

Emp\_Proj(SSN,ProjNumber, Hours, EmpName, ProjName, ProjLoc)

Emp\_Loc(EmpName, ProjLoc)

Emp\_Proj2(SSN,ProjNumber, Hours, ProjName,ProjLoc)

Emp\_Proj :

SSN01	P001	20	Ali	Proyek1	Yogya
SSN02	P001	45	Agus	Proyek1	Yogya
SSN01	P002	60	Ali	Proyek2	Solo

Emp\_Loc:

Ali	Yogya
Agus	Yogya
Ali	Solo

Emp\_Proj2:

SSN01	P001	20	Proyek1	Yogya
SSN02	P001	45	Proyek1	Yogya
SSN01	P002	60	Proyek2	Solo

Join Emp\_Loc dgn Emp\_Proj2

Ali	<i>Yogya</i>	SSN01	P001	20	Proyek1	<i>Yogya</i>
Ali	<i>Yogya</i>	SSN02	P001	45	Proyek1	<i>Yogya</i>
Agus	<i>Yogya</i>	SSN01	P001	20	Proyek1	<i>Yogya</i>
Agus	<i>Yogya</i>	SSN02	P001	45	Proyek1	<i>Yogya</i>
Ali	<i>Solo</i>	SSN01	P002	60	Proyek2	<i>Solo</i>

Penyebab masalah di atas adalah pemecahan yang tidak mempertimbangkan adanya FUNCTIONAL DEPENDENCY antara kolom-kolom yang ada.

Apakah Functional Dependency (Kebergantungan Fungsional /

Secara Fungsi)?

$X \rightarrow Y$ , Y dikatakan bergantung fungsional terhadap X ( Y is functionally dependent on X) atau X dikatakan menentukan Y secara fungsional (X functionally determines Y)

jika untuk baris-baris dalam tabel bila nilai X-nya sama maka nilai Y-nya harus juga sama

Contoh dalam matematika

$Y=X^2$ , yaitu Y fungsi dari X karena Y adalah kuadrat dari X yang berarti untuk setiap nilai X berbeda, nilai Y pasti berbeda

$$X=2 \quad Y=4$$

$$X=1 \quad Y=1$$

$$X=-3 \quad Y=9$$

$$X=-2 \quad Y=4$$

$$X=5 \quad Y=25$$

$$X=2 \quad Y=4$$

Apakah  $Y \rightarrow X$ ?? TIDAK !! Knp? Karena untuk Y yang sama, ada nilai X yang berbeda (lihat untuk  $Y=4$ , da dua nilai X yaitu '2' dan '-2')

Untuk kasus bagaimana functional dependency-nya?

Emp\_Proj :

SSN01	P001	20	Ali	Proyek1	Yogya
SSN02	P001	45	Agus	Proyek1	Yogya
SSN01	P002	60	Ali	Proyek2	Solo

SSN menentukan ProjNum? TIDAK

SSN menentukan EmpName ? YA

SSN menentukan ProjLoc? TIDAK

ProjNum menentukan ProjLoc? YA

ProjNum menentukan ProjName? YA

T1(SSN, EmpName)

{(SSN01, Ali), (SSN02, Agus)}

T2(ProjNum, ProjLoc, ProjName)

{(P001, Proyek1, Yogya), (P002, Proyek2, Solo)}

T3(SSN, ProjNum, hours)

{(SSN01, P001, 20), (SSN01, P002, 60), (SSN02, P001, 45)}

Hint (himbauan) : jangan menyimpan ***detail*** data yang berasal dari dua atau lebih entitas yang berbeda ke dalam satu tabel yang sama.

Dari EMP\_PROJ terlihat bahwa ada detail data karyawan (EmpName) dan detail data proyek (ProjName, ProjLoc) yang disimpan dalam tabel yang sama.

Dari dekomposisi menjadi T1, T2 dan T3 seperti di atas, terlihat pada T1 dan T2 tidak ada campur aduk data dari entitas yang berbeda, sedang pada T3 ada data yang berasal dari entitas-entitas yg berbeda, tetapi BUKAN DETAILnya (hanya kunci untuk perelasian saja), yaitu SSN dan ProjNum.



## Normalisasi

### 1. 1NF (First Normal Form)

mensyaratkan bahwa setiap attribute harus bernilai *atomic* (tunggal, tak bisa dipecah)

DEPARTMENT(DeptName, DeptNumber, DeptMngSSN, Deptloc)

Contoh baris --> ('Research', 5, 132211619, {Yogya, Solo, Purworejo})

-->

('Research', 5, 132211619, 'Yogya')

('Research', 5, 132211619, 'Solo')

('Research', 5, 132211619, 'Purworejo')

('Publikasi', 7, 132211619, 'Purworejo')

select \* from D where upper(DeptLoc)='SOLO'

Update anomalies

Insert? Modify? Delete?

### 2. 2NF

**Syarat:**

***Semua atribut bukan kunci harus bergantung secara penuh (full functional dependency) kepada atribut kunci (candidate/primary key).***

$X \twoheadrightarrow Y$  (Y bergantung fungsional penuh) bila Y bergantung fungsional pada X dan penghilangan satu atau lebih atribut dari X menyebabkan kebergantungan tidak berlaku lagi

$\succ \langle$  ***partial functional dependency***

***Update anomalies???***

DEPARTMENT(DeptName, DeptNumber, DeptMngSSN, Deptloc)

*Dept1(DeptNum, DeptName, DeptMngSSN)*

*T1(DeptNum, DeptLoc)*

***Dept1* dan *T1* keduanya sudah memenuhi syarat normal kedua.**

### 3. 3NF (Third Normal Form)

Syarat:

atribut yang bukan kunci tidak boleh bergantung secara TRANSITIF terhadap atribut kunci.

TRANSITIVE *dependency* ( $X \twoheadrightarrow Y$  dan  $Y \twoheadrightarrow Z$  maka  $X \twoheadrightarrow Z$ )