**LOFIAN**: Jurna**l** Tekn**o**logi In**f**ormas**i** d**a**n Komu**n**ikasi
https://ejournal.umbp.ac.id/index.php/lofian/
Vol. 2, No. 1, September 2022, Halaman 24-28

e-ISSN: 2798-9593
p-ISSN: 2798-9836

# Analisa Pembuatan Aplikasi Pencarian Kata Dalam Dokumen Teks Dengan Metode Hashing

Jimmy Peranginangin[1], Fauzi Haris Simbolon[2]

[1,2]*Universitas Mandiri Bina Prestasi*
*Jl. Letjen. Djamin Ginting No. 285-287, Padang Bulan, Medan Baru, Medan, Sumatera Utara, Indonesia - 20155*

[1]*jimmy.mbp@gmail.com,* [2]*farisboys@dsn.umbp.ac.id*

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

**Abstrak**

Pencarian (searching) merupakan suatu kegiatan yang sering sekali dilakukan oleh pengguna komputer dengan tujuan untuk mendapatkan suatu informasi. Proses pencarian data sangat bergantung pada cara penyimpanan, media penyimpanan serta kerumitan data pada saat penyimpanan. Salah satu cara penyimpanan data yang di kenal adalah penyimpanan data secara acak (random) dan secara berurutan (sequential). Analisis pencarian alamat data dengan empat teknik menggunakan algoritma yang sesuai dengan teknik masing-masing pada metoda Hashing dengan data teks yang tingkat kerumitan data yang sama, yaitu: Teknik dengan sisa pembagian, Teknik Pemenggalan, Teknik Lipatan, dan Teknik Perkalian. Setelah mengaplikasikan dalam perangkat lunak, maka dapat diambil kesimpulan bahwa proses pencarian lokasi penyimpanan dalam satu dokumen teks dengan menggunakan empat teknik pada metoda Hashing memberikan alamat lokasi penyimpanan dan waktu yang berbeda. Dan hasil yang berbeda pada keempat teknik pada metoda Hashing bertujuan untuk mencegah tubrukan (collision).

*Kata Kunci*: Pencarian data, metode hashing, penyimpanan data, collision, aplikasi.

## 1. Pendahuluan

Pencarian (searching) merupakan suatu kegiatan yang sering sekali dilakukan oleh pengguna komputer dengan tujuan untuk mendapatkan suatu informasi. Data yang dicari dapat berupa sebuah file teks maupun file database. Proses pencarian data sangat bergantung pada cara penyimpanan, media penyimpanan serta kerumitan data pada saat penyimpanan. Untuk mengidentifikasi setiap data pada tabel, maka data harus dinyatakan dengan suatu nilai yang dikenal sebagai kunci dengan tujuan sebagai acuan data (ciri ciri data yang spesifik).

Kecepatan suatu algoritma dalam menemukan data merupakan hal yang sangat penting, makin baik algoritma maka semakin cepat menemukan data yang akan dicari. Salah satu cara penyimpanan data yang di kenal adalah penyimpanan data secara acak (random) dan secara berurutan (sequential).

Metode yang sering dipakai untuk mendapatkan posisi dalam satu file (alamat relatif) suatu data adalah metode Hashing (Scatter storage) yang meliputi suatu perhitungan aritmatika pada nilai kunci untuk menghasilkan satu bilangan bulat/integer yang disimpan dalam direktori dan direktori disimpan sebagai satu array. Fungsi Hashing merupakan penentuan alamat record pada file data dengan melakukan perhitungan pada nilai kunci dari record tersebut. Dalam menyimpan suatu data sering ditemukan data yang sama dan ini akan menyebabkan tubrukan (collision). Suatu pendekatan yang sederhana untuk mencegah terjadinya tubrukan dengan cara menyimpan elemen yang bertubrukan dalam ruang berikutnya yang tersedia dengan alamat indeks.

## 2. Tinjauan Pustaka

### *2.1. Pencarian (Searching)*

Pencarian (Searching) adalah proses untuk menemukan dan mendapatkan suatu nilai berdasarkan satu kunci (key) yang dimana bisa juga disebutkan sebagai acuan data. Data yang dimaksud pada pencarian tersebut bisa juga bersifat integer. Jadi, didalam pencarian sebuah data bukanlah tidak mungkin akan terjadi sebuah tubrukan data/nilai yang sama dalam sebuah dokumen teks.

