

Pencarian Transformasi Kata Terpendek (Word Ladder) pada Kamus Bahasa Indonesia Menggunakan Breadth-First Search

Muhammad Adam Mirza - 18223015

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia

E-mail: adammirza310@gmail.com, 18223015@std.stei.itb.ac.id

Abstrak—*Word ladder* merupakan persoalan pencarian urutan transformasi kata dari kata awal menuju kata tujuan melalui perubahan satu huruf pada setiap langkah. Penelitian ini menerapkan algoritma *Breadth-First Search* untuk mencari jalur transformasi terpendek pada kamus Bahasa Indonesia. Setiap kata dimodelkan sebagai simpul graf, sedangkan hubungan antarkata dibentuk apabila dua kata memiliki panjang yang sama dan berbeda tepat satu huruf. Dataset kamus dibersihkan dengan mempertahankan kata tunggal berhuruf alfabet, kemudian pencarian dilakukan terhadap sepuluh pasangan kata uji. Hasil pengujian menunjukkan bahwa seluruh pasangan kata berhasil ditemukan jalurnya, dengan rata-rata 2,0 transformasi, rata-rata 794,2 simpul dikunjungi, dan rata-rata waktu eksekusi 11,7796 ms. Berdasarkan hasil tersebut, *Breadth-First Search* terbukti sesuai untuk persoalan *word ladder* karena mampu menjamin jalur terpendek pada graf tak berbobot.

Kata kunci—*Breadth-First Search*, graf, kamus Bahasa Indonesia, pencarian jalur terpendek, *word ladder*

I. PENDAHULUAN

Persoalan *word ladder* merupakan salah satu bentuk permasalahan pencarian jalur dalam graf. Pada persoalan ini, suatu kata awal perlu diubah menjadi kata tujuan melalui serangkaian transformasi. Setiap transformasi hanya diperbolehkan mengubah satu huruf, dan setiap kata antara yang terbentuk harus merupakan kata valid dalam kamus. Dengan aturan tersebut, persoalan *word ladder* dapat dipandang sebagai persoalan pencarian jalur terpendek pada graf tak berbobot.

Dalam konteks Strategi Algoritma, persoalan ini relevan karena menunjukkan penerapan algoritma pencarian graf secara langsung. Setiap kata dapat direpresentasikan sebagai simpul, sedangkan sisi menghubungkan dua kata yang berbeda tepat satu huruf. Karena setiap transformasi memiliki bobot yang sama, algoritma Breadth-First Search dapat digunakan untuk menjamin bahwa jalur pertama yang ditemukan dari simpul awal menuju simpul tujuan merupakan jalur dengan jumlah transformasi paling sedikit.

Penelitian ini bertujuan untuk menerapkan algoritma Breadth-First Search pada kamus Bahasa Indonesia, mengevaluasi hasil transformasi yang diperoleh, serta mengamati jumlah simpul yang dikunjungi dan waktu eksekusi pada beberapa pasangan kata. Dengan demikian, makalah ini tidak hanya memaparkan teori BFS, tetapi juga menyajikan implementasi dan pengujian sederhana terhadap data kata Bahasa Indonesia.

Persoalan transformasi kata juga memiliki nilai pedagogis karena aturan masalahnya sederhana, tetapi representasi grafnya dapat menjadi cukup besar ketika

kamus diperluas. Setiap kata tidak berdiri sendiri, melainkan memiliki hubungan dengan kata lain berdasarkan perubahan huruf. Hubungan ini membuat kamus dapat dipandang sebagai ruang keadaan, sedangkan algoritma pencarian bertugas menemukan urutan perpindahan dari satu keadaan ke keadaan lain.

Pemodelan tersebut sejalan dengan konsep dasar strategi algoritma, yaitu memilih pendekatan penyelesaian yang sesuai dengan karakteristik masalah. Karena setiap langkah transformasi memiliki bobot yang seragam, penggunaan algoritma berbasis prioritas biaya tidak diperlukan. Algoritma BFS sudah cukup untuk memperoleh solusi optimal dalam arti jumlah transformasi minimum, selama graf yang dibentuk tidak memiliki bobot berbeda pada setiap sisi.

Batasan penelitian ditetapkan agar ruang lingkup pembahasan tetap terarah. Penelitian ini hanya membahas transformasi kata dengan panjang yang sama, sehingga operasi penyisipan dan penghapusan huruf tidak digunakan. Selain itu, transformasi hanya diperbolehkan pada tingkat karakter, bukan pada tingkat fonetik, morfologi, atau makna. Dengan batasan tersebut, hasil penelitian dapat difokuskan pada penerapan BFS sebagai algoritma pencarian jalur pada graf tak berbobot.

