Nama: Izzatus Shofwa Ar Rafei

NIM: 25091397038

Kelas: 2025B

1. Gambaran Umum Program

   Program ini merupakan simulasi struktur data Hash Table menggunakan metode Linear Probing untuk menangani tabrakan (collision). Program dibuat dengan Python dan divisualisasikan menggunakan library matplotlib.

   Tujuan utama program ini adalah untuk memperlihatkan bagaimana proses penyimpanan data ke dalam hash table terjadi, mulai dari perhitungan index, terjadinya collision, proses probing, hingga perhitungan load factor. Selain itu, program dilengkapi animasi dan kontrol keyboard agar pengguna dapat melihat prosesnya secara bertahap.

2. Konsep Dasar Hash Table

   Hash table adalah struktur data yang menyimpan data dalam bentuk pasangan key dan index. Untuk menentukan posisi penyimpanan suatu key, digunakan fungsi hash dengan rumus:

   index = hash(key) % ukuran_tabel

   Rumus tersebut menghasilkan index tujuan berdasarkan ukuran tabel yang tersedia. Namun, karena beberapa key bisa saja menghasilkan index yang sama, maka dapat terjadi collision (tabrakan).

3. Linear Probing

   Program ini menggunakan metode Linear Probing untuk menangani collision. Jika index hasil hash sudah terisi, maka program akan mencari index kosong berikutnya dengan cara maju satu langkah secara berurutan:

   index = (index + 1) % ukuran_tabel

   Penggunaan operator modulus (%) bertujuan agar ketika index sudah berada di posisi terakhir, perhitungan kembali ke index 0. Proses ini dilakukan berulang sampai ditemukan bucket yang kosong.

Contoh sederhana:

Misalkan ukuran tabel adalah 5 dan suatu key menghasilkan index 2. Jika index 2 sudah terisi, maka program akan mencoba index 3, lalu 4, lalu kembali ke 0 jika diperlukan.

4. Fungsi plan_inserts()

   Fungsi plan_inserts() adalah bagian utama yang mengatur proses penyimpanan key ke dalam tabel.

   Langkah-langkah yang dilakukan fungsi ini adalah:

   a. Membuat tabel kosong dengan ukuran tertentu.
   b. Mengambil key satu per satu untuk dimasukkan ke dalam tabel.
   c. Menghitung index awal menggunakan fungsi hash.
   d. Mengecek apakah terjadi collision.
   e. Jika collision terjadi, melakukan linear probing sampai menemukan slot kosong.
   f. Menyimpan key ke dalam slot yang tersedia.
   g. Menghitung load factor setiap kali selesai melakukan insert.
   h. Menyimpan setiap tahap proses sebagai frame untuk animasi.

5. Load Factor

   Load factor adalah ukuran tingkat kepadatan hash table.

   Rumusnya adalah:

   load factor = jumlah_data_terisi / ukuran_tabel

   Nilai load factor berada antara 0 sampai 1. Semakin mendekati 1, berarti tabel semakin penuh.Jika tabel semakin penuh, kemungkinan terjadinya collision juga semakin besar.

6. Bagian Visualisasi

   Program menampilkan dua bagian utama:

   1. Bagian atas menampilkan hash table dalam bentuk kotak-kotak (bucket).
      a. Bucket yang sedang diproses akan diberi highlight.
      b. Jika terjadi collision, bucket tersebut akan ditandai secara visual.
      c. Informasi seperti index hash, jumlah probing, dan load factor juga ditampilkan.

2. Bagian bawah menampilkan grafik load factor.
    a. Grafik menunjukkan peningkatan load factor setiap insert.
    b. Terdapat indikator titik terakhir untuk menunjukkan kondisi terkini.

7. Kontrol Keyboard

   Program dapat dikendalikan menggunakan keyboard, sehingga pengguna dapat mengatur jalannya animasi.

   Fungsi tombol yang tersedia adalah:

   a. SPACE : untuk pause dan resume animasi
   b. Panah kanan : maju satu langkah saat dalam keadaan pause
   c. Panah kiri : mundur satu langkah saat dalam keadaan pause
   d. R : mengulang dari awal
   e. ESC atau Q : keluar dari program

   Fitur ini memungkinkan pengguna mempelajari proses hash table secara perlahan dan detail.