



Penerapan Algoritma Apostolico Crochemore Pada Perancangan Aplikasi Laundry Mobile Berbasis Android

Feri Berkat

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

E-mail : feriberkat06@gmail.com

Abstrak—Jasa *laundry* merupakan salah satu bisnis jasa yang bergerak di bidang cuci pakaian yang *notabene* diperuntukkan bagi kalangan menengah ke atas, yang mana mereka tidak punya waktu untuk mencuci pakaiannya sendiri sehingga mempercayakan pencucian pakaiannya ke jasa *laundry* yang ada di sekitarnya. Hadirnya jasa *laundry* pada kehidupan masyarakat umum memberikan dampak positif dengan mengurangi aktivitas di luar aktivitas yang begitu banyak baik aktivitas kuliah bahkan aktivitas pekerjaan. Pengguna jasa *laundry* biasa ditemukan adalah mahasiswa – mahasiswi hingga pekerja kantoran, dan lain - lain. Jasa laundry pada umumnya menawarkan jasa pencucian kaos, kemeja celana, selimut, *bad cover*, dan lain-lain. Masing – masing penyedia jasa *laundry* pada umumnya memberikan tarif yang berbeda dimulai dengan harga 5 ribu rupiah per kilo gram hingga 30 ribu rupiah per kilo gram tergantung dengan kualitas yang ditawarkan oleh penyedia jasa *laundry* tersebut. Adanya selisih harga yang ditawarkan oleh penyedia jasa *laundry* tentunya konsumen ingin mencari penyedia jasa *laundry* yang sesuai dengan kondisi keuangan maupun kualitas yang diinginkan, hal tersebut membuat calon pengguna jasa *laundry* perlu membanding harga dan kualitas yang disediakan jasa *laundry*. Untuk menemukan informasi yang mengadung tentang harga dan kualitas yang ditawarkan jasa laundry sulit untuk dijangkau dengan mudah oleh calon pengguna jasa *laundry* yang ada dikarenakan jarak setiap penyedia jasa *laundry* yang berjauhan.

Kata Kunci: Laundry; Android; Algoritma Apostolico-Crochemore

Abstract—Laundry service is one of the service businesses engaged in washing clothes which incidentally is intended for the upper middle class, where they do not have time to wash their own clothes so they entrust their laundry to laundry services around them. The presence of laundry services in the lives of the general public has a positive impact by reducing activities outside of the many activities, both college activities and even work activities. Users of laundry services are usually found, from students to office workers, and others. Laundry services generally offer services for washing t-shirts, shirts, pants, blankets, bad covers, and others. Each laundry service provider generally provides different rates starting with a price of 5 thousand rupiah per kilo gram to 30 thousand rupiah per kilo gram depending on the quality offered by the laundry service provider. The existence of a price difference offered by laundry service providers, of course, consumers want to find a laundry service provider that is in accordance with financial conditions and the desired quality, this makes prospective laundry service users need to compare the price and quality provided by laundry services. To find information that contains information about the price and quality offered by laundry services, it is difficult to be easily reached by prospective users of existing laundry services due to the distance between each laundry service provider.

Keywords: Laundry; Android; Apostolico-Crochemore Algorithm

1. PENDAHULUAN

Jasa *laundry* merupakan salah satu bisnis jasa yang bergerak di bidang cuci pakaian yang *notabene* diperuntukkan bagi kalangan menengah ke atas, yang mana mereka tidak punya waktu untuk mencuci pakaiannya sendiri sehingga mempercayakan pencucian pakaiannya ke jasa *laundry* yang ada di sekitarnya. Hadirnya jasa *laundry* pada kehidupan masyarakat umum memberikan dampak positif dengan mengurangi aktivitas di luar aktivitas yang begitu banyak baik aktivitas kuliah bahkan aktivitas pekerjaan.

