

PENGEMBANGAN APLIKASI ANDROID UNTUK PENGENALAN CITRA NOMOR SERTIFIKAT HALAL MUI DENGAN *LIBRARY TESSERACT OPTICAL CHARACTER RECOGNITION (OCR)*

Muhtadii, Hilmy A. Tawakal

Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jl. Lenteng Agung Raya No. 20, Srengseng Sawah, Jakarta Selatan, 12640
mjusuf92@yahoo.com

Abstrak

Semakin beragamnya makanan dan minuman di pasaran membuat konsumen muslim harus bisa memilih mana makanan dan minuman yang halal dan mana yang tidak. Salah satu metode pengenalan produk halal adalah dengan pemberian sertifikat halal bermasa berlaku yang dikeluarkan oleh Majelis Ulama Indonesia. Maka dari itu diperlukan sebuah aplikasi yang dapat mengidentifikasi keaslian sertifikat halal MUI berdasarkan nomor sertifikatnya. Aplikasi dapat mengidentifikasi nomor sertifikat halal secara manual dan otomatis menggunakan citra. Untuk dapat mengenali karakter pada sebuah citra akan digunakan sebuah sistem *Optical Character Recognition (OCR)*. OCR merupakan sistem komputer yang dapat membaca karakter pada sebuah citra. Dari sekian banyak sistem OCR yang ada, pada penelitian ini akan menggunakan tesseract OCR karena tesseract memiliki tingkat akurasi yang cukup besar yaitu diatas 94%. Aplikasi akan dikembangkan dengan metode pengembangan perangkat lunak dengan metode *waterfall* versi sommerville. Dengan menggunakan metode *waterfall*, tahapan-tahapan pengembangan akan jelas, nyata dan praktis. Dengan begitu pengembangan aplikasi menjadi lebih terjadwal dan mudah dikontrol serta pendokumentasian tahapan pengembangan dapat dilakukan dengan baik. Hasil penelitian ini adalah sebuah aplikasi android yang dapat mengidentifikasi nomor sertifikat halal secara manual dan otomatis menggunakan citra.

Kata kunci : Sertifikat Halal, Majelis Ulama Indonesia, Citra, *Optical Character Recognition, Tesseract, Android*

Abstract

Nowadays, more increasing the various kinds of foods and drinks in the market, make a lot of muslim customers have to aware and choose which are the halal foods and drinks and which are not halal. A simple method to recognize the halal products is by giving the products the MUI's halal certificate which has a validation period and must be passed the terms and conditions that are applied by Indonesian Ulama Council (Majelis Ulama Indonesia/MUI). Therefore, it should be needed an application which can be use to identify the originality and authenticity of the MUI's halal Certificates based on its numbers. The application can be used to identify the numbers of MUI's Halal Certificate by manually or automatically using the image of the numbers. To recognize the character of an image numbers, it uses an Optical Character Recognition (OCR) system. OCR is a computerized system that can read the characters of an image, such as numbers and letters. From many usefulness OCR system, the research of the application uses the Tesseract OCR, because Tesseract itself has a high accuracy level, which is above 94%. The application would have been developed using the software development system with waterfall method in sommerville version. By using the waterfall method, the development steps could be clear enough, realistic and more practical. Therefore, the development of the application should be more scheduled and more easier to control, and the documentation of the development steps could be more properly implemented. The result of this research is an android based application that can be used to identify and recognize the MUI's halal certificate numbers, both manually and automatically by using its image.

Keywords : Halal Certificate, Indonesian Ulama Council, Image, *Optical Character Recognition, Tesseract, Android*

1. PENDAHULUAN

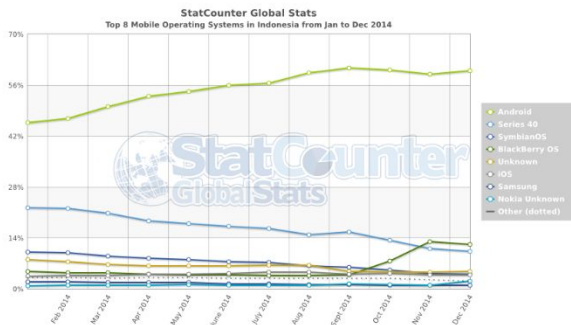
Semakin beragamnya makanan dan minuman yang tersedia di pasaran membuat para konsumen muslim

harus bisa memilih mana makanan dan minuman yang halal dan mana yang tidak. Oleh karena itu, pengenalan akan produk halal harus terus ditingkatkan kualitas dan kuantitasnya. Salah satu metode pengenalan produk

halal adalah dengan pemberian sertifikat halal bermasa berlaku yang dikeluarkan oleh Majelis Ulama Indonesia (MUI).

Namun dengan hanya pemberian label halal pada kemasan dirasa kurang karena para produsen tetap banyak yang berani melakukan kecurangan dengan memberikan label halal tanpa melalui prosedur yang telah ditetapkan oleh MUI. Seperti kasus yang terjadi beberapa tahun belakangan, ditemukan adanya logo halal MUI pada Bakso Planetaria 56 yang mengandung daging babi. Padahal data yang terkoneksi dengan MUI Pusat tidak terdapat nama merk maupun nama produsen bakso tersebut [1].

