





# ANGGOTA KELOMPOK :

1. Feros Pedrosa Valentino (103112400055)
2. Felix Pedrosa Valentino (103112400056)
3. Shafa Farel Azka (103112400259)





# JUDUL & TEMA

1. Judul: Sistem Reservasi Hotel "Tilem Suites"
2. Tema: Implementasi Struktur Data Multi Linked List (MLL)
3. Jenis Relasi: One-to-Many (1-to-N). Satu Kamar bisa memiliki banyak riwayat tamu.



# DESKRIPSI & TUJUAN

1. Studi Kasus: Manajemen data reservasi hotel yang mencakup data kamar dan tamu.
2. Masalah: Bagaimana menyimpan data kamar yang statis namun memiliki riwayat tamu yang dinamis (bertambah/berkurang).
3. Solusi: Menggunakan struktur data Multi Linked List (MLL) relasi 1-to-N.
4. Tujuan:
  - a. Efisiensi manajemen memori.
  - b. Mengorganisir hubungan hierarkis antara Kamar (Induk) dan Tamu (Anak).



# KONSEP STRUKTUR DATA (MLL)



## JENIS RELASI

1-to-N (Satu Kamar memiliki Banyak Tamu).



## PARENT (INDUK)

1. Representasi: Data Kamar (Nomor, Tipe, Harga).
2. Struktur: Single Linked List.
3. Pengurutan: Ascending (Kecil ke Besar berdasarkan Nomor Kamar).



## CHILD (ANAK)

1. Representasi: Data Tamu (Nama, Tanggal Check-In).
2. Struktur: Single Linked List yang "menempel" pada setiap Node Parent.
3. Sifat: Dinamis (Insert Last).



# IMPLEMENTASI PARENT (DATA KAMAR)



## STRUCT NODEKAMAR

1. int nomorKamar (Key)
2. string tipe, long harga, string fasilitas[ ]
3. NodeTamu\* tamuHead → Pointer penghubung ke Child
4. NodeKamar\* next → Pointer ke kamar berikutnya



## ALGORITMA INSERTION: INSERT SORTED

1. Data kamar tidak dimasukkan sembarangan.
2. Node baru disisipkan secara terurut berdasarkan Nomor Kamar (Ascending: 101 - > 102 -> 201).
3. Memastikan denah hotel selalu rapi saat ditampilkan.



## VALIDASI

Mencegah duplikasi nomor kamar saat penginputan.



# IMPLEMENTASI CHILD (DATA TAMU)



## STRUCT NODETAMU

1. string namaTamu
2. string tanggalCheckIn
3. NodeTamu\* next →  
*Pointer ke tamu berikutnya dalam satu kamar*



## ALGORITMA INSERTION: INSERT LAST

Tamu baru ditambahkan di urutan paling belakang dari list tamu pada kamar tersebut.



## VALIDASI

1. Sistem mengecek ketersediaan kamar.
2. Mengecek apakah tanggal check-in bentrok dengan tamu yang sudah ada (berdasarkan logika).



# ALGORITMA UTAMA - PARENT (KAMAR)



## CREATE KAMAR (INSERT SORTED)

1. Data kamar dimasukkan secara terurut agar denah hotel rapi.
2. Logika: Cek apakah nomor kamar < Head, atau cari posisi di tengah list.

Catatan : Untuk menambah kamar harus secara manual melalui code program (tidak ada menu create kamar di program)



## DELETE KAMAR

1. Menghapus node kamar dari list.

Catatan: Menerapkan pencegahan Memory Leak dengan menghapus seluruh data Tamu (Child) di dalamnya terlebih dahulu sebelum menghapus Kamar.



# ALGORITMA UTAMA - CHILD (TAMU)



## CHECK-IN (INSERT LAST)

Menambahkan tamu baru ke antrian paling belakang pada kamar tertentu.



## VALIDASI RESERVASI

1. Sistem mengecek apakah tanggal check-in bentrok dengan tamu lain di kamar yang sama.
2. Rule: Satu kamar hanya bisa diisi satu tamu. Jika ada tamu lain yang ingin mengisi kamar tersebut, kita harus check out tamu yang ada di kamar tersebut



## CHECK-OUT (DELETE BY VALUE)

Mencari nama tamu tertentu dalam list child, lalu menghapus node tersebut (update pointer next).



# TRAVERSAL & PENCARIAN



## NESTED TRAVERSAL (LOOP BERSARANG)

1. Digunakan pada fitur viewDenahHotel.
2. Loop Luar: Menelusuri setiap Kamar.
3. Loop Dalam: Menelusuri setiap Tamu di dalam kamar tersebut.



## PENCARIAN LINEAR

1. Digunakan untuk mencari posisi tamu (fitur "Cari Tamu").
2. Menelusuri seluruh kamar dan seluruh tamu untuk menemukan nama yang dicari.





# FITUR PROGRAM

1. View Denah Hotel: Menampilkan status seluruh kamar beserta tamu yang sedang menginap (Nested Traversal).
2. Check-In & Check-Out: Manajemen tamu masuk dan keluar.
3. Pencarian Tamu: Mencari lokasi kamar tamu berdasarkan nama (Search Global).
4. Pindah Kamar: Memindahkan data tamu dari satu kamar ke kamar lain (Gabungan Delete & Insert).
5. Edit Data: Mengubah nama tamu jika terjadi kesalahan input.
6. Rekapitulasi: Statistik total kamar, harga termurah, dan kamar terlaris.





# KESIMPULAN

1. Fleksibilitas: MLL memungkinkan penambahan data tamu tanpa batas (selama memori tersedia) tanpa perlu mengubah ukuran array statis.
2. Penggunaan Insert Sorted pada Parent memudahkan pencarian dan keteraturan pada data.
3. Struktur data dinamis memungkinkan penambahan jumlah tamu dan kamar tanpa batas (selama memori tersedia).







**THANK YOU!**