")&" !huruf2.equals("al`f lam"))"{
"            hukum = cekTajwid("ta`win", "uruf2);
"        } else {
"            hukum = cekTajwid("al", "uruf2);
"        }
"    } else if (harakat2.equals("su`un") || harakat2.contains("ta`n")
"        || huruf2.equals("al`f lam"))"{
"        if (harakat2.equals("su`un") "& !huruf1.equals("al`f
lam"))"{
"            if (huruf2.equals("nu`") || huruf2.equals("mi`"))"{
"                hukum = cekTajwid(huruf2, huruf3);
"            }
"        } else if (harakat2.contains("ta`n") "& !
huruf2.equals("al`f lam")
"        && !huruf3.equals("al`f lam"))"{
"            hukum = cekTajwid("ta`win", "uruf3);
"        } else {
"            hukum = cekTajwid("al", "uruf3);
"        }
"        System.out.print("ma`uk cek 2");"
"        this.hukum = hukum;
"        System.out.println("te` hukum >>" +"hukum);
"        return hukum;
"    }
}
```

Gambar 4.15. *Source code* cek tajwid

Setelah tajwid dicek, selanjutnya dilakukan transliterasi. Transliterasi ini juga merupakan kombinasi dari 3 huruf dan harakat, sehingga jika salah satu

dari keenam kombinasi tersebut tidak ada maka transliterasi tidak dapat dilakukan. Fungsi transliterasi terdapat dalam *source code* pada Gambar 4.16 dan Gambar 4.17.

```

for (int i = 0; i < harakat.length; i++) {
    System.out.println(harakat[i].split(" ")[0]);
    if (harakat[i].split(" ")[0].equals("fa`hah"))"{
        if (huruf[i].equals("al`f") "|
huruf[i].equals("ha`zah"))"        {bacaHuruf[i] = "a""
"    } else if (huruf[i].equals("ai`"))"{
"        bacaHuruf[i] = "'a'"
"    } else {
"        if (huruf[i].contains("a`"))"{
"            bacaHuruf[i] = huruf[i].split("a`")[0] + "a""
"        } else if (huruf[i].contains("i`"))"{
"            bacaHuruf[i] = huruf[i].split("i`")[0] + "a""
"        } else if (huruf[i].contains("u`"))"{
"            bacaHuruf[i] = huruf[i].split("u`")[0] + "a""
"        } else {
"            bacaHuruf[i] = huruf[i].split("o`")[0] + "a""
"        }
"    }
" } else if (harakat[i].split(" ")[0].equals("ka`rah"))"{
"     if (huruf[i].equals("al`f") "|
huruf[i].equals("ha`zah"))"
"         {bacaHuruf[i] = "i""
"     } else if (huruf[i].equals("ai`"))"{
"         bacaHuruf[i] = "'i'"
"     } else {
"         if (huruf[i].contains("a`"))"{
"             bacaHuruf[i] = huruf[i].split("a`")[0] + "i""
"         } else if (huruf[i].contains("i`"))"{
"             bacaHuruf[i] = huruf[i].split("i`")[0] + "i""
"         } else if (huruf[i].contains("u`"))"{
"             bacaHuruf[i] = huruf[i].split("u`")[0] + "i""
"         } else {
"             bacaHuruf[i] = huruf[i].split("o`")[0] + "i""
"         }
"     }
" } else if (harakat[i].split(" ")[0].equals("dh`mmah"))"{
"     if (huruf[i].equals("al`f") "| huruf[i].equals("ha`zah"))"
"     {bacaHuruf[i] = "u""
"     } else if (huruf[i].equals("ai`"))"{
"         bacaHuruf[i] = "'u'"
"     } else {
"         if (huruf[i].contains("a`"))"{
"             bacaHuruf[i] = huruf[i].split("a`")[0] + "u""
"         } else if (huruf[i].contains("i`"))"{
"             bacaHuruf[i] = huruf[i].split("i`")[0] + "u""
"         } else if (huruf[i].contains("u`"))"{
"             bacaHuruf[i] = huruf[i].split("u`")[0] + "u""
"         } else {
"             bacaHuruf[i] = huruf[i].split("o`")[0] + "u""
"         }
"     }
" }
" }
}

```

Gambar 4.16. *Source code* transliterasi a

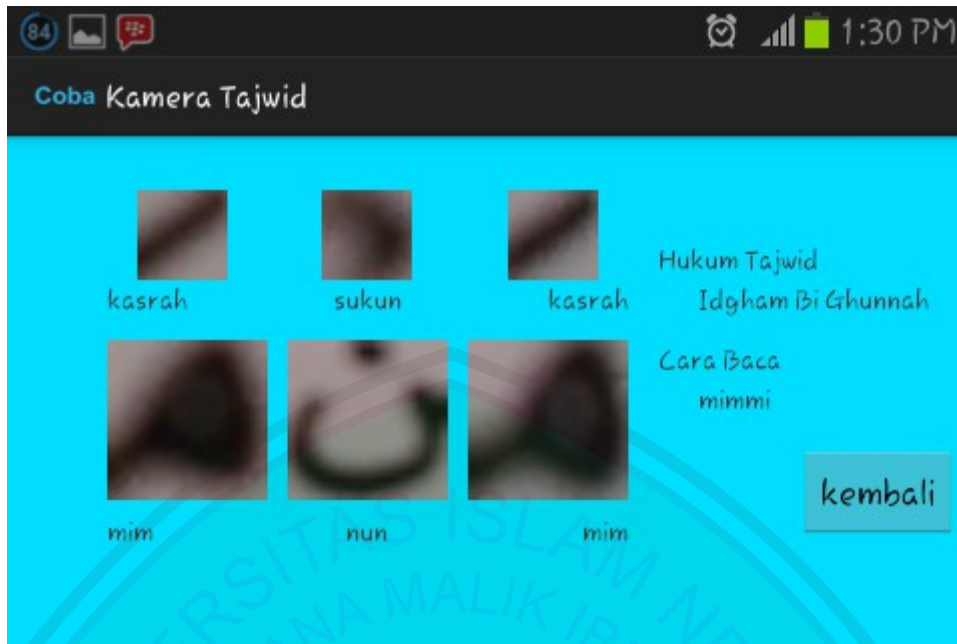
```

    } else if (harakat[i].split(" ")[0].equals("fa`hatain"))"{
        if (huruf[i].equals("al`f") "|
huruf[i].equals("ha`zah"))"{
            bacaHuruf[i] = "an";
"        } else if (huruf[i].equals("ai`"))"{
            bacaHuruf[i] = "'a";
"        } else {
            if (huruf[i].contains("a`"))"{
                bacaHuruf[i] = huruf[i].split("a`")[0] + "an";