II. DASAR TEORI

A. Graf

Graf merupakan struktur yang terdiri atas himpunan simpul dan himpunan sisi [3]. Dalam penelitian ini, graf didefinisikan sebagai $G = (V, E)$, dengan V sebagai himpunan kata dan E sebagai himpunan pasangan kata yang dapat ditransformasikan melalui perubahan tepat satu huruf. Graf yang digunakan bersifat tak berbobot karena setiap transformasi dianggap memiliki biaya yang sama, yaitu satu langkah.

B. Breadth-First Search

Breadth-First Search adalah algoritma penelusuran graf yang bekerja dengan mengunjungi simpul berdasarkan tingkat kedalaman [1, 4]. Simpul awal dikunjungi terlebih dahulu, kemudian seluruh tetangganya dikunjungi sebelum pencarian berlanjut ke tingkat berikutnya. Struktur antrean digunakan agar simpul yang lebih dahulu ditemukan diproses lebih dahulu.

Pada graf tak berbobot, BFS memiliki sifat penting, yaitu mampu menemukan jalur dengan jumlah sisi paling sedikit dari simpul awal ke simpul tujuan [1, 4]. Hal ini terjadi karena seluruh simpul pada jarak k dari simpul awal selalu diproses sebelum simpul pada jarak $k + 1$. Oleh karena itu, ketika simpul tujuan pertama kali

ditemukan, jarak yang diperoleh merupakan jarak minimum.

C. Word Ladder

Word ladder adalah persoalan transformasi kata yang dapat dipandang sebagai persoalan pencarian pada ruang keadaan, dengan setiap kata valid sebagai keadaan dan setiap perubahan satu huruf sebagai perpindahan antar-keadaan [2]. Dua kata dianggap bertetangga apabila keduanya memiliki panjang yang sama dan berbeda tepat satu karakter. Sebagai contoh, kata batu dapat diubah menjadi bata karena hanya huruf terakhir yang berubah. Sebaliknya, batu dan bisa tidak bertetangga karena terdapat lebih dari satu perbedaan karakter.

D. Kompleksitas BFS

Kompleksitas waktu BFS pada graf secara umum adalah $O(|V| + |E|)$ karena setiap simpul dan setiap sisi akan diproses paling banyak satu kali [1], [4]. Pada persoalan word ladder, nilai $|V|$ bergantung pada jumlah kata dalam kamus yang memiliki panjang sama dengan kata awal dan kata tujuan. Sementara itu, nilai $|E|$ bergantung pada banyaknya pasangan kata yang berbeda tepat satu huruf. Jika kamus memiliki banyak kata dengan pola huruf yang saling berdekatan, jumlah sisi dapat meningkat dan pencarian dapat mengunjungi lebih banyak simpul sebelum mencapai kata tujuan.

Pada implementasi sederhana, pembentukan semua sisi secara eksplisit dapat menjadi mahal karena setiap kata harus dibandingkan dengan kata lain. Jika terdapat n kata dengan panjang m , pemeriksaan seluruh pasangan kata membutuhkan biaya hingga $O(n^2 \times m)$. Oleh karena itu, penelitian ini menggunakan pendekatan indeks pola agar daftar kandidat tetangga dapat diperoleh tanpa membandingkan setiap pasangan kata secara langsung.

Walaupun indeks pola mempercepat pencarian tetangga, BFS tetap memiliki karakteristik penelusuran melebar. Artinya, algoritma tidak hanya mengikuti satu jalur yang terlihat paling menjanjikan, tetapi memeriksa seluruh kandidat pada tingkat kedalaman yang sama. Sifat tersebut menyebabkan BFS dapat mengunjungi banyak simpul, namun sifat yang sama juga menjadi alasan mengapa BFS mampu menjamin jalur terpendek pada graf tak berbobot.

III. METODOLOGI

A. Dataset Kamus

Dataset yang digunakan berupa daftar entri kata Bahasa Indonesia yang diperoleh dari repositori kumpulan kata Bahasa Indonesia berbasis KBBI [5]. Karena data kamus mentah dapat memuat frasa, tanda baca, kata berulang dengan tanda hubung, atau karakter non-alfabet, tahap pembersihan data dilakukan sebelum pembentukan graf. Pembersihan tersebut diperlukan agar simpul graf hanya merepresentasikan kata tunggal, sesuai dengan definisi persoalan *word ladder*.