Disamping itu seiring dengan perkembangan zaman, teknologi juga ikut berkembang pesat. Salah satu teknologi yang sangat populer pada saat ini adalah *smartphone*. Berdasarkan portal berita teknologi techinasia.com, bisnis *smartphone* semakin pesat perkembangannya di Indonesia, seperti yang dikemukakan Djatmiko Wardoyo selaku Direktur *Marketing and Communications* Erajaya, bahwa hingga akhir tahun 2014 diperkirakan kenaikan penjualan *smartphone* tumbuh sekitar 10-15 persen. Pada tahun 2013, angka penjualan *handphone* mencapai 55 juta unit dengan rincian, 72 persen untuk *feature phone* dan 28 persen untuk *smartphone*. Sedangkan di tahun ini ia memperkirakan angkanya akan bergeser ke 65 persen untuk *feature phone* dan 35 persen untuk *smartphone* [2].



Gambar 1. Statistik sistem operasi *smartphone* di Indonesia tahun 2014, riset oleh StatCount (<http://gs.statcounter.com>, 2016).

Data laporan hasil riset dari StatCounter - *Website Analisa Statistik*. Android merupakan sistem operasi yang mendominasi peredaran *smartphone* di tanah air dengan pembagian pasar sebesar 59,91 persen. Diperingkat kedua, BlackBerry dengan pembagian pasar sebesar 12,18 persen. Dan diperingkat ketiga, ternyata Indonesia masih di dominasi dengan *feature phone* - Series 40 dari Nokia yang memiliki akses internet - dengan pembagian pasar sebesar 10,36 persen [3].

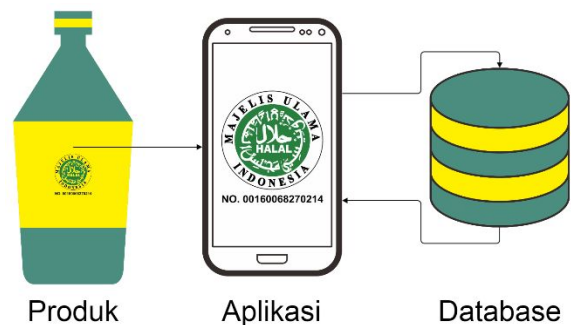
Selain pada sisi *hardware*, perkembangan teknologi juga berkembang pesat pada sisi *software*. Salah satunya pada teknologi *input*. Seperti yang diketahui, *input* dapat dilakukan dengan berbagai macam cara, salah satunya dengan menggunakan *keyboard*. Akan

tetapi dengan berkembangnya kesibukan manusia, maka diperlukan suatu metode yang lebih praktis dalam melakukan *input*. Untuk mengatasi masalah tersebut, maka penulis memanfaatkan teknologi *Optical Character Recognition (OCR)*. OCR adalah sebuah sistem komputer yang dapat membaca karakter termasuk huruf, baik yang berasal dari sebuah mesin cetak (*printer* atau mesin ketik) maupun yang berasal dari tulisan tangan. Dengan adanya sistem OCR maka pengguna dapat lebih leluasa memasukan kata ataupun kalimat karena pengguna tidak harus memakai papan *keyboard* [4].

Sistem OCR menggunakan sebuah citra untuk dapat mengenali karakter. Citra adalah representasi dari dua dimensi untuk bentuk fisik nyata dari tiga dimensi. Citra dalam perwujudannya dapat bermacam-macam, seperti gambar berwarna yang bergerak pada pesawat televisi. Citra dapat dikatakan pula sebagai sebuah gambar, foto yang ditampilkan atau bentuk lain yang memberikan representasi visual tentang sebuah objek atau pemandangan. Pada pemrosesan citra digital sebuah gambar bilangan array 2 dimensi, yang setiap barisnya adalah representasi piksel pada gambar setiap barisnya. Minimum nilai piksel = 0 (hitam) dan maksimum = 225 (putih) [5].

Diantara *library OCR*, tesseract adalah salah satu OCR *open source* yang paling akurat yang ada sekarang. Berdasarkan penelitian yang dilakukan oleh Ray Smith, Daria Antonova dan Dar-Shyang Lee, Tesseract memiliki tingkat akurasi yang cukup besar, yaitu diatas 94% [6]. Dengan menggunakan Tesseract sebagai *library OCR* diharapkan akan menghasilkan akurasi yang besar pula.

Tujuan dari penelitian ini yaitu antara lain: mengimplementasikan tesseract pada aplikasi pengenalan citra nomor sertifikat halal, mengetahui presentase tingkat keakuratan tesseract dalam mengenali nomor sertifikat halal yang biasanya dicetak sangat kecil pada kemasan produk dan mempermudah masyarakat untuk mengetahui produk-produk halal.



Gambar 2. Ilustrasi cara kerja aplikasi.

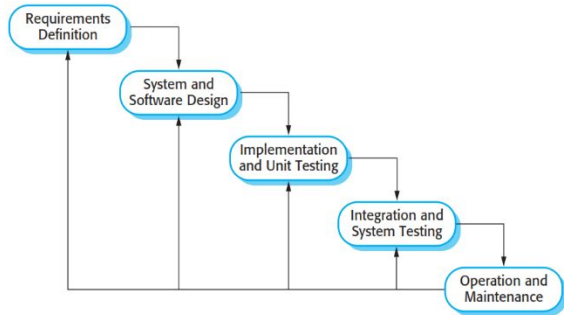
2. METODE

A. Metode Penelitian

Aplikasi dikembangkan dengan menggunakan metode pengembangan perangkat lunak *waterfall* versi

Sommerville. Alasan penulis menggunakan metode *waterfall* dikarenakan metode ini mempunyai tahapan-tahapan yang jelas, nyata dan praktis. Setiap tahap harus diselesaikan terlebih dahulu untuk menghindari terjadinya *error* pada tahap berikutnya. Dengan begitu pengembangan aplikasi menjadi lebih terjadwal dan mudah untuk kontrol. Disamping itu dengan metode *waterfall*, pendokumentasian tahapan pengembangan aplikasi dapat dilakukan dengan baik yang memudahkan pembuatan laporan penelitian.

