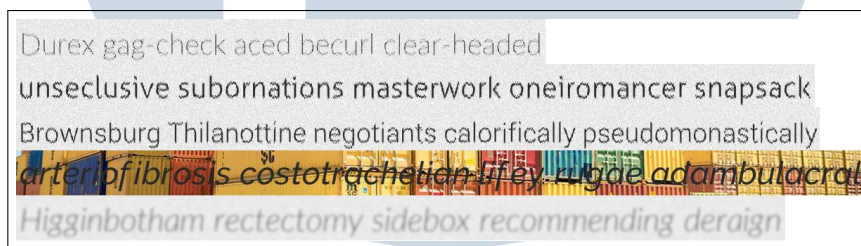


## BAB 3 METODOLOGI PENELITIAN

### 3.1 Pengumpulan Dataset

Dataset yang dikumpulkan akan digunakan untuk melatih model EasyOCR. Dataset akan dibuat secara otomatis oleh TextRecognitionDataGenerator yang dapat dikustomisasikan untuk menghasilkan data dengan kualitas yang bervariasi. TextRecognitionDataGenerator dapat di-install menggunakan pip, setelah itu dapat dijalankan melalui Command Prompt dengan *command* "trdg". Jumlah dan tingkat kustomisasi dapat ditentukan melalui *command* "trdg" dan daftar parameternya terdapat di dokumentasi resmi TextRecognitionDataGenerator.



Gambar 3.1. Contoh data yang dihasilkan TextRecognitionDataGenerator

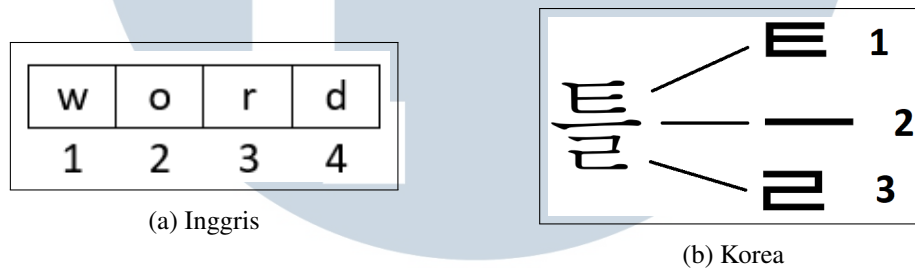
### 3.2 Train Model

Setelah dataset untuk pelatihan sudah dikumpulkan, pelatihan model dapat dimulai. Model akan dilatih menggunakan *script* yang telah disediakan oleh EasyOCR yang merupakan modifikasi dari *repository* deep-text-recognition-benchmark.

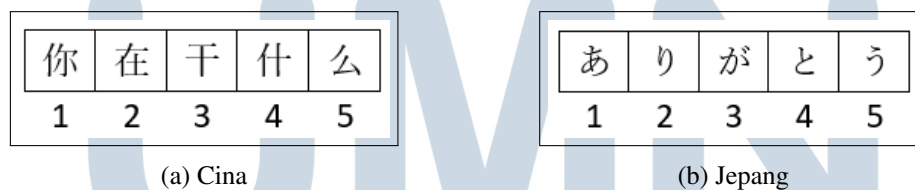
### 3.3 Evaluasi Model

Evaluasi model akan menggunakan metode CER dengan dataset berupa huruf A-Z, a-z, dan angka 0-9 dengan 5 font serif dan 5 font sans-serif dan memiliki variasi polos, italic, bold, dan bold-italic. Untuk bahasa yang menggunakan karakter unik seperti Cina, Jepang (kanji), dan Korea tetap akan dilakukan evaluasi dengan metode CER, tetapi hasil yang didapat tidak mencakup semua karakter yang ada sehingga nilai CER dapat turun naik tergantung kesulitan deteksi karakter tersebut. Kondisi yang dapat menyebabkan kesalahan dalam deteksi tulisan akan dibahas lebih lanjut di Bab 4.

Metode untuk mengevaluasi nilai CER dari sebuah model adalah pertama menghitung jumlah kesalahan yang muncul dan jumlah huruf di gambar awal (*ground truth*). Untuk menghitung nilai CER, kita harus menghitung jumlah perubahan yang perlu dilakukan untuk mengubah tulisan output menjadi tulisan *ground truth*. Untuk metode perhitungannya akan sedikit berbeda untuk empat bahasa yang dipilih karena perbedaan sistem bahasanya, dimana Inggris dan Korea menggunakan sistem huruf yang digabung menjadi sebuah kata, Cina dan Jepang menggunakan sistem karakter yang setiap karakternya memiliki artinya sendiri. Untuk bahasa Inggris dan Korea, pertama setiap katanya akan dipecah dan dihitung huruf-huruf yang menyusun kata tersebut (lihat Gambar 3.2). Untuk bahasa Cina dan Jepang, kalimatnya akan dipecah dan dihitung karakter yang menyusun kalimat tersebut (lihat Gambar 3.3).

