

Dokumentasi Fungsi Hash

Data Structure



**Cakrawala
University**

Nama: Irfan Luthfiardi Anhar

Nim: 24110300007

Kelas: DS06

1. Pendahuluan

Hashing adalah teknik penting dalam struktur data untuk melakukan pencarian dan penyimpanan data secara efisien. Tujuan dari tugas ini adalah mengevaluasi empat metode fungsi hash terhadap empat jenis dataset menggunakan chaining untuk penanganan kolisi. Metrik yang digunakan adalah jumlah kolisi, load factor, dan panjang rantai maksimum (max chain length).

2. Deskripsi Fungsi Hash

Program mengevaluasi empat fungsi hash:

- **Division Method:** $h(k) = k \bmod m$
- **Multiplication Method:** $h(k) = \text{floor}(m * (k * A \bmod 1))$, dengan $A = 0.6180339887$
- **Mid-Square Method:** Mengkuadratkan k , mengambil 4 digit tengah, lalu mod m
- **Folding Method:** Memecah angka jadi pasangan digit, menjumlahkannya, lalu mod m

Semua fungsi menerima key numerik, yang dihasilkan dari string melalui konversi sederhana (stringToInt).

3. Deskripsi Dataset

Empat dataset diuji untuk variasi distribusi:

- **Dataset 1 – Berurutan:** 100 bilangan berurutan dari 1000–1099
- **Dataset 2 – Random:** 150 bilangan acak
- **Dataset 3 – Terstruktur:** 120 angka dengan urutan yang terstruktur
- **Dataset 4 – Campuran:** 80 ID unik berisi karakter, angka, dan symbol

4. Ukuran Tabel Hash

Tiga ukuran tabel hash yang digunakan yaitu:

- $m = 31$
- $m = 101$
- $m = 503$

5. Metodologi Evaluasi

Untuk setiap kombinasi dataset, ukuran tabel, dan fungsi hash, dihitung metrik:

- **Collisions:** Jumlah bentrokan di semua bucket (elemen ke- n dalam satu bucket dihitung sebagai $n-1$ kolisi)
- **Load Factor:** Rasio jumlah item terhadap ukuran tabel (n/m)
- **Max Chain Length:** Panjang maksimum dari satu bucket
- **Best Method:** Metode terbaik setiap dataset

6. Contoh Hasil Output

1) Dataset 1:

a) Table size $m = 31$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	69	3.23	4	Multiplication
Multiplication	69	3.23	4	
Mid-Square	69	3.23	9	
Folding	69	3.23	4	

b) Table size $m = 101$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	0	0.99	1	Division
Multiplication	13	0.99	2	
Mid-Square	29	0.99	6	
Folding	0	0.99	1	

c) Table size $m = 503$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	0	0.20	1	Multiplication
Multiplication	0	0.20	1	
Mid-Square	6	0.20	2	
Folding	0	0.20	1	

2) Dataset 2:

a) Table size $m = 31$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	199	4.84	9	Multiplication
Multiplication	199	4.84	8	
Mid-Square	199	4.84	11	
Folding	199	4.84	12	

b) Table size $m = 101$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	72	1.49	4	Multiplication
Multiplication	67	1.49	5	
Mid-Square	75	1.49	5	

Folding	70	1.49	5	
---------	----	------	---	--

c) Table size $m = 503$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	16	0.30	3	Division
Multiplication	18	0.30	2	
Mid-Square	18	0.30	2	
Folding	61	0.30	5	

3) Dataset 3:

a) Table size $m = 31$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	89	3.87	4	Mid-Square
Multiplication	89	3.87	5	
Mid-Square	89	3.87	5	
Folding	100	3.87	10	

b) Table size $m = 101$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	19	1.19	2	Mid-Square
Multiplication	27	1.19	2	
Mid-Square	103	1.19	10	
Folding	100	1.19	10	

c) Table size $m = 503$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	0	0.24	1	Division
Multiplication	0	0.24	1	
Mid-Square	6	0.24	2	
Folding	100	0.24	10	

4) Dataset 4:

a) Table size $m = 31$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	40	2.19	5	Multiplication
Multiplication	40	2.19	5	

Mid-Square	42	2.19	6	
Folding	41	2.19	6	

b) Table size $m = 101$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	20	0.67	4	Mid-Square
Multiplication	20	0.67	4	
Mid-Square	14	0.67	3	
Folding	20	0.67	3	

c) Table size $m = 503$

Metode Hash	Collision	Load Factor	Max Chain Length	Best Method
Division	3	0.14	2	Division
Multiplication	5	0.14	2	
Mid-Square	4	0.14	2	
Folding	13	0.14	3	

7. Hasil Analisis

- **Dataset 1 (Berurutan):** Metode Multiplication menunjukkan distribusi yang lebih merata dibanding Division, Mid-Square, dan Folding.
- **Dataset 2 (Random):** Semua metode menunjukkan performa yang cukup baik, namun Mid-Square dan Folding cenderung lebih stabil pada ukuran tabel besar.
- **Dataset 3 (Terstruktur):** Karena kemiripan pola angka, metode seperti Division menghasilkan banyak kolisi. Multiplication tetap menunjukkan kinerja lebih baik.
- **Dataset 4 (Campuran):** Karena banyak karakter non-digit, konversi ke integer (stringToInt) menyebabkan penyebaran yang cukup acak, dan Multiplication serta Mid-Square cenderung unggul.

8. Kesimpulan

Berdasarkan hasil evaluasi terhadap empat metode hashing Division, Multiplication, Mid-Square, dan Folding dengan berbagai ukuran tabel dan tipe dataset, dapat disimpulkan bahwa metode Division menunjukkan performa paling stabil dan efisien, terutama untuk data numerik dan terstruktur, dengan jumlah collision dan panjang rantai maksimum yang rendah saat ukuran tabel cukup besar (misalnya $m = 503$). Metode Multiplication memiliki hasil yang cukup sebanding namun tidak secara konsisten lebih unggul. Sementara itu, Mid-Square dan Folding cenderung lebih cocok digunakan untuk data string kompleks atau tidak teratur, meskipun keduanya rentan menghasilkan collision tinggi pada dataset berurutan dan ukuran tabel kecil. Secara umum, pemilihan metode hash yang optimal sangat bergantung pada karakteristik data dan ukuran tabel hash yang digunakan, dengan tabel berukuran prima dan cukup besar menjadi faktor kunci untuk meminimalkan collision dan distribusi yang merata.