Pengguna jasa *laundry* biasa ditemukan adalah mahasiswa – mahasiswi hingga pekerja kantoran, dan lain - lain. Jasa laundry pada umumnya menawarkan jasa pencucian kaos, kemeja celana, selimut, *bad cover*, dan lain-lain. Masing – masing penyedia jasa *laundry* pada umumnya memberikan tarif yang berbeda dimulai dengan harga 5 ribu rupiah per kilo gram hingga 30 ribu rupiah per kilo gram tergantung dengan kualitas yang ditawarkan oleh penyedia jasa *laundry* tersebut. Adanya selisih harga yang ditawarkan oleh penyedia jasa *laundry* tentunya konsumen ingin mencari penyedia jasa *laundry* yang sesuai dengan kondisi keuangan maupun kualitas yang diinginkan, hal tersebut membuat calon pengguna jasa *laundry* perlu membanding harga dan kualitas yang disediakan jasa *laundry*. Untuk menemukan informasi yang mengadung tentang harga dan kualitas yang ditawarkan jasa laundry sulit untuk dijangkau dengan mudah oleh calon pengguna jasa *laundry* yang ada dikarenakan jarak setiap penyedia jasa *laundry* yang berjauhan.

Agar dapat menyelesaikan masalah tersebut dalam penelitian ini penulis melakukan sebuah penelitian terhadap penyedia jasa laundry di beberapa tempat untuk mengutip informasi tentang harga dan kualitas beserta informasi lainnya yang ditawarkan oleh penyedia jasa *laundry*. Kemudian pada tahap yang selanjutnya penulis melakukan perancangan aplikasi *laundry mobile* berbasis *android* yang dirancang menggunakan *tools editor eclipse juno* dan *database SQLite*. Dikarenakan pencarian data informasi jasa *laundry* pada aplikasi *laundry mobile* berbasis *android* yang dirancang pada penelitian ini adalah berupa *string* atau karakter maka untuk mempermudah *user* aplikasi tersebut untuk memperoleh pencapaian hasil pencarian tersebut diperlukan penerapan algoritma *string matching*.

Penerapan algoritma *string matching* pada perancangan aplikasi *laundry mobile* berbasis *android* tersebut dinilai dapat mengurangi waktu yang dibutuhkan dalam proses pencarian informasi jasa *laundry* yang diharapkan. Algoritma *string matching* yang diterapkan oleh penulis dalam penelitian ini untuk dipergunakan pada fitur pencarian di dalam aplikasi *laundry mobile* berbasis *android* yang dirancang adalah algoritma *apostolico-chocrome*. Algoritma *apostolico-crochemore* merupakan salah satu jenis algoritma pencarian *string* dari banyak jenis algoritma pencarian *string*.



Algoritma *apostolico-crochemore* adalah algoritma sederhana yang melakukan perbandingan karakter pada teks sebanyak $3n/2$ pada kasus terburuknya. algoritma *apostolico-crochemore* terdiri dari dua *fase*, yaitu *fase* proses awal (*preprocessing*) dan *fase* pencarian *string*. Pada *fase* proses awal dilakukan fungsi pinggiran untuk menentukan jumlah langkah pergeseran *pattern* terbesar dengan menggunakan perbandingan sebelum pencarian *string*. pada *fase* pencarian *string* dilakukan perbandingan *pattern* pada teks.

Pada penelitian ini peneliti tertarik melakukan penelitian dengan menerapkan algoritma *apostolico-crochemore*. Peneliti berharap penelitian ini dapat memberikan hasil yang lebih baik terhadap pencarian informasi jasa *laundry*.

2. METODOLOGI PENELITIAN

2.1 String Matching

String Matching merupakan salah satu algoritma yang digunakan untuk mempercepat proses pencarian kata yang diinginkan. *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic* atau *statistical matching*. Algoritma *string matching* telah sering digunakan sebelumnya seperti contoh pada proses pencocokan *string* berdasarkan persamaan teks data yaitu *Brute Force*. Dalam hal ini, dipilih algoritma *brute force* karena algoritma ini dapat digunakan untuk melakukan pencarian *string* atau teks. Algoritma *brute force* adalah algoritma untuk mencocokkan *pattern* dengan semua teks antara 0 dan $n-m$ untuk menemukan keberadaan *pattern* dalam teks. Berdasarkan arah pencariannya, algoritma ini diklasifikasikan sebagai algoritma yang membaca *string* dari kiri ke kanan [2].

String matching adalah proses pencarian semua kemunculan *query* yang selanjutnya disebut *pattern* ke dalam *string* yang lebih panjang. *String matching* dirumuskan dengan :

$$x = x [0...m-1] \quad (1)$$

$$y = y [0...n-1] \quad (2)$$

dimana : $x = \text{pattern}$ $m = \text{panjang pattern}$ $y = \text{Teks}$ $n = \text{Panjang teks}$ [3].

