Menyelesaikan Teka Teki Silang Menggunakan Pencocokan String dan Regex

Dimas Bagoes Hendrianto - 13522112
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13522112@std.stei.itb.ac.id

***Abstract*—Teka-teki silang adalah permainan kata yang populer di mana pemain diminta untuk mengisi ruang kosong dalam diagram persegi panjang dengan kata-kata yang cocok berdasarkan petunjuk horizontal dan vertikal yang diberikan. Abstrak ini membahas hubungan antara teka-teki silang dan string matching, terutama dalam konteks pendekatan komputasional. Dalam string matching, tujuan utamanya adalah mencocokkan string input dengan pola atau kumpulan string yang telah ditentukan. Pendekatan ini digunakan dalam penyelesaian otomatis teka-teki silang, di mana algoritma mencari kata-kata yang sesuai dengan petunjuk dan memastikan konsistensi solusi dengan kata-kata yang sudah terisi. Abstrak ini juga mengeksplorasi berbagai teknik string matching yang diterapkan dalam pemecahan teka-teki silang, termasuk algoritma pencocokan pola seperti algoritma *Boyer-Moore* dan *Knuth-Morris-Pratt*. Dengan memahami keterkaitan antara teka-teki silang dan string matching secara komputasional, diharapkan dapat memberikan wawasan baru dalam pengembangan algoritma cerdas untuk menyelesaikan teka-teki silang secara efisien.**

Kata kunci—String matching; Knuth-Morris-Pratt (KMP); Boyer-Moore (BM); Regular Expression (Regex); Teka-Teki silang

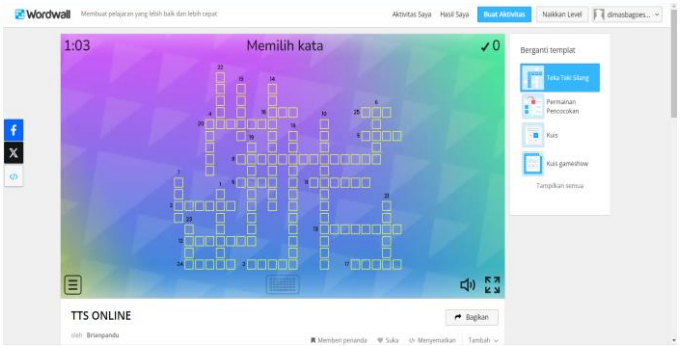
I. PENDAHULUAN

Teka-teki silang merupakan permainan klasik yang telah digemari banyak orang selama bertahun-tahun. Permainan ini menantang pemain untuk mengisi kotak-kotak kosong dengan huruf yang sesuai, berdasarkan petunjuk yang diberikan.Seiring perkembangan teknologi, muncullah berbagai metode baru untuk menyelesaikan teka-teki silang, termasuk penggunaan teknik pemrograman. Salah satu teknik yang populer adalah dengan menggunakan pencocokan string dan regex (regular expression).

Dalam pendahuluan ini, kita akan membahas bagaimana pencocokan string dan regex dapat digunakan untuk menyelesaikan teka-teki silang secara otomatis. Pada dasarnya, pencocokan string memungkinkan kita untuk mencari pola tertentu dalam sebuah teks. Pola ini dapat berupa kata, frasa, atau bahkan karakter tunggal. Regex, di sisi lain, merupakan alat yang lebih kuat yang memungkinkan kita untuk mencari pola yang lebih kompleks, seperti pola yang mengandung karakter tertentu atau yang memiliki struktur tertentu.

Dengan menggunakan pencocokan string dan regex, kita dapat membuat program yang dapat secara otomatis mencari kata-kata yang sesuai dengan petunjuk teka-teki silang. Program ini dapat membaca petunjuk dan kotak kosong, kemudian menggunakan pencocokan string dan regex untuk mencari kata-kata yang sesuai dengan petunjuk dan kotak tersebut.

Penggunaan pencocokan string dan regex untuk menyelesaikan teka-teki silang memiliki beberapa keuntungan. Pertama, metode ini dapat menyelesaikan teka-teki silang dengan lebih cepat dan akurat daripada manusia. Kedua, metode ini dapat digunakan untuk menyelesaikan teka-teki silang yang sangat sulit, untuk kata yang jarang muncul atau jarang digunakan yang mungkin tidak dapat diselesaikan oleh manusia. Teknik ini masih dalam tahap pengembangan, namun memiliki potensi besar untuk merevolusi cara kita menyelesaikan teka-teki silang.