Proses-proses yang akan dilakukan pada metode *waterfall* versi Sommerville adalah sebagai berikut:



Gambar 3. Alur proses metode *waterfall* versi sommerville (Ian Sommerville, 2011).

1. Requirements Analysis and Definition

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh *software* yang akan dibangun. Hal ini sangat penting mengingat *software* harus dapat berinteraksi dengan elemen-elemen lain seperti *hardware*, *database*, dsb [7].

2. System and Software Design

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik *hardware* maupun *software* dengan membentuk arsitektur sistem secara keseluruhan [8]. Proses *software design* untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk *blueprint* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya [7].

3. Implementation and Unit Testing

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya [8].

4. Integration and System Testing

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer* [8].

5. Operation and Maintenance

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru [8].

B. Analisis Kebutuhan

Kebutuhan-kebutuhan dalam pengembangan aplikasi didefinisikan dan dikumpulkan untuk mulai merancang aplikasi. Batasan-batasan dalam pengembangan aplikasi juga harus didefinisikan agar tidak terjadi kesalahan ketika implementasi dan pengujian pada tahap selanjutnya

C. Desain Aplikasi

Aplikasi dikembangkan menggunakan *library* tesseract OCR. Tesseract OCR adalah sistem Optical Character Recognition (OCR) *open source* yang dapat digunakan di berbagai sistem operasi. OCR adalah suatu proses mengkonversi citra menjadi teks.

D. Implementasi

Pada tahap ini seluruh kebutuhan-kebutuhan di terjemahkan ke dalam bahasa pemrograman. Pemrograman dilakukan pada unit-unit kecil yang nanti akan digabungkan menjadi satu kesatuan. Sebelum digabungkan, kebutuhan-kebutuhan akan diperiksa terlebih dahulu untuk memastikan apakah semua kebutuhan sudah terpenuhi. Implementasi dirancang dan dibangun dengan menggunakan spesifikasi perangkat keras dan perangkat lunak, antara lain :

a. Perangkat Keras:

- Notebook Acer Aspire 4732Z

Sebagai *tool* untuk melakukan proses desain dan implementasi aplikasi.

- Samsung Galaxy S II GT-I9100

Sebagai *tool* untuk melakukan proses pengujian aplikasi.

b. Perangkat Lunak:

- Ubuntu 12.04.5 LTS (Precise Pangolin)

Sebagai *interface* antara peneliti dengan perangkat keras yang digunakan untuk proses desain dan implementasi aplikasi.

- Android 4.1.2

Sebagai sistem operasi untuk *tool* yang digunakan pada proses pengujian.

- Eclipse

Sebagai *tool Integrated Development Environment (IDE)* untuk mempercepat proses *coding* pengembangan aplikasi

- Android SDK.

Sebagai *library application* agar aplikasi dapat berjalan di sistem operasi Android.

E. Pengujian

Pada tahap ini, aplikasi yang telah selesai dikembangkan akan diuji berdasarkan kebutuhan yang telah didefinisikan serta tingkat akurasi dalam mengenali nomor sertifikat halal. Jika masih terdapat kesalahan-kelalahan yang tidak ditemukan pada tahap sebelumnya, maka aplikasi akan kembali ke tahap sebelumnya untuk diperbaiki.

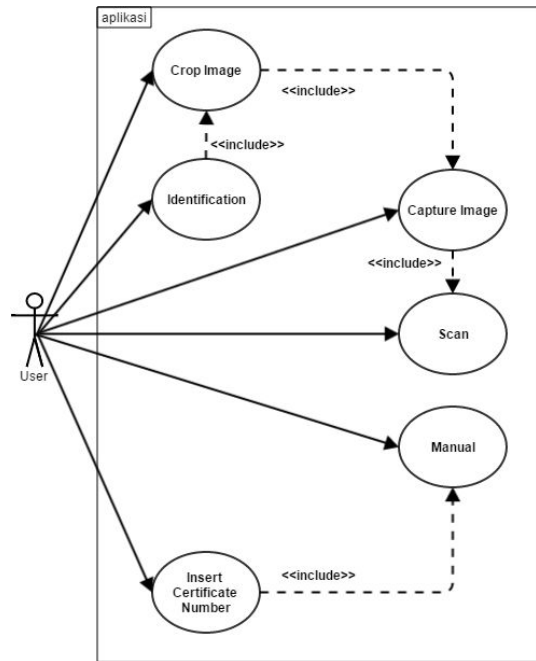
3. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Pada *use case* terdapat 6 kebutuhan fungsional pengguna yaitu *scan*, *manual*, *capture image*, *crop image*, *identification* dan *insert certificate number*.

Tabel 1. Daftar requirement yang terdapat pada aplikasi.

