



PENERAPAN ALGORITMA KNUTH MORRIS PRATT (KMP) PADA PENCARIAN DATA DI SQL LIKE OPERATORS

Fenina Adline Twince Tobing¹, Alex Chandra^{2*}, Rena Nainggolan³

¹Program Studi Informatika, Fakultas Teknik dan Informatika Universitas Multimedia Nusantara,
Tangerang, Banten, Indonesia

²Program Studi Manajemen Informatika, AMIK Widya Loka, Medan, Sumatera Utara, Indonesia

³Program Studi Komputerisasi Akuntansi, Universitas Methodist Indonesia,
Medan, Sumatera Utara, Indonesia

Email : ¹fenina.tobing@umn.ac.id, ²alexsiburian03@gmail.com

Received: January 30, 2022. **Revised:** February 28, 2022. **Accepted:** March 25, 2022.

DOI : <https://doi.org/10.54593/awl.v3i1.82>

Abstrak

Penelitian ini membahas bagaimana database tracking system dapat menemukan data yang diinginkan berdasarkan SQL LIKE Operator yang telah ditentukan dengan menggunakan Knuth Morris Pratt Algorithm (KMP) dalam pencarian data pada SQL. Structured Query Language (SQL) adalah sekumpulan perintah khusus yang digunakan untuk mengakses data dalam database relasional. Untuk mencari data SQL pada query operator LIKE yang telah ditentukan dapat dilakukan melalui pencocokan string pada data yang ada untuk mendapatkan hasil. Pencocokan String adalah algoritma untuk mencari semua kemunculan string pendek yang disebut pola dalam string yang lebih panjang yang disebut teks. Knuth Morris Pratt Algorithm (KMP) adalah pencocokan string dalam teks dari kiri ke kanan dengan mencocokkan karakter per pola karakter dengan karakter dalam teks yang sesuai. Hasil penelitian menggunakan metode KMP pada query SQL berjalan dengan baik dalam melakukan pencarian data menggunakan Operator LIKE dan kemudahan dalam mengimplementasikan algoritma KMP dalam pencarian data pada SQL harus disesuaikan dengan *wildcard* pada operator LIKE.

Kata kunci—Pencocokan String, SQL, Knuth Morris Pratt (KMP).

Abstract

Application is an action taken, both individually and in groups to achieve the goals that have been formulated. This journal discusses how the database tracking system can find the desired data based on the SQL LIKE Operator that has been determined by using the Knuth Morris Pratt Algorithm (KMP) in searching data on the SQL. Structured Query Language (SQL) is a set of special commands that are used to access data in a relational database. To search for SQL data on the LIKE operator query that has been determined through matching a string on existing data to get the results. String Matching is an algorithm for searching for all occurrences of short strings called patterns in a longer string called text. Knuth Morris Pratt Algorithm (KMP) is matching strings in text from left to right by matching characters per character pattern with characters in the corresponding text. The results of the study using the KMP method on SQL queries went well in conducting data searches using the LIKE Operator and it was easy to implement the KMP algorithm in searching data in SQL.

Keywords— String Matching, SQL, Knuth Morris Pratt (KMP)



JURNAL WIDYA This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).



1 Pendahuluan (Introduction)

Pada zaman sekarang ini perusahaan-perusahaan besar seperti Google, Microsoft, Apple Inc, dll telah banyak menggunakan database, salah satunya menggunakan SQL sebagai tempat dalam proses pengumpulan dan pengolahan suatu kumpulan data. Dengan adanya database, data yang telah terkumpul dapat disimpan sehingga data tersebut akan tersusun secara ringkas dan rapi, serta dapat mengecek apakah ada data yang duplikat, tidak valid, atau data yang tidak penting. Basis data juga dapat diakses kapan saja dan di mana saja oleh siapa saja dengan syarat terhubung ke basis data melalui internet. Wikimedia Foundation menyatakan “Basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis” [1].

Basis data bisa dikatakan mirip dengan perpustakaan. Saat mencari buku di perpustakaan akan memakan waktu yang cukup lama untuk menemukannya, apalagi jika perpustakaannya sangat besar. Perbedaan dengan database terletak pada kecepatan pencarian data, dengan database pencarian akan dilakukan oleh proses komputer pada sistem.

Dalam query SQL yang merupakan bahasa untuk mengakses data dalam database relasional [2]. Dalam SQL terdapat query SQL yang menggunakan operator LIKE untuk menampilkan tabel berdasarkan pencarian karakter sederhana dengan cara mencocokkan pola pada LIKE dengan data yang ingin dibutuhkan oleh kolom tabel.

Karena pencarian karakter harus sesuai dengan pola dengan teks data, maka dibuatlah aplikasi untuk mencari data pada operator SQL LIKE dengan algoritma Knuth Morris Pratt (KMP). Melalui metode ini, data yang ingin dicari dapat ditemukan sesuai dengan pola LIKE yang telah ditentukan.