Dalam proses pencarian terdapat beberapa ketergantungan untuk memudahkan pencarian antara lain:

1. Media tempat penyimpanan data (memori, tape, disk).
2. Karakteristik Data yang akan disimpan.

e-ISSN: 2798-9593
p-ISSN: 2798-9836

3. Jumlah data yang akan disimpan untuk proses pencarian yang secepat– cepatnya.

*2.2. Teknik-teknik Pencarian (Searching)*

Dalam teknik pencarian data ada dua hal yang harus diperhatikan dan dilakukan. Adapun hal tersebut adalah sebagai berikut:

1. Pencarian Berurutan
   Metoda yang paling sederhana dari sejumlah metoda pencarian adalah pencarian berurutan. Secara garis besar metoda ini bisa dijelaskan sebagai berikut. Dari vektor yang diketahui data yang dicari dibandingkan satu persatu sampai tersebut ditemukan atau tidak ditemukan. Pada saat data yang dicari sudah ketemu, maka proses pencarian langsung dihentikan. Tetapi jika data yang dicari belum ketemu, maka pencarian diteruskan sampai seluruh data dibandingkan. Satu persatu sampai data tersebut ditemukan atau tidak ditemukan. Pada saat data yang dicari belum ketemu, maka proses pencarian langsung dihentikan. Tetapi jika data yang dicari belum ketemu, maka pencarian diteruskan sampai seluruh data dibandingkan.
2. Pencarian Berurutan Berindeks
   Metoda pencarian berurutan berindeks adalah metoda lain dari beberapa pencarian yang dapat menaikkan efisiensi pencarian pada tabel yang sudah dalam keadaan urut. Dalam hal ini diperlukan tabel tambahan yang disebut dengan tabel indeks. Setiap elemen dalam tabel indeks berisi suatu kunci dan pionter yang menunjuk kerekaman tersebut. Elemen-elemen dalam tabel indeks juga harus diurutkan seperti halnya dalam tabel asli. Kita bisa menyimpan tabel diatas untuk diimplementasikan menggunakan suatu struktur data tersebut untuk mendukung pencarian data berurutan berindeks dengan beberapa cara. Pemakaian indeks juga bisa diimplementasikan menggunakan senarai berantai seperti halnya dengan penggunaan larik.
   Jika tabel data yang digunakan cukup besar sehingga meskipun sudah digunakan tabel berurutan terindeks efisiensi pencarian yang diinginkan tidak bisa dicapai (mungkin karena indeks terlalu besar untuk mengurangi pencarian berurutan, atau karena indeksnya terlalu kecil sehingga kunci-kunci yang berdekatan cukup jauh satu sama lain), seringkali digunakan indeks sekunder. Indeks sekunder ini juga berfungsi sama dengan indeks primer. Penghapusan elemen dari tabel berurutan terindeks akan lebih mudah apabila digunakan suatu tanda khusus (flag) yang menunjukkan bahwa suatu elemen sudah dihapus.

*2.3. Hashing*

Hashing adalah pengembangan dari teknik pencarian data (Searching). Teknik ini berkembang seiring dengan perkembangan teknologi sistem basis data, yaitu data yang disimpan di dalam media penyimpan dimana datanya begitu banyak, seperti harddisk dan media sejenisnya, sehingga untuk mengakses data tersebut dibutuhkan waktu yang singkat. Dengan metode hashing inilah, mulai ditemukan dan dikembangkan trik yang mempercepat proses pencarian data yang biasanya tersimpan di dalam memori.