B. Pembersihan Data

Pembersihan data dilakukan dengan mengubah seluruh entri menjadi huruf kecil, menghapus entri kosong, menghapus entri yang mengandung spasi, menghapus entri yang mengandung angka atau tanda baca, serta menghapus duplikat. Dengan aturan tersebut, entri seperti adat pepadun dan anak-anak tidak digunakan karena tidak merepresentasikan satu kata tunggal berhuruf alfabet.

C. Pembentukan Tetangga Kata

Pendekatan langsung untuk mencari tetangga adalah membandingkan setiap kata dengan seluruh kata lain yang memiliki panjang sama. Namun, pendekatan tersebut dapat menghasilkan biaya komputasi yang besar. Oleh karena itu, implementasi menggunakan indeks pola *wildcard* sebagai optimasi pencarian kandidat tetangga, sehingga setiap kata tidak perlu dibandingkan secara langsung dengan seluruh kata lain [1]. Sebagai contoh, kata batu menghasilkan pola **atu*, *b*tu*, *ba*u*, dan *bat**. Kata lain yang memiliki salah satu pola yang sama akan menjadi kandidat tetangga.

Dua kata kemudian dianggap bertetangga apabila berbeda tepat satu huruf. Dengan cara ini, proses pencarian tetangga dapat dilakukan lebih efisien karena kandidat tidak perlu dibandingkan dengan seluruh kata dalam kamus.

D. Pencarian Jalur dengan BFS

Pencarian dimulai dari kata awal. Kata awal dimasukkan ke dalam antrean dan ditandai sebagai telah dikunjungi. Setiap kali sebuah kata dikeluarkan dari antrean, seluruh tetangganya diperiksa. Tetangga yang belum pernah dikunjungi akan dimasukkan ke dalam antrean dan dicatat induknya. Pencatatan induk digunakan untuk membangun kembali jalur transformasi setelah kata tujuan ditemukan.

E. Rekonstruksi Jalur

Selama proses BFS, setiap simpul baru yang ditemukan disimpan bersama simpul induknya. Informasi induk tersebut tidak diperlukan untuk menentukan apakah kata tujuan dapat dicapai, tetapi diperlukan untuk membangun kembali urutan transformasi. Ketika kata tujuan ditemukan, proses rekonstruksi dilakukan dari kata tujuan menuju kata awal dengan mengikuti hubungan induk secara terbalik, kemudian urutan tersebut dibalik agar menjadi jalur dari kata awal ke kata tujuan.

Sebagai contoh, apabila kata tujuan bisa ditemukan melalui induk bata, kata bata ditemukan melalui induk batu, maka jalur akhir yang terbentuk adalah batu, bata, bata, bisa. Mekanisme ini lebih efisien dibandingkan menyimpan seluruh jalur pada setiap elemen antrean karena memori yang digunakan hanya berupa pemetaan simpul ke simpul induk.

F. Validasi Masukan

Validasi masukan dilakukan sebelum BFS dijalankan. Kata awal dan kata tujuan harus memiliki panjang yang sama karena transformasi hanya dilakukan melalui perubahan huruf, bukan penambahan atau penghapusan huruf. Selain itu, kedua kata harus terdapat di dalam kamus yang telah dibersihkan. Apabila salah satu kata tidak ditemukan dalam kamus, pencarian tidak dilanjutkan karena simpul awal atau simpul tujuan tidak berada dalam graf.

Validasi ini juga penting dari sudut pandang pengalaman pengguna program. Tanpa validasi, program dapat mengembalikan hasil tidak ditemukan tanpa menjelaskan penyebabnya. Dengan validasi, kesalahan seperti salah ketik, kata yang tidak ada dalam kamus, atau perbedaan panjang kata dapat diidentifikasi secara langsung.

G. Representasi Ruang Keadaan

Dalam penelitian ini, ruang keadaan didefinisikan sebagai himpunan seluruh kata yang lolos proses pembersihan dan memiliki panjang yang sama dengan kata awal. Pembatasan berdasarkan panjang kata dilakukan

karena perubahan satu huruf tidak mengubah panjang *string*. Dengan demikian, kata berpanjang empat tidak perlu dibandingkan dengan kata berpanjang lima atau enam. Pembatasan ini mengurangi ukuran ruang pencarian dan membuat pembentukan indeks pola menjadi lebih efisien.

Setiap keadaan memiliki aksi berupa perubahan salah satu huruf menjadi huruf lain sehingga menghasilkan kata yang tetap ada dalam kamus. Namun, implementasi tidak membangkitkan seluruh kemungkinan huruf dari a sampai z secara eksplisit. Sebaliknya, kandidat tetangga diperoleh melalui indeks *wildcard*. Pendekatan ini lebih langsung karena hanya kandidat yang memang terdapat dalam kamus yang akan dipertimbangkan sebagai tetangga.

