



Perbandingan Performansi Metode String Matching Menggunakan Metode Naive String Matching, Knuth Morris Pratt, Boyer Moore, Rabin Karp dan SQL Query Like Untuk Pencarian Data Konsumen

Oleh: Isep Lutpi Nur (2113191079)

Pembimbing: Gunawansyah S.T. M.Kom

Program Studi Teknik Informatika

Universitas Sangga Buana YPKP





Latar Belakang

- Pencarian data atau informasi sangat penting dan sering dilakukan oleh banyak orang.
- Pada umumnya string matching menggunakan metode SQL Query Like.
- Terdapat beberapa metode algoritma yang dapat digunakan dalam pencarian string.



Batasan Penelitian

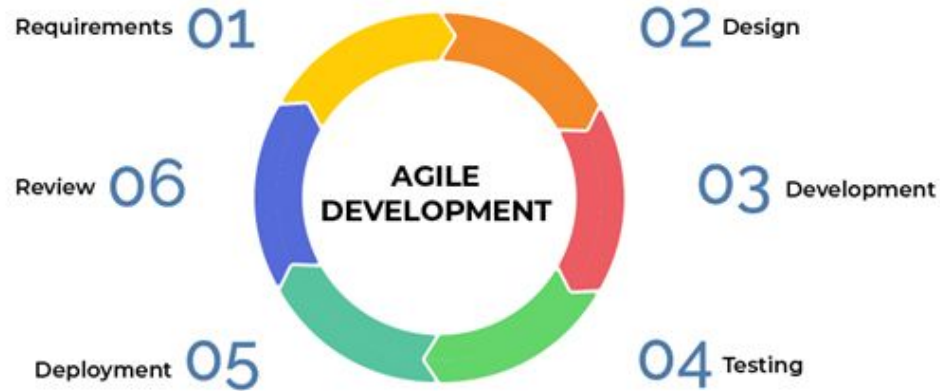
- Penelitian ini hanya akan membahas performa dari algoritma string matching dalam mencari pola pada sebuah string.
- Penelitian ini dilakukan dengan membandingkan waktu eksekusi dan memory yang dibutuhkan algoritma tersebut.
- Penelitian ini hanya akan membahas algoritma string matching yang paling tepat dalam pencarian data berdasarkan karakteristik data dan kebutuhan pengguna.



