



Pertemuan 3 : ER Diagram

Basis Data

Agung Nugroho, M.Kom
Teknik Informatika – S1
Fakultas Teknik
Universitas Pelita Bangsa



Entity Relationship Diagram

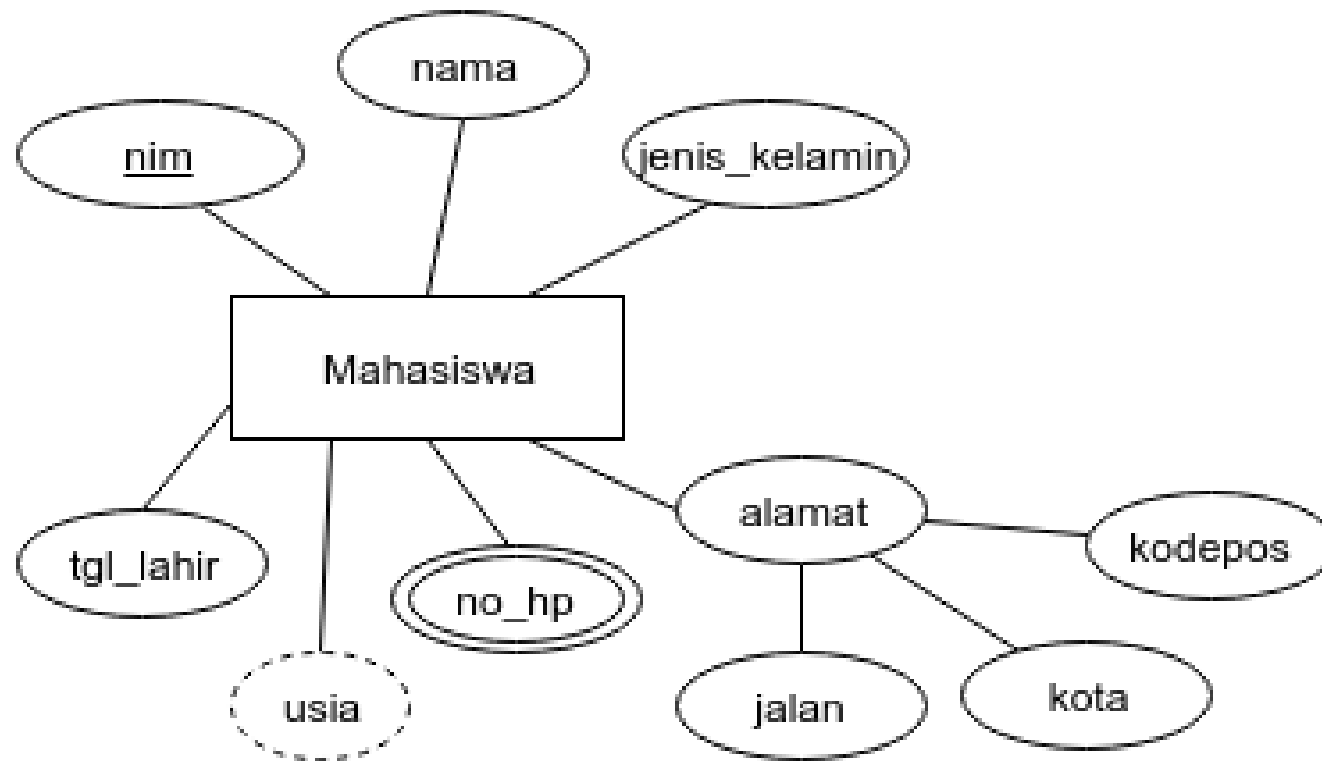
Latihan

Latihan

Level 1

- Buatlah ER-D untuk Entitas Mahasiswa dengan Atribut NIM, Nama, Alamat (Jalan, Kota, Kode Pos), Tanggal Lahir, Usia, Jenis Kelamin, No HP.

Entitas Mahasiswa dengan atributnya

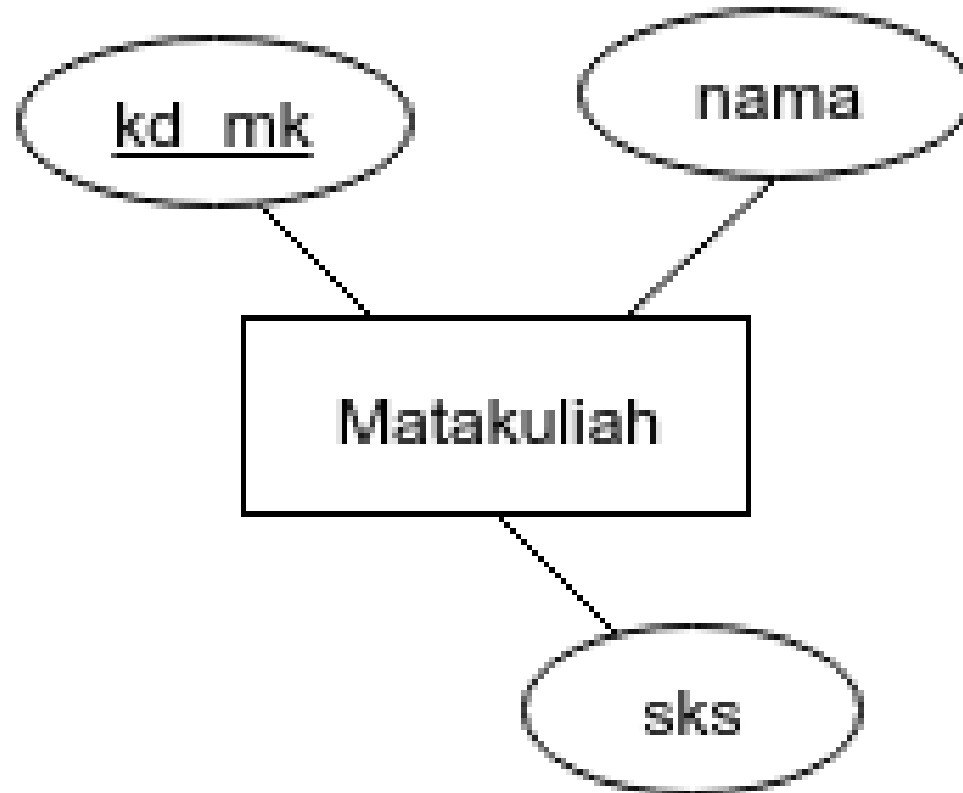


Latihan

Level 1

- Buatlah ER-D untuk Entitas Mahasiswa dengan Atribut NIM, Nama, Alamat (Jalan, Kota, Kode Pos), Tanggal Lahir, Usia, Jenis Kelamin, No HP.
- Buatlah ER-D untuk Entitas Mata Kuliah dengan atribut Kode Mata Kuliah, Nama Mata Kuliah dan SKS

Entitas Mata Kuliah dengan atributnya

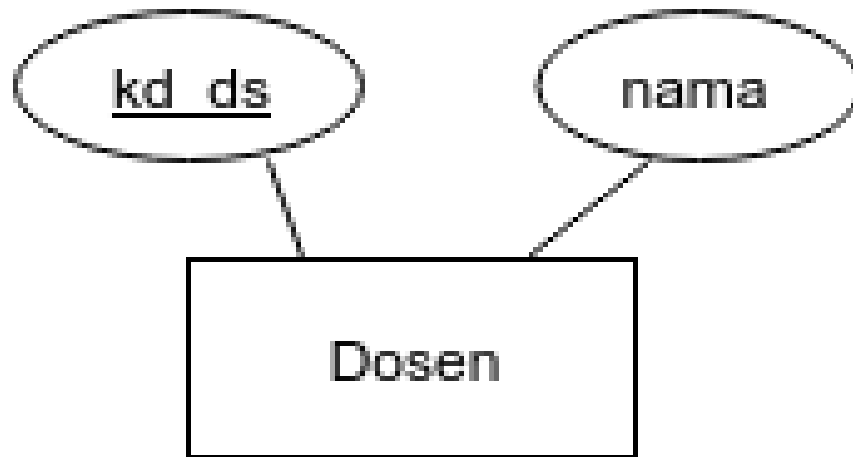


Latihan

Level 1

- Buatlah ER-D untuk Entitas Mahasiswa dengan Atribut NIM, Nama, Alamat (Jalan, Kota, Kode Pos), Tanggal Lahir, Usia, Jenis Kelamin, No HP.
- Buatlah ER-D untuk Entitas Mata Kuliah dengan atribut Kode Mata Kuliah, Nama Mata Kuliah dan SKS
- Buatlah ER-D untuk Entitas Dosen yang memiliki Kode Dosen, Nama Dosen.

