

PERANCANGAN APLIKASI PENGOLAHAN CITRA TEKS ARAB DAN PENERJEMAHANNYA KE DALAM BAHASA INDONESIA MENGGUNAKAN SMARTPHONE ANDROID

Wendi Zarman¹, Sri Nurhayati², Heriyanto³,

^{1,2,3} Jurusan Teknik Komputer Unikom, Bandung

¹wendizar@gmail.com, ²serieid@yahoo.com, ³heriynt89@gmail.com

ABSTRAK

Keterbatasan seseorang dalam memahami suatu bahasa asing terkadang menjadi kendala manakala orang tersebut sedang bepergian jauh ke suatu negara, terutama negara yang menggunakan huruf pada tulisan yang tidak terlalu umum digunakan. Alat penerjemah konvensional yang ada saat ini seperti kamus, dinilai belum cukup efektif untuk mengatasi kendala tersebut, dikarenakan alasan kepraktisan. Untuk itu diperlukan suatu alternatif solusi untuk mengatasi kendala tersebut.

Dalam penelitian ini akan dirancang suatu aplikasi pada *smartphone* Android yang dapat menerjemahkan teks Arab dengan cara mengambil gambar teks tersebut, untuk kemudian diterjemahkan oleh program aplikasi. Dasar dari aplikasi yang dibangun ialah dengan melakukan proses pengolahan citra terhadap gambar teks Arab yang diambil untuk kemudian dilakukan proses pengenalan dan penerjemahan teks secara online. Model aplikasi dirancang menggunakan pendekatan objek dengan salah satu toolsnya adalah UML.

Kata kunci : Teks arab, Pengolahan citra, Android, UML

1. PENDAHULUAN

Arab Saudi merupakan salah satu dari sekian banyak negara di dunia yang paling sering dikunjungi oleh warga Indonesia. Negara Arab Saudi memiliki keunikan dari segi penggunaan huruf dalam berbagai bentuk tulisan, yang tidak umum dipakai di negara Indonesia. Hal tersebut mungkin menjadi kendala bagi warga Indonesia yang tidak mengerti bahasa Arab ketika sedang berkunjung kesana, khususnya ketika berada di lokasi yang tidak banyak dilengkapi dengan papan informasi jalan dalam tulisan yang lebih umum digunakan. Kondisi tersebut dapat menyebabkan seseorang tersesat dan juga bisa berdampak terhadap suatu pelanggaran yang berlaku di lokasi tersebut. Keberadaan alat penerjemah konvensional yang ada saat ini seperti kamus dinilai belum cukup efektif dikarenakan alasan kepraktisan.

Di sisi lain, keberadaan *smartphone* Android yang ada saat ini, dengan beragam fitur yang ditawarkan dan harga yang semakin terjangkau, menjadikan perangkat tersebut banyak diminati oleh warga Indonesia. Berangkat dari fakta tersebut muncul gagasan untuk merancang suatu aplikasi pada *smartphone* Android yang bisa mengenali teks Arab pada suatu citra dan menerjemahkan teks tersebut ke dalam bahasa

Indonesia. Tujuan dibangunnya aplikasi tersebut ialah agar warga Indonesia yang sedang berkunjung ke negara Arab dapat menggunakan perangkat Android miliknya sebagai alat bantu penerjemah teks Arab dari hasil foto.

2. TINJAUAN PUSTAKA

2.1. Teks Arab

Teks Arab ditulis secara *cursive* (bergaya tulisan tangan) dan terdiri dari 28 huruf yang kesemuanya merupakan konsonan. Banyak huruf Arab yang terlihat mirip, akan tetapi satu sama lain dibedakan oleh letak titik yang disebut *rasm*. Titik ini merupakan bagian integral suatu huruf Arab, yang membedakan antar huruf dan pengucapannya. Salah satu contohnya huruf ب (ba') dan ت (ta') yang memiliki bentuk dasar yang sama.

Masing-masing huruf Arab mempunyai berbagai bentuk huruf sesuai dengan posisinya (di depan, tengah, belakang atau terpisah). Penulisan huruf Arab dimulai dari kanan ke kiri secara horizontal, namun khusus untuk penulisan nomor dimulai dari kiri ke kanan.

