Muhammad Aufa Dhiya'uddin 25091397052
Struktur Data 2025-B
Pekan-3 : Penjelasan mengenai code yang diberikan

Program ini adalah implementasi visual dari **Hash Table** dengan metode **Linear Probing** untuk menangani collision. Hash table adalah struktur data yang menyimpan data dalam bentuk pasangan key-value, di mana posisi penyimpanan ditentukan oleh fungsi hash. Program dibuat dengan Python dan matplotlib, dilengkapi kontrol keyboard untuk menjalankan simulasi langkah demi langkah.

A. ANALOGI PARKIRAN MOBIL

Bayangkan sebuah **gedung parkir** dengan 20 slot (indeks 0-19). Setiap mobil yang masuk harus parkir di slot tertentu.

1. Petugas Parkir (Fungsi Hash)

Petugas menentukan slot parkir berdasarkan plat nomor:

text

slot = hash(plat_nomor) % jumlah_slot

2. Proses Parkir (Penyisipan Data)

- Mobil "B 1234 AB" → dihitung → parkir di slot 5
- Mobil "B 5678 CD" → dihitung → parkir di slot 8

3. Tabrakan (Collision)

Masalah muncul jika dua mobil diarahkan ke slot yang sama:

- Mobil A (B 1234 AB) → slot 5
- Mobil B (D 9876 EF) → slot 5 juga! (padahal sudah terisi)

4. Solusi Linear Probing

Aturannya: "Jika slot tujuan sudah terisi, coba slot berikutnya sampai dapat kosong."

Mobil B yang seharusnya di slot 5, karena sudah terisi, maka dicoba slot 6. Jika slot 6 juga terisi, coba slot 7, dan seterusnya. Jika sampai slot 19 penuh, lanjut ke slot 0, 1, 2 (sistem putaran).

5. Istilah Penting:

- **Start**: Slot awal dari hasil hash
- **Probes**: Jumlah percobaan sampai dapat slot kosong
- **Load Factor**: Rasio slot terisi / total slot

B. CARA KERJA PROGRAM

1. Alur Program:

1. Program membuat key "k0", "k1", "k2"... lalu mengacaknya
2. Setiap key dimasukkan ke hash table ukuran 20 slot

3. Jika slot tujuan kosong → langsung ditempatkan
4. Jika slot tujuan terisi → terjadi collision
5. Terapkan linear probing: cek slot berikutnya berurutan
6. Setiap langkah direkam sebagai frame untuk animasi
7. Load factor dihitung setelah setiap penyisipan

2. Visualisasi Tampilan:

- **Grid 20 kotak**: Mewakili slot hash table
- **Warna kotak**:
    - Biru muda: slot hasil hash (start)
    - Biru tua: slot yang sedang diproses
    - Merah: terjadi collision
- **Informasi teks**: Menampilkan key, start, idx, probes, fase
- **Grafik load factor**: Menunjukkan perkembangan keterisian

3. Kontrol Keyboard:

Tombol	Fungsi
SPACE	Menjeda/melanjutkan
→	Maju satu langkah (saat pause)
←	Mundur satu langkah (saat pause)
R	Mulai dari awal
ESC / Q	Keluar

C. CONTOH SIMULASI SEDERHANA

Setup: Ukuran table = 5 slot, Keys = k0, k1, k2, k3

Langkah	Key	Hash	Start	Proses	Hasil
1	k0	3	3	Kosong	k0 di slot 3

Langkah	Key	Hash	Start	Proses	Hasil
2	k1	4	4	Kosong	k1 di slot 4
3	k2	3	3	Collision! Cek 4 (isi), cek 0 (kosong)	k2 di slot 0
4	k3	4	4	Collision! Cek 0 (isi), cek 1 (kosong)	k3 di slot 1

Hasil akhir:

- Slot 0: k2
- Slot 1: k3
- Slot 2: kosong
- Slot 3: k0
- Slot 4: k1
- Load factor = 4/5 = 80%

D. KELEBIHAN & KEKURANGAN LINEAR PROBING

Kelebihan:

1. Mudah diimplementasikan dan dipahami
2. Data tersimpan berdekatan (cache-friendly)
3. Efisien untuk load factor rendah

Kekurangan:

1. **Primary Clustering**: Data mengelompok membentuk cluster
2. Performa turun drastis saat load factor > 70%

E. CARA MENJALANKAN PROGRAM

bash

Install library yang diperlukan

pip install matplotlib numpy

Jalankan program

python hash_table_premium_keyboard.py

Dengan ukuran kustom

python hash_table_premium_keyboard.py --size 12 --nkeys 11

Simpan sebagai GIF

python hash_table_premium_keyboard.py --save gif

Kesimpulan: Program ini adalah alat pembelajaran interaktif yang efektif untuk memahami konsep hash table, collision, dan linear probing melalui visualisasi yang jelas dan kontrol yang intuitif.