Entitas Dosen dengan atributnya

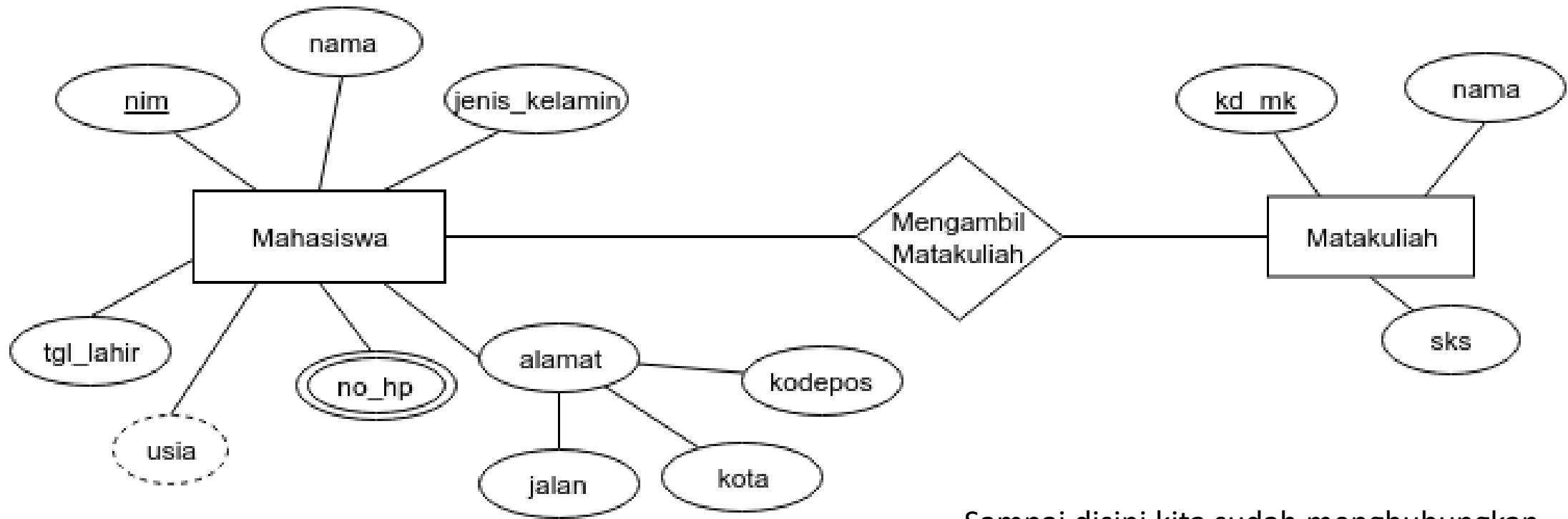


Latihan

Level 2 (Relationship)

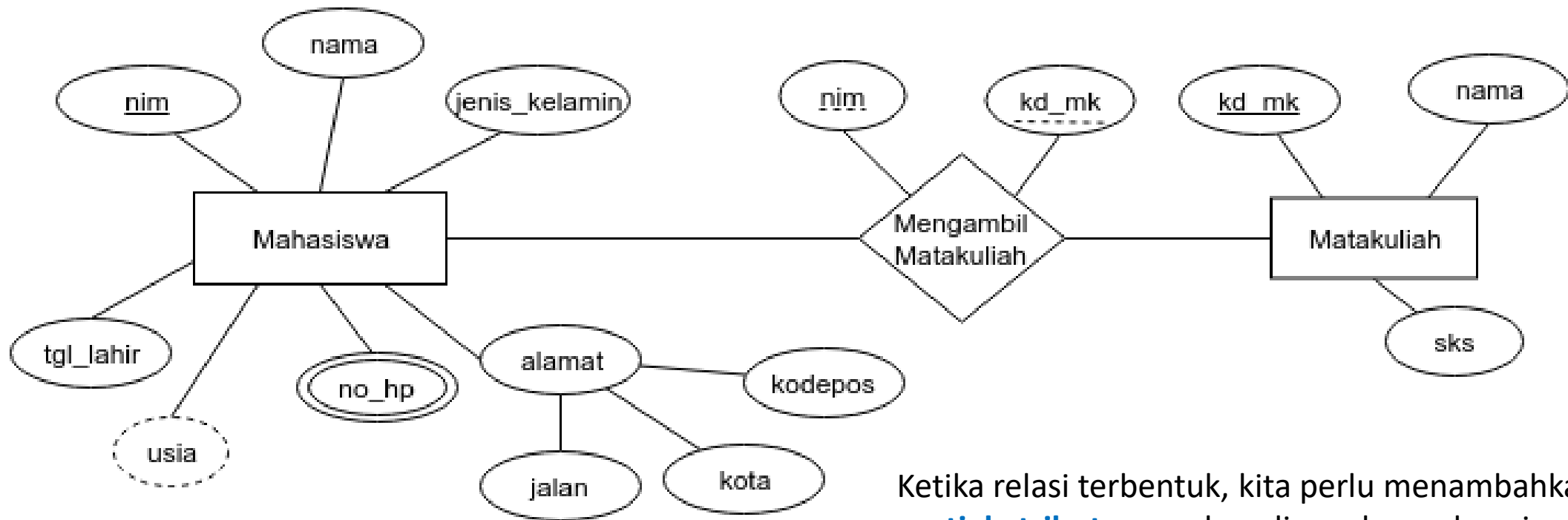
- Setiap mahasiswa mengambil mata kuliah
- Setiap mata kuliah diajar oleh Dosen
- Setiap Mahasiswa juga memiliki dosen wali untuk membantu konseling dalam perkuliahan selain mengajar mahasiswa

Setiap mahasiswa mengambil mata kuliah



Sampai disini kita sudah menghubungkan relasi antara mahasiswa dengan matakuliah, akan tetapi belum ditentukan **kardinalitasnya**.

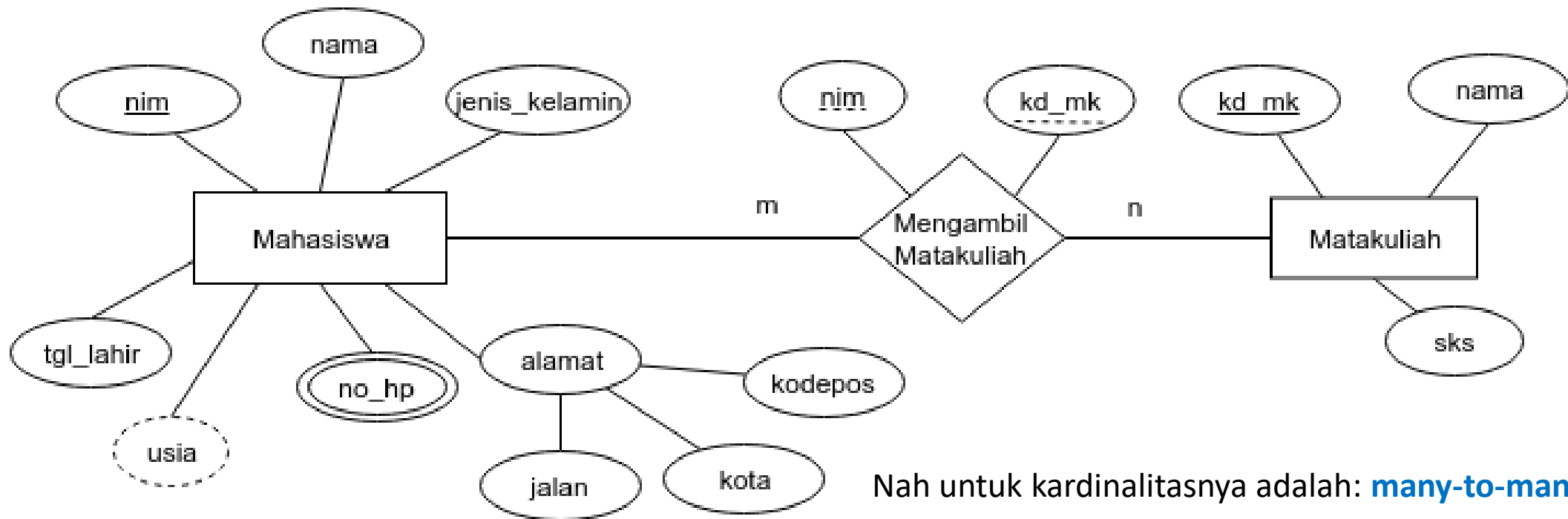
Setiap mahasiswa mengambil mata kuliah



Ketika relasi terbentuk, kita perlu menambahkan **partial atribut** yang akan digunakan sebagai penghubung relasinya, dalam hal ini nantinya akan menjadi **Foreign Key**.

partial atributnya adalah: **nim**, dan **kd_mk**

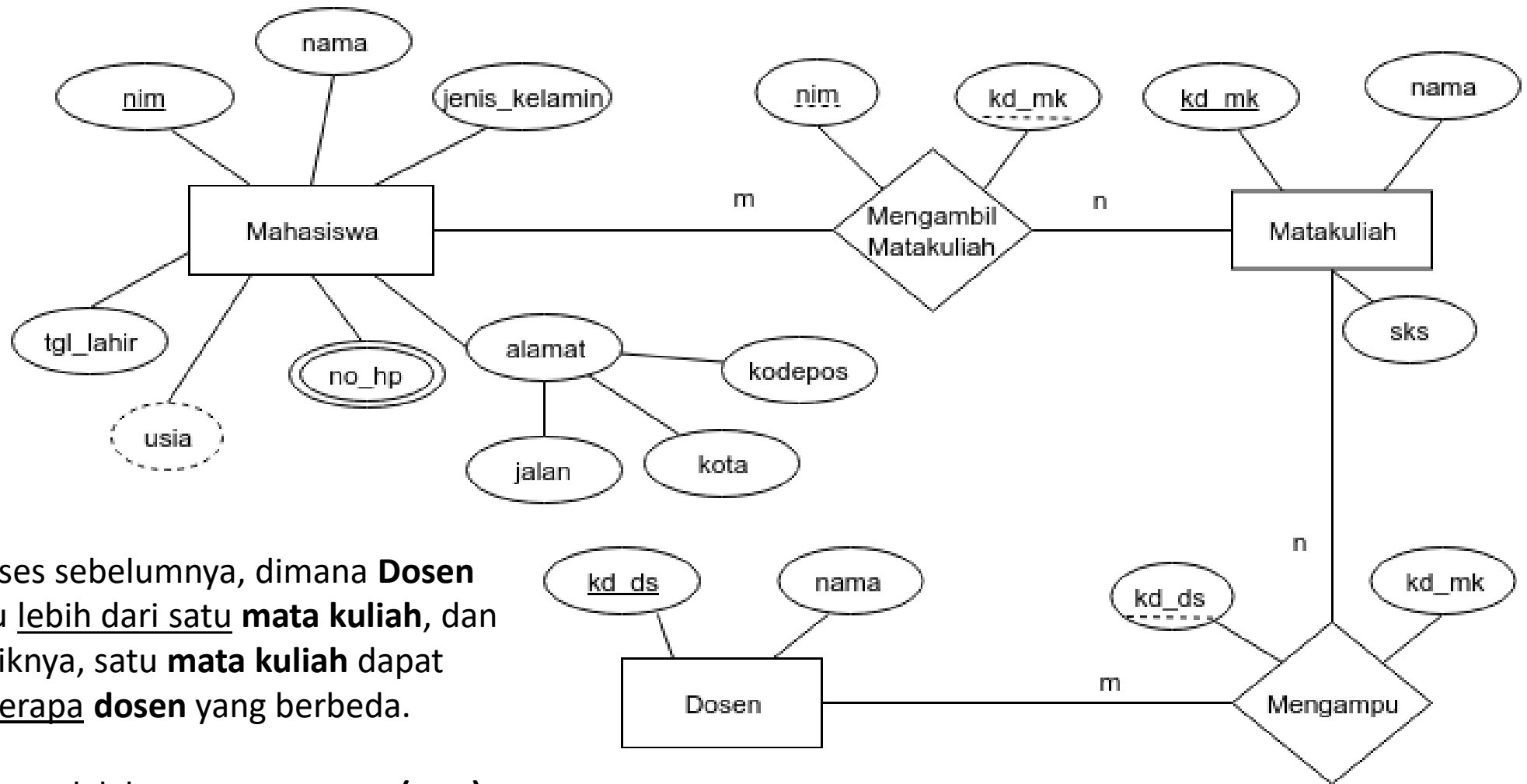
Setiap mahasiswa mengambil mata kuliah



Nah untuk kardinalitasnya adalah: **many-to-many** (m:n)