2.1.1 Algoritma String Matching

Algoritma *String Matching* adalah algoritma yang dapat digunakan untuk penyelesaian masalah pencocokan *string*, secara garis besar algoritma *string matching* dibedakan menjadi 3 bagian berdasarkan arah pencocokan *string*, yaitu :

1. Arah pembacaan dari kiri ke kanan.

Algoritma yang termasuk kategori ini adalah *Brute Force*, *Levenshtein Distance*, *Morris* dan *Pratt* (yang kemudian dikembangkan oleh Knuth, Morris, dan Pratt).

2. Arah pembacaan dari kanan ke kiri.

Algoritma yang termasuk kategori ini adalah Boyer dan Moore yang kemudian dikembangkan menjadi algoritma *turbo Boyer-Moore*, *tuned Boyer-Moore*, dan *Zhu-Takaoka*.

3. Arah pencarian yang ditentukan program. Algoritma yang termasuk kategori ini adalah algoritma *Colussi*, *Crochemore-Perrin* [2].

2.2 Algoritma Apostolico-Crochemore

Algoritma *Apostolico-Crochemore* adalah salah satu jenis algoritma pencarian *string* dari banyak jenis algoritma pencarian *string*. Algoritma *Apostolico-Crochemore* adalah algoritma sederhana yang melakukan perbandingan karakter pada teks sebanyak $3n/2$ pada kasus terburuknya. Algoritma *Apostolico-Crochemore* terdiri dari dua *fase*, yaitu *fase* proses awal (*preprocessing*) dan *fase* pencarian *string*. Pada *fase* proses awal dilakukan fungsi pinggiran untuk menentukan jumlah langkah pergeseran *pattern* terbesar dengan menggunakan perbandingan sebelum pencarian *string*. Pada *fase* pencarian *string* dilakukan perbandingan *pattern* pada teks. Algoritma *Apostolico-Crochemore* mirip dengan algoritma *Knuth-Morris-Path (KMP)*. Fungsi pinggiran (*border function*) yang digunakan mirip seperti fungsi pinggiran pada algoritma *KMP* dan proses pencariannya sama-sama dimulai dari kiri ke kanan. Akan tetapi, tahapan pencarian (proses urutan perbandingan) berbeda dengan algoritma *KMP* [1].

2.2.1 Preprocessing Apostolico-Crochemore

Preprocessing (*Fase* proses awal) terhadap *pattern* x adalah dengan menghitung fungsi pinggiran yang mengindikasikan pergeseran x terbesar yang mungkin dengan menggunakan perbandingan yang dibentuk sebelum *fase* pencarian *string*. Dengan adanya fungsi pinggiran ini dapat dicegah pergeseran yang tidak berguna, seperti halnya pada algoritma *brute force*. Fungsi pinggiran hanya bergantung pada karakter-karakter di dalam *pattern*, dan bukan pada karakter-karakter di dalam teks. Fungsi pinggiran pada algoritma *Apostolico-Crochemore* adalah *kmpNext* [1].

2.2.2 Fase Pencocokan String

Proses pencarian (pencocokan) *pattern* dengan teks pada algoritma *Apostolico-Crochemore* adalah sebagai berikut. Misalnya x adalah *string* dari *pattern* yang akan dicari, y adalah teks yang akan dicocokkan dengan *pattern*, m adalah panjang x , dan n adalah panjang y .

$I = 0$ jika x adalah karakter tunggal yang dipangkatkan ($x = c^m$ dengan c di dalam S) dan l adalah posisi karakter pertama dari x yang berbeda dari $x[0]$ ($x = a^l b$ untuk a, b di dalam S , u di dalam S^* , dan $a^l b$). Setiap perbandingan dilakukan dengan posisi yang berpola dengan urutan sebagai berikut : $l, l+1, \dots, m-2, m-1, 0, 1, \dots, l-1$.



Selama fase pencarian, yang harus diperhatikan (i, j, k) di mana :

- jendela karakter diposisikan pada teks $y[j .. j+m-1]$
- $0 \leq k \leq l$ dan $x[0 .. k-1] = y[j .. j+k-1]$
- $1 \leq i < m$ dan $x[1 .. i-1] = y[j+1 .. i+j-1]$

Inisialisasi awal dari (i, j, k) adalah $(1, 0, 0)$.

Perhitungan untuk menentukan (i, j, k) berikutnya mempertimbangkan tiga kasus yang bergantung pada nilai i . Ketiga kasus tersebut sebagai berikut :

1. $i = 1$

Jika $x[i] = y[i+j]$ maka (i, j, k) berikutnya adalah $(i+1, j, k)$.

