Penerapan OCR dalam Mesin Pencari File dengan Konten Tertentu

Fajar Merah Diwangkara
20/459264/PA/19925

Muhammad Faqih Husaen
19/442478/PA/19227

Ryan Novianno
20/462191/PA/20163

I. Latar Belakang

Digitalisasi merupakan sebuah proses yang mengubah informasi analog menjadi informasi digital. Pengubahan dilakukan dengan bantuan teknologi digital seperti komputer, smartphone, scanner, dan kamera. Proses ini bertujuan untuk mendapatkan efisiensi dan optimalisasi dalam berbagai hal. Salah satu kegiatan digitalisasi yang paling sering kita lakukan adalah mengubah dokumen berbasis kertas menjadi dokumen gambar menggunakan bantuan scanner ataupun kamera. Dokumen hasil digitalisasi tersebut biasanya akan disimpan dalam satu tempat atau folder.

Permasalahan akan muncul ketika terdapat banyak dokumen dan kita ingin mencari dokumen yang memuat informasi tertentu. Permasalahan tersebut adalah komputer tidak mengenali informasi text yang ada di dalam gambar tersebut sehingga harus melakukan pencarian dokumen secara manual. Proses pencarian secara manual membutuhkan banyak waktu dan usaha. Permasalahan tersebut dapat diatasi dengan sistem temu kembali informasi yang dapat menemukan informasi sesuai query dari pengguna. Sistem yang dikembangkan menggunakan OCR yang berfungsi mengekstrak informasi text yang ada pada gambar. OCR merupakan Optical Character Recognition yang dapat mengenali informasi berupa karakter dari suatu dokumen gambar.

II. Studi Terkait

Paper yang kami jadikan referensi berjudul Content-based Image Retrieval using Tesseract OCR Engine and Levenshtein Algorithm. Paper tersebut ditulis oleh Charles Adjetey dan Kofi Sarpong Adu-Manu pada tahun 2021. Pada paper tersebut menjelaskan tentang Image Retrieval System (IRS) yaitu suatu aplikasi yang memungkinkan seseorang untuk mencari gambar yang tersimpan di lokasi manapun di suatu jaringan. Paper ini memberikan teknik untuk mendapatkan informasi dokumen gambar secara penuh berdasarkan pencarian pengguna. Untuk meningkatkan performa hasil, penulis mengkombinasikan mesin Optical Character Recognition (OCR) dan algoritma pencocokan teks yang ditingkatkan dalam pengimplementasiannya, yaitu Mesin Tesseract OCR dan algoritma Levenshtein terintegrasi untuk melakukan pencarian gambar.

III. Pendekatan yang Diusulkan

Untuk menyelesaikan permasalahan yang disebutkan di latar belakang, kami mengajukan untuk membuat sebuah mesin pencari yang dapat mencari file di komputer dengan isi tertentu. Alur dari mesin pencari yang kami usulkan seperti berikut: user memasukkan daftar file dan daftar direktori sebagai lokasi pencarian, lalu sistem pencari akan mengekstrak informasi teks dari seluruh file pada daftar file dan kemudian melakukan transformasi data. Hasil transformasi selanjutnya digunakan untuk melakukan indexing, setelah itu user memasukkan query untuk melakukan pencarian file dengan isi tertentu.

IV. Metode

Pertama-tama, user akan membuka program mesin pencari. Jika sebelumnya user sudah pernah melakukan indexing, maka mesin pencari akan memberi user pilihan apakah user ingin menggunakan hasil indexing sebelumnya atau melakukan indexing yang baru. Jika user ingin menggunakan hasil indexing sebelumnya, mesin pencari akan memberi pilihan apakah user ingin menambah file/folder yang ingin ditambahkan ke proses indexing atau tidak. Jika user tidak ingin menambah file/folder baru yang ingin ditambahkan ke proses indexing, maka user dapat melakukan pencarian file dengan query.

