



Hak cipta dan penggunaan kembali:

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

Copyright and reuse:

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

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Berikut adalah beberapa penelitian terkait yang telah dilakukan oleh peneliti terdahulu.

A. Penelitian yang dilakukan oleh Ozlem Durmaz Incel dan Mustafa Incel dengan judul “*Etiket: System for tracking the contents of the packaged food products*” memiliki permasalahan bagaimana menyediakan informasi seputar produk makanan seperti konten makanan, informasi gizi, dan alergen yang terkandung dalam makanan kemasan tersebut. Penelitian ini dilakukan dengan menggunakan aplikasi untuk memindai *barcode* untuk mendeteksi produk makanan tersebut. Aplikasi *mobile* yang dibuat oleh peneliti ini memiliki langkah-langkah yang cukup rumit untuk mendapatkan hasil akhir[8]. Hasil dari penelitian ini menyatakan bahwa aplikasi yang dibuat memudahkan pemantauan konten produk makanan kemasan. Hasil penelitian ini akan penulis gunakan sebagai perbandingan dengan penelitian yang penulis dibuat.

B. Penelitian yang dilakukan oleh S. Ch, S. Mahna, N. Kashyap pada tahun 2015 dengan judul “*Optical Character Recognition on Handheld Devices*” memiliki permasalahan bagaimana mengimplementasi sistem OCR pada perangkat *mobile* untuk melakukan *scan* gambar. Sistem ini juga menyediakan fitur seperti *no-typing*, edit data mentah, terjemahan sekejap, dan utilisasi memori. Penelitian ini menghasilkan rata-rata akurasi OCR sekitar 93%[9]. Konstruksi OCR yang dijelaskan pada penelitian ini akan penulis gunakan dalam penelitian penulis.

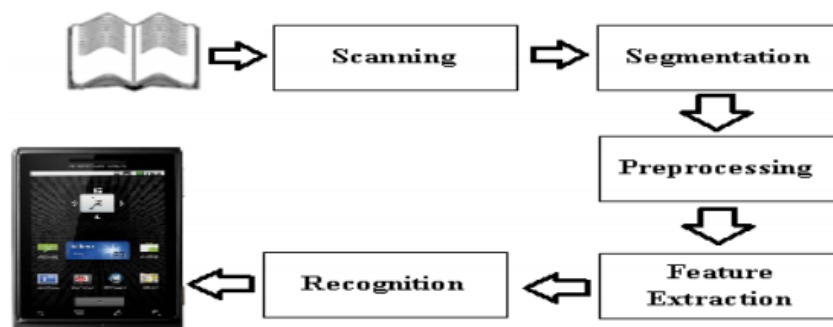
C. Penelitian yang dilakukan oleh P. Wankhede, S. Mohod pada tahun 2015 dengan judul “*A Method for Web Application Vulnerabilities Detection by Using Boyer-Moore String Matching Algorithm*” memiliki permasalahan bagaimana mendeteksi vulnerabilitas aplikasi web menggunakan algoritma Boyer-Moore *String Matching*. Penelitian ini menghasilkan rata-rata akurasi metode Boyer-Moore sebesar 100% dan metode ini dianggap memiliki performa yang baik dalam kemampuan mendeteksi vulnerabilitas secara akurat[10].

2.2 Algoritma Optical Character Recognition

Optical Character Recognition (OCR) merupakan proses menerjemahkan teks cetak, tulisan tangan, tulisan ketik, atau sebaliknya dari *hard copy* menjadi *soft copy* yang bisa dimanipulasi oleh komputer. Hal ini juga bisa disebut sebagai digitalisasi sebuah dokumen[11].

Kinerja dari OCR bergantung dengan kualiti dari dokumen yang diinput. OCR dirancang untuk memproses gambar yang mengandung hampir semuanya adalah teks, dengan sangat sedikit non-teks atau gambar yang didapat dari tangkapan (*capture*) kamera *mobile*[12].

Gambar 2.1 dibawah ini mengilustrasikan langkah-langkah OCR untuk merekognisi teks. Langkah-langkah tersebut adalah *scanning*, *segmentation*, *preprocessing*, *feature extraxtion*, dan *recognition*[12].



Gambar 2.1 Optical Character Recognition (OCR)

A. *Scanning*

Input gambar ke dalam OCR melalui langkah *scanning*. Gambar yang *input* bisa berupa tulisan tangan, *printed text* seperti buku, majalah, koran, dll. *Scanning* dilakukan menggunakan kamera *mobile* android. Langkah ini merupakan proses digitalisasi dokumen analog menjadi *gray scale* sehingga bisa menghemat penyimpanan memori dan usaha komputasional.

