



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**DESAIN DAN ANALISIS ALGORITMA KOMPUTASI STRING
DENGAN METODE MEET IN THE MIDDLE DAN DYNAMIC
PROGRAMMING PADA PERMASALAHAN KLASIK SPOJ 9967
PLAYING WITH WORDS**

DEWANGGA WINASFORCEPTA WINARDI
NRP 5113 100 098

Dosen Pembimbing 1
Rully Soelaiman, S.Kom., M.Kom.

Dosen Pembimbing 2
Ir. F.X. Arunanto, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

Halaman ini sengaja dikosongkan



TUGAS AKHIR - KI141502

**DESAIN DAN ANALISIS ALGORITMA KOMPUTASI STRING
DENGAN METODE MEET IN THE MIDDLE DAN DYNAMIC
PROGRAMMING PADA PERMASALAHAN KLASIK SPOJ 9967
PLAYING WITH WORDS**

**DEWANGGA WINASFORCEPTA WINARDI
NRP 5113 100 098**

**Dosen Pembimbing 1
Rully Soelaiman, S.Kom., M.Kom.**

**Dosen Pembimbing 2
Ir. F.X. Arunanto, M.Sc.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017**

Halaman ini sengaja dikosongkan

UNDERGRADUATE THESES - KI141502

**ALGORITHM DESIGN AND ANALYSIS FOR STRING COM-
PUTATION USING MEET IN THE MIDDLE TECHNIQUE AND
DYNAMIC PROGRAMMING IN SPOJ CLASSIC PROBLEM
9967 PLAYING WITH WORDS**

DEWANGGA WINASFORCEPTA WINARDI
NRP 5113 100 098

Supervisor 1
Rully Soelaiman, S.Kom., M.Kom.

Supervisor 2
Ir. F.X. Arunanto, M.Sc.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

Halaman ini sengaja dikosongkan

DESAIN DAN ANALISIS ALGORITMA KOMPUTASI STRING DENGAN METODE MEET IN THE MIDDLE DAN DYNAMIC PROGRAMMING PADA PERMASALAHAN KLASIK SPOJ 9967 PLAYING WITH WORDS

Nama : DEWANGGA WINASFORCEPTA WINARDI
NRP : 5113 100 098
Jurusan : Teknik Informatika FTIF - ITS
Pembimbing I : Rully Soelaiman, S.Kom., M.Kom.
Pembimbing II : Ir. F.X. Arunanto, M.Sc.

Abstrak

Diberikan dua buah string $orig1$ dan $orig2$. Diberikan tiga tahapan proses enkripsi untuk menghasilkan string $ad1$ dan $ad2$. Tahap pertama adalah string $orig1$ diacak urutan karakter-karakternya. Tahap kedua adalah string $orig2$ diacak urutan karakter-karakternya. Tahap terakhir adalah salah satu karakter dari string $orig1$ atau $orig2$ diganti dengan karakter sebelum atau sesudahnya dalam alfabet. Jarak dua buah string didefinisikan sebagai jumlah dari selisih mutlak dari karakter-karakter pada posisi yang sama. Diberikan sebuah bilangan bulat X yang merupakan jarak dari string $orig1$ dan string $ad1$ dijumlahkan dengan jarak dari string $orig2$ dan string $ad2$. Tentukan jumlah kemungkinan kombinasi string $orig1$ dan $orig2$ jika diberikan string $ad1$, $ad2$ dan nilai X .

Meet-in-the-middle adalah sebuah teknik pencarian dengan paradigma divide and conquer yang membagi permasalahan menjadi dua, lalu menyelesaikannya masing-masing, lalu menggabungkannya kembali.

Dynamic programming adalah sebuah paradigma untuk mendapatkan nilai optimal dari beberapa kemungkinan jawaban, dimana permasalahan tersebut memiliki submasalah tumpang tindih dan

struktur optimal.

Pada tugas akhir ini akan dirancang penyelesaian masalah yang disampaikan pada paragraf pertama dengan menggunakan teknik meet-in-the-middle dan pendekatan dynamic programming.

*Solusi yang dikembangkan berjalan dengan kompleksitas waktu $O(2^{|S|} * MAX_DIST)$, dimana $|S|$ adalah panjang string yang diberikan dan MAX_DIST adalah jarak antar string maksimal.*

Kata Kunci: string, divide and conquer, meet-in-the-middle, dynamic programming

**ALGORITHM DESIGN AND ANALYSIS FOR STRING
COMPUTATION USING MEET IN THE MIDDLE TECHNI-
QUE AND DYNAMIC PROGRAMMING IN SPOJ CLASSIC
PROBLEM 9967 PLAYING WITH WORDS**

Name : DEWANGGA WINASFORCEPTA WINARDI
NRP : 5113 100 098
Major : Informatics Department Faculty of IT - ITS
Supervisor I : Rully Soelaiman, S.Kom., M.Kom.
Supervisor II : Ir. F.X. Arunanto, M.Sc.

Abstract

TO DO

**Keywords: string, divide and conquer, meet-in-the-middle,
dynamic programming**

Halaman ini sengaja dikosongkan

DAFTAR ISI

SAMPUL	i
ABSTRAK	vii
ABSTRACT	ix
LEMBAR PENGESAHAN	vii
DAFTAR ISI	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvii
DAFTAR KODE SUMBER	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Metodologi	4
1.6 Sistematika Penulisan	5
2 DASAR TEORI	7
2.1 Deskripsi Permasalahan	7
2.2 Analisa Submasalah Optimal	8
2.2.1 Membagi Permasalahan Menjadi Dua Sub- masalah yang <i>Independent</i>	10

2.2.2	Submasalah Optimal untuk Menghitung Jumlah Kemungkinan String Awal Tanpa Operasi <i>Replace</i>	12
2.2.3	Submasalah Optimal untuk Menghitung Jumlah Kemungkinan String Awal dengan Operasi <i>Replace</i>	12
2.2.4	Penggabungan Hasil Perhitungan Submasalah	12
2.3	Pemodelan Relasi Rekurens	12
2.4	Meet In The Middle	12
2.5	Dynamic Programming	12
3	DESAIN	13
3.1	Desain Umum Sistem	13
3.2	Desain Algoritma	13
3.2.1	Desain Fungsi <i>Preprocess</i>	13
3.2.2	Desain Fungsi Perhitungan Tanpa Operasi <i>Replace</i>	13
3.2.3	Desain Fungsi Perhitungan Dengan Operasi <i>Replace</i>	14
3.2.4	Desain Fungsi Penggabungan Hasil Perhitungan Submasalah	14
4	IMPLEMENTASI	15
4.1	Lingkungan Implementasi	15
4.2	Rancangan Data	15
4.2.1	Data Masukan	15
4.2.2	Data Keluaran	15
4.3	Implementasi Algoritma	16
4.3.1	Header-Header yang Diperlukan	16
4.3.2	Implementasi Variabel Global	16
4.3.3	Implementasi Fungsi Main	16
4.3.4	Implementasi Fungsi <i>Preprocess</i>	16
4.3.5	Implementasi Fungsi Perhitungan Tanpa Operasi <i>Replace</i>	16

4.3.6	Implementasi Fungsi Perhitungan Dengan Operasi Replace	17
4.3.7	Implementasi Fungsi Penggabungan Hasil Perhitungan Submasalah	17
5	UJI COBA DAN EVALUASI	19
5.1	Lingkungan Uji Coba	19
5.2	Data Uji Coba	19
5.2.1	Dataset pada Pengujian Daring SPOJ . . .	19
5.2.2	Dataset Generator	19
5.3	Skenario Uji Coba	19
5.3.1	Uji Coba Kebenaran	20
5.3.2	Uji Coba Peforma	20
5.4	Analisis Hasil Uji Coba	20
6	KESIMPULAN	21
6.1	Kesimpulan	21

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1	Ilustrasi teknik <i>meet in the middle</i>	9
2.2	Ilustrasi tahapan teknik <i>meet in the middle</i>	11

Halaman ini sengaja dikosongkan

DAFTAR KODE SUMBER

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Pada bab ini akan dijelaskan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Memasuki abad ke-20 perkembangan teknologi informasi semakin cepat. Tidak dapat dipungkiri bahwa hampir semua aspek kehidupan mulai terkomputerisasi. Tujuannya adalah konsistensi dan akurasi. Dengan desain dan implementasi yang tepat sebuah program komputer dapat membantu manusia untuk melakukan sebuah proses yang jika dikerjakan oleh manusia secara langsung akan membutuhkan waktu yang lama dan rentan terhadap error. Keunggulan inilah yang menyebabkan semakin banyak penggunaan teknologi informasi pada aspek-aspek kehidupan manusia.

Kesadaran akan keunggulan komputer dalam melakukan suatu pekerjaan menyebabkan semakin besar pula keinginan manusia untuk menyelesaikan masalah-masalah yang sebelumnya tidak dapat diselesaikan tanpa bantuan komputer. Masalah-masalah ini seringkali berkaitan dengan data yang memiliki ukuran sangat besar. Dari data tersebut tentu terdapat informasi-informasi tertentu yang ingin diambil. Dengan metode yang tepat informasi yang diperlukan dapat diambil dengan waktu yang relatif singkat.

Besarnya ukuran data tentu menjadi sebuah masalah tersendiri bagi para ilmuwan. Perkembangan perangkat keras yang ada belum dapat mengimbangi dengan kebutuhan komputasi yang semakin besar. Untuk inilah penyelesaian sebuah masalah membutuhkan pendekatan

an yang tepat, sehingga dapat ditemukan algoritma dan struktur data yang sesuai dengan permasalahan yang ada.

Topik Tugas Akhir ini mengacu pada permasalahan klasik SPOJ 9967 *Playing With Words*. Diberikan dua buah string *orig1* dan *orig2*. Diberikan tiga tahapan proses enkripsi untuk menghasilkan string *ad1* dan *ad2* sebagai berikut:

1. String *orig1* diacak urutan karakter-karakternya.
2. String *orig2* diacak urutan karakter-karakternya.
3. Salah satu karakter dari string *orig1* atau *orig2* diganti dengan karakter sebelum atau sesudahnya dalam alfabet.

Jarak dua buah string didefinisikan sebagai jumlah dari selisih mutlak dari karakter-karakter pada posisi yang sama. Diberikan sebuah bilangan bulat X yang merupakan jarak dari string *orig1* dan string *ad1* dijumlahkan dengan jarak dari string *orig2* dan string *ad2*. Berapakah jumlah kemungkinan kombinasi string *orig1* dan *orig2* jika diberikan string *ad1*, *ad2* dan nilai X .

Untuk menyelesaikan permasalahan di atas, penulis akan menggunakan pendekatan solusi dengan teknik *meet in the middle* dan *dynamic programming*. Selain dapat menjawab pertanyaan dengan benar, waktu juga menjadi salah satu faktor penting untuk memberikan gambaran tentang performa dari algoritma yang dirancang.

Hasil dari Tugas Akhir ini diharapkan dapat memberikan gambaran mengenai performa algoritma dengan teknik *meet in the middle* dan *dynamic programming*.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Penerapan *meet in the middle* dan *dynamic programming* sebagai pendekatan untuk menyelesaikan permasalahan klasik

SPOJ 9967 *Playing With Words*.

2. Pengimplementasian desain algoritma yang didasari oleh metode *meet in the middle* dan *dynamic programming* untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.

1.3 Batasan Masalah

Permasalahan yang dibahas pada Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Algoritma yang digunakan diilhami oleh metode *meet in the middle* dan *dynamic programming*.
2. Implementasi algoritma menggunakan bahasa pemrograman C++.
3. *Dataset* yang digunakan untuk menguji algoritma yang telah dirancang adalah *dataset* pada permasalahan klasik SPOJ 9967 *Playing With Words*.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Melakukan analisis dan mendesain algoritma yang diilhami oleh metode *meet in the middle* dan *dynamic programming* untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.
2. Melakukan implementasi algoritma dengan metode *meet in the middle* dan *dynamic programming* untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.
3. Mengevaluasi algoritma dengan metode *meet in the middle* dan *dynamic programming* untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.

1.5 Metodologi

Metodologi yang digunakan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir
Pada tahap ini dilakukan penyusunan proposal tugas akhir yang berisi permasalahan dan gagasan solusi yang akan diteliti pada permasalahan klasik SPOJ 9967 *Playing With Words*.
2. Studi literatur
Pada tahap ini dilakukan pencarian informasi dan studi literatur mengenai pengetahuan atau metode yang dapat digunakan dalam penyelesaian masalah. Informasi didapatkan dari materi-materi yang berhubungan dengan algoritma dan struktur data yang digunakan untuk penyelesaian permasalahan ini, materi-materi tersebut didapatkan dari buku, jurnal, maupun internet.
3. Desain
Pada tahap ini dilakukan desain rancangan algoritma yang digunakan dalam solusi untuk pemecahan permasalahan klasik SPOJ 9967 *Playing With Words*.
4. Implementasi perangkat lunak
Pada tahap ini dilakukan implementasi atau realiasi dari rancangan desain algoritma yang telah dibangun pada tahap desain ke dalam bentuk program.
5. Uji coba dan evaluasi
Pada tahap ini dilakukan uji coba kebenaran implementasi dan uji coba generalisasi. Pengujian kebenaran dilakukan pada sistem penilaian daring SPOJ sesuai dengan masalah yang dikerjakan untuk diuji apakah luaran dari program telah sesuai. Uji coba generalisasi digunakan untuk menguji struktur data generalisasi pada variasi permasalahan lain.
6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan penyusunan buku Tugas Akhir yang berisi dokumentasi hasil pengerjaan Tugas Akhir.

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan buku Tugas Akhir ini:

1. BAB I: PENDAHULUAN
Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi dan sistematika penulisan Tugas Akhir.
2. BAB II: DASAR TEORI
Bab ini berisi dasar teori mengenai permasalahan dan algoritma penyelesaian yang digunakan dalam Tugas Akhir
3. BAB III: DESAIN
Bab ini berisi desain algoritma dan struktur data yang digunakan dalam penyelesaian permasalahan.
4. BAB IV: IMPLEMENTASI
Bab ini berisi implementasi berdasarkan desain algoritma yang telah dilakukan pada tahap desain.
5. BAB V: UJI COBA DAN EVALUASI
Bab ini berisi uji coba dan evaluasi dari hasil implementasi yang telah dilakukan pada tahap implementasi.
6. BAB VI: KESIMPULAN
Bab ini berisi kesimpulan yang didapat dari hasil uji coba yang telah dilakukan.

Halaman ini sengaja dikosongkan

BAB 2

DASAR TEORI

Pada bab ini akan dijelaskan mengenai dasar teori yang menjadi dasar pengerjaan Tugas Akhir ini.

2.1 Deskripsi Permasalahan

Amr M. sangat curiga dengan sebuah iklan yang terdiri dari 2 buah string *ad1* dan *ad2*. Ia menduga kedua string tersebut menyembipkan pesan tersembunyi. Setelah menghubungi sumber yang terpercaya, ia menemukan proses untuk menyembunyikan pesan tersebut. Pesan asli yang dibawa selalu berupa dua buah string *orig1* dan *orig2*. Berikut adalah langkah-langkah transformasi pesan asli menjadi pesan pada iklan:

1. Karakter-karakter pada string *orig1* diacak urutannya.
2. Karakter-karakter pada string *orig2* diacak urutannya.
3. Salah satu karakter dari string *orig1* atau *orig2* diganti dengan karakter sebelum atau sesudahnya dalam alfabet.

Langkah - langkah di atas akan menghasilkan string *ad1* dan *ad2* dari *orig1* dan *orig2* secara berurutan. Contohnya untuk string *orig1* = "bcd" dan *orig2* = "wcy" dapat menghasilkan string *ad1* = "dcb" dan *ad2* = "cxy" di mana "cxy" berasal dari "wcy" yang diacak menjadi "cwxy" dan karakter 'w' digantikan dengan karakter 'x'.

Setelah melakukan riset, Amr menemukan sebuah jarak X , di mana X adalah $jarak(orig1 + ad1)$ ditambah dengan $jarak(orig2 + ad2)$. Jarak antara dua string didefinisikan sebagai jumlah dari selisih absolut dari karakter-karakter pada posisi yang sama. Contohnya $jarak("ab", "cd") = |'a' - 'c'| + |'b' - 'd'| = 4$. Diberikan

string $ad1$, $ad2$ dan sebuah bilangan bulat X . Hitung jumlah kemungkinan string $orig1$ dan $orig2$ yang mungkin.

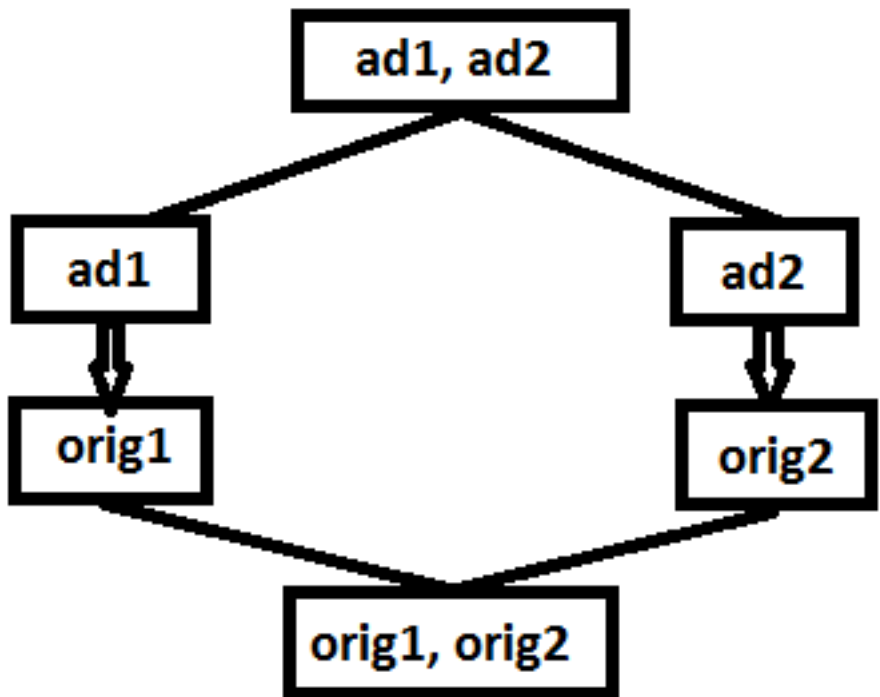
Permasalahan ini adalah suatu permasalahan optimasi yang bertujuan untuk mencari banyak kemungkinan hasil yang ada. Teknik *meet in the middle* dapat diterapkan pada permasalahan ini karena permasalahan ini memiliki kriteria dapat dipecah menjadi beberapa submasalah yang dapat diselesaikan masing-masing tanpa bergantung dengan satu sama lain. Pendekatan *dynamic programming* dapat menyelesaikan masing-masing submasalah dari permasalahan ini karena submasalah yang dihasilkan memiliki kriteria submasalah optimal dan submasalah tumpang tindih.

2.2 Analisa Submasalah Optimal

Pada subbab ini akan dijelaskan mengenai submasalah-submasalah yang jawabannya dapat membangun jawaban akhir. Tujuan utama dari permasalahan yang diberikan adalah untuk mencari jumlah kemungkinan string $orig1$ dan $orig2$ dari string $ad1$ dan $ad2$ yang memiliki jarak $dist(orig1, ad1) + dist(orig2, ad2) = X$. Sebagai contoh, dengan string $ad1 = "c"$ dan $ad2 = "n"$ dan $X = 1$, terdapat 4 kombinasi string $orig1$ dan $orig2$, yaitu $\{\{orig1 = "b", orig2 = "n"\}, \{orig1 = "d", orig2 = "n"\}, \{orig1 = "c", orig2 = "m"\}, \{orig1 = "c", orig2 = "o"\}\}$.

Teknik membagi permasalahan menjadi submasalah yang *independent* sering disebut dengan teknik *meet in the middle*. Teknik ini akan membagi permasalahan yang akan diberikan menjadi dua submasalah untuk masing-masing string yang diberikan. Sehingga string $ad1$ dan $ad2$ dapat diselesaikan masing-masing. Teknik ini tentu akan menghasilkan operasi yang lebih optimal dibandingkan dengan harus menghitung $ad1$ dan $ad2$ menjadi $orig1$ dan $orig2$ secara bersamaan.

Berdasarkan ilustrasi pada gambar 2.1, terlihat bahwa string $ad1$



Gambar 2.1: Ilustrasi teknik *meet in the middle*.

dan *ad2* bersifat *independent* karena memiliki karakteristik dapat diselesaikan tanpa saling bergantung satu sama lain.

Berdasarkan ilustrasi pada gambar 2.2, teknik *meet in the middle* akan membagi proses penyelesaian masalah menjadi tiga tahap, yaitu:

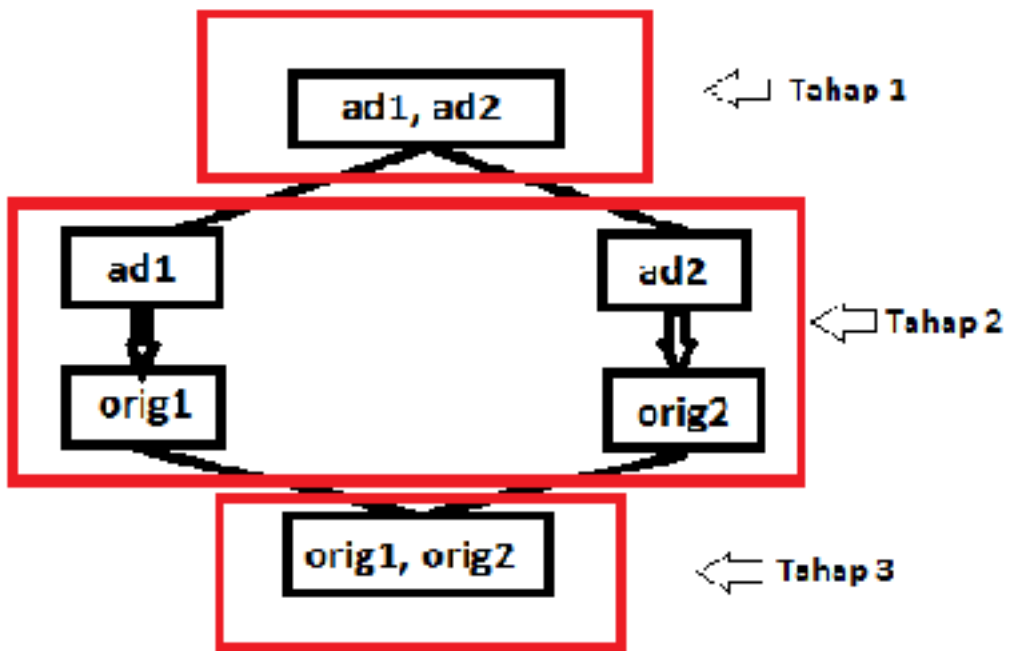
1. Tahap 1: Pembagian permasalahan menjadi submasalah yang *independent*.
2. Tahap 2: Penyelesaian submasalah.
3. Tahap 3: Penggabungan hasil penyelesaian submasalah.

2.2.1 Membagi Permasalahan Menjadi Dua Submasalah yang *Independent*

Pada permasalahan klasik SPOJ 9967 *Playing With Words*, permasalahan akan dibagi menjadi dua submasalah yang *independent*. Permasalahan akan dibagi berdasarkan string yang diberikan. Dimana artinya permasalahan akan dibagi menjadi dua submasalah, yaitu submasalah untuk menghitung banyak kemungkinan string *orig1* dari string *ad1* dan submasalah untuk menghitung banyak kemungkinan string *orig2* dari string *ad2*.

Tahap berikutnya yaitu menyelesaikan masing-masing submasalah. Karena submasalah yang kita pecah tidak saling bergantung satu sama lain, kita dapat menyelesaikannya secara terpisah. Pada permasalahan klasik SPOJ 9967 *Playing With Words*, kita dapat menghitung jumlah kemungkinan string *orig1* dari string *ad1* tanpa mempedulikan apa yang terjadi pada string *ad2*. Begitu pula sebaliknya, kita dapat menghitung jumlah kemungkinan string *orig2* dari string *ad2* tanpa mempedulikan apa yang terjadi pada string *ad1*.

Pada proses transformasi string *orig1* dan *orig2* menjadi string *ad1* dan *ad2*, terdapat proses *replace* yaitu mengganti salah satu karakter pada string *orig1* atau *orig2*. Sehingga untuk dapat menerapkan



Gambar 2.2: Ilustrasi tahapan teknik *meet in the middle*.

metode *meet in the middle* pada permasalahan ini, harus melakukan proses-proses perhitungan berikut:

1. Menghitung jumlah kemungkinan string *orig1* dari string *ad1* tanpa menggunakan operasi *replace*.
2. Menghitung jumlah kemungkinan string *orig1* dari string *ad1* dengan menggunakan operasi *replace*.
3. Menghitung jumlah kemungkinan string *orig2* dari string *ad2* tanpa menggunakan operasi *replace*.
4. Menghitung jumlah kemungkinan string *orig2* dari string *ad2* dengan menggunakan operasi *replace*.

Jawaban akhir yang berupa jumlah kemungkinan string *orig1* dan *orig2* yang mungkin adalah hasil penjumlahan dari hasil perkalian proses 1 dengan proses 4 dengan hasil perkalian proses 2 dengan proses 3.

2.2.2 Submasalah Optimal untuk Menghitung Jumlah Kemungkinan String Awal Tanpa Operasi *Replace*

2.2.3 Submasalah Optimal untuk Menghitung Jumlah Kemungkinan String Awal dengan Operasi *Replace*

2.2.4 Penggabungan Hasil Perhitungan Submasalah

2.3 Pemodelan Relasi Rekurens

2.4 Meet In The Middle

2.5 Dynamic Programming

BAB 3

DESAIN

Pada bab ini akan dibahas tentang desain algoritma untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.

3.1 Desain Umum Sistem

Pada subbab ini akan dijelaskan tentang rancangan umum sistem dalam bentuk *pseudocode*.

3.2 Desain Algoritma

Pada subbab ini akan dijelaskan tentang desain algoritma dengan rinci dalam bentuk *pseudocode* dengan detail masing-masing fungsi.

3.2.1 Desain Fungsi *Preprocess*

Pada bagian ini akan dijelaskan tentang desain fungsi *preprocessing* data-data yang akan digunakan pada fungsi-fungsi lainnya.

3.2.2 Desain Fungsi Perhitungan Tanpa Operasi *Replace*

Pada bagian ini akan dijelaskan tentang desain fungsi perhitungan kemungkinan string tanpa operasi *replace* berdasarkan fungsi rekursen yang telah dijelaskan pada bab 2.

3.2.3 Desain Fungsi Perhitungan Dengan Operasi *Replace*

Pada bagian ini akan dijelaskan tentang desain fungsi perhitungan kemungkinan string dengan operasi *replace* berdasarkan fungsi rekuren yang telah dijelaskan pada bab 2.

3.2.4 Desain Fungsi Penggabungan Hasil Perhitungan Submasalah

Pada bagian ini akan dijelaskan tentang desain fungsi untuk menggabungkan hasil perhitungan dua submasalah yang mana merupakan langkah terakhir dari teknik *meet in the middle* untuk membentuk jawaban akhir dari permasalahan klasik SPOJ 9967 *Playing With Words*.

BAB 4

IMPLEMENTASI

Pada bab ini akan dijelaskan tentang implementasi algoritma berdasarkan desain yang telah dipaparkan pada Bab 3.

4.1 Lingkungan Implementasi

Pada subbab ini akan dipaparkan tentang lingkungan (hardware dan software) implementasi.

4.2 Rancangan Data

Pada subbab ini akan dipaparkan dan dijelaskan tentang rancangan dataset yang akan digunakan untuk menguji algoritma yang telah dibangun.

4.2.1 Data Masukan

Pada bagian ini akan dijelaskan tentang format data masukan yang akan digunakan untuk menguji algoritma yang telah dibangun.

4.2.2 Data Keluaran

Pada bagian ini akan dijelaskan tentang format data keluaran yang dihasilkan oleh algoritma yang dibangun berdasarkan format keluaran yang diberikan pada permasalahan klasik SPOJ 9967 *Playing With Words*.

4.3 Implementasi Algoritma

Pada subbab ini akan dipaparkan *source code* implementasi dari masing-masing fungsi berdasarkan desain yang telah dijelaskan pada Bab 3.

4.3.1 Header-Header yang Diperlukan

Pada bagian ini akan dipaparkan *source code* yang berisi header-header yang diberikan.

4.3.2 Implementasi Variabel Global

Pada bagian ini akan dipaparkan *source code* yang berisi variabel global yang bertujuan memudahkan implementasi desain yang telah disampaikan dan dipaparkan pada Bab 3.

4.3.3 Implementasi Fungsi Main

Pada bagian ini akan dipaparkan *source code* yang berisi Fungsi main yang bertugas memanggil dan mengintegrasikan fungsi-fungsi lainnya untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.

4.3.4 Implementasi Fungsi *Preprocess*

Pada bagian ini akan dipaparkan *source code* yang berisi fungsi *preprocess* yang bertugas untuk memproses data-data awal yang dibutuhkan untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.

4.3.5 Implementasi Fungsi Perhitungan Tanpa Operasi *Replace*

Pada bagian ini akan dipaparkan *source code* yang berisi fungsi perhitungan jumlah karakter tanpa operasi *replace*.

4.3.6 Implementasi Fungsi Perhitungan Dengan Operasi Replace

Pada bagian ini akan dipaparkan *source code* yang berisi fungsi perhitungan jumlah karakter dengan operasi *replace*.

4.3.7 Implementasi Fungsi Penggabungan Hasil Perhitungan Submasalah

Pada bagian ini akan dipaparkan *source code* yang berisi fungsi untuk menghitung jawaban akhir yang merupakan langkah terakhir dari teknik *meet in the middle* untuk menyelesaikan permasalahan klasik SPOJ 9967 *Playing With Words*.

Halaman ini sengaja dikosongkan

BAB 5

UJI COBA DAN EVALUASI

Pada bab ini akan dipaparkan tentang metode dan analisis dari uji coba algoritma yang sudah dibangun pada Bab 4.

5.1 Lingkungan Uji Coba

pada subbab ini akan dipaparkan tentang lingkungan (hardware dan software) uji coba.

5.2 Data Uji Coba

Pada subbab ini akan dipaparkan tentang berbagai sumber data pengujian.

5.2.1 Dataset pada Pengujian Daring SPOJ

Pada bagian ini dijelaskan tentang data dari pengujian daring SPOJ yang akan digunakan untuk uji kebenaran.

5.2.2 Dataset Generator

Pada bagian ini akan dipaparkan dan dijelaskan tentang generator untuk menghasilkan data uji coba untuk menguji performa dan menganalisis pengaruh berbagai variabel terhadap performa algoritma.

5.3 Skenario Uji Coba

Pada subbab ini akan dijelaskan tentang metode-metode pengujian algoritma yang telah dibangun.

5.3.1 Uji Coba Kebenaran

Pada bagian ini akan dijelaskan tentang metode pengujian kebenaran dengan menggunakan bantuan pengujian daring SPOJ.

5.3.2 Uji Coba Peforma

Pada bagian ini akan dijelaskan tentang metode pengujian peforma menggunakan data uji yang telah diproduksi dan menampilkan hasilnya dalam bentuk grafik.

5.4 Analisis Hasil Uji Coba

Pada subbab ini akan dipaparkan dan dijelaskan tentang analisis hasil dari uji coba yang telah dilakukan. Seperti contohnya analisis pengaruh berbagai variabel yang ada terhadap peforma algoritma telah yang dibangun.

BAB 6

KESIMPULAN

6.1 Kesimpulan

Berisi kesimpulan dari hasil analisis algoritma yang telah dirancang.

Halaman ini sengaja dikosongkan