Kebutuhan	Keterangan
<i>Scan</i>	Merupakan menu untuk memulai proses identifikasi dengan menggunakan citra sebagai <i>input</i> .
<i>Manual</i>	Merupakan menu untuk memulai proses identifikasi secara manual dengan menggunakan <i>keyboard smartphone</i>
<i>Capture Image</i>	Mengambil citra menggunakan kamera <i>smartphone</i>
<i>Crop Image</i>	Memotong citra hanya pada bagian nomor sertifikatnya saja
<i>Identification</i>	Mengenali nomor sertifikat halal yang terdapat pada sebuah citra menggunakan <i>library tesseract</i>
<i>Insert Certificate Number</i>	Memasukan nomor sertifikat halal pada <i>form</i> yang telah disediakan

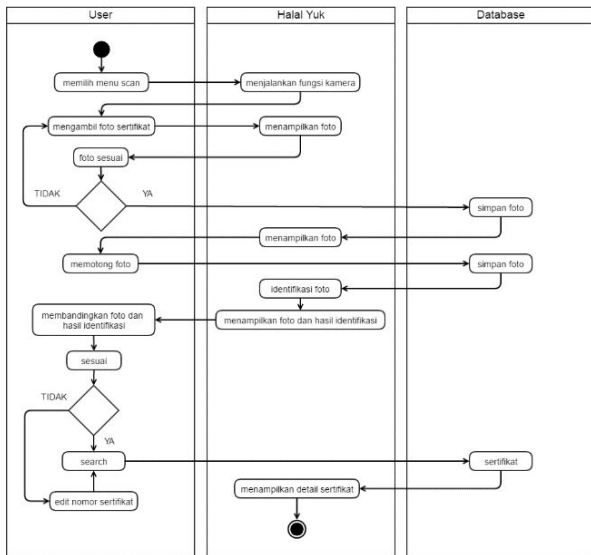


Gambar 4. Use case diagram aplikasi.

B. Desain Sistem

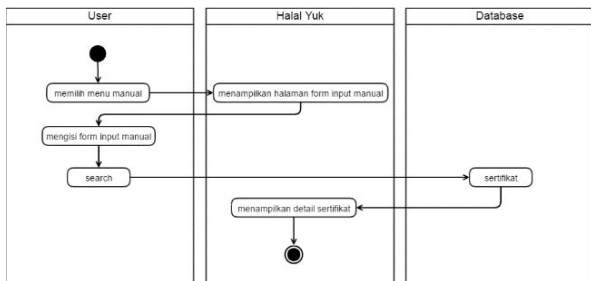
Berdasarkan *use case* pada tahap sebelumnya dapat dibuat sebuah *activity diagram* yang menghubungkan antar komponen-komponen.

Pada menu *scan*, *user* diharuskan mengambil citra nomor sertifikat halal menggunakan kamera *smartphone*. Kemudian bila citra sudah didapatkan, aplikasi akan menjalankan fungsi *crop*. *User* dapat memotong citra tepat hanya pada bagian nomor sertifikatnya agar tesseract lebih optimal dalam mengidentifikasi nomor sertifikat halal. Aplikasi akan menampilkan citra setelah dipotong dan hasil identifikasinya agar *user* dapat membandingkan keduanya. Bila tidak sesuai *user* dapat megeditnya dan bila sudah sesuai *user* dapat memulai mencari data berdasarkan nomor sertifikat halal tersebut di-*database*. Terakhir bila nomor sertifikat halal ditemukan aplikasi akan menampilkan detail sertifikat halal tersebut



Gambar 5. Activity diagram menu scan aplikasi.

Sedangkan pada menu manual, aplikasi akan menampilkan halaman *form input manual*. User dapat memasukkan nomor sertifikat halal pada *form* yang telah disediakan kemudian menekan tombol OK untuk memulai proses pencarian di-*database*. Bila hasil pencarian ditemukan, aplikasi akan menampilkan detail sertifikat halal tersebut.



Gambar 6. Activity diagram menu manual aplikasi.

Untuk penyimpanan data di *database*, akan digunakan *database* SQLite. SQLite merupakan sebuah basis data yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat *public domain* yang dikerjakan oleh D. Ricard Hipp. SQLite memiliki fitur relasional *database*, hampir sama dengan SQL pada desktop hanya saja SQLite membutuhkan memori sedikit [9].

a. Struktur Data

Pada aplikasi ini data daftar sertifikat disimpan di *database* *certificate* dengan nama tabel *certificate*. Adapun struktur tabel *certificate* adalah sebagai berikut:

cid	name	type	notnull	dfilt_value	pk
1 0	id	INTEGER	0	NULL	1
2 1	certificate_no	TEXT	0	NULL	0
3 2	product_name	TEXT	0	NULL	0
4 3	company	TEXT	0	NULL	0
5 4	valid_from	TEXT	0	NULL	0
6 5	valid_to	TEXT	0	NULL	0

Gambar 7. Struktur tabel *certificate*.

Tabel *certificate* memiliki 6 kolom yaitu *id* bertipe *integer primary key* dan *certificate_no*, *product_name*, *company*, *valid_from* dan *valid_to* bertipe TEXT. Kolom *certificate_no* tidak bertipe INTEGER karena ada beberapa nomor sertifikat dimulai dengan awalan angka 0. Sedangkan kolom *valid_from* dan kolom *valid_to* tidak ber-tipe DATE karena SQLite hanya memiliki tipe data NULL, INTEGER, REAL, TEXT, dan BLOB. Dengan menggunakan tipe data TEXT pemanggilan data saat proses *coding* juga akan menjadi lebih mudah.

b. Data

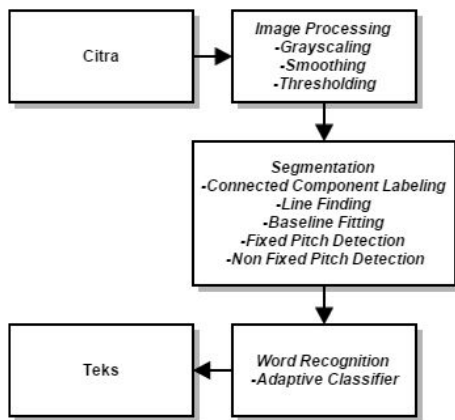
Tabel *certificate* akan diisi dengan data-data sesuai dengan nama kolomnya sebagai berikut:

id	certificate_no	product_name	company	valid_from	valid_to
1 1	00010002311297	Beef Sausages VILLADORP	KEMANG FOOD INDUSTRIES, PT	2015-10-07	2017-10-06

Gambar 8. Data sertifikat halal disimpan pada tabel *certificate*.

