

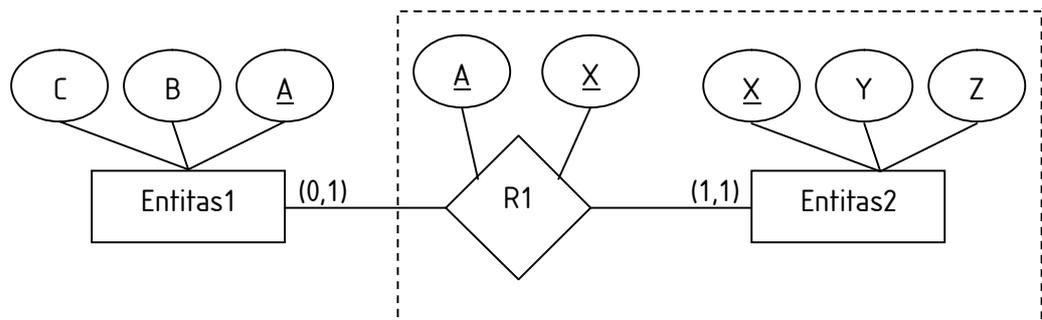
Pertemuan	: 12 dan 13
Pokok Bahasan	: Konversi dari diagram ER ke record logika dan menetapkan ke relasi tabel Setelah mempelajari pokok bahasan ini mahasiswa diharapkan dapat
TIU	: pengetahuan tentang teknik konversi dari diagram ER ke struktur record logika Setelah mempelajari pokok bahasan ini mahasiswa diharapkan dapat :
TIK	: 1. Mengkonversi diagram_ER ke struktur record logika (1 : 1) 2. Mengkonversi diagram_ER ke struktur record logika (1 : N) 3. Mengkonversi diagram_ER ke struktur record logika (N : M) 4. Menjelaskan definisi relasi Tabel 5. Memahami teknik konversi 6. Melakukan konversi dari record logika ke relasi tabel 7. Melakukan penentuan struktur penyimpanan

## 1. Konversi diagram ER ke relasi tabel

### a. Teknik konversi untuk kardinalitas 1:1

-  Penambahan atribut-atribut relasi ke salah satu tabel dari kedua himpunan entitas.
-  Pemilihan tabel yang tepat adalah himpunan entitas dengan derajat minimum terbesar.
-  Jika derajat minimumnya sama, maka pilih tabel dengan jumlah row (baris) paling sedikit.

Contoh :



Tabel yang terbentuk 2 buah :

Tabel "Entitas1"

A	B	C

Tabel "Entitas2"

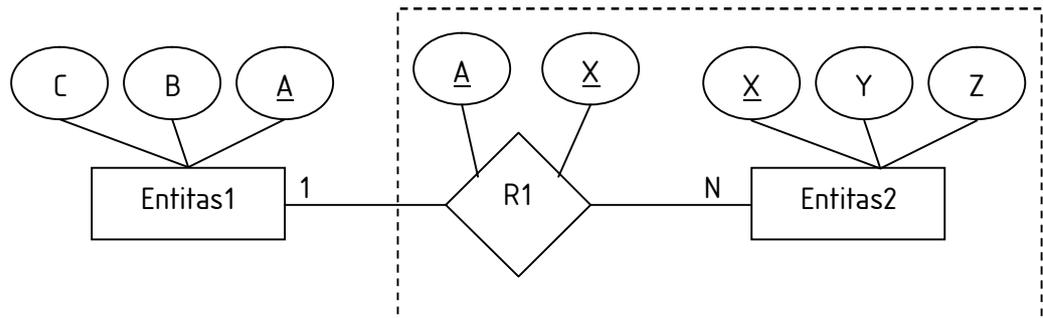
X	Y	Z	A

Ket : Karena derajat minimum pada entitas 2 lebih besar, maka seluruh atribut yang melekat (termasuk atribut tambahan jika ada) dari relasi R1 akan melekat pada tabel entitas2 dan karena atribut X sudah ada pada entitas2 maka tidak perlu dibuat dua kali dalam tabel entitas2.

b. Teknik konversi untuk kardinalitas 1:N

- ☒ Biasanya akan terbentuk 2 File (Tabel)
- ☒ Atribut-atribut dari relasi akan melekat pada salah satu tabel himpunan entitas yang derajat kardinalitas relasinya N(Banyak).