Fungsi Hashing (Hashing Function) adalah rumusan yang memetakan elemen ke dalam tabel Hash. Terdapat 4 hal penting yang terkait, yaitu:

1. Home address adalah alamat yang dihasilkan oleh fungsi Hashing yang menunjukkan lokasi elemen tersebut di dalam tabel Hash.
2. Collision adalah tabrakan yang terjadi bila terdapat 2 atau lebih elemen yang di-hash ke lokasi yang sama sehingga elemen-elemen tersebut mempunyai home address yang sama.
3. Digit Selection adalah teknik Hashing yang melakukan pemilihan digit dari key sebelum dilakukan division. Digit yang dipilih sebaiknya yang bisa menghasilkan nilai yang random.
4. Rehashing adalah teknik yang digunakan bila terjadi Collision, yang biasa dikenakan dengan istilah Collision Resolution strategies.

*2.4. Teknik Pencarian dengan Metoda Hashing*

Teknik hashing meliputi perhitungan aritmatika pada nilai kunci untuk menghasilkan satu bilangan bulat/integer. Bilangan bulat ini merupakan alamat relatif dimana nilai kunci disimpan didirektori dan direktori disimpan sebagai array. Untuk mencari alamat relative dengan cara menghitung kunci di salah satu petunjuk array.

Ada 7 teknik yang digunakan dalam metoda hashing yaitu:

1. Teknik Sisa Pembagian (Division Remainder)
   Merupakan teknik yang paling sederhana dengan membagi satu nilai kunci dengan ukuran tabel (tempat data yang akan dimasukkan dan diproses) atau direktori dimana sisa pembagian merupakan alamat relative untuk data atau record. Kekurangan teknik ini adalah jika hasil pembagian adalah 0 (nol). Maka nilai indeksnya adalah 0 sebagai indeks. Juga dengan teknik ini diasumsikan tidak adanya tubrukan. Alamat dihasilkan dari suatu nilai dengan perhitungan MOD.

e-ISSN: 2798-9593
p-ISSN: 2798-9836

Jika ukuran tabel n dan nilai kunci = 648 maka f (648) MOD 11 = 10. Jadi nilai 648 disimpan dialamat indeks 10.

2. Teknik Pemenggalan (Truncation)
   Merupakan teknik yang paling mudah digunakan bagi programmer dengan melakukan penghilangan beberapa digit pertama dari data atau terakhir, teknik ini mempunyai kekurangan karena terbatasnya ukuran ruang alamat untuk bentuk kelipatan 10. Pemenggalan dilakukan dengan menghilangkan digit k yang pertama atau yang terakhir dari sejumlah n digit.
3. Teknik Lipatan (Folding)
   Dalam teknik ini nilai kunci dibagi menjadi beberapa bagian, masing-masing memiliki jumlah digit yang sama. Bagian ini dapat dilipat antara satu dengan bagian yang lain. Hasil penjumlahannya setelah dilipat digit dengan order paling tinggi dipenggal menjadi alamat relatif record atau data.
4. Teknik Perkalian (Multiplication)
   Teknik ini dilakukan dengan cara membagi bagi satu nilai kunci dan kemudian menjumlah bagian bagiannya seperti yang dilakukan di teknik lipatan. Bagian dari salah satu kunci bisa dipilih untuk kemudian dilakukan dan hasilnya merupakan alamat relatif record atau data.
5. File Terurut (Sequential)
   Adalah jenis data organisasi urut berdasarkan urutan pemasukan data yaitu berdasarkan nomor record yang mana data yang ditambahkan selalu menempati urutan berikutnya. Sekumpulan record yang disimpan dalam media penyimpanan sekunder komputer dapat diakses secara berurutan mulai dari record pertama sampai record terahkir. Record terahkir adalah rekaman fiktif yang menandai akhir dari arsip. Struktur file sequential berindeks dibuat dengan tujuan untuk kebutuhan akses data secara langsung.
6. File Acak
   Adalah jenis data dengan tanpa organisasi pengurutan, yang mana data yang ditambahkan selalu menempati urutan berikutnya tanpa melihat kunci atribut dari data yang masuk. Untuk bisa mengakses data yang acak memerlukan lokasi atau alamat file, dalam menentukan alamat file ada tiga teknik yaitu: Pengalamatan absolut/tetap, Pengalamatan relative, dan Pengalamatan indeks
7. Pengaksesan Secara Acak
   Ketika suatu record dibaca atau ditulis, maka satu nilai kunci dihubungkan dengan record yang diketahui. Untuk menghubungkan nilai kunci tersebut dengan suatu alamat bisa alamat absolut maupun relatif dengan direktori yang sering digunakan. Direktori merupakan suatu tabel dengan dua kolom dimana kolom pertama berisi nilai kunci dan dalam kedua berisi sekumpulan alamat record yang ditunjukkan oleh nilai kunci.