Gambar 1.1 Tampilan Antarmuka Laman wordwall.net

Sumber : https://wordwall.net/id/resource/3359492/tts-online

II. LANDASAN TEORI

A. *Algoritma String Matching*

Algoritma string matching adalah teknik yang digunakan dalam ilmu komputer untuk menemukan kemunculan sebuah pola (pattern) tertentu dalam sebuah teks atau string yang lebih besar. Pola ini dapat berupa kata, frasa, karakter tunggal, atau bahkan struktur teks yang kompleks. Tujuan utamanya adalah untuk menemukan posisi atau indeks di mana pola tersebut cocok dengan teks yang diberikan. Terdapat 2 komponen utama yaitu:

1. *Text (T)*

Ini adalah string yang lebih besar tempat akan mencari polanya. Bisa berupa apa saja, mulai dari kalimat dalam dokumen hingga sebaris kode.

2. *Pattern (P)*

Urutan karakter tertentu yang ingin ditemukan dalam teks. Itu bisa berupa satu kata, frasa tertentu, atau bahkan rangkaian karakter yang lebih kompleks.

Berikut adalah contoh dari string matcing dengan teks T dan pattern P

T : Strategi **algo**ritma

P : **algo**

Algoritma string matching ini telah diaplikasikan ke berbagai bidang sebagai contoh, fitur pencarian kata pada teks editor, mesin pencarian kata seperti google, analisis citraa, bioinformatik seperti pencocokan DNA, dan yang akan dibahas pada makalah ini yaitu teka-teki silang.

B. Algoritma Knuth-Morris-Pratt (KMP)

Algoritma Knuth-Morris-Pratt (KMP) adalah salah satu algoritma pencocokan string (string matching) yang terkenal dengan efisiensi dan kecepatannya dalam mencari pola (pattern) di dalam teks (string). Algoritma ini dikembangkan secara independen oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, dan dipublikasikan bersama pada tahun 1977.

Algoritma KMP menggunakan tabel untuk mempercepat proses pencarian pola. Tabel ini, yang disebut tabel Failure Function (tabel F), berisi informasi tentang berapa banyak karakter yang perlu dilewati jika terjadi ketidakcocokan antara pola dan teks.

Pembuatan tabel F dilakukan sebelum proses pencarian dimulai. Berikut adalah langkah-langkahnya:

1. Inisialisasi tabel F: Isi elemen pertama tabel F dengan 0.
2. Iterasi: Untuk setiap elemen berikutnya dalam tabel F (dilambangkan dengan i), lakukan langkah-langkah berikut:

   Bandingkan karakter ke-i pola dengan karakter ke-i teks.

   Jika cocok, lanjutkan ke langkah berikutnya.

   Jika tidak cocok, lakukan salah satu dari dua hal:

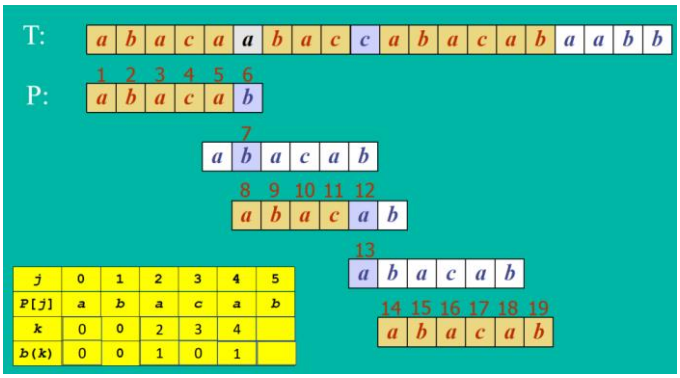
   Jika i = 0, set elemen ke-i tabel F dengan 0.

   Jika i > 0, bandingkan karakter ke-i pola dengan karakter ke-(i - nilai elemen ke-(i - 1) tabel F) teks.
3. Pencarian Pola: Lakukan pencarian pola dari awal teks.

   Bandingkan setiap karakter pola dengan karakter teks yang sesuai.

   Jika cocok, lanjutkan ke karakter berikutnya.

   Jika tidak cocok, gunakan tabel F untuk melompat ke karakter yang tepat dalam teks.