"            } else if (huruf[i].contains("i`"))"{
                bacaHuruf[i] = huruf[i].split("i`")[0] + "an";
"            } else if (huruf[i].contains("u`"))"{
                bacaHuruf[i] = huruf[i].split("u`")[0] + "an";
"            } else {
                bacaHuruf[i] = huruf[i].split("o`")[0] + "an";
"            }}
        } else if (harakat[i].split(" ")[0].equals("ka`ratain"))"{
            if (huruf[i].equals("al`f") "|
huruf[i].equals("ha`zah"))"{                bacaHuruf[i] = "in";
"        } else if (huruf[i].equals("ai`"))"{
            bacaHuruf[i] = "'i";
"        } else {
            if (huruf[i].contains("a`"))"{
                bacaHuruf[i] = huruf[i].split("a`")[0] + "in";
"            } else if (huruf[i].contains("i`"))"{
                bacaHuruf[i] = huruf[i].split("i`")[0] + "in";
"            } else if (huruf[i].contains("u`"))"{
                bacaHuruf[i] = huruf[i].split("u`")[0] + "in";
"            } else {
                bacaHuruf[i] = huruf[i].split("o`")[0] + "in";
"            }}
        } else if (harakat[i].split(" ")[0].equals("dh`mmatain"))"{
            if (huruf[i].equals("al`f") "|
huruf[i].equals("ha`zah"))"{                bacaHuruf[i] = "un";
"        } else if (huruf[i].equals("ai`"))"{
            bacaHuruf[i] = "'u";
"        } else {
            if (huruf[i].contains("a`"))"{
                bacaHuruf[i] = huruf[i].split("a`")[0] + "un";
"            } else if (huruf[i].contains("i`"))"{
                bacaHuruf[i] = huruf[i].split("i`")[0] + "un";
"            } else if (huruf[i].contains("u`"))"{
                bacaHuruf[i] = huruf[i].split("u`")[0] + "un";
"            } else {
                bacaHuruf[i] = huruf[i].split("o`")[0] + "un";
"            }}
    }
}

```

Gambar 4.17. Source code transliterasi b

Transliterasi akan ditampilkan berdasarkan kombinasi antara 3 huruf dan 3 harakat selesai. Hasil akhir dari aplikasi ini terdapat pada Gambar 4.18.



Gambar 4.18. Halaman Hasil

Pada Halaman Hasil pada Gambar 4.18 ditampilkan semua imCrop, nama masing-masing dari huruf dan harakat hasil pencocokan *array grayscale* dengan *template* menggunakan metode *template matching* dengan korelasi Pearson, hasil pencocokan hukum tajwid dan hasil transliterasi. Pada Halaman Hasil juga terdapat tombol kembali untuk mengulang proses aplikasi mulai dari awal lagi.

4.2. Uji Coba Sistem

Uji coba yang dilakukan meliputi akurasi aplikasi dalam melakukan pencocokan huruf dan harakat, pengecekan tajwid sampai transliterasi. Pengujian ini dilakukan pada al-Quran terbitan j-Art. *Device Android* yang digunakan adalah Samsung Galaxy Young GT-S6310. Digunakan kamera dengan resolusi 3 megapiksel dengan ram sebesar 700 megabyte dan kecepatan prosessor mencapai 1 gigaHz. Aplikasi ini diuji coba dalam sistem operasi Android *jelly bean* 4.1.2. Pengujian dilakukan dengan kondisi intensitas sedang, tidak terlalu gelap maupun terlalu terang. Pengujian dilakukan dengan jarak optimal sekitar 15 cm dari objek.














Pencocokan dilakukan dengan membuka aplikasi dan masuk pada Halaman Kamera kemudian menyotot al-Quran menggunakan fitur kamera pada aplikasi. Kemudian *capture* dilakukan dengan tombol “Ambil”. Selanjutnya masuk pada Halaman Utama untuk melakukan *cropping* manual dengan menggunakan bingkai *cropping*. Setelah semua bingkai *cropping* terpasang dengan benar, dilakukan *cropping* dengan menekan tombol “crop”. Selanjutnya Halaman Hasil akan muncul dan menampilkan hasil pencocokan huruf dan harakat, pengecekan tajwid, dan transliterasi.




Data pencocokan huruf menggunakan 78 data uji. Data uji pencocokan huruf berupa *image* huruf hasil *capture* menggunakan aplikasi implementasi. Hasil pencocokan huruf pada aplikasi meliputi semua huruf hijaiyah. Data pencocokan huruf terdapat pada Tabel 4.1.