Contoh :



Tabel yang terbentuk 2 buah :

Tabel "Entitas1"

A	B	C

Tabel "Entitas2"

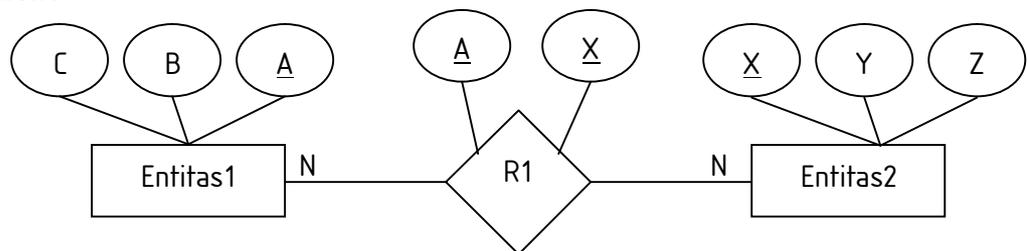
X	Y	Z	A

Ket : Karena kardinalitas pada entitas2 bernilai N (banyak), maka seluruh atribut (termasuk atribut tambahan jika ada) dari relasi R1 akan melekat pada tabel entitas2 dan karena atribut X sudah ada pada entitas2 maka tidak perlu dibuat dua kali dalam tabel entitas2.

c. Teknik konversi untuk kardinalitas N:N

- ☒ Biasanya File (tabel) yang terbentuk akan berjumlah 3 Buah.
- ☒ Relasi dari kedua himpunan entitas akan membentuk suatu tabel baru.

Contoh :



Tabel yang terbentuk 3 buah :

Tabel "Entitas1"

A	B	C

Tabel "Entitas2"

X	Y	Z

Tabel "R1"

A	X

Ket : Garis putus-putus diatas hanya untuk membantu menunjukkan arah penggabungan atribut dari relasinya, bukan bagian dari notasi diagram ER.

## Struktur Penyimpanan dan Macam-Macam Atribut

### A. Struktur Penyimpanan

- ☐ Merupakan cara penempatan (penyusunan) baris-baris data didalam tabel.
- ☐ Sangat menentukan kecepatan akses dan manipulasi data.
- ☐ Jenis struktur penyimpanan penyimpanan yang umumnya didukung oleh sebagian besar DBMS antara lain ; Heap, Hash, Sekuensial Berindeks, File Berindeks.

#### 1. Heap

- ☐ Merupakan struktur penyimpanan paling sederhana karena itu paling hemat dalam ruang penyimpanan.
- ☐ Data disimpan sesuai kedatangannya (kronologis-nya) mulai dari posisi awal ruang penyimpanan dan seterusnya.
- ☐ Bila ada penghapusan suatu record maka record-record dibawahnya akan mengisi tempat kosong tersebut.
- ☐ Pencarian data secara Sekuensial (ditelusuri dari baris pertama hingga baris terakhir) sehingga berjalan lambat.
- ☐ Struktur penyimpan Heap lebih cocok untuk tabel berukuran kecil dan jarang berubah.
- ☐ Tidak cocok untuk tabel berukuran besar atau bersifat dinamis (sering berubah) atau yang sering menjadi acuan (referensi) bagi tabel lain.

#### 2. Hash

- ☐ Merupakan struktur penyimpanan berdasarkan nilai alamat fisik dari hasil perhitungan (Hashing)
- ☐ Saat sebuah baris data disimpan DBMS akan menerapkan fungsi hashing terhadap field yang menjadi Key hasilnya berupa alamat fisik dimana baris data tersebut ditempatkan.
- ☐ Karena itu baris data tidak tersusun sesuai kedatangannya.

#### 3. Sekuensial Berindeks

- ☐ Merupakan struktur penyimpanan berdasarkan nilai indeks primer-nya.
- ☐ Baris data dengan indeks primer paling kecil akan ditempatkan di awal tabel meskipun baru terakhir disimpan
- ☐ Waktu menyimpan biasanya lebih lama (untuk memposisikan kembali record-record-nya) jika terjadi penambahan/penghapusan yang menyangkut indeks primernya.

#### 4. File Berindeks

- ☐ Merupakan pengembangan dari struktur penyimpanan Heap.
- ☐ Sama dengan Heap baris data disusun/diurutkan sesuai kedatangannya.
- ☐ Berbeda dengan Heap, tabel diberi indeks yang disusun berdasarkan nilai Key untuk mempercepat pencarian data.
- ☐ Indeks yang digunakan untuk penyusunan data dapat bervariasi dimana data yang baru masuk tidak disusun langsung tetapi hanya ditandai berdasarkan nomor indeks-nya.