H. Kriteria Keberhasilan

Suatu pencarian dianggap berhasil apabila kata tujuan dapat dicapai dari kata awal melalui satu atau lebih transformasi yang seluruhnya valid dalam kamus. Apabila kata tujuan ditemukan, program menampilkan jalur, jumlah transformasi, jumlah kata pada jalur, jumlah simpul yang dikunjungi, dan waktu eksekusi. Apabila kata tujuan tidak ditemukan, program menampilkan status gagal disertai jumlah simpul yang telah dikunjungi. Dengan kriteria ini, hasil pencarian dapat dianalisis baik pada kasus berhasil maupun tidak berhasil.

IV. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi Program

Program diimplementasikan menggunakan bahasa Python. Fungsi utama program terdiri atas pembacaan kamus, pembersihan kata, pembentukan indeks pola, pencarian tetangga, algoritma BFS, dan rekonstruksi jalur. Program juga menampilkan jumlah transformasi, jumlah kata pada jalur, jumlah simpul yang dikunjungi, serta waktu eksekusi dalam milidetik.

B. Skenario Pengujian

Pengujian dilakukan terhadap sepuluh pasangan kata yang memiliki panjang empat dan lima huruf. Setiap pasangan diuji untuk mengetahui apakah jalur transformasi dapat ditemukan, berapa jumlah langkah yang dibutuhkan, berapa simpul yang dikunjungi, serta berapa waktu eksekusi pencarian. Hasil pengujian ditampilkan pada Tabel I.

TABEL I. HASIL PENGUJIAN TRANSFORMASI KATA

Awal	Tujuan	Panjang	Status	Langkah	Jalur	Simpul	Waktu (ms)
batu	bisa	4	Ya	3	batu→bata→bita→bisa	1576	12,5730
batu	jauh	4	Ya	4	batu→baku→bakh→bauh→jauh	1791	8,7900
buku	luka	4	Ya	2	buku→buka→luka	404	6,5166
kata	kota	4	Ya	1	kata→kota	229	6,2690
kaya	saya	4	Ya	1	kaya→saya	127	5,8768
baru	biru	4	Ya	1	baru→biru	77	6,1354
makan	salam	5	Ya	3	makan→malan→malam→salam	1873	17,9162
batuk	jatuh	5	Ya	3	batuk→patuk→patuh→jatuh	1732	20,1620
dapur	kapur	5	Ya	1	dapur→kapur	59	17,0130
telur	teluk	5	Ya	1	telur→teluk	74	16,5441

Berdasarkan Tabel I, seluruh pasangan kata berhasil ditemukan jalurnya. Rata-rata jumlah transformasi adalah 2,0 langkah, rata-rata jumlah simpul yang dikunjungi adalah 794,2 simpul, dan rata-rata waktu eksekusi adalah 11,7796 ms. Nilai tersebut menunjukkan bahwa pencarian dapat dilakukan dalam waktu yang relatif kecil pada data uji yang digunakan, meskipun jumlah simpul yang dikunjungi dapat berbeda jauh antara pasangan kata.

Kolom jalur pada Tabel I menunjukkan urutan kata yang diperoleh dari proses rekonstruksi induk. Setiap tanda panah menunjukkan satu transformasi. Dengan demikian, jumlah transformasi pada kolom langkah sama dengan jumlah perpindahan antar kata pada kolom jalur. Kolom simpul menunjukkan banyaknya simpul yang dikunjungi selama pencarian, bukan hanya jumlah simpul yang muncul pada jalur akhir.

Perbedaan antara jumlah simpul pada jalur dan jumlah simpul yang dikunjungi menjadi hal penting dalam analisis BFS. Jalur akhir hanya menyatakan solusi yang dipilih, sedangkan jumlah simpul yang dikunjungi menunjukkan luasnya ruang pencarian yang telah diperiksa sebelum solusi ditemukan. Pada beberapa kasus, BFS dapat menemukan jalur pendek tetapi tetap mengunjungi banyak simpul karena banyak kandidat lain berada pada kedalaman yang sama atau lebih rendah.