Gambar 2.1 Contoh Implementasi algoritma KMP

Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

C. Algotima Boyer-Moore (BM)

Algoritma Boyer-Moore (BM) adalah salah satu algoritma pencocokan string (string matching) yang terkenal dengan kecepatan dan kecerdasannya dalam mencari pola (pattern) di dalam teks (string). Algoritma ini dikembangkan oleh Robert S. Boyer dan Moore pada tahun 1977 dan dipublikasikan pada tahun 1981.

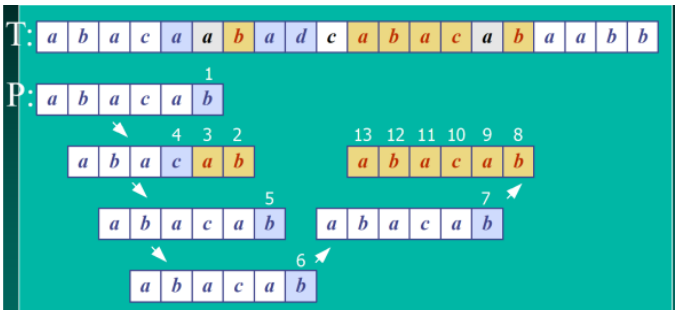
Algoritma Boyer-Moore merupakan algoritma pattern matching yang memanfaatkan dua teknik yang unik, yaitu: 1. The looking-glass technique Teknik ini merupakan teknik yang mencari pattern P pada teks T dengan bergerak mundur dari kanan ke kiri, dimulai dari indeks terakhir. 2. The character-jump technique Teknik ini berarti indeks akan lompat dengan cara yang berbeda, Apabila mismatch terjadi pada T[i] == x, dan karakter pada pattern P[j] tidak sama dengan T[i], maka dilakukan penggeseran karakter.

Terdapat tiga kemungkinan kasus dalam penggeseran karakter pada teknik character-jump, yaitu

1. Kasus pertama
   Jika pattern P memiliki karakter x di sebelah kiri dari indeks j, maka geser P ke kanan sedemikian rupa sehingga P sejajar dengan indeks i, yaitu indeks dimana x terdekat berada.

2. Kasus kedua
   Jika pattern P memiliki karakter x, tetapi bukan di sebelah kiri dari indeks j, maka geser P ke kanan sebanyak 1 karakter menuju T[i+1].

3. Kasus ketiga
   Jika karakter x bukan merupakan substring dari pattern P, maka geser pattern P sedemikian rupa sehingga pattern P[0] sejajar dengan T[i+1]



Gambar 2.2 Contoh Implementasi algoritma BM

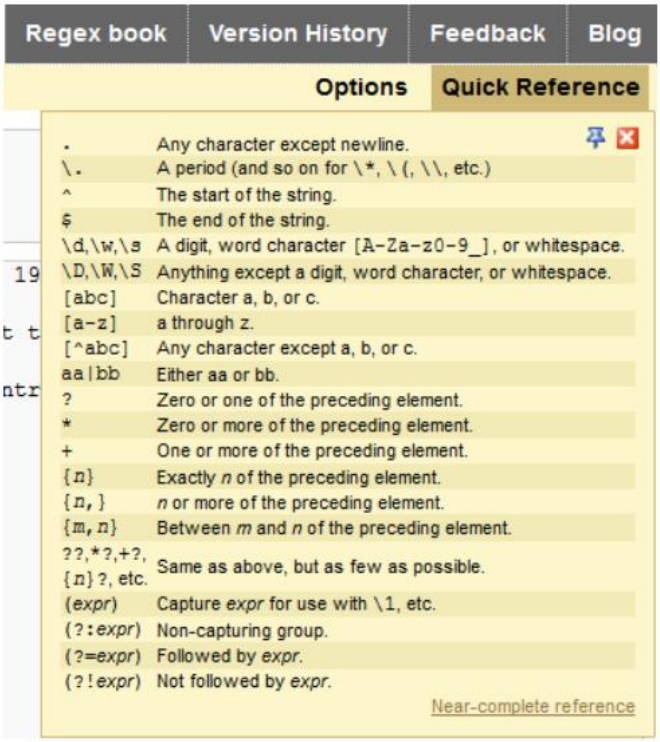
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

D. Regular Expression (Regex)

Regular Expressions (regex) adalah sekumpulan karakter yang digunakan untuk mencocokkan pola dalam teks. Konsep ini sangat berguna dalam berbagai aplikasi pemrograman, seperti pencarian, validasi, dan manipulasi string. Regex bekerja dengan menggunakan kombinasi karakter biasa dan karakter khusus yang dikenal sebagai metakarakter. Misalnya, simbol titik (.) mewakili sembarang karakter tunggal, sedangkan tanda bintang (*) menunjukkan bahwa karakter sebelumnya dapat muncul nol kali atau lebih. Regex sering digunakan dalam bahasa pemrograman seperti Perl, Python, JavaScript, dan banyak lainnya karena kemampuannya yang kuat dalam memproses teks.