Metode Pegumpulan Data

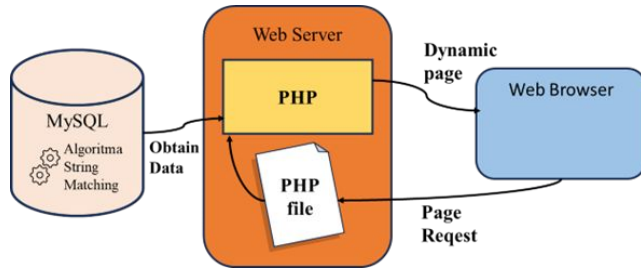
- Studi pustaka
- Eksperimen

Metode Pengembangan Perangkat Lunak

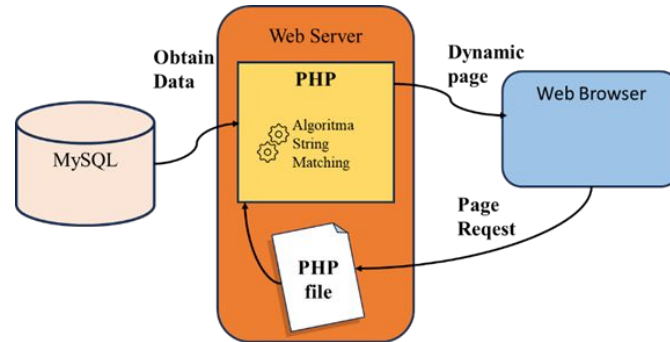


Analisis dan perancangan sistem

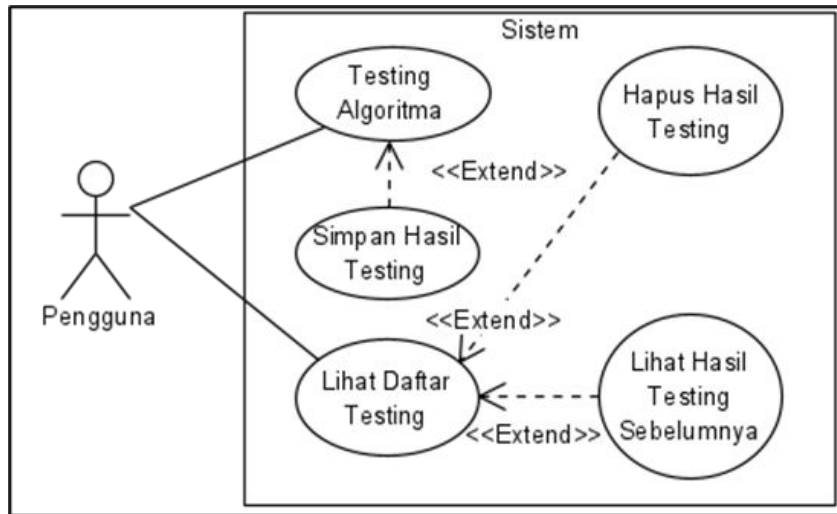
Sistem yang berjalan



Sistem yang diusulkan



Use Case Diagram



Metode penelitian

- Naive String Matching
- Knuth Morris Pratt
- Boyer Moore
- Rabin Karp
- SQL Query Like



Naive String Matching

Berikut adalah pseudocode algoritma naive string matching:

NAIVE-STRING-MATCHING (T, P)

1. $n \leftarrow \text{length } [T]$
2. $m \leftarrow \text{length } [P]$
3. for $i \leftarrow 0$ to $n - m$
4. do if $P [1 \dots m] = T [i + 1 \dots i + m]$
5. then return i



Knuth Morris Pratt

Berikut adalah langkah-langkah operasional algoritma KMP:

1. Mulai dari kiri, bandingkan setiap karakter string yang dicari dengan string yang sedang dikombinasikan.
2. Jika ada ketidakcocokan:
 - Lihat tabel lompatan untuk karakter yang tidak cocok.
 - Geser string yang dicari sejauh yang ditentukan oleh tabel, atau geser ke kanan jika tidak ada entri di tabel.
3. Ulangi proses ini sampai string yang dicari dipindahkan sepenuhnya melalui string yang sedang dikombinasikan.



Boyer Moore

Dalam penggunaan algoritma boyer moore secara umum terdiri dari dua teknik yang harus dilakukan yaitu:

1. The looking-glass technique
2. Fungsi Last Occurrence



Rabin Karp

- Algoritma Rabin-Karp adalah algoritma pencocokan string yang menggunakan fungsi hash sebagai pembanding antara string yang dicari (m) dengan substring pada teks (n).
- Apabila hash value keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya.
- Apabila hasil keduanya tidak sama, maka substring akan bergeser ke kanan.
- Pergeseran dilakukan sebanyak $(n-m)$ kali. Perhitungan nilai hash yang efisien pada saat pergeseran akan mempengaruhi performa dari algoritma ini



SQL Query Like

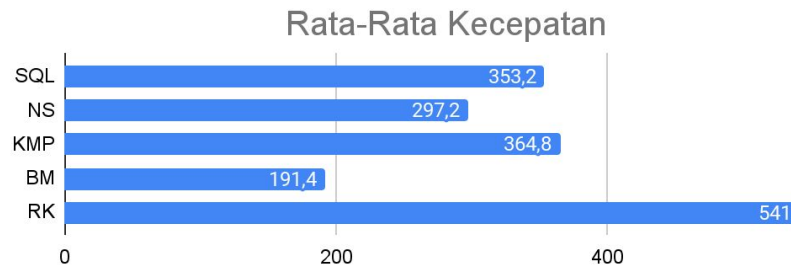
SQL Query Like adalah sebuah pernyataan (statement) yang digunakan dalam bahasa SQL untuk melakukan pencarian data yang cocok dengan pola tertentu pada sebuah tabel atau database.