2 Tinjauan Literatur

A. String Matching

Algoritma String Matching (Pencocokan String) adalah algoritma untuk mencari semua kemunculan string pendek (pola) yang muncul dalam teks. Polanya adalah string dengan panjang karakter 'm' ($m < n$). Teks berupa string panjang yang panjangnya 'n' karakter [3].

B. Query SQL

Ariata.C menyatakan, "Query SQL adalah bahasa pemrograman yang berguna untuk memanipulasi data - menambah, menghapus, dan mengubah data." [4].

Query adalah sejenis kemampuan untuk menampilkan data dari database yang diambil dari tabel di database, dengan sintaks atau perintah yang akan ditulis sesuai dengan keinginan engineer dalam menampilkan data. Ada tiga perintah utama dalam operasi akses database, yaitu:

1) SELECT

Digunakan untuk memilih data dari database [5].

2) FROM

Digunakan untuk menentukan tabel untuk memilih atau menghapus data [6].

3) WHERE

Digunakan untuk mengekstrak hanya record yang memenuhi persyaratan tertentu [7].

Contoh penggunaan query untuk mengakses database di SQL adalah sebagai berikut.

```
SELECT LAST_NAME, JOB_ID, SALARY
FROM EMPLOYEES
WHERE SALARY > 2000 AND JOB_ID = '
ST_CLERK';
```





Pada query diatas akan diakses beberapa data EMPLOYEES yang memiliki gaji lebih dari 2000 dan memiliki JOB_ID 'ST_CLERK'.

Dalam WHERE, berbagai perintah relasional digunakan, seperti LIKE dan perintah operasi lainnya. Dengan perintah LIKE, semua string dalam database akan dicari berdasarkan teks/huruf yang dimasukkan. String yang akan dicari dapat digunakan dapat ditentukan pada dua *wildcard* yang sering digunakan bersama dengan operator LIKE:

- 1) '%' Tanda persen mewakili nol, satu, atau beberapa karakter.
- 2) '_' Garis bawah mewakili satu karakter yang biasa disebut sebagai karakter khusus [8].

Sintaks atau perintah sederhana dari query LIKE adalah sebagai berikut.

```
SELECT JOB_ID  
FROM EMPLOYEES  
WHERE JOB_ID LIKE '%CLERK%';
```

Pada query di atas akan diakses data JOB_ID yang namanya memiliki kata 'CLERK' di setiap teks [9].

C. Algoritma Knuth Morris Pratt

Algoritma KMP (Knuth-Morris-Pratt) merupakan salah satu algoritma yang dapat digunakan untuk mencari dimana suatu string (dalam hal ini disebut pattern) ditemukan pada string lain yang berukuran lebih besar [10].

Secara sistematis, langkah-langkah yang diambil oleh algoritma Knuth-Morris-Pratt saat mencocokkan string:

- 1) Algoritma Knuth Morris Pratt (KMP) mulai mencocokkan pola di awal teks.
- 2) Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per pola karakter dengan karakter dalam teks yang sejajar, hingga salah satu kondisi berikut terpenuhi:
 - a) Karakter dalam pola dan teks yang dibandingkan tidak cocok (mismatch).
 - b) Semua karakter dalam pola cocok. Kemudian algoritma akan memberitahukan penemuan pada posisi tersebut.
- 3) Algoritma kemudian menggeser pola berdasarkan tabel berikutnya, kemudian mengulangi langkah 'b' hingga pola berada di akhir teks.





Algoritma Pseudocode KMP dalam tahap pencarian oleh Wikipedia (2019).

```

procedure KMPSearch(
  input m, n: integer,
  input P: array[0..n-1] of char,
  input T: array[0..m-1] of char,
  output ketemu: array[0..m-1] of boolean
)

Deklarasi:
i, j, next: integer
kmpNext: array[0..n] of integer

Algoritme:
preKMP(n, P, kmpNext)
i:=0
while (i<= m-n) do
  j:=0
  while (j < n and T[i+j] = P[j]) do
    j:=j+1
  endwhile
  if(j >= n) then
    ketemu[i]:=true;
  endif
  next:= j - kmpNext[j]
  i:= i+next
endwhile

```

Algoritma ini hanya membutuhkan ruang $O(n)$ memori internal jika teks dibaca dari file eksternal. Semua nilai O tidak bergantung pada ukuran ruang alfabet [11].

Contoh penerapan algoritma KMP dalam pencarian string pada teks sesuai pola sebagai berikut.