Jika $x[i] \neq y[i+j]$ maka (i, j, k) berikutnya adalah $(1, j+1, \max\{0, k-1\})$.

2. $1 < i < m$ Jika $x[i] = y[i+j]$ maka (i, j, k) berikutnya adalah $(i+1, j, k)$.

Jika $x[i] \neq y[i+j]$ maka ada dua kasus yang muncul yang bergantung pada nilai $kmpNext[i]$:

a. $kmpNext[i] \leq l$ maka (i, j, k) berikutnya $(1, i+j-kmpNext[i], \max\{0, kmpNext[i]\})$.

b. $kmpNext[i] > l$ maka (i, j, k) berikutnya $(kmpNext[i], i+j-kmpNext[i], kmpNext[i] - 1)$.

3. $i = m$

Jika $k < l$ dan $x[k] = y[j+k]$ maka (i, j, k) berikutnya adalah $(i, j, k+1)$.

4. Sebaliknya salah satu dari $k < l$ dan $x[k] \neq y[j+k]$, atau $k = l$ (jika $k = l$ kemunculan x diberitahukan) pada kedua kasus tersebut perhitungan (i, j, k) berikutnya sama seperti perhitungan pada kasus $1 < i < m$ [1].

2.3 Aplikasi

Aplikasi adalah Program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian Aplikasi Secara Umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user* [11].

2.4 Laundry

Laundry atau Binatu adalah fasilitas di mana pakaian dicuci dan dikeringkan. Binatu biasa menggunakan mesin cuci otomatis yang disebut sebagai laundromat atau merk dagang umum. Dengan perkembangan model busana dan aneka perniknya, misalnya jenis-jenis pakaian yang memiliki banyak motif yang tidak mudah untuk mencuci sendiri dan berbagai jenis perlengkapan lainnya seperti pelayanan cuci handuk, cuci sepatu, cuci selimut, cuci *badcover*, cuci boneka, cuci tas dan lain-lain. Dengan perkembangan kain serta modelnya maka pencucian dan perawatannya menjadi sulit, disinilah [jasa laundry](#) mulai diperlukan dan saat ini perkembangan usaha laundry semakin maju.

Jasa *laundry* tidak hanya sekedar untuk tempat mencuci pakaian melainkan sebagai tempat perawatan pakaian agar menjadi lebih bersih dan awet, dan faktor serba instan dan praktis menjadi *trend* bagi masyarakat saat ini. Begitu pentingnya keberadaan *laundry* untuk saat ini maka membuat bisnis ini begitu berkembang pesat. Kalau jaman dulu jasa laundry masih dikelola oleh kebanyakan di hotel hotel tapi saat ini baik kalangan bermodal atau masyarakat umum mulai memanfaatkan jasa laundry [4].

2.5 Alat Bantu Perancangan Sistem

2.5.1 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan *requirement*, membuat analisis dan desain, menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung, yang hanya berfungsi untuk melakukan pemodelan [5].

3. HASIL DAN PEMBAHASAN

3.1 Penerapan Algoritma

Apostolico-crochemore pada penelitian ini bertujuan untuk menyelesaikan masalah dalam pencarian informasi jasa *laundry*. Sebelum melakukan proses pencarian terlebih dahulu dilakukan *fase preprocessing* dengan menghitung jumlah pergeseran *pattern* terbesar yang mungkin terjadi dengan menggunakan perbandingan karakter *pattern* dengan karakter *text*. *Fase preprocessing* pada algoritma *apostolico-crochemore* menggunakan tabel pergeseran *kmpNext*. Untuk mendapatkan nilai pergeseran, *kmpNext[i]* untuk indeks 0 diberi nilai -1 sebagai inisialisasi. Selanjutnya, bandingkan karakter pada indeks 0 dengan karakter pada indeks 1. Jika karakternya sama, maka *kmpNext[1]* bernilai -1 Namun apabila karakternya berbeda pada saat melakukan proses pencocokan *string* maka *kmpNext[1]* bernilai 0. Adapun contoh penyelesaian masalah pencarian informasi jasa *laundry* menerapkan algoritma *apostolico-crochemore* dapat dilihat pada contoh kasus berikut :