B. *Segmentation*

Pada tahap ini melakukan proses menemukan letak teks. Segmentasi akan membedakan teks dari gambar dan grafik. Namun, sistem OCR masih sering kesulitan pada tahap ini ketika karakter terhubung dengan grafik.

C. *Preprocessing*

Tahap ini akan memperbaiki masalah yang ditimbulkan ketika proses *scanning* dimana bisa terdapat *noise* sehingga menyebabkan sulit mengenali karakter. Tahap ini terdiri dari melakukan *smoothing* dan normalisasi. *Smoothing* memiliki beberapa peraturan yang akan diterapkan ke konten gambar dengan bantuan teknik *filling and thinning* sedangkan normalisasi bertanggung jawab untuk menangani ukuran, kemiringan, dan rotasi dari karakter.

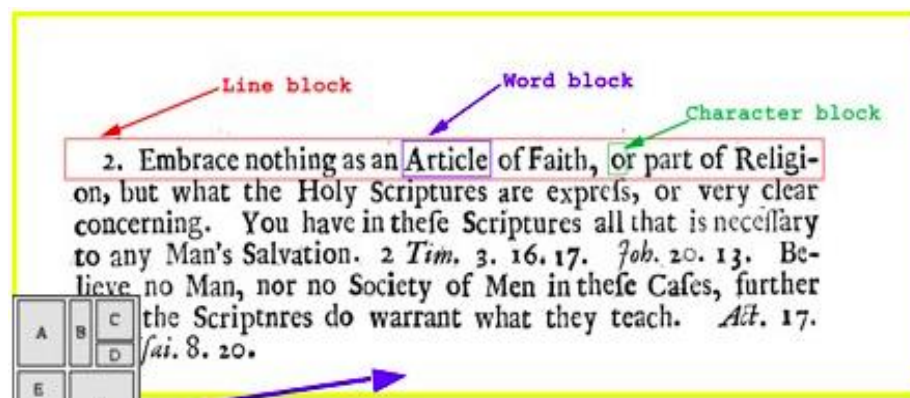
D. *Feature Extraction*

Tahap ini akan mengekstrak fitur (karakteristik) dari simbol. Simbol akan dikarakteristik lalu atribut yang tidak penting akan ditinggal.

E. *Recognition*

Karakter akan diidentifikasi dan kata demi kata akan direkognisi. Rekognisi akan mengekstrak teks dari gambar dokumen.

Di bawah ini merupakan identifikasi kata dan blok karakter. Gambar 2.2 menunjukkan ilustrasi pengidentifikasian. OCR akan bekerja dengan proses *layout analysis* untuk menemukan blok teks, kalimat/blok *line*, blok kata, dan blok karakter. Blok karakter dan kata selanjutnya akan dipecah menjadi beberapa bagian komponen lalu dibandingkan dengan berbagai font dan bahasa pada kamus karakter milik OCR *engine*. Ketika telah terjadi kecocokan semua karakter pada suatu blok kata, selanjutnya adalah membandingkan kata tersebut dengan kamus kata lengkap yang tersedia untuk bahasa tersebut[13].



Gambar 2.2 Identifikasi Kata dan Blok Karakter.

2.3 Algoritma Boyer Moore

Boyer Moore merupakan algoritma pencocokan kata atau biasa disebut *string matching* yang dikembangkan oleh Bob Boyer dan J.S. Moore pada tahun 1977. *String matching* merupakan suatu proses pencarian semua kemunculan *query* yang selanjutnya disebut dengan *pattern* ke dalam *string* untuk kesesuaian kebutuhan informasi yang dibutuhkan[14].

Metode Boyer Moore memiliki cara kerja yaitu dengan mencocokkan karakter dari sebelah kanan *pattern* sehingga pencariannya lebih cepat[14]. Dengan melakukan pencocokan karakter dari sebelah kanan, maka akan lebih banyak informasi yang didapatkan sehingga kita dapat melompati (*skip*) karakter

yang mungkin tidak cocok. *String matching* dilakukan dengan mencocokkan kata yang telah direkognisi oleh OCR dengan kata yang ada di *database*.

Pada tahap ini yang akan dilakukan adalah mencocokkan *string* dengan menggunakan algoritma Boyer Moore. Adapun cara kerja algoritma ini akan dijelaskan dengan contoh berikut ini[14].