Tabel 4.1. Pencocokan Huruf

No	Image Huruf	Hasil Pencocokan Aplikasi	Hasil Sebenarnya	Benar / Salah
1		alif (ا)	alif (ا)	Benar
2		ba (ب)	ba (ب)	Benar
3		ya (ي)	ba (ب)	Salah
4		ba (ب)	ba (ب)	Benar
5		nun (ن)	Ta (ت)	Salah
6		Ta (ت)	Ta (ت)	Benar
7		Ta (ت)	Ta (ت)	Benar
8		Ta (ت)	Ta (ت)	Benar
9		sa (ث)	sa (ث)	Benar
10		sa (ث)	sa (ث)	Benar

11		sa (ث)	sa (ث)	Benar
12		sin (س)	jim (ج)	Salah
13		jim (ج)	jim (ج)	Benar
14		jim (ج)	jim (ج)	Benar
15		ha (ح)	ha (ح)	Benar
16		ha (ح)	ha (ح)	Benar
17		kha (خ)	kha (خ)	Benar
18		kha (خ)	kha (خ)	Benar
19		dal (د)	dal (د)	Benar
20		zal (ذ)	dal (د)	Salah
21		zal (ذ)	zal (ذ)	Benar
22		zal (ذ)	zal (ذ)	Benar
23		ro (ر)	ro (ر)	Benar
24		zai (ز)	zai (ز)	Benar
25		sin (س)	sin (س)	Benar
26		sin (س)	sin (س)	Benar
27		sin (س)	sin (س)	Benar
28		syin (ش)	syin (ش)	Benar
29		syin (ش)	syin (ش)	Benar
30		sad (ص)	sad (ص)	Benar
31		mim (م)	sad (ص)	Salah
32		sad (ص)	sad (ص)	Benar
33		dad (ض)	dad (ض)	Benar
34		dad (ض)	dad (ض)	Benar














35		<i>dad</i> (ض)	<i>dad</i> (ض)	Benar
36		<i>za</i> (ظ)	<i>ta</i> (ط)	Salah
37		<i>za</i> (ظ)	<i>ta</i> (ط)	Salah
38		<i>za</i> (ظ)	<i>za</i> (ظ)	Benar
39		<i>za</i> (ظ)	<i>za</i> (ظ)	Benar
40		<i>ain</i> (ع)	<i>ain</i> (ع)	Benar
41		<i>ain</i> (ع)	<i>ain</i> (ع)	Benar
42		<i>gain</i> (غ)	<i>gain</i> (غ)	Benar
43		<i>gain</i> (غ)	<i>gain</i> (غ)	Benar
44		<i>gain</i> (غ)	<i>gain</i> (غ)	Benar
45		<i>fa</i> (ف)	<i>fa</i> (ف)	Benar
46		<i>fa</i> (ف)	<i>fa</i> (ف)	Benar
47		<i>fa</i> (ف)	<i>fa</i> (ف)	Benar
48		<i>qaf</i> (ق)	<i>qaf</i> (ق)	Benar
49		<i>qaf</i> (ق)	<i>qaf</i> (ق)	Benar
50		<i>qaf</i> (ق)	<i>qaf</i> (ق)	Benar
51		<i>qaf</i> (ق)	<i>qaf</i> (ق)	Benar
52		<i>kaf</i> (ك)	<i>kaf</i> (ك)	Benar
53		<i>kaf</i> (ك)	<i>kaf</i> (ك)	Benar
54		<i>kaf</i> (ك)	<i>kaf</i> (ك)	Benar
55		<i>lam</i> (ل)	<i>lam</i> (ل)	Benar
56		<i>lam</i> (ل)	<i>lam</i> (ل)	Benar
57		<i>lam</i> (ل)	<i>lam</i> (ل)	Benar
58		<i>lam</i> (ل)	<i>lam</i> (ل)	Benar















59		<i>mim (م)</i>	<i>mim (م)</i>	Benar
60		<i>mim (م)</i>	<i>mim (م)</i>	Benar
61		<i>mim (م)</i>	<i>mim (م)</i>	Benar
62		<i>nun (ن)</i>	<i>nun (ن)</i>	Benar
63		<i>nun (ن)</i>	<i>nun (ن)</i>	Benar
64		<i>Ta (ت)</i>	<i>nun (ن)</i>	Salah
65		<i>nun (ن)</i>	<i>nun (ن)</i>	Benar
66		<i>wawu (و)</i>	<i>wawu (و)</i>	Benar
67		<i>wawu (و)</i>	<i>wawu (و)</i>	Benar
68		<i>Ha (ه)</i>	<i>Ha (ه)</i>	Benar
69		<i>Ha (ه)</i>	<i>Ha (ه)</i>	Benar
70		<i>Ha (ه)</i>	<i>Ha (ه)</i>	Benar
71		<i>ya (ي)</i>	<i>ya (ي)</i>	Benar
72		<i>ya (ي)</i>	<i>ya (ي)</i>	Benar
73		<i>ya (ي)</i>	<i>ya (ي)</i>	Benar
74		<i>alif lam (ال)</i>	<i>alif lam (ال)</i>	Benar
75		<i>lam alif (لا)</i>	<i>lam alif (لا)</i>	Benar
76		<i>Ta marbutah (ة)</i>	<i>Ta marbutah (ة)</i>	Benar
77		<i>Ta marbutah (ة)</i>	<i>Ta marbutah (ة)</i>	Benar
78		<i>hamzah (ء)</i>	<i>hamzah (ء)</i>	Benar

Data pencocokan harakat menggunakan 40 data uji. Data uji pencocokan harakat berupa *image* harakat hasil *capture* menggunakan aplikasi implementasi.. Hasil pencocokan harakat meliputi harakat *fathah*, *fathatain*, *kasrah*, *kasratain*,

dhammah, dhammatain, sukun, dan alif panjang. Data pencocokan harakat terdapat pada Tabel 4.2.