2.2. Pengolahan Citra

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi

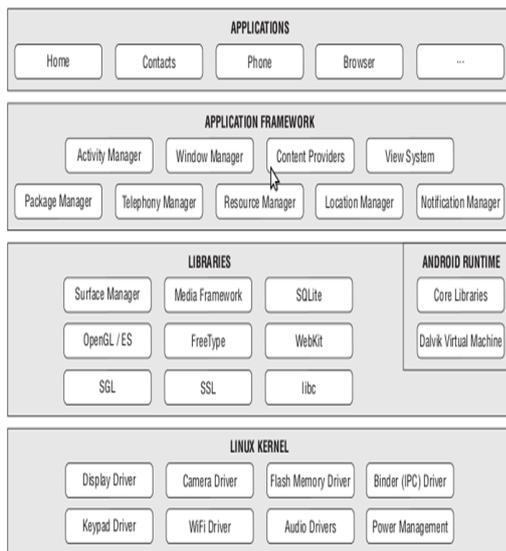
menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital pengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik (*array* yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu.

Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital.

2.3. Anroid

Android merupakan suatu *software stack* untuk *perangkat* mobile. Di dalamnya terdapat sistem operasi, *middleware*, dan *key application*. Aplikasi pada *platform* ini dikembangkan dalam bahasa pemrograman java.

Sistem operasi Android dibagi menjadi lima bagian dalam empat lapisan utama, seperti yang dapat di lihat pada gambar 2.1 di bawah ini.



Gambar 2.1 Arsitektur Android

2.4. OCR

Salah satu aplikasi yang penting dalam dunia pengolahan citra adalah pengenalan objek (*object* recognition). Aplikasi yang paling banyak dijumpai adalah OCR (*Optical Character Recognition*). Aplikasi OCR sering digunakan untuk mengidentifikasi citra huruf

untuk kemudian diubah ke dalam bentuk file tulisan. Aplikasi OCR juga digunakan di dunia industri seperti industri elektronik untuk mengenali label-label yang ada pada *circuit board*.

2.5. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak lainnya.

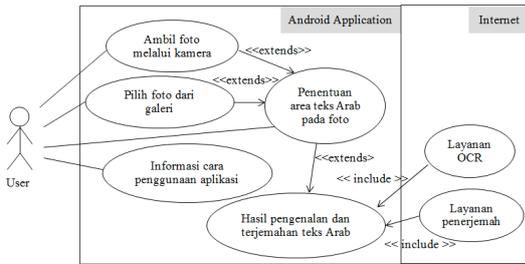
UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu[4]:

1. *Use Case Diagram*, untuk memodelkan proses bisnis.
2. *Conceptual Diagram*, untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram*, untuk memodelkan pengiriman pesan (*message*) antar objek.
4. *Collaboration Diagram*, untuk memodelkan interaksi antar objek.
5. *State Diagram*, untuk memodelkan perilaku objek di dalam sistem.
6. *Activity Diagram*, untuk memodelkan perilaku *Use Case* dan objek di dalam sistem.
7. *Class Diagram*, untuk memodelkan struktur kelas.
8. *Object Diagram*, untuk memodelkan struktur objek.
9. *Component Diagram*, untuk memodelkan komponen objek.
10. *Deployment Diagram*, untuk memodelkan distribusi aplikasi.

3. PERANCANGAN SISTEM

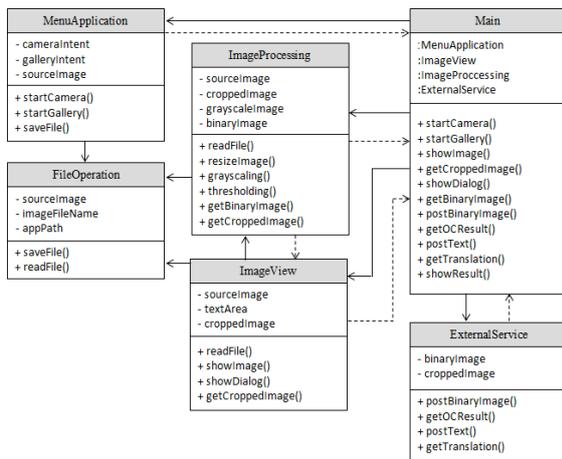
Pada sistem aplikasi yang dirancang terdapat lima buah *uses case* dan *user* sebagai aktor tunggal yang berperan dalam sistem.