Teks (S) : PENGOLAHAN CITRA DIGITAL

Pattern (P) : CITRA

Tahapan pencarian *pattern* (P) dalam Teks (S):

Tabel 2.1 Analisa Penentuan Nilai OH

Pattern (P)	C	I	T	R	A
<i>Occurrence Heuristic</i> (OH)	4	3	2	1	0

Langkah-Langkah:

a. Pada pergeseran pertama karakter “A” pada pattern tidak cocok dengan karakter “O” pada teks, maka pergeseran selanjutnya berdasarkan nilai dari tabel OH. Pada tabel OH karakter O tidak terdapat pada tabel, sehingga pergeseran selanjutnya adalah sebanyak jumlah karakter “A” pada pattern yaitu 5. Tabel 2.2 menunjukkan langkah ke-1.

Tabel 2.2 Langkah ke-1 Cara Kerja Algoritma Boyer Moore

Langkah Ke-1																								
Posisi Teks (S):	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Teks (S):	P	E	N	G	O	L	A	H	A	N		C	I	T	R	A		D	I	G	I	T	A	L
Pattern (P):	C	I	T	R	A																			

b. Pada pergeseran Ke-2 karakter “A” pada pattern tidak cocok dengan karakter “N” pada teks, maka pergeseran selanjutnya berdasarkan nilai dari tabel OH. Pada tabel OH karakter N tidak terdapat pada tabel, sehingga pergeseran selanjutnya adalah sebanyak jumlah karakter “A” pada pattern yaitu 5. Tabel 2.3 menunjukkan langkah ke-2.

Tabel 2.3 Langkah ke-2 Cara Kerja Algoritma Boyer Moore

Langkah Ke -2																								
Posisi Teks (S):	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Teks (S):	P	E	N	G	O	L	A	H	A	N		C	I	T	R	A		D	I	G	I	T	A	L
Pattern (P):						C	I	T	R	A														

- c. Pada pergeseran Ke-3 karakter “A” pada pattern tidak cocok dengan karakter “R” pada teks, maka pergeseran selanjutnya berdasarkan nilai dari tabel OH. Pada tabel OH karakter R terdapat pada tabel, sehingga pergeseran selanjutnya adalah sebanyak jumlah karakter “R” pada tabel OH pada yaitu 1. Tabel 2.4 menunjukan langkah ke-3.

Tabel 2.4 Langkah ke-3 Cara Kerja Algoritma Boyer Moore

Langkah Ke -3																								
Posisi Teks (S):	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Teks (S):	P	E	N	G	O	L	A	H	A	N		C	I	T	R	A		D	I	G	I	T	A	L
Pattern (P):											C	I	T	R	A									

- d. Pada pergeseran Ke-4 karakter “A” pada pattern cocok dengan karakter “A” pada teks, maka pergeseran selanjutnya dimundurkan satu langkah. Tabel 2.5 menunjukan langkah ke-4.

Tabel 2.5 Langkah ke-4 Cara Kerja Algoritma Boyer Moore

Langkah Ke -4																								
Posisi Teks (S):	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Teks (S):	P	E	N	G	O	L	A	H	A	N		C	I	T	R	A		D	I	G	I	T	A	L
Pattern (P):											C	I	T	R	A									

- e. Pada pergeseran selanjutnya dilakukan sampai pada pergeseran ke-12 karakter “C” pada pattern dengan karakter “C” pada text cocok.

2.4 Android

Android adalah sistem operasi berbasis *mobile* yang *open source* dan didesain terutama untuk perangkat *mobile* yang menggunakan sistem layar sentuh seperti *smartphone* dan *tablet*. Teknologi Android dikembangkan secara kontinu dan semakin banyak digunakan pada berbagai perangkat. Android dibuat dengan

menggunakan *software development kit* (SDK) spesial yang tersedia secara gratis untuk diunduh dari internet serta mampu untuk bekerja di berbagai macam sistem operasi. Seluruh *framework* aplikasi juga bisa digunakan kembali dengan komponen tertentu. Hal ini yang menjadi salah satu kelebihan Android. [15]

Pada penelitian ini akan menggunakan Android dengan versi sistem operasi Nougat 7.0 untuk menguji aplikasi rekognisi alergen. Penelitian memilih Android versi Nougat karena merupakan sistem operasi yang paling banyak digunakan dengan 30.8% pengguna Android menggunakan Nougat. Android Nougat memiliki beberapa kelebihan seperti mendukung *split screen*, *data saver*, tampilan notifikasi baru, baterai yang lebih irit, mode malam, dan menggunakan bahasa pemrograman Java 8[16].

Penelitian ini menggunakan Android Studio sebagai IDE untuk mengembangkan aplikasi android. yang menggunakan versi Android Studio 3.5.3 dengan SDK Tools 26.1.1 dan *platform* versi API 28:Android 9.0 (Pie) *revision* 6.