Tabel 4.2. Pencocokan Harakat

No	Image Harakat	Hasil Pencocokan Aplikasi	Hasil Sebenarnya	Benar / Salah
1		<i>fathah</i>	<i>fathah</i>	Benar
2		<i>fathah</i>	<i>fathah</i>	Benar
3		<i>fathah</i>	<i>fathah</i>	Benar
4		<i>fathah</i>	<i>fathah</i>	Benar
5		<i>fathah</i>	<i>fathah</i>	Benar
6		<i>fathatain</i>	<i>fathatain</i>	Benar
7		<i>sukun</i>	<i>fathatain</i>	Salah
8		<i>fathatain</i>	<i>fathatain</i>	Benar
9		<i>fathatain</i>	<i>fathatain</i>	Benar
10		<i>sukun</i>	<i>fathatain</i>	Salah
11		<i>kasrah</i>	<i>kasrah</i>	Benar
12		<i>kasrah</i>	<i>kasrah</i>	Benar
13		<i>kasrah</i>	<i>kasrah</i>	Benar
14		<i>kasrah</i>	<i>kasrah</i>	Benar
15		<i>kasrah</i>	<i>kasrah</i>	Benar
16		<i>kasratain</i>	<i>kasratain</i>	Benar
17		<i>kasratain</i>	<i>kasratain</i>	Benar
18		<i>kasratain</i>	<i>kasratain</i>	Benar
19		<i>kasratain</i>	<i>kasratain</i>	Benar
20		<i>kasratain</i>	<i>kasratain</i>	Benar

21		<i>dhammah</i>	<i>dhammah</i>	Benar
22		<i>dhammah</i>	<i>dhammah</i>	Benar
23		<i>dhammah</i>	<i>dhammah</i>	Benar
24		<i>dhammah</i>	<i>dhammah</i>	Benar
25		<i>dhammah</i>	<i>dhammah</i>	Benar
26		<i>dhammatain</i>	<i>dhammatain</i>	Benar
27		<i>dhammatain</i>	<i>dhammatain</i>	Benar
28		<i>dhammatain</i>	<i>dhammatain</i>	Benar
29		<i>dhammatain</i>	<i>dhammatain</i>	Benar
30		<i>dhammatain</i>	<i>dhammatain</i>	Benar
31		<i>sukun</i>	<i>sukun</i>	Benar
32		<i>sukun</i>	<i>sukun</i>	Benar
33		<i>sukun</i>	<i>sukun</i>	Benar
34		<i>sukun</i>	<i>sukun</i>	Benar
35		<i>sukun</i>	<i>sukun</i>	Benar
36		<i>alif panjang</i>	<i>alif panjang</i>	Benar
37		<i>alif panjang</i>	<i>alif panjang</i>	Benar
38		<i>alif panjang</i>	<i>alif panjang</i>	Benar
39		<i>alif panjang</i>	<i>alif panjang</i>	Benar
40		<i>alif panjang</i>	<i>alif panjang</i>	Benar

Dalam pengecekan tajwid digunakan 41 data uji. Data uji berupa kombinasi dari 3 huruf dan 3 harakat. Hasil pengecekan tajwid meliputi hukum *Idzhar*, *Idgham Bi Ghunnah*, *Idgham Bilaa Ghunnah*, *Iqlab*, *Ikhfa*, *Idgham*

Mistlain, Ikhfa Syafawi, Idzhar Syafawi Idzhar Qamariyah, Idgham Syamsiyah, dan Idzhar Wajib. Data pengecekan tajwid terdapat pada Tabel 4.3.

Tabel 4.3. Pengecekan Tajwid

N o	Huruf 1 Harakat 1	Huruf 2 Harakat 2	Huruf 3 Harakat 3	Cek Tajwid	Hasil Sebenarnya	Benar / Salah
1	<i>hamzah dhammatai n</i>	<i>alif kasrah</i>	<i>lam fathah</i>	<i>Idzhar</i>	<i>Idzhar</i>	Benar
2	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ha dhammah</i>	<i>Idzhar</i>	<i>Idzhar</i>	Benar
3	<i>hamzah dhammatai n</i>	<i>ain fathah</i>	<i>lam fathah</i>	<i>Idzhar</i>	<i>Idzhar</i>	Benar
4	<i>sa kasratain</i>	<i>gain fathah</i>	<i>ya sukun</i>	<i>Idzhar</i>	<i>Idzhar</i>	Benar
5	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ha fathah</i>	<i>Idzhar</i>	<i>Idzhar</i>	Benar
6	<i>alif kasrah</i>	<i>nun sukun</i>	<i>kha kasrah</i>	<i>Idzhar</i>	<i>Idzhar</i>	Benar
7	<i>mim fathah</i>	<i>nun sukun</i>	<i>ya dhammah</i>	<i>Idgham Bi Ghunnah</i>	<i>Idgham Bi Ghunnah</i>	Benar
8	<i>mim kasrah</i>	<i>nun sukun</i>	<i>nun fathah</i>	<i>Idgham Bi Ghunnah</i>	<i>Idgham Bi Ghunnah</i>	Benar
9	<i>qaf dhammatai n</i>	<i>mim fathah</i>	<i>nun fathah</i>	<i>Idgham Bi Ghunnah</i>	<i>Idgham Bi Ghunnah</i>	Benar
10	<i>ba fathatain</i>	<i>wawu fathah</i>	<i>lam fathah</i>	<i>Idgham Bi Ghunnah</i>	<i>Idgham Bi Ghunnah</i>	Benar
11	<i>lam dhammatai n</i>	<i>lam fathah</i>	<i>ha dhammah</i>	<i>Idgham Bilaa Ghunnah</i>	<i>Idgham Bilaa Ghunnah</i>	Benar
12	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ro fathah</i>	<i>Idgham Bilaa Ghunnah</i>	<i>Idgham Bilaa Ghunnah</i>	Benar
13	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ba fathah</i>	<i>Iqlab</i>	<i>Iqlab</i>	Benar
14	<i>kaf dhammah</i>	<i>nun sukun</i>	<i>ta dhammah</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar
15	<i>alif fathah</i>	<i>nun sukun</i>	<i>zal fathah</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar
16	<i>alif dhammah</i>	<i>nun sukun</i>	<i>za dhammah</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar
17	<i>lam fathah</i>	<i>nun sukun</i>	<i>Ta fathah</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar
18	<i>ta fathah</i>	<i>nun sukun</i>	<i>za dhammah</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar

19	<i>dad dhammatain</i>	<i>fa fathah</i>	<i>zai fathah</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar
20	<i>ta marbutoh dhammatain</i>	<i>qaf fathah</i>	<i>dal sukun</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar
21	<i>alif kasrah</i>	<i>nun sukun</i>	<i>kaf dhammah</i>	<i>Ikhfa</i>	<i>Ikhfa</i>	Benar
22	<i>kaf dhammaf</i>	<i>mim sukun</i>	<i>mim fathah</i>	<i>Idgham Mistlain</i>	<i>Idgham Mistlain</i>	Benar
23	<i>Ta dhammah</i>	<i>mim sukun</i>	<i>ba kasrah</i>	<i>Ikhfa Syafawi</i>	<i>Ikhfa Syafawi</i>	Benar
24	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>alif kasrah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
25	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>ta fathah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
26	<i>Ha dhammah</i>	<i>mim sukun</i>	<i>jim fathah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
27	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>ain kasrah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
28	<i>Ha Dhammah</i>	<i>mim sukun</i>	<i>fa kasrah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
29	<i>Ha dhammah</i>	<i>mim sukun</i>	<i>kaf dhammah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
30	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>wawu fathah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
31	<i>lam fathah</i>	<i>mim sukun</i>	<i>ya dhammah</i>	<i>Idzhar Syafawi</i>	<i>Idzhar Syafawi</i>	Benar
32	<i>alif lam sukun</i>	<i>fa kasrah</i>	<i>ta sukun</i>	<i>Idzhar Qamariyah</i>	<i>Idzhar Qamariyah</i>	Benar
33	<i>alif lam sukun</i>	<i>gain fathah</i>	<i>ro kasrah</i>	<i>Idzhar Qamariyah</i>	<i>Idzhar Qamariyah</i>	Benar
34	<i>alif lam sukun</i>	<i>ha fathah</i>	<i>qaf dhammah</i>	<i>Idzhar Qamariyah</i>	<i>Idzhar Qamariyah</i>	Benar
35	<i>alif lam sukun</i>	<i>fa alif panjang</i>	<i>sin kasrah</i>	<i>Idzhar Qamariyah</i>	<i>Idzhar Qamariyah</i>	Benar
36	<i>alif lam sukun</i>	<i>mim fathah</i>	<i>sin alif panjang</i>	<i>Idzhar Qamariyah</i>	<i>Idzhar Qamariyah</i>	Benar
37	<i>alif lam sukun</i>	<i>sad alif panjang</i>	<i>lam kasrah</i>	<i>Idgham Syamsiyah</i>	<i>Idgham Syamsiyah</i>	Benar
38	<i>alif lam sukun</i>	<i>za alif panjang</i>	<i>lam kasrah</i>	<i>Idgham Syamsiyah</i>	<i>Idgham Syamsiyah</i>	Benar
39	<i>alif lam sukun</i>	<i>lam fathah</i>	<i>zal kasroh</i>	<i>Idgham Syamsiyah</i>	<i>Idgham Syamsiyah</i>	Benar

40	<i>alif lam sukun</i>	<i>nun fathah</i>	<i>sin dhammah</i>	<i>Idgham Syamsiyah</i>	<i>Idgham Syamsiyah</i>	Benar
41	<i>dal dhammah</i>	<i>nun sukun</i>	<i>ya fathah</i>	<i>Idzhar Wajib</i>	<i>Idzhar Wajib</i>	Benar

Data pengecekan transliterasi menggunakan 41 data uji. Data uji berupa kombinasi dari 3 huruf dan 3 harakat. Hasil pengecekan transliterasi berupa cara baca dari kombinasi dari 3 huruf dan 3 harakat yang menjadi data uji. Data pengecekan transliterasi terdapat pada Tabel 4.4.

Tabel 4.4. Pengecekan Transliterasi

No	Huruf 1 Harakat 1	Huruf 2 Harakat 2	Huruf 3 Harakat 3	Transliterasi	Hasil Sebenarnya	Benar / Salah
1	<i>hamzah dhammatain</i>	<i>alif kasrah</i>	<i>lam fathah</i>	<i>unila</i>	<i>unila</i>	Benar
2	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ha dhammah</i>	<i>minhu</i>	<i>minhu</i>	Benar
3	<i>hamzah dhammatain</i>	<i>ain fathah</i>	<i>lam fathah</i>	<i>un'ala</i>	<i>un'ala</i>	Benar
4	<i>sa kasratain</i>	<i>gain fathah</i>	<i>ya sukun</i>	<i>singai</i>	<i>singai</i>	Benar
5	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ha fathah</i>	<i>minha</i>	<i>minha</i>	Benar
6	<i>alif kasrah</i>	<i>nun sukun</i>	<i>kha kasrah</i>	<i>inkhi</i>	<i>inkhi</i>	Benar
7	<i>mim fathah</i>	<i>nun sukun</i>	<i>ya dhammah</i>	<i>mayyu</i>	<i>mayyu</i>	Benar
8	<i>mim kasrah</i>	<i>nun sukun</i>	<i>nun fathah</i>	<i>minna</i>	<i>minna</i>	Benar
9	<i>qaf dhammatain</i>	<i>mim fathah</i>	<i>nun fathah</i>	<i>qummina</i>	<i>qummina</i>	Benar
10	<i>ba fathatain</i>	<i>wawu fathah</i>	<i>lam fathah</i>	<i>bawwala</i>	<i>bawwala</i>	Benar
11	<i>lam dhammatain</i>	<i>lam fathah</i>	<i>ha dhammah</i>	<i>lullahu</i>	<i>lullahu</i>	Benar
12	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ro fathah</i>	<i>mirra</i>	<i>mirra</i>	Benar
13	<i>mim kasrah</i>	<i>nun sukun</i>	<i>ba fathah</i>	<i>mimba</i>	<i>mimba</i>	Benar
14	<i>kaf dhammah</i>	<i>nun sukun</i>	<i>ta dhammah</i>	<i>kungtu</i>	<i>kungtu</i>	Benar
15	<i>alif fathah</i>	<i>nun sukun</i>	<i>zal fathah</i>	<i>angza</i>	<i>angza</i>	Benar
16	<i>alif dhammah</i>	<i>nun sukun</i>	<i>za dhammah</i>	<i>unzu</i>	<i>unzu</i>	Benar

17	<i>lam fathah</i>	<i>nun sukun</i>	<i>ta fathah</i>	<i>langta</i>	<i>langta</i>	Benar