Pengujian Sistem

- Uji Coba Kasus Rata-rata (Average Case)
- Pengujian Dengan Karakteristik Teks Khusus

Uji Coba Kasus Rata-rata (Average Case)

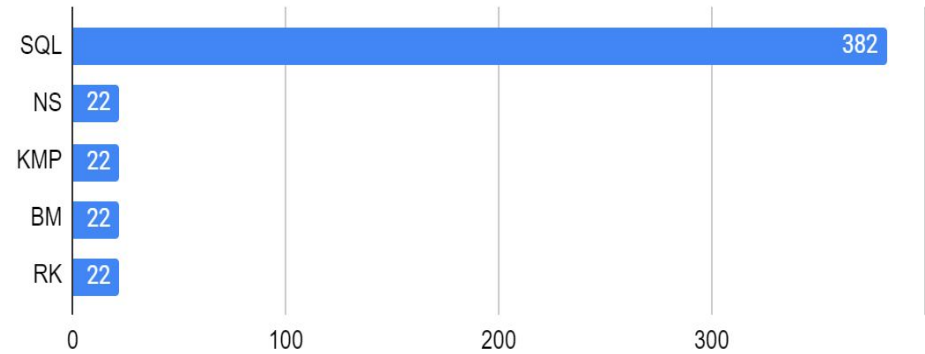
Pada skenario ini akan dilakukan pengujian dengan menggunakan kata kunci “ANDY” selama 5 kali dengan jumlah data 250.000



Pengujian	Kecepatan (ms)				
	SQL	NS	KMP	BM	RK
1	355	292	368	167	553
2	363	311	370	208	566
3	364	277	366	173	530
4	341	294	357	177	542
5	343	312	363	232	514
Rata-Rata	353,2	297,2	364,8	191,4	541

Penggunaan memory

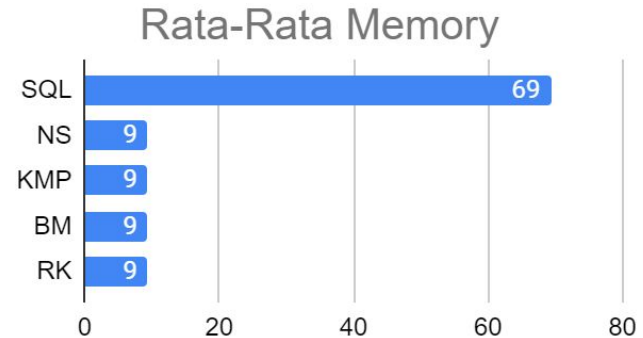
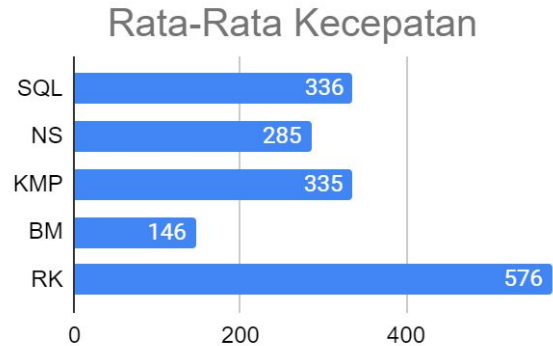
Pengujian	Memory (KB)				
	SQL	NS	KMP	BM	RK
1	382	22	22	22	22
2	382	22	22	22	22
3	382	22	22	22	22
4	382	22	22	22	22
5	382	22	22	22	22
Rata-Rata	382	22	22	22	22





Pengujian Dengan Karakteristik Teks Khusus

Untuk pengujian dengan karakteristik teks khusus ini menggunakan pattern/Kata kunci “123” dengan jumlah pengujian selama 5 kali dengan jumlah data 250.000





Kesimpulan

- Algoritma Boyer Moore lebih cepat dibanding algoritma yang lain. selisih kecepatan 161,8ms dengan SQL Query Like atau 45,8% lebih cepat berdasarkan pengujian performansi dengan skenario kasus rata-rata untuk pencarian data dengan pattern/kata kunci “ANDY” selama 5 kali dengan jumlah data 250,000.
- Penggunaan memory rata-rata memori yang digunakan SQL lebih banyak dibandingkan algoritma yang lain.



Saran

- Melakukan uji coba menggunakan spesifikasi perangkat keras yang lebih unggul dan lebih variatif.
- Melakukan uji coba dengan melibatkan jumlah pengguna yang lebih besar pada waktu yang sama.
- Uji coba dilakukan menggunakan hosting server.
- Dalam penelitian ini, 250.000 data pelanggan digunakan dan jumlah ini dapat ditingkatkan.



Terima Kasih