Penggunaan regex dapat sangat bervariasi dari yang sederhana hingga yang kompleks. Misalnya, untuk mencocokkan alamat email, regex seperti ^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$ digunakan untuk memastikan bahwa string yang dimasukkan memenuhi pola umum dari sebuah alamat email. Regex ini mengecek keberadaan karakter alfanumerik, simbol tertentu seperti titik dan garis bawah, serta memastikan ada tanda @ dan domain yang valid. Selain itu, regex juga dapat digunakan untuk ekstraksi data dari teks, seperti menemukan semua nomor telepon dalam dokumen atau memvalidasi format tanggal.

Salah satu keunggulan utama regex adalah fleksibilitasnya dalam menangani teks. Namun, kompleksitas regex juga dapat menjadi tantangan, terutama ketika bekerja dengan pola yang sangat rumit. Untuk mempermudah, banyak editor teks dan IDE menyediakan fitur pencarian dan penggantian berbasis regex, sehingga pengguna dapat dengan cepat menemukan dan memodifikasi teks sesuai kebutuhan. Meskipun mempelajari regex memerlukan waktu dan latihan, pemahaman mendalam mengenai cara kerjanya dapat sangat meningkatkan efisiensi dan kemampuan dalam pemrosesan teks.



Gambar 2.3 Tabel Notasi Regex

Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/String-Matching-dengan-Regex-2019.pdf

E. Algoritma Longest Common Subsequence

Algoritma Longest Common Subsequence (LCS) adalah algoritma yang digunakan untuk menemukan subsekuen terpanjang yang sama antara dua atau lebih string (urutan karakter). Subsekuen adalah urutan karakter yang diambil dari string tanpa harus berurutan dalam string aslinya.

Algoritma LCS bekerja dengan membangun tabel dua dimensi yang berisi informasi tentang subsekuen terpanjang yang sama antara setiap prefiks dari string pertama dan setiap prefiks dari string kedua. Berikut adalah langkah-langkahnya:

1. Inisialisasi tabel: Buat tabel dua dimensi dengan ukuran (m+1) x (n+1), di mana m adalah panjang string pertama dan n adalah panjang string kedua. Isi semua elemen tabel dengan nilai 0.
2. Isi tabel: Isi tabel secara bertahap dari kiri ke kanan dan atas ke bawah. Pada setiap elemen tabel, lakukan salah satu dari tiga hal berikut:

   Jika karakter pada string pertama dan string kedua pada posisi yang sama sama, tambahkan 1 ke nilai elemen tabel dan set nilai elemen tabel menjadi nilai maksimum dari elemen di atasnya, di sebelah kirinya, dan diagonal kiri atasnya.

   Jika karakter pada string pertama dan string kedua pada posisi yang sama tidak sama, set nilai elemen tabel menjadi nilai maksimum dari elemen di atasnya dan di sebelah kirinya.

3. Rekonstruksi LCS: Setelah tabel terisi, rekonstruksi LCS dengan melacak kembali dari elemen tabel dengan nilai maksimum di pojok kanan bawah ke elemen tabel dengan nilai 0.

Diberikan dua string, X dan Y, tujuan dari algoritma LCS adalah menemukan subsekuens terpanjang yang terdapat dalam kedua string tersebut dalam urutan yang sama. Misalnya, jika X = "AGGTAB" dan Y = "GXTXAYB", maka LCS dari X dan Y adalah "GTAB".

F. Situs Word Wall

Wordwall.net adalah sebuah platform online yang dirancang untuk membuat, membagikan, dan menggunakan berbagai jenis kegiatan pendidikan interaktif. Situs ini menyediakan alat yang memungkinkan guru dan pendidik untuk membuat aktivitas pembelajaran yang menarik dan dinamis, seperti kuis, teka-teki silang, permainan pencocokan, dan banyak lagi. Dengan antarmuka yang intuitif dan berbagai template yang tersedia, Wordwall.net membantu guru merancang kegiatan yang dapat disesuaikan dengan kebutuhan dan tujuan pembelajaran mereka.