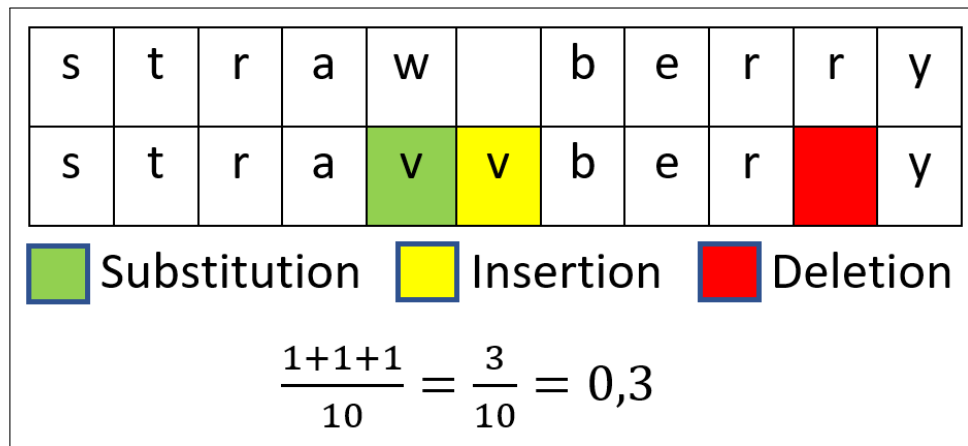


Gambar 3.2. Contoh pemecahan kata bahasa Inggris dan Korea



Gambar 3.3. Contoh pemecahan kalimat bahasa Jepang dan Cina

Contoh perhitungan nilai CER bisa dilihat di ..., dimana terjadi satu *substitution*, satu *deletion*, dan satu *insertion*. Dengan total tiga kesalahan dan jumlah huruf *ground truth* sebanyak 10, menggunakan rumus CER yang telah dibahas di bab 2, nilai CER yang didapatkan adalah 0,3 atau 30% akurasi yang berarti setiap 100 huruf terdapat 30 kesalahan.



Gambar 3.4. Contoh perhitungan nilai CER

### 3.4 Perancangan Backend

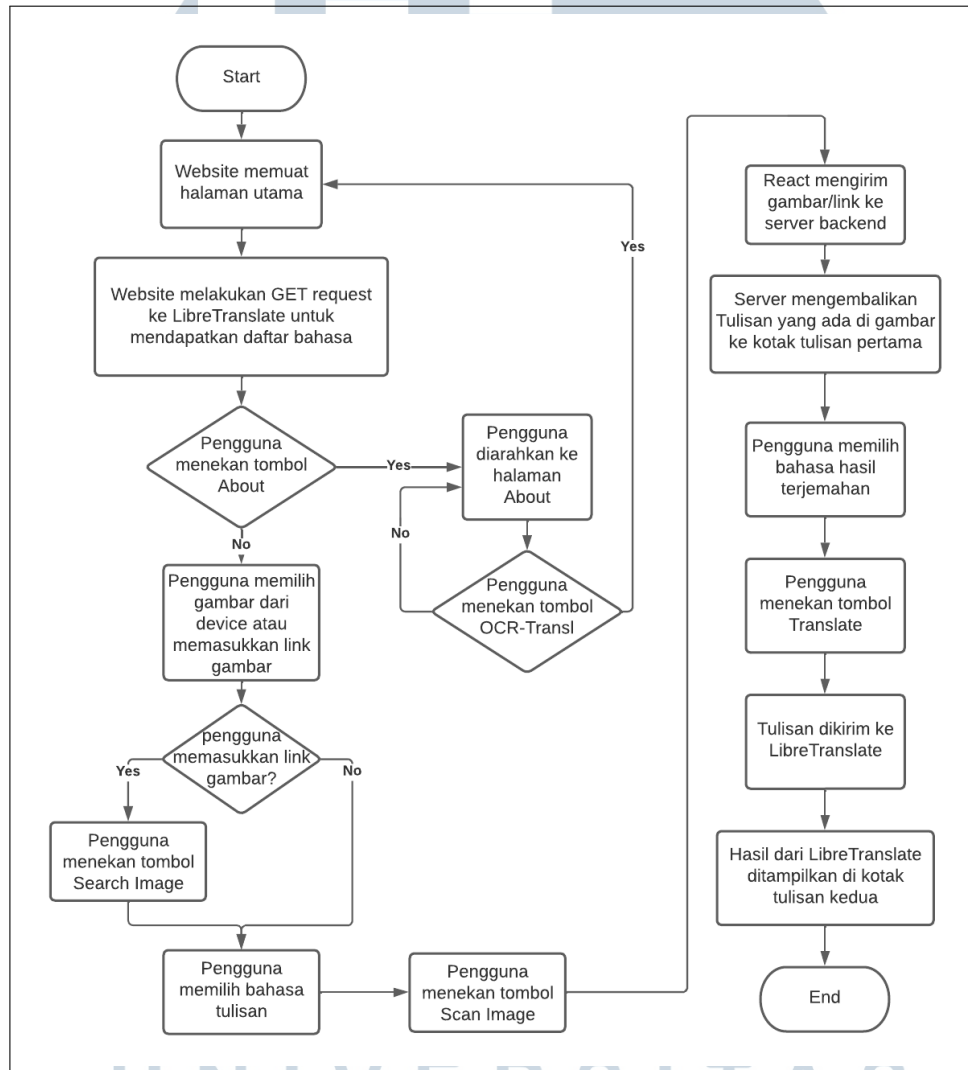
Tampilan website akan dirancang menggunakan Javascript library React untuk membuat single-page application. Untuk backend yang akan memproses gambar, Python dapat digunakan untuk memuat kode yang menjalankan model EasyOCR. Frontend dan backend lalu akan dihubungkan dengan Flask untuk mengubah backend Python menjadi sebuah server yang dapat menerima *request* dari frontend. Penerjemahan tulisan yang telah di-ekstrak dari gambar dapat dilakukan dengan membuat sebuah request ke *open-source* API bernama LibreTranslate. Supaya React dapat melakukan *request* ke server dan API, di-install Axios untuk menangani *request* yang dikirim.