## 3. Pembahasan

### *3.1. Metoda Hashing*

Analisis pencarian alamat data dengan empat teknik menggunakan algoritma yang sesuai dengan teknik masing-masing pada metoda Hashing dengan data teks yang tingkat kerumitan data yang sama. Data teks yang sudah disiapkan terdiri dari data hasil nilai ujian mahasiswa seperti pada tabel 1.

Tabel 1. Data Hasil Ujian

| No | NIM | Nama Mahasiswa | UTS | UAS |
|---|---|---|---|---|
| 1. | 20065 | Karianto | 56 | 90 |
| 2. | 20068 | Rudi | 60 | 76 |
| 3. | 20067 | Subianto | 57 | 40 |
| 4. | 20064 | Subandrio | 89 | 90 |
| 5. | 20061 | Antony Salim | 58 | 78 |
| 6. | 20066 | Ciputra | 78 | 60 |
| 7. | 20063 | Sodomo | 89 | 45 |
| 8. | 20062 | Cut Keke | 78 | 97 |

UTS = Ujian Tengah Semester
UAS = Ujian Akhir Semester
Dimana ukuran field:

| | |
|---|---|
| Nomor Urut | : 5 byte |
| Nim | : 5 byte |
| NamaMahasiswa | : 20 byte |
| Nilai UTS | : 5 byte |
| Nilai UAS | : 5 byte |

Ukuran field rata rata = 40/5 = 8 byte
Ukuran record = 8 x 5 = 40 byte
Jumlah record sebanyak 8 record

Kunci pencarian adalah NIM dan diasumsikan ukuran tabel = 11

### *3.2. Teknik dengan sisa pembagian*

Pada teknik ini data teks yang akan di proses adalah data yang dimasukkan pada form inputan dan data tersebut dibandingkan dengan file teks, proses yang pertama dilakukan adalah menghitung ukuran dari data teks, lalu data yang hendak dicari sisa pembagiannya, kemudian sisa pembagiannya yang menjadi alamat relatif data yang dicari. Jadi diperlukan data yang akan dicari sebagai nilai kunci.

e-ISSN: 2798-9593
p-ISSN: 2798-9836

Untuk Lokasi dengan nilai kunci 20061=20061 mod 11=8
Untuk Lokasi dengan nilai kunci 20062=20062 mod 11=9
Untuk Lokasi dengan nilai kunci 20063=20063 mod 11=10
Untuk Lokasi dengan nilai kunci 20064=20064 mod 11=11
Untuk Lokasi dengan nilai kunci 20065=20065 mod 11=12
Untuk Lokasi dengan nilai kunci 20066=20066 mod 11=13
Untuk Lokasi dengan nilai kunci 20067=20067 mod 11=14
Untuk Lokasi dengan nilai kunci 20068=20068 mod 11=15

Tabel 2. Hashing dengan Teknik Sisa-Pembagian

| Indeks | 61 | 62 | 63 | 64 | … |
|---|---|---|---|---|---|
| Nilai Kunci | 20061 | 20062 | 20063 | 20064 | … |

*3.3. Teknik Pemenggalan*

Pada teknik ini data yang dicari akan diinput kedalam form inputan lalu data tersebut dibandingkan dengan data teks dengan proses penghilangan 3 digit pertama dari data yang dicari. Hasil penghilangan tersebut merupakan alamat relatif dari data yang dicari yang ditampilkan pada form output.