Dimana, setiap **mahasiswa** (banyak mahasiswa) mengambil **mata kuliah** (satu mahasiswa mata kuliah yang bisa diambil juga lebih dari satu)

Setiap mata kuliah diajar oleh dosen



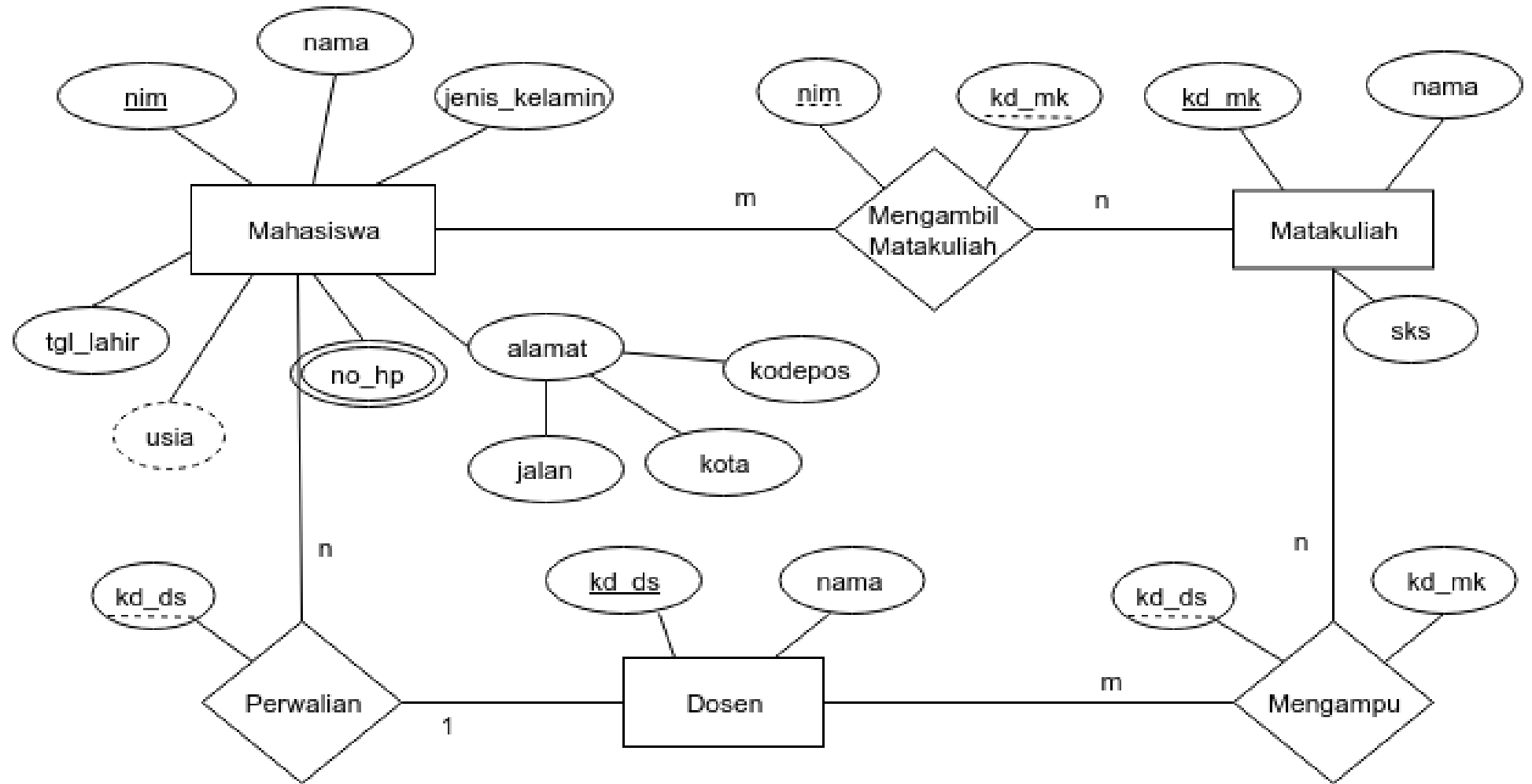
Sama seperti proses sebelumnya, dimana **Dosen** boleh mengampu lebih dari satu mata kuliah, dan begitu juga sebaliknya, satu **mata kuliah** dapat diampu oleh beberapa dosen yang berbeda.

Jadi kardinalitasnya adalah: **many-to-many (m:n)**

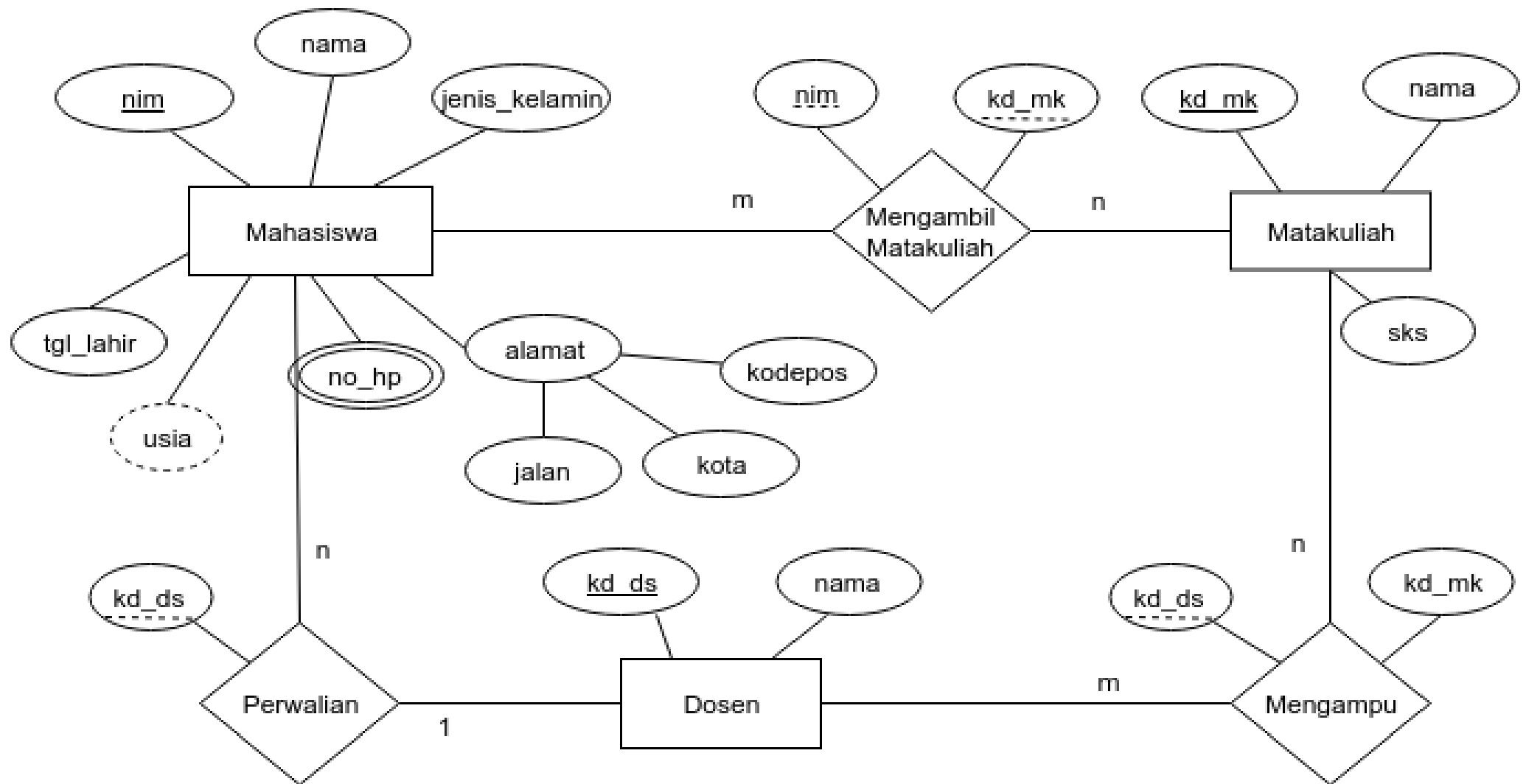
Setiap mahasiswa memiliki dosen wali

Setiap (banyak) mahasiswa memiliki satu dosen wali.

Jadi **kardinalitasnya** adalah: **many-to-one (n:1)**



Sampai disini apakah sudah selesai?