### 3.5 Perancangan Frontend

#### 3.5.1 Flowchart Website

Berikut adalah alur kerja website yang dirancang dari saat dibuka pengguna hingga tulisan telah diterjemahkan. Pertama pengguna langsung diarahkan ke halaman utama yang terdapat header, body, dan footer. Header terdapat nama website yang ketika diklik akan mengarahkan user ke halaman utama dan menu About yang mengarahkan user ke halaman About ketika diklik. Footer terdapat link untuk dua *resource* yang paling penting dalam perancangan fitur website yaitu EasyOCR dan LibreTranslate. Di halaman utama, user dapat meng-*upload* gambar dari device mereka atau *copy and paste* link yang merujuk ke gambar di internet. Setelah memilih gambar, user akan diminta untuk memilih bahasa yang terdapat di gambar

tersebut supaya server dapat memilih link mana yang akan dipakai. Setelah user mengklik tombol "scan image", gambar yang telah dipilih akan dikirim ke server dan server akan mengembalikan output berupa tulisan yang terdapat di gambar. Tulisan yang didapat bisa di *copy and paste* ke aplikasi penerjemah lain atau menggunakan LibreTranslate yang ditambahkan di website sebagai fitur tambahan.

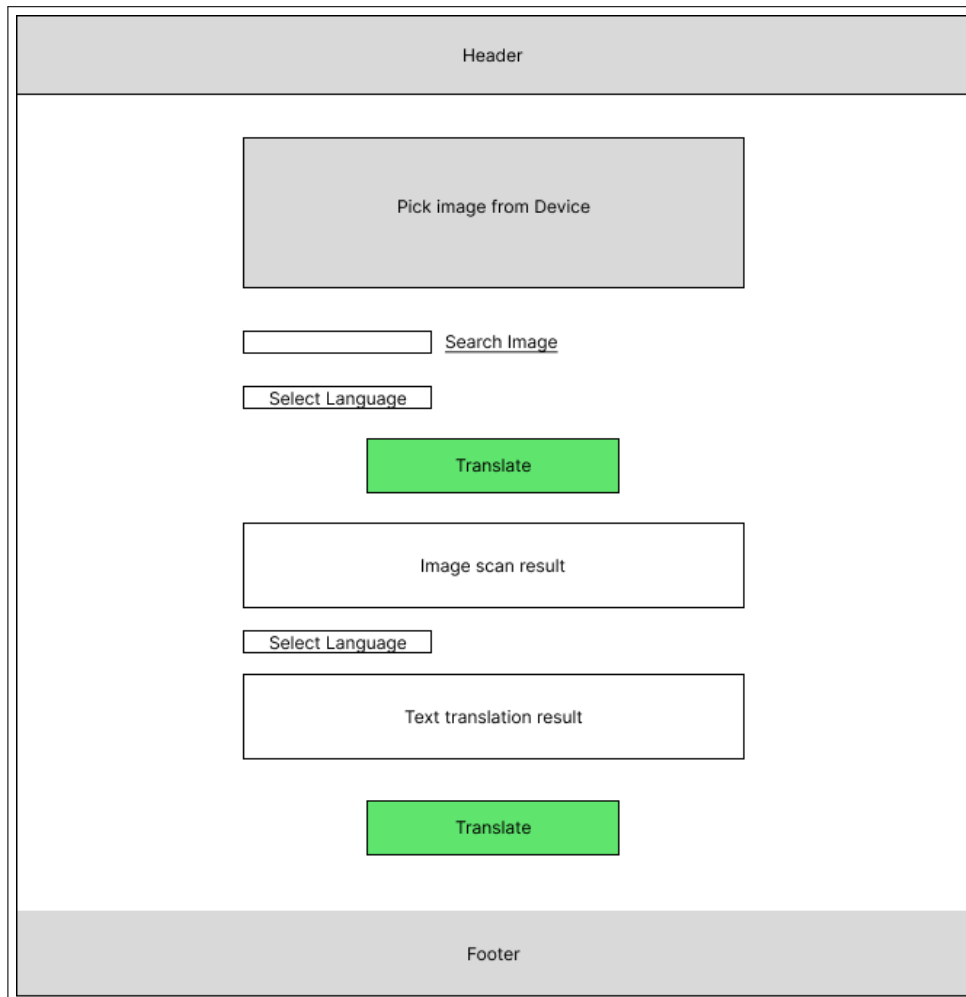


Gambar 3.5. Flowchart website

### 3.5.2 Rancangan Tampilan Website

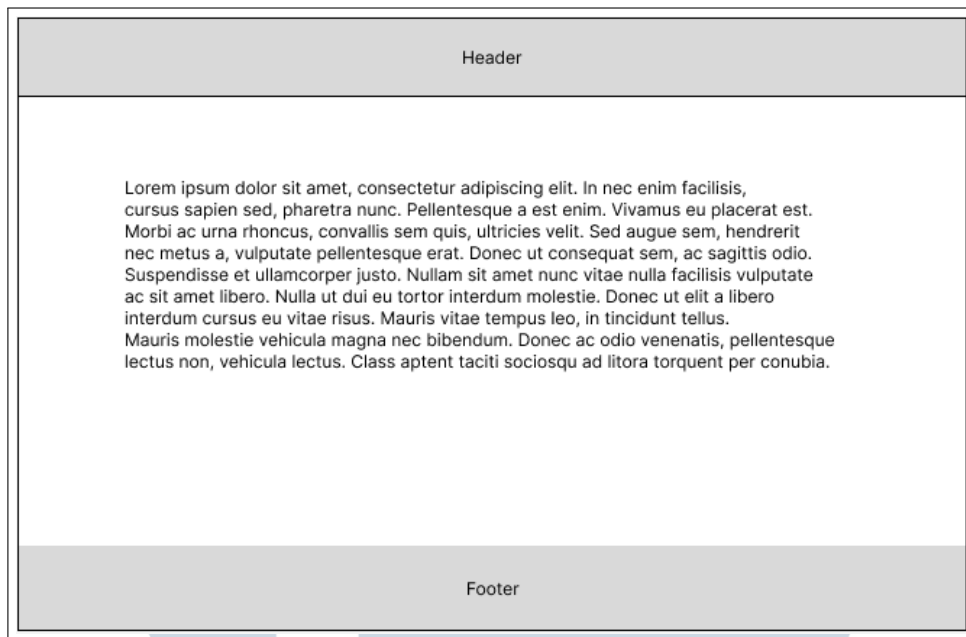