Perancangan Aplikasi Pengolahan Citra Teks Arab Dan Penerjemahannya Ke Dalam Bahasa Indonesia Menggunakan *Smartphone* Android



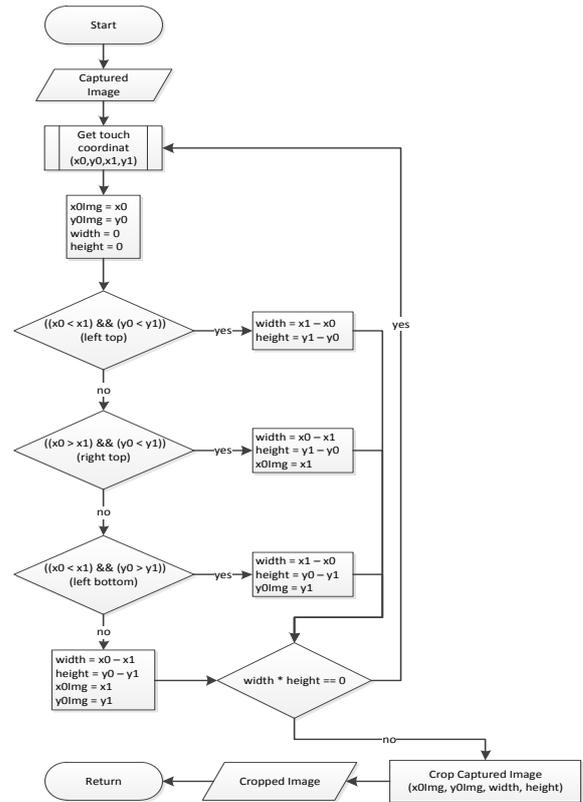
Gambar 3.1. Diagram Use Case

Penggambaran secara abstrak struktur dari aplikasi yang akan dibangun, *class-class* yang terlibat, serta hubungan antar *class* untuk saling berkomunikasi satu sama lain. Struktur *class* dari aplikasi yang akan dibangun dapat dilihat pada gambar 3.2.

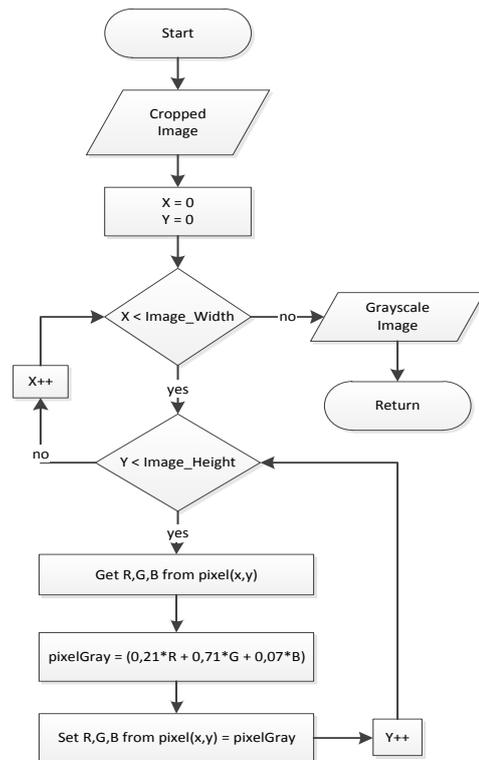


Gambar 3.5. Diagram Class

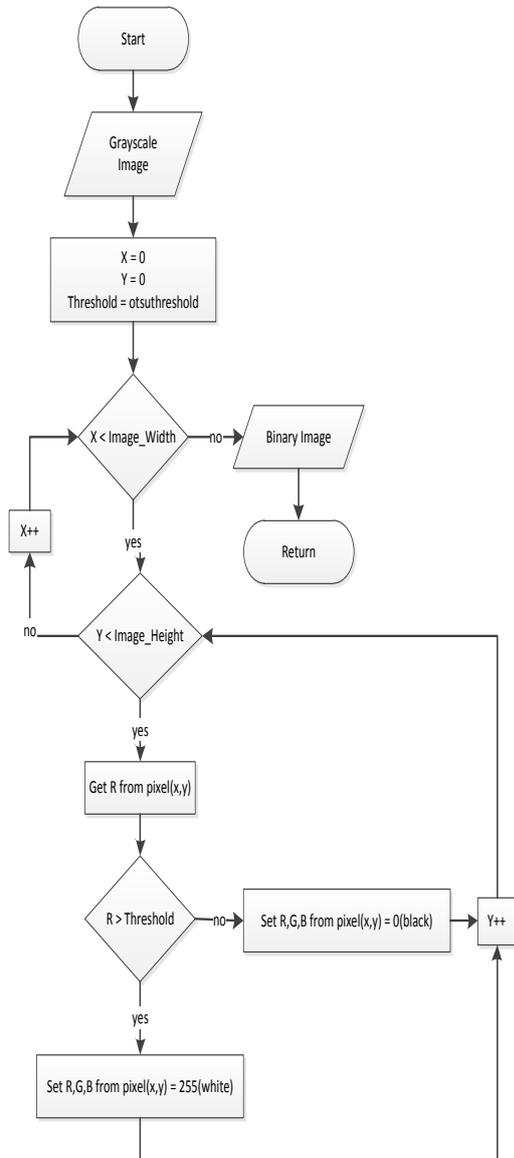
Rancangan prosedural dari aplikasi yang dibuat terdapat tiga buah proses secara berurut yaitu *cropping*, *grayscale* dan *thresholding*.