- *id* sebagai INTEGER PRIMARY KEY yang membedakan setiap baris *record* yang ada pada tabel *certificate*.
- *certificate_no* sebagai TEXT berisikan nomor sertifikat halal setiap produk.
- *product_name* sebagai TEXT berisikan nama produk .
- *company* sebagai TEXT berisikan nama perusahaan pembuat produk.
- *valid_from* sebagai TEXT berisikan tanggal sertifikat halal mulai berlaku.
- *valid_to* sebagai TEXT berisikan tanggal habis masa berlaku sertifikat halal.

Penelitian ini menghasilkan aplikasi Android untuk pengenalan citra nomor sertifikat halal MUI dengan *library* tesseract OCR. Pada tahap identifikasi foto proses yang dilakukan adalah mengkonversi citra menjadi teks menggunakan *library* tesseract OCR.



Gambar 9. Alur proses tesseract dalam mengenali karakter (Hanny Rindiani, Karlina Khiyarin Nisa).

Untuk dapat mengenali karakter, tesseract melakukan 3 proses besar yang terdiri dari:

1. Image Processing

Merupakan tahap awal yang berfungsi merubah citra berwarna menjadi biner. Pada prosesnya, *image processing* melakukan 3 tahapan, yaitu *grayscale*, *smoothing* dan *thresholding*.

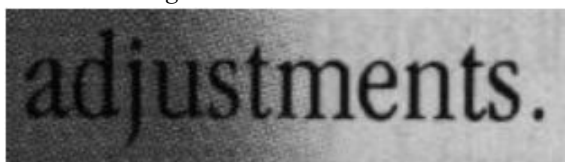
- *Grayscale*

Merupakan proses mengubah citra berwarna menjadi citra yang hanya memiliki derajat keabuan. Pada proses perubahan, *output* dalam tahap ini akan digunakan untuk binerisasi citra. Program akan melakukan pengulangan untuk *scan* citra perpixel pada koordinat (x,y) sebesar ukuran panjang dan lebar citra.

- *Smoothing*

Selanjutnya dilakukan *smoothing*, yaitu proses penghalusan citra yang bertujuan mengurangi *noise*.

- *Thresholding*



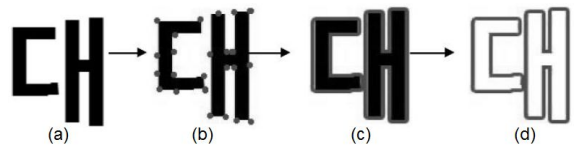
Gambar 10. Tahap *thresholding* (Hanny Rindiani, Karlina Khiyarin Nisa).

Thresholding merupakan suatu proses memisahkan *background* dengan objek yang ingin diamati dengan mengubah gambar menjadi hitam putih. Pada tahap ini *thresholding* dapat mengubah gambar berwarna maupun *grayscale* menjadi citra *biner* dengan mengubah masing-masing pixel dalam kisaran tertentu.

2. Segmentation

Setelah proses *image processing* selesai, proses selanjutnya adalah *segmentation*. Pada proses ini terdapat 4 proses yaitu *connected component labeling*, *line finding*, *baseline fitting*, *fixed pitch*, dan *non fixed pitch detection*.

- *Connected Component Labeling*



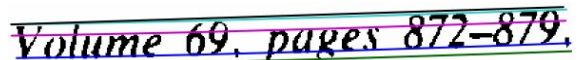
Gambar 11. Proses pencarian outline. (a) karakter yang ingin dicari outline-nya (b) pembelahan pada karakter (c) blob membentuk outline (d) outline karakter (Hanny Rindiani, Karlina Khiyarin Nisa).

Proses untuk mendeteksi komponen-komponen karakter yang saling terhubung. Pada proses ini tesseract melakukan pencarian sepanjang citra kemudian mengidentifikasi pixel latar depan (*outline*), proses pembelahan akan terus dilakukan sampai semua pixel terluar ditandai sebagai *blob* atau karakter potensial.

- *Line Finding*

Merupakan algoritma untuk mencari baris pada teks. Pada tesseract, algoritma *line finding* dirancang supaya halaman miring dapat dikenali tanpa harus melakukan *deskew* (proses untuk mengubah halaman yang miring menjadi tegak lurus) sehingga tidak menurunkan kualitas citra. Pada tahap ini pencarian baris pada teks dilakukan menggunakan *blob filtering* dan *line construction*.

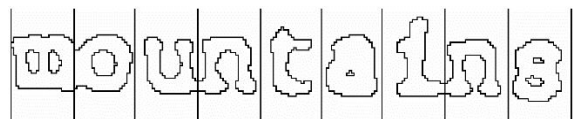
- *Baseline Fitting*



Gambar 12. Tahap *baseline fitting algorithm* (Hanny Rindiani, Karlina Khiyarin Nisa).

Setelah baris teks telah ditemukan, garis pangkal (*baseline*) dicocokkan secara lebih tepat menggunakan *quadratic spline*. *Quadratic spline* merupakan metode untuk menghasilkan titik pada sebuah rentang data yang telah diketahu sebelumnya. *Quadratic spline* menggunakan polinomial dengan *degree* rendah sehingga mampu membentuk garis dengan halus.

- *Fixed Pitch Detection*



Gambar 13. Tahap *fixed pitch detection* (Hanny Rindiani, Karlina Khiyarin Nisa).