Jika user ingin membuat indexing baru, maka hasil indexing sebelumnya akan dihapus. Jika user belum pernah melakukan indexing, atau jika user ingin membuat indexing baru, atau jika user ingin menambah file/folder ke indexing yang sudah ada, maka akan dilakukan proses pengindexan. Pada proses pengindexan, user akan memberikan daftar file dan folder sebagai lokasi indexing

A. Crawling

Untuk metode crawling, mesin pencari akan menelusuri semua direktori yang diberikan oleh user, lalu akan melakukan indexing untuk semua file yang ada pada folder tersebut, lalu jika pada folder terdapat folder lagi, maka mesin pencari akan menelusuri folder yang ada dalam folder tersebut dan seterusnya sampai tidak menemukan folder lagi.

B. Transformasi Data

Untuk transformasi data, mesin pencari menggunakan proses tokenisasi, penghapusan tanda baca, penghapusan kata stopword, dan melakukan stemming, Tokenisasi merupakan proses mengubah kalimat menjadi kata penyusunnya berdasarkan spasi. Kata stopwords merupakan kata umum yang tidak mengandung informasi penting sehingga bisa dihilangkan. Stemming merupakan proses mengubah kata menjadi

ke bentuk dasarnya. Hal ini dilakukan dengan menghilangkan awalan, sisipan, dan akhiran yang terdapat pada kata. Untuk stemming, penghapusan imbuhan kata, dan penghapusan kata stopword, mesin pencari untuk saat ini hanya mendukung kata dan kalimat bahasa inggris saja.

C. Indexing

Proses indexing mesin pencari yang kami usulkan dimulai dengan pertama kali melakukan ekstraksi teks dari direktori file hasil crawling. Jika file berupa gambar, sistem temu kembali akan memanfaatkan OCR untuk melakukan ekstraksi teks [1]. Setelah diekstraksi, teks kemudian disederhanakan bentuknya melalui proses transformasi data dan kemudian baru dimasukkan proses indexing. Algoritma indexing yang dipakai mesin pencari yang kami usulkan yaitu Single-pass In-memory Indexing (SPIMI).

SPIMI adalah salah satu metode indexing pada Information Retrieval yang mana proses pembangunan inverted indexnya dapa terjadi di memori dan di disk. Metode SPIMI memiliki cara kerja membentuk dictionary yang memuat kata atau term beserta posting list. Posting list berisi dokumen id dari term yang bersangkutan. Dictionary tersebut kemudian akan dimasukkan ke dalam file-file block. File-file block kemudian akan digabungkan untuk membentuk inverted index secara keseluruhan.

D. Kompresi Indeks

Setelah melakukan indexing, proses selanjutnya adalah melakukan kompresi indeks. Kompresi indeks merupakan teknik yang digunakan untuk lebih mengefisienkan indeks yang dibuat baik dari segi kapasitas dan performa. Terdapat dua jenis kompresi yaitu lossless compression dan lossy compression. Pada lossless compression, setiap informasi akan dipertahankan, sementara pada lossy compression, terdapat sebagian informasi yang hilang. Metode kompresi indeks yang dipakai mesin pencari yang kami usulkan adalah variable byte code.

Variable byte code (VB Code) merupakan metode kompresi yang melakukan kompresi pada posting list. VB code memiliki cara kerja melakukan encoding terhadap nilai gap atau selisih dari dokumen id term. Proses encoding dilakukan menggunakan nilai bytes dengan 7 bit terakhir disebut payload dan 1 bit pertama merupakan bit lanjutan.

V. Hasil

A. Crawling

Metode crawling dari sistem temu kembali kami menghasilkan daftar direktori dari setiap file yang terbaca. Sistem kami saat ini dapat membaca file dengan format pdf, doc, txt, png, dan jpg. Sebagai contoh, jika STKI kami berawal dari folder satu, dan susunan folder dan file seperti pada gambar di atas, maka STKI kami akan membaca filesatu.txt. Lalu STKI kami akan masuk ke folderdua dan foldertiga lalu membaca filetiga.txt dan filedua.txt. Terakhir, STKI kami akan masuk ke folderempat dan membaca fileempat.txt. Kemudian didapatkan hasil akhir yang berisi direktori dari keempat file tersebut.

