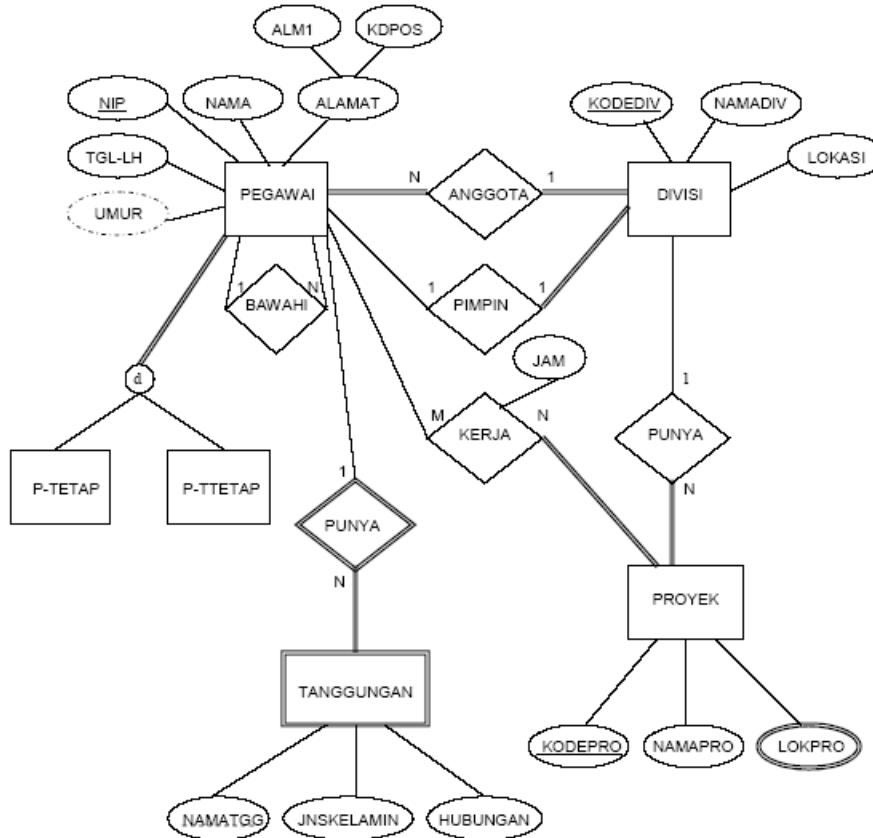


Manajemen Informasi dan Basisdata

ERD yang digunakan pada kasus transformasi:



Ketentuan transformasi ERD ke skema relasional:

1. Transformasi Entitas Kuat

Untuk mentransformasikan ke ERD ke skema relasional, pertama kali yang perlu dikerjakan adalah entitas-entitas kuat, dengan ketentuan:

Setiap **entitas kuat** ditransformasi menjadi **satu relasi/tabel** yang memiliki semua **atribut tunggal** dari entitas tersebut, dengan atribut kunci (PK) sesuai dengan yang ada di ERD.

Sedangkan untuk atribut komposit hanya dimuat komponen atributnya saja.

Contoh dari ERD di atas:

PEGAWAI (NIP, NAMA, TGL-LH, ALM1, KDPOS, UMUR....)

DIVISI (KODEDIV, NAMADIV, LOKASI....)

PROYEK (KODEPRO, NAMAPRO....)

Keterangan:

- Atribut dengan garis bawah: primary key (PK)
- Kurung jangan ditutup dulu, karena atribut mungkin masih ditambah dari hasil transformasi berikutnya.

2.Transformasi Atribut Bernilai Banyak

Setiap entitas yang mempunyai atribut bernilai banyak dibuat relasi baru dimana kunci utamanya merupakan gabungan dari kunci utama dari relasi asal dengan atribut bernilai banyak.

LOKASIPRO (KODEPRO, LOKPRO)

Keterangan:

- PK : KODEPRO+LOKPRO
- KODEPRO: FK yang bereferensi ke atribut kunci KODEPRO pada relasi PROYEK

3. Transformasi Entitas Lemah

Setiap Entitas Lemah ditransformasi menjadi relasi yang memiliki semua atribut tunggal pada entitas tersebut dan kunci utamanya adalah gabungan dari kunci parsial dan kunci utama dari induknya.