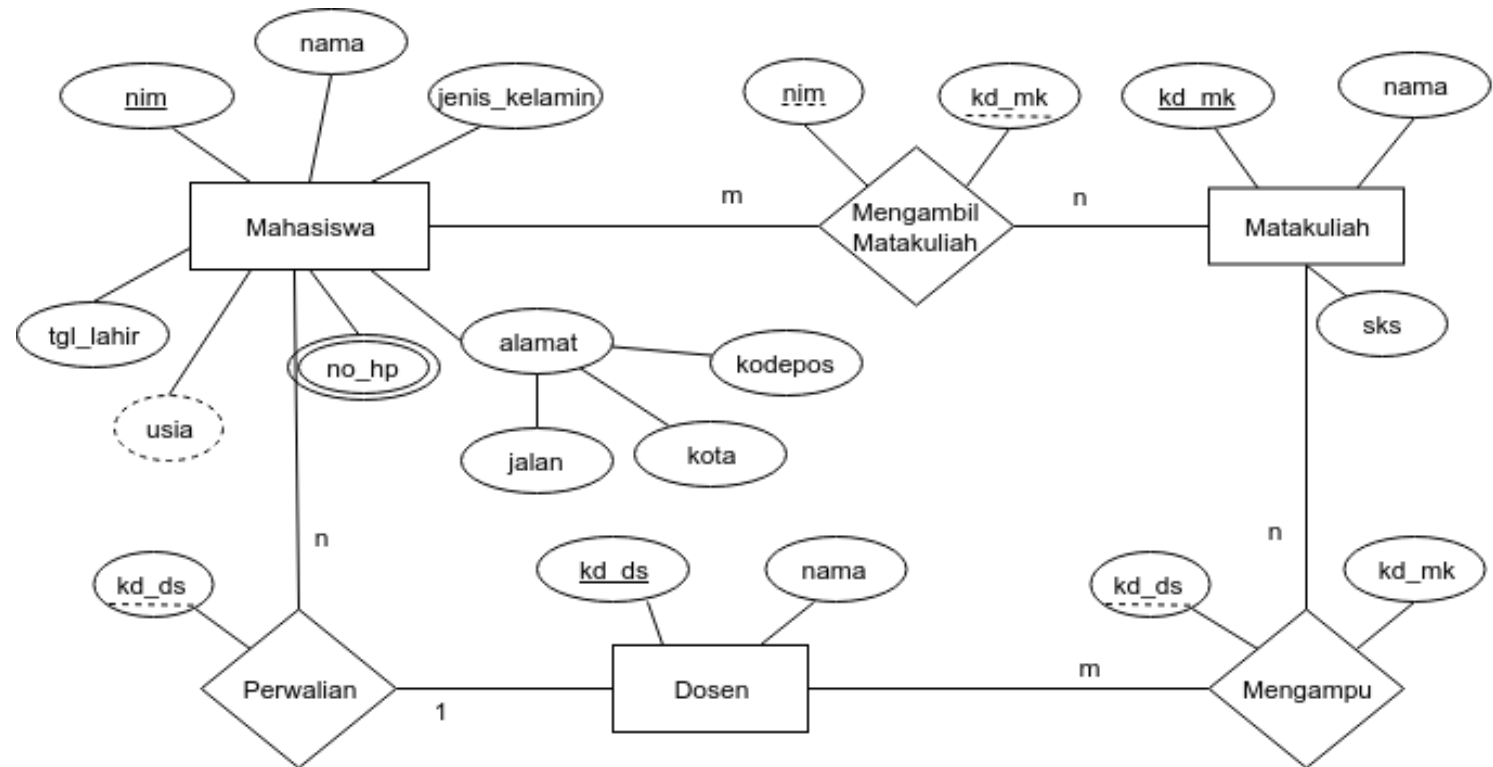
Sampai disini apakah sudah selesai?

Coba kita lihat kembali:

Apabila terjadi relasi, maka diperlukan **partial key atribut** yang seharusnya ditambahkan pada entitas yang berelasi.

Kita lihat pada relasi **Dosen** dengan **Mahasiswa**, dengan kardinalitas **one-to-many** (1:n), maka diperlukan **partial key atribut** untuk menghubungkan relasinya.

Maka, **partial key atribut** **kd_ds** perlu ditambahkan pada entitas **Mahasiswa**.



Sampai disini apakah sudah selesai?

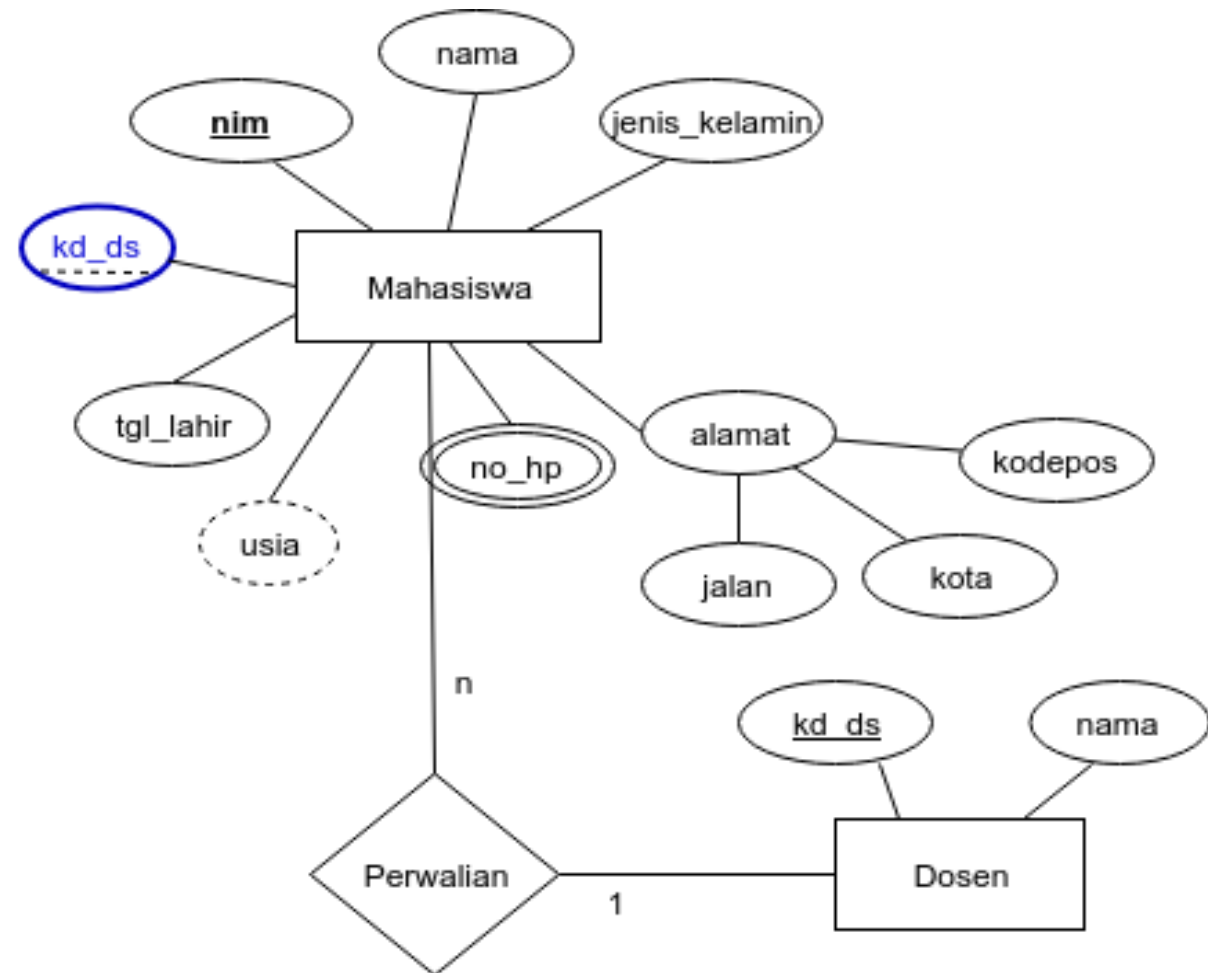
Coba kita lihat kembali:

Apabila terjadi relasi, maka diperlukan **partial key atribut** yang seharusnya ditambahkan pada entitas yang berelasi.

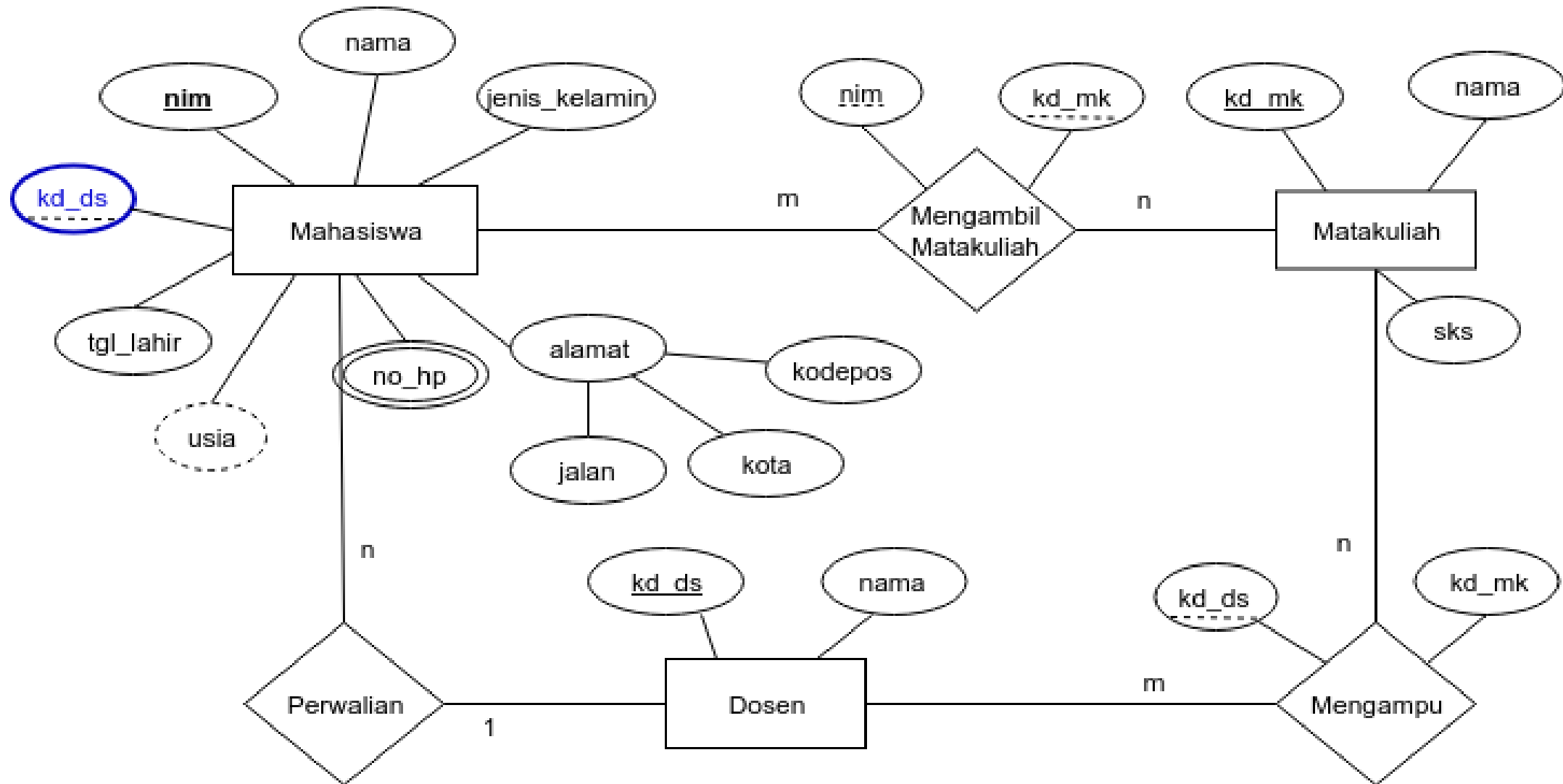
Kita lihat pada relasi **Dosen** dengan **Mahasiswa**, dengan kardinalitas **one-to-many** (1:n), maka diperlukan **partial key atribut** untuk menghubungkan relasinya.