Teks : ACTCTCCATGATTAGTCACTCTCCACTATCCTA

Pola : TAGTCACTC

Cara untuk mengatasi masalah tersebut harus terlebih dahulu mencari data awalan agar pergeseran string yang cocok dapat lebih terorganisir. Dengan membuat tabel yang terdiri dari kolom-kolom (Indexes, Patterns, Prefixes, Prefixes, Suffixes) untuk mencari nilai prefix dari pattern, dilakukan dengan cara mencocokkan prefix dan suffix pada kedua data tersebut.

INDEX	PATTERN	PREFIX	PREFIXES	SUFFIXES
1	T	0	-	-
2	TA	0	T	A
3	TAG	0	T, TA	AG, A
4	TAGT	1	T, TA, TAG	AGT, GT, T
5	TAGTC	0	T, TA, TAG, TAGT	AGTC, GTC, TC, C
6	TAGTCA	0	T, TA, TAG, TAGT, TAGTC	AGTCA, GTCA, TCA, CA, A
7	TAGTCAC	0	T, TA, TAG, TAGT, TAGTC, TAGTCA	AGTCAC, GTCAC, TCAC, CAC, AC, C
8	TAGTCACT	1	T, TA, TAG, TAGT, TAGTC, TAGTCA, TAGTCACT	AGTCACT, GTCACT, TCACT, CACT, ACT, CT, T
9	TAGTCACTC	0	T, TA, TAG, TAGT, TAGTC, TAGTCA, TAGTCACT	AGTCACTC, GTCACTC, TCACTC, CACTC, ACTC, CTC, TC, C

Gambar 1: Pencarian Awalan





Dari query diatas akan didapatkan hasil seperti gambar berikut.

```

SQL> SELECT JOB_ID
FROM EMPLOYEES
WHERE JOB_ID LIKE '%CLE%';

JOB_ID
-----
PU_CLERK
PU_CLERK
PU_CLERK
PU_CLERK
PU_CLERK
PU_CLERK
SH_CLERK

```

Gambar 4: Hasil Data dalam Query SQL

Hal pertama yang harus dilakukan adalah melihat substring yang telah ditentukan oleh operator LIKE, pola apa yang harus dicari. Pada Gambar 4 dapat dilihat bahwa pola atau substring memiliki wildcard '%' yang artinya mewakili beberapa karakter termasuk null dan karakter di antara wildcard ditulis 'CLE' karena di antara karakter tersebut terdapat '%' maka panjang karakter antar 'CLE' disesuaikan dengan teks yang ada. Contoh teks yang akan digunakan adalah 'PU_CLERK' dan pola 'CLE'.

Selanjutnya, tentukan awalan dalam pola awalan dan akhiran 'CLE'.

INDEX	PATTERN	PREFIX	PREFIXES	SUFFIXES
1	C	0	-	-
2	CL	0	C	L
3	CLE	0	C, CL	L, LE

Gambar 5: Pencarian Awalan untuk Pola 'CLE'

Index	1	2	3
Pattern	C	L	E
Prefix	0	0	0

Gambar 6: Nilai Awalan dalam Pola

Setelah menemukan awalan, dapat mencocokkan pola dengan teks. Pergeseran dilakukan sesuai dengan nilai awalan masing-masing karakter yang terdapat pada Gambar 6.

STRING MATCHING KNUTH MORRIS PRATT							
1	2	3	4	5	6	7	8
P	U	-	C	L	E	R	K
C	L	E					
	C	L	E				
		C	L	E			
			C	L	E		

Gambar 7: Mencari Data dalam Pola 'CLE'





Pada Gambar 7 dilakukan pencarian data dari kiri ke kanan dalam proses pencocokan pola dengan teks. Terdapat warna merah yang artinya pola karakter tidak sesuai dengan karakter pada teks sehingga digeser sesuai dengan nilai awalan dan nilai awalan yang semuanya '0' kemudian hanya digeser 1 kali hingga karakter dan teks yang sama pola yang telah diberi warna hijau.

Contoh penerapan algoritma KMP dengan query SQL pada operator LIKE menggunakan dua wildcard '%' dan '_' pada substring atau polanya.

```
SELECT LAST_NAME
FROM EMPLOYEES
WHERE LAST_NAME LIKE '%a_';
```

Dari query di atas, dapatkan data berikut

```
SQL> SELECT LAST_NAME
FROM EMPLOYEES
WHERE LAST_NAME LIKE '%a_';

LAST_NAME
-----
De Haan
Doran
Fay
Kochhar
Kumar
Sullivan
```

Gambar 8: Hasil Data dalam Query SQL

Pada Gambar 8 hasil data pada operator LIKE memiliki 2 wildcard yang berbeda yaitu '%' dan '_' yang artinya jika terdapat wildcard, maka ini mewakili 1 karakter atau karakter khusus dalam sistem SQL. Oleh karena itu pola yang dicari tidak hanya 'a' tetapi 'a [karakter_ khusus]' Karakter khusus dalam jurnal ini direpresentasikan sebagai '#'. Bisa dilihat dari pencarian awalan terlebih dahulu.