Gambar 3.2. Flowchart Cropping



Gambar 3.3. Flowchart Grayscale



Gambar 3.4. Flowchart Thresholding

4. PENGUJIAN

4.1. Pengujian Warna

Untuk pengujian warna dilakukan dengan menguji pada gambar yang berlatar belakang warna terang dapat dilihat pada gambar 4.1 dan gelap dapat dilihat pada gambar 4.2.



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan



(d) Hasil dari server OCR

Gambar 4.1 Hasil dari OCR dikenali dengan “Parkir Mobil”



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan

Tidak dikenali

(d) Hasil dari server OCR

Gambar 4.2 Hasil penerjemahan tidak dikenali

Dari gambar 4.1. dapat dilihat bahwa dengan gambar warna teks yang lebih terang dari latar belakang menunjukkan gambar biner yang dihasilkan lebih bersih dibandingkan hasil percobaan yang lain. Tingkat gangguan dari gambar biner yang dihasilkan sangat berpengaruh terhadap hasil pengenalan teks oleh server OCR. Semakin tinggi tingkat gangguan pada gambar biner, menyebabkan hasil pengenalan teks tidak dapat dikenali dengan baik bahkan menyebabkan teks tersebut sama sekali tidak dapat dikenali oleh server OCR. Proses binarisasi gambar dengan menggunakan metode Otsu sangat rentan terhadap intensitas perbedaan warna pada area gambar. Semakin kecil tingkat perbedaan warna pada area gambar menyebabkan proses pemisahan antara teks dan latarbelakang menjadi kurang akurat

4.2. Pengujian Teks Miring

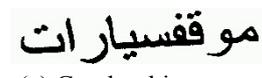
Pengujian dengan teks miring dapat dilihat pada gambar 4.3 dan 4.4.



(a) Hasil pengambilan gambar oleh kamera



(b) hasil pemotongan gambar secara manual



(c) Gambar biner yang dihasilkan

موقفسيارات

(d) Hasil dari server OCR

Gambar 4.3 Hasil pengujian teks dengan kemiringan wajar dikenali

Perancangan Aplikasi Pengolahan Citra Teks Arab Dan Penerjemahannya Ke Dalam Bahasa Indonesia Menggunakan *Smartphone* Android



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan

حممم

(d) Hasil dari server OCR

Gambar 4.4 Hasil pengujian teks dengan kemiringan wajar dikenali

Dari gambar 4.3 dan 4.4, diketahui bahwa server OCR masih mampu mengenali teks dengan kondisi gambar teks yang miring selama kemiringan gambar teks tersebut masih dalam kategori wajar.

4.3. Pengujian Posisi Pengambilan Gambar Teks

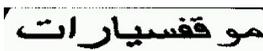
Pengujian pengambilan gambar teks berdasarkan posisi dilakukan dengan pengambilan foto dari posisi bawah, pengambilan foto dari posisi atas, pengambilan foto dari posisi samping kiri, dan pengambilan foto dari posisi samping kanan.



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan

موظسيارات

(d) Hasil dari server OCR

Gambar 4.5 Hasil pengujian pengambilan foto dari posisi bawah



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan

مزي سهار ابات

(d) Hasil dari server OCR

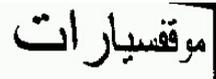
Gambar 4.6 Hasil pengujian pengambilan foto dari posisi atas



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan

موشياراتاة

(d) Hasil dari server OCR

Gambar 4.7 Hasil pengujian pengambilan foto dari posisi samping kiri



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan

ح؛

(d) Hasil dari server OCR

Gambar 4.8 Hasil pengujian pengambilan foto dari posisi samping kanan

Dari empat pengujian diatas, denna data yang sama terlihat bahwa posisi pengambilan gambar teks yang dilakukan pada masing-masing pengujian menyebabkan teks pada gambar tidak terdefinisi dengan baik sehingga pengujian diatas menghasilkan hasil pengenalan yang salah.