Maka, **partial key atribut** **kd_ds** perlu ditambahkan pada entitas **Mahasiswa**.

Sehingga, hasilnya seperti berikut -->



Sampai disini apakah sudah selesai?



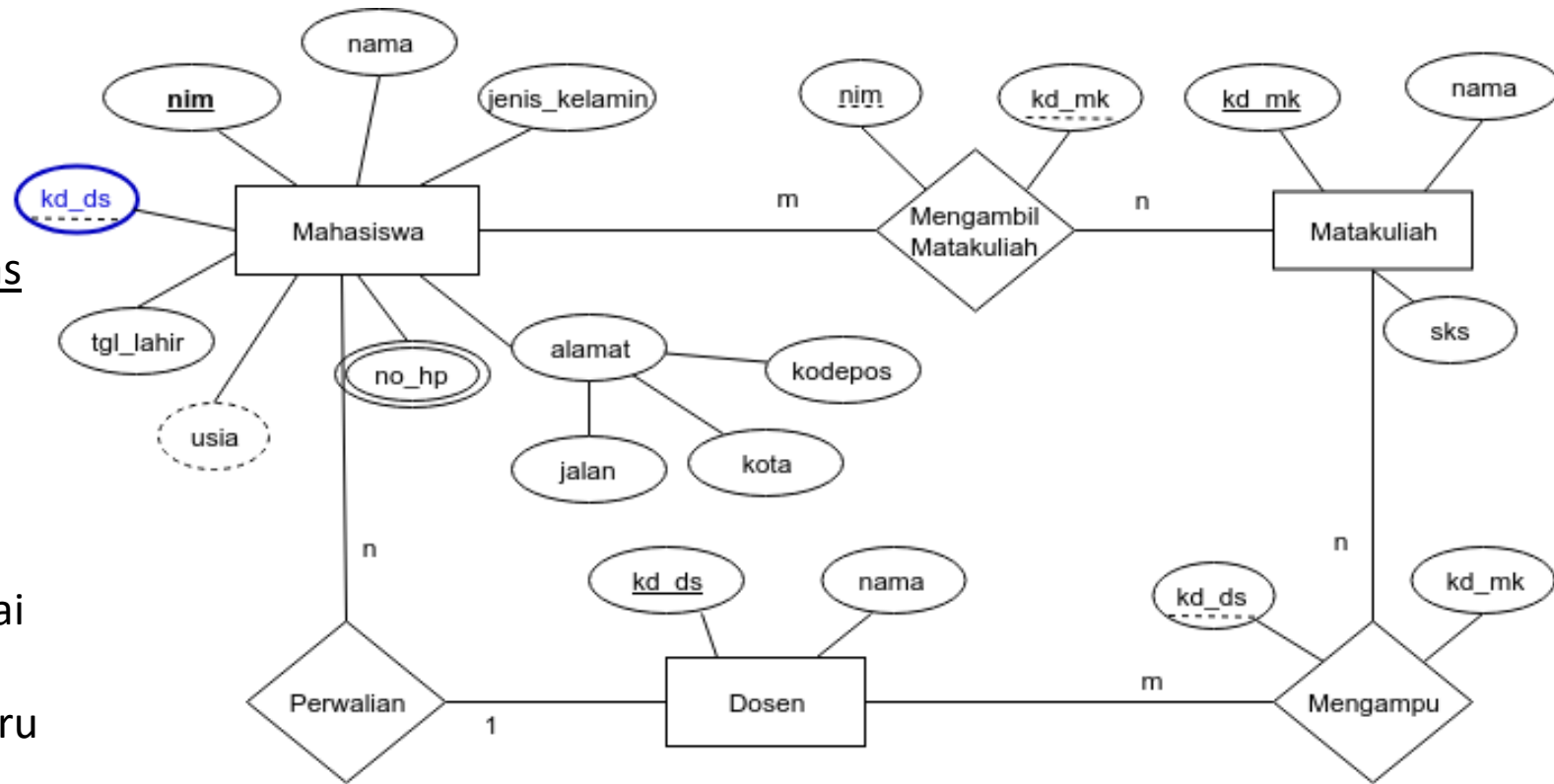
Sampai disini apakah sudah selesai?

Coba kita perhatikan kembali:

Apabila terjadi relasi **many-to-many**, maka secara otomatis akan terbentuk sebuah entitas baru yang membawa atribut **partial key**.

Itu artinya, relasi **mengambil matakuliah** dan relasi **mengampu** akan menjadi entitas baru.

Mengambil matakuliah --> kita anggap sebagai entitas baru misal **krs_mahasiswa**, dan **Mengampu** --> kita anggap sebagai entitas baru misal **jadwal_mengajar**.



Sampai disini apakah sudah selesai?

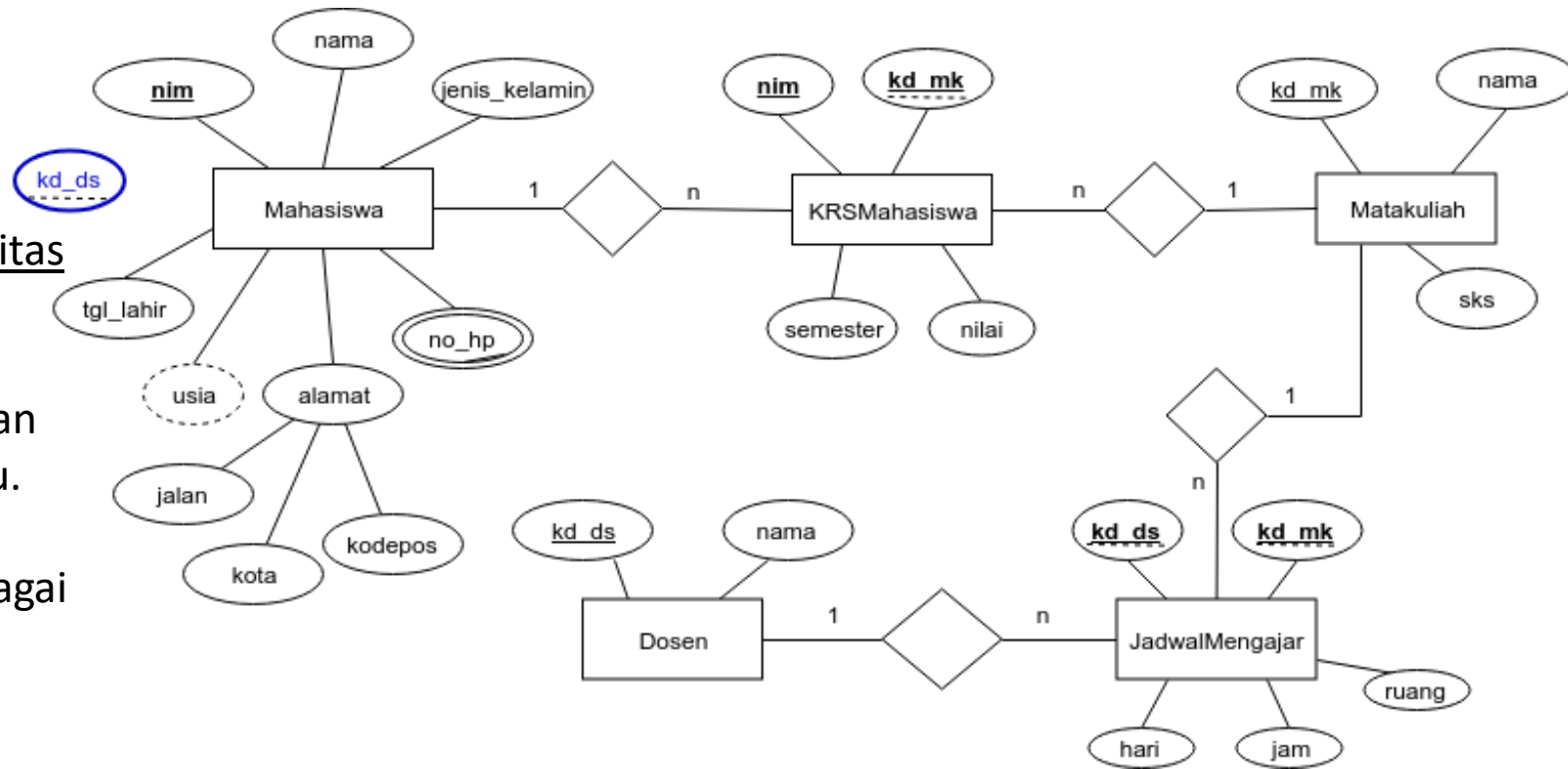
Coba kita perhatikan kembali:

Apabila terjadi relasi **many-to-many**, maka secara otomatis akan terbentuk sebuah entitas baru yang membawa atribut **partial key**.