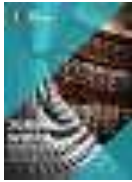
INDEX	PATTERN	PREFIX	PREFIXES	SUFFIXES
1	a	0	-	-
2	a#	0	a	#

Gambar 9: Pencarian Awalan untuk Pola yang Memiliki Karakter Khusus

Index	1	2
Pattern	a	#
Prefix	0	0

Gambar 10: Nilai Awalan dalam Pola





Setelah mendapatkan nilai prefix, maka dapat dicari setiap string yang memiliki proses yang mirip dengan string sebelumnya, namun yang berbeda dari sebelumnya adalah salah satu karakter dalam teks akan berubah menjadi karakter khusus sebagai karakter representatif. Teks yang harus dicoba adalah 'De Haan' di tabel 'LAST_NAME' di database karyawan

STRING MATCHING KNUTH MORRIS PRATT						
1	2	3	4	5	6	7
D	e		H	a	a	n
D	e		H	a	a	#
a	#					
	a	#				
		a	#			
			a	#		
				a	#	
					a	#
						a

Gambar 11: Menemukan Data di Pettern 'a [special_characters]'

Pada Gambar 11 proses pencarian datanya sama dari kiri ke kanan sampai ditemukan setiap pola karakter dengan teks yang cocok satu sama lain. Pencarian berisi karakter 'n' yang diubah menjadi '#' karena hanya sebagai perwakilan karakter dan ada karakter lain tetapi tidak selesai karena indeks atau perwakilan karakter tidak sesuai dengan pola yang ada sehingga 1- terjadi pergeseran waktu ke kanan dan hasilnya terlihat pada tabel yang berwarna hijau.

4 Hasil dan Pembahasan

Penerapan metode algoritma KMP telah terbukti bahwa algoritma KMP dapat melakukan pencarian data pada operator SQL LIKE dengan baik. Karena algoritma KMP melakukan proses dari kiri ke kanan, jika ada sintaks seperti 'LIKE' _a%' akan mudah ditemukan dan akan menghemat perkiraan waktu saat mencari data.

5 Kesimpulan

Berdasarkan hasil perbandingan yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Dalam pencarian data pada operator SQL LIKE menggunakan Algoritma KMP untuk setiap pola yang ada harus disesuaikan dengan *wildcard* pada operator LIKE karena merupakan bagian dari penentu pola yang akan dicocokkan dengan teks atau data pada tabel yang telah dipilih dan juga sebagai penentu nilai awalan untuk perpindahan karakter pada saat proses pencocokan setiap karakter.
2. Dalam hal perbandingan, diharapkan dapat menjadi referensi dan bahan pembelajaran untuk meneliti dengan algoritma pencocokan string lainnya dalam pencarian data di operator SQL LIKE atau operator lain yang mendukung algoritma pencocokan string dalam pencarian datanya.





Referensi (Reference)

- [1] Wikimedia Foundation. "Pangkalan Data". Wikipedia.org. https://id.wikipedia.org/wiki/Pangkalan_data (accessed 17 November 2021)
- [2] Wikimedia Foundation. "SQL". Wikipedia.org. <https://id.wikipedia.org/wiki/SQL> (accessed 17 November 2021)
- [3] Wikimedia Foundation. "Algoritme Pencarian String". Wikipedia.org. https://id.wikipedia.org/wiki/Algoritme_pencarian_string (accessed 17 November 2021)
- [4] C. Ariata (Juni 3, 2019). "Apa itu Query". <https://www.hostinger.co.id/tutorial/apa-itu-query/> (accessed 17 November 2021)
- [5] W3schools. "SQL SELECT Statement". W3schools.com. https://www.w3schools.com/sql/sql_select.asp (accessed 17 November 2021)
- [6] W3schools. "SQL FROM Keyword". W3schools.com. https://www.w3schools.com/sql/sql_ref_from.asp (accessed 17 November 2021)
- [7] W3schools. "SQL WHERE Clause". W3schools.com. https://www.w3schools.com/sql/sql_where.asp (accessed 17 November 2021)
- [8] W3schools. "SQL LIKE Operator". W3schools.com. https://www.w3schools.com/sql/sql_like.asp (accessed 17 November 2021)
- [9] Singh. Puja, Pottle .Brian (April 2009). "Oracle Database 11g: SQL Fundamentals I". (hlm 112-113). (accessed 17 November 2021)
- [10] Pip. (19 Mei 2016). "Algoritma KMP". Piptools.net. <https://piptools.net/algoritma-kmp-knuth-morris-pratt/> (accessed 17 November 2021)
- [11] Wikimedia Foundation. "Algoritma Knuth Morris Pratt". Wikipedia.org. https://id.wikipedia.org/wiki/Algoritme_Knuth-Morris-Pratt (accessed 17 November 2021)