Platform ini sangat fleksibel dan dapat digunakan di berbagai perangkat, termasuk komputer, tablet, dan ponsel pintar, sehingga memudahkan akses bagi pengguna di berbagai situasi belajar, baik di dalam kelas maupun di luar kelas. Salah satu fitur utama Wordwall.net adalah kemampuannya untuk mengonversi satu jenis aktivitas ke dalam beberapa format lain dengan mudah. Misalnya, sebuah kuis pilihan ganda dapat diubah menjadi teka-teki kata atau permainan memori hanya dengan beberapa klik. Hal ini memungkinkan pendidik untuk menggunakan kembali konten yang sama dalam berbagai bentuk yang menarik dan menantang bagi siswa.

Gambar 2.4 Tampilan Teka Teki Silang Laman wordwall.net

Sumber : https://wordwall.net/id/resource/3359492/tts-online

III. PEMBAHASAN

Pada bagian ini akan dijelaskan bagaimana penggunaan algoritma string matching dan regex untuk mencocokkan input dari pengguna dengan teks dan panjangnya yang harus diketikkan dalam situs web wordwall adalah sebagai berikut:

1. Masukan Pengguna ke Wordwall

Pengguna akan memilih terlebih dahulu soal nomor berapa yang ingin dikerjakan terlebih dahulu. Umumnya pemain akan memilih soal-soal yang telah memiliki huruf bantu atau kata dengan huruf paling sedikit yang secara logika akan lebih mudah dan lebih cepat untuk diselesaikan. Worldwall menerima input pengguna berupa teks yang harus diketikkan. Pengguna diberikan deskripsi dari sebuah kata yang harus diketikkan dengan tepat pada layar. Kata tersebut mungkin sudah terisi dari beberapa huruf "pembantu" yang tentunya bisa mempermudah dalam menebak kata yang dimasksud. Pengguna diharapkan mengetikkan setiap karakter dengan benar, mulai dari huruf, angka, hingga tanda baca sesuai dengan yang ada pada layar. Setiap karakter yang diinput oleh pengguna akan langsung diproses pada poin kedua.

2. Pencocokan Kata di Wordwall

Dalam aplikasi teka-teki silang online seperti Wordwall, pencocokan string digunakan untuk memverifikasi apakah jawaban yang diberikan oleh pemain sesuai dengan kata-kata yang diharapkan dalam teka-teki tersebut. Teknik ini mendukung pengalaman interaktif yang mendidik dan menyenangkan bagi pengguna, karena memberikan umpan balik langsung mengenai benar atau salahnya jawaban yang diinput.

Pada intinya, pencocokan string memanfaatkan algoritma perbandingan seperti perbandingan langsung (direct comparison), di mana setiap karakter dalam input pengguna dibandingkan dengan karakter yang sesuai dalam jawaban yang benar. Misalnya, jika jawaban yang benar adalah "apple" dan pengguna menginput "appl", sistem akan mendeteksi bahwa ada perbedaan pada panjang dan karakter terakhir. Pencocokan dapat dibuat lebih canggih dengan mengabaikan perbedaan kapitalisasi huruf (case-insensitive) dan mengabaikan spasi atau tanda baca yang tidak relevan.

Gambar 3.1 Tampilan Teka Teki Silang Laman wordwall.net

Sumber : https://wordwall.net/id/resource/3359492/tts-online

Untuk membuat pengalaman pengguna lebih ramah dan mengurangi frustrasi, beberapa aplikasi teka-teki silang menggunakan teknik fuzzy matching atau pencocokan mendekati. Teknik ini memungkinkan toleransi terhadap kesalahan ketik kecil atau variasi dalam jawaban. Misalnya, algoritma Levenshtein distance dapat digunakan untuk menghitung jumlah perubahan yang diperlukan untuk mengubah input pengguna menjadi jawaban yang benar. Jika perubahan ini berada dalam batas toleransi yang ditetapkan, jawaban masih dapat dianggap benar. Ini sangat berguna dalam konteks pendidikan, di mana tujuan utamanya adalah pembelajaran dan bukan hanya penilaian yang kaku.

