

farm-life

Learning

- Student can apply MVC and OOP concept
- Student can analyze the task and find solution

Summary

Seorang pengusaha perkebunan bernama Vincent ingin mengatur dan mengetahui proses dari hasil kebun tersebut. Vincent memiliki banyak kebun dan masing-masing kebun tersebut di manage oleh seseorang yang dia percayai. Dalam 1 perkebunan dapat memiliki banyak hewan ternak yang menghasilkan hasil ternak sesuai dengan jenisnya.

Masing-masing kebun dapat memiliki lebih dari 1 jenis ternak. Ternak memiliki hasil ternak sesuai jenisnya. Buatlah aplikasi **Farm Management** menggunakan konsep MVC dan OOP.

Buat command - command seperti berikut :

```
node farm.js help
node farm.js show
node farm.js create <farm_name>
node farm.js details <farm_name>
node farm.js add <farm_name> <livestock_type>
node farm.js sell <farm_name> <livestock_id>
node farm.js totalProduce <farm_name>
node farm.js sort <sort_type>
```

Karena akan mengambil dari terminal maka menggunakan **process.argv**.

Tasks

Task 1

Buatlah class **Farm** yang memiliki :

key	description
id	Berisi nomor unik dari Farm (nomor id)
name	Berisi nama dari Farm , tipe data string
livestocks	Berisi array dari Livestock yang memiliki inheritance,

Task 2

Buatlah class **Livestock** yang memiliki informasi sebagai berikut :

key	description
id	Berisi nomor unik dari Livestock (nomor id)
type	Berisi tipe dari Livestock , tipe data string

Task 3

Buatlah class inheritance **Chicken**, **Cow**, **Sheep** dengan menambahkan key pada masing - masing class seperti berikut :

- **Chicken** menambahkan key : *eggs*
- **Cow** menambahkan key : *milks*
- **Sheep** menambahkan key : *wools*

Task 4

Buatlah **data.json** untuk menampung data **Farm**.

Struktur dari **data.json** adalah sebagai :

```
[
  {
    "id": 1,
    "name": "Green Field",
    "livestocks": [
      {
        "id": 1,
        "type": "chicken",
        "eggs": 10
      },
      {
        "id": 2,
        "type": "cow",
        "milks": 5
      },
      {
        "id": 3,
        "type": "sheep",
        "wools": 3
      }
    ]
  }
]
```

Task 5

Aplikasikan command - command berikut ini :

- Command *show*

```
node farm.js show

// Result
// Farms list :
// 1. "Green Field" has 5 livestock
// 2. "Blue Soil" has 3 livestock
```

- Command `_create <farm_name>`

```
node farm.js create "Brown Land"

// Result
// "Brown Land" has been bought!
```

- Command `_detail <farm_name>`

```
node farm.js "Green Field"

//Result
// Green Field stats :
// 1. Chicken, eggs : 10,
// 2. Cow, milks : 5,
// 3. Sheep, wools : 3,
```

- Command `_add <farm_name> <livestock_type>`

Ketika menambahkan livestock, akan menghasilkan hasil ternak secara random dar *Range 1 - 10*.

```
node farm.js "Brown Land" chicken

// Result
// A chicken has been added to "Brown Land"!
```

- Command `sell <farm_name> <livestock_type>`

```
node farm.js sell "Brown Land" 1

// Result
// A chicken has been sold!

// Error / not found
// Livestock not found.
```

- Command *totalProduce* <*farm_name*>

```
node farm.js totalProduce "Green Field"

// Result
// "Green Field" has produced 18 goods
```

- Command *sort* <*sort_type*>

Sort Types :

- produce
- livestock

```
node farm.js sort produce

// Result :
/*
  Sort Produce :
  1. Green Field : 30 goods
  2. Sunny Day : 15 goods
  3. Brown Lans : 7 goods
*/

node farm.js sort livestock

// Result :
/*
  Sort Livestocks :
  1. Green Field : 5 Livestocks
  2. Sunny Day : 3 Livestocks
  3. Brown Land : 1 Livestocks
*/
```