TANGGUNGAN (NIP, NAMATGG, JNSKELAMIN, HUBUNGAN)

Keterangan:

- PK : NIP+NAMATGG
- NIP: FK yang bereferensi ke atribut NIP pada relasi PEGAWAI

4. Transformasi Hubungan Entitas Berderajat Satu, Kardinalitas 1:N

Pada setiap hubungan entitas berderajat satu dengan kardinalitas satu ke banyak pada suatu entitas, maka pada relasi yang merepresentasikan entitas tersebut perlu ditambahkan kunci asing yang bereferensi ke kunci utama dari relasi itu sendiri

PEGAWAI (NIP, NAMA,TGL-LH, ALM1, KDPOS, UMUR, KODEATASAN,)

Keterangan:

KODEATASAN: FK yang bereferensi pada NIP pada Pegawai

5. Transformasi Hubungan Berderajat Satu, Kardinalitas M:N

Pada setiap hubungan berderajat satu dengan kardinalitas banyak ke banyak pada suatu entitas, perlu dibuat relasi baru dimana kunci utamanya merupakan gabungan dari kunci utama relasi asal dan kunci utama relasi asal itu sendiri yang diberi nama lain.

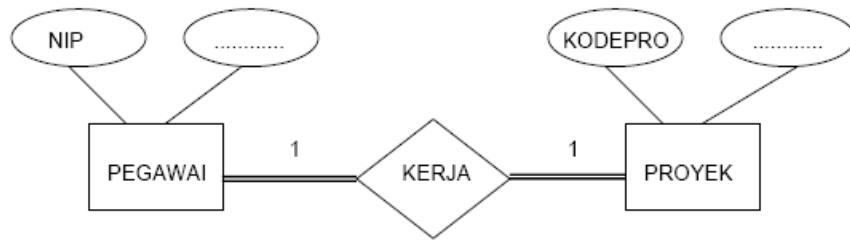
KOMPONENBRG (KODEBRG, KODEKOMP, JUMLAH)

Keterangan:

KODEKOMP: FK yang bereferensi ke KODEBRG di KOMPONENBRG

6. Transformasi Hubungan Berderajat Dua, Kardinalitas 1:1, dan Partisipasi Total

Setiap hubungan berderajat dua dengan kardinalitas satu ke satu dan kedua-duanya berpartisipasi secara total, maka perlu dibuat suatu relasi gabungan dimana kunci utamanya dapat dipilih salah satu



PEGAWAI (NIP,..., KODEPRO,...)

Keterangan:

PK: NIP atau KODEPRO

7. Transformasi Hubungan Berderajat Dua 1:1 dan Salah Satu Berpartisipasi Total

Pada setiap hubungan berderajat dua dengan kardinalitas satu ke satu dan salah satu berpartisipasi secara total, maka **pada relasi yang berpartisipasi total** mempunyai FK (kunci asing) yang berasal dari kunci utama relasi yang berpartisipasi sebagian.

DIVISI (KODEDIV,NAMADIV, LOKASI, KODEMANAGER, ...)

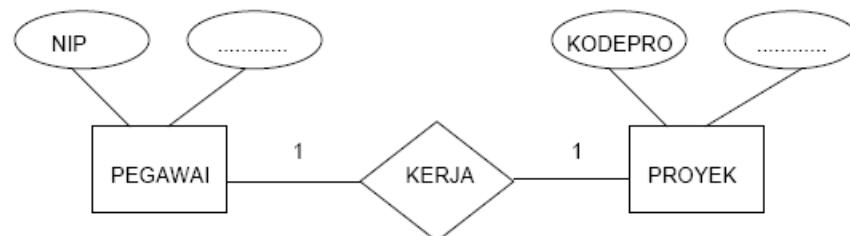
Keterangan:

KODEMANAGER: FK yang bereferensi ke NIP di relasi Pegawai

8. Transformasi Hubungan Berderajat Dua 1:1 dan Keduanya Berpartisipasi Parsial

Pada setiap hubungan berderajat dua dengan kardinalitas satu ke satu dan kedua-duanya berpartisipasi secara parsial, maka perlu dibuat suatu relasi baru dimana kunci utamanya merupakan gabungan dari kunci utama kedua relasi.