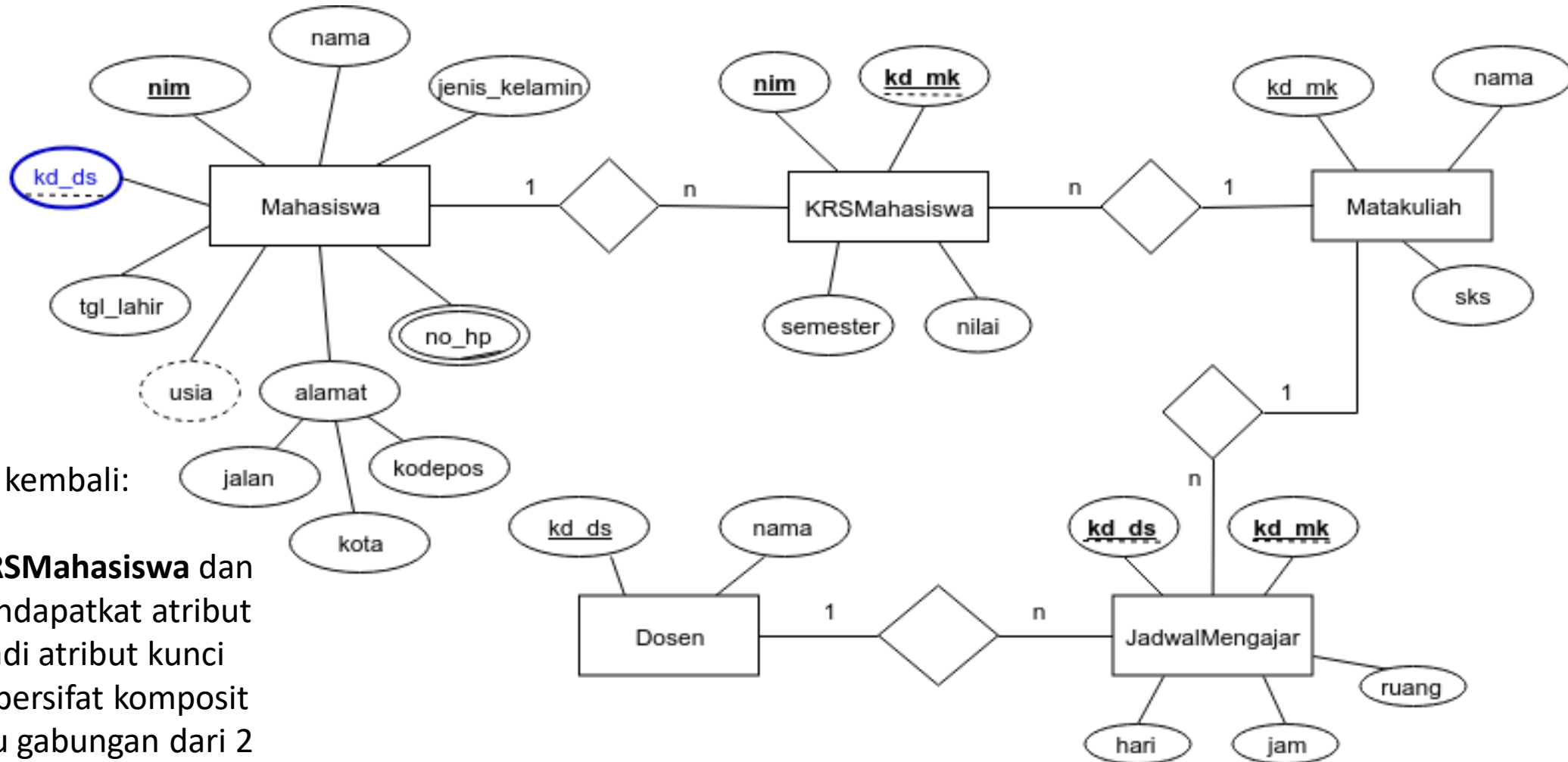
Itu artinya, relasi **mengambil matakuliah** dan relasi **mengampu** akan menjadi entitas baru.

Mengambil matakuliah --> kita anggap sebagai entitas baru misal **krs_mahasiswa**, dan **Mengampu** --> kita anggap sebagai entitas baru misal **jadwal_mengajar**.

Sehingga kita dapatkan **ER-D** baru ==>



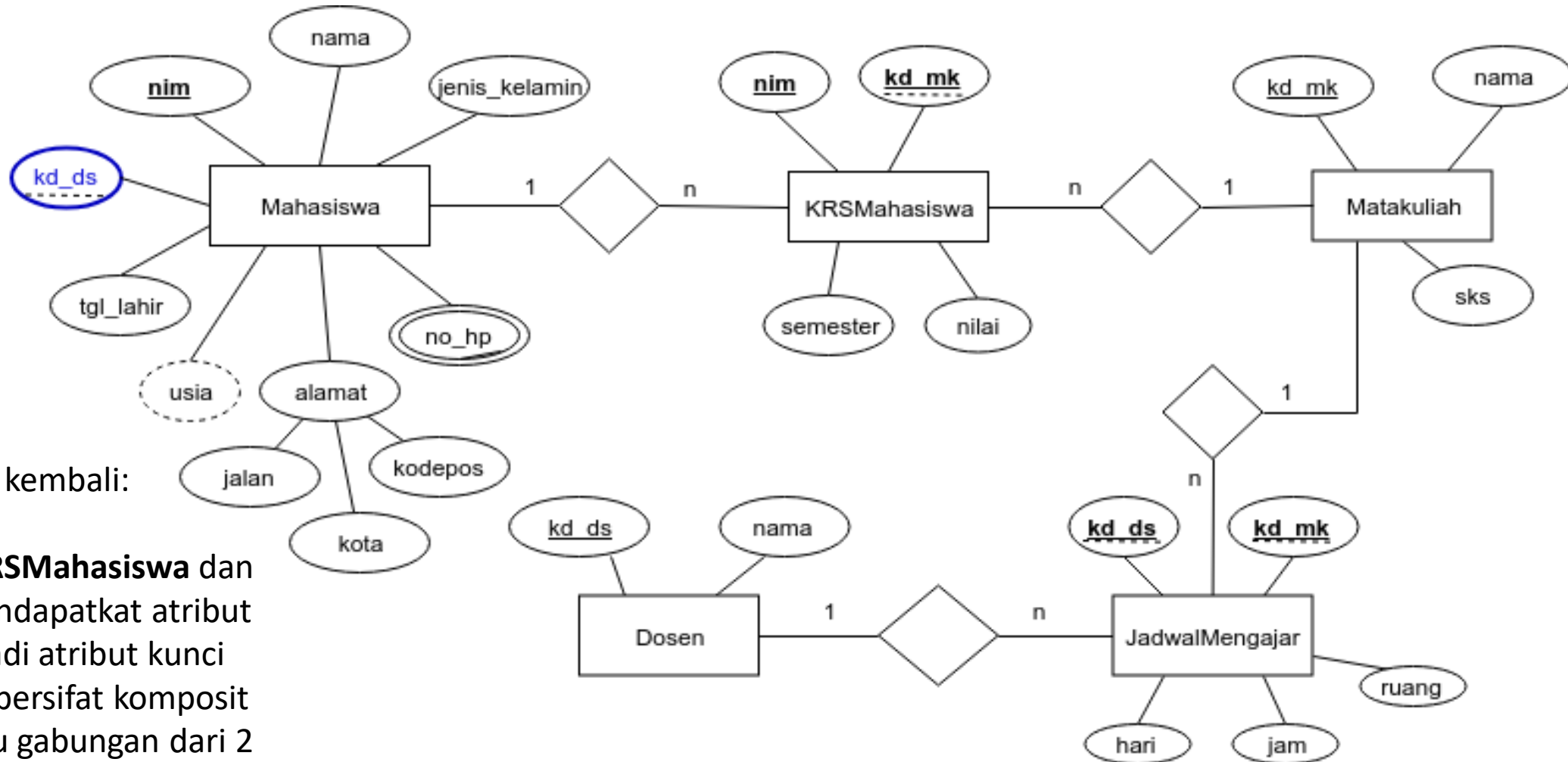
Sampai disini apakah sudah selesai?



Coba kita perhatikan kembali:

Entitas yang baru, **KRSMahasiswa** dan **JadwalMengajar** mendapat atribut baru dan yang menjadi atribut kunci (**key atribut**) adalah bersifat komposit (**composite key**), yaitu gabungan dari 2 atribut yang berasal dari **partial key atribut**.