Untuk Lokasi dengan nilai kunci f(20061)= 61
Untuk Lokasi dengan nilai kunci f(20062)= 62
Untuk Lokasi dengan nilai kunci f(20063)=63
Untuk Lokasi dengan nilai kunci f(20064)=64
Untuk Lokasi dengan nilai kunci f(20065)=65
Untuk Lokasi dengan nilai kunci f(20066)=66
Untuk Lokasi dengan nilai kunci f(20067)= 67
Untuk Lokasi dengan nilai kunci f(20068)= 68
Untuk Lokasi dengan nilai kunci f(20069)= 69
Untuk Lokasi dengan nilai kunci f(20070)= 70

*3.4. Teknik Lipatan*

Pada teknik ini data yang dicari akan diinput kedalam form inputan lalu nilai kunci dari data yang dicari dibagi menjadi beberapa bagian, masing masing memiliki jumlah digit yang sama, jika jumlah digit tidak sama maka ditambahkan angka 0 (nol) pada awal data seperti dibawah ini.

Untuk kunci 20060 = 20 060 → dilipat 060 + 020 → 80. Jadi alamat relatif nilai kunci 20060 adalah pada lokasi 80
Untuk kunci 20061 = 20 061 → dilipat 061 + 020 → 81. Jadi alamat relatif nilai kunci 20061 adalah pada lokasi 81
Untuk kunci 20062 = 20 062 → dilipat 062 + 020 → 82. Jadi alamat relatif nilai kunci 20062 adalah pada lokasi 82
Untuk kunci 20063 = 20 063 → dilipat 063 + 020 → 83. Jadi alamat relatif nilai kunci 20063 adalah pada lokasi 83
Untuk kunci 20064 = 20 064 → dilipat 064 + 020 → 84. Jadi alamat relatif nilai kunci 20064 adalah pada lokasi 84
Untuk kunci 20065 = 20 065 → dilipat 065 + 020 → 95. Jadi alamat relatif nilai kunci 20065 adalah pada lokasi 95
Untuk kunci 20066 = 20 066 → dilipat 066 + 020 → 96. Jadi alamat relatif nilai kunci 20066 adalah pada lokasi 96
Untuk kunci 20067 = 20 067 → dilipat 067 + 020 → 97. Jadi alamat relatif nilai kunci 20067 adalah pada lokasi 97
Untuk kunci 20068 = 20 068 → dilipat 068 + 020 → 88. Jadi alamat relatif nilai kunci 20068 adalah pada lokasi 88
Untuk kunci 20069 = 20 068 → dilipat 069 + 020 → 89. Jadi alamat relatif nilai kunci 20069 adalah pada lokasi 89
Untuk kunci 20070 = 20 070 → dilipat 070 + 020 → 90. Jadi alamat relatif nilai kunci 20070 adalah pada lokasi 90

*3.5. Teknik Perkalian*

Teknik ini dilakukan dengan cara membagi bagi satu nilai kunci dan kemudian menjumlah bagian-bagiannya seperti yang dilakukan pada teknik lipatan, bagian dari satu kunci bisa dipilih untuk kemudian dikalikan. Perkalian bagian-bagian dari kunci untuk menyebarkan alamat relatif yang dihitung dan bisa mengurangi resiko terjadinya tubrukan. Keberhasilan teknik ini bergantung dari sifat-sifat dasar dari kunci yang digunakan.

Untuk kunci 20060 dipenggal = 20 dan 060 → dikalikan menjadi 1200. Jadi alamat relatif nilai kunci 20060 adalah pada lokasi 1200
Untuk kunci 20061 dipenggal = 20 dan 061 → dikalikan menjadi 1220. Jadi alamat relatif nilai kunci 20061 adalah pada lokasi 1220
Untuk kunci 20062 dipenggal = 20 dan 062 → dikalikan menjadi 1240. Jadi alamat relatif nilai kunci 20062 adalah pada lokasi 1240

