1. DATABASE MODEL
   1. Form request add field
      1. createdDate
      2. updatedDate
   2. problem nosql vs relational
      1. fetch data ke tiap table (relational udah dicover, coba cari tahu tentang eager vs lazy)
      2. gpp embedded. Paling yang harus diperhatikan fetch data nya.
2. API SPEC

Dokumentasi semua use case yang ada. Use case akan di translate menjadi 1 api / method.

Pengantar:

* Httprequestmethod: POST (modify/create data), GET (fetch data) , PUT (modify/update data), PATCH (modify data), DELETE(delete)
* Naming convention for API
  + Misal kita punya use case CRUD Employee, ada 4 api berarti, create, update, get, delete
  + Naming formatnya: {context-path}/{entity}/
  + Entity -> entitas yang dituju. Misal employee
  + Context path -> application name
  + Cth: missal nama aplikasinya: inventory-service
  + /inventory-service/employee
    - Create -> POST: /inventory-service/employee

Parameter: bisa berupa JSON body atau request parameter

Json body seunggukan variable dalam 1 object

Request parameter itu primitive data type.

Note: semua request dan response dalam bentuk JSON.

Tujuannya api spec:

* Mendetailkan use case:
  + Misal create Employee:
    - Create employee ini request nya apa2 saja. Supaya bisa menyimpan 1 data

Employee?

* + - Response nya apa?
    - Path nya apa?
    - Perlu header kah?
    - Perlu parameter kah?
* Task: Identifikasi dari To Do List project semua method yang ada
  + Request & response nya
  + http request
  + header
  + request parameter
  + request path

Header vs parameter (baca2 lagi)

API Create Employee Data