PENJELASAN CODE

WAHYU ADI PRATAMA
25091397067/2025B

SARJANA TERAPAN MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA

1. **Code tentang apa?**

   Hash Table Premium + (Linear Probing) — matplotlib + keyboard controls (Windows CMD safe)

2. **Gambaran Awal**

   Anggap saja kita punya area parkir yang terdiri dari sejumlah slot (seperti ember/bucket dalam hash table). Setiap mobil yang datang memiliki plat nomor (key). Tujuan kita adalah memarkir mobil di slot tertentu berdasarkan aturan yang sudah disepakati.

3. **Analogi (Dalam Parkiran Mobil)**
   - **Hash Table** = Area parkir dengan slot bernomor 0 sampai N-1.
   - **Key** (misal "k0", "k1", ...) = Plat nomor mobil.
   - **Fungsi hash** = Petugas parkir yang menghitung slot awal berdasarkan plat nomor. Misal, hash(plat) % N memberi nomor slot pertama yang harus dicoba.
   - **Linear Probing** = Jika slot yang dituju sudah terisi, mobil akan mencoba slot berikutnya (secara melingkar) sampai menemukan slot kosong.
   - **Collision** = Dua mobil ingin parkir di slot yang sama (slot penuh).
   - **Load Factor** = Rasio jumlah mobil yang sudah parkir dibanding total slot. Semakin penuh, semakin sering terjadi collision.

4. **Cara Kerja Program**

   Program ini membuat simulasi visual proses memarkir mobil satu per satu ke dalam area parkir. Setiap langkah direkam dalam frame (seperti snapshot).

   1. **Persiapan Data**
      - Sejumlah mobil (keys) diacak urutan kedatangannya.
      - Ukuran parkir (size) ditentukan, misal 20 slot.

   2. **Proses Insert dengan Linear Probing**

      Untuk setiap mobil:
      - Mulai: Hitung slot awal (start = hash(key) % size).
      - Cek slot: Jika kosong, langsung parkir. Jika terisi, terjadi collision.
      - Collision: Mobil pindah ke slot berikutnya (idx = (idx + 1) % size). Proses ini diulang sampai ketemu slot kosong.
      - Parkir: Setelah slot kosong ditemukan, mobil diparkir di situ.

      Setiap kejadian (mulai, collision, pindah, parkir, jeda) direkam sebagai frame untuk animasi. Juga dihitung load factor setelah setiap mobil berhasil parkir.

   3. **Visualisasi**
      - Tabel parkir digambar sebagai kotak-kotak (slot) dengan nomor indeks.
        - Warna gelap menandakan slot yang sedang diperiksa.
        - Warna lebih gelap saat collision.

- Informasi di atas tabel menunjukkan:
    - Frame ke berapa, mobil apa yang sedang diproses.
    - Slot awal, slot yang sedang dicek, jumlah probe (berapa kali pindah).
    - Fase (start, collision, probe, place, pause).
    - Status pause/play.
- Grafik Load Factor di bawah menunjukkan pertumbuhan kepadatan parkir.

4. **Program ini bisa dikendalikan langsung dari keyboard saat jendela gambar terbuka:**

Tombol	Fungsi
SPACE	Pause / lanjutkan animasi otomatis.
kanan	Jika sedang pause, maju satu frame (lihat langkah berikutnya).
kiri	Jika sedang pause, mundur satu frame (lihat langkah sebelumnya).
R	Mulai ulang animasi dari awal (otomatis pause).
ESC / Q	Tutup jendela dan keluar program

Dengan kontrol ini, kita bisa mengamati proses collision dan pencarian slot secara detail, langkah demi langkah.

5. **Fitur Tambahan**
    - Program bisa menerima argumen baris perintah, misal “--size 12 --nkeys 11” untuk mengubah jumlah slot dan mobil.
    - Bisa menyimpan animasi sebagai GIF dengan “--save gif” (perlu pustaka Pillow).

5. **Contoh Sederhana**

Misal parkir dengan 5 slot (indeks 0–4) dan 3 mobil: "Mobil A", "Mobil B", "Mobil C". Fungsi hash sederhana: ambil huruf pertama, ubah ke angka (A=0, B=1, C=2), lalu modulo 5.

1. **Mobil A** (hash=0): slot 0 kosong → langsung parkir.
2. **Mobil B** (hash=1): slot 1 kosong → parkir.
3. **Mobil C** (hash=2): slot 2 kosong → parkir.

Sekarang coba dengan mobil yang sama tapi urutan berbeda, misal "Mobil C" dulu, lalu "Mobil A", lalu "Mobil B":

1. **Mobil C** (hash=2): slot 2 kosong → parkir.
2. **Mobil A** (hash=0): slot 0 kosong → parkir.
3. **Mobil B** (hash=1): slot 1 kosong → parkir. Masih aman.

Sekarang tambah mobil ke-4 "Mobil D" (hash=3): slot 3 kosong → parkir. Ke-5 "Mobil E" (hash=4): slot 4 kosong → parkir. Sekarang parkir penuh (load factor = 5/5 = 100%).

Jika ada mobil baru "Mobil F" (hash=0), karena slot 0 sudah terisi, dia akan coba slot 1 (terisi), slot 2 (terisi), slot 3 (terisi), slot 4 (terisi), lalu kembali ke slot 0? Tapi karena semua penuh, tidak ada slot kosong. Dalam program ini, kita asumsikan jumlah mobil tidak melebihi kapasitas (karena kita tentukan nkeys ≤ size), jadi tidak terjadi overflow.

6. Inti Code

Kode tersebut terdiri dari dua fungsi utama:

- plan_inserts: Membuat daftar frame dengan merekam setiap langkah (mulai, collision, probe, place, pause). Ini seperti skenario film.
- main: Mengatur gambar, animasi, dan interaksi keyboard.

Dengan matplotlib, kita membuat animasi yang berjalan maju terus (otomatis) kecuali di-pause. Setiap kali tombol ditekan, fungsi on_key mengubah status (frame index, pause) dan memperbarui tampilan.

7. Kesimpulan

Program ini adalah alat bantu belajar yang sangat interaktif untuk memahami konsep hash table dengan linear probing. Analogi parkiran mobil memudahkan kita membayangkan prosesnya: mobil datang, cari slot, jika penuh geser ke slot sebelah, dan seterusnya. Visualisasi dan kontrol keyboard memungkinkan kita menyelami setiap detail kejadian, termasuk melihat bagaimana load factor meningkat seiring banyaknya mobil.

8. Source Code

Percobaan_StrukDat_Pertama.py