C. Metrik Pengujian

Terdapat empat metrik utama yang dicatat pada pengujian. Metrik pertama adalah status ditemukan atau tidaknya jalur. Metrik kedua adalah jumlah transformasi, yaitu jumlah sisi pada jalur dari kata awal menuju kata tujuan. Metrik ketiga adalah jumlah simpul yang dikunjungi, yaitu banyaknya kata yang telah dimasukkan ke himpunan visited selama BFS berjalan. Metrik keempat adalah waktu eksekusi dalam milidetik, yang mencakup proses pembentukan indeks pola untuk kelompok kata dengan panjang yang sesuai dan proses pencarian BFS.

Metrik jumlah transformasi digunakan untuk menilai hasil utama algoritma, sedangkan jumlah simpul yang dikunjungi dan waktu eksekusi digunakan untuk mengamati perilaku pencarian. Dua pasangan kata dapat memiliki jumlah transformasi yang sama, tetapi jumlah simpul yang dikunjungi dapat berbeda karena struktur hubungan antar kata pada graf tidak seragam.

D. Lingkungan Pengujian

Pengujian dilakukan pada program Python yang dijalankan melalui antarmuka baris perintah. Program membaca berkas kamus, membersihkan data, membentuk indeks pola, menerima kata awal dan kata tujuan, lalu menampilkan hasil pencarian. Karena pengujian hanya dilakukan untuk membandingkan hasil antar pasangan kata, waktu eksekusi digunakan sebagai indikasi relatif, bukan sebagai tolok ukur performa mutlak.

E. Alur Eksekusi Program

Alur eksekusi program dimulai dari pembacaan berkas kamus. Setelah kamus dimuat, program meminta kata awal dan kata tujuan dari pengguna. Program kemudian memeriksa kesamaan panjang kedua kata dan keberadaan kedua kata di dalam kamus. Jika validasi berhasil, program menyaring kata berdasarkan panjang yang sesuai, membentuk indeks pola, menjalankan BFS, dan melakukan rekonstruksi jalur apabila kata tujuan ditemukan.

Pemisahan fungsi dalam program dilakukan agar setiap bagian memiliki tanggung jawab yang jelas. Fungsi pembacaan kamus bertanggung jawab terhadap data masukan, fungsi graf bertanggung jawab terhadap pembentukan tetangga, sedangkan fungsi BFS bertanggung jawab terhadap proses pencarian. Pemisahan ini memudahkan pengujian dan memudahkan penjelasan implementasi dalam makalah.

F. Langkah-Langkah Algoritma

Secara umum, proses BFS yang digunakan dapat dinyatakan sebagai berikut. Pertama, masukkan kata awal ke antrian. Kedua, selama antrian tidak kosong, ambil kata terdepan. Ketiga, apabila kata tersebut sama dengan kata tujuan, bangun jalur dari data induk dan hentikan pencarian. Keempat, jika belum sama, bangkitkan seluruh tetangga kata tersebut. Kelima, setiap tetangga yang belum dikunjungi dimasukkan ke antrian dan dicatat induknya. Jika antrian habis tanpa menemukan kata tujuan, pencarian dinyatakan gagal.

Langkah-langkah tersebut memperlihatkan bahwa BFS tidak membutuhkan fungsi heuristik, berbeda dari pendekatan pencarian berbasis estimasi seperti Greedy Best First Search atau A* [2]. Urutan kunjungan sepenuhnya ditentukan oleh antrian dan tingkat kedalaman simpul.

V. HASIL DAN ANALISIS

A. Analisis Jalur Transformasi

Hasil pengujian menunjukkan bahwa BFS mampu menemukan jalur transformasi untuk seluruh pasangan kata. Pasangan kata seperti kata menuju kota, kaya menuju saya, baru menuju biru, dapur menuju kapur, dan telur menuju teluk hanya membutuhkan satu transformasi. Hal ini menunjukkan bahwa kedua kata pada pasangan tersebut bertetangga langsung dalam graf karena hanya berbeda satu huruf.

Pasangan lain membutuhkan lebih dari satu transformasi. Sebagai contoh, batu menuju bisa membutuhkan tiga transformasi melalui jalur batu, bata, bita, dan bisa. Jalur tersebut menunjukkan bahwa BFS tidak sekadar mencari kata yang mirip dengan tujuan, tetapi menelusuri hubungan antarkata secara bertahap hingga kata tujuan ditemukan.

B. Analisis Jumlah Simpul yang Dikunjungi

Jumlah simpul yang dikunjungi tidak selalu sebanding dengan jumlah langkah pada jalur akhir. Pasangan dapur menuju kapur hanya mengunjungi 59 simpul, sedangkan pasangan makan menuju salam mengunjungi 1873 simpul. Perbedaan tersebut dipengaruhi oleh struktur graf pada kelompok kata dengan panjang tertentu. Apabila kata awal memiliki banyak hubungan tidak langsung sebelum mencapai kata tujuan, BFS perlu memeriksa lebih banyak simpul.

