

## **BAB 2**

### **LANDASAN TEORI**

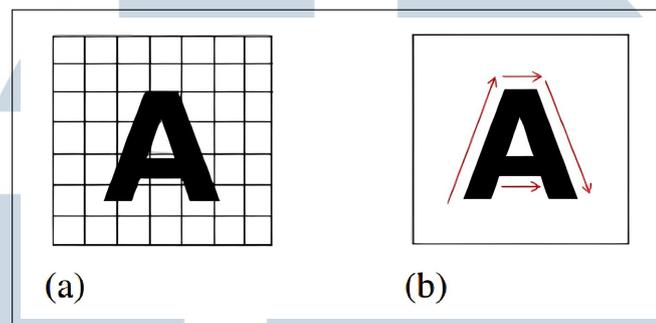
#### **2.1 Tanggal Kadaluwarsa**

Sesuai dengan Peraturan Pemerintah No.69 Tahun 1999 tentang Label dan Iklan Pangan pada Pasal 3 ayat (2) yang menerangkan bahwa keterangan label pangan sekurang-kurangnya memuat tanggal, bulan, dan tahun kadaluwarsa. Pengertian dari tanggal, bulan, dan tahun kadaluwarsa (tanggal kadaluwarsa) ini sebenarnya mengalami perubahan, sebelumnya dalam Peraturan Menteri Kesehatan No.346/MEN.KES/PER/IX/1983, pengertian dari tanggal kadaluwarsa adalah batas waktu akhir suatu makanan dapat dikonsumsi (*expired date*), kemudian setelahnya dalam Peraturan Menteri Kesehatan No.180/MEN.KES/PER/IV/1985, pengertian dari tanggal kadaluwarsa adalah batas waktu akhir suatu makanan dijamin mutunya sepanjang penyimpanannya mengikuti petunjuk yang diberikan oleh produsen (*best before*). Dengan kata lain, makanan yang telah melewati *expired date* sudah tidak layak untuk dikonsumsi, sedangkan makanan yang telah melewati *best before* masih layak untuk dikonsumsi tetapi mutu dari makanan tersebut telah mengalami penurunan [8]. Meskipun dikatakan bahwa makanan yang telah melewati batas tanggal *best before* masih layak untuk dikonsumsi hanya saja mengalami penurunan mutu. Akan tetapi, apabila konsumen tidak menyimpan makanan tersebut di tempat yang dianjurkan oleh produsen maka akan membuat makanan tersebut mengalami penurunan mutu yang drastis hingga makanan tersebut tidak layak untuk dikonsumsi lagi [15]. Sehingga tanggal kadaluwarsa merupakan hal yang penting untuk diperhatikan oleh konsumen ketika membeli atau menyimpan produk.

#### **2.2 Optical Character Recognition (OCR)**

*Optical Character Recognition* (OCR) merupakan sebuah aplikasi komputer yang dimanfaatkan untuk melakukan pengidentifikasian terhadap citra (gambar) tulisan berupa huruf atau angka yang selanjutnya dikonversikan ke dalam bentuk file tulisan. OCR memiliki prinsip dasar untuk meniru bagaimana cara manusia dapat menafsirkan teks dari suatu objek, dimulai dari melihat dan membaca objek yang mengandung teks kemudian memproses dan menginterpretasikan teks yang ada pada objek tersebut. Selanjutnya teks tersebut akan disimpan ke dalam bentuk digital [3].

OCR memiliki peranan dalam mengenali sebuah tulisan pada sebuah gambar dan dapat merubahnya ke dalam bentuk ASCII. Sesuai dengan Gambar 2.1 di bawah, terdapat dua macam *character recognition* yang akan dijelaskan sebagai berikut [2].