18	<i>ta fathah</i>	<i>nun sukun</i>	<i>za dhammah</i>	<i>tangzu</i>	<i>tangzu</i>	Benar
19	<i>dad dhammatain</i>	<i>fa fathah</i>	<i>zai fathah</i>	<i>dungfaza</i>	<i>dungfaza</i>	Benar
20	<i>ta marbutoh dhammatain</i>	<i>qaf fathah</i>	<i>dal sukun</i>	<i>tungqadj</i>	<i>tungqadj</i>	Benar
21	<i>alif kasrah</i>	<i>nun sukun</i>	<i>kaf dhammah</i>	<i>ingku</i>	<i>ingku</i>	Benar
22	<i>kaf dhammaf</i>	<i>mim sukun</i>	<i>mim fathah</i>	<i>kummi</i>	<i>kummi</i>	Benar
23	<i>ta dhammah</i>	<i>mim sukun</i>	<i>ba kasrah</i>	<i>tumbi</i>	<i>tumbi</i>	Benar
24	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>alif kasrah</i>	<i>kumi</i>	<i>kumi</i>	Benar
25	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>ta fathah</i>	<i>kumta</i>	<i>kumta</i>	Benar
26	<i>ha dhammah</i>	<i>mim sukun</i>	<i>jim fathah</i>	<i>humja</i>	<i>humja</i>	Benar
27	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>ain kasrah</i>	<i>kum'i</i>	<i>kum'i</i>	Benar
28	<i>ha dhammah</i>	<i>mim sukun</i>	<i>fa kasrah</i>	<i>humfi</i>	<i>humfi</i>	Benar
29	<i>ha dhammah</i>	<i>mim sukun</i>	<i>kaf dhammah</i>	<i>humku</i>	<i>humku</i>	Benar
30	<i>kaf dhammah</i>	<i>mim sukun</i>	<i>wawu fathah</i>	<i>kumwa</i>	<i>kumwa</i>	Benar
31	<i>lam fathah</i>	<i>mim sukun</i>	<i>ya dhammah</i>	<i>lamyu</i>	<i>lamyu</i>	Benar
32	<i>alif lam sukun</i>	<i>fa kasrah</i>	<i>ta sukun</i>	<i>alfit</i>	<i>alfit</i>	Benar
33	<i>alif lam sukun</i>	<i>gain fathah</i>	<i>ro kasrah</i>	<i>algari</i>	<i>algari</i>	Benar
34	<i>alif lam sukun</i>	<i>ha fathah</i>	<i>qaf dhammah</i>	<i>alhaqu</i>	<i>alhaqu</i>	Benar
35	<i>alif lam sukun</i>	<i>fa alif panjang</i>	<i>sin kasrah</i>	<i>alfasi</i>	<i>slfasi</i>	Benar
36	<i>alif lam sukun</i>	<i>mim fathah</i>	<i>sin alif panjang</i>	<i>almasaa</i>	<i>almasaa</i>	Benar
37	<i>alif lam sukun</i>	<i>sad alif panjang</i>	<i>lam kasrah</i>	<i>assaali</i>	<i>assaali</i>	Benar
38	<i>alif lam sukun</i>	<i>za alif panjang</i>	<i>lam kasrah</i>	<i>azzaali</i>	<i>azzaali</i>	Benar

39	<i>alif lam sukun</i>	<i>lam fathah</i>	<i>zal kasroh</i>	<i>allazi</i>	<i>allazi</i>	Benar
40	<i>alif lam sukun</i>	<i>nun fathah</i>	<i>sin dhammah</i>	<i>annasu</i>	<i>annasu</i>	Benar
41	<i>Dal dhammah</i>	<i>nun sukun</i>	<i>ya fathah</i>	<i>dunya</i>	<i>dunya</i>	Benar

Pada Tabel 4.1 terdapat 8 kesalahan dari 78 uji coba pencocokan huruf. Pada Tabel 4.2 terdapat 2 kesalahan dari 40 uji coba pencocokan harakat. Pada Tabel 4.3 tidak terdapat kesalahan dari 41 uji coba pencocokan tajwid. Pada Tabel 4.4 juga tidak terdapat kesalahan dari 41 uji coba pencocokan transliterasi. Perhitungan akurasi dari masing-masing uji coba disajikan pada Tabel 4.5.

Tabel 4.5. Perhitungan Akurasi

No	Nama pencocokan	Persentase kebenaran
1	Pencocokan Huruf	$(70/78)*100\% = 89\%$
2	Pencocokan Harakat	$(38/40)*100\% = 95\%$
3	Pengecekan Tajwid	$(41/41)*100\% = 100\%$
4	Transliterasi	$(41/41)*100\% = 100\%$
Total Akurasi		96%

Berdasarkan Tabel 4.5 dapat disimpulkan bahwa aplikasi Implementasi *Template Matching* Dengan Korelasi Pearson Untuk Mendeteksi Hukum Tajwid Citra Tulisan Al-Quran Pada *Platform* Android memiliki akurasi pencocokan huruf sebesar 89%, pencocokan harakat 95%, pencocokan tajwid 100%, dan transliterasi 100%. Kemudian rata-rata dari semua akurasi tersebut adalah 96%.

Aplikasi ini memiliki kelemahan dalam hal pencocokan huruf dan harakat. Kesalahan dalam pencocokan huruf dan harakat berpengaruh pada pencocokan tajwid dan transliterasi. Hal ini disebabkan karena pencocokan tajwid dan transliterasi didasarkan pada kombinasi antara 3 huruf dan 3 harakat yang diperoleh melalui pencocokan *array grayscale* dengan *template* menggunakan metod *template matching* dengan korelasi Pearson.

Kesalahan pencocokan dapat terjadi karena kesalahan pada saat proses *capture image* maupun pada saat melakukan *cropping* menggunakan bingkai *cropping*. Pada saat proses *capture image*, dapat terjadi kesalahan jika terdapat intensitas cahaya yang terlalu gelap maupun terlalu terang sehingga gambar menjadi tidak jelas atau buram yang mengakibatkan input aplikasi menjadi tidak valid. Juga dapat karena gambar diambil terlalu jauh atau terlalu dekat sehingga gambar menjadi tidak terlihat jelas. Sedangkan kesalahan pada saat melakukan *cropping* menggunakan bingkai *cropping* dapat terjadi ketika bingkai *cropping* tidak tepat pada huruf maupun harakat yang dimaksud.