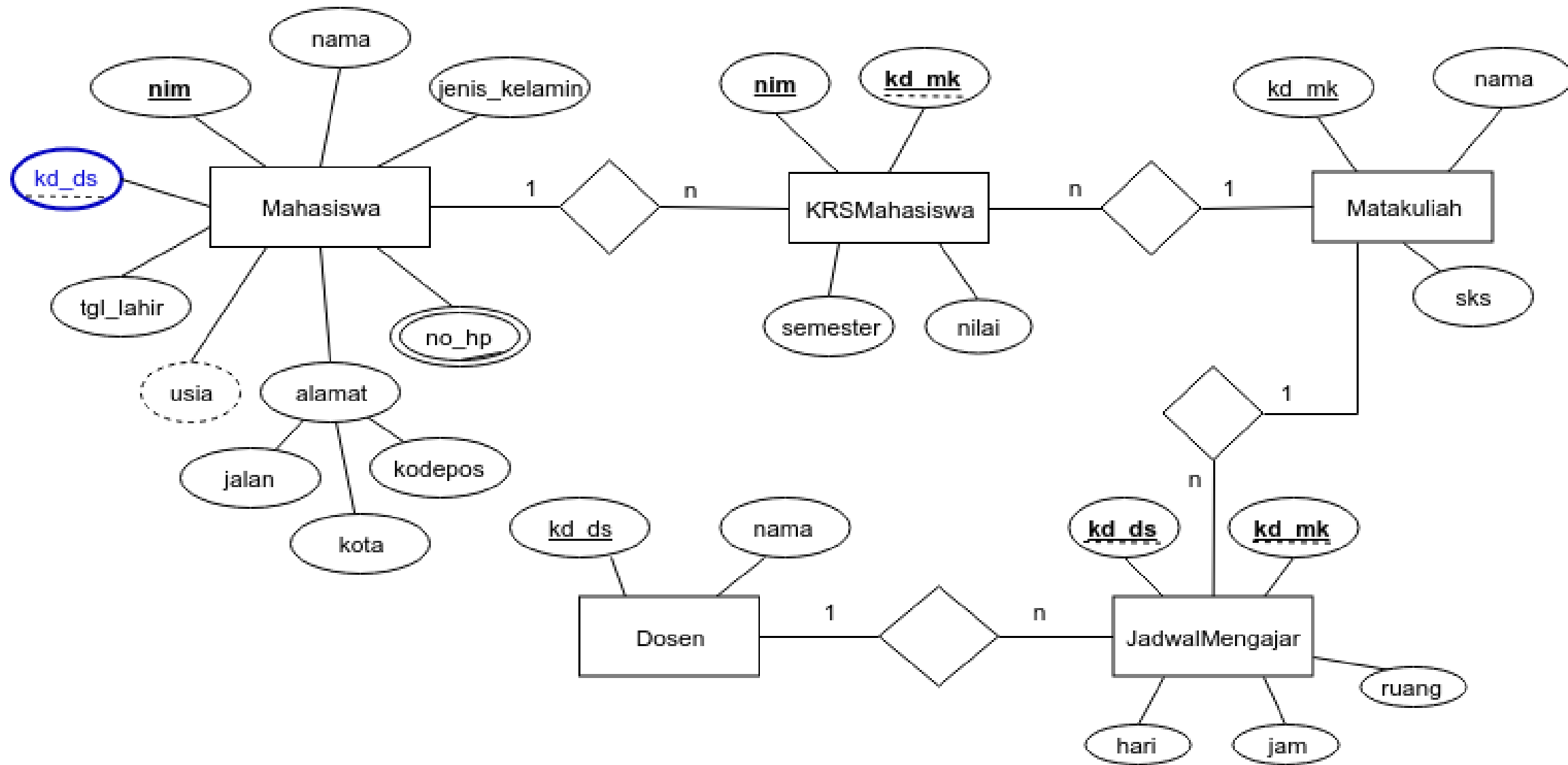
Sampai disini apakah sudah selesai?



Coba kita perhatikan kembali:

Entitas yang baru, **KRSMahasiswa** dan **JadwalMengajar** mendapat atribut baru dan yang menjadi atribut kunci (**key atribut**) adalah bersifat komposit (**conposite key**), yaitu gabungan dari 2 atribut yang berasal dari **partial key atribut**.

Sampai disini apakah masalah selesai?

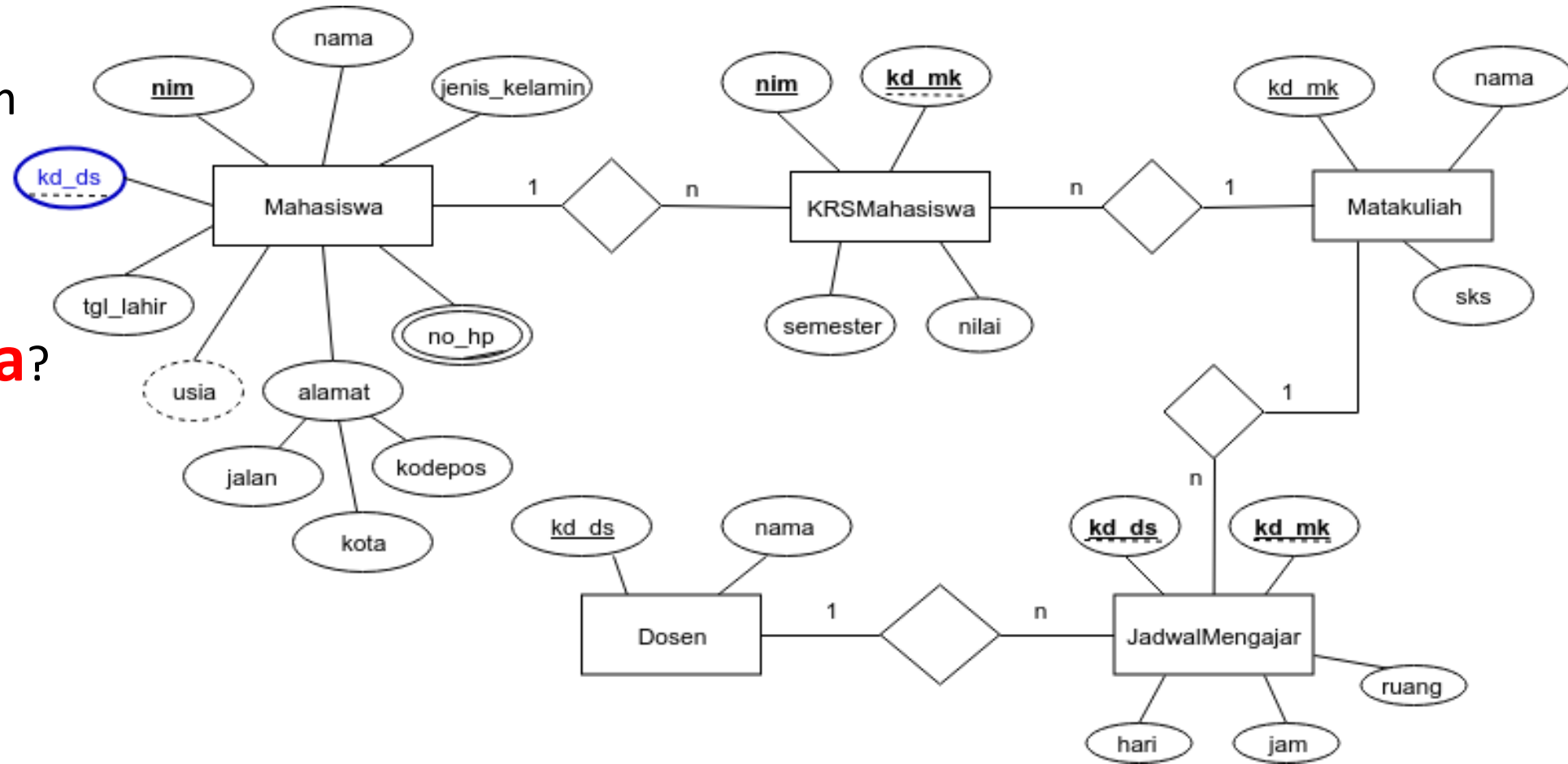


Sampai disini apakah masalah selesai?

Ternyata masih belum selesai,

lalu,

Apa masalahnya?



Sampai disini apakah masalah selesai?

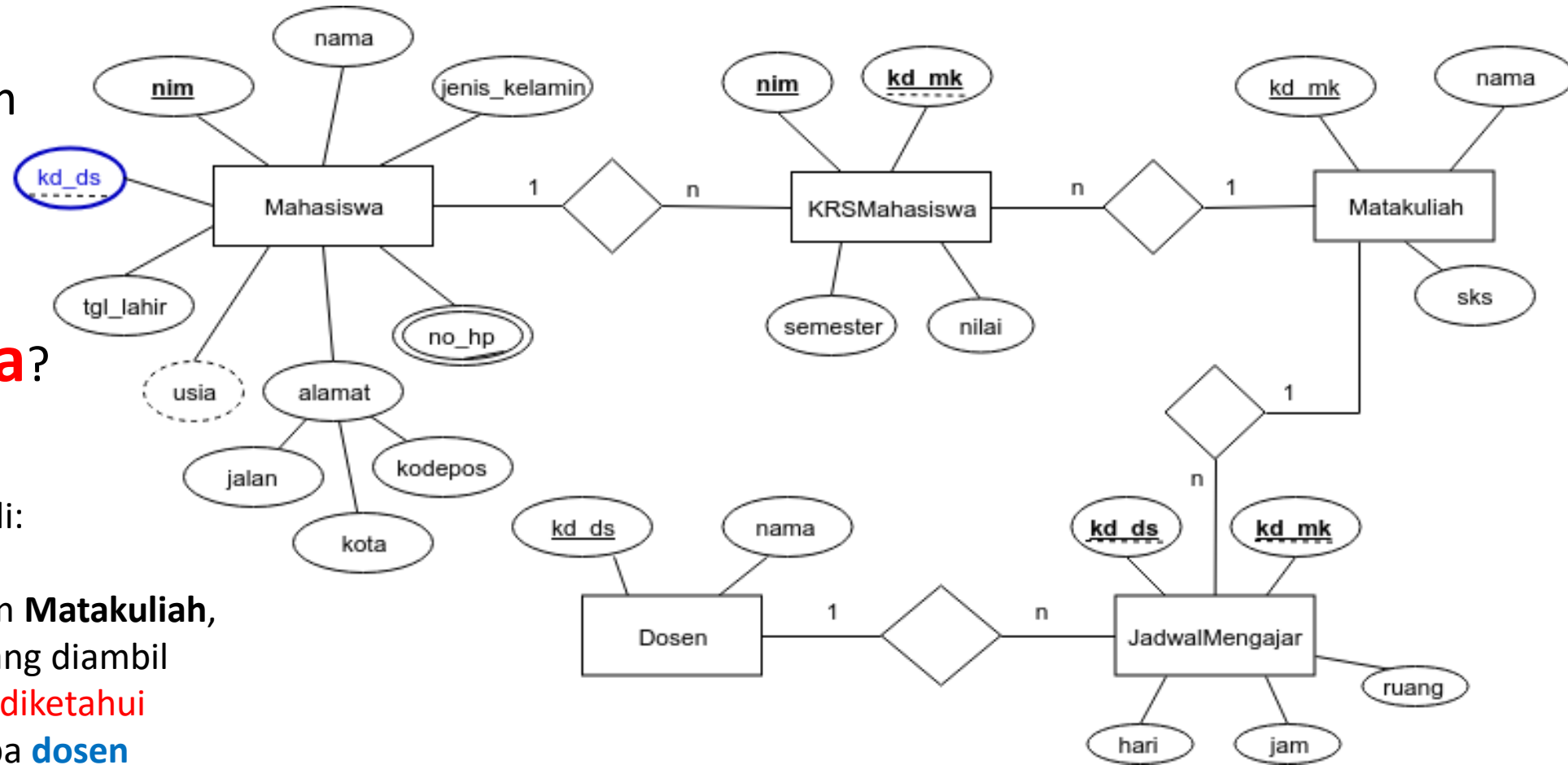
Ternyata masih belum selesai,

lalu,

Apa masalahnya?