Proses selanjutnya tesseract memperkirakan lebar karakter yang dideteksi. Proses ini mendeteksi karakter yang memiliki lebar tetap. Bila karakter berhasil dideteksi, selanjutnya tesseract melakukan pemotongan (*chopping*) karakter sehingga karakter pada teks menjadi terpotong-potong. Potongan ini yang selanjutnya akan diklasifikasikan.

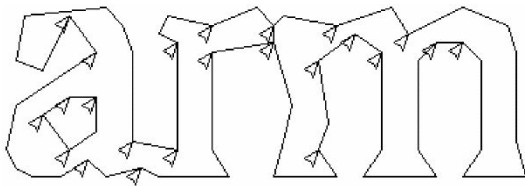
- *Non Fixed Pitch Detection*

Bila teks yang digunakan tidak memiliki lebar

garis tepi tetap, tesseract akan melakukan algoritma *non fixed pitch detection* dengan cara mengukur batasan kesenjangan antara *base line* dengan *mean line* (garis tengah). Ruang pada citra yang mendekati nilai *threshold* akan diklasifikasikan dengan *fuzzy* sehingga akan ditentukan sebagai bagian fitur yang akan dikenali ataupun tidak.

- *Chopping Characters*

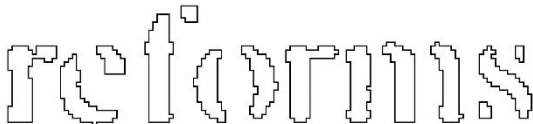
Pada tahap ini kandidat untuk titik-titik pemisahan ditemukan dari simpul cekung, poligonal *outline* dan titik cekung yang berlawanan.



Gambar 14. Kandidat titik pemotongan (Hanny Rindiani, Karlina Khyyarin Nisa).

Pada gambar 14 diatas menunjukkan kumpulan kandidat titik pemisahan dengan panah dan pemotongan terpilih sebagai sebuah garis melewati *outline* di mana huruf 'r' menyentuh huruf 'm'. Pemotongan dilakukan sesuai dengan urutan prioritas. Setiap pemotongan yang gagal tidak sepenuhnya dibuang tetapi disimpan oleh *associator* sehingga dapat digunakan kembali jika dibutuhkan.

- *Associating Broken Characters*



Gambar 15. Karakter terputus yang dapat dikenali (Hanny Rindiani, Karlina Khyyarin Nisa).

Pada tahap pemotongan karakter tidak menghasilkan karakter yang cukup bagus, kemudian dilanjutkan ke *associator*. *Associator* membuat pencarian dengan menarik kandidat *state* baru dari antrian prioritas dan mengevaluasi kandidat dengan mengklasifikasikan kombinasi fragmen yang belum terklasifikasi.

3. Word Recognition



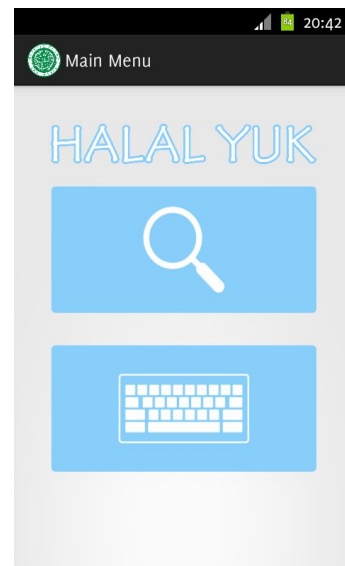
Gambar 16. Tahap *adaptive classifier* (Hanny Rindiani, Karlina Khyyarin Nisa).

Setelah proses *segmentation* selesai dilakukan, proses selanjutnya adalah *word recognition*. Proses ini

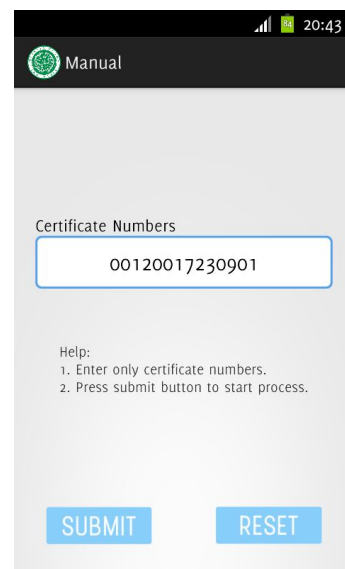
merupakan proses pengenalan bentuk karakter. Pada tahap ini terdapat proses *adaptive classifier*, tahap pengenalan elemen-elemen poligon yang dipecah menjadi bagian yang lebih pendek dengan panjang yang sama sehingga dimensi panjang dieliminasi dari *vektor* fitur. Beberapa fitur pendek dicocokkan dengan setiap fitur prototipikal dari *training*, hal ini membuat proses klasifikasi lebih kuat terhadap karakter yang terputus [10].

C. Implementasi

Pada tahap implementasi menghasilkan aplikasi yang dapat mengenali nomor sertifikat halal MUI berdasarkan citra. Tampilan aplikasi hasil implementasi dapat dilihat pada gambar-gambar dibawah ini:



Gambar 17. Hasil implementasi halaman main menu.