Text (y) : Kaos Oblong



Pattern (x) : Oblong

Tabel 1. *kmpNext*

<i>I</i>	0	1	2	3	4	5
<i>x[i]</i>	O	B	L	O	N	G
<i>kmpNext[i]</i>	-1	0	0	1	0	0

$1 = 1$

Tahap pencocokan karakter menggunakan algoritma *apostolic-crochemore* dapat dilihat pada table berikut

Tabel 2. Tahap 1

K	A	O	S	O	B	L	O	N	G
	1								
O	B	L	O	N	G				

Berdasarkan pencocokan karakter pada tahap 1 tidak ditemukan kecocokan karakter terhadap “B” dengan “A”, maka *Pattern* digeser sebanyak : $1 - kmpNext[1] = 1 - 0 = 1$ langkah

Tabel 3. Tahap 2

K	A	O	S	O	B	L	O	N	G
		1							
O	B	L	O	N	G				

Berdasarkan pencocokan karakter pada tahap 2 tidak ditemukan kecocokan karakter terhadap “B” dengan “O”, maka *Pattern* digeser sebanyak : $1 - kmpNext[1] = 1 - 0 = 1$ langkah

Tabel 4. Tahap 3

K	A	O	S	O	B	L	O	N	G
			1						
O	B	L	O	N	G				

Berdasarkan pencocokan karakter pada tahap 3 tidak ditemukan kecocokan karakter terhadap “B” dengan “S”, maka *Pattern* digeser sebanyak : $1 - kmpNext[1] = 1 - 0 = 1$ langkah

Tabel 5. Tahap 4

K	A	O	S	O	B	L	O	N	G
				1					
O	B	L	O	N	G				

Berdasarkan pencocokan karakter pada tahap 4 tidak ditemukan kecocokan karakter terhadap “B” dengan “”, maka *Pattern* digeser sebanyak : $1 - kmpNext[1] = 1 - 0 = 1$ langkah

Tabel 6. Tahap 5

K	A	O	S	O	B	L	O	N	G
				1					
O	B	L	O	N	G				

Berdasarkan pencocokan karakter pada tahap 5 tidak ditemukan kecocokan karakter terhadap “B” dengan “O”, maka *Pattern* digeser sebanyak : $1 - kmpNext[1] = 1 - 0 = 1$ langkah

Tabel 7. Tahap 6

K	A	O	S	O	B	L	O	N	G
				6	1	2	3	4	5
				O	B	L	O	N	G

Berdasarkan pencocokan karakter pada tahap 6 ditemukan kecocokan seluruh karakter (x) dengan karakter (y), maka pergeseran *pattern* dihentikan

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan penulis maka dapat ditarik kesimpulan bahwa pencarian informasi jasa *loundry* pada *Express Londry* dalam penelitian ini menggunakan kata kunci nama *laundry*. Implementasi algoritma *apostolico-crochemore* dapat mempercepat proses pencocokan *string* untuk pencarian informasi jasa *loundry*.



REFERENCES

- [1] Sindy Gita Ratri, "Penggunaan Algoritma Apostolico-Crochemore Pada Proses Pencarian String di Dalam Teks," 2007.
- [2] Mesran, "Implementasi Algoritma Brute Force Dalam Pencarian Data Katalog Buku Perpustakaan," 2014.
- [3] Fince Tinus Waruwu, "Implementasi Algoritma Knuth Morris Pratt Pada Aplikasi Kamus Istilah Latin Flora dan Fauna Berbasis Android," 2014.
- [4] <http://www.pewangilaundry.co.id/pengertian-usaha-laundry.html>
- [5] Widodo, *Menggunakan UML untuk Memodelkan Analisis & Desain Sistem Berorientasi*. Bandung: Informatika, 2011.
- [6] R. A. S and M. Shalahuddin, *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Informatika, 2014.
- [7] Nasruddin Safaat H, *Pemograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android.*, 2015.
- [8] M. Hilmi Masruri dan Java Creativity, *Buku Pintar Android.*, 2015.
- [9] Alfa Satyaputra dan Eva Maulina Aritonang, *Java For Beginners with Eclipse 4.2 Juno*. Jakarta: PT Elex Media Komputindo, 2012.
- [10] Uli fitrianti & mutammimul ula, Implementasi Algoritma Levenshtein Distance dan Algoritma Knuth Morris Pratt Pada Aplikasi Asmaul Husna Berbasis Android, *Jurnal Sistem Informasi*, ISSN : 259 8-599X, 2017.