Gambar 2.1. (a) *Offline Character Recognition* (b) *Online Character Recognition*

Sumber: [16]

#### 1. *Offline Character Recognition*

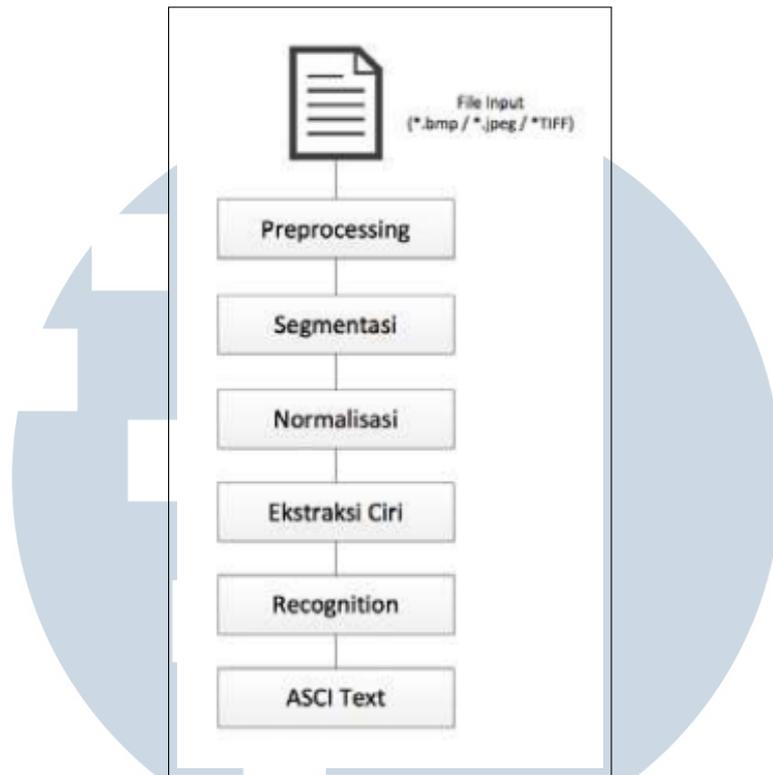
*Offline Character Recognition* merupakan proses pembacaan karakter yang dimulai dengan menyimpan dan mengolah tangkapan (*capture*) gambar terlebih dahulu sebelum menampilkan hasil dari pembacaan tersebut (teks).

#### 2. *Online Character Recognition*

*Online Character Recognition* merupakan proses pembacaan karakter yang langsung melakukan pembacaan terhadap gambar dan menampilkan hasilnya (teks) secara langsung selama gambar dalam posisi tangkapan (*capture*).

Secara umum, OCR dapat melakukan proses pembacaan karakter dari suatu objek gambar melalui proses yang terdapat pada Gambar 2.2.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.2. Diagram Proses OCR

Sumber: [17]

Dari gambar diatas, dapat dilihat bahwa untuk menghasilkan suatu teks mentah oleh OCR, tiap karakter yang ada pada citra objek tersebut harus melewati serangkaian proses yang akan dijelaskan sebagai berikut [2–4].

1. *Preprocessing*

*Preprocessing* merupakan tahapan awal dari proses penerjemahan OCR. Proses ini dilakukan untuk menghilangkan bagian-bagian yang tidak diperlukan pada objek gambar untuk proses selanjutnya.

2. Segmentasi

Segmentasi merupakan proses untuk memisahkan area pengamatan pada setiap karakter yang terdeteksi pada objek gambar. Proses ini akan memisahkan antara latar dan masing-masing karakter yang terdeteksi.

3. Normalisasi

Normalisasi merupakan proses untuk mengubah dimensi dan ketebalan pada setiap karakter yang terdeteksi.

#### 4. Ekstraksi Ciri (*Feature Extraction*)

Ekstraksi ciri atau *feature extraction* merupakan proses untuk mendapatkan ciri-ciri tertentu dari setiap karakter yang terdeteksi.