Sifat BFS yang menjelajahi simpul berdasarkan tingkat kedalaman menyebabkan algoritma dapat mengunjungi banyak simpul meskipun jalur akhir relatif pendek. Kondisi tersebut merupakan konsekuensi dari jaminan optimalitas BFS pada graf tak berbobot. Untuk memperoleh jalur terpendek, algoritma harus memastikan bahwa seluruh kemungkinan jalur yang lebih pendek telah diperiksa terlebih dahulu.

C. Analisis Waktu Eksekusi

Waktu eksekusi pada pengujian berada pada rentang 5,8768 ms sampai 20,1620 ms. Waktu terendah diperoleh pada pasangan kaya menuju saya, sedangkan waktu tertinggi diperoleh pada pasangan batuk menuju jatuh. Secara umum, waktu eksekusi dipengaruhi oleh jumlah kata pada kelompok panjang yang sama, jumlah tetangga yang dihasilkan oleh indeks pola, serta jumlah simpul yang dikunjungi selama proses BFS.

Walaupun demikian, waktu eksekusi tidak selalu meningkat secara linear terhadap jumlah simpul yang

dikunjungi karena terdapat pengaruh struktur data dan proses pembentukan indeks pola. Pada implementasi ini, indeks pola dibangun untuk kata-kata dengan panjang yang sama dengan kata awal dan kata tujuan. Dengan demikian, biaya awal pembentukan indeks juga berkontribusi terhadap waktu total pencarian.

D. Keterbatasan

Penelitian ini memiliki beberapa keterbatasan. Pertama, kualitas hasil sangat bergantung pada kualitas kamus yang digunakan. Kata yang jarang digunakan atau entri yang tidak umum dapat menghasilkan jalur yang secara algoritmik valid, tetapi kurang alami secara kebahasaan. Kedua, penelitian ini belum membedakan kelas kata, makna, atau tingkat kelaziman penggunaan kata. Ketiga, pengujian masih dilakukan pada jumlah pasangan kata yang terbatas.

E. Kesesuaian BFS dengan Word Ladder

BFS sesuai untuk persoalan word ladder karena setiap transformasi memiliki bobot yang sama, sehingga pencarian dapat diperlakukan sebagai pencarian jalur terpendek pada graf tak berbobot [1], [4]. Dalam kondisi tersebut, tujuan utama bukan mencari jalur dengan biaya numerik tertentu, melainkan mencari jalur dengan jumlah langkah paling sedikit. Apabila algoritma lain yang bersifat heuristik digunakan tanpa jaminan optimalitas, jalur yang ditemukan dapat lebih pendek secara perkiraan tetapi belum tentu merupakan jalur minimum yang sebenarnya.

Kesesuaian tersebut terlihat dari pasangan kata yang memiliki hubungan langsung. Pada pasangan kata menuju kota, BFS menemukan jalur satu langkah karena kata tujuan berada pada tingkat pertama dari kata awal. Pada pasangan yang membutuhkan beberapa tahap, BFS tetap memeriksa tingkat pencarian secara berurutan sehingga jalur yang ditemukan tetap merupakan jalur terpendek berdasarkan graf yang dibentuk.

F. Pengaruh Kualitas Kamus

Kualitas kamus memiliki pengaruh langsung terhadap hasil transformasi karena setiap entri dalam dataset diperlakukan sebagai simpul valid dalam graf [5]. Apabila kamus memuat kata yang tidak baku, kata serapan yang jarang digunakan, atau entri yang kurang lazim, jalur yang diperoleh dapat terlihat kurang natural meskipun sah menurut aturan algoritmik. Sebagai contoh, jalur batu menuju jauh memuat kata bakh dan bauh pada data uji. Apabila kata tersebut dianggap tidak layak secara kebahasaan, kamus perlu dikurasi lebih lanjut agar jalur yang dihasilkan lebih sesuai dengan penggunaan Bahasa Indonesia sehari-hari.

Dengan demikian, keberhasilan algoritma tidak hanya ditentukan oleh BFS, tetapi juga oleh definisi validitas kata pada dataset. Dari sudut pandang algoritmik, setiap entri yang berada di dalam kamus dianggap sebagai simpul valid. Dari sudut pandang kebahasaan, tidak semua simpul tersebut memiliki tingkat kelaziman yang sama. Perbedaan ini perlu dijelaskan agar hasil pengujian tidak ditafsirkan sebagai penilaian linguistik, melainkan sebagai hasil pencarian pada graf yang dibentuk dari dataset tertentu.