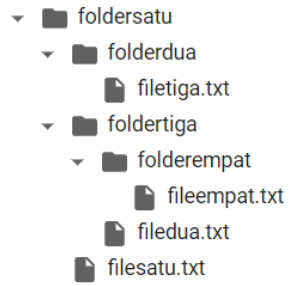
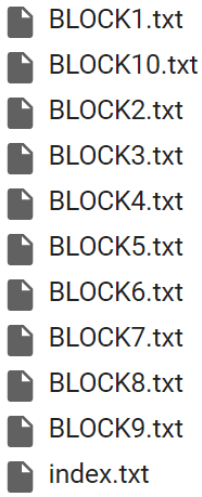


Fig. 1: Hasil Crawling

B. Indexing

Proses indexing menghasilkan file-file block dalam format txt. Setiap block berisi term/kata dan juga posting list yang berisi dokumen id. File-file block kemudian akan digabungkan untuk membentuk inverted index secara keseluruhan.



(a) Block hasil indexing.

```
'0116unknown': [20, 20, 20, 20, 20, 20, 20, 20, 20],
'0117unknown': [20, 20, 20, 20, 20, 20, 20, 20, 20],
'0118unknown': [20, 20, 20, 20, 20, 20, 20, 20, 20, 20],
'0119unknown': [20, 20, 20, 20, 20, 20, 20],
'0120unknown': [20, 20, 20, 20, 20],
'0121unknown': [20],
'0123unknown': [20, 20],
'0124unknown': [20, 20],
'01254773date': [20],
'0125unknown': [20, 20],
'0126unknown': [20, 20],
'0127unknown': [20],
'01282465date': [20],
'0128unknown': [20],
'0129unknown': [20, 20],
'01302950date': [20],
```

(b) Term tiap blocknya.

Fig. 2: Hasil Indexing

Proses indexing menghasilkan file-file block dalam format txt. Setiap block berisi term/kata dan juga posting list yang

berisi dokumen id. File-file block kemudian akan digabungkan untuk membentuk inverted index secara keseluruhan.

C. Kompresi Indeks

Kompresi indeks menghasilkan indeks yang menghabiskan lebih sedikit kapasitas disk dan memiliki waktu pencarian yang lebih cepat. Perbedaan dari hasil pembuatan indeks dan hasil kompresi indeks terletak pada bagian posting list. Hasil kompresi indeks memiliki postling list yang sudah diencoding menggunakan VB code.

DISKUSI

Proses penyiapan (crawling dan transformasi data) serta pembuatan indeks (indexing dan kompresi) telah berhasil diimplementasikan. Sistem temu kembali yang kami kembangkan mencari file lokal yang memiliki konten sesuai dengan query yang dimasukkan pengguna sehingga kami tidak melakukan crawling di internet seperti pada umumnya. Sebagai gantinya kami melakukan crawling secara lokal yang menghasilkan daftar direktori dari setiap file. Daftar tersebut digunakan selanjutnya digunakan untuk melakukan ekstraksi teks yang memanfaatkan fitur OCR. Proses ektraksi yang kami terapkan masih memiliki kekurangan yaitu hanya dapat membaca file dengan format pdf, docx, txt, png, dan jpg. Selanjutnya untuk bagian transformasi data, kami baru menghilangkan stopword dan mengembalikan bentuk kata berbahasa Inggris. Proses pembuatan indeks menggunakan SPIMI masih dapat ditingkatkan kembali terutama dari segi performa.

REFERENCES

[1] C. Adjetey and K. S. Adu-Manu, “Content-based Image Retrieval using Tesseract OCR Engine and Levenshtein Algorithm,” International Journal of Advanced Computer Science and Applications, vol. 12, no. 7, pp. 666–675, 2021, doi: 10.14569/IJACSA.2021.0120776.