

NAMA : Muhammad alfaras Aditya rizky  
KELAS : 2025B  
NIM : 065

## PENJELASAN KODE SIMULASI HASH TABLE DENGAN LINEAR PROBING

### Sistem Loker Motor di Area Kampus

Bayangkan sebuah kampus menyediakan **15 loker parkir motor** yang diberi nomor dari **0 sampai 14**.

Setiap mahasiswa yang datang membawa motor tidak memilih tempat secara acak, melainkan diarahkan ke **nomor loker tertentu** berdasarkan perhitungan dari **NIM** mereka.

Perhitungan ini disebut **fungsi hash**.

**Contoh:**

- NIM **2501** → hasil perhitungan **4** → motor ditempatkan di **Loker 4**
- NIM **2502** → hasil perhitungan **9** → motor ditempatkan di **Loker 9**

Dengan sistem ini, petugas parkir tidak perlu mengecek semua loker satu per satu. Ia langsung tahu ke mana harus mengarahkan motor. Inilah kelebihan hash table: **akses cepat dan efisien**.

---

### Collision dan Linear Probing

Permasalahan muncul ketika **dua NIM menghasilkan nomor loker yang sama**.

Misalnya:

- NIM **2510** juga menghasilkan angka **4**

Namun, **Loker 4 sudah terisi** oleh motor sebelumnya. Keadaan ini disebut **collision** (tabrakan data).

### Cara Mengatasinya: Linear Probing

Jika terjadi collision, sistem akan:

1. Mengecek **loker berikutnya (5)**
2. Jika masih penuh, lanjut ke **loker 6**
3. Terus bergeser ke depan hingga menemukan loker kosong

Apabila sudah mencapai **loker terakhir (14)** dan belum ada yang kosong, pencarian akan **kembali ke loker 0** dan berlanjut secara melingkar.

---

### Contoh Proses Penempatan Motor

NIM	Slot Awal	Slot Akhir	Jumlah Probe	Keterangan
2501	4	4	0	Langsung masuk
2502	9	9	0	Langsung masuk
2510	4	5	1	Bertabrakan dengan 2501
2520	5	6	1	Bertabrakan dengan 2510
2530	4	7	3	Melewati 3 loker penuh

Mahasiswa dengan NIM **2530** harus melewati beberapa loker yang sudah terisi sebelum menemukan tempat kosong.

---

## **Clustering**

Kondisi ketika **banyak loker berdekatan terisi penuh** disebut **clustering**.

Akibatnya, proses pencarian tempat menjadi lebih lama karena sistem harus melakukan probing berulang kali.