Atau: Salah satu relasi memiliki kunci asing (FK) yg bereferensi ke relasi lainnya, tapi boleh null.



PEKERJAANPRO (NIP, KODEPRO,...)

Keterangan:

- PK : NIP+KODEPRO
- NIP: FK ke NIP di Pegawai
- KODEPRO: FK ke Kodepro di Proyek

9. Transformasi Relationship Berderajat Dua 1:N dan Sisi N Berpartisipasi Total

Pada setiap hubungan berderajat dua dengan kardinalitas satu ke banyak dan pada entitas yang mempunyai kardinalitas N berpartisipasi secara total, maka pada relasi bersisi N dimasukkan kunci asing yang berasal dari kunci utama relasi bersisi 1

PROYEK (KODEPRO, NAMAPRO, KODEDIV, ...)

Keterangan:

KODEDIV: FK ke relasi Divisi

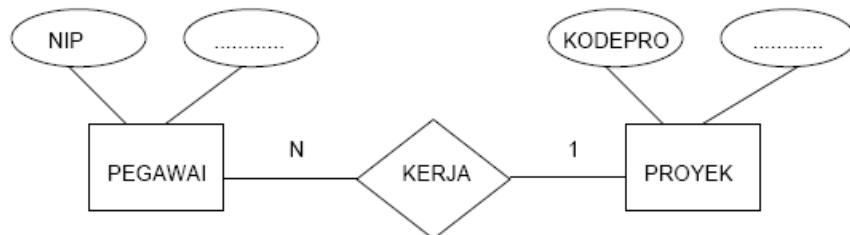
10. Transformasi Hubungan Berderajat Dua 1:N dan Sisi N Berpartisipasi Parsial

Pada setiap hubungan berderajat dua dengan kardinalitas satu ke banyak dan pada entitas yang mempunyai kardinalitas N berpartisipasi secara parsial, maka perlu dibuat suatu relasi baru dimana kunci utamanya merupakan gabungan dari kunci utama kedua relasi. Atau: Seperti no 9 tapi FK boleh null

PEKERJAANPRO (NIP, KODEPRO)

Keterangan:

- PK : NIP+KODEPRO
- FK : NIP dan KODEPRO



11. Transformasi Hubungan Berderajat Dua M:N

Pada hubungan berderajat dua dengan kardinalitas banyak ke banyak, perlu dibuat suatu relasi baru dimana kunci utamanya merupakan gabungan dari kunci utama kedua relasi.

PEKERJAANPRO (NIP, KODEPRO, ..., JAM)

Keterangan:

- PK : NIP+KODEPRO
- FK : NIP dan KODEPRO

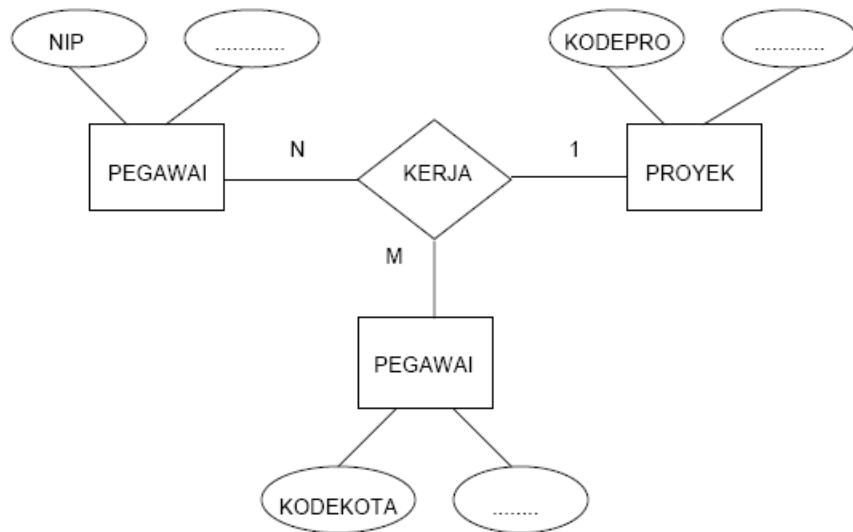
12. Transformasi Hubungan Berderajat Tiga

Pada setiap hubungan berderajat tiga, dibuat relasi baru dimana kunci utamanya merupakan gabungan dari kunci utama ketigarelasi.