Implementasi pencocokan string dalam teka-teki silang online memerlukan pemrograman yang teliti. Platform seperti

Wordwall mungkin menggunakan bahasa pemrograman web seperti JavaScript untuk menangani logika pencocokan di sisi klien. Ketika pengguna menginput jawaban dan menekan tombol 'submit', fungsi JavaScript akan menjalankan algoritma pencocokan dan memberikan umpan balik langsung. Umpan balik ini bisa berupa penanda visual seperti warna hijau untuk jawaban benar dan merah untuk jawaban salah, atau pesan teks yang memberikan petunjuk lebih lanjut. Dengan memberikan umpan balik langsung, pengguna dapat belajar dari kesalahan mereka dan memahami materi dengan lebih baik.

3. Pemanfaatan String Matching

```
public static int[,] LCSLength(string X, string Y)
{
    int m = X.Length;
    int n = Y.Length;
    int[,] c = new int[m + 1, n + 1];

    for (int i = 0; i <= m; i++)
    {
        for (int j = 0; j <= n; j++)
        {
            if (i == 0 || j == 0)
                c[i, j] = 0;
            else if (X[i - 1] == Y[j - 1])
                c[i, j] = c[i - 1, j - 1] + 1;
            else
                c[i, j] = Math.Max(c[i - 1, j], c[i, j - 1]);
        }
    }

    return c;
}

public static string PrintLCS(int[,] c, string X, string Y, int i, int j)
{
    if (i == 0 || j == 0)
        return "";

    if (X[i - 1] == Y[j - 1])
        return PrintLCS(c, X, Y, i - 1, j - 1) + X[i - 1];

    if (c[i, j - 1] > c[i - 1, j])
        return PrintLCS(c, X, Y, i, j - 1);
    else
        return PrintLCS(c, X, Y, i - 1, j);
}
```

Gambar 3.2 Potongan Kode Algoritma yang Digunakan Dalam Bahasa c#

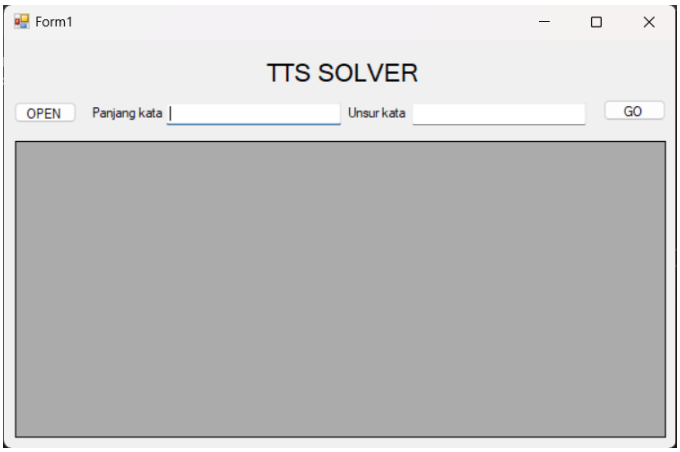
Sumber : Dokumentasi Penulis

Fungsi LCSLength mengambil dua string, X dan Y, dan menghitung panjang subsekuens terpanjang yang ada di kedua string tersebut. Fungsi ini menggunakan pendekatan pemrograman dinamis dan menghasilkan tabel dua dimensi c yang menyimpan panjang LCS untuk setiap kombinasi substring dari X dan Y. Fungsi ini dimulai dengan menginisialisasi tabel c dengan ukuran (m+1) x (n+1), di mana m adalah panjang dari X dan n adalah panjang dari Y. Dalam tabel ini, setiap entri c[i, j] mewakili panjang LCS dari substring X[0..i-1] dan Y[0..j-1]. Algoritma ini mengisi tabel dengan iterasi melalui setiap karakter dari X dan Y, dan membandingkan karakter tersebut. Jika karakter cocok, nilai diagonal sebelumnya di tabel ditambah satu. Jika tidak cocok, nilai maksimum antara nilai di atas atau di sebelah kiri diambil.

Fungsi PrintLCS bertanggung jawab untuk membangun dan mengembalikan subsekuens terpanjang berdasarkan tabel c yang dihasilkan oleh fungsi LCSLength. Fungsi ini memanfaatkan rekursi untuk melacak kembali langkah-langkah dari tabel c untuk menentukan karakter-karakter yang membentuk LCS. Jika karakter terakhir dari substring saat ini cocok (X[i-1] == Y[j-1]), karakter tersebut ditambahkan ke hasil LCS, dan fungsi dipanggil kembali dengan indeks yang dikurangi satu. Jika karakter tidak cocok, fungsi memilih jalur dengan nilai terbesar antara c[i, j-1] dan c[i-1, j], untuk memastikan jalur yang diambil memberikan subsekuens terpanjang. Fungsi ini berlanjut hingga mencapai salah satu batas dari string, yaitu ketika i atau j sama dengan nol.