Tampilan website akan dirancang menggunakan Javascript library React menggunakan TailwindCSS dan Sass sebagai *styling*. Terdapat dua halaman yaitu halaman utama dan halaman About. Halaman utama sebagai halaman dimana terdapat semua fitur yang ditawarkan website. Halaman About berisi tujuan dan cara

kerja website yang dirancang.



Gambar 3.6. Rancangan halaman utama

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.7. Rancangan halaman About

### 3.6 Integrasi Frontend dan Backend

Supaya Frontend dapat mengirim *request* ke backend dan LibreTranslate API, digunakan Axios untuk menangani *request*. Dan untuk menerima *request* dari frontend, salah satu caranya adalah menggunakan Flask untuk mengubah backend menjadi sebuah server, hal itu dilakukan dengan membuat route yang menerima tipe *request* yang ditentukan dan menjalankan fungsi yang ditetapkan untuk route tersebut. Setelah model EasyOCR selesai memproses gambar yang diterima, hasilnya akan dikirim kembali sebagai *response* terhadap *request* yang diterima.

### 3.7 Deployment

Deployment aplikasi React akan menggunakan layanan yang disediakan oleh GitHub yaitu github-pages sehingga website akan di-host di domain GitHub. Deployment server backend yang berupa Python gagal dikarenakan ukuran folder EasyOCR yang disebabkan oleh jumlah dependency yang digunakan melebihi batas penyimpanan layanan hosting seperti Heroku, PythonAnywhere, Vercel, dan Google Cloud Platform sehingga antara gagal upload file atau memunculkan error random.