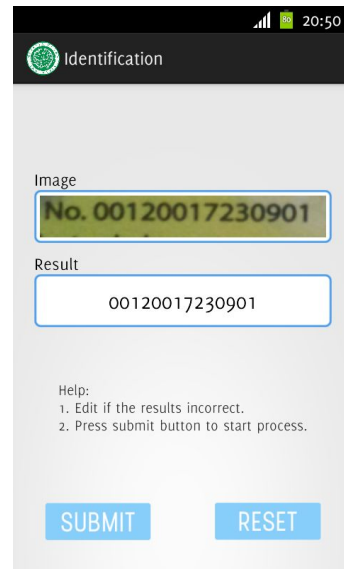
Gambar 18. Hasil implementasi halaman manual



Gambar 19. Hasil implementasi fungsi kamera.



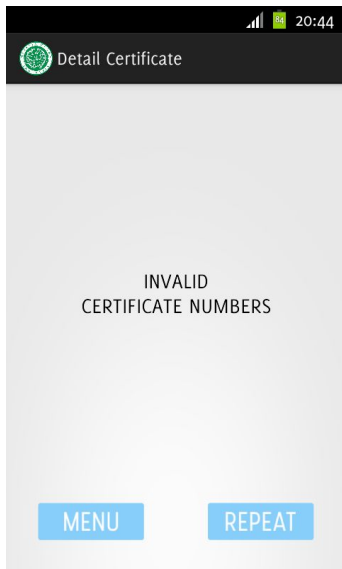
Gambar 20. Hasil implementasi fungsi crop.



Gambar 21. Hasil implementasi halaman identification.



Gambar 22. Hasil implementasi halaman output success.



Gambar 23. Hasil implementasi halaman output failure.

D. Pengujian

Berdasarkan hasil pengujian, menunjukkan bahwa seluruh fungsi yang ada pada aplikasi telah berjalan dengan baik. Berikut ini hasil pengujian yang telah dilakukan pada 15 sampel produk dengan bahan kemasan yang berbeda-beda. Aplikasi dikatakan berhasil bila tesseract dapat mengidentifikasi keseluruhan nomor sertifikat halal pada kemasan produk.

a. Kemasan Karton/Kardus

Sampel Nomor	Nomor Sertifikat	Percobaan	Citra	Terdeteksi	Hasil
1	00120017230901	1	00120017230901	00120017230901	SUKSES
1		2	00120017230901	00120017230901	SUKSES
1		3	00120017230901	00120017230901	SUKSES
1		4	00120017230901	00120017230901	SUKSES
1		5	00120017230901	00120017230901	SUKSES
2	01201130690615	1	01201130690615	01201130690615	SUKSES
2		2	01201130690615	01201130690615	SUKSES
2		3	01201130690615	01201130690615	SUKSES
2		4	01201130690615	01201130690615	SUKSES
2		5	01201130690615	01201130690615	SUKSES
3	00100014251200	1	00100014251200	00100014251200	SUKSES
3		2	00100014251200	00100014251200	SUKSES
3		3	00100014251200	00100014251200	SUKSES
3		4	00100014251200	00100014251200	SUKSES
3		5	00100014251200	00100014251200	SUKSES
4	00100007301297	1	00100007301297	-	GAGAL
4		2	00100007301297	-	GAGAL
4		3	00100007301297	-	GAGAL
4		4	00100007301297	00100007301291	GAGAL
4		5	00100007301297	-	GAGAL
5	00100014251200	1	00100014251200	00100014251200	SUKSES
5		2	00100014251200	00100014251200	SUKSES
5		3	00100014251200	00100014251200	SUKSES
5		4	00100014251200	00100014251200	SUKSES
5		5	00100014251200	00100014251200	SUKSES

b. Kemasan Kaleng

Sampel Nomor	Nomor Sertifikat	Percobaan	Citra	Terdeteksi	Hasil
1	00120065810713	1	00120065810713	00120056310113	GAGAL
1		2	00120065810713	-	GAGAL
1		3	00120065810713	0012006681011	GAGAL
1		4	00120065810713	-	GAGAL
1		5	00120065810713	001200	GAGAL
2	00030008060498	1	00030008060498	-	GAGAL
2		2	00030008060498	000300	GAGAL
2		3	00030008060498	-	GAGAL
2		4	00030008060498	-	GAGAL
2		5	00030008060498	-	GAGAL
3	00040021820902	1	00040021820902	-	GAGAL
3		2	00040021820902	-	GAGAL
3		3	00040021820902	-	GAGAL
3		4	00040021820902	-	GAGAL
3		5	00040021820902	00	GAGAL
4	00100024040303	1	00100024040303	00100024040303	SUKSES
4		2	00100024040303	00100024040303	SUKSES
4		3	00100024040303	00100024040303	SUKSES
4		4	00100024040303	00100024040303	SUKSES
4		5	00100024040303	00100024040303	SUKSES
5	00100014251200	1	00100014251200	00100074251200	GAGAL
5		2	00100014251200	00100014251200	SUKSES
5		3	00100014251200	00100014251200	SUKSES
5		4	00100014251200	001007425720	GAGAL
5		5	00100014251200	00500142012011	GAGAL

keakuratan tesseract sebesar 80%.