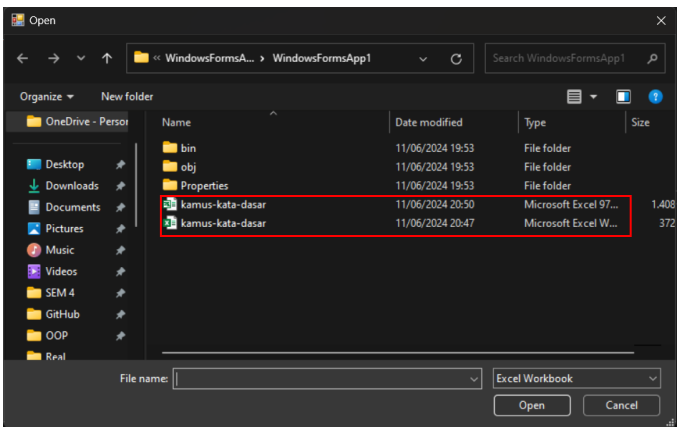
4. Masukan Pengguna ke Wordwall Solver

Pengguna akan memilih terlebih file berisi kata kata atau kamus yang ingin dijadikan acuan dalam memecahkan persoalan di wordwall ini, fle yang dimasukkan harus dalam format xls atau xlsx. Pemain akan memasukkan juga panjang dari kata yang ingin ia cari diikuti dengan huruf bantu apa yang telah tersedia.



Gambar 3.3 Tampilan Aplikasi

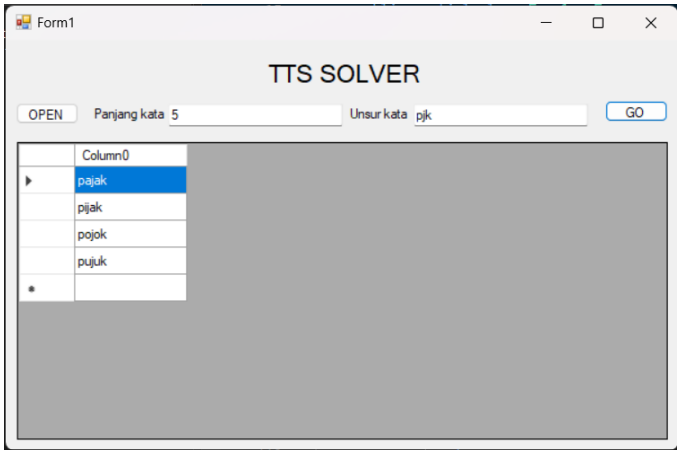
Sumber : Dokumentasi Penulis



Gambar 3.4 Tampilan File Manager

Sumber : Dokumentasi Penulis

Setelah pengguna memasukkan data tadi maka akan kata – kata yang sesuai akan dicari dari kamus dan ditampilkan kata – kata yang relevan. Sebagai contoh pada Gambar 3.1 terdapat persoalan dengan kata yang memiliki panjang 5 dengan huruf bantu “pjk”. Apabila kita menggunakan aplikasi TTS Solver ini maka akan tamapak beberapa kata hasil proses yaitu pajak, pijak, pojok, dna pujuk seperti gambar di bawah.



Gambar 3.5 Tampilan Hasil Aplikasi

Sumber : Dokumentasi Penulis

Gambar 3.6 Tampilan Hasil Input ke Wordladder

: Sumber : https://wordwall.net/id/resource/3359492/tts-online

Dipilih satu dari keempat kata hasil pengolahan kata oleh TTS Solver yaitu kata “pajak”. Kemudian setelah dimasukkan ke laman jawaban Wordwall didapati kalau hasilnya benar. Contoh barusan barulah satu dari seian banyaknya kata yang harus dipecahkan dalam permainan kata TTS ini. Pengguna bisa melakukan pengulangan untuk setiap soalnya dengan memanfaatkan aplikasi TTS Solver sehingga menjadi lebih cepat dan mudah

IV. KESIMPULAN

Regular expressions (regex) dan teknik string matching memiliki peran penting dalam menyelesaikan teka-teki silang dengan efisien dan akurat. Regex, dengan kemampuannya untuk mencocokkan pola dalam teks, memungkinkan pengguna untuk menemukan kata atau frasa yang memenuhi kriteria tertentu dalam sebuah database kata. Misalnya, jika sebuah petunjuk meminta kata yang dimulai dengan huruf 'A' dan berakhir dengan 'E', regex seperti `^A.*E$` dapat digunakan untuk secara cepat menemukan semua kata yang memenuhi kriteria tersebut, mempercepat proses pencarian dibandingkan metode manual.