4.4. Pengujian Pada Sample Gambar Asli



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan



(d) Hasil dari server OCR



(e) Screenshot hasil tampilan akhir

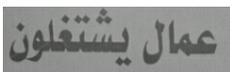
Gambar 4.9 Hasil pengujian sample gambar ke-1



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan



(d) Hasil dari server OCR

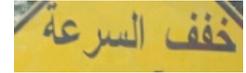


(e) Screenshot hasil tampilan akhir

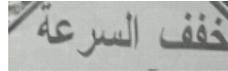
Gambar 4.10 Hasil pengujian sample gambar ke-2



(a) Hasil pemotongan gambar secara manual



(b) Gambar grayscale yang dihasilkan



(c) Gambar biner yang dihasilkan



(d) Hasil dari server OCR



(e) Screenshot hasil tampilan akhir

Gambar 4.11 Hasil pengujian sample gambar ke-3

Dari ketiga *sample* gambar yang diuji pada aplikasi, terdapat satu gambar yang tidak dapat dikenali oleh server OCR dengan baik yaitu *sample* gambar ke-3. Pada *sample* gambar ke-3 terlihat dari hasil pemotongan gambar teks terdapat garis hitam dengan tingkat warna yang sama dengan teks sehingga garis tersebut menjadi bagian dari teks yang akan dikenali oleh server OCR. Hal tersebut menyebabkan terjadinya kesalahan pada hasil pengenalan teks.

5. SIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil pengujian yang telah dilakukan, diperoleh kesimpulan bahwa program aplikasi yang dibuat dapat digunakan sebagai alat bantu untuk mengenali teks Arab, dengan beberapa kondisi sebagai berikut:

1. Gambar teks yang akan dikenali memiliki perbedaan warna yang jelas dengan latar belakang teks tersebut.
2. Posisi pengambilan gambar teks hanya bisa dilakukan dari arah depan untuk mendapatkan hasil pengenalan yang optimal.
3. Hasil pemotongan gambar harus dilakukan serapih mungkin, yang artinya bahwa teks tidak ikut terpotong pada saat proses penyeleksian dan potongan

Perancangan Aplikasi Pengolahan Citra Teks Arab Dan Penerjemahannya Ke Dalam Bahasa Indonesia Menggunakan *Smartphone* Android

gambar tersebut tidak terganggu oleh objek lain, selain teks yang ingin dikenali.

4. Jika pada saat proses pengambilan gambar ternyata menghasilkan teks yang miring, maka teks tersebut masih bisa dikenali selama masih dalam kategori kemiringan yang wajar.
5. Gambar teks yang yang ingin dikenali tidak terlalu kecil.

5.2. Saran

Beberapa saran untuk pengembangan selanjutnya, diantaranya:

1. Gambar teks yang akan dikenali memiliki perbedaan warna yang jelas dengan latar belakang teks tersebut.
2. Posisi pengambilan gambar teks hanya bisa dilakukan dari arah depan untuk mendapatkan hasil pengenalan yang optimal.
3. Hasil pemotongan gambar harus dilakukan serapih mungkin, yang artinya bahwa teks tidak ikut terpotong pada saat proses penyeleksian dan potongan gambar tersebut tidak terganggu oleh objek lain, selain teks yang ingin dikenali.
4. Jika pada proses pengambilan gambar ternyata menghasilkan teks yang miring, maka teks tersebut masih bisa dikenali selama masih dalam kategori kemiringan yang wajar.
5. Gambar teks yang yang ingin dikenali tidak terlalu kecil.

6. DAFTAR PUSTAKA

- [1] Putra, Darma. (2010). *Pengolahan Citra Digital*. Yogyakarta: Andi.
- [3] Lee, Wei-Meng. (2012). *Beginning Android 4 Application Development*. Idianapolis: John Wiley & Sons, inc
- [4] Nugroho Adi (2010). *Rekayasa Perangkat Lunak Menggunakan UML Dan Java*. Diakses tanggal 25 Juli 2012, dari <http://books.google.co.id/>
- [5] *Dokumentasi ABBYY Cloud OCR*. Diakses tanggal 15 Juli 2012, dari <http://ocrsdk.com/documentation/quick-start/>
- [6] *Microsoft Translator Java API*. Diakses tanggal 15 Juli 2012, dari <http://code.google.com/p/microsoft-translator-java-api/>