- Pada kemasan kaleng tingkat keakuratan tesseract sebesar 28%.
- Pada kemasan plastik tingkat keakuratan tesseract sebesar 40%.

c. Kemasan Plastik

Sempel Nomor	Nomor Sertifikat	Percobaan	Citra	Terdeteksi	Hasil
1	00100037591205	1	00100037591205	00100037591205	SUKSES
1		2	00100037591205	00100037591205	SUKSES
1		3	00100037591205	90190937591205	GAGAL
1		4	00100037591205	00100037591205	SUKSES
1		5	00100037591205	00100037591205	SUKSES
2	00100063511212	1	00100063511212	0010000351121	GAGAL
2		2	00100063511212	0010005353313	GAGAL
2		3	00100063511212	00100003511211	GAGAL
2		4	00100063511212	00100063511212	SUKSES
2		5	00100063511212	00100063511212	SUKSES
3	00100037591205	1	00100037591205	00100037591205	SUKSES
3		2	00100037591205	10919993259120	GAGAL
3		3	00100037591205	00100037591205	SUKSES
3		4	00100037591205	00109037591205	GAGAL
3		5	00100037591205	00100037591205	SUKSES
4	00140012700600	1	00140012700600	00135012706606	GAGAL
4		2	00140012700600	00145012700600	GAGAL
4		3	00140012700600	00145632766600	GAGAL
4		4	00140012700600	270	GAGAL
4		5	00140012700600	145011500600	GAGAL
5	00170069380614	1	00170069380614	00170069380614	SUKSES
5		2	00170069380614	007009360614	GAGAL
5		3	00170069380614	001360693806	GAGAL
5		4	00170069380614	36669380614	GAGAL
5		5	00170069380614	001700693614	GAGAL

4. KESIMPULAN DAN SARAN

A. Kesimpulan

Berikut kesimpulan berdasarkan pengujian tesseract OCR yang telah dilakukan:

1. Semakin kecil *font* nomor sertifikat halal yang terdapat pada kemasan produk, semakin kecil juga kemungkinan tesseract dapat mengidentifikasinya.
2. Bagus tidaknya kualitas kamera *smartphone* yang digunakan, mempengaruhi tingkat keberhasilan tesseract dalam mengidentifikasi karakter pada citra.
3. Aplikasi paling sesuai digunakan pada produk dengan kondisi *font* nomor sertifikat halal yang cukup besar dan tidak saling berimpitan.
4. Tesseract OCR kurang cocok bila diimplementasikan pada aplikasi pengenalan citra nomor sertifikat halal, karena nomor sertifikat halal pada kemasan produk yang beredar dipasaran saat ini sebagian besar menggunakan ukuran *font* yang sangat kecil.
5. Aplikasi yang telah dikembangkan menggunakan tesseract memiliki tingkat keakuratan yang berbeda-beda pada beberapa macam kemasan produk. Berikut tingkat keakuratan tesseract pada tiga macam kemasan produk berdasarkan hasil pengujian:

- Pada kemasan karton/kardus tingkat

B. Saran

Pada penelitian ini, citra yang digunakan masih bersifat *original* hasil pengambilan kamera *device* yang digunakan. Untuk penelitian selanjutnya, diharapkan dilakukan pengolahan citra dahulu sebelum citra tersebut diolah tesseract. Pengolahan-pengolahan citra yang dapat dilakukan agar hasil identifikasi tesseract menjadi lebih maksimal antara lain *rescaling*, *binarisation*, *noise removal*, *rotation/deskewing* dan *border removal*.

5. DAFTAR PUSTAKA

- [1] Rayhan, "MUI: Pemalsuan Logo Halal MUI Harus Diusut Tuntas," 19 Desember 2012. [Online]. Available: <https://www.islampos.com/mui-pemalsuan-logo-halal-mui-harus-diusut-tuntas-33527/>. [Diakses 17 Desember 2015].
- [2] E. Putri, "Pertumbuhan *smartphone* capai kenaikan 15 persen di 2014," 18 September 2014. [Online]. Available: <https://id.techinasia.com/smartphone-terus-meran-gkak-naik-di-2014/>. [Diakses 17 Desember 2015].
- [3] K. K. Wijaya, "Android dan browser Opera dominasi pengguna mobile Indonesia selama 2014," 6 Januari 2015. [Online]. Available: <https://id.techinasia.com/android-opera-dominasi-smartphone-indonesia-2014/>. [Diakses 17 Desember 2015].
- [4] N. Rellyani, "Penerapan Optical Character Recognition (OCR) Untuk Rancang Bangun Aplikasi Translator Pada Platform Android," 7 Mei 2012. [Online]. Available: <http://digilib.unila.ac.id/14761/4/BAB%20II.pdf>. [Diakses 23 Desember 2015].
- [5] E. S. Utami, "Pembacaan Plat Nomor Kendaraan Menggunakan Metode Jaringan Syaraf Tiruan (JST) Backpropagation Berbasis Image Processing," p. 5, 2009.
- [6] R. Smith, D. Antonova dan D.-S. Lee, "Adapting the Tesseract Open Source OCR Engine for Multilingual OCR," 25 Juli 2009. [Online]. Available: <http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/35248.pdf>. [Diakses 23 Desember 2015].
- [7] Rojali dan A. G. Salman, "Studi dan Implementasi Steganografi Citra JPEG

Menggunakan Metode Spread Spectrum Pada Perangkat Mobile Berbasis Android,” 2012. [Online]. Available: <http://library.binus.ac.id/eColls/eThesdoc/Bab2DOC/2012-1-00388-IF%20Bab2001.doc>. [Diakses 23 Desember 2015].

- [8] I. Sommerville, *Software Engineering*, Boston: Pearson Education, Inc, 2011, pp. 30-31.
- [9] A. Darmawan, “SQLite–Database Handal Tanpa Server,” 3 Agustus 2013. [Online]. Available: <http://ariefdarmawan.com/1006/sqlitedatabase-handal-tanpa-server>. [Diakses 29 Desember 2015].
- [10] H. Rindiani dan K. K. Nisa, “Aplikasi Android untuk Pengenalan Citra Karakter Jepang dengan Library Tesseract,” [Online]. Available: <http://apps.cs.ipb.ac.id/ojs/files/journals/3/articles/324/submission/review/324-854-1-RV.pdf>. [Diakses 23 Desember 2015].