Coba kita perhatikan kembali:

Relasi antara **Mahasiswa** dan **Matakuliah**, terlihat disitu mata kuliah yang diambil oleh mahasiswa **belum** bisa **diketahui jadwal perkuliahan** dan siapa **dosen pengampunya**.



Sampai disini apakah masalah selesai?

Ternyata masih belum selesai,

lalu,

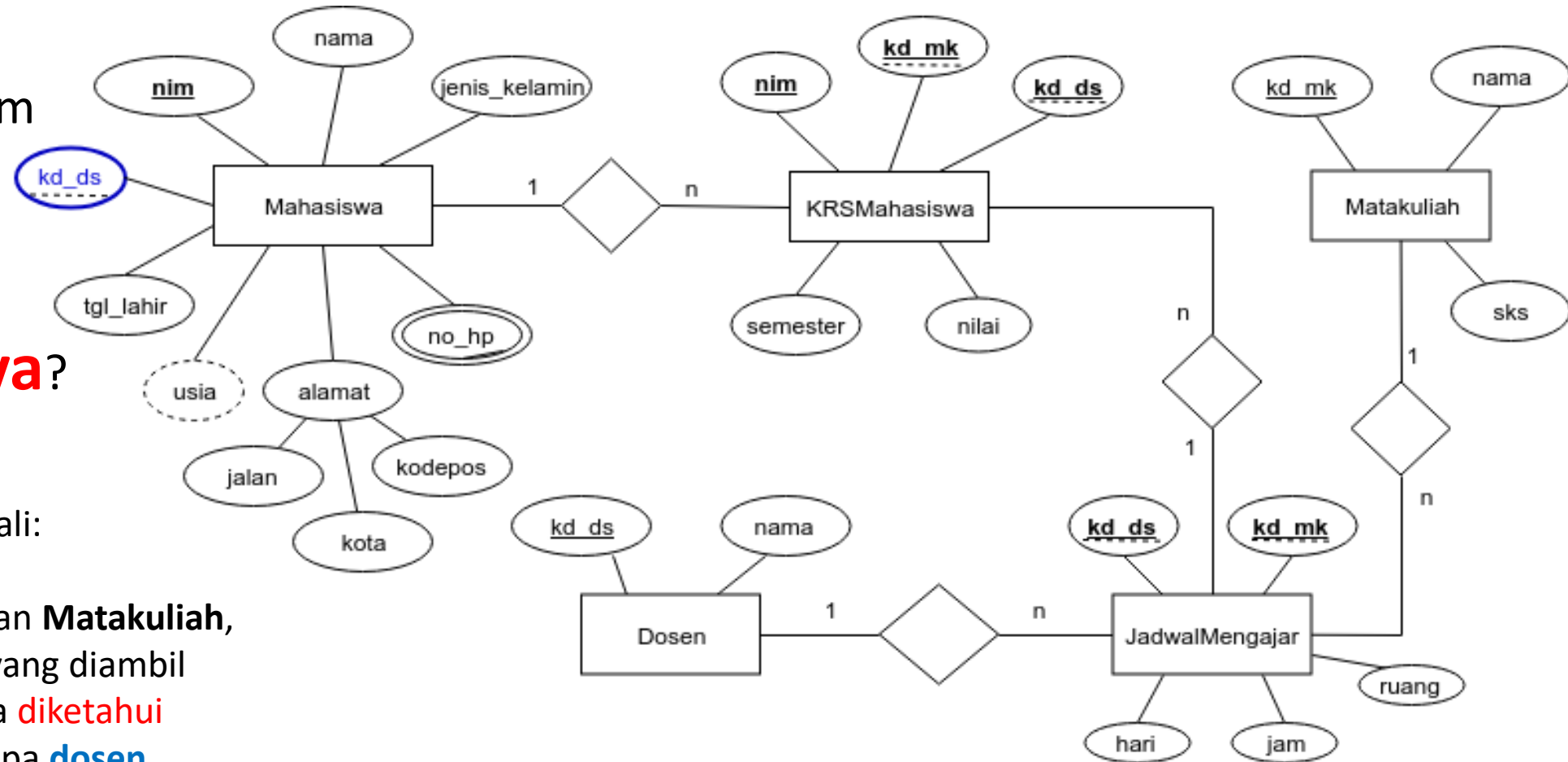
Apa masalahnya?

Coba kita perhatikan kembali:

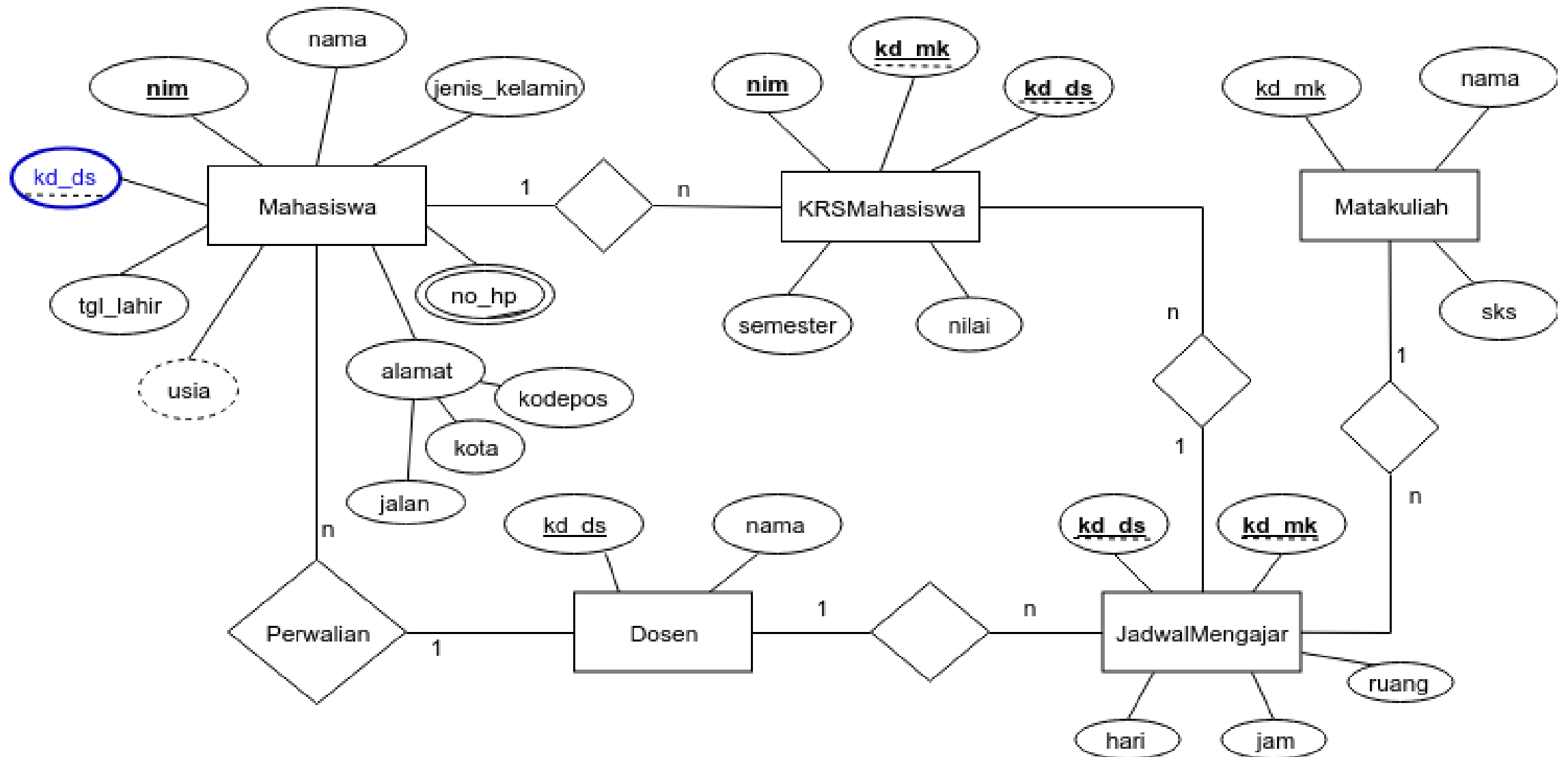
Relasi antara **Mahasiswa** dan **Matakuliah**, terlihat disitu mata kuliah yang diambil oleh mahasiswa **belum** bisa **diketahui jadwal perkuliahan** dan siapa **dosen pengampunya**.

Maka, perlu kita **sesuaikan relasinya** ==>

KRSMahasiswa sekarang **berelasi** dengan **JadwalMengajar**



Hasil Akhir



Transform ER-D

Selanjutnya kita transformasikan hasil akhir **ER-D** kedalam **Mapping Table**.

Tabel:

Mahasiswa (nim, nama, jenis_kelamin, tgl_lahir, jalan, kota, kodepos, no_hp, kd_ds)

Dosen (kd_ds, nama)

Matakuliah (kd_mk, nama, sks)

JadwalMengajar (kd_ds, kd_mk, hari, jam, ruang)

KRSMahasiswa (nim, kd_mk, kd_ds, semester, nilai)