Teknik string matching juga sangat berguna dalam konteks ini, terutama ketika mencoba mencocokkan pola kata dalam teka-teki silang yang kompleks. Algoritma seperti Knuth-Morris-Pratt (KMP) atau Boyer-Moore dapat digunakan untuk menemukan kata atau frasa dalam kumpulan teks besar dengan efisiensi tinggi. Dalam aplikasi teka-teki silang, string matching memungkinkan pengguna untuk dengan cepat menemukan posisi dan frekuensi kata dalam database, membantu dalam mengidentifikasi jawaban yang benar berdasarkan petunjuk yang diberikan.

Kombinasi antara regex dan string matching meningkatkan kemampuan untuk menyelesaikan teka-teki silang dengan meminimalisir waktu yang dibutuhkan untuk mencari dan mencocokkan kata. Regex dapat digunakan untuk mendefinisikan pola-pola spesifik yang kemudian dicari menggunakan algoritma string matching. Dengan pendekatan ini, tidak hanya pencarian menjadi lebih cepat, tetapi juga lebih akurat, karena pola pencarian dapat disesuaikan dengan lebih tepat sesuai dengan petunjuk yang diberikan dalam teka-teki silang.

Secara keseluruhan, penggunaan regex dan teknik string matching memberikan solusi yang kuat dan efisien dalam penyelesaian teka-teki silang. Kedua alat ini memungkinkan pemecah teka-teki untuk menangani tantangan dengan cara yang lebih sistematis dan terstruktur, mengurangi kesalahan dan meningkatkan akurasi hasil. Dengan demikian, regex dan string matching tidak hanya mempercepat proses pencarian dan pencocokan kata, tetapi juga meningkatkan kualitas dan kepuasan dalam menyelesaikan teka-teki silang, menjadikannya alat yang tak ternilai bagi para penggemar dan penyelesai teka-teki profesional.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur yang sebesar-besarnya kepada Tuhan Yang Maha Esa sehingga penulis dapat menyelesaikan makalah ini dengan tepat waktu. Keberhasilan ini tidak lepas dari bimbingan dan dukungan banyak pihak. Pertama-tama, penulis ingin menyampaikan rasa terima kasih yang mendalam kepada dosen pengampu mata kuliah IF2211 Strategi Algoritma K-02, Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc., atas ilmu yang telah disampaikan selama perkuliahan dan bimbingan yang diberikan selama satu semester penuh. Ilmu dan arahan dari beliau sangat membantu penulis dalam memahami materi dan menyelesaikan makalah ini. Selanjutnya, penulis juga ingin mengucapkan terima kasih yang sebesar-besarnya kepada teman-teman IF K-02 yang selalu siap memberikan bantuan dan dukungan ketika penulis menghadapi kesulitan. Kebersamaan dan kerjasama yang telah terjalin selama ini sangat berarti bagi penulis, baik dalam proses belajar maupun dalam penyelesaian tugas-tugas akademik. Penulis berharap makalah ini dapat memberikan manfaat yang signifikan, baik secara langsung maupun tidak langsung, bagi para pembaca. Semoga hasil karya ini dapat menjadi referensi yang berguna dan menambah wawasan di bidang strategi algoritma. Akhir kata, penulis menyadari bahwa makalah ini masih jauh dari sempurna, oleh karena itu, penulis sangat

mengharapkan kritik dan saran yang membangun dari pembaca untuk perbaikan di masa mendatang.

LINK GITHUB

https://github.com/dimasb1954/MakalahStima.git

REFERENSI

[1] Munir, Rinaldi. 2021. Pencocokan string (String/Pattern Matching) *https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf* Diakses 11 Juni 2024.

[2] Munir, Rinaldi. 2019. String Matching dengan Regex https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/String-Matching-dengan-Regex-2019.pdf Diakses 11 Juni 2024.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024

Dimas Bagoes Hendrianto
13522112