

Nama : Frederico Steven Kwok

NPM : 242310037

Kelas : TI-24-PA

Matkul : Lab. Desain dan Analisis Algoritma

Github : https://github.com/RicoSteven120206/Desain_Analisis_Algoritma.git

Tugas : Buatlah sebuah program sederhana dari hash table yang menggunakan metode double hashing!

1. Screenshot Source Code

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  const int emptySlot = -1;
6  const int deleteSlot = -2;
7
8  class DoubleHashing {
9  private:
10     vector<int> hashTable;
11     int tableSize;
12     int prime;
13
14 public:
15     DoubleHashing(int size) {
16         this->tableSize = size;
17         this->prime = 7;
18         hashTable.assign(tableSize, emptySlot);
19     }
20
21     int hash1(int key) {
22         return key % tableSize;
23     }
24
25     int hash2(int key) {
26         return prime - (key % prime);
27     }
28
29     void insert(int key) {
30         if (search(key) != -1) {
31             cout << "-> Kunci " << key << " sudah ada di dalam tabel." << endl;
32             return;
33         }
34     }
```

```

34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
};

int h1 = hash1(key);

if (hashTable[h1] == emptySlot || hashTable[h1] == deleteSlot) {
    hashTable[h1] = key;
    cout << "-> Kunci " << key << " berhasil disisipkan di indeks " << h1 << "." << endl;
    return;
}

int h2 = hash2(key);
for (int i = 1; i < tableSize; ++i) {
    int probeIndex = (h1 + i * h2) % tableSize;

    if (hashTable[probeIndex] == emptySlot || hashTable[probeIndex] == deleteSlot) {
        hashTable[probeIndex] = key;
        cout << "-> Kunci " << key << " disisipkan di indeks " << probeIndex << " (setelah " << i << " kali probe)." << endl;
        return;
    }
}

cout << "-> Tabel hash penuh, tidak bisa menyisipkan kunci " << key << "." << endl;
}

int search(int key) {
    int h1 = hash1(key);
    int h2 = hash2(key);

    for (int i = 0; i < tableSize; ++i) {
        int probeIndex = (h1 + i * h2) % tableSize;

        if (hashTable[probeIndex] == key) {
            return probeIndex;
        }

        if (hashTable[probeIndex] == emptySlot) {
            return -1;
        }
    }

    return -1;
}

void remove(int key) {
    int index = search(key);
    if (index != -1) {
        hashTable[index] = deleteSlot;
        cout << "-> Kunci " << key << " di indeks " << index << " berhasil dihapus." << endl;
    } else {
        cout << "-> Kunci " << key << " tidak ditemukan di dalam tabel." << endl;
    }
}

void display() {
    cout << "\n===== ISI TABEL HASH =====\n";
    for (int i = 0; i < tableSize; i++) {
        cout << "Indeks " << i << ":\t";
        if (hashTable[i] == emptySlot) {
            cout << "[ Kosong ]" << endl;
        } else if (hashTable[i] == deleteSlot) {
            cout << "[ Dihapus ]" << endl;
        } else {
            cout << hashTable[i] << endl;
        }
    }
    cout << "===== \n";
}

```

```

101
102 int main() {
103     int table_size;
104
105     int choice, key;
106
107     cout<<"Masukkan Jumlah Table Size : ";
108     cin>>table_size;
109
110     DoubleHashing ht(table_size);
111
112     do {
113         cout<<"1. Sisipkan Kunci (Insert)"<<endl;
114         cout<<"2. Cari Kunci (Search)"<<endl;
115         cout<<"3. Hapus Kunci (Delete)"<<endl;
116         cout<<"4. Tampilkan Tabel Hash"<<endl;
117         cout<<"5. Selesai"<<endl;
118         cout<<"Pilihan Anda: ";
119         cin>>choice;
120
121         if(choice == 1) {
122             cout<<"Masukkan kunci yang akan disisipkan: ";
123             cin>>key;
124             ht.insert(key);
125         } else if (choice == 2) {
126             cout<<"Masukkan kunci yang akan dicari: ";
127             cin>>key;
128             int index = ht.search(key);
129             if (index != -1) {
130                 cout << "-> Kunci " << key << " ditemukan di indeks " << index << "." << endl;
131             } else {
132                 cout << "-> Kunci " << key << " tidak ditemukan." << endl;
133             }
134         } else if (choice == 3) {
135             cout<<"Masukkan kunci yang akan dihapus: ";
136             cin>>key;
137             ht.remove(key);
138         } else if (choice == 4) {
139             ht.display();
140         } else if (choice == 5) {
141             cout<<"Program Selesai"<<endl;
142         } else {
143             cout<<"Pilihan tidak valid. Silakan coba lagi"<<endl;
144         }
145     } while (choice != 5);
146
147     return 0;
148 }

```

2. Screenshot Output

```
Masukkan Jumlah Table Size : 10
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: 1
Masukkan kunci yang akan disisipkan: 12
-> Kunci 12 berhasil disisipkan di indeks 2.
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: 1
Masukkan kunci yang akan disisipkan: 4
-> Kunci 4 berhasil disisipkan di indeks 4.
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: 1
Masukkan kunci yang akan disisipkan: 56
-> Kunci 56 berhasil disisipkan di indeks 6.
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: 1
Masukkan kunci yang akan disisipkan: 44
-> Kunci 44 disisipkan di indeks 9 (setelah 1 kali probe).
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
```

```

Pilihan Anda: 1
Masukkan kunci yang akan disisipkan: 31
-> Kunci 31 berhasil disisipkan di indeks 1.
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: 1
Masukkan kunci yang akan disisipkan: 26
-> Kunci 26 disisipkan di indeks 8 (setelah 1 kali probe).
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: 4

===== ISI TABEL HASH =====
Indeks 0:      [ Kosong ]
Indeks 1:      31
Indeks 2:      12
Indeks 3:      [ Kosong ]
Indeks 4:      4
Indeks 5:      [ Kosong ]
Indeks 6:      56
Indeks 7:      [ Kosong ]
Indeks 8:      26
Indeks 9:      44
=====
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda:

Pilihan Anda: 2
Masukkan kunci yang akan dicari: 4
-> Kunci 4 ditemukan di indeks 4.
1. Sisipkan Kunci (Insert)

```

```

Pilihan Anda: 3
Masukkan kunci yang akan dihapus: 4
-> Kunci 4 di indeks 4 berhasil dihapus.
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: 4

===== ISI TABEL HASH =====
Indeks 0:      [ Kosong ]
Indeks 1:      31
Indeks 2:      12
Indeks 3:      [ Kosong ]
Indeks 4:      [ Dihapus ]
Indeks 5:      [ Kosong ]
Indeks 6:      56
Indeks 7:      [ Kosong ]
Indeks 8:      26
Indeks 9:      44
=====
1. Sisipkan Kunci (Insert)
2. Cari Kunci (Search)
3. Hapus Kunci (Delete)
4. Tampilkan Tabel Hash
5. Selesai
Pilihan Anda: _
Pilihan Anda: 5
Program Selesai

```

3. Screenshot Jam Pengerjaan

01.18
07/07/2025 2