#### 5. *Recognition*

*Recognition* merupakan tahapan akhir dari proses penerjemahan OCR. Proses ini dilakukan untuk dapat mengenali setiap karakter yang tengah diamati (terdeteksi) tersebut dengan menggunakan cara membandingkan ciri-ciri yang telah didapatkan sebelumnya melalui proses ekstraksi ciri dengan ciri-ciri karakter yang terdapat pada basis data.

Salah satu metode pengenalan pada tahap *recognition* adalah metode *Template Matching Correlation*. Metode ini memiliki rata-rata tingkat keberhasilan terhadap pengenalan karakter sebesar 92,90% yang dimana tingkat keberhasilan sebesar itu sudah cukup untuk membangun sebuah sistem OCR yang baik. Metode ini dilakukan dengan mencari dan menemukan bagian-bagian (ciri) dari karakter yang terdeteksi pada objek gambar yang sesuai dengan referensi (basis data). Proses yang terjadi pada metode ini akan diawali dengan dimasukkannya objek gambar yang kemudian akan mengalami proses *preprocessing* meliputi binerisasi (*grayscale*, merupakan proses untuk mengubah warna gambar yang awalnya RGB menjadi hitam putih), segmentasi, dan normalisasi terlebih dahulu. Setelahnya baru akan diproses pada proses pengenalan dan pencocokan karakter antara karakter yang terdeteksi dengan karakter yang terdapat pada basis data [2, 3].

### 2.3 Flutter

Flutter merupakan *framework* yang dikembangkan oleh Google bersifat *open source* untuk membuat aplikasi yang bersifat *multi-platform* hanya dengan satu basis kode [18]. Dengan kata lain, dengan penggunaan Flutter, pengembang cukup hanya perlu melakukan pembangunan aplikasi dengan satu basis kode yang sama (Dart) tetapi aplikasi yang dihasilkan dapat digunakan di berbagai platform seperti web, android, maupun iOS. Flutter terbagi menjadi dua komponen sebagai berikut [19].

#### 1. SDK (*Software Development Kit*)

SDK (*Software Development Kit*) merupakan sekumpulan alat yang digunakan untuk dapat membuat aplikasi berjalan di berbagai platform.

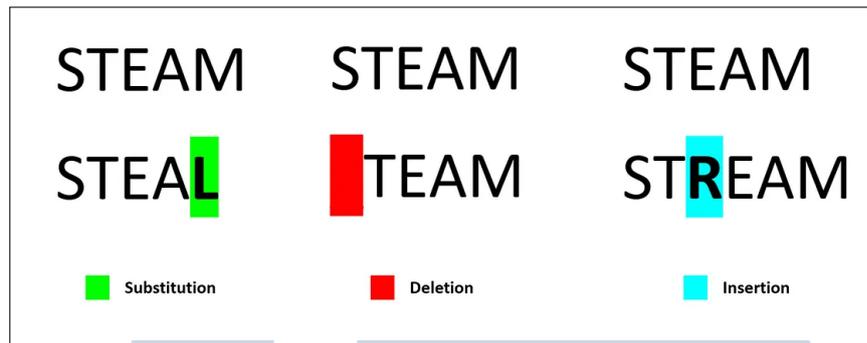
## 2. *Framework* UI

*Framework* UI merupakan komponen UI atau perlengkapan yang dapat digunakan dalam membangun aplikasi dan dapat dikustomisasi sesuai dengan keperluan pengembangan.