Disamping itu, aplikasi ini juga belum mampu membedakan hukum tajwid yang terdapat dalam satu maupun dua kalimat sehingga belum mampu melakukan pencocokan hukum *Mad*. Hukum *Mad* yang dimaksud adalah hukum *Mad Wajib* dan *Mad Jaiz* dimana *Mad Wajib* terjadi ketika terdapat huruf *mad* bertemu *hamzah* dalam satu kalimat sedangkan *Mad Jaiz* terjadi ketika terdapat huruf *mad* bertemu *hamzah* dalam dua kalimat.

4.3 Integrasi Aplikasi Dengan Pembelajaran Dan Tata Cara Membaca Al-Quran Yang Baik Dan Benar

Al-Quran adalah sumber ilmu dan juga mukjizat terbesar Rasulullah saw. Mempelajari al-Quran sangat diperlukan untuk memperkuat dan memperkokoh keimanan seorang muslim. Mempelajari al-Quran telah diperintahkan oleh Rasulullah saw sebagaimana tercantum dalam hadis berikut :

قال رسول الله صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ : « خَيْرُكُمْ مَنْ تَعَلَّمَ الْقُرْآنَ وَعَلَّمَهُ » رواه البخاري .

Rasulullah saw. bersabda: "Sebaik-baik engkau semua ialah orang yang mempelajari al-Quran dan mengajarkannya pula." (HR. Bukhari, Kitab *Riyadhus-Shalihin*, No. 993).

Rasulullah saw juga menjanjikan surga kepada orang yang gemar membaca al-Quran, mengingat-ingat kandungan al-Quran serta mengamalkan isi al-Quran. Hal tersebut sebagaimana tercantum dalam hadis berikut :

النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ : « يُقَالُ لِصَاحِبِ الْقُرْآنِ : اقْرَأْ وَارْتَقِ وَرَتِّلْ كَمَا كُنْتَ تُرْتِّلُ فِي الدُّنْيَا ، فَإِنَّ مَنْزِلَتَكَ عِنْدَ آخِرِ آيَةٍ تَقْرَأُهَا » رواه أبو داود ، والترمذي

Nabi saw bersabda: "Dikatakanlah - nanti ketika akan masuk syurga - kepada orang yang mempunyai al-Quran - yakni gemar membaca, mengingat-ingat kandungannya serta mengamalkan isinya: "Bacalah dan naiklah darjatmu - dalam syurga - serta *tartilkanlah* -yakni membaca perlahan-lahan- sebagaimana engkau *mentartilkan*nya dulu ketika di dunia, sebab sesungguhnya tempat kedudukanmu adalah pada akhir ayat yang engkau baca," (HR. Abu Dawud dan Tirmidzi, Kitab *Riyadhus-Shalihin*, No. 1001).

Maksud dari hadis tersebut yakni muslim yang membaca keseluruhan al-Quran akan memiliki kedudukan tertinggi namun jika terdapat muslim yang tidak membaca sampai keseluruhan al-Quran, tentu kedudukannya berada di bawah muslim yang membaca keseluruhan al-Quran. Hal ini berarti kedudukan muslim yang membaca al-Quran adalah berdasarkan jumlah bacaan al-Quran yang dibacanya.

Disamping itu, Rasulullah saw juga memerintahkan untuk *mentartilkan* bacaan al-Quran, yakni membacanya sesuai *makhraj* serta baik dan benar. Membaca al-Quran dengan sesuai *makhraj* serta baik dan benar berarti harus mengikuti kaidah tajwid yang berlaku. Hal ini karena ilmu tajwid merupakan ilmu untuk membaca al-Quran dengan baik dan benar. Hal ini ditegaskan oleh Rasulullah saw dalam hadisnya :

النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ : « مَنْ لَمْ يَتَعَنَّ بِالْقُرْآنِ فَلَيْسَ مِنَّا » رواه أبو داود

Nabi saw bersabda: "Barangsiapa yang tidak *bertaghanni* dengan al-Quran - yakni di waktu membacanya, maka ia bukanlah termasuk golongan kita." (HR. Abu Dawud, Kitab *Riyadhus-Shalihin*, No. 1007).

Bertaghanni pada hadis tersebut memiliki arti membaguskan suara, namun makna lebih luasnya adalah membaca dengan baik dan benar. Hal ini berarti jika membaca al-Quran tanpa menggunakan kaidah tajwid yang baik dan benar, berarti orang tersebut bukan termasuk golongan Rasulullah saw.

Dengan aplikasi Implementasi *Template Matching* dengan Korelasi Pearson Pearson untuk Mendeteksi Hukum Tajwid Citra Tulisan al-Quran pada *Platform* Android, maka aplikasi ini membantu menentukan hukum tajwid dan menampilkan transliterasi bacaan al-Quran. Meski masih memiliki kekurangan, namun aplikasi ini cukup mampu membantu untuk menentukan hukum bacaan nun mati dan tanwin, mim mati, dan *al-ta'rif*.

Dengan adanya aplikasi ini, diharapkan dapat membantu umat muslim untuk menentukan hukum tajwid dengan citra yang diambil melalui media *mobile*. Di samping itu, diharapkan juga adanya pengembangan penelitian untuk menjadikan teknologi citra dan *mobile* menjadi lebih baik dan beragam terutama dalam pembelajaran tentang islam baik pada lingkup ilmu tajwid, al-Quran maupun yang lainnya.

BAB V PENUTUP

5.1 Kesimpulan

1. Aplikasi Implementasi Template Matching Dengan Korelasi Pearson Untuk Mendeteksi Hukum Tajwid Citra Tulisan Al-Quran Pada Platform Android dibangun dengan cara melakukan pengumpulan data sebagai *template*, selanjutnya melakukan perancangan desain sistem dan *interface*. Langkah pertama dalam merancang desain sistem dan *interface* adalah membangun Halaman Kamera beserta fitur halaman tersebut berupa fitur *capture* kamera dan *scaling image* (250x200). Langkah kedua membuat Halaman Utama beserta fitur halaman tersebut, berupa fitur *cropping* berdasarkan bingkai *cropping*, binerisasi, *cropping* berdasarkan nilai biner terluar, dan *scaling image* (20x20). Langkah ketiga membuat Halaman Hasil beserta fitur halaman tersebut berupa pencocokan menggunakan *template matching by* korelasi Pearson, cek tajwid dan transliterasi.
2. Akurasi dari aplikasi ini mencapai 96% dari jumlah semua uji coba yang dilakukan. Dalam pencocokan huruf, akurasi yang dihasilkan adalah sebesar 89% dari 78 uji coba. Sedangkan dalam pencocokan harakat, didapatkan akurasi sebesar 95% dari 40 uji coba. Akurasi dari pengecekan tajwid mendapatkan 100% dari 41 uji coba. Akurasi dari pengecekan transliterasi juga mendapatkan 100% dari 41 uji coba.

5.2 Saran

1. Diharapkan aplikasi ini dapat dikembangkan lebih lanjut untuk meningkatkan akurasi sehingga hasil yang dihasilkan menjadi semakin *valid*.
2. Diperlukan fitur *cropping* otomatis yang mampu memotong *image* huruf dan harakat sehingga tidak diperlukan *cropping* manual yang sederhana.
3. Diperlukan tambahan fitur yang mampu memisahkan antar kalimat dalam al-Quran dan *rule tajwid* sehingga aplikasi dapat melakukan pencocokan hukum *Mad* dan hukum lainnya.



Daftar Pustaka

- An-Nawawi, Al Imam. 1995. *Riyadhus-Shalihin*.
- Bluman, Allan G. 2004 *Elementary Statistic A Step by Step Approach Fifth Edition*, New York : Mc Graw Hill companies.
- Das, M. Swamy et al.2010. *Segmentation Of Overlapping Text Lines, Characters In Printed Telugu Text Document Images*.
- Hastiana, Runmi. 2010. *Segmentasi Citra Digital Pembuluh Darah Mata Untuk Mendeteksi Tingkat Keparahan Diabetic Retinopathy*, Skripsi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Hendry, Jans. 2011. *Using Profile Projection To Segment Character In Image (Matlab)*.
- Ichwanti, Nurul. 2013. *Implementasi Pemrograman Citra Metode Cropping Menggunakan Csharp Dan Matlab*.
- Irwan B.S., M.Kom. 2013. *Aplikasi Korelasi Pearson Dalam Membangun Model Tree-Augmented Network (TAN)*, MATICS (jurnal Ilmu Komputer dan Teknologi Informasi UIN Malang), vol 5.
- Latecki, Longin Jan. 2002. *Template Matching Persentation Based on a project by Roland Mieziank*.
- Leksono, Bowo et al. 2010. *Aplikasi Metode Template Matching Untuk Klasifikasi Sidik Jari*.
- Mardiyah, Ainatul. 2008. *Pemisahan Tulisan Arab Sambung Menjadi Pola Huruf Hijaiah*, Skripsi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Muthohhar, Ahmad bin, Abdur Rahman Al-Murofi dan Adnin, H. 2012. *Tuhfah Al-Tullab*, Surabaya: Tri Jayaguna Offset.
- Mutholib, Abdul et al. 2013. *Development of Portable Automatic Number Plate Recognition System on Android Mobile Phone*.
- Pramana, Indra et al. 2010. *Tracking Object Menggunakan Metode Template Matching Berbasis Stereo Vision*.
- Pratama, Widiyanto. 2011. *Tutorial Android Progamming*.
- Putra, Dharma. 2010. *Pengolahan Citra Digital*, Yogyakarta: Penerbit Andi.

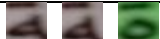
- Ridwan, Asep H. 2007. *Tahsin, Tartil Dan Tahfidz*.
- Rochmah, Nurma Zubaidatur. 2010. *Aplikasi Pencarian Hukum Bacaan (Tajwid) Pada Juz Amma*, Skripsi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Rodrigues, Roberto J. dan Thomé, Antonio Carlos Gay. 1996. *Cursive Character Recognition – A Character Segmentation Method Using Projection Profile-Based Technique*.
- Wardhana, Arditya Wishnu dan Yudi Prayudi. 2008. *Penggunaan Metode Template Matching Untuk Identifikasi Kecacatan Pada PCB*.
- Zainuddin. 2011. *Aplikasi Pengenalan Karakter (Alfabet, Angka) dengan Dua Bahasa (Indonesia, Inggris) Penerjemah*, Skripsi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.



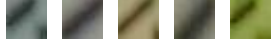

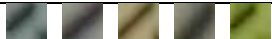
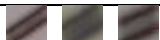



Lampiran

Tabel *Template Huruf*

No	Nama Sampel	Data Image
1	<i>alif</i>	
2	<i>ba</i>	
3	<i>ta</i>	
4	<i>sa</i>	
5	<i>jim</i>	
6	<i>ha</i>	
7	<i>kha</i>	
8	<i>dal</i>	
9	<i>zal</i>	
10	<i>ro</i>	
11	<i>zai</i>	
12	<i>sīn</i>	
13	<i>syīn</i>	
14	<i>sad</i>	
15	<i>dad</i>	
16	<i>ta</i>	
17	<i>za</i>	
18	<i>ain</i>	
19	<i>gain</i>	
20	<i>fa</i>	
21	<i>qaf</i>	
22	<i>kaf</i>	
23	<i>lam</i>	
24	<i>mim</i>	
25	<i>nun</i>	
26	<i>wawu</i>	
27	<i>ha</i>	
28	<i>lam alif</i>	
28	<i>hamzah</i>	
29	<i>ya</i>	
30	<i>alif lam</i>	

31	<i>ta marbutoh</i>	
----	--------------------	---

Tabel *Template* Harakat

No	Nama Sampel	Data Image
1	<i>fathah</i>	
2	<i>fathatain</i>	
3	<i>kasrah</i>	
4	<i>kasratain</i>	
5	<i>dhammah</i>	
6	<i>dhammatain</i>	
7	<i>sukun</i>	
8	<i>alif panjang</i>	