PEKERJAANPRO (NIP, KODEPRO, KODEKOTA,...)

Keterangan:

- PK: NIP+KODEPRO+KODEKOTA
- FK: NIP, KODEPRO, KODEKOTA



13. Transformasi Spesialisasi atau Generalisasi

Terdapat 3 cara :

- a) Untuk superclass dan setiap subclass dibuat relasi tersendiri, dimana pada masing-masing subclass ditambahkan kunci utama dari superclass.
- b) Setiap subclass dibuat relasi tersendiri dimana pada masing-masing subclass ditambahkan semua atribut dari superclass. Kunci utama relasi adalah kunci utama dari superclass.
- c) Untuk superclass yang dispesialisasi dengan disjoint, ditransformasikan dengan membuat satu relasi tunggal yang memuat semua atribut dari superclass maupun subclass. Kunci utama relasi adalah kunci utama dari superclass.

Skema relasional Pegawai (generalisasi dgn disjoint):

PEGAWAI (NIP, NAMA, TGLLH, ALM1, KDPOS, UMUR, KODEATASAN, GAPOK, HONOR)

Skema relasional final lainnya:

DIVISI (KODEDIV, NAMADIV, LOKASI, KODEMANAGER)

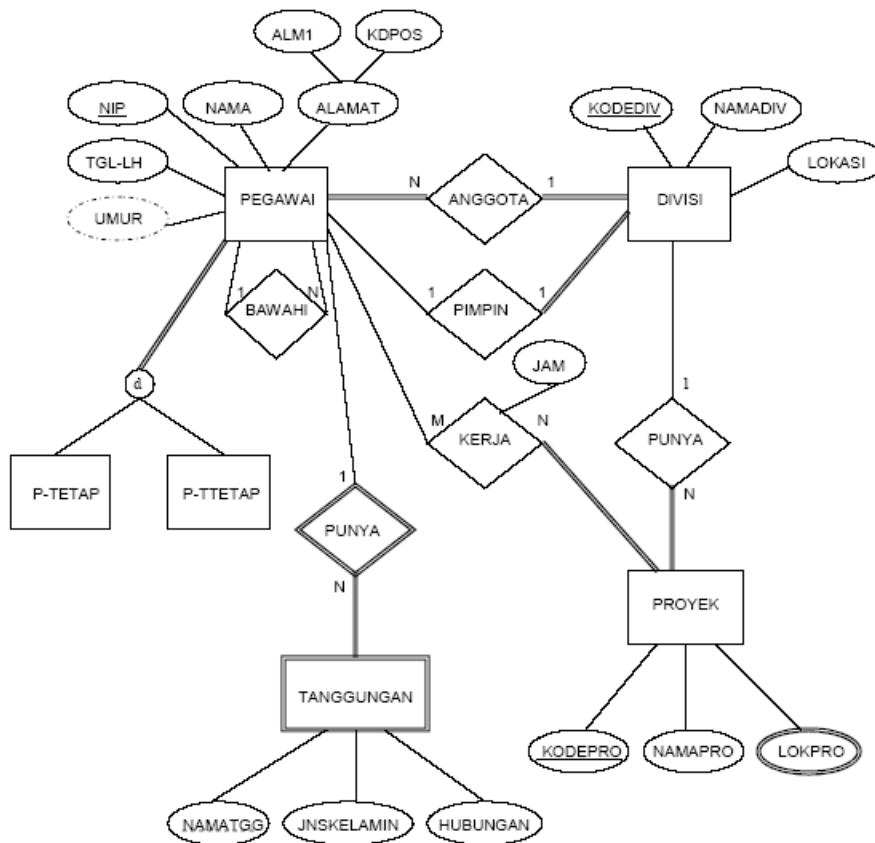
PROYEK (KODEPRO, NAMAPRO, KODEDIV)

LOKASIPRO (KODEPRO, LOKPRO)

PEKERJAANPRO (NIP, KODEPRO, JAM)

TANGGUNGAN (NIP, NAMATGG, JNSKELAMIN, HUBUNGAN)

Pada tahap akhir, perlu diperiksa/dicek lagi apakah semua entitas (kuat dan lemah), atribut dan hubungan sudah ditransformasi ditransformasi.



~~~00~~~