G. Implikasi terhadap Perancangan Eksperimen

Pemilihan pasangan kata uji perlu mempertimbangkan variasi panjang kata dan variasi tingkat kesulitan. Pasangan yang berbeda satu huruf digunakan untuk memeriksa apakah program dapat mengenali tetangga langsung. Pasangan yang membutuhkan beberapa transformasi

digunakan untuk memeriksa kemampuan BFS dalam menemukan jalur bertahap. Pasangan yang tidak memiliki jalur juga dapat ditambahkan pada penelitian lanjutan untuk mengamati perilaku BFS ketika seluruh komponen graf yang relevan telah dieksplorasi.

Pada pengujian ini, seluruh pasangan yang digunakan memiliki jalur. Hal tersebut memudahkan analisis awal karena setiap hasil dapat dibandingkan berdasarkan jumlah langkah, jumlah simpul, dan waktu eksekusi. Namun, untuk evaluasi yang lebih luas, pasangan tanpa jalur sebaiknya juga dimasukkan agar dapat terlihat biaya pencarian ketika BFS tidak berhasil menemukan kata tujuan.

H. Perbandingan dengan Pendekatan Brute Force

Apabila seluruh pasangan kata dibandingkan secara langsung untuk membentuk graf, jumlah pemeriksaan dapat bertambah sangat besar. Untuk n kata, terdapat sekitar $n(n-1)/2$ pasangan yang perlu diperiksa. Setiap pemeriksaan masih membutuhkan perbandingan karakter sepanjang m . Pendekatan tersebut masih dapat digunakan untuk kamus kecil, tetapi kurang efisien ketika jumlah kata meningkat.

Penggunaan indeks pola membuat pencarian tetangga lebih terarah. Kata yang tidak berbagi pola wildcard tidak perlu diperiksa sebagai kandidat tetangga. Dengan demikian, implementasi tetap mempertahankan hasil yang sama dengan perbandingan langsung, tetapi mengurangi jumlah kandidat yang perlu diproses. Perbedaan ini menjadi kontribusi teknis penting dalam implementasi karena persoalan word ladder sangat bergantung pada kecepatan memperoleh daftar tetangga.

I. Interpretasi Hasil Eksperimen

Hasil eksperimen perlu ditafsirkan sebagai evaluasi terhadap graf yang terbentuk dari kamus tertentu. Jalur yang ditemukan oleh BFS bersifat optimal pada graf tersebut, tetapi bukan berarti selalu menjadi jalur yang paling natural menurut penutur bahasa. Apabila kamus berubah, jalur yang ditemukan juga dapat berubah karena simpul dan sisi pada graf ikut berubah. Oleh karena itu, hasil eksperimen tidak dapat dipisahkan dari proses pembersihan dan pemilihan data.

Pada pasangan batu menuju bisa, jalur yang ditemukan adalah batu, bata, bita, bisa. Jalur ini terdiri atas tiga transformasi dan setiap transformasi mengubah satu huruf. Pada pasangan makan menuju salam, jalur yang ditemukan adalah makan, malan, malam, salam. Hasil tersebut memperlihatkan bahwa BFS dapat bergerak melalui kata antara yang tidak selalu intuitif pada awal pencarian, selama kata tersebut memenuhi aturan graf dan terdapat dalam kamus.

J. Ancaman terhadap Validitas

Ancaman terhadap validitas penelitian terutama berasal dari kualitas dataset dan keterbatasan pasangan uji. Jika kamus mengandung entri yang kurang baku, hasil jalur dapat memuat kata yang kurang dikenal. Jika pasangan uji terlalu sedikit, kesimpulan mengenai performa waktu juga belum dapat digeneralisasi untuk seluruh kamus. Oleh karena itu, hasil yang diperoleh dalam makalah ini sebaiknya dipandang sebagai bukti penerapan metode, bukan sebagai pengukuran performa final untuk seluruh kosakata Bahasa Indonesia.

Ancaman lain berasal dari variasi lingkungan eksekusi. Waktu eksekusi dapat berubah bergantung pada perangkat

keras, versi Python, dan beban sistem ketika program dijalankan. Oleh sebab itu, analisis waktu lebih tepat digunakan sebagai indikasi relatif antar pasangan uji dalam lingkungan yang sama, bukan sebagai angka mutlak yang berlaku pada semua lingkungan.

