REQUEST FOR COMMENTS

1. Summary

Membuat software warehouse management system untuk memonitor aktifitas operational pada warehouse.

1. Problem and Motivation

* Mengatur dan menjaga pasokan stock produk
* Melihat dan menerima orderan dari pelanggan
* Melacak pengiriman barang

1. Detailed Design

* ERD

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| warehouse |  | product |  | transaction |  | customer |  | region |
| id(PK) |  | id(PK) |  | id(PK) |  | id(PK) |  | id(PK) |
| product\_id(FK) |  | category |  | Customer\_id  (FK) |  | region\_id(FK) |  | cities |
| region\_id(FK) |  | stock |  | warehouse\_id (FK) |  | address |  |  |
| employee\_id (PK) |  | price |  | product\_id(FK) |  | email |  |  |
|  |  |  |  | quantity |  | phone |  |  |
|  |  |  |  | total |  |  |  |  |
|  |  |  |  | created\_at |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| shipping |  | employee |  |  |  |  |  |  |
| id(PK) |  | id(PK) |  |  |  |  |  |  |
| transaction\_id (FK) |  | name |  |  |  |  |  |  |
| employee\_id (FK) |  | phone |  |  |  |  |  |  |
| timestamp |  |  |  |  |  |  |  |  |

* API Contract

# Create list as API

product = [{'name': 'pakan ikan', 'price':25000,'stock':60},

{'name': 'pakan udang','price': 30000,'stock':40},

{'name': 'benih ikan', 'price': 30000,'stock':15}]

# Create function to modify API list

class order(Resource):

def get(self,name):

index = next((index for (index, d) in enumerate(product) if d["name"] == name), None)

for items in product:

if items['name'] == name:

return product[index]

return {'item': 'unavailable'},404

def post(self,name):

index = next((index for (index, d) in enumerate(product) if d["name"] == name), None)

for items in product:

if items['name'] == name:

return items['stock'] + 1

return product[index]

def put(self,name1,price1,stock1,name2,price2,stock2):

index = next((index for (index, d) in enumerate(pokedex) if d["name"] == name1), None)

for items in product:

if items['name'] == name1:

product.pop(index)

items = {'name': name2, 'type': type2}

product.insert(index,items)

def delete(self,name):

index = next((index for (index, d) in enumerate(product) if d["name"] == name), None)

for items in product:

if items['name'] == name:

return items['stock'] - 1

return product[index]