e-ISSN: 2798-9593
p-ISSN: 2798-9836

Untuk kunci 20063 dipenggal = 20 dan 063 → dikalikan menjadi 1260. Jadi alamat relatif nilai kunci 20063 adalah pada lokasi 1260
Untuk kunci 20064 dipenggal = 20 dan 064 → dikalikan menjadi 1280. Jadi alamat relatif nilai kunci 20064 adalah pada lokasi 1280
Untuk kunci 20065 dipenggal = 20 dan 065 → dikalikan menjadi 1300. Jadi alamat relatif nilai kunci 20065 adalah pada lokasi 1300
Untuk kunci 20066 dipenggal = 20 dan 066 → dikalikan menjadi 1320. Jadi alamat relatif nilai kunci 20066 adalah pada lokasi 1320
Untuk kunci 20067 dipenggal = 20 dan 067 → dikalikan menjadi 1340. Jadi alamat relatif nilai kunci 20067 adalah pada lokasi 1340
Untuk kunci 20068 dipenggal = 20 dan 068 → dikalikan menjadi 1360. Jadi alamat relatif nilai kunci 20068 adalah pada lokasi 1360
Untuk kunci 20069 dipenggal = 20 dan 069 → dikalikan menjadi 1380. Jadi alamat relatif nilai kunci 20069 adalah pada lokasi 1380
Untuk kunci 20070 dipenggal = 20 dan 070 → dikalikan menjadi 1400. Jadi alamat relatif nilai kunci 20070 adalah pada lokasi 1400

## 4. Kesimpulan

Setelah mengaplikasikan dalam perangkat lunak, maka dapat diambil kesimpulan bahwa proses pencarian lokasi penyimpanan dalam satu dokumen teks dengan menggunakan empat teknik pada metoda Hashing memberikan alamat lokasi penyimpanan dan waktu yang berbeda. Dan hasil yang berbeda pada keempat teknik pada metoda Hashing bertujuan untuk mencegah tubrukan (collision).

## Referensi

[1] Benedict, M., Budiman, A., & Rachmawati, D. (2017). PERBANDINGAN ALGORITMA MESSAGE DIGEST-5 (MD5) DAN GOSUDARSTVENNYI STANDARD (GOST) PADA HASHING FILE DOKUMEN. JTIK (Jurnal Teknik Informatika Kaputama), 1(1), 52-63.
[2] Erwin Daniel Sitanggang et al 2019 J. Phys.: Conf. Ser. 1235 012061
[3] Irawan, B., Sitanggang, E., & Achmady, S. (2021). Sistem Pendukung Keputusan Tingkat Kepuasan Pasien Terhadap Mutu Pelayanan Rumah Sakit Berdasarkan Metode ServQual. Jurnal of Computer Engineering, System and Science (CESS), 6(1), 10-19. https://doi.org/10.24114/cess.v6i1.21023
[4] Mahardika, G., Indriati, I., & Adikara, P. (2021). Klasifikasi Dokumen Berita Menggunakan Feature Hashing Dan Jaringan Saraf Tiruan. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 5(13). Diambil dari https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/9348
[5] Pambudi, A., Kusyanti, A., & Data, M. (2018). Perancangan Sistem Pengamanan Data Transaksi Pada Database Terdistribusi Menggunakan Metode Hashing. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 3(1), 247-252. Diambil dari https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4085
[6] Qiu, N., Hu, X., Wang, P., & Yang, H. (2016). Research on Optimization Strategy to Data Clustered Storage of Consistent Hashing Algorithm. TELKOMNIKA: Telecommunication, Computing, Electronics and Control, 14(3), 824-830. doi: http://doi.org/10.12928/telkomnika.v14i3.3550
[7] Wulandari, D., & Rusjdi, D. (2017). RANCANG BANGUN MEDIA PEMBELAJARAN TEKNIK HASHING SEBAGAI UPAYA UNTUK MENINGKATKAN KEMAMPUAN MAHASISWA TERHADAP PEMROSESAN DAN PENCARIAN FILE SECARA ACAK DALAM MATAKULIAH SISTEM BERKAS. Jurnal Ilmiah FIFO, 9(1), 57-67. doi:http://dx.doi.org/10.22441/fifo.2017.v9i1.007
[8] Yudhana, A., Fadlil, A., & Prianto, E. (2018). Performance Analysis of Hashing Mathods on the Employment of App. International Journal of Electrical and Computer Engineering (IJECE), 8(5), 3512-3522. doi: http://10.11591/ijece.v8i5.pp3512-3522
[9] Yulianingsih. (2017). Implementasi Algoritma Jaro-Winkler dan Levenstein Distance dalam Pencarian Data pada Database. STRING (Satuan Tulisan Riset dan Inovasi Teknologi), 2(1), 18-27. doi: https://doi.org/10.30998/string.v2i1.1720