K. Rekomendasi Perbaikan Implementasi

Implementasi yang digunakan dalam penelitian ini sudah cukup untuk menunjukkan penerapan BFS pada persoalan word ladder. Namun, beberapa perbaikan masih dapat dilakukan agar program lebih sesuai untuk eksperimen berskala besar. Pertama, indeks pola dapat disimpan satu kali setelah kamus dibaca, kemudian digunakan kembali untuk beberapa pasangan kata dengan panjang yang sama. Pada implementasi saat ini, pembentukan indeks dilakukan pada setiap pencarian sehingga waktu eksekusi mencakup biaya persiapan indeks.

Kedua, kata dalam kamus dapat dikelompokkan berdasarkan panjang sejak awal. Pengelompokan tersebut membuat program tidak perlu melakukan penyaringan ulang untuk setiap pencarian. Ketiga, hasil pencarian dapat dilengkapi dengan informasi jumlah tetangga rata-rata pada kelompok panjang kata tertentu. Informasi tersebut berguna untuk menjelaskan mengapa sebagian pasangan kata mengunjungi lebih banyak simpul dibandingkan pasangan lain.

Keempat, kamus dapat diberi lapisan validasi tambahan berdasarkan daftar kata baku atau frekuensi penggunaan. Dengan cara tersebut, jalur yang dihasilkan tidak hanya valid secara algoritmik, tetapi juga lebih wajar secara kebahasaan. Perbaikan ini tidak mengubah algoritma BFS, tetapi memperbaiki kualitas graf yang menjadi masukan bagi algoritma.

L. Replikasi Eksperimen

Agar eksperimen dapat direplikasi, berkas pasangan kata uji disimpan dalam format CSV yang memuat kolom kata awal dan kata tujuan. Program eksperimen membaca setiap baris pada berkas tersebut, menjalankan BFS untuk masing-masing pasangan, lalu menyimpan hasil ke berkas CSV keluaran. Format ini dipilih karena mudah diperiksa secara manual dan dapat langsung digunakan sebagai bahan tabel pada makalah.

Replikasi eksperimen dilakukan dengan tiga langkah. Pertama, kamus mentah dibersihkan menjadi daftar kata tunggal. Kedua, pasangan kata uji disusun dengan memastikan bahwa setiap kata terdapat di dalam kamus. Ketiga, program eksperimen dijalankan untuk menghasilkan data jalur, jumlah langkah, jumlah simpul yang dikunjungi, dan waktu eksekusi. Dengan alur tersebut, pengujian dapat diperluas tanpa mengubah kode utama program.

VI. KESIMPULAN

Penelitian ini menunjukkan bahwa algoritma Breadth-First Search dapat diterapkan untuk menyelesaikan persoalan pencarian transformasi kata terpendek pada kamus Bahasa Indonesia. Pemodelan kata sebagai simpul dan relasi beda satu huruf sebagai sisi membuat persoalan word ladder dapat diselesaikan sebagai pencarian jalur terpendek pada graf tak berbobot.

Selain itu, penelitian lanjutan dapat menambahkan analisis terhadap komponen graf berdasarkan panjang kata. Analisis tersebut dapat menunjukkan kelompok panjang kata mana yang memiliki konektivitas tinggi dan kelompok

mana yang cenderung menghasilkan banyak simpul terisolasi. Informasi tersebut dapat memperkaya pembahasan mengenai hubungan antara karakteristik kamus dan performa algoritma pencarian.

TAUTAN REPOSITORI

<https://github.com/AdmMRZ/word-ladder-bfs-stima>

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada dosen pengampu IF2211 Strategi Algoritma atas materi pembelajaran strategi pencarian graf, serta pihak penyedia sumber data kata Bahasa Indonesia untuk bahan pengujian. Terima kasih juga kepada keluarga atas doa dan dukungannya, serta teman-teman yang selalu membantu dan memberikan motivasi.

REFERENSI

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA: MIT Press, 2022.
- [2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ: Pearson, 2021.
- [3] D. E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, 3rd ed. Boston, MA: Addison-Wesley, 1997.
- [4] R. Munir, *Breadth/Depth First Search (BFS/DFS)*. Slide Kuliah IF2211 Strategi Algoritma. Bandung: Program Studi Teknik Informatika, Institut Teknologi Bandung, 2026.
- [5] Damzaky, kumpulan-kata-bahasa-indonesia-KBBI, *GitHub repository*. [Online]. Available: <https://github.com/damzaky/kumpulan-kata-bahasa-indonesia-KBBI>.

PERNYATAAN

Dengan ini dinyatakan bahwa makalah ini merupakan tulisan sendiri, bukan saduran atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 17 Juni 2026



Muhammad Adam Mirza 18223015