Sesuai dengan sifat yang dimiliki oleh Flutter yaitu *open source*, Flutter menyediakan sarana untuk para pengembang dapat berkontribusi dalam pembuatan *package*. Setiap pengembang memiliki kesempatan untuk membuat dan mengunggah *package* yang telah dibuatnya ke situs resmi milik Flutter (pub.dev). *Package* bertujuan untuk mempercepat proses pengembangan aplikasi sehingga pada saat pembuatan suatu proyek tidak perlu membuat semua *widget* atau fitur dari awal (*scratch*). *Package* yang telah diunggah oleh para pengembang dapat digunakan oleh pengembang lainnya dengan cara menginstalkannya ke dalam proyek Flutter yang ingin digunakan melalui file (pubspec.yaml). *Package* berisi sekumpulan *widget* baru yang memiliki fitur-fitur yang sebelumnya belum ada pada *widget* bawaan dari Flutter. Contohnya seperti *package image\_picker* yang dapat digunakan untuk mengakses kamera dan galeri foto pada *smartphone* ataupun *package pin\_code\_field* yang merupakan *widget* untuk membuat sebuah fitur *pin code* yang biasanya digunakan pada saat melakukan verifikasi OTP [20].

### 2.4 **Levenshtein Distance**

Algoritma Levenshtein *Distance* ditemukan oleh Vladimir Levenshtein pada tahun 1965. Algoritma ini merupakan algoritma untuk mengukur jumlah perbedaan antara dua buah *string*. Algoritma ini akan melakukan perhitungan jumlah minimum dari upaya transformasi suatu kata (*string*) menjadi kata (*string*) lain. Transformasi ini meliputi *insertions* (penyisipan), *deletions* (penghapusan), dan *substitutions* (penggantian) yang dapat dilihat pada Gambar 2.3 di bawah ini [21–23].



Gambar 2.3. Transformasi Algoritma Levenshtein *Distance*

Sumber: [21]

Berdasarkan Gambar 2.3, terdapat 3 (tiga) buah transformasi pada algoritma ini yang akan dijelaskan sebagai berikut.

1. *Insertion* (Penyisipan)

*Insertion* merupakan penambahan karakter pada suatu kata (*string*). Contohnya dapat dilihat pada Gambar 2.3 yang mengubah kata "STEAM" menjadi "STREAM" dengan penambahan karakter "R" diantara karakter "T" dan "E".

2. *Deletion* (Penghapusan)

*Deletion* merupakan penghapusan karakter pada suatu kata (*string*). Contohnya dapat dilihat pada Gambar 2.3 yang mengubah kata "STEAM" menjadi "TEAM" dengan penghapusan karakter "S" yang terdapat pada karakter pertama kata.

3. *Substitution* (Penggantian)

*Substitution* merupakan penggantian karakter pada suatu kata (*string*). Contohnya dapat dilihat pada Gambar 2.3 yang mengubah kata "STEAM" menjadi "STEAL" dengan penggantian karakter "M" yang terdapat pada karakter terakhir kata menjadi karakter "L".

## 2.5 Character Error Rate (CER)

Dalam perhitungan keakuratan OCR, umumnya menggunakan perhitungan *Character Error Rate* (CER) [24]. Perhitungan CER didasari oleh konsep *Levenshtein Distance*, yang menghitung jumlah minimum karakter dari transformasi suatu kata (*string*) *ground truth* (kata awal/referensi) dengan kata

(*string*) hasil keluaran (*output*) OCR [21]. Rumus perhitungan CER dapat dilihat sebagai berikut.

$$CER = \frac{S+D+I}{N} \quad (2.1)$$

Dengan keterangan:

- S = jumlah *Substitutions*
- D = jumlah *Deletions*
- I = jumlah *Insertions*
- N = jumlah karakter pada *ground truth* (kata referensi)

Dan untuk mendapatkan persentase dari CER maka hasil dari perhitungan CER dapat dikalikan dengan 100%. Ada kalanya hasil persentase CER melebihi 100%, terutama untuk perhitungan yang memiliki nilai *insertion* yang tinggi. Oleh karena itu, dapat dilakukan normalisasi dengan menggunakan rumus perhitungan CER sebagai berikut.

$$CER_{normalized} = \frac{S+D+I}{S+D+I+C} \quad (2.2)$$

Dengan keterangan:

- S = jumlah *Substitutions*
- D = jumlah *Deletions*
- I = jumlah *Insertions*
- C = jumlah karakter yang benar

U M N  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA