



**UNIVERSITAS INDONESIA**

**PENCOCOKAN UNTAI EKSAK  
DENGAN MENGGUNAKAN  
UKURAN JARAK *HAMMING***

**SKRIPSI**

**MUHAMAD RAFLY FADILLAH  
0606067490**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
PROGRAM STUDI SARJANA MATEMATIKA  
DEPOK  
JUNI 2011**



**UNIVERSITAS INDONESIA**

**PENCOCOKAN UNTAI EKSAK  
DENGAN MENGGUNAKAN  
UKURAN JARAK *HAMMING***

**SKRIPSI**


**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana sains**

**MUHAMAD RAFLY FADILLAH  
0606067490**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
PROGRAM STUDI SARJANA MATEMATIKA  
DEPOK  
JUNI 2011**

## HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.


Nama : Muhamad Rafly Fadillah  
NPM : 0606067490  
Tanda Tangan :   
Tanggal : 09 Juni 2011

## HALAMAN PENGESAHAN

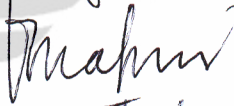
Skripsi ini diajukan oleh :  
Nama : Muhamad Rafly Fadillah  
NPM : 0606067490  
Program Studi : Sarjana Matematika  
Judul Skripsi : Pencocokan Untai Eksak dengan Menggunakan  
Ukuran Jarak *Hamming*

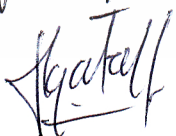
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia

### DEWAN PENGUJI

Pembimbing : Prof. Dr. Djati Kerami (  )

Penguji : Dr. Alhaji Akbar B., M.Sc. (  )

Penguji : Bevina D. Handari, Ph.D. (  )

Penguji : Gatot F. Hertono, Ph.D. (  )

Ditetapkan di : Depok  
Tanggal : 09 Juni 2011

## KATA PENGANTAR

Tiada kata yang patut terucap selain puji syukur kehadirat Allah SWT. yang telah melimpahkan rahmat-Nya yang begitu besar hingga pada akhirnya penulis dapat menyelesaikan skripsi ini dengan baik. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Sains Jurusan Matematika pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.

Dalam prosesnya, tentulah tidak mudah bagi penulis untuk menyelesaikan skripsi tanpa adanya bimbingan serta bantuan dari berbagai pihak. Oleh sebab itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Djati Kerami, selaku dosen pembimbing yang penulis cintai. Sungguh besar jasa beliau yang dengan sabar membimbing penulis dalam menyelesaikan penyusunan skripsi ini. Semoga Allah membalas segala kebaikan beliau. Amin.
2. Bapak Dr. Yudi Satria selaku Kepala Departemen Matematika FMIPA UI beserta staf-stafnya.
3. Mama dan Papa, yang telah membesarkan, mendidik, dan mengayomi penulis tiada henti, sehingga menjadi pendorong bagi penulis untuk menyelesaikan skripsi ini dengan sebaik-baiknya. Semoga Allah selalu melimpahkan rahmat-Nya kepada mereka. Amin.
4. Ibu Dra. Nora Hariadi, M.Si. selaku dosen Pembimbing Akademik selama penulis menuntut ilmu di Departemen Matematika FMIPA UI.
5. Ibu Ruruh Wuryani, selaku teman seperjuangan juga sebagai guru bagi penulis, yang sama-sama berjuang dalam menyelesaikan tugas akhir. Semangat Bu, pasti bisa dan harus bisa!
6. Eka, Elies, dan Ellia, adik-adikku tercinta, tawa canda mereka yang senantiasa memberikan ketenangan batin dalam diri penulis.
7. Alm. H. Sarca (Mbah Olot), Alm. H. Armala (Bapa Kolot), Almrh. Hj. Emun (Mak Olot), Hj. Hamsah (Mak Kolot), Hj. Saerah (Mak Ērah) beserta seluruh keluarga besar penulis, yang tak pernah lupa mendo'akan penulis agar

menjadi orang yang berguna bagi agama, bangsa, dan negara. Amiin.

8. Bu Bela, Bu Suarsih, Bu Siti Nurrohmah, Bu Dian Lestari, Bu Sri Mardiyati, Mbak Helen, Mbak Rahmi, Mbak Milla, Pak Suryadi M.T., Pak Alhadi, serta seluruh Bapak dan Ibu dosen Departemen Matematika FMIPA UI yang telah memberikan motivasi kepada penulis hingga terselesaikannya skripsi ini.
9. Mbak Santi yang selalu bersedia menjawab pertanyaan penulis seputar birokrasi akademik meskipun dalam masa kehamilan, terima kasih Mbak semoga anak Mbak menjadi anak yang berbakti kepada orang tuanya, Amiin. Mbak Rusmi, Pak Salman, Pak Saliman, serta seluruh staf tata usaha Departemen Matematika FMIPA UI, terima kasih semuanya.
10. Farah, Yunita, Rifza, Reza, terima kasih atas segalanya yang telah kalian berikan demi terselesaikannya skripsi ini dengan baik.
11. Seluruh rekan-rekan seperjuangan angkatan 2006 yang telah maupun akan menyelesaikan skripsinya, terima kasih atas semangatnya.
12. Sutisna, Rama, Budi, Teguh, Rendy, Billy, Ali, Pangky, Poe, serta seluruh rekan-rekan 2006-2007 yang sama-sama berjuang menyelesaikan skripsi.
13. Puspa, terima kasih atas semuanya hingga akhirnya penulis bisa menyelesaikan skripsinya.
14. Afni, Dede, Yos, Umbu, Mita, May TA, serta seluruh rekan-rekan angkatan 2002-2011, terima kasih atas semangatnya.
15. Indry, anak-anak “Rukit”, Ibnu, B’jo, Irfan, Kerry, serta para sahabat yang telah banyak membantu menyelesaikan skripsi ini, terima kasih semuanya.

Penulis berharap dan berdo’a kepada Allah SWT. agar senantiasa berkenan membalas segala kebaikan dari semua pihak yang telah membantu. Amiin.

Tak ada gading yang tak retak, tiada satupun makhluk yang sempurna. Begitupun dengan penulis yang tidak luput dari kesalahan. Untuk itu, penulis mohon saran dan kritik untuk penyempurnaan skripsi ini. Semoga skripsi ini bisa memberikan manfaat yang berguna bagi pengembangan ilmu pengetahuan.

**Penulis**

2011

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Muhamad Rafly Fadillah  
NPM : 0606067490  
Program Studi : Sarjana Matematika  
Departemen : Matematika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

**Pencocokan Untai Eksak dengan Menggunakan Ukuran Jarak *Hamming***

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 09 Juni 2011

Yang menyatakan



(Muhamad Rafly Fadillah)

## ABSTRAK

Nama : Muhamad Rafly Fadillah  
Program Studi : Matematika  
Judul : Pencocokan Untai Eksak dengan Menggunakan Ukuran Jarak  
*Hamming*

Masalah pencocokan untaı terbagi menjadi dua yakni pencocokan untaı eksak dan hampiran. Pada skripsi ini, masalah pencocokan untaı yang dibahas adalah pencocokan untaı eksak. Inti dari masalah pencocokan untaı eksak adalah mencari semua posisi kemunculan suatu untaı pada untaı yang lain.

Salah satu ukuran yang membedakan dua buah untaı adalah jarak Hamming. Jika diberikan dua buah untaı, misalkan  $x$  dan  $y$  dengan panjang yang sama, maka jarak Hamming antara keduanya adalah banyaknya karakter pada untaı  $x$  yang berbeda dengan karakter pada untaı  $y$  pada posisi yang bersesuaian.

Dalam pencocokan untaı eksak, panjang untaı yang akan dicocokkan tidak akan selalu sama sehingga jarak Hamming antara keduanya tidak dapat dihitung. Dengan merangkaikan sejumlah untaı hampa pada untaı yang lebih pendek sehingga panjangnya menjadi sama, jarak Hamming antara keduanya barulah dapat dihitung. Dengan pemikiran inilah akhirnya ukuran jarak Hamming digunakan dalam pencocokan untaı eksak untuk mencari semua posisi kemunculan suatu untaı pada untaı lain yang lebih panjang.

Kata Kunci : teori untaı, pencocokan untaı eksak, jarak hamming.  
xiv + 67 halaman; 25 gambar; 2 tabel  
Daftar Pustaka : 7 (1983-2010)



## ABSTRACT

Name : Muhamad Rafly Fadillah  
Program Study : Mathematics  
Title : Exact String Matching by Using Hamming Distance Measure

String matching problem is divided into exact and approximate string matching. In this thesis, we discuss the exact string matching. The main problem of exact string matching is to find all position of a string in other string.

The measure that used in this thesis is Hamming distance. Given two strings namely the  $x$  and  $y$  with the equal length, Hamming distance of the two is the number of positions at which the corresponding characters are different.

In string matching problem, Hamming distance between them can't always be calculated because the strings are not always have the same length. By concating a number of empty string with shortest string until the length to be same, then Hamming distance could be calculated. Finally, Hamming distance measure is used by exact string matching to find all position of a string in other string.

Key Words : string theory, exact string matching, hamming distance.  
xiv + 67 halaman; 25 gambar; 2 tabel  
Bibliography : 7 (1983-2010)

## DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS .....	iii
HALAMAN PENGESAHAN .....	iv
KATA PENGANTAR .....	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI .....	vii
ABSTRAK .....	viii
ABSTRACT .....	ix
DAFTAR ISI .....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN .....	xiv
<b>1. PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah dan Ruang Lingkup .....	3
1.3. Jenis Penelitian dan Metode yang Digunakan.....	3
1.4. Tujuan Penelitian .....	4
<b>2. LANDASAN TEORI .....</b>	<b>5</b>
2.1. Untai .....	5
2.1.1. Definisi Alfabet .....	5
2.1.2. Definisi Untai .....	5
2.1.3. Himpunan Untai .....	6
2.1.4. Panjang Untai dan Posisi Simbol pada Untai.....	7
2.1.5. Operasi pada Untai .....	8
2.1.6. Eksponensiasi pada Untai .....	9
2.1.7. Subuntai, Prefiks, dan Sufiks .....	10
2.1.8. Balikan .....	10
2.1.9. Operasi antar Dua Himpunan Untai .....	11
2.1.10. Penutup Kleene .....	12
2.2. Pencocokan Untai Eksak .....	12
2.3. Jarak .....	13
<b>3. PENCOCOKAN UNTAI EKSAK DENGAN UKURAN JARAK HAMMING .....</b>	<b>15</b>
3.1. Pembentukan Algoritma Pencocokan Untai Eksak .....	15
3.2. Jarak Hamming .....	26
3.2.1. Definisi Jarak Hamming.....	26
3.2.2. Algoritma Jarak Hamming .....	29
3.3. Metode Pencocokan Untai Eksak dengan Ukuran Jarak Hamming.	34
<b>4. IMPLEMENTASI DAN SIMULASI PROGRAM .....</b>	<b>62</b>
4.1. Implementasi .....	62
4.2. Simulasi Program .....	63

<b>5. KESIMPULAN .....</b>	<b>66</b>
<b>DAFTAR PUSTAKA .....</b>	<b>67</b>



## DAFTAR TABEL

Tabel 3.1. Tabel algoritma $jarak\_hamming2(x, y)$ .....	32
Tabel 3.2. Perhitungan jarak Hamming pada contoh 3.10. dengan menggunakan tabel .....	33



## DAFTAR GAMBAR

Gambar 3.1.	Contoh pencocokan untai eksak .....	16
Gambar 3.2.	Pergeseran untai $x$ pada contoh 3.1. ....	17
Gambar 3.3.	Pergeseran untai $x$ pada contoh 3.2. ....	18
Gambar 3.4.	Pergeseran untai $x$ pada contoh 3.3. ....	19
Gambar 3.5.	Pergeseran untai $x$ pada contoh 3.4. ....	20
Gambar 3.6.	Pergeseran untai $x$ pada contoh 3.5. ....	21
Gambar 3.7.	Pergeseran untai $x$ pada contoh 3.6. ....	22
Gambar 3.8.	Contoh kasus terburuk pada algoritma <i>brute_force_pencocokan_untai_eksak(x, y)</i> .....	23
Gambar 3.9.	Pergeseran untai $x$ pada contoh 3.7. ....	25
Gambar 3.10.	Contoh perhitungan jarak Hamming pada contoh 3.8. ....	27
Gambar 3.11.	Perbandingan yang dilakukan pada algoritma <i>jarak_hamming(x, y)</i> .....	30
Gambar 3.12.	Penerapan algoritma <i>jarak_hamming(x, y)</i> pada contoh 3.9....	31
Gambar 3.13.	Kemungkinan posisi untai hampa pada contoh 3.11. ....	35
Gambar 3.14.	Ilustrasi pergeseran untai $x$ .....	35
Gambar 3.15.	Kemungkinan posisi untai hampa pada contoh 3.12. ....	38
Gambar 3.16.	Kemungkinan posisi untai hampa pada contoh 3.13. ....	41
Gambar 3.17.	Kemungkinan posisi untai hampa yang dirangkaikan pada untai <i>den</i> .....	54
Gambar 3.18.	Kemungkinan posisi untai hampa yang dirangkaikan pada untai <i>per</i> .....	58
Gambar 3.19.	Kemungkinan posisi untai hampa yang dirangkaikan pada untai <i>ayam</i> .....	60
Gambar 4.1.	Tampilan awal pada <i>command window</i> .....	63
Gambar 4.2.	Tampilan input untai berikutnya pada <i>command window</i> .....	63
Gambar 4.3.	Tampilan kemungkinan posisi pertama pada <i>command window</i> .....	64
Gambar 4.4.	Tampilan kemungkinan posisi ke-2 hingga ke-4 pada <i>command window</i> .....	64
Gambar 4.5.	Tampilan kemungkinan posisi ke-5 dan ke-6 pada <i>command window</i> .....	65
Gambar 4.6.	Tampilan hasil akhir pada <i>command window</i> .....	65

## DAFTAR LAMPIRAN

Lampiran 1	Program interaktif pencocokan untai eksak lebih dari 1 subuntai .....	<i>CD</i>
Lampiran 2	Program interaktif penghitungan jarak Hamming .....	<i>CD</i>
Lampiran 3	Program interaktif pencocokan untai eksak dengan menggunakan ukuran jarak Hamming .....	<i>CD</i>



# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Dalam teori komputasi, sebuah untai didefinisikan sebagai suatu barisan berhingga dari simbol-simbol atau karakter-karakter anggota himpunan alfabet  $V$ . Misal diberikan sebuah alfabet  $V = \{a, b, \dots, z\}$ , maka untai-untai yang dapat dibentuk sepanjang alfabet  $V$  tersebut antara lain adalah *aabbccdd*, *semangka*, *matematika*, dan sebagainya. Jika  $V = \{0, 1\}$ , maka untai-untai yang dapat dibentuk antara lain ialah 0111011, 00001, 01111, dan lain sebagainya. Sebuah untai bisa saja tidak mengandung satupun karakter, dalam hal ini disebut dengan untai hampa yang dinotasikan dengan  $\epsilon$  atau  $e$ . (Lewis, 1998).

Selain itu, ada sebuah istilah yang cukup erat kaitannya dengan untai, yakni jarak antara dua buah untai. Misalkan  $a$  dan  $b$  anggota dari himpunan tak kosong  $H$ . Jarak antara  $a$  dan  $b$  adalah suatu fungsi  $d$  yang memetakan  $(a, b)$  ke suatu bilangan real nonnegatif, atau  $d: H \times H \rightarrow \mathbb{R}^+ \cup \{0\}$  (Bartle, 2000). Jarak antara  $a$  dan  $b$  dinotasikan dengan  $d(a, b)$  harus memenuhi sifat-sifat nonnegatifitas, simetris, serta memenuhi ketaksamaan segitiga. Sedangkan jarak antara dua buah untai didefinisikan sebagai sebuah ukuran dari banyaknya operasi yang paling sedikit dilakukan pada untai untuk merubah untai tersebut menjadi untai yang lain. Operasi untai itu sendiri bisa berupa penghapusan, penyisipan, maupun penggantian satu atau lebih karakter pada suatu untai.

Salah satu ukuran jarak antara dua buah untai ialah Jarak Hamming (*Hamming Distance*). Nama ini diambil dari nama seorang ilmuwan, yakni Richard Hamming. Ia memperkenalkan ukuran jarak tersebut pada makalahnya yang berjudul *Error detecting and error correcting codes* tahun 1950. Jika diberikan dua buah untai,  $x$  dan  $y$  dengan panjang yang sama, misalkan  $n$ , maka jarak Hamming antara untai  $x$  dan untai  $y$  adalah banyaknya karakter pada untai  $x$  yang berbeda dengan karakter pada untai  $y$  pada posisi yang bersesuaian (*Hamming Distance*, 2011).

Misal diberikan untai  $x = \text{sate}$  dan untai  $y = \text{soto}$ . Dari kedua untai tersebut, terdapat 2 buah karakter pada untai  $x$  yang berbeda dengan karakter pada untai  $y$  pada posisi yang bersesuaian, yakni karakter pada posisi ke-2 dan ke-4, sehingga jarak Hamming antara untai  $x$  dan  $y$  adalah 2.

Setelah melihat beberapa ulasan tentang untai di atas, ada sebuah masalah yang tidak asing lagi didengar dalam ilmu komputer yaitu masalah pencocokan untai (*string matching problem*). Masalah pencocokan untai ini adalah sebuah masalah yang kerap ditemui dalam kegiatan kita sehari-hari khususnya yang berhubungan dengan suatu teks. Contoh implementasi dari masalah pencocokan untai ini adalah pencocokan sebuah untai kata pada *Microsoft Word* ataupun teks editor yang lain. Misalkan, ingin dicari sebuah kata pada sebuah halaman yang kemudian akan diganti dengan kata yang lain, maka dapat kita gunakan tools “*find and replace*” yang terdapat pada *Microsoft Word* ataupun teks editor yang lain.

Dalam kasus yang lebih besar lagi, pencocokan untai digunakan pada website, yakni dengan memasukkan kata kunci sebagaimana yang telah diimplementasikan pada mesin pencari seperti *Yahoo* maupun *Google*. Misalkan ingin dicari sebuah dokumen pada mesin pencari *Google* yang mengandung suatu kata tertentu. Jika diperhatikan, terkadang pengguna internet mengetikkan kata-kata yang salah sehingga *Google* memberikan saran pencarian untuk mencari kata-kata yang benar. Biasanya *Google* memberikan pernyataan “Mungkin maksud anda adalah: ...”. Walaupun demikian, jika dokumen yang mengandung kata yang diinput ada di dalam *database Google*, maka *link* dokumen tersebut tetap ditampilkan.

Masalah pencocokan untai ini terbagi menjadi dua jenis, yakni masalah pencocokan untai eksak dan *approximate*. Pada skripsi ini, masalah pencocokan untai yang dibahas adalah masalah pencocokan untai eksak (*exact string matching problem*). Masalah utama dalam pencocokan untai eksak ini adalah mencari semua posisi kemunculan sebuah untai di dalam untai yang lain (Lewis, 1998). Contohnya, diberikan sebuah untai  $y = \text{jakarta}$  dengan panjang  $|y| = n = 7$  dan untai  $x = \text{akar}$  dengan panjang  $|x| = m = 4$ , maka terlihat bahwa untai  $x$  ada di dalam untai  $y$  pada posisi ke-2.



Pada contoh pencocokan untai di atas, terlihat bahwa panjang untai  $x$  tidak sama dengan panjang untai  $y$ . Jika merujuk kepada definisi jarak Hamming, maka jarak Hamming antara kedua untai tersebut tidak dapat dihitung. Akan tetapi, dengan metode tertentu dua buah untai yang memiliki panjang yang berbeda dapat dicari jarak Hamming-nya. Sehingga pada akhirnya dapat dicari semua posisi kemunculan suatu untai pada untai yang lain berdasarkan ukuran jarak Hamming.

## 1.2. Perumusan Masalah dan Ruang Lingkup

Dari uraian latar belakang di atas, penulis merumuskan beberapa masalah sebagai berikut:

- a. Bagaimanakah teknik mengukur jarak Hamming jika dua buah untai yang diberikan memiliki panjang yang tidak sama?
- b. Bagaimanakah membangun suatu algoritma pencocokan untai eksak menggunakan ukuran jarak Hamming?
- c. Bagaimanakah kompleksitas waktu yang diperlukan oleh algoritma pencocokan untai tersebut?

Adapun ruang lingkup dalam pembahasan tugas akhir ini adalah masalah pencocokan untai eksak dengan panjang untai yang diinput kurang dari atau sama dengan 500 (lima ratus) karakter.

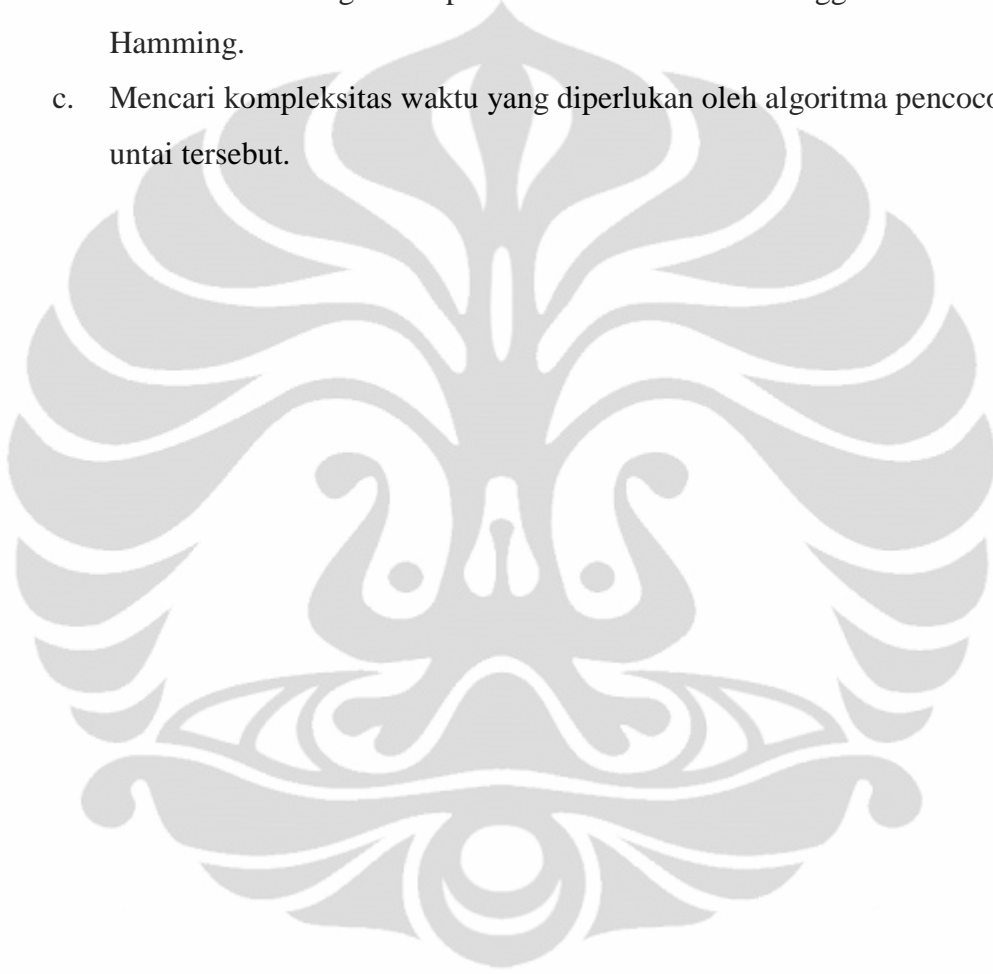
## 1.3. Jenis Penelitian dan Metode yang Digunakan

Jenis penelitian yang digunakan dalam penyusunan tugas akhir ini adalah studi literatur. Sedangkan metode yang digunakan adalah metode kombinatorik, yakni mencari segala macam kemungkinan dari posisi untai hampa  $\wedge$  yang akan ditambahkan pada untai yang lebih pendek sehingga didapatkan posisi untai  $\wedge$  yang akan meminimalkan jarak Hamming. Selain itu, mencari setiap kemungkinan posisi pencocokan untai yang telah diperoleh sebelumnya berdasarkan definisi-definisi yang telah dibuat.

#### 1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

- a. Mencari teknik untuk mengukur jarak Hamming antara dua buah untai yang panjangnya tidak sama.
- b. Membuat suatu algoritma pencocokan untai eksak menggunakan ukuran jarak Hamming.
- c. Mencari kompleksitas waktu yang diperlukan oleh algoritma pencocokan untai tersebut.



## BAB 2 LANDASAN TEORI

Pada bab ini akan dipaparkan beberapa teori dasar tentang untai, pencocokan untai eksak, dan jarak yang dijadikan sebagai landasan dalam penelitian ini. Secara umum, definisi yang digunakan dalam bab ini diambil dari (Lewis & Papadimitriou, 1998)

### 2.1. Untai

Berikut ini adalah beberapa teori dasar tentang untai secara matematis.

#### 2.1.1. Definisi Alfabet

Misal diberikan suatu himpunan berhingga  $V$  yang terdiri dari simbol-simbol. Maka himpunan inilah yang kemudian disebut dengan abjad atau alfabet (*alphabet*) yang terkadang dinotasikan juga dengan  $\Sigma$ . Contohnya, Abjad Latin  $\{a, b, c, \dots, z\}$  ataupun Abjad Yunani  $\{\alpha, \beta, \gamma, \dots, \omega\}$ . Abjad yang terutama sekali berhubungan dengan teori komputasi adalah Abjad Biner  $\{0, 1\}$ .

Dalam kehidupan nyata, terdapat banyak sekali simbol-simbol dengan bentuk yang beragam. Maka untuk penyederhanaan, yang kita gunakan sebagai simbol di sini adalah huruf, angka, dan simbol khusus lain seperti \$ ataupun #.

#### 2.1.2. Definisi Untai

Untai (*string*) sepanjang alfabet  $V$  adalah sebuah barisan berhingga simbol-simbol anggota alfabet  $V$  tersebut. Contohnya, *aabbccdd*, *cgtcttgc*, dan *matematika* adalah untai sepanjang alfabet  $\{a, b, c, \dots, z\}$  sedangkan 0111011, 00001, dan 01111 adalah untai sepanjang  $\{0, 1\}$ . Dalam bahasa sehari-hari, untai yang mempunyai makna dalam bahasa alami disebut juga sebagai suatu kata.

Terkadang sebuah untai hanya mengandung satu simbol saja, maka untai seperti ini disebut dengan untai tunggal. Contohnya,  $a$  adalah untai yang hanya mengandung satu simbol saja yaitu  $a$ . Dalam hal ini, simbol  $a$  bisa disebut sebagai untai  $a$ .

Sebuah untai juga bisa saja tidak mengandung satupun simbol, untai seperti ini disebut dengan untai hampa yang dinotasikan dengan  $\wedge$  atau  $e$ .

### 2.1.3. Himpunan Untai

Himpunan dari semua untai, termasuk untai hampa, sepanjang alfabet  $V$  dinotasikan dengan  $V^*$ . Dalam hal ini,  $V^*$  disebut sebagai penutup (*closure*) dari himpunan  $V$ . Contohnya, jika  $V = \{a, b\}$ , maka penutup dari himpunan  $V$  adalah  $V^* = \{\wedge, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, \dots\}$ . Apabila untai hampa  $\wedge$  tidak dimasukkan dalam himpunan untai tersebut, maka himpunan untai tersebut menjadi  $V^* - \{\wedge\}$  dan ditulis dengan  $V^+$ . Di sini, himpunan hampa  $\emptyset$  atau  $\{\}$  merupakan himpunan yang tidak mengandung satupun untai.

#### **Teorema:**

“Jika  $V$  adalah himpunan alfabet yang berhingga, maka  $V^*$  merupakan himpunan yang tak berhingga tetapi terhitung.”

Untuk membuktikannya, harus ditunjukkan bahwa terdapat pemetaan bijektif  $f: \mathbb{N} \rightarrow V^*$ .

Pertama, ambil sebuah himpunan alfabet  $V = \{a_1, a_2, \dots, a_n\}$ , dengan  $a_1, a_2, \dots, a_n$  yang berbeda dan telah terurut (diatur urutannya). Maka, setiap anggota dari  $V^*$  dapat dicacah dengan cara sebagai berikut:

- a. Untuk setiap  $k \geq 0$ , semua untai dengan panjang  $k$  dicacah sebelum semua untai yang panjangnya  $k + 1$ .
- b. Untai-untai yang panjangnya tepat  $k$  dicacah secara leksikografi, yaitu  $a_{i_1} \dots a_{i_k}$  mendahului  $a_{j_1} \dots a_{j_k}$ , dengan aturan, untuk sembarang  $m$  dengan  $0 \leq m \leq k - 1$ ,  $i_p = j_p$  untuk  $p = 1, \dots, m$ , dan  $i_{m+1} < j_{m+1}$ .

Contohnya, jika diberikan  $V = \{0,1\}$ , maka urutan setiap anggota dari  $V^*$  adalah  $\wedge, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \dots$

Dengan demikian, jika diberikan  $V = \{a_1, a_2\}$ , maka urutannya adalah sebagai berikut:

$\wedge$   
 $a_1$   
 $a_2$   
 $a_1 a_1$   
 $a_1 a_2$   
 $a_2 a_1$   
 $a_2 a_2$   
 $a_1 a_1 a_1$   
 $a_1 a_1 a_2$   
 $a_1 a_2 a_1$   
 $a_1 a_2 a_2$   
 $\dots \dots \dots$

Jika  $V$  adalah Abjad Latin dan urutan dari  $V = \{a_1, a_2, \dots, a_{26}\}$  adalah seperti biasa yaitu  $\{a, b, \dots, z\}$ , maka urutan leksikografi untuk untai-untai yang panjangnya sama diurutkan seperti dalam kamus. Meskipun demikian, aturan yang tercantum pada butir a dan b di atas yang mendaftar untai terpendek sebelum untai yang lebih panjang untuk semua untai anggota  $V^*$  berbeda dengan urutan pada kamus.

#### 2.1.4. Panjang Untai dan Posisi Simbol pada Untai

Panjang dari sebuah untai  $x$  merupakan banyaknya simbol yang membentuk untai  $x$  tersebut, dinotasikan dengan  $|x|$ . Contohnya, jika  $x = 101$ , maka  $|x| = |101| = 3$ , jika  $x = \wedge$ , maka  $|x| = |\wedge| = 0$ .

Sebuah untai  $x$  anggota dari  $V^*$  juga dapat direpresentasikan sebagai suatu fungsi, yaitu  $x: \{1, \dots, |x|\} \rightarrow V$ . Nilai fungsi dari  $x(i)$  untuk  $1 \leq i \leq |x|$ , dengan  $i$  bilangan bulat, adalah simbol pada posisi ke- $i$  dari  $x$ . Contohnya, jika  $x = telepon$ , maka  $x(1) = t$  dan  $x(4) = x(2) = e$ . Pada dasarnya, simbol  $e$

pada posisi ke-4 identik dengan simbol  $e$  pada posisi ke-2. Akan tetapi, untuk membedakan simbol yang identik tersebut pada posisi yang berbeda dalam untai, kita sebut mereka sebagai kejadian yang berbeda dari suatu simbol. Yaitu, simbol  $\sigma$  anggota dari  $V$  muncul pada posisi ke- $i$  dari untai  $x$  anggota  $V^*$  jika  $x(i) = \sigma$ .

Untuk selanjutnya, simbol pada posisi ke- $i$  dari  $x$  akan ditulis dengan  $x_i$  untuk  $1 \leq i \leq |x|$ ,  $i$  bilangan bulat.

### 2.1.5. Operasi pada Untai

Dua buah untai sepanjang alfabet yang sama dapat dirangkai menjadi bentuk ketiga dengan operasi perangkaian (*concatenation*). Perangkaian dari untai  $x$  dan  $y$  yang ditulis dengan  $x \circ y$  atau disingkat menjadi  $xy$ , adalah untai  $x$  yang diikuti oleh untai  $y$ . Misal diberikan untai  $w = x \circ y = xy$  untuk  $w, x$ , dan  $y$  anggota  $V^*$ , sifat-sifat yang berlaku pada perangkaian untai tersebut adalah:

- $|w| = |x| + |y|$ ,
- $w(j) = x(j)$  untuk  $j = 1, \dots, |x|$ , dan
- $w(|x| + j) = y(j)$  untuk  $j = 1, \dots, |y|$ .

Contohnya, jika  $x = 01$  dan  $y = 001$ , maka  $x \circ y = 01 \circ 001 = 01001$ , jika  $x = \text{mata}$  dan  $y = \text{hari}$ , maka  $x \circ y = \text{mata} \circ \text{hari} = \text{matahari}$ .

Berdasarkan definisi operasi perangkaian tersebut, didapat  $x \circ \wedge = \wedge \circ x = x$  untuk sembarang untai  $x$ .

Diketahui bahwa monoid  $(G, *)$  adalah suatu himpunan tak kosong  $G$  dengan operasi ‘ $*$ ’ yang bersifat:

- Tertutup terhadap operasi ‘ $*$ ’.
- Asosiatif,  $(x * y) * z = x * (y * z)$  untuk setiap  $x, y$ , dan  $z$  anggota  $G$ .
- Memiliki elemen identitas  $e$ , identitas sehingga  $x * e = e * x = x$ , untuk setiap  $x$  anggota  $G$ .

Dapat dibuktikan bahwa himpunan  $V^*$  dengan operasi perangkaian merupakan monoid karena memenuhi sifat-sifat berikut:

- Tertutup terhadap operasi perangkaian.

Bukti: Ambil  $x$  dan  $y$  anggota  $V^*$  dengan panjang masing-masing  $m$  dan  $n$ .

$$\text{Maka, } x = x_1x_2 \dots x_m \text{ dan } y = y_1y_2 \dots y_n$$

dengan  $x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n$  adalah simbol-simbol anggota  $V$ .

Sehingga  $xy = x_1x_2 \dots x_mx_1y_2 \dots y_n$  merupakan untai yang terdiri dari simbol-simbol anggota  $V$  juga.

Akibatnya,  $xy$  juga anggota  $V^*$ .

- b. Asosiatif, yaitu:  $(xy)z = x(yz)$  untuk sembarang untai  $x, y$ , dan  $z$ .

Bukti: Ambil  $x, y$ , dan  $z$  anggota  $V^*$  dengan panjang masing-masing  $p, q$ , dan  $r$ .

Maka,  $x = x_1 \dots x_p, y = y_1 \dots y_q$ , dan  $z = z_1 \dots z_r$ ,

dengan  $x_1 \dots x_p, y_1 \dots y_q, z_1 \dots z_r$  adalah simbol-simbol anggota  $V$ .

$$\begin{aligned} \text{Sehingga : } (xy)z &= (x_1 \dots x_p y_1 \dots y_q) z_1 \dots z_r \\ &= x_1 \dots x_p y_1 \dots y_q z_1 \dots z_r \\ &= x_1 \dots x_p (y_1 \dots y_q z_1 \dots z_r) \\ &= x(yz) \end{aligned}$$

- c. Memiliki elemen identitas, yaitu untai hampa  $\wedge$ .

Sedemikian sehingga  $x \wedge = \wedge x = x$ , untuk setiap untai  $x$  anggota  $V^*$ .

#### 2.1.6. Eksponensiasi pada Untai

Untuk setiap untai  $x$  dan setiap bilangan bulat nonnegatif  $i$ , untai  $x^i$  didefinisikan secara induktif sebagai:

$$\begin{aligned} x^0 &= \wedge, & \text{yaitu untai hampa} \\ x^{i+1} &= x^i \circ x, & \text{untuk setiap } i = 0, 1, 2, \dots \end{aligned}$$

Sehingga, untuk  $i = 0$ , didapat  $x^1 = x$ . Contoh lain, jika  $x = do$  dan  $i = 1$  maka berdasarkan definisi di atas didapat:

$$\begin{aligned} (do)^{1+1} &= (do)^1 \circ do & \text{dengan } i = 1 \\ (do)^2 &= ((do)^0 \circ do) \circ do & \text{dengan } i = 0 \\ &= (\wedge \circ do) \circ do \\ &= do \circ do \\ &= dodo \end{aligned}$$

### 2.1.7. Subuntai, Prefiks, dan Sufiks

Suatu untai  $z$  dikatakan subuntai (*substring*) dari untai  $w$  jika dan hanya jika terdapat untai  $x$  dan  $y$  sedemikian sehingga  $w = xzy$ . Contohnya, jika diberikan untai  $z = \text{masuk}$  dan untai  $w = \text{pemasukan}$ , maka terdapat untai  $x = \text{pe}$  dan untai  $y = \text{an}$  sedemikian sehingga  $w = \text{pemasukan} = xzy$ . Dengan kata lain, untai  $z$  adalah subuntai dari untai  $w$ .

Untai  $x$  dan  $y$  dapat pula berupa untai hampa  $\epsilon$  baik sendiri-sendiri maupun secara bersamaan, sehingga setiap untai adalah subuntai dari dirinya sendiri. Dengan mengambil  $x = w$  dan  $z = y = \epsilon$ , terlihat bahwa  $w = w\epsilon\epsilon = xzy$ , sehingga untai hampa  $\epsilon$  adalah subuntai dari setiap untai. Suatu untai bisa saja memiliki subuntai yang sama. Contohnya, untai *ababab* memiliki tiga kali kemunculan untai *ab* dan dua kali kemunculan untai *abab*.

Jika  $w = zy$  untuk suatu  $y$ , maka  $z$  disebut awalan atau prefiks dari  $w$ . Sedangkan, jika  $w = xz$  untuk suatu  $x$ , maka  $z$  disebut akhiran atau sufiks dari  $w$ . Contohnya, jika diberikan  $z = \text{mata}$  dan  $w = \text{matahari}$ , maka terdapat  $y = \text{hari}$  sedemikian sehingga  $w = \text{matahari} = zy$ , maka  $z$  adalah awalan dari  $w$ , sedangkan jika diberikan  $w = \text{kacamata}$ , maka terdapat  $x = \text{kaca}$  sedemikian sehingga  $w = \text{kacamata} = xz$ , maka  $z$  adalah akhiran dari  $w$ .

### 2.1.8. Balikan

Balikan (*reversal*) dari untai  $x$ , dinotasikan dengan  $x'$ , adalah untai yang dieja dari belakang ke depan. Contohnya, jika diberikan  $x = \text{kursi}$ , maka  $x' = (\text{kursi})' = \text{isruk}$ .

Definisi formal dapat ditulis secara induksi sebagai berikut:

- Jika  $x$  adalah untai dengan panjang 0, maka  $x' = x = \epsilon$ .
- Jika  $x$  adalah untai dengan panjang  $n + 1 > 0$ , maka  $x = ua$  untuk suatu  $a$  anggota himpunan alfabet  $V$ ,  $u$  anggota  $V^*$ , dan  $x' = au'$ .

Berdasarkan definisi di atas, dapat dibuktikan bahwa untuk sembarang untai  $x$  dan  $y$ , berlaku  $(xy)' = y'x'$  dengan menggunakan induksi matematika, yakni sebagai berikut,



Langkah Awal : Ambil  $x$  anggota  $V^*$ . Jika  $|y| = 0$ , maka  $y = \wedge$ , dan

$$\begin{aligned}(xy)' &= (x\wedge)' \text{ karena } y = \wedge \\ &= x' \text{ karena } x\wedge = x \\ &= \wedge x' \text{ definisi perangkaian untai hampa} \\ &= \wedge' x' \text{ definisi balikan untai hampa } (\wedge' = \wedge) \\ &= y'x' \text{ karena } y = \wedge\end{aligned}$$

Hipotesis induksi : Jika  $|y| \leq n$ , maka  $(xy)' = y'x'$ .

Langkah induksi : Misalkan  $|y| = n + 1$ , maka  $y = ua$  untuk  $u$  anggota  $V^*$  dan  $a$  anggota  $V$ . Sehingga  $|u| = n$ .

$$\begin{aligned}(xy)' &= (x(ua))' \text{ karena } y = ua \\ &= ((xu)a)' \text{ karena perangkaian bersifat asosiatif} \\ &= a(xu)' \text{ dengan definisi balikan dari } (wu)a \\ &= au'x' \text{ dengan hipotesis induksi } (|u| = n) \\ &= (ua)'x' \text{ dengan definisi balikan dari } ua \\ &= y'x' \text{ karena } y = ua\end{aligned}$$

Terbukti bahwa untuk sembarang untai  $x$  dan  $y$ , berlaku  $(xy)' = y'x'$ . Contohnya,  $(karp\text{et})' = (p\text{et})'(k\text{ar})' = t\text{epr}ak$ .

### 2.1.9. Operasi antar Dua Himpunan Untai

Misalkan  $A$  dan  $B$  adalah subhimpunan dari  $V^*$ , dengan  $V$  adalah suatu alfabet. Perangkaian dari  $A$  dan  $B$ , dinotasikan dengan  $AB$ , adalah himpunan dari semua untai yang berbentuk  $xy$  dengan  $x$  adalah untai anggota  $A$  dan  $y$  adalah untai anggota  $B$ . Contohnya, jika diberikan himpunan  $A = \{0, 11\}$  dan  $B = \{1, 10, 110\}$ , maka  $AB = \{01, 010, 0110, 111, 1110, 11110\}$  dan  $BA = \{10, 111, 100, 1011, 1100, 11011\}$ . Dari contoh tersebut terlihat bahwa  $AB \neq BA$ .

Berdasarkan definisi perangkaian dua himpunan untai di atas, dapat didefinisikan  $A^n$  untuk  $n = 0, 1, 2, \dots$  secara rekursif sebagai berikut:

$$\begin{aligned}A^0 &= \{\wedge\} \\ A^{n+1} &= A^n A, \quad \text{untuk } n = 0, 1, 2, \dots\end{aligned}$$

Contohnya, jika diberikan  $A = \{1, 00\}$ , maka:

$$A^0 = \{\epsilon\}$$

$$A^1 = A^0 A = \{\epsilon\}A = A = \{1, 00\}$$

$$A^2 = A^1 A = \{1, 00\}A = \{11, 100, 001, 0000\}$$

$$\begin{aligned} A^3 &= A^2 A = \{11, 100, 001, 0000\}A \\ &= \{111, 1100, 1001, 10000, 0011, 00100, 00001, \\ &\quad 000000\} \end{aligned}$$

### 2.1.10. Penutup Kleene

Misalkan  $A$  adalah subhimpunan dari  $V^*$ , maka Penutup Kleene (*Kleene Closure*) dari  $A$ , ditulis  $A^*$ , adalah himpunan yang mengandung semua rangkaian dari untai-untai anggota  $A$ . (Rosen, 2009). Atau:

$$A^* = \bigcup_{k=0}^{\infty} A^k \quad (2.1)$$

Contohnya, jika diberikan untai  $A = \{0\}$ ,  $B = \{0, 1\}$ ,  $C = \{11\}$ , dan  $D = \{00, 01\}$ , maka Penutup Kleene dari  $A$  adalah perangkaian untai 0 dengan dirinya sendiri yaitu  $A^* = \{0^n \mid n = 0, 1, 2, \dots\}$ , penutup Kleene dari  $B$  adalah sembarang perangkaian 0 ataupun 1, dalam hal ini adalah himpunan semua untai sepanjang abjad biner  $V = \{0, 1\}$ , penutup Kleene dari  $C$  adalah perangkaian untai 11 dengan dirinya sendiri. Sehingga  $C^*$  adalah himpunan untai yang terdiri dari sejumlah genap 1 yaitu  $C^* = \{1^{2n} \mid n = 0, 1, 2, \dots\}$ , dan penutup Kleene dari  $D$  adalah himpunan untai yang terdiri dari perangkaian untai 00 dengan untai 01 dan anggota  $D$  itu sendiri yaitu  $D^* = \{00, 01, 0000, 0001, 0100, 0101, \dots\}$ .

Contoh lain misalnya  $E = \{a, b\}$ . Maka penutup Kleene dari  $E$  adalah  $E = \{a, b, aa, ab, ba, bb, aaa, aab, aba, abb, \dots\}$ .

## 2.2. Pencocokan Untai Eksak

Diberikan dua buah untai, misal  $x$  dan  $y$ , masing-masing sepanjang alfabet  $V$ . Misalkan untai  $y = y_1 y_2 \dots y_n$  dengan panjang  $n$  dan untai  $x = x_1 x_2 \dots x_m$  dengan panjang  $m$ ,  $m \leq n$ . Maka, masalah pencocokan untai eksak adalah

menemukan apakah untai  $x$  ada di dalam untai  $y$ . Jika ada, akan dicari semua posisi kemunculan untai  $x$  pada untai  $y$  tersebut. Contohnya, jika diberikan  $y = atcgtcacgtc$  dan  $x = cgt$  maka untai  $x$  ada di dalam untai  $y$  yaitu:

$$y = at\underline{cgt}c\underline{acg}tc$$

Dalam hal ini, untai  $x$  disebut juga sebagai subuntai dari  $y$ . Sedangkan jika diberikan untai lain yaitu  $z = agt$  maka untai  $z$  tidak ada di dalam untai  $y$ .

### 2.3. Jarak

Diberikan suatu himpunan tak kosong  $H$  dan dua buah elemen  $a$  dan  $b$  anggota himpunan  $H$  tersebut. Dalam bukunya, Bartle & Sherbert (2000) mengatakan bahwa jarak antara  $a$  dan  $b$  yang dinotasikan dengan  $d(a, b)$  didefinisikan sebagai fungsi  $d$  yang memetakan  $(a, b)$  ke suatu bilangan real nonnegatif, atau  $d: H \times H \rightarrow \mathbb{R}^+ \cup \{0\}$ , yang memenuhi sifat-sifat berikut:

- a.  $d(a, b) \geq 0$
- b.  $d(a, b) = 0$  jika  $a = b$
- c.  $d(a, b) = d(b, a)$
- d.  $d(a, c) \leq d(a, b) + d(b, c)$  untuk  $c$  anggota  $H$

Salah satu contoh fungsi jarak dalam matematika adalah jarak *Euclid* yang didefinisikan sebagai berikut:

- a. Jarak Euclid di  $\mathbb{R}$

Misalkan, ambil dua buah bilangan real  $a$  dan  $b$  anggota himpunan  $\mathbb{R}$ , maka jarak antara  $a$  dan  $b$  adalah:

$$d(a, b) = |a - b| \quad (2.2)$$

- b. Jarak Euclid di  $\mathbb{R}^2$

Misalkan, ambil dua buah titik  $A(a_1, a_2)$  dan  $B(b_1, b_2)$  anggota himpunan  $\mathbb{R}^2$ , maka jarak antara  $A$  dan  $B$  adalah:

$$d(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \quad (2.3)$$

c. Jarak Euclid di  $\mathbb{R}^3$ 

Misalkan, ambil dua buah vektor  $\bar{p}(a_1, a_2, a_3)$  dan  $\bar{q}(b_1, b_2, b_3)$  anggota himpunan  $\mathbb{R}^3$ , maka jarak antara  $\bar{p}$  dan  $\bar{q}$  adalah:

$$d(\bar{p}, \bar{q}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2} \quad (2.4)$$

d. Jarak Euclid di  $\mathbb{R}^n$ 

Jika diberikan dua buah vektor  $\bar{p}(a_1, a_2, \dots, a_n)$  dan  $\bar{q}(b_1, b_2, \dots, b_n)$  anggota himpunan  $\mathbb{R}^n$ , maka jarak antara  $\bar{p}$  dan  $\bar{q}$  adalah:

$$d(\bar{p}, \bar{q}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}$$

atau

$$d(\bar{p}, \bar{q}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (2.5)$$

### BAB 3

## PENCOCOKAN UNTAI EKSAK DENGAN UKURAN JARAK HAMMING

Pada dasarnya, algoritma pencocokan untai eksak merupakan suatu algoritma yang bertujuan untuk menemukan posisi dari suatu untai di dalam untai yang lain. Dalam melakukan proses pencarian, banyak sekali pendekatan yang dapat dilakukan dan pendekatan-pendekatan tersebut memiliki kelebihan dan kekurangannya masing-masing. Dalam bab ini, dibahas mengenai sebuah metode pencocokan untai eksak dengan menggunakan jarak Hamming sebagai ukurannya, kompleksitas algoritma dari metode pencocokan untai tersebut, serta sifat-sifat yang diperoleh.

#### 3.1. Pembentukan Algoritma Pencocokan Untai Eksak

Berdasarkan definisi yang telah dijelaskan dalam bab 2 sebelumnya, suatu untai sepanjang alfabet  $V$  adalah sebuah barisan berhingga simbol-simbol anggota alfabet  $V$  tersebut. Untuk mempermudah pemahaman, maka simbol-simbol pembentuk untai ini akan kita sebut sebagai karakter.

Misal diberikan dua buah untai,  $x = ong$  dan  $y = lontong$ , harus diperlihatkan apakah untai  $x$  ada di dalam untai  $y$ . Jika ya, maka harus ditunjukkan pula di mana posisi kemunculan untai  $x$  tersebut di dalam untai  $y$ . Berikut ini adalah penjelasannya.

$$y = lontong \quad \rightarrow |y| = 7$$

$$x = ong \quad \rightarrow |x| = 3$$

- Periksa apakah  $x_1$  sama dengan  $y_1$ , ternyata  $x_1 = o \neq l = y_1$ . Maka, untai  $x$  digeser satu langkah ke kanan.
- Periksa apakah  $x_1$  sama dengan  $y_2$ , ternyata  $x_1 = o = y_2$ . Maka, periksa karakter berikutnya dari untai  $x$  dan  $y$ .
  - Apakah  $x_2$  sama dengan  $y_3$ , ternyata  $x_2 = n = y_3$ . Maka, periksa karakter berikutnya dari untai  $x$  dan  $y$ .

- Apakah  $x_3$  sama dengan  $y_4$ , ternyata  $x_3 = g \neq t = y_4$ . Maka, untai  $x$  digeser kembali satu langkah ke kanan dan periksa kembali karakter pertama dari untai  $x$  dengan karakter yang bersesuaian dari untai  $y$ .
- Periksa apakah  $x_1$  sama dengan  $y_3$ , ternyata  $x_1 = o \neq n = y_3$ . Maka, untai  $x$  digeser kembali satu langkah ke kanan.
- Periksa apakah  $x_1$  sama dengan  $y_4$ , ternyata  $x_1 = o \neq t = y_4$ . Maka, untai  $x$  digeser kembali satu langkah ke kanan.
- Periksa apakah  $x_1$  sama dengan  $y_5$ , ternyata  $x_1 = o = y_5$ . Maka, periksa karakter berikutnya dari untai  $x$  dan  $y$ .
  - Apakah  $x_2$  sama dengan  $y_6$ , ternyata  $x_2 = n = y_6$ . Maka, periksa karakter berikutnya dari untai  $x$  dan  $y$ .
  - Apakah  $x_3$  sama dengan  $y_7$ , ternyata  $x_3 = g = y_7$ . Karena semua karakter dalam untai  $x$  sudah diperiksa, maka disimpulkan bahwa untai  $x$  ada di dalam untai  $y$  pada posisi ke-5 atau untai  $x$  adalah subuntai dari untai  $y$ .

Posisi Karakter :	1	2	3	4	5	6	7	
$y =$	<i>l</i>	<i>o</i>	<i>n</i>	<i>t</i>	<i>o</i>	<i>n</i>	<i>g</i>	
Posisi ke-1 :		<i>o</i>	<i>n</i>	<i>g</i>				
Posisi ke-2 :			<i>o</i>	<i>n</i>	<i>g</i>			
Posisi ke-3 :				<i>o</i>	<i>n</i>	<i>g</i>		
Posisi ke-4 :					<i>o</i>	<i>n</i>	<i>g</i>	
Posisi ke-5 :						<i>o</i>	<i>n</i>	<i>g</i>

**Gambar 3.1.** Contoh pencocokan untai eksak

Secara umum, misal diberikan untai  $x = x_1x_2 \dots x_m$  dan  $y = y_1y_2 \dots y_n$  dengan  $m \leq n$ , ingin dibuat suatu algoritma yang dapat menentukan apakah untai  $x$  ada di dalam untai  $y$ , dan dapat menunjukkan di mana posisinya jika ada. Dengan asumsi bahwa jika untai  $x$  ada di dalam untai  $y$ , maka hanya terdapat satu subuntai dari  $y$  yang sama dengan untai  $x$ . Jika diketahui  $m = 1$ , berarti untai  $x$  hanya terdiri dari satu karakter saja yaitu  $x = x_1$ , maka algoritma yang dapat dibentuk adalah sebagai berikut:

**Algoritma brute\_force\_pencocokan\_untai\_eksak1(x, y)****Input** : Untai  $x$  dengan panjang 1 dan unta  $y$  dengan panjang  $n \geq 1$ .**Output** : Posisi karakter dari unta  $y$  yang sama dengan unta  $x$ , atau 0 jika tidak ada karakter dari unta  $y$  yang sama dengan unta  $x$ .

```

1. posisi = 0
2. for j = 1 to n
3.     if  $x_1 = y_j$  then
4.         posisi = j
5.         break
6.     else
7.         next j
8.     end if
9. end for
10. display posisi /* posisi karakter dari unta y yang sama dengan unta x. */

```

**Contoh 3.1.**Jika diberikan:  $x = p \rightarrow |x| = m = 1$  $y = tempe \rightarrow |y| = n = 5$ 

Maka, perjalanan algoritmanya adalah sebagai berikut:

 $posisi = 0$ .

- $j = 1, x_1 = p \neq t = y_1$ , maka lanjutkan  $j$ .
- $j = 2, x_1 = p \neq e = y_2$ , maka lanjutkan  $j$ .
- $j = 3, x_1 = p \neq m = y_3$ , maka lanjutkan  $j$ .
- $j = 4, x_1 = p = y_4$ , maka  $posisi = j = 4$ .

Output : 4. Artinya, unta  $x$  ada di dalam unta  $y$  pada posisi ke-4. Sehingga  $x$  adalah subunta dari  $y$ .

Posisi Karakter :	1	2	3	4	5
$y =$	$t$	$e$	$m$	<b><math>p</math></b>	$e$
Posisi ke-1 :	$p$				
Posisi ke-2 :		$p$			
Posisi ke-3 :			$p$		
Posisi ke-4 :				<b><math>p</math></b>	

**Gambar 3.2.** Pergeseran unta  $x$  pada contoh 3.1.

**Contoh 3.2.**

Jika diberikan:  $x = d \quad \rightarrow |x| = m = 1$

$y = \text{tempe} \quad \rightarrow |y| = n = 5$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$posisi = 0$ .

- $j = 1, x_1 = d \neq t = y_1$ , maka lanjutkan  $j$ .
- $j = 2, x_1 = d \neq e = y_2$ , maka lanjutkan  $j$ .
- $j = 3, x_1 = d \neq m = y_3$ , maka lanjutkan  $j$ .
- $j = 4, x_1 = d \neq p = y_4$ , maka lanjutkan  $j$ .
- $j = 5 = n, x_1 = d \neq e = y_5$ , maka lanjutkan  $j$ .
- $j = 6 > 5 = n$ , maka selesai.

Output : 0. Artinya, untai  $x$  tidak ada di dalam untai  $y$ . Sehingga  $x$  bukan subuntai dari  $y$ .

Posisi Karakter :	1	2	3	4	5
$y =$	$t$	$e$	$m$	$p$	$e$
Posisi ke-1 :	$d$				
Posisi ke-2 :		$d$			
Posisi ke-3 :			$d$		
Posisi ke-4 :				$d$	
Posisi ke-5 :					$d$

**Gambar 3.3.** Pergeseran untai  $x$  pada contoh 3.2.

Jika diketahui  $m = 2$ , berarti untai  $x$  terdiri dari dua buah karakter yaitu  $x_1$  dan  $x_2$  sehingga  $x = x_1x_2$ , maka algoritma yang dapat dibentuk adalah sebagai berikut:

**Algoritma brute\_force\_pencocokan\_untai\_eksak2( $x, y$ )**

**Input** : Untai  $x$  dengan panjang 2 dan untai  $y$  dengan panjang  $n \geq 2$ .

**Output** : Posisi karakter awal subuntai dari untai  $y$  yang sama dengan untai  $x$ , atau 0 jika tidak ada subuntai dari untai  $y$  yang sama dengan untai  $x$ .

1.  $posisi = 0$
2. **for**  $j = 1$  **to**  $n - 1$
3.     **if**  $x_1 = y_j$  **then**
4.         **if**  $x_2 = y_{j+1}$  **then**
5.              $posisi = j$



```

6.          break
7.          else
8.          next j
9.          end if
10.         else
11.          next j
12.         end if
13.     end for
14. display posisi /* posisi karakter awal subuntai dari untai y yang sama
                    dengan untai x. */

```

### Contoh 3.3.

Jika diberikan:  $x = mp \rightarrow |x| = m = 2$

$y = tempe \rightarrow |y| = n = 5$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$posisi = 0$ .

- $j = 1, x_1 = m \neq t = y_1$ , maka lanjutkan  $j$ .
- $j = 2, x_1 = m \neq e = y_2$ , maka lanjutkan  $j$ .
- $j = 3, x_1 = m = y_3$ .  
 $x_2 = p = y_4$ , maka  $posisi = j = 3$ .

Output : 3. Artinya, untai  $x$  ada di dalam untai  $y$  pada posisi ke-3. Sehingga  $x$  adalah subuntai dari  $y$ .

Posisi Karakter :	1	2	3	4	5
$y =$	$t$	$e$	<b><math>m</math></b>	<b><math>p</math></b>	$e$
Posisi ke-1 :	$m$	$p$			
Posisi ke-2 :		$m$	$p$		
Posisi ke-3 :			<b><math>m</math></b>	<b><math>p</math></b>	

**Gambar 3.4.** Pergeseran untai  $x$  pada contoh 3.3.

### Contoh 3.4.

Jika diberikan:  $x = mo \rightarrow |x| = m = 2$

$y = tempe \rightarrow |y| = n = 5$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$posisi = 0$ .

- $j = 1, x_1 = m \neq t = y_1$ , maka lanjutkan  $j$ .
- $j = 2, x_1 = m \neq e = y_2$ , maka lanjutkan  $j$ .
- $j = 3, x_1 = m = y_3$ .  
 $x_2 = o \neq p = y_4$ , maka lanjutkan  $j$ .
- $j = 4 = n - 1, x_1 = m \neq e = y_4$ , maka lanjutkan  $j$ .
- $j = 5 > 4 = n - 1$ , maka selesai.

Output : 0. Artinya, untai  $x$  tidak ada di dalam untai  $y$ . Sehingga  $x$  bukan subuntai dari  $y$ .

Posisi Karakter :	1	2	3	4	5
$y =$	$t$	$e$	$m$	$p$	$e$
Posisi ke-1 :	$m$	$o$			
Posisi ke-2 :		$m$	$o$		
Posisi ke-3 :			$m$	$o$	
Posisi ke-4 :				$m$	$o$

Gambar 3.5. Pergeseran untai  $x$  pada contoh 3.4.

Untuk  $m$  secara umum, berarti untai  $x$  terdiri dari  $m$  buah karakter yaitu  $x_1, x_2, \dots, x_m$  sehingga  $x = x_1x_2 \dots x_m$ , maka algoritma yang dapat dibentuk adalah sebagai berikut:

**Algoritma brute\_force\_pencocokan\_untai\_eksak( $x, y$ )**

**Input** : Untai  $x$  dengan panjang  $m$  dan untai  $y$  dengan panjang  $n$ .  $m > 0$ ,  $n > 0$ , dan  $m \leq n$ .

**Output** : Posisi karakter awal subuntai dari untai  $y$  yang sama dengan untai  $x$ , atau 0 jika tidak ada subuntai dari untai  $y$  yang sama dengan untai  $x$ .

1.  $posisi = 0$
2. **for**  $j = 1$  **to**  $n - (m - 1)$
3.      $i = 1$
4.     **while**  $i \leq m$  **and**  $x_i = y_{i+j-1}$  **do**
5.          $i = i + 1$
6.     **end while**
7.     **if**  $i > m$  **then**
8.          $posisi = j$
9.         **break**
10.    **else**
11.    **next j**
12. **end if**

13. **end for**  
 14. **display posisi** /\* posisi karakter awal subuntai dari untai y yang sama dengan untai x. \*/

### Contoh 3.5.

Jika diberikan:  $x = top \quad \rightarrow |x| = m = 3$

$y = ketoprak \quad \rightarrow |y| = n = 8$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$posisi = 0$ .

- $j = 1, i = 1 \leq 3 = m$  tetapi  $x_1 = t \neq k = y_1$ , maka lanjutkan  $j$ .
- $j = 2, i = 1 \leq 3 = m$  tetapi  $x_1 = t \neq e = y_2$ , maka lanjutkan  $j$ .
- $j = 3, i = 1 \leq 3 = m$  dan  $x_1 = t = y_3$ , maka  $i = 2$ .  
 $i = 2 \leq 3 = m$  dan  $x_2 = o = y_4$ , maka  $i = 3$ .  
 $i = 3 \leq 3 = m$  dan  $x_3 = p = y_5$ , maka  $i = 4$ .  
 $i = 4 > 3 = m$ , maka  $posisi = j = 3$ .

Output : 3. Artinya, untai  $x$  ada di dalam untai  $y$  pada posisi ke-3. Sehingga  $x$  adalah subuntai dari  $y$ .

Posisi Karakter :	1	2	3	4	5	6	7	8
$y =$	<i>k</i>	<i>e</i>	<i>t</i>	<i>o</i>	<i>p</i>	<i>r</i>	<i>a</i>	<i>k</i>
Posisi ke-1 :	<i>t</i>	<i>o</i>	<i>p</i>					
Posisi ke-2 :		<i>t</i>	<i>o</i>	<i>p</i>				
Posisi ke-3 :			<i>t</i>	<i>o</i>	<i>p</i>			

**Gambar 3.6.** Pergeseran untai  $x$  pada contoh 3.5.

### Contoh 3.6.

Jika diberikan:  $x = topan \quad \rightarrow |x| = m = 5$

$y = ketoprak \quad \rightarrow |y| = n = 8$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$posisi = 0$ .

- $j = 1, i = 1 \leq 5 = m$  tetapi  $x_1 = t \neq k = y_1$ , maka lanjutkan  $j$ .
- $j = 2, i = 1 \leq 5 = m$  tetapi  $x_1 = t \neq e = y_2$ , maka lanjutkan  $j$ .
- $j = 3, i = 1 \leq 5 = m$  dan  $x_1 = t = y_3$ , maka  $i = 2$ .

$i = 2 \leq 5 = m$  dan  $x_2 = o = y_4$ , maka  $i = 3$ .

$i = 3 \leq 5 = m$  dan  $x_3 = p = y_5$ , maka  $i = 4$ .

$i = 4 \leq 5 = m$  tetapi  $x_4 = a \neq r = y_6$ , maka lanjutkan  $j$ .

- $j = 4 = n - (m - 1)$ ,  $i = 1 \leq 5 = m$  tetapi  $x_1 = t \neq o = y_4$ , maka lanjutkan  $j$ .
- $j = 5 > 4 = n - (m - 1)$ , maka selesai.

Output : 0. Artinya, untai  $x$  tidak ada di dalam untai  $y$ . Sehingga  $x$  bukan subuntai dari  $y$ .

Posisi Karakter :	1	2	3	4	5	6	7	8
$y =$	<i>k</i>	<i>e</i>	<i>t</i>	<i>o</i>	<i>p</i>	<i>r</i>	<i>a</i>	<i>k</i>
Posisi ke-1 :	<i>t</i>	<i>o</i>	<i>p</i>	<i>a</i>	<i>n</i>			
Posisi ke-2 :		<i>t</i>	<i>o</i>	<i>p</i>	<i>a</i>	<i>n</i>		
Posisi ke-3 :			<i>t</i>	<i>o</i>	<i>p</i>	<i>a</i>	<i>n</i>	
Posisi ke-4 :				<i>t</i>	<i>o</i>	<i>p</i>	<i>a</i>	<i>n</i>

**Gambar 3.7.** Pergeseran untai  $x$  pada contoh 3.6.

Contoh kasus terburuk yang mungkin terjadi pada algoritma *brute\_force\_pencocokan\_untai\_eksak*( $x, y$ ) adalah pada saat akan mencocokkan untai  $x = aaa \dots aab$  yakni untai yang terdiri dari rangkaian karakter  $a$  sebanyak  $m - 1$  kali yang diakhiri dengan sebuah karakter  $b$  dan untai  $y = aaaaaa \dots aaaaab$  yakni untai yang terdiri dari rangkaian karakter  $a$  sebanyak  $n - 1$  kali yang diakhiri dengan sebuah karakter  $b$ . Perbandingan-perbandingan yang dilakukan dapat dilihat pada gambar berikut.

Posisi Karakter	:	1 ... .. n
$y =$	:	$a \dots a a a a \dots a a \dots a b$
Posisi ke-1	:	$a \dots a b$
Posisi ke-2	:	$a \dots a b$
Posisi ke-3	:	$a \dots a b$
:	:	...
:	:	...
:	:	...
:	:	...
:	:	...
Posisi ke- $(n - m)$	:	$a \dots a b$
Posisi ke- $(n - m + 1)$	:	$a \dots a b$

**Gambar 3.8.** Contoh kasus terburuk pada algoritma *brute\_force\_pencocokan\_untai\_eksak(x,y)*

Dari gambar terlihat bahwa terdapat  $(n - m + 1)$  posisi yang mungkin dengan masing-masing  $m$  kali perbandingan. Maka total perbandingan yang dilakukan adalah sebanyak  $m(n - m + 1) = mn - m^2 + m$  kali. Sehingga kompleksitas waktu yang diperlukan adalah  $O(mn)$ .

Dalam pencocokan untai eksak, terkadang untai  $x$  muncul beberapa kali di dalam untai  $y$ . Pada pembahasan sebelumnya, diasumsikan bahwa jika untai  $x$  ada di dalam untai  $y$ , maka hanya terdapat satu subuntai dari  $y$  yang sama dengan untai  $x$ . Sekarang diasumsikan bahwa jika untai  $x$  ada di dalam untai  $y$ , maka terdapat satu atau lebih subuntai dari  $y$  yang sama dengan untai  $x$ . Maka algoritma yang dapat dibentuk adalah sebagai berikut:

**Algoritma brute\_force\_pencocokan\_untai\_eksak\_lebih\_dari\_1\_subuntai(x,y)**

**Input** : Untai  $x$  dengan panjang  $m$  dan untai  $y$  dengan panjang  $n$ .  $m > 0$ ,  $n > 0$ , dan  $m \leq n$ .

**Output** : Posisi-posisi karakter awal subuntai dari untai  $y$  yang sama dengan untai  $x$ , atau 0 jika tidak ada subuntai dari untai  $y$  yang sama dengan untai  $x$ .

1.  $posisi[0] = 0$
2.  $k = 0$
3. **for**  $j = 1$  **to**  $n - (m - 1)$
4.      $i = 1$
5.     **while**  $i \leq m$  **and**  $x_i = y_{i+j-1}$  **do**
6.          $i = i + 1$

```

7.   end while
8.   if  $i > m$  then
9.        $k = k + 1$ 
10.     $posisi[k] = j$ 
11.   else
12.       next  $j$ 
13.   end if
14. end for
15. if  $k = 0$ 
16.    display 0  /* jika tidak ada subuntai dari untai  $y$  yang sama dengan
                    untai  $x$ . */
17. else
18.    for  $i = 1$  to  $k$ 
19.        display  $posisi[i]$   /* posisi karakter awal subuntai dari untai  $y$ 
                                yang sama dengan untai  $x$ . */
20.    end for
21. end if

```

Algoritma tersebut dituangkan ke dalam Program Interaktif Pencocokan Untai Eksak Lebih Dari 1 Subuntai yang dapat dilihat pada lampiran 1 ataupun dalam lampiran CD.

### Contoh 3.7.

Jika diberikan:  $x = cgt \quad \rightarrow |x| = m = 3$   
 $y = ccgtcgta \quad \rightarrow |y| = n = 8$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$posisi[0] = 0$ .

$k = 0$ .

- $j = 1, i = 1$ .

$i = 1 \leq 3 = m$  dan  $x_1 = c = y_1$ , maka  $i = 2$ .

$i = 2 \leq 3 = m$  tetapi  $x_2 = g \neq c = y_2$ , maka lanjutkan  $j$ .

- $j = 2, i = 1$ .

$i = 1 \leq 3 = m$  dan  $x_1 = c = y_2$ , maka  $i = 2$ .

$i = 2 \leq 3 = m$  dan  $x_2 = g = y_3$ , maka  $i = 3$ .

$i = 3 \leq 3 = m$  dan  $x_3 = t = y_4$ , maka  $i = 4$ .

$i = 4 > 3 = m$ , maka  $k = k + 1 = 0 + 1 = 1$  dan  $posisi[1] = j = 2$ .

- $j = 3, i = 1$ .

$i = 1 \leq 3 = m$  tetapi  $x_1 = c \neq g = y_3$ , maka lanjutkan  $j$ .

- $j = 4, i = 1$ .

$i = 1 \leq 3 = m$  tetapi  $x_1 = c \neq t = y_4$ , maka lanjutkan  $j$ .

- $j = 5, i = 1$ .

$i = 1 \leq 3 = m$  dan  $x_1 = c = y_5$ , maka  $i = 2$ .

$i = 2 \leq 3 = m$  dan  $x_2 = g = y_6$ , maka  $i = 3$ .

$i = 3 \leq 3 = m$  dan  $x_3 = t = y_7$ , maka  $i = 4$ .

$i = 4 > 3 = m$ , maka  $k = k + 1 = 1 + 1 = 2$  dan  $posisi[2] = j = 5$ .

- $j = 6 = n - (m - 1), i = 1$ .

$i = 1 \leq 3 = m$  tetapi  $x_1 = c \neq g = y_6$ , maka lanjutkan  $j$ .

- $j = 7 > 6 = n - (m - 1)$ .

$k = 2 > 0$ , maka

$i = 1, posisi[1] = 2$ .

$i = 2 = k, posisi[2] = 5$ .

Output : 2 dan 5. Artinya, untai  $x$  ada di dalam untai  $y$  pada posisi ke-2 dan ke-5. Sehingga  $x$  adalah subuntai dari  $y$ .

Posisi Karakter :	1	2	3	4	5	6	7	8
$y =$	$c$	$c$	$g$	$t$	$c$	$g$	$t$	$a$
Posisi ke-1 :	$c$	$g$	$t$					
Posisi ke-2 :		$c$	$g$	$t$				
Posisi ke-3 :			$c$	$g$	$t$			
Posisi ke-4 :				$c$	$g$	$t$		
Posisi ke-5 :					$c$	$g$	$t$	
Posisi ke-6 :						$c$	$g$	$t$

**Gambar 3.9.** Pergeseran untai  $x$  pada contoh 3.7.

Dari contoh di atas, terlihat bahwa untai  $x$  ada di dalam untai  $y$  dan terdapat dua subuntai dari  $y$  yang sama dengan untai  $x$ . Artinya, untai  $x$  muncul dua kali pada posisi ke-2 dan ke-5.

### 3.2. Jarak Hamming

#### 3.2.1. Definisi Jarak Hamming

Diberikan dua buah untai  $x = x_1x_2 \dots x_n$  dan  $y = y_1y_2 \dots y_n$  sepanjang alfabet  $V$  dengan panjang yang sama, misal  $|x| = |y| = n$ , sedemikian sehingga  $x$  dan  $y$  anggota dari  $V^*$ . Jarak Hamming antara untai  $x$  dan untai  $y$  tersebut adalah fungsi  $HD$  yang memetakan  $(x, y)$  ke suatu bilangan real nonnegatif, atau  $HD: V^* \times V^* \rightarrow \mathbb{R} + \{0\}$ .

Fungsi  $HD$  didefinisikan sebagai berikut:

$$\begin{aligned} HD(x, y) &= d(x_1, y_1) + d(x_2, y_2) + \dots + d(x_n, y_n) \\ &= \sum_{i=1}^n d(x_i, y_i) \end{aligned} \quad (3.1)$$

dengan :  $x_i, y_i \in V$  untuk  $i = 1, 2, \dots, n$

$$d(x_i, y_i) = 0 \quad \text{jika } x_i = y_i \text{ dan}$$

$$d(x_i, y_i) = 1 \quad \text{jika } x_i \neq y_i$$

Berdasarkan definisi di atas, dapat diartikan bahwa jarak Hamming antara untai  $x$  dan untai  $y$  yang memiliki panjang yang sama merupakan banyaknya karakter pada untai  $x$  yang berbeda dengan karakter pada untai  $y$  yang berada pada posisi yang bersesuaian. Dengan kata lain, jumlah minimum penggantian karakter yang diperlukan untuk merubah untai  $x$  menjadi untai  $y$  ataupun sebaliknya.

#### Contoh 3.8.

Jika diberikan untai  $x = \text{gelas}$  dan  $y = \text{beras}$ , terlihat bahwa  $|x| = |y| = 5$ , maka jarak Hamming antara kedua untai tersebut adalah

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^5 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &= d(g, b) + d(e, e) + d(l, r) + d(a, a) + d(s, s) \\ &= 1 + 0 + 1 + 0 + 0 \\ &= 2 \end{aligned}$$



Jadi, jarak Hamming antara untai  $x = gelas$  dan  $y = beras$  adalah  $HD(x, y) = 2$  karena  $x_1 = g \neq b = y_1$  dan  $x_3 = l \neq r = y_3$ .

$i :$	1	2	3	4	5
$x =$	$g$	$e$	$l$	$a$	$s$
$y =$	$b$	$e$	$r$	$a$	$s$

**Gambar 3.10.** Contoh perhitungan jarak Hamming pada contoh 3.8.

Selanjutnya, akan dibuktikan bahwa untuk setiap  $x, y$ , dan  $z$  anggota  $V^*$ , jarak Hamming memenuhi semua sifat-sifat jarak yang telah dijelaskan sebelumnya.

- a. Akan dibuktikan bahwa  $HD(x, y) \geq 0$ .

Ambil sembarang  $x$  dan  $y$  anggota  $V^*$  dengan  $|x| = |y| = n$ , maka

$$HD(x, y) = d(x_1, y_1) + d(x_2, y_2) + \cdots + d(x_n, y_n)$$

$$\begin{aligned} \text{Min } HD(x, y) &= \text{Min } d(x_1, y_1) + \text{Min } d(x_2, y_2) + \cdots + \text{Min } d(x_n, y_n) \\ &= 0 + 0 + \cdots + 0 \\ &= 0 \end{aligned}$$

Maka  $HD(x, y) \geq 0$ .

- b. Akan dibuktikan bahwa  $HD(x, y) = 0$  jika  $x = y$ .

Ambil sembarang  $x$  dan  $y$  anggota  $V^*$  dengan  $|x| = |y| = n$  sedemikian sehingga  $x = y$ , maka

$$\begin{aligned} HD(x, y) &= d(x_1, y_1) + d(x_2, y_2) + \cdots + d(x_n, y_n) \\ &= d(y_1, y_1) + d(y_2, y_2) + \cdots + d(y_n, y_n) \\ &= 0 + 0 + \cdots + 0 \\ &= 0 \end{aligned}$$

Maka  $HD(x, y) = 0$  jika  $x = y$ .

- c. Akan dibuktikan bahwa  $HD(x, y) = HD(y, x)$ .

Ambil sembarang  $x$  dan  $y$  anggota  $V^*$  dengan  $|x| = |y| = n$ , maka

$$\begin{aligned} HD(x, y) &= d(x_1, y_1) + d(x_2, y_2) + \cdots + d(x_n, y_n) \\ &= d(y_1, x_1) + d(y_2, x_2) + \cdots + d(y_n, x_n) \end{aligned}$$

$$= HD(y, x)$$

Maka  $HD(x, y) = HD(y, x)$ .

d. Akan dibuktikan bahwa  $HD(x, z) \leq HD(x, y) + HD(y, z)$ .

Ambil sembarang  $x, y$ , dan  $z$  anggota  $V^*$  dengan  $|x| = |y| = |z| = n$ .

Untuk  $x = z$ , maka  $d(x_i, z_i) = 0$  untuk setiap  $i = 1, 2, \dots, n$  sedemikian sehingga  $HD(x, z) = 0$ . Sebelumnya diketahui bahwa  $0 \leq HD(x, y)$  dan  $0 \leq HD(y, z)$  sehingga  $0 \leq HD(x, y) + HD(y, z)$ . Maka

$$HD(x, z) = 0 \leq HD(x, y) + HD(y, z).$$

Untuk  $x \neq z$ , akan digunakan induksi matematika dengan menginduksi panjang dari  $x, y$ , maupun  $z$  yaitu  $n$ .

Langkah awal. Misalkan  $n = 1$ , maka  $HD(x, z) = d(x_1, z_1) = 1$  dan  $y \neq x$  atau  $y \neq z$ . Sehingga  $HD(x, y) + HD(y, z) = d(x_1, y_1) + d(y_1, z_1)$  akan bernilai 1 atau bahkan 2. Maka  $HD(x, z) = 1 \leq HD(x, y) + HD(y, z)$ .

Jadi, terbukti bahwa  $HD(x, z) \leq HD(x, y) + HD(y, z)$  untuk  $n = 1$

Hipotesis induksi. Jika  $|x| = |y| = |z| = k$ , maka

$$\begin{aligned} HD(x, z) &\leq HD(x, y) + HD(y, z) \text{ atau} \\ d(x_1, z_1) + \dots + d(x_k, z_k) &\leq [d(x_1, y_1) + d(y_1, z_1)] \\ &\quad + [d(x_2, y_2) + d(y_2, z_2)] + \dots \\ &\quad + [d(x_k, y_k) + d(y_k, z_k)] \end{aligned}$$

Langkah Induksi. Misalkan  $|x| = |y| = |z| = k + 1$ .

$$HD(x, y) = d(x_1, y_1) + \dots + d(x_k, y_k) + d(x_{k+1}, y_{k+1})$$

$$HD(y, z) = d(y_1, z_1) + \dots + d(y_k, z_k) + d(y_{k+1}, z_{k+1})$$

$$HD(x, z) = d(x_1, z_1) + \dots + d(x_k, z_k) + d(x_{k+1}, z_{k+1})$$

Maka,

$$\begin{aligned} HD(x, z) &= d(x_1, z_1) + \dots + d(x_k, z_k) + d(x_{k+1}, z_{k+1}) \\ &\leq [d(x_1, y_1) + d(y_1, z_1)] + [d(x_2, y_2) + d(y_2, z_2)] + \dots \\ &\quad + [d(x_k, y_k) + d(y_k, z_k)] + d(x_{k+1}, z_{k+1}) \\ &\leq [d(x_1, y_1) + d(y_1, z_1)] + [d(x_2, y_2) + d(y_2, z_2)] + \dots \\ &\quad + [d(x_k, y_k) + d(y_k, z_k)] \\ &\quad + [d(x_{k+1}, y_{k+1}) + d(y_{k+1}, z_{k+1})] \\ &= HD(x, y) + HD(y, z) \end{aligned}$$

Sehingga,

$$HD(x, z) \leq HD(x, y) + HD(y, z).$$

Dari uraian di atas, terbukti bahwa untuk setiap  $x$ ,  $y$ , dan  $z$  anggota  $V^*$ , jarak Hamming memenuhi sifat-sifat jarak, yaitu:

- a.  $HD(x, y) \geq 0$
- b.  $HD(x, y) = 0$  jika  $x = y$
- c.  $HD(x, y) = HD(y, x)$
- d.  $HD(x, z) \leq HD(x, y) + HD(y, z)$

### 3.2.2. Algoritma Jarak Hamming

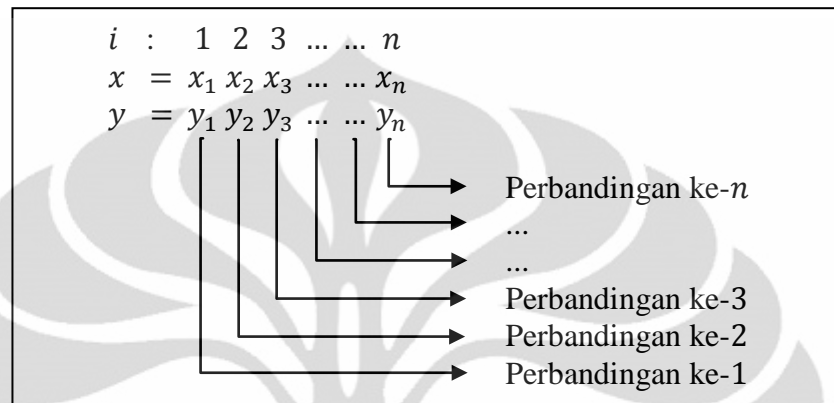
Berdasarkan definisi jarak Hamming di atas, dapat dibentuk suatu algoritma untuk menghitung jarak Hamming antara dua buah untai yang memiliki panjang yang sama. Misal diberikan dua buah untai,  $x$  dan  $y$ , yang memiliki panjang yang sama yaitu  $n$ , dengan  $x = x_1x_2 \dots x_n$  dan  $y = y_1y_2 \dots y_n$ . Maka algoritma penghitungan jarak Hamming yang dapat dibentuk adalah sebagai berikut:

**Algoritma jarak\_hamming( $x, y$ )**  
**Input** : Untai  $x$  dan untai  $y$  dengan panjang  $n$ .  
**Output** : Jarak Hamming antara untai  $x$  dan untai  $y$ .

1. **if**  $n = 0$  **then**
2.      $HD = 0$
3. **else**
4.      $HD = 0$
5.     **for**  $i = 1$  **to**  $n$
6.         **if**  $x_i = y_i$  **then**
7.              $HD = HD + 0$
8.         **else**
9.              $HD = HD + 1$
10.         **end if**
11.     **end for**
12. **end if**
13. **display**  $HD$  /\*  $HD$  adalah jarak Hamming antara untai  $x$  dan untai  $y$ . \*/

Algoritma tersebut dituangkan ke dalam Program Interaktif Penghitungan Jarak Hamming yang dapat dilihat pada lampiran 2 ataupun dalam lampiran CD.

Di dalam algoritma tersebut terlihat bahwa terdapat  $n$  kali perbandingan yang dilakukan untuk membandingkan dua buah karakter dari masing-masing untai untuk setiap posisi pada untai. Sehingga kompleksitas waktu yang dibutuhkan adalah  $O(n)$ .



**Gambar 3.11.** Perbandingan yang dilakukan pada algoritma  $\text{jarak\_hamming}(x, y)$

### Contoh 3.9.

Jika diberikan:  $x = \text{biskuit} \rightarrow |x| = n = 7$

$y = \text{bauksit} \rightarrow |y| = n = 7$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$n = 7 > 0$ .

$HD = 0$ .

- $i = 1, x_1 = b = y_1$ , maka  $HD = HD + 0 = 0 + 0 = 0$ .
- $i = 2, x_2 = i \neq a = y_2$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .
- $i = 3, x_3 = s \neq u = y_3$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .
- $i = 4, x_4 = k = y_4$ , maka  $HD = HD + 0 = 2 + 0 = 2$ .
- $i = 5, x_5 = u \neq s = y_5$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .
- $i = 6, x_6 = i = y_6$ , maka  $HD = HD + 0 = 3 + 0 = 3$ .
- $i = 7 = n, x_7 = t = y_7$ , maka  $HD = HD + 0 = 3 + 0 = 3$ .

Output : 3. Artinya, jarak Hamming antara untai  $x$  dan  $y$  adalah 3.

$i$	:	1	2	3	4	5	6	7
$x$	=	b	i	s	k	u	i	t
$y$	=	b	a	u	k	s	i	t

**Gambar 3.12.** Penerapan algoritma  $jarak\_hamming(x, y)$  pada contoh 3.9.

Perhitungan jarak Hamming juga dapat dilakukan dalam bentuk tabel atau matriks berdasarkan algoritma sebagai berikut:

**Algoritma jarak\_hamming2( $x, y$ )**  
**Input** : Untai  $x$  dan untaian  $y$  dengan panjang  $n$ .  
**Output** : Jarak Hamming antara untaian  $x$  dan untaian  $y$ .

1. **if**  $n = 0$  **then**
2.      $C_{n,n} = 0$
3. **else**
4.      $C_{0,0} = 0$
5.     **for**  $i = 1$  **to**  $n$
6.         **for**  $j = 1$  **to**  $n$
7.             **if**  $i = j$  **then**
8.                 **if**  $x_i = y_j$  **then**
9.                      $C_{i,j} = C_{i-1,j-1} + 0$
10.                    **else**
11.                      $C_{i,j} = C_{i-1,j-1} + 1$
12.                    **end if**
13.             **else**
14.                  $C_{i,j} = 0$
15.             **end if**
16.         **end for**
17.     **end for**
18. **end if**
19. **display**  $C_{n,n}$  /\*  $C_{n,n}$  adalah jarak Hamming antara untaian  $x$  dan untaian  $y$ . \*/

Pada algoritma di atas, terjadi dua kali pengulangan (*looping for*) masing-masing sebanyak  $n$  kali sehingga kompleksitas waktu pada algoritma tersebut adalah  $O(n^2)$ . Akibatnya, perhitungan jarak Hamming dengan algoritma  $jarak\_hamming2(x, y)$  jelas memerlukan waktu yang lebih lama dibandingkan dengan menggunakan algoritma  $jarak\_hamming(x, y)$ . Sehingga, algoritma penghitungan jarak Hamming yang akan digunakan untuk pencocokan untaian eksak

adalah algoritma  $jarak\_hamming(x, y)$ . Akan tetapi, setidaknya algoritma  $jarak\_hamming2(x, y)$  ini dapat menunjukkan bahwa perhitungan jarak Hamming dapat pula dilakukan dengan menggunakan tabel atau matriks.

**Tabel 3.1.** Tabel algoritma  $jarak\_hamming2(x, y)$

-	-	0	1	...	...	$n$
-	-	-	$x_1$	...	...	$x_n$
0	-	0	0	0	0	0
1	$y_1$	0	$c_{1,1}$	0	0	0
...	...	0	0	...	0	0
...	...	0	0	0	...	0
$n$	$y_n$	0	0	0	0	$c_{n,n}$

Pada tabel di atas, sel-sel yang diperhatikan hanyalah sel yang berada pada diagonal tabelnya saja. Sedangkan sel-sel pada bagian atas kanan dan kiri bawah diisi dengan 0. Pengisian tabel dimulai dari sel  $(0, 0)$  yang diisi dengan 0 dan berlanjut ke sel  $(1, 1)$ , jika  $x_1 \neq y_1$  maka sel  $(1, 1)$  ini akan berisi nilai sel  $(0, 0)$  di tambah 1. Begitu seterusnya hingga mencapai sel yang berada pada pojok kanan bawah, yaitu sel  $(n, n)$ . Sel  $(n, n)$  inilah yang merupakan jarak Hamming antara untai  $x$  dan  $y$ .

**Contoh 3.10.**

Jika diberikan:  $x = \text{sate} \quad \rightarrow |x| = n = 4$   
 $y = \text{soto} \quad \rightarrow |y| = n = 4$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$$n = 4 > 0.$$

$$C_{0,0} = 0.$$

- $i = 1.$

$$j = 1, i = j, x_1 = s = y_1, \text{ maka } C_{1,1} = C_{0,0} + 0 = 0.$$

$$j = 2, i \neq j, \text{ maka } C_{1,2} = 0.$$

$j = 3, i \neq j$ , maka  $C_{1,3} = 0$ .

$j = 4 = n, i \neq j$ , maka  $C_{1,4} = 0$ .

- $i = 2$ .

$j = 1, i \neq j$ , maka  $C_{2,1} = 0$ .

$j = 2, i = j, x_2 = a \neq o = y_2$ , maka  $C_{2,2} = C_{1,1} + 1 = 1$ .

$j = 3, i \neq j$ , maka  $C_{2,3} = 0$ .

$j = 4 = n, i \neq j$ , maka  $C_{2,4} = 0$ .

- $i = 3$ .

$j = 1, i \neq j$ , maka  $C_{3,1} = 0$ .

$j = 2, i \neq j$ , maka  $C_{3,2} = 0$ .

$j = 3, i = j, x_3 = t = y_3$ , maka  $C_{3,3} = C_{2,2} + 0 = 1$ .

$j = 4, i \neq j$ , maka  $C_{3,4} = 0$ .

- $i = 4 = n$ .

$j = 1, i \neq j$ , maka  $C_{4,1} = 0$ .

$j = 2, i \neq j$ , maka  $C_{4,2} = 0$ .

$j = 3, i = j$ , maka  $C_{4,3} = 0$ .

$j = 4 = n, i \neq j, x_4 = e \neq o = y_4$ , maka  $C_{4,4} = C_{3,3} + 1 = 2$ .

Output : 2. Artinya, jarak Hamming antara untai  $x$  dan  $y$  adalah 2.

**Tabel 3.2.** Penghitungan jarak Hamming pada contoh 3.10. dengan menggunakan tabel

-	-	0	1	2	3	4
-	-	-	s	a	t	e
0	-	0	0	0	0	0
1	s	0	0	0	0	0
2	o	0	0	1	0	0
3	t	0	0	0	1	0
4	o	0	0	0	0	2

Dari contoh di atas, terlihat bahwa sel  $C_{n,n} = C_{4,4} = 2$ . Artinya jarak Hamming antara untai  $x$  dan  $y$  adalah 2.

### 3.3. Metode Pencocokan Untai Eksak dengan Ukuran Jarak Hamming

Setelah melihat pembahasan sebelumnya tentang pencocokan untaik eksak dan jarak Hamming, terlihat bahwa ada keterkaitan antara keduanya. Jika diberikan dua buah untaik, yaitu  $x$  dan  $y$  yang memiliki panjang yang sama, misal  $|x| = |y| = n$ , dengan jarak Hamming  $HD(x, y) = 0$ , maka untaik  $x$  dan  $y$  merupakan dua buah untaik yang sama. Dengan kata lain, dapat dinyatakan bahwa untaik  $x$  adalah subuntaik dari untaik  $y$  atau untaik  $x$  ada di dalam untaik  $y$  pada posisi ke-1. Lain halnya jika kedua buah untaik yang diberikan memiliki panjang yang tidak sama, misal  $|x| = m$  dan  $|y| = n$  dengan  $m \leq n$ , maka pencocokan untaik eksak dapat dilakukan akan tetapi jarak Hamming antara kedua buah untaik tersebut tidak dapat dihitung.

Seandainya untaik  $x$ , yang lebih pendek dari untaik  $y$ , dirangkaikan dengan sebuah karakter tertentu sebanyak  $n - m$  di belakang ataupun di depan untaik  $x$  sedemikian sehingga panjang dari untaik  $x$  menjadi sama dengan panjang dari untaik  $y$  yaitu  $|x| = |y| = n$ , maka jarak Hamming antara keduanya barulah dapat dihitung. Atas pemikiran inilah akhirnya ditetapkan bahwa karakter yang harus dirangkaikan tersebut adalah untaik hampa atau  $\wedge$  dengan mendefinisikan kembali bahwa panjang dari sebuah untaik hampa adalah 1.

Berikut ini adalah contoh perangkaian sebuah untaik  $x$  yang memiliki panjang  $m$  dengan beberapa untaik hampa sedemikian sehingga  $|x| = |y| = n$  dan dapat dihitung jarak Hamming-nya.

#### Contoh 3.11.

Misal, diberikan untaik  $x = adi$  dan  $y = radio$ . Terlihat bahwa  $|x| = m = 3$  tidak sama dengan  $|y| = n = 5$ . Maka untaik  $x$  harus dirangkaikan dengan untaik hampa sebanyak  $n - m = 5 - 3 = 2$  buah dengan beberapa kemungkinan posisi, yaitu:

- a. Kemungkinan posisi pertama :  $x = adi\wedge\wedge$ , sehingga

$$y = r a d i o$$

$$x = a d i \wedge \wedge$$

- b. Kemungkinan posisi kedua :  $x = \wedge adi\wedge$ , sehingga

$$y = r a d i o$$



$$x = \text{^} a d i \text{^}$$

c. Kemungkinan posisi ketiga :  $x = \text{^} \text{^} adi$ , sehingga

$$y = r a d i o$$

$$x = \text{^} \text{^} a d i$$

Posisi Karakter	:	1	2	3	4	5
$y = r a d i o$			<b>a</b>	<b>d</b>	<b>i</b>	<b>o</b>
Kemungkinan Posisi ke-1	:	<b>a</b>	<b>d</b>	<b>i</b>	<b>^</b>	<b>^</b>
Kemungkinan Posisi ke-2	:	<b>^</b>	<b>a</b>	<b>d</b>	<b>i</b>	<b>^</b>
Kemungkinan Posisi ke-3	:	<b>^</b>	<b>^</b>	<b>a</b>	<b>d</b>	<b>i</b>

**Gambar 3.13.** Kemungkinan posisi untai hampa pada contoh 3.11.

Dari contoh tersebut, terlihat bahwa dari untai  $x$  dengan  $|x| = m = 3$  dan untai  $y$  dengan  $|y| = n = 5$  menghasilkan 3 kemungkinan posisi penempatan untai hampa yang dirangkaikan dengan untai  $x$ .

Jika diberikan untai  $x$  dan  $y$  dengan panjang masing-masing  $|x| = m$  dan  $|y| = n$  sedemikian sehingga  $x = x_1 x_2 \dots x_m$  dan  $y = y_1 y_2 \dots y_{m-1} y_m y_{m+1} \dots y_n$ . Maka untai  $x$  harus dirangkaikan dengan untai hampa sebanyak  $n - m$  buah dengan beberapa kemungkinan posisi. Banyaknya kemungkinan posisi perangkaian untai hampa dengan untai  $x$  pada dasarnya merupakan banyaknya pergeseran untai  $x$  sejauh satu langkah ke kanan hingga mencapai bagian akhir dari untai  $y$  ditambah satu posisi awal.

$i$	:	1	...	...	...	...	...	$n$
$y$	=	$y_1$	...	...	...	...	...	$y_n$
$x$	=	$x_1$	...	$x_m$				
$y$	=	$y_1$	...	...	...	...	...	$y_n$
$x$	=		$x_1$	...	$x_m$			
	:							
$y$	=	$y_1$	...	...	...	...	...	$y_n$
$x$	=					$x_1$	...	$x_m$

**Gambar 3.14.** Ilustrasi pergeseran untai  $x$

Dari gambar di atas, jika diberikan:

$m = 1$ , maka banyaknya kemungkinan posisi adalah  $n$ .

$m = 2$ , maka banyaknya kemungkinan posisi adalah  $n - 1$ .

$m = 3$ , maka banyaknya kemungkinan posisi adalah  $n - 2$ .

:

:

$m = n - 1$ , maka banyaknya kemungkinan posisi adalah 2.

$m = n$ , maka banyaknya kemungkinan posisi adalah 1.

Berdasarkan teori barisan, kemungkinan-kemungkinan posisi tersebut membentuk suatu barisan aritmetika dengan  $U_1 = n$  dan  $b = -1$ . Maka didapat rumus suku ke- $m$ :

$$\begin{aligned} U_m &= U_1 + (m - 1)b \\ &= n + (m - 1)(-1) \\ &= n - (m - 1) \\ &= n - m + 1 \end{aligned} \tag{3.2}$$

Sehingga, untuk  $m = 1, 2, 3, \dots, n$ , banyaknya kemungkinan posisi untai hampa yang harus dirangkaikan dengan untai  $x$  adalah  $n - m + 1$ . Dalam bentuk kombinasi, dapat dituliskan sebagai berikut:

$$\begin{aligned} C_1^{n-m+1} &= \frac{(n - m + 1)!}{1! (n - m)!} \\ &= \frac{(n - m + 1)(n - m)!}{(n - m)!} = n - m + 1 \end{aligned} \tag{3.3}$$

Setelah didapatkan banyaknya kemungkinan posisi untai hampa yang harus dirangkaikan dengan untai yang lebih pendek, barulah dapat dihitung jarak Hamming untuk setiap kemungkinan posisi pencocokan untai. Berikut ini adalah beberapa contoh pencocokan untai eksak menggunakan ukuran jarak Hamming.

### Contoh 3.12.

Misal, diberikan untai  $x = bur$  dan  $y = bubuk$ . Terlihat bahwa panjang dari  $x$  yaitu  $|x| = m = 3$  tidak sama dengan  $|y| = n = 5$ . Maka untai  $x$  harus dirangkaikan dengan untai hampa sebanyak  $n - m = 5 - 3 = 2$  buah dengan  $n - m + 1 = 5 - 3 + 1 = 3$  kemungkinan posisi, yaitu:

- a. Kemungkinan posisi pertama :  $x = bur^{^^}$ , sehingga

$$y = b u b u r$$

$$x = b u r ^ ^$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^5 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &= d(b, b) + d(u, u) + d(r, b) + d(^, u) + d(^, r) \\ &= 0 + 0 + 1 + 1 + 1 \\ &= 3 \end{aligned}$$

- b. Kemungkinan posisi kedua :  $x = ^bur^$ , sehingga

$$y = b u b u r$$

$$x = ^ b u r ^$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^5 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &= d(^, b) + d(b, u) + d(u, b) + d(r, u) + d(^, r) \\ &= 1 + 1 + 1 + 1 + 1 \\ &= 5 \end{aligned}$$

- c. Kemungkinan posisi terakhir :  $x = ^^bur$ , sehingga

$$y = b u b u r$$

$$x = ^ ^ b u r$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^5 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &= d(^, b) + d(^, u) + d(b, b) + d(u, u) + d(r, r) \\ &= 1 + 1 + 0 + 0 + 0 \\ &= 2 \end{aligned}$$

Posisi Karakter	:	1	2	3	4	5
	$y =$	$b$	$u$	$b$	$u$	$r$
Kemungkinan Posisi ke-1	:	$b$	$u$	$r$	$\wedge$	$\wedge$
Kemungkinan Posisi ke-2	:	$\wedge$	$b$	$u$	$r$	$\wedge$
Kemungkinan Posisi ke-3	:	$\wedge$	$\wedge$	$b$	$u$	$r$

**Gambar 3.15.** Kemungkinan posisi untai hampa pada contoh 3.12.

Dari contoh tersebut, terlihat bahwa untai  $x$  berada pada posisi ke-3 di dalam untai  $y$ . Posisi tersebut juga merupakan posisi pencocokan untai yang memiliki jarak Hamming terkecil yaitu 2 dibandingkan dengan jarak Hamming pada kemungkinan posisi pencocokan untai yang lain.

**Contoh 3.13.**

Misal diberikan untai  $x = cgt$  dan  $y = ccgtcgta$ . Terlihat bahwa  $|x| = m = 3$  dan  $|y| = n = 8$ . Maka untai  $x$  harus dirangkaikan dengan untai hampa sebanyak  $n - m = 8 - 3 = 5$  buah dengan  $n - m + 1 = 8 - 3 + 1 = 6$  kemungkinan posisi, yaitu:

- a. Kemungkinan posisi pertama :  $x = cgt\wedge\wedge\wedge\wedge$ , sehingga

$$\begin{array}{cccccccc} y & = & c & c & g & t & c & g & t & a \\ x & = & c & g & t & \wedge & \wedge & \wedge & \wedge & \wedge \end{array}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^8 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &\quad + d(x_6, y_6) + d(x_7, y_7) + d(x_8, y_8) \\ &= d(c, c) + d(g, c) + d(t, g) + d(\wedge, t) + d(\wedge, c) + d(\wedge, g) \\ &\quad + d(\wedge, t) + d(\wedge, a) \\ &= 0 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \\ &= 7 \end{aligned}$$

- b. Kemungkinan posisi kedua :  $x = \wedge cgt\wedge\wedge\wedge$ , sehingga

$$\begin{array}{cccccccc} y & = & c & c & g & t & c & g & t & a \\ x & = & \wedge & c & g & t & \wedge & \wedge & \wedge & \wedge \end{array}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^8 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &\quad + d(x_6, y_6) + d(x_7, y_7) + d(x_8, y_8) \\
 &= d(^, c) + d(c, c) + d(g, g) + d(t, t) + d(^, c) + d(^, g) \\
 &\quad + d(^, t) + d(^, a) \\
 &= 1 + 0 + 0 + 0 + 1 + 1 + 1 + 1 \\
 &= 5
 \end{aligned}$$

c. Kemungkinan posisi ketiga :  $x = \text{^}^cgt\text{^}^$ , sehingga

$$\begin{array}{cccccccc}
 y & = & c & c & g & t & c & g & t & a \\
 x & = & \text{^} & \text{^} & c & g & t & \text{^} & \text{^} & \text{^}
 \end{array}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^8 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &\quad + d(x_6, y_6) + d(x_7, y_7) + d(x_8, y_8) \\
 &= d(^, c) + d(^, c) + d(c, g) + d(g, t) + d(t, c) + d(^, g) \\
 &\quad + d(^, t) + d(^, a) \\
 &= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 \\
 &= 8
 \end{aligned}$$

d. Kemungkinan posisi keempat :  $x = \text{^}^cgt\text{^}^$ , sehingga

$$\begin{array}{cccccccc}
 y & = & c & c & g & t & c & g & t & a \\
 x & = & \text{^} & \text{^} & \text{^} & c & g & t & \text{^} & \text{^}
 \end{array}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^8 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &\quad + d(x_6, y_6) + d(x_7, y_7) + d(x_8, y_8) \\
 &= d(^, c) + d(^, c) + d(^, g) + d(c, t) + d(g, c) + d(t, g) \\
 &\quad + d(^, t) + d(^, a)
 \end{aligned}$$

$$= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$$

$$= 8$$

e. Kemungkinan posisi kelima :  $x = \wedge\wedge\wedge\wedge cgt\wedge$ , sehingga

$$y = c \ c \ g \ t \ c \ g \ t \ a$$

$$x = \wedge \ \wedge \ \wedge \ \wedge \ c \ g \ t \ \wedge$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$HD(x, y) = \sum_{i=1}^8 d(x_i, y_i)$$

$$= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5)$$

$$+ d(x_6, y_6) + d(x_7, y_7) + d(x_8, y_8)$$

$$= d(\wedge, c) + d(\wedge, c) + d(\wedge, g) + d(\wedge, t) + d(c, c) + d(g, g)$$

$$+ d(t, t) + d(\wedge, a)$$

$$= 1 + 1 + 1 + 1 + 0 + 0 + 0 + 1$$

$$= 5$$

f. Kemungkinan posisi terakhir :  $x = \wedge\wedge\wedge\wedge\wedge cgt$ , sehingga

$$y = c \ c \ g \ t \ c \ g \ t \ a$$

$$x = \wedge \ \wedge \ \wedge \ \wedge \ \wedge \ c \ g \ t$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$HD(x, y) = \sum_{i=1}^8 d(x_i, y_i)$$

$$= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5)$$

$$+ d(x_6, y_6) + d(x_7, y_7) + d(x_8, y_8)$$

$$= d(\wedge, c) + d(\wedge, c) + d(\wedge, g) + d(\wedge, t) + d(\wedge, c) + d(c, g)$$

$$+ d(g, t) + d(t, a)$$

$$= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$$

$$= 8$$

Posisi Karakter	:	1	2	3	4	5	6	7	8
$y =$		<i>c</i>	<i>c</i>	<i>g</i>	<i>t</i>	<i>c</i>	<i>g</i>	<i>t</i>	<i>a</i>
Kemungkinan Posisi ke-1	:	<i>c</i>	<i>g</i>	<i>t</i>	^	^	^	^	^
Kemungkinan Posisi ke-2	:	^	<i>c</i>	<i>g</i>	<i>t</i>	^	^	^	^
Kemungkinan Posisi ke-3	:	^	^	<i>c</i>	<i>g</i>	<i>t</i>	^	^	^
Kemungkinan Posisi ke-4	:	^	^	^	<i>c</i>	<i>g</i>	<i>t</i>	^	^
Kemungkinan Posisi ke-5	:	^	^	^	^	<i>c</i>	<i>g</i>	<i>t</i>	^
Kemungkinan Posisi ke-6	:	^	^	^	^	^	<i>c</i>	<i>g</i>	<i>t</i>

**Gambar 3.16.** Kemungkinan posisi untai hampa pada contoh 3.13.

Pada contoh ini, untai  $x$  berada pada posisi ke-2 dan ke-5 di dalam untai  $y$ . Posisi tersebut sekaligus sebagai posisi yang memiliki jarak Hamming terkecil yaitu 5 dibandingkan dengan jarak Hamming pada posisi-posisi yang lain.

Dari kedua contoh di atas, terlihat bahwa jarak Hamming terkecil yang diperoleh dari sekian banyak kemungkinan posisi pencocokan di atas adalah sama dengan banyaknya untai hampa yang harus dirangkaikan dengan untai  $x$ , yaitu  $n - m$ . Kemudian untai  $x$  juga merupakan subuntai dari  $y$ , dan posisi untai  $x$  di dalam  $y$  berada pada kemungkinan posisi dengan jarak Hamming yang terkecil. Sehingga, jika diberikan untai  $x$  dan  $y$  dengan panjang masing-masing  $|x| = m$  dan  $|y| = n$ ,  $m \leq n$ , sedemikian sehingga  $x = x_1x_2 \dots x_m$  dan  $y = y_1y_2 \dots y_n$ , maka untai  $x$  harus dirangkaikan dengan untai hampa sebanyak  $n - m$  buah dengan  $n - m + 1$  kemungkinan posisi. Jika jarak Hamming terkecil diperoleh hanya pada posisi ke- $i$  dan besarnya sama dengan  $n - m$ , maka untai  $x$  ada di dalam untai  $y$  dan hanya satu yakni pada posisi ke- $i$  tersebut.

Berdasarkan metode yang telah dijelaskan di atas, disusunlah sebuah algoritma pencocokan untai eksak dengan menggunakan ukuran jarak Hamming sebagai berikut:

**Algoritma pencocokan untai eksak dengan ukuran jarak hamming( $x, y$ )**  
**Input** : Untai  $x$  dan untai  $y$  dengan panjang masing-masing  $m$  dan  $n$  dengan  $m \leq n$ .  
**Output** : Posisi karakter awal subuntai dari  $y$  yang sama dengan untai  $x$  atau 0 jika tidak ada subuntai dari  $y$  yang sama dengan untai  $x$ , serta jarak Hamming antara untai  $x$  dan untai  $y$  tersebut.

1. **if**  $m = 0$  **then**
2.      $k = 2$

```

3.     posisi[1] = 0
4.     posisi[2] = n + 1
5.     HD_Min = n
6.     else if n = 0 then
7.         k = 1
8.         posisi[1] = 0
9.         HD_Min = 0
10.    else
11.        posisi[0] = 0
12.        if m ≠ n then
13.            for i = 1 to n - m
14.                xm+i = '^'
15.            end for
16.        end if
17.        HD_Min = n
18.        k = 0
19.        for j = 1 to n - m + 1
20.            HD = 0
21.            for i = 1 to n
22.                if xi = yi then
23.                    HD = HD + 0
24.                else
25.                    HD = HD + 1
26.                end if
27.            end for
28.            if HD = n - m then
29.                HD_Min = HD
30.                k = k + 1
31.                posisi[k] = j
32.            else if HD ≤ HD_Min then
33.                HD_Min = HD
34.            else
35.                next j
36.            end if
37.            if m ≠ n then
38.                for i = m + j to j + 1 step -1
39.                    xi = xi-1
40.                end for
41.                xj = '^'
42.            end if
43.        end for
44.    end if
45.    if k = 0 then
46.        display 0    /* jika tidak ada subuntai dari untai y yang sama dengan
                        untai x. */
47.    end if
48.    else

```



```

49.   for  $i = 1$  to  $k$ 
50.       display  $posisi[i]$    /* posisi karakter awal subuntai dari untai y
                                   yang sama dengan untai x. */
51.   end for
52. end if
53. display  $HD\_Min$    /*  $HD\_Min$  adalah jarak Hamming minimum antara
                                   untai  $x$  dan untai  $y$ . */

```

#### Algoritma

*pencocokan\_untai\_eksak\_dengan\_ukuran\_jarak\_hamming(x, y)* ini dituangkan ke dalam Program Interaktif Pencocokan Untai Eksak dengan Menggunakan Ukuran Jarak Hamming yang dapat dilihat pada lampiran 3 ataupun dalam lampiran CD.

Pada algoritma tersebut terdapat dua kali perbandingan yang cukup signifikan, yakni perbandingan untuk menentukan setiap posisi pencocokan sebanyak  $n - m + 1$  kali dan perbandingan untuk menghitung jarak Hamming dari setiap posisi, masing-masing sebanyak  $n$  kali. Maka, total perbandingan yang cukup signifikan pada algoritma tersebut adalah  $n(n - m + 1) = n^2 - nm + n$ . Sehingga kompleksitas waktunya adalah  $O(n^2)$ .

#### Contoh 3.14.

Jika diberikan:  $x = top \quad \rightarrow |x| = m = 3 \neq 0$   
 $y = ketoprak \quad \rightarrow |y| = n = 8 \neq 0$

Maka, perjalanan algoritmanya adalah sebagai berikut:

$posisi[1] = 0$ .

$m \neq n$ , maka

$$i = 1, x_4 = \wedge.$$

$$i = 2, x_5 = \wedge.$$

$$i = 3, x_6 = \wedge.$$

$$i = 4, x_7 = \wedge.$$

$$i = 5 = n - m, x_8 = \wedge.$$

$HD\_Min = n = 8$ .

$k = 0$ .

- $j = 1, HD = 0$ .

$i = 1, x_1 = t \neq k = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = o \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = p \neq t = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = \wedge \neq o = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 5, x_5 = \wedge \neq p = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$i = 6, x_6 = \wedge \neq r = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$i = 7, x_7 = \wedge \neq a = y_7$ , maka  $HD = HD + 1 = 6 + 1 = 7$ .

$i = 8 = n, x_8 = \wedge \neq k = y_8$ , maka  $HD = HD + 1 = 7 + 1 = 8$ .

$HD = 8 \leq HD_{Min}$ , maka  $HD_{Min} = HD = 8$ .

$m \neq n$ , maka

$i = 4, x_4 = x_3 = p$ .

$i = 3, x_3 = x_2 = o$ .

$i = 2 = j + 1, x_2 = x_1 = t$ .

$x_1 = \wedge$ .

- $j = 2, HD = 0$ .

$i = 1, x_1 = \wedge \neq k = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = t \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = o \neq t = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = p \neq o = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 5, x_5 = \wedge \neq p = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$i = 6, x_6 = \wedge \neq r = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$i = 7, x_7 = \wedge \neq a = y_7$ , maka  $HD = HD + 1 = 6 + 1 = 7$ .

$i = 8 = n, x_8 = \wedge \neq k = y_8$ , maka  $HD = HD + 1 = 7 + 1 = 8$ .

$HD = 8 \leq HD_{Min}$ , maka  $HD_{Min} = HD = 8$ .

$m \neq n$ , maka

$i = 5, x_5 = x_4 = p$ .

$i = 4, x_4 = x_3 = o$ .

$i = 3 = j + 1, x_3 = x_2 = t$ .

$x_2 = \wedge$ .

- $j = 3, HD = 0$ .

$i = 1, x_1 = \wedge \neq k = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = t = y_3$ , maka  $HD = HD + 0 = 2 + 0 = 2$ .

$i = 4, x_4 = o = y_4$ , maka  $HD = HD + 0 = 2 + 0 = 2$ .

$i = 5, x_5 = p = y_5$ , maka  $HD = HD + 0 = 2 + 0 = 2$ .

$i = 6, x_6 = \wedge \neq r = y_6$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 7, x_7 = \wedge \neq a = y_7$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 8 = n, x_8 = \wedge \neq k = y_8$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$HD = 5 = n - m$ , maka  $HD\_Min = HD = 5, k = k + 1 = 0 + 1 = 1$ ,

$posisi[1] = j = 3$ .

$m \neq n$ , maka

$i = 6, x_6 = x_5 = p$ .

$i = 5, x_5 = x_4 = o$ .

$i = 4 = j + 1, x_4 = x_3 = t$ .

$x_3 = \wedge$ .

- $j = 4, HD = 0$ .

$i = 1, x_1 = \wedge \neq k = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = \wedge \neq t = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = t \neq o = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 5, x_5 = o \neq p = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$i = 6, x_6 = p \neq r = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$i = 7, x_7 = \wedge \neq a = y_7$ , maka  $HD = HD + 1 = 6 + 1 = 7$ .

$i = 8 = n, x_8 = \wedge \neq k = y_8$ , maka  $HD = HD + 1 = 7 + 1 = 8$ .

$HD = 8 > 5 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$i = 7, x_7 = x_6 = p$ .

$i = 6, x_6 = x_5 = o$ .

$i = 5 = j + 1, x_5 = x_4 = t$ .

$x_4 = \wedge$ .

- $j = 5, HD = 0$ .

$i = 1, x_1 = \wedge \neq k = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = \wedge \neq t = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = \wedge \neq o = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 5, x_5 = t \neq p = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$i = 6, x_6 = o \neq r = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$i = 7, x_7 = p \neq a = y_7$ , maka  $HD = HD + 1 = 6 + 1 = 7$ .

$i = 8 = n, x_8 = \wedge \neq k = y_8$ , maka  $HD = HD + 1 = 7 + 1 = 8$ .

$HD = 8 > 5 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$i = 8, x_8 = x_7 = p$ .

$i = 7, x_7 = x_6 = o$ .

$i = 6 = j + 1, x_6 = x_5 = t$ .

$x_5 = \wedge$ .

- $j = 6 = n - m + 1, HD = 0$ .

$i = 1, x_1 = \wedge \neq k = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = \wedge \neq t = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = \wedge \neq o = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 5, x_5 = \wedge \neq p = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$i = 6, x_6 = t \neq r = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$i = 7, x_7 = o \neq a = y_7$ , maka  $HD = HD + 1 = 6 + 1 = 7$ .

$i = 8 = n, x_8 = p \neq k = y_8$ , maka  $HD = HD + 1 = 7 + 1 = 8$ .

$HD = 8 > 5 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$i = 9, x_9 = x_8 = p$ .

$i = 8, x_8 = x_7 = o$ .

$i = 7 = j + 1, x_7 = x_6 = t$ .

$x_6 = \wedge$ .

- $k = 1 > 0$ , maka  $i = 1 = k, posisi[1] = 3$ .

Output : 3 dan 5. Artinya, untai  $x$  ada di dalam untai  $y$  pada posisi ke-3 dengan jarak Hamming 5. Sehingga  $x$  adalah subuntai dari  $y$ .

Setelah mendapatkan algoritma tentang pencocokan untai eksak dengan menggunakan ukuran jarak Hamming, terdapat beberapa kasus yang akan dipaparkan berikut ini.

- a. Misal diberikan untai  $x = y = \text{rendang}$ , untai  $x$  dan  $y$  memiliki panjang yang sama yaitu 7.

Maka, perjalanan algoritmanya adalah sebagai berikut:

$$\text{posisi}[1] = 0.$$

$$HD\_Min = n = 7.$$

$$k = 0.$$

$$j = 1 = n - m + 1, HD = 0.$$

$$i = 1, x_1 = r = y_1, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 2, x_2 = e = y_2, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 3, x_3 = n = y_3, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 4, x_4 = d = y_4, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 5, x_5 = a = y_5, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 6, x_6 = n = y_6, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 7, x_7 = g = y_7, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$HD = 0 = n - m, \text{ maka } HD\_Min = HD = 0, k = k + 1 = 0 + 1 = 1,$$

$$\text{posisi}[1] = j = 1.$$

$$k = 1 > 0, \text{ maka } i = 1 = k, \text{ posisi}[1] = 1.$$

Output : 1 dan 0. Artinya, untai  $x$  ada di dalam untai  $y$  pada posisi ke-1 dengan jarak Hamming 0. Karena jarak Hamming-nya 0, maka  $x$  sama dengan  $y$ .

Berdasarkan sifat dari definisi jarak Hamming juga jelas terlihat bahwa jarak Hamming antara keduanya adalah 0. Dari kasus tersebut, dapat diambil sebuah sifat bahwa untai  $x$  dan  $y$  adalah dua buah untai yang sama jika dan hanya jika jarak Hamming antara  $x$  dan  $y$  sama dengan nol atau  $HD(x, y) = 0$ .

- b. Misal diberikan untai  $y = \text{rendang}$  dan untai  $x$  yang tidak mengandung satu karakter pun. Terlihat bahwa  $|y| = n = 7$  dan  $|x| = m = 0$ . Untuk melakukan pencocokan untai dengan ukuran jarak Hamming, maka untai  $x$  akan berisi rangkaian untai hampa sebanyak  $n - m = 7 - 0 = 7$  buah

dengan  $n - m + 1 = 7 - 0 + 1 = 1$  kemungkinan posisi, yaitu

$x = \wedge \wedge \wedge \wedge \wedge \wedge \wedge$ , sehingga

$y = r e n d a n g$

$x = \wedge \wedge \wedge \wedge \wedge \wedge \wedge$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^7 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &\quad + d(x_6, y_6) + d(x_7, y_7) \\
 &= d(\wedge, r) + d(\wedge, e) + d(\wedge, n) + d(\wedge, d) + d(\wedge, a) + d(\wedge, n) \\
 &\quad + d(\wedge, g) \\
 &= 1 + 1 + 1 + 1 + 1 + 1 + 1 \\
 &= 7 \\
 &= |y|
 \end{aligned}$$

Didapat jarak Hamming yaitu  $7 = n - m = n$ . Jika menggunakan algoritma, maka perjalanan algoritmanya adalah sebagai berikut:

$m = 0$ , maka  $k = 2$ ,  $posisi[1] = 0$ ,  $posisi[2] = n + 1 = 7 + 1 = 8$ , dan

$$HD\_Min = n = 7.$$

$k = 2 > 0$ , maka

$$i = 1, posisi[1] = 0.$$

$$i = 2 = k, posisi[2] = 8.$$

Output : 0, 8, dan 8. Karena terdapat dua posisi dan posisi kedua dari untai  $x$  lebih dari panjang  $y$ , artinya untai  $x$  merupakan untai yang tidak mengandung satu karakter pun dengan jarak Hamming sama dengan panjang  $y$  yakni 8.

Terlihat bahwa jarak Hamming antara  $x$  dan  $y$  sama dengan panjang untai  $y$ . Dari kasus ini, dapat diambil sifat bahwa jika panjang  $x$  sama dengan 0 dan panjang  $y$  sama dengan  $n$ , maka jarak Hamming antara  $x$  dan  $y$  sama dengan panjang  $y$  yaitu  $n$ .

- c. Misal diberikan untai  $x = den$  dan untai  $y = dendeng$ . Terlihat bahwa  $|x| = m = 3$  dan  $|y| = n = 7$ . Untuk mencocokkan kedua untai, maka untai  $x$

harus dirangkaikan dengan untai hampa sebanyak  $n - m = 7 - 3 = 4$  buah dengan  $n - m + 1 = 7 - 3 + 1 = 5$  kemungkinan posisi, yaitu:

- Kemungkinan posisi pertama :  $x = den^{^^^}$ , sehingga

$$y = d e n d e n g$$

$$x = d e n \wedge \wedge \wedge \wedge$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^7 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &\quad + d(x_6, y_6) + d(x_7, y_7) \\ &= d(d, d) + d(e, e) + d(n, n) + d(\wedge, d) + d(\wedge, e) + d(\wedge, n) \\ &\quad + d(\wedge, g) \\ &= 0 + 0 + 0 + 1 + 1 + 1 + 1 \\ &= 4 \end{aligned}$$

- Kemungkinan posisi kedua :  $x = ^den^{^^}$ , sehingga

$$y = d e n d e n g$$

$$x = \wedge d e n \wedge \wedge \wedge$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^7 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &\quad + d(x_6, y_6) + d(x_7, y_7) \\ &= d(\wedge, d) + d(d, e) + d(e, n) + d(n, d) + d(\wedge, e) + d(\wedge, n) \\ &\quad + d(\wedge, g) \\ &= 1 + 1 + 1 + 1 + 1 + 1 + 1 \\ &= 7 \end{aligned}$$

- Kemungkinan posisi ketiga :  $x = ^^den^{^^}$ , sehingga

$$y = d e n d e n g$$

$$x = \wedge \wedge d e n \wedge \wedge$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^7 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &\quad + d(x_6, y_6) + d(x_7, y_7) \\
 &= d(^, d) + d(^, e) + d(d, n) + d(e, d) + d(n, e) + d(^, n) \\
 &\quad + d(^, g) \\
 &= 1 + 1 + 1 + 1 + 1 + 1 + 1 \\
 &= 7
 \end{aligned}$$

- Kemungkinan posisi keempat :  $x = \wedge \wedge \wedge den \wedge$ , sehingga

$$y = d e n d e n g$$

$$x = \wedge \wedge \wedge d e n \wedge$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^7 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &\quad + d(x_6, y_6) + d(x_7, y_7) \\
 &= d(^, d) + d(^, e) + d(^, n) + d(d, d) + d(e, e) + d(n, n) \\
 &\quad + d(^, g) \\
 &= 1 + 1 + 1 + 0 + 0 + 0 + 1 \\
 &= 4
 \end{aligned}$$

- Kemungkinan posisi kelima :  $x = \wedge \wedge \wedge \wedge den$ , sehingga

$$y = d e n d e n g$$

$$x = \wedge \wedge \wedge \wedge d e n$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^7 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &\quad + d(x_6, y_6) + d(x_7, y_7) \\
 &= d(^, d) + d(^, e) + d(^, n) + d(^, d) + d(d, e) + d(e, n) \\
 &\quad + d(n, g) \\
 &= 1 + 1 + 1 + 1 + 1 + 1 + 1
 \end{aligned}$$



$$= 7$$

Didapat jarak Hamming terkecil yaitu  $4 = n - m$  pada posisi ke-1 dan ke-4.  
Jika menggunakan algoritma, maka perjalanan algoritmanya adalah sebagai berikut:

$$posisi[1] = 0.$$

$m \neq n$ , maka

$$i = 1, x_4 = \wedge.$$

$$i = 2, x_5 = \wedge.$$

$$i = 3, x_6 = \wedge.$$

$$i = 4 = n - m, x_7 = \wedge.$$

$$HD\_Min = n = 7.$$

$$k = 0.$$

- $j = 1, HD = 0.$

$$i = 1, x_1 = d = y_1, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 2, x_2 = e = y_2, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 3, x_3 = n = y_3, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 4, x_4 = \wedge \neq d = y_4, \text{ maka } HD = HD + 1 = 0 + 1 = 1.$$

$$i = 5, x_5 = \wedge \neq e = y_5, \text{ maka } HD = HD + 1 = 1 + 1 = 2.$$

$$i = 6, x_6 = \wedge \neq n = y_6, \text{ maka } HD = HD + 1 = 2 + 1 = 3.$$

$$i = 7 = n, x_7 = \wedge \neq g = y_7, \text{ maka } HD = HD + 1 = 3 + 1 = 4.$$

$$HD = 4 = n - m, \text{ maka } HD\_Min = HD = 4,$$

$$k = k + 1 = 0 + 1 = 1, posisi[1] = j = 1.$$

$m \neq n$ , maka

$$i = 4, x_4 = x_3 = n.$$

$$i = 3, x_3 = x_2 = e.$$

$$i = 2 = j + 1, x_2 = x_1 = d.$$

$$x_1 = \wedge.$$

- $j = 2, HD = 0.$

$$i = 1, x_1 = \wedge \neq d = y_1, \text{ maka } HD = HD + 1 = 0 + 1 = 1.$$

$$i = 2, x_2 = d \neq e = y_2, \text{ maka } HD = HD + 1 = 1 + 1 = 2.$$

$$i = 3, x_3 = e \neq n = y_3, \text{ maka } HD = HD + 1 = 2 + 1 = 3.$$

$$i = 4, x_4 = n \neq d = y_4, \text{ maka } HD = HD + 1 = 3 + 1 = 4.$$

$i = 5, x_5 = \wedge \neq e = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$i = 6, x_6 = \wedge \neq n = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$i = 7 = n, x_7 = \wedge \neq g = y_7$ , maka  $HD = HD + 1 = 6 + 1 = 7$ .

$HD = 7 > 4 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$i = 5, x_5 = x_4 = n$ .

$i = 4, x_4 = x_3 = e$ .

$i = 3 = j + 1, x_3 = x_2 = d$ .

$x_2 = \wedge$ .

- $j = 3, HD = 0$ .

$i = 1, x_1 = \wedge \neq d = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = d \neq n = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = e \neq d = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 5, x_5 = n \neq e = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$i = 6, x_6 = \wedge \neq n = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$i = 7 = n, x_7 = \wedge \neq g = y_7$ , maka  $HD = HD + 1 = 6 + 1 = 7$ .

$HD = 7 > 4 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$i = 6, x_6 = x_5 = n$ .

$i = 5, x_5 = x_4 = e$ .

$i = 4 = j + 1, x_4 = x_3 = d$ .

$x_3 = \wedge$ .

- $j = 4, HD = 0$ .

$i = 1, x_1 = \wedge \neq d = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = \wedge \neq n = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = d = y_4$ , maka  $HD = HD + 0 = 3 + 0 = 3$ .

$i = 5, x_5 = e = y_5$ , maka  $HD = HD + 0 = 3 + 0 = 3$ .

$i = 6, x_6 = n = y_6$ , maka  $HD = HD + 0 = 3 + 0 = 3$ .

$i = 7 = n, x_7 = \wedge \neq g = y_7$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$HD = 4 = n - m$ , maka  $HD\_Min = HD = 4$ ,

$$k = k + 1 = 1 + 1 = 2, \text{posisi}[2] = j = 4.$$

$m \neq n$ , maka

$$i = 7, x_7 = x_6 = n.$$

$$i = 6, x_6 = x_5 = e.$$

$$i = 5 = j + 1, x_5 = x_4 = d.$$

$$x_4 = \wedge.$$

- $j = 5 = n - m + 1, HD = 0.$

$$i = 1, x_1 = \wedge \neq d = y_1, \text{ maka } HD = HD + 1 = 0 + 1 = 1.$$

$$i = 2, x_2 = \wedge \neq e = y_2, \text{ maka } HD = HD + 1 = 1 + 1 = 2.$$

$$i = 3, x_3 = \wedge \neq n = y_3, \text{ maka } HD = HD + 1 = 2 + 1 = 3.$$

$$i = 4, x_4 = \wedge \neq d = y_4, \text{ maka } HD = HD + 1 = 3 + 1 = 4.$$

$$i = 5, x_5 = d \neq e = y_5, \text{ maka } HD = HD + 1 = 4 + 1 = 5.$$

$$i = 6, x_6 = e \neq n = y_6, \text{ maka } HD = HD + 1 = 5 + 1 = 6.$$

$$i = 7, x_7 = n \neq g = y_7, \text{ maka } HD = HD + 1 = 6 + 1 = 7.$$

$HD = 7 > 5 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$$i = 8, x_8 = x_7 = n.$$

$$i = 7, x_7 = x_6 = e.$$

$$i = 6 = j + 1, x_6 = x_5 = d.$$

$$x_5 = \wedge.$$

$k = 2$ , maka

$$i = 1, \text{posisi}[1] = 1.$$

$$i = 2 = k, \text{posisi}[2] = 4.$$

Output : 1, 4, dan 4. Artinya, untai  $x$  ada di dalam untai  $y$  pada posisi ke-1 dan ke-4 dengan jarak Hamming 4. Sehingga  $x$  adalah subuntai dari  $y$ .

Posisi Karakter	:	1	2	3	4	5	6	7
$y =$		<b>d</b>	<b>e</b>	<b>n</b>	<b>d</b>	<b>e</b>	<b>n</b>	<b>g</b>
Kemungkinan Posisi ke-1	:	<b>d</b>	<b>e</b>	<b>n</b>	^	^	^	^
Kemungkinan Posisi ke-2	:	^	<b>d</b>	<b>e</b>	<b>n</b>	^	^	^
Kemungkinan Posisi ke-3	:	^	^	<b>d</b>	<b>e</b>	<b>n</b>	^	^
Kemungkinan Posisi ke-4	:	^	^	^	<b>d</b>	<b>e</b>	<b>n</b>	^
Kemungkinan Posisi ke-5	:	^	^	^	^	<b>d</b>	<b>e</b>	<b>n</b>

**Gambar 3.17.** Kemungkinan posisi untai hampa yang dirangkaikan pada untai *den*.

Pada kasus ini, diketahui bahwa  $|x| = m = 3$  dan  $|y| = n = 7$  mengharuskan perangkaian untai hampa dengan untai  $x$  sebanyak  $n - m = 7 - 3 = 4$  buah dan menghasilkan 5 kemungkinan posisi penempatan untai hampa. Kemudian didapat jarak Hamming terkecil yaitu  $4 = n - m$  pada posisi ke-1 dan ke-4 dibandingkan dengan jarak Hamming pada posisi-posisi yang lain. Karena terdapat dua kemungkinan posisi dengan jarak Hamming terkecil, maka untai  $x$  merupakan subuntai dari untai  $y$  yang berada posisi ke-2 dan ke-5 dalam untai  $y$ . Jadi, terdapat  $k$  subuntai dari  $y$  yang sama dengan  $x$  jika dan hanya jika jarak Hamming antara  $x$  dan  $y$  pada ke- $k$  posisi tersebut merupakan jarak Hamming minimum yang sama dengan  $n - m$ .

Contoh lain, diberikan untai  $x = \textit{per}$  dan untai  $y = \textit{pemek}$ . Terlihat bahwa  $|x| = m = 3$  dan  $|y| = n = 6$ . Untuk mencocokkan kedua untai, maka untai  $x$  harus dirangkaikan dengan untai hampa sebanyak  $n - m = 6 - 3 = 3$  buah dengan  $n - m + 1 = 6 - 3 + 1 = 4$  kemungkinan posisi, yaitu:

- Kemungkinan posisi pertama :  $x = \textit{per}^{\wedge\wedge\wedge}$ , sehingga

$$y = \textit{p e m p e k}$$

$$x = \textit{p e r}^{\wedge\wedge\wedge}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^6 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &\quad + d(x_6, y_6) \\ &= d(p, p) + d(e, e) + d(r, m) + d(^{\wedge}, p) + d(^{\wedge}, e) + d(^{\wedge}, k) \end{aligned}$$

$$= 0 + 0 + 1 + 1 + 1 + 1$$

$$= 4$$

- Kemungkinan posisi kedua :  $x = \text{^per^}$ , sehingga

$$y = p e m p e k$$

$$x = \text{^ p e r ^}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$HD(x, y) = \sum_{i=1}^6 d(x_i, y_i)$$

$$= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) + d(x_6, y_6)$$

$$= d(\text{^}, p) + d(p, e) + d(e, m) + d(r, p) + d(\text{^}, e) + d(\text{^}, k)$$

$$= 1 + 1 + 1 + 1 + 1 + 1$$

$$= 6$$

- Kemungkinan posisi ketiga :  $x = \text{^^per^}$ , sehingga

$$y = p e m p e k$$

$$x = \text{^ ^ p e r ^}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$HD(x, y) = \sum_{i=1}^6 d(x_i, y_i)$$

$$= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) + d(x_6, y_6)$$

$$= d(\text{^}, p) + d(\text{^}, e) + d(p, m) + d(e, p) + d(r, e) + d(\text{^}, k)$$

$$= 1 + 1 + 1 + 1 + 1 + 1$$

$$= 6$$

- Kemungkinan posisi keempat :  $x = \text{^^^per}$ , sehingga

$$y = p e m p e k$$

$$x = \text{^ ^ ^ p e r}$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$HD(x, y) = \sum_{i=1}^6 d(x_i, y_i)$$

$$= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5)$$

$$\begin{aligned}
& +d(x_6, y_6) \\
& = d(\wedge, p) + d(\wedge, e) + d(\wedge, m) + d(p, p) + d(e, e) + d(r, k) \\
& = 1 + 1 + 1 + 0 + 0 + 1 \\
& = 4
\end{aligned}$$

Didapat jarak Hamming terkecil yaitu 4. Jika menggunakan algoritma, maka perjalanan algoritmanya adalah sebagai berikut:

$posisi[1] = 0$ .

$m \neq n$ , maka

$$i = 1, x_4 = \wedge.$$

$$i = 2, x_5 = \wedge.$$

$$i = 3 = n - m, x_6 = \wedge.$$

$HD\_Min = n = 6$ .

$k = 0$ .

•  $j = 1, HD = 0$ .

$$i = 1, x_1 = p = y_1, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 2, x_2 = e = y_2, \text{ maka } HD = HD + 0 = 0 + 0 = 0.$$

$$i = 3, x_3 = r \neq m = y_3, \text{ maka } HD = HD + 0 = 0 + 1 = 1.$$

$$i = 4, x_4 = \wedge \neq p = y_4, \text{ maka } HD = HD + 1 = 1 + 1 = 2.$$

$$i = 5, x_5 = \wedge \neq e = y_5, \text{ maka } HD = HD + 1 = 2 + 1 = 3.$$

$$i = 6 = n, x_6 = \wedge \neq k = y_6, \text{ maka } HD = HD + 1 = 3 + 1 = 4.$$

$$HD = 4 \leq HD\_Min, \text{ maka } HD\_Min = HD = 4.$$

$m \neq n$ , maka

$$i = 4, x_4 = x_3 = r.$$

$$i = 3, x_3 = x_2 = e.$$

$$i = 2 = j + 1, x_2 = x_1 = p.$$

$$x_1 = \wedge.$$

•  $j = 2, HD = 0$ .

$$i = 1, x_1 = \wedge \neq p = y_1, \text{ maka } HD = HD + 1 = 0 + 1 = 1.$$

$$i = 2, x_2 = p \neq e = y_2, \text{ maka } HD = HD + 1 = 1 + 1 = 2.$$

$$i = 3, x_3 = e \neq m = y_3, \text{ maka } HD = HD + 1 = 2 + 1 = 3.$$

$$i = 4, x_4 = r \neq p = y_4, \text{ maka } HD = HD + 1 = 3 + 1 = 4.$$

$$i = 5, x_5 = \wedge \neq e = y_5, \text{ maka } HD = HD + 1 = 4 + 1 = 5.$$

$i = 6 = n, x_6 = \wedge \neq k = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .  
 $HD = 6 > 4 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$$i = 5, x_5 = x_4 = r.$$

$$i = 4, x_4 = x_3 = e.$$

$$i = 3 = j + 1, x_3 = x_2 = p.$$

$$x_2 = \wedge.$$

- $j = 3, HD = 0$ .

$i = 1, x_1 = \wedge \neq p = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = p \neq m = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = e \neq p = y_4$ , maka  $HD = HD + 0 = 3 + 1 = 4$ .

$i = 5, x_5 = r \neq e = y_5$ , maka  $HD = HD + 0 = 4 + 1 = 5$ .

$i = 6 = n, x_6 = \wedge \neq k = y_6$ , maka  $HD = HD + 1 = 5 + 1 = 6$ .

$HD = 6 > 4 = HD\_Min$ , maka lanjutkan  $j$ .

$m \neq n$ , maka

$$i = 6, x_6 = x_5 = r.$$

$$i = 5, x_5 = x_4 = e.$$

$$i = 4 = j + 1, x_4 = x_3 = p.$$

$$x_3 = \wedge.$$

- $j = 4 = n - m + 1, HD = 0$ .

$i = 1, x_1 = \wedge \neq p = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = \wedge \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = \wedge \neq m = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = p = y_4$ , maka  $HD = HD + 0 = 3 + 0 = 3$ .

$i = 5, x_5 = e = y_5$ , maka  $HD = HD + 0 = 3 + 0 = 3$ .

$i = 6 = n, x_6 = r \neq k = y_6$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$HD = 4 \leq HD\_Min$ , maka  $HD\_Min = HD = 4$ .

$m \neq n$ , maka

$$i = 7, x_7 = x_6 = r.$$

$$i = 6, x_6 = x_5 = e.$$

$$i = 5 = j + 1, x_5 = x_4 = p.$$

$$x_4 = \wedge.$$

$k = 0$ , maka hasilnya 0.

Output : 0 dan 4. Artinya, untai  $x$  tidak ada di dalam untai  $y$  tetapi jarak Hamming-nya adalah 4. Sehingga  $x$  bukan subuntai dari  $y$ .

Posisi Karakter	: 1 2 3 4 5 6
$y =$	$p e m p e k$
Kemungkinan Posisi ke-1	: $p e r \wedge \wedge \wedge$
Kemungkinan Posisi ke-2	: $\wedge p e r \wedge \wedge$
Kemungkinan Posisi ke-3	: $\wedge \wedge p e r \wedge$
Kemungkinan Posisi ke-4	: $\wedge \wedge \wedge p e r$

**Gambar 3.18.** Kemungkinan posisi untai hampa yang dirangkaikan pada untai  $per$ .

Pada contoh ini, didapat jarak Hamming terkecil yaitu 4, namun tidak sama dengan  $n - m$ , maka untai  $x$  bukan subuntai dari untai  $y$ .

- d. Jika diberikan untai  $x = ayam$  dan untai  $y = telur$ . Terlihat bahwa  $|x| = m = 4$  dan  $|y| = n = 5$ . Juga tidak terdapat satupun karakter pada untai  $x$  yang sama dengan karakter pada untai  $y$ . Untuk mencocokkan kedua untai, maka untai  $x$  harus dirangkaikan dengan untai hampa sebanyak  $n - m = 5 - 4 = 1$  buah dengan  $n - m + 1 = 5 - 4 + 1 = 2$  kemungkinan posisi, yaitu:

- Kemungkinan posisi pertama :  $x = ayam\wedge$ , sehingga
 
$$y = t e l u r$$

$$x = a y a m \wedge$$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned}
 HD(x, y) &= \sum_{i=1}^5 d(x_i, y_i) \\
 &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\
 &= d(a, t) + d(y, e) + d(a, l) + d(m, u) + d(\wedge, r) \\
 &= 1 + 1 + 1 + 1 + 1 \\
 &= 5 \\
 &= |y|
 \end{aligned}$$



- Kemungkinan posisi kedua :  $x = \text{^}ayam$ , sehingga

$y = t e l u r$

$x = \text{^} a y a m$

Maka, jarak Hamming antara untai  $x$  dan  $y$  di atas adalah:

$$\begin{aligned} HD(x, y) &= \sum_{i=1}^5 d(x_i, y_i) \\ &= d(x_1, y_1) + d(x_2, y_2) + d(x_3, y_3) + d(x_4, y_4) + d(x_5, y_5) \\ &= d(\text{^}, t) + d(a, e) + d(y, l) + d(a, u) + d(m, r) \\ &= 1 + 1 + 1 + 1 + 1 \\ &= 5 \\ &= |y| \end{aligned}$$

Didapat jarak Hamming terkecil yaitu  $5 = |y|$ . Jika menggunakan algoritma, maka perjalanan algoritmanya adalah sebagai berikut:

$posisi[1] = 0$ .

$m \neq n$ , maka

$$i = 1 = n - m, x_5 = \text{^}.$$

$HD\_Min = n = 5$ .

$k = 0$ .

- $j = 1, HD = 0$ .

$i = 1, x_1 = a \neq t = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1$ .

$i = 2, x_2 = y \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2$ .

$i = 3, x_3 = a \neq l = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3$ .

$i = 4, x_4 = m \neq u = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4$ .

$i = 5, x_5 = \text{^} \neq r = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5$ .

$HD = 5 \leq HD\_Min$ , maka  $HD\_Min = HD = 5$ .

$m \neq n$ , maka

$$i = 5, x_5 = x_4 = m.$$

$$i = 4, x_4 = x_3 = a.$$

$$i = 3, x_3 = x_2 = y.$$

$$i = 2 = j + 1, x_2 = x_1 = a.$$

$$x_1 = \text{^}.$$

- $j = 2 = n - m + 1, HD = 0.$

$i = 1, x_1 = \wedge \neq t = y_1$ , maka  $HD = HD + 1 = 0 + 1 = 1.$

$i = 2, x_2 = a \neq e = y_2$ , maka  $HD = HD + 1 = 1 + 1 = 2.$

$i = 3, x_3 = y \neq l = y_3$ , maka  $HD = HD + 1 = 2 + 1 = 3.$

$i = 4, x_4 = a \neq u = y_4$ , maka  $HD = HD + 1 = 3 + 1 = 4.$

$i = 5, x_5 = m \neq r = y_5$ , maka  $HD = HD + 1 = 4 + 1 = 5.$

$HD = 5 \leq HD\_Min$ , maka  $HD\_Min = HD = 5.$

$m \neq n$ , maka

$i = 6, x_6 = x_5 = m.$

$i = 5, x_5 = x_4 = a.$

$i = 4, x_4 = x_3 = y.$

$i = 3 = j + 1, x_3 = x_2 = a.$

$x_2 = \wedge.$

$k = 0$ , maka hasilnya 0.

Output : 0 dan 5. Artinya, untai  $x$  tidak ada di dalam untai  $y$  tetapi jarak Hamming-nya adalah 5. Sehingga  $x$  bukan subuntai dari  $y$ .

Posisi Karakter	:	1	2	3	4	5
		$y$	$=$	$t$	$e$	$l$
				$a$	$y$	$a$
Kemungkinan Posisi ke-1	:	$a$	$y$	$a$	$m$	$\wedge$
Kemungkinan Posisi ke-2	:	$\wedge$	$a$	$y$	$a$	$m$

**Gambar 3.19.** Kemungkinan posisi untai hampa yang dirangkaikan pada untai ayam.

Pada contoh ini, didapat jarak Hamming yang sama dari dua kemungkinan yaitu 5. Diketahui pula bahwa tidak terdapat satupun karakter dari untai  $x$  yang sama dengan karakter pada untai  $y$ . Sehingga jarak Hamming yang didapat untuk semua kemungkinan sama dengan  $n = 5$  namun untai  $x$  dipastikan bukan subuntai dari untai  $y$ .

Terlihat dari contoh ini, jika tidak terdapat satupun karakter dari untai  $x$  yang sama dengan karakter pada untai  $y$ , maka jarak Hamming yang didapat untuk

semua kemungkinan sama dengan panjang  $y$  yaitu  $n$ . Sehingga, untai  $x$  dipastikan bukan subuntai dari untai  $y$ .

Berikut ini adalah ringkasan dari beberapa kasus yang telah dipaparkan di atas.

- a. Untai  $x$  dan  $y$  adalah dua buah untai yang sama jika dan hanya jika jarak Hamming antara  $x$  dan  $y$  sama dengan nol atau  $HD(x, y) = 0$ .
- b. Jika panjang  $x$  sama dengan 0 dan panjang  $y$  sama dengan  $n$ , maka jarak Hamming antara  $x$  dan  $y$  sama dengan panjang  $y$  yaitu  $n$ .
- c. Terdapat  $k$  subuntai dari  $y$  yang sama dengan  $x$  jika dan hanya jika jarak Hamming antara  $x$  dan  $y$  pada ke- $k$  posisi tersebut merupakan jarak Hamming minimum yang sama dengan  $n - m$ .
- d. Jika tidak terdapat satupun karakter dari untai  $x$  yang sama dengan karakter pada untai  $y$ , maka jarak Hamming yang didapat untuk semua kemungkinan sama dengan panjang  $y$  yaitu  $n$ . Sehingga, untai  $x$  dipastikan bukan subuntai dari untai  $y$ .

## BAB 4

### IMPLEMENTASI DAN SIMULASI PROGRAM

#### 4.1. Implementasi

Permasalahan pencocokan untai (*string matching*) merupakan permasalahan yang sudah tidak asing lagi dalam dunia komputer. Contoh implementasi dari permasalahan pencocokan untai ini adalah pencocokan sebuah untai kata pada *Microsoft Word* ataupun teks editor yang lain. Misalkan, dicari sebuah kata “*ekstra*” pada sebuah halaman yang kemudian akan diganti dengan kata “*super*”. Dengan menggunakan tools “*find and replace*” yang terdapat pada *Microsoft Word* ataupun teks editor yang lain, maka akan tampil semua kata yang mengandung kata “*ekstra*” maupun kata “*ekstra*” itu sendiri dan selanjutnya tinggal dipilih bagian kata mana yang akan diganti dengan kata “*super*”.

Dalam kasus yang lebih besar lagi, pencocokan untai digunakan pada website dengan memasukkan kata-kata kunci sebagaimana yang telah diimplementasikan pada mesin pencari seperti *Yahoo* maupun *Google*. Ambil contoh mesin pencari yang sedang populer, *Google*, jika diperhatikan terkadang pengguna internet mengetikkan kata-kata yang salah sehingga *Google* memberikan saran pencarian untuk mencari kata-kata yang benar. Misalkan dicari kata “*bakso*”, namun pengguna salah mengetikkan kata sehingga menjadi “*baksa*”. Terkadang *Google* memberikan pernyataan “Mungkin maksud anda adalah: *bakso*”. Walaupun demikian, semua dokumen yang mengandung kata “*baksa*” tetap ditampilkan. Saat menampilkan semua dokumen yang mengandung kata “*baksa*”, pencocokan untai yang digunakan adalah pencocokan untai eksak, namun pada saat menampilkan pernyataan “Mungkin maksud anda adalah: *bakso*”, maka pencocokan hampiran untai yang digunakan.

Dalam ilmu biologi komputasi atau yang lebih dikenal dengan *bioinformatic*, pencocokan untaipun digunakan untuk mencocokkan suatu untai DNA. Namun biasanya, yang digunakan di sini adalah pencocokan hampiran untai, karena tujuan utama pencocokan untai DNA biasanya mencari kemiripan

antara dua untai DNA dari dua buah objek berbeda yang tidak 100% memiliki untai DNA yang sama.

#### 4.2. Simulasi Program

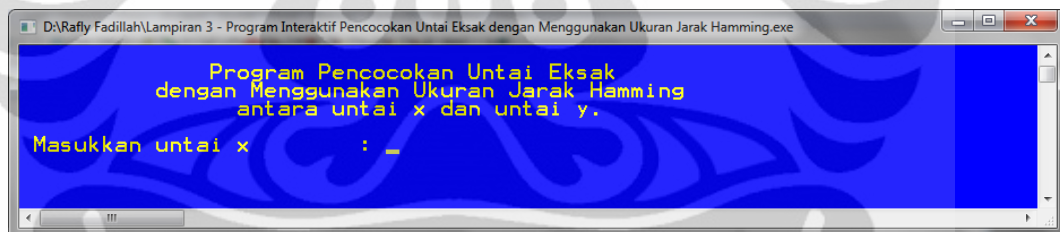
Setelah mendapatkan algoritma yang diinginkan yakni algoritma *pencocokan\_untai\_eksak\_dengan\_ukuran\_jarak\_hamming(x,y)*, maka dibuatlah suatu program sederhana yang dapat melakukan pencocokan untai eksak dengan menggunakan ukuran jarak Hamming. Program ini dibuat dengan menggunakan bahasa C pada aplikasi *Borland C++ Ver. 5.02*. Program dapat dilihat pada lampiran 3 ataupun pada lampiran (CD). Berikut ini adalah simulasinya.

Misalkan, kita gunakan contoh 3.14. pada bab sebelumnya.

Jika diberikan:  $x = top \rightarrow |x| = m = 3 \neq 0$

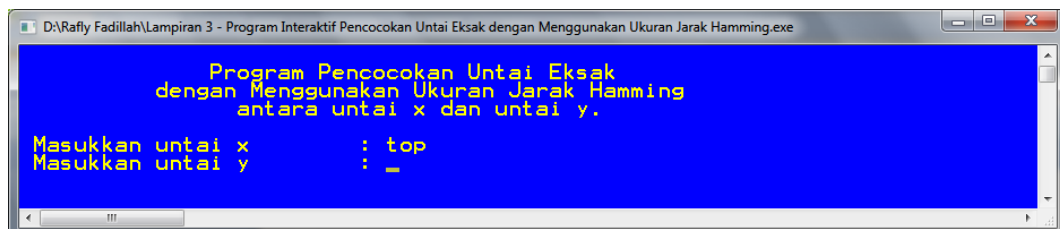
$y = ketoprak \rightarrow |y| = n = 8 \neq 0$

Maka, simulasi programnya adalah sebagai berikut.



**Gambar 4.1.** Tampilan awal pada *command window*

Kita masukkan untai  $x = top$ , kemudian tekan enter. Maka akan muncul permintaan untuk memasukkan untai berikutnya, yaitu untai  $y$ .



**Gambar 4.2.** Tampilan input untai berikutnya pada *command window*

Kemudian kita masukkan untai  $y = \text{ketoprak}$ , kemudian tekan enter. Karena panjang untai  $x$  tidak sama dengan panjang untai  $y$ , maka akan dilakukan perangkaian untai hampa dengan untai  $x$  sebanyak  $n - m = 5$  buah dengan  $n - m + 1 = 6$  kemungkinan posisi. Maka akan muncul 6 kemungkinan penempatan untai hampa serta jarak Hamming untuk masing-masing kemungkinan posisi seperti yang akan ditampilkan berikut ini.

```

D:\Rafly Fadillah\Lampiran 3 - Program Interaktif Pencocokan Untai Eksak dengan Menggunakan Ukuran Jarak Hamming.exe

Program Pencocokan Untai Eksak
dengan Menggunakan Ukuran Jarak Hamming
antara untai x dan untai y.

Masukkan untai x      : top
Masukkan untai y      : ketoprak

panjang x = m = 3
panjang y = n = 8

Karena panjang untai x dan untai y tidak sama, maka harus dilakukan
penambahan untai hampa pada untai x.
Sehingga didapatkan kemungkinan-kemungkinan posisinya sebagai berikut:

1. Kemungkinan posisi ke-1
   y = ketoprak
   x = top^^^^
   HD = 8          --> HD_Min = 8

```

**Gambar 4.3.** Tampilan kemungkinan posisi pertama pada *command window*

Pada kemungkinan posisi pertama, didapat jarak Hamming-nya yaitu 8.

```

D:\Rafly Fadillah\Lampiran 3 - Program Interaktif Pencocokan Untai Eksak dengan Menggunakan Ukuran Jarak Hamming.exe

2. Kemungkinan posisi ke-2
   y = ketoprak
   x = ^top^^^^
   HD = 8          --> HD_Min = 8

3. Kemungkinan posisi ke-3
   y = ketoprak
   x = ^^top^^
   HD = n - m = 5  --> HD_Min = 5

4. Kemungkinan posisi ke-4
   y = ketoprak
   x = ^^top^^
   HD = 8          --> HD_Min = 5

```

**Gambar 4.4.** Tampilan kemungkinan posisi ke-2 hingga ke-4 pada *command window*



## BAB 5

### KESIMPULAN

Pada bab ini, akan diberikan beberapa kesimpulan yang dihasilkan dari pembahasan pada bab-bab sebelumnya sehingga menjawab semua tujuan dari penulisan skripsi ini.

- a. Jarak Hamming hanya dapat berlaku untuk dua buah untai yang memiliki panjang yang sama. Pada pencocokan untai eksak, panjang dua buah untai yang diberikan tidak harus sama. Jadi, jika untai  $x$  dan  $y$  yang diberikan memiliki panjang yang tidak sama, maka untai yang lebih pendek harus dirangkaikan dengan untai hampa agar panjang kedua buah untai menjadi sama. Jika  $|x| = m$ ,  $|y| = n$ , dan  $m < n$ , maka untai  $x$  harus dirangkaikan dengan untai hampa sebanyak  $n - m$  buah dengan  $n - m + 1$  kemungkinan posisi.
- b. Dari  $n - m + 1$  kemungkinan posisi, dicari posisi dengan jarak Hamming terkecil. Jika didapat jarak Hamming terkecil sama dengan  $n - m$  pada posisi ke- $i$ , maka disimpulkan bahwa untai  $x$  ada di dalam untai  $y$  pada posisi ke- $i$ .
- c. Dengan menggunakan metode-metode yang telah dibahas sebelumnya, akhirnya didapatkanlah suatu algoritma pencocokan untai eksak menggunakan ukuran jarak Hamming yaitu  
*Algoritma pencocokan\_untai\_eksak\_dengan\_ukuran\_jarak\_hamming( $x, y$ )*  
dengan kompleksitas waktu  $O(n^2)$ .



## DAFTAR PUSTAKA

- Bartle, Robert G., & Sherbert, Donald R. (2000). *Introduction to real analysis* (3<sup>rd</sup> ed.). New York: John Wiley & Sons.
- Davis, Martin D., & Weyuker, Elaine W. (1983). *Computability, complexity, and language, fundamentals of theoretical computer science*. New York: Academic Press.
- Faro, Simone, & Lecroq, Thierry. (2010). *The exact string matching problem: a comprehensive experimental evaluation*. New York: Springer.
- Lecroq, Thierry, & Charras, Christian. (2004). *Handbook of exact string-matching algorithms*. London: King's College.
- Lewis, Harry R., & Papadimitriou, Christos H. (1998). *Elements of the theory of computation* (2<sup>nd</sup> ed.). New Jersey: Prentice Hall.
- Rosen, Kenneth H. (2009). *Discrete mathematics and its applications* (4<sup>th</sup> ed.). New York: McGraw-Hill.
- Wibisono, Samuel. (2008). *Matematika diskrit* (edisi ke-2). Yogyakarta: Graha Ilmu.

## LAMPIRAN-LAMPIRAN

### Lampiran 1. Listing program interaktif pencocokan untai eksak lebih dari 1 subuntai

```
#include <dos.h> //sleep();
#include <stdio.h> //gets();
#include <conio.h> //clrscr(); gotoxy();
#include <stdlib.h>
#include <string.h> //strlen();
#include <ctype.h> //tolower();

char x[500], y[500], x2[500], y2[500], opsi[500];
int m, n, i, j, k, posisi[500];

void brute_force(void);

main()
{
    clrscr();
    opsi[0] = 's';
    while (opsi[0] == 's' || opsi[0] == 'x' || opsi[0] == 'y')
    {
        clrscr();
        printf(" \n");
        printf(" \t Program Pencocokan Untai Eksak \n");
        printf(" \t Apakah untai x ada di dalam untai y \n");
        printf(" \n\n");

        if (opsi[0] == 's')
        {
            printf(" Masukkan untai x \t : ");
            gets(x);
            m = strlen(x);
            printf(" Masukkan untai y \t : ");
            gets(y);
            n = strlen(y);
        }

        if (opsi[0] == 'x')
        {
            printf(" Masukkan untai x \t : \n");
            printf(" Masukkan untai y \t : %s ", y);
            gotoxy(28,6); gets(x);
            m = strlen(x);
            printf(" \n");
        }

        if (opsi[0] == 'y')
        {
            printf(" Masukkan untai x \t : %s ", x);
            printf(" \n");
            printf(" Masukkan untai y \t : ");
            gets(y);
            n = strlen(y);
        }

        while (m == 0 || n == 0)
        {
```

```

gotoxy(1,8); printf(" Tidak Boleh Kosong !!!");

if (m == 0)
{
    gotoxy(28,6); gets(x);
    m = strlen(x);
}

else
{
    gotoxy(28,7); gets(y);
    n = strlen(y);
}

strcpy(x2,x);
strcpy(y2,y);
strlwr(x2);
strlwr(y2);

printf(" \n");
printf(" Untai x \t\t : %s\n", x);
printf(" Panjang Untai x \t : %d\n", m);
printf(" Untai y \t\t : %s\n", y);
printf(" Panjang Untai y \t : %d\n", n);
printf(" \n\n", n);

for (k = 0; k < 70; k++)
{
    printf("-");
}

printf("\n\n");

if (m <= n)
{
    brute_force();
}

else
{
    printf(" \t Untai x \t = '%s'\n\n", x);
    printf(" \t\t TIDAK MUNGKIN ADA di dalam\n\n");
    printf(" \t Untai y \t = '%s'.\n\n",y);
    printf(" \t Karena |x| = %d lebih besar dari |y| = %d. ", m, n);
}

getchar();
printf(" \n\n");
printf(" Anda Ingin Coba Lagi ? \n\n");
printf(" \t Jika ingin merubah x, \t\t pilih 'x' \n\n");
printf(" \t Jika ingin merubah y, \t\t pilih 'y' \n\n");
printf(" \t Jika ingin merubah x dan y, \t pilih 's' \n\n");
printf(" \t Jika ingin keluar, \t\t pilih 'e' \n\n");
printf(" Pilihan Anda : ");
gets(opsi);
tolower(opsi[0]);

while (strlen(opsi) != 1 || (opsi[0] != 'x' && opsi[0] != 'y' &&
opsi[0] != 's' && opsi[0] != 'e'))
{
    printf(" Input Salah. Ulangi : ");
    gets(opsi);
    tolower(opsi[0]);
}

```

```

    }
}

void brute_force(void)
{
    k = 0;
    for (j = 0; j <= n-m; j++)
    {
        i = 0;
        while (i < m && x[i] == y[i+j])
        {
            i++;
        }

        if (i >= m)
        {
            k++;
            posisi[k] = j+1;
        }
    }

    if (k > 1)
    {
        printf(" \t Untai x \t = '%s'\n\n", x);
        printf(" \t\t ADA di dalam\n\n");
        printf(" \t Untai y \t = '%s'\n\n", y);
        printf(" \t dengan karakter awal untaian x berada pada posisi ke ");

        for (i = 1; i <= k-1; i++)
        {
            printf("%d, ", posisi[i]);
        }

        printf("dan %d. ", posisi[k]);
    }

    else if (k == 1)
    {
        if(m == n)
        {
            printf(" \t Untai x \t = '%s'\n\n", x);
            printf(" \t\t SAMA DENGAN\n\n");
            printf(" \t Untai y \t = '%s'.", y);
        }
        else
        {
            printf(" \t Untai x \t = '%s'\n\n", x);
            printf(" \t\t ADA di dalam\n\n");
            printf(" \t Untai y \t = '%s'\n\n", y);
            printf(" \t dengan karakter awal untaian x berada pada posisi ke
            %d.", posisi[k]);
        }
    }

    else
    {
        printf(" \t Untai x \t = '%s'\n\n", x);
        printf(" \t\t TIDAK ADA di dalam\n\n");
        printf(" \t Untai y \t = '%s'.", y);
    }
}
}

```

## Lampiran 2. Listing program interaktif penghitungan jarak hamming

```
#include <dos.h>           //sleep();
#include <stdio.h>         //gets();
#include <conio.h>         //clrscr(); gotoxy();
#include <stdlib.h>
#include <string.h>       //strlen();
#include <ctype.h>        //tolower();

char x[500], y[500], x2[500], y2[500], opsi[500];
int m, n, i, k, HD;

main()
{
    clrscr();
    opsi[0] = 's';
    while (opsi[0] == 's' || opsi[0] == 'x' || opsi[0] == 'y')
    {
        clrscr();
        printf(" \n");
        printf(" \t      Program Penghitungan Jarak Hamming \n");
        printf(" \t      Antara untai x dengan untai y. \n");
        printf(" \n\n");

        if (opsi[0] == 's')
        {
            printf(" Masukkan untai x \t : ");
            gets(x);
            m = strlen(x);
            printf(" Masukkan untai y \t : ");
            gets(y);
            n = strlen(y);

            while (m != n)
            {
                printf(" \n\n");
                printf(" Panjang Untai Harus Sama !!! \n\n");
                printf(" Untai y : "); gets(y);
                n = strlen(y);
            }
        }

        if (opsi[0] == 'x')
        {
            printf(" Masukkan untai x \t : \n");
            printf(" Masukkan untai y \t : %s ", y);
            gotoxy(28,6); gets(x);
            m = strlen(x);

            while (m != n)
            {
                printf(" \n\n");
                printf(" Panjang Untai Harus Sama !!! \n\n");
                printf(" Untai x : "); gets(x);
                m = strlen(x);
            }
        }

        if (opsi[0] == 'y')
        {
            printf(" Masukkan untai x \t : %s ", x);
            printf(" \n");
            printf(" Masukkan untai y \t : ");
        }
    }
}
```

```

    gets(y);
    n = strlen(y);

    while (m != n)
    {
        printf(" \n\n");
        printf(" Panjang Untai Harus Sama !!! \n\n");
        printf(" Untai y : "); gets(y);
        n = strlen(y);
    }
}

HD = 0;

for (i = 0; i <= n-1; i++)
{
    if (x[i] != y[i])
    {
        HD = HD + 1;
    }
}

printf(" \n\n");
printf(" Maka, Jarak Hamming antara untai x dan y adalah %d.", HD);
printf(" \n\n");
printf(" Anda Ingin Coba Lagi ? \n\n");
printf(" \t Jika ingin merubah x, \t\t pilih 'x' \n\n");
printf(" \t Jika ingin merubah y, \t\t pilih 'y' \n\n");
printf(" \t Jika ingin merubah x dan y, \t pilih 's' \n\n");
printf(" \t Jika ingin keluar, \t\t pilih 'e' \n\n");
printf(" Pilihan Anda : ");
gets(opsi);
tolower(opsi[0]);

while (strlen(opsi) != 1 || (opsi[0] != 'x' && opsi[0] != 'y' &&
opsi[0] != 's' && opsi[0] != 'e'))
{
    printf(" Input Salah. Ulangi : ");
    gets(opsi);
    tolower(opsi[0]);
}

clrscr();
}
}

```

### Lampiran 3. Listing program interaktif pencocokan untai eksak dengan menggunakan ukuran jarak hamming

```

#include <dos.h>          //sleep();
#include <stdio.h>       //gets();
#include <conio.h>       //clrscr(); gotoxy();
#include <stdlib.h>
#include <string.h>     //strlen();
#include <ctype.h>     //tolower();

char x[500], y[500], x2[500], y2[500], opsi[500];
int cocok[500], posisi[500];
int m, n, i, k, v, w, a, b, c, j, HD, HD_Min;

void brute_force_HD(void);

```

```

void hasil(void);
void tampilan(void);

main()
{
    clrscr();
    opsi[0] = 's';
    while (opsi[0] == 's' || opsi[0] == 'x' || opsi[0] == 'y')
    {
        clrscr();
        printf(" \n");
        printf(" \t      Program Pencocokan Untai Eksak \n");
        printf(" \t dengan Menggunakan Ukuran Jarak Hamming \n");
        printf(" \t      antara untaian x dan untaian y. ");
        printf(" \n\n");

        if (opsi[0] == 's')
        {
            printf(" Masukkan untaian x \t : ");
            gets(x);
            m = strlen(x);
            printf(" Masukkan untaian y \t : ");
            gets(y);
            n = strlen(y);

            while (n < m)
            {
                printf(" \n\n");
                printf(" Panjang Untaian y Harus Lebih Dari atau Sama Dengan  

                Panjang Untaian x !!! \n\n");
                printf(" Untaian y : "); gets(y);
                n = strlen(y);
            }
        }

        if (opsi[0] == 'x')
        {
            printf(" Masukkan untaian x \t : \n");
            printf(" Masukkan untaian y \t : %s ", y);
            gotoxy(28,6); gets(x);
            m = strlen(x);
            printf(" \n");

            while (m > n)
            {
                printf(" \n\n");
                printf(" Panjang Untaian x Harus Kurang Dari atau Sama Dengan  

                Panjang Untaian y !!! \n\n");
                printf(" Untaian x : "); gets(x);
                m = strlen(x);
            }
        }

        if (opsi[0] == 'y')
        {
            printf(" Masukkan untaian x \t : %s ", x);
            printf(" \n");
            printf(" Masukkan untaian y \t : ");
            gets(y);
            n = strlen(y);

            while (n < m)
            {
                printf(" \n\n");
                printf(" Panjang Untaian y Harus Lebih Dari atau Sama Dengan

```

```

        Panjang Untai x !!! \n\n");
        printf(" Untai y : "); gets(y);
        n = strlen(y);
    }
}

strcpy(x2,x);
strcpy(y2,y);
strlwr(x2);
strlwr(y2);
v = 0;
w = 0;

/* 1. Jika |x| = |y| */

if (m == n)
{
    printf(" \n");
    printf(" |x| = %d = |y| ", n);
    posisi[1] = 1;
    /* 1.a. Jika |x| = |y| = 0 */

    if (m == 0)
    {
        printf(" \n\n", n);
        printf(" Kedua untai merupakan untai hampa, maka Jarak Hamming
        = 0. ");
        getchar();
    }

    /* 1.b. Jika |x| = |y| != 0 */

    else
    {
        k = 1;
        brute_force_HD();
        HD_Min = HD;
        printf(" \n\n", n);
        printf(" Didapat Jarak Hamming Minimum = %d. ", HD_Min);
        printf(" \n\n");
        printf(" Sehingga untai x");

        /* 1.b.1. Jika x = y */

        if (HD == 0)
        {
            printf(" sama");
        }

        /* 1.b.2. Jika x != y */

        else
        {
            printf(" tidak sama");
        }

        printf(" dengan untai y. ");
        getchar();
    }
}

/* 2. Jika |x| != |y| */

else
{

```



```

printf(" \n");
printf(" panjang x = m = %d\n", m);
printf(" panjang y = n = %d", n);

for (i = 0; i <= n-m-1; i++)
{
    x2[m+i] = '^';
}

/* 2.a. Jika |x| = 0 */

if (m == 0)
{
    printf(" \n\n");
    printf(" Karena panjang untai x adalah 0, maka untai x akan
        diisi \n");
    printf(" dengan untai hampa sebanyak %d. \n\n", m);
    printf(" Sehingga didapatkan posisinya sebagai berikut: ");
    getchar();
    printf(" \n\n");
    printf(" \t y = %s", y2);
    printf(" \n");
    printf(" \t x = %s", x2);
    posisi[1] = 1;
    printf(" \n\n");
    printf(" \t HD = n - m = %d \t --> HD_Min = %d ", n, n);
    printf(" \n\n");
    printf(" Didapat Jarak Hamming Minimum = %d. \n", n);
    printf(" Dari hanya satu kemungkinan posisi karena untai x
        merupakan untai hampa. ");
}

/* 2.b. Jika |x| != 0 dan |y| != 0 */

else
{
    printf(" \n\n");
    printf(" Karena panjang untai x dan untai y tidak sama, maka
        harus dilakukan \n");
    printf(" penambahan untai hampa pada untai x. \n\n");
    printf(" Sehingga didapatkan kemungkinan-kemungkinan posisinya
        sebagai berikut: ");
    getchar();

    HD_Min = n;
    for (k = 1; k <= n-m+1; k++)
    {
        printf(" \n\n");
        printf(" %d. Kemungkinan posisi ke-%d ", k, k);
        printf(" \n\n");
        printf(" \t y = %s", y2);
        printf(" \n");
        printf(" \t x = %s", x2);
        brute_force_HD();

        if (HD == n-m)
        {
            HD_Min = HD;
            v++;
            cocok[v] = k;
            printf(" \n\n");
            printf(" \t HD = n - m = %d \t --> HD_Min = %d ", HD,
                HD_Min);
        }
    }
}

```

```

else
{
    if (HD == HD_Min)
    {
        w++;
        posisi[w] = k;
    }
    if (HD < HD_Min)
    {
        HD_Min = HD;
        w = 1;
        posisi[w] = k;
    }

    printf(" \n\n");
    printf(" \t HD = %d \t --> HD_Min = %d ", HD, HD_Min);
}

for (i = m+k-1; i >= k; i--)
{
    x2[i] = x2[i-1];
}

x2[k-1] = '^';
getchar();
}

printf(" \n\n");
printf(" Didapat Jarak Hamming Minimum = %d. \n", HD_Min);
printf(" Dari kemungkinan posisi yang ");

/* 2.b.1. Jika x subuntai dari y */

if (HD_Min == n-m)
{
    if (v == 1)
    {
        printf("ke-%d. ", cocok[v]);
    }
    else
    {
        for (k = 1; k <= v-1; k++)
        {
            printf("ke-%d, ", cocok[k]);
        }

        printf("dan ke-%d. ", cocok[v]);
    }

    printf(" \n\n");
    printf(" Sehingga untai x ada di dalam untai y pada posisi
");

    if (v == 1)
    {
        printf("ke-%d. ", cocok[v]);
    }
    else
    {
        for (k = 1; k <= v-1; k++)
        {
            printf("ke-%d, ", cocok[k]);
        }
    }
}

```

```

        printf("dan ke-%d. ", cocok[v]);
    }
}

/* 2.b.2. Jika x bukan subuntai dari y */

else
{
    if (w == 1)
    {
        printf("ke-%d. ", posisi[w]);
    }

    else
    {
        for (k = 1; k <= w-1; k++)
        {
            printf("ke-%d, ", posisi[k]);
        }

        printf("dan ke-%d. ", posisi[w]);
    }

    printf(" \n\n");
    printf(" Akan tetapi, untai x tidak ada di dalam untai y");
}
}

/* 3. Tampilan jika |y| != 0 */

if (n != 0)
{
    printf(" \n\n");
    printf(" \t\tTekan ENTER untuk melihat hasilnya");
    getchar();

    /* 3.a. Tampilan jika |x| = 0 */

    if (m == 0)
    {
        hasil();
        gotoxy(5,31); printf(" Hanya terdapat satu kemungkinan posisi
        karena untai x merupakan untai hampa. ");
        gotoxy(5,32); printf(" Sehingga didapat satu-satunya Jarak
        Hamming Minimum = %d. \n", n);
        gotoxy(5,35); printf(" Tekan 'ENTER' untuk melanjutkan. ");
        getchar();
    }

    /* 3.b. Tampilan jika |x| != 0 */

    else
    {
        char pilihan = 'p';
        while (pilihan == 'p' || pilihan == 'P')
        {
            hasil();

            if (m == n)
            {
                gotoxy(5,31); printf(" Didapat Jarak Hamming Minimum =
                %d. ", HD_Min);
                gotoxy(5,32); printf(" Sehingga untai x");
            }
        }
    }
}
}

```

```

/* 3.b.1. Tampilan jika x = y */
    if (HD == 0)
    {
        printf(" sama");
    }

/* 3.b.2. Tampilan jika x != y */

    else
    {
        printf(" tidak sama");
    }

    printf(" dengan untai y. ");
}
else
{
    gotoxy(5,31); printf(" Didapat Jarak Hamming Minimum =
%d. \n", HD_Min);
    gotoxy(5,32); printf(" Dari kemungkinan posisi yang ");

/* 3.b.3. Tampilan jika x subuntai dari y */
    if (HD_Min == n-m)
    {
        if (v == 1)
        {
            printf("ke-%d. ", cocok[v]);
        }
        else
        {
            for (k = 1; k <= v-1; k++)
            {
                printf("ke-%d, ", cocok[k]);
            }
            printf("dan ke-%d. ", cocok[v]);
        }

        gotoxy(5,33); printf(" Sehingga untai x ada di dalam
untai y pada posisi ");

        if (v == 1)
        {
            printf("ke-%d. ", cocok[v]);
        }
        else
        {
            for (k = 1; k <= v-1; k++)
            {
                printf("ke-%d, ", cocok[k]);
            }
            printf("dan ke-%d. ", cocok[v]);
        }
    }

/* 3.b.4. Tampilan jika x bukan subuntai dari y */

    else

```

```

        {
            if (w == 1)
            {
                printf("ke-%d. ", posisi[w]);
            }

            else
            {
                for (k = 1; k <= w-1; k++)
                {
                    printf("ke-%d, ", posisi[k]);
                }

                printf("dan ke-%d. ", posisi[w]);
            }

            gotoxy(5,33); printf(" Akan tetapi, untai x tidak ada
            di dalam untai y. ");
        }
    }

    gotoxy(5,35); printf(" Tekan 'p' untuk mengulangi atau
    lainnya untuk melanjutkan. ");
    pilihan = getche();
}
}

k = 0;

while (x2[k] != NULL || y2[k] != NULL || cocok[k+1] != NULL ||
posisi[k+1] != NULL)
{
    x2[k] = NULL;
    y2[k] = NULL;
    cocok[k+1] = NULL;
    posisi[k+1] = NULL;
    k++;
}

clrscr();
printf(" \n\n");
printf(" Anda Ingin Coba Lagi ? \n\n");
printf(" \t Jika ingin merubah x, \t\t pilih 'x' \n\n");
printf(" \t Jika ingin merubah y, \t\t pilih 'y' \n\n");
printf(" \t Jika ingin merubah x dan y, \t pilih 's' \n\n");
printf(" \t Jika ingin keluar, \t\t pilih 'e' \n\n");
printf(" Pilihan Anda : ");
gets(opsi);
tolower(opsi[0]);

while (strlen(opsi) != 1 || (opsi[0] != 'x' && opsi[0] != 'y' &&
opsi[0] != 's' && opsi[0] != 'e'))
{
    printf(" Input Salah. Ulangi : ");
    gets(opsi);
    tolower(opsi[0]);
}
}

void brute_force_HD()
{
    HD = n-m;
}

```

```

for (i = k-1; i < m+k-1; i++)
{
    if (x2[i] == y2[i])
    {
        HD = HD+0;
    }

    else
    {
        HD = HD+1;
    }
}

}

void hasil()
{
    k = 0;
    c = 1;

    while (x2[k] != NULL || y2[k] != NULL || posisi[k+1] != NULL)
    {
        x2[k] = NULL;
        y2[k] = NULL;

        if (cocok[1] != NULL)
        {
            posisi[k+1] = NULL;
        }

        k++;
    }

    clrscr();

    for (k = 1; k <= n+7; k++)
    {
        gotoxy(30+k,11); printf("=");
        gotoxy(30+k,16); printf("=");

        if (k <= 6)
        {
            gotoxy(30,10+k); printf("||");
            gotoxy(38+n,10+k); printf("||");
        }
    }

    gotoxy(33,13); printf("y = %s", y);
    strcpy(x2,x);

    for (i = 0; i <= n-m-1; i++)
    {
        x2[m+i] = '^';
    }

    if (m == 0 || n == 0)
    {
        k = 1;
        tampilan();
        sleep(1);
    }

    else
    {
        for (k = 1; k <= n-m+1; k++)

```

```

    {
        tampilan();

        for (i = m+k-1; i >= k; i--)
        {
            x2[i] = x2[i-1];
        }

        x2[k-1] = '^';
        sleep(1);
    }
}

gotoxy(35,35);
}

void tampilan()
{
    gotoxy(33,14); printf("x = %s", x2);
    gotoxy(1,1);

    if (k == cocok[c])
    {
        c = c+1;

        for (i = 1; i <= 3; i++)
        {
            sleep(1);

            for (j = 1; j <= 6+2*i; j++)
            {
                gotoxy(30-2*i,10-i+j); printf("||");
                gotoxy(38+n+2*i,10-i+j); printf("||");
                gotoxy(1,1);
            }
        }

        sleep(1);

        for (i = 1; i <= 3; i++)
        {
            for (j = 1; j <= 6+2*i; j++)
            {
                gotoxy(30-2*i,10-i+j); printf(" ");
                gotoxy(38+n+2*i,10-i+j); printf(" ");
                gotoxy(1,1);
            }
        }
    }
}
}

```