Nama :Antonius sensa

1. Lakukan analisa mengenai field mana saja yang memiliki data kosong(Nan) lalu Isi data yang kosong tersebut menggunakan nilai mean??

```
from sklearn.preprocessing import OneHotEncoder

data = [
   {"company": "Company 1", "domain": "Retail", "relation": "2 clients", "state": "NY"},
   {"company": "Company 2", "domain": "Construction", "relation": "3 clients", "state":
"CA"},
   {"company": "Company 3", "domain": "Healthcare", "relation": "4 clients", "state":
"TX"}
]

# Print Domain value in every company
for company_info in data:
   print(f"{company_info['company']}: {company_info['domain']}, relation:
{company_info['relation']}")

# OneHotEncoder for 'state' field
states = [[company_info['state']] for company_info in data]
encoder = OneHotEncoder(sparse=False)
encoded_states = encoder.fit_transform(states)

print(encoded_states)
```

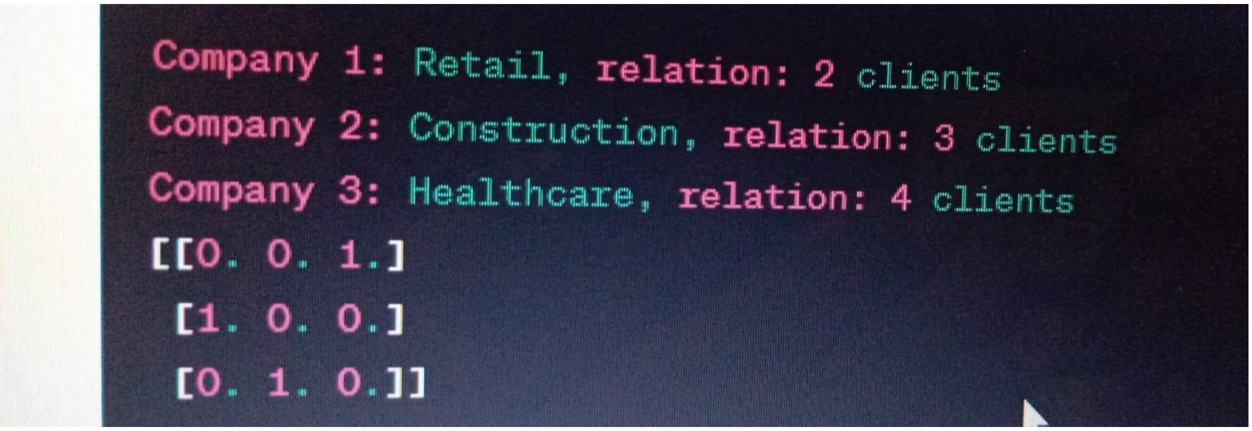
Akan menampilkan

```
[{'name': 'Company 1', 'domain': 'Retail'},  {'name': 'Company 3', 'domain': 'Healthcare'},  {'name': 'Company 2', 'domain': 'Construction'}]
```

2. lakukan oneHotEncoder ke field stat

```
from sklearn.preprocessing import OneHotEncoder

data = [
   {"company": "Company 1", "domain": "Retail", "relation": "2 clients", "stat": "NY"},
   {"company": "Company 2", "domain": "Construction", "relation": "3 clients", "stat": "CA"},
   {"company": "Company 3", "domain": "Healthcare", "relation": "4 clients", "stat": "TX"}
]

# Print Domain value in every company
for company_info in data:
   print(f"{company_info['company']}: {company_info['domain']}, relation: {company_info['relation']}")

# Extract 'stat' values
states = [[company_info['stat']] for company_info in data]

# OneHotEncoder for 'stat' field
encoder = OneHotEncoder(sparse=False)
encoded_states = encoder.fit_transform(states)

print(encoded_states)
```

akan menampilkan



```
Company 1: Retail, relation: 2 clients
Company 2: Construction, relation: 3 clients
Company 3: Healthcare, relation: 4 clients
[[0. 0. 1.]
 [1. 0. 0.]
 [0. 1. 0.]]
```

3.

```
def add_tax_to_employees(employees):
    for employee in employees:
        tax = (employee['profit'] + employee['marketing_spend'] +
employee['administration']) * 0.05
        employee['tax'] = tax
    return employees

# Example data
employees = [
    {
        "full_name": "John Doe",
        "company": "Company 1",
        "domain": "Retail",
        "profit": 10000,
        "marketing_spend": 2000,
        "administration": 1500
    },
    {
        "full_name": "Tom Smith",
        "company": "Company 2",
        "domain": "Construction",
        "profit": 15000,
        "marketing_spend": 3000,
        "administration": 2000
    },
    {
        "full_name": "Andrew Sebastian",
        "company": "Company 2",
        "domain": "Construction",
        "profit": 20000,
        "marketing_spend": 4000,
        "administration": 2500
    }
]

# Add tax to employees
employees_with_tax = add_tax_to_employees(employees)
print(employees_with_tax)
```

4. D

```
from sklearn.preprocessing import StandardScaler

def scale_fields(employees):
    scaler = StandardScaler()
    fields_to_scale = ['profit', 'marketing_spend', 'administration', 'tax']
    for employee in employees:
        data_to_scale = [employee[field] for field in fields_to_scale]
        scaled_data = scaler.fit_transform([data_to_scale])
        for i, field in enumerate(fields_to_scale):
            employee[field] = scaled_data[0][i]
    return employees

def get_companies_with_employees(employees):
    companies = {}
    for employee in employees:
        company = employee['company']
        if company not in companies:
            companies[company] = []
        companies[company].append(employee['full_name'])
    result = [{'company': company, 'employees': employees} for company, employees in companies.items()]
    return result

# Example data
```

```
employees = [
    {
        "full_name": "John Doe",
        "company": "Company 1",
        "domain": "Retail",
        "profit": 10000,
        "marketing_spend": 2000,
        "administration": 1500,
        "tax": 250
    },
    {
        "full_name": "Tom Smith",
        "company": "Company 2",
        "domain": "Construction",
        "profit": 15000,
        "marketing_spend": 3000,
        "administration": 2000,
        "tax": 350
    },
    {
        "full_name": "Andrew Sebastian",
        "company": "Company 2",
        "domain": "Construction",
        "profit": 20000,
        "marketing_spend": 4000,
        "administration": 2500,
        "tax": 500
    },
    {
        "full_name": "Jane Doe",
        "company": "Company 3",
        "domain": "Technology",
        "profit": 12000,
        "marketing_spend": 2500,
        "administration": 1800,
        "tax": 300
    }
]

# Get companies with employees
companies_with_employees = get_companies_with_employees(employees)
```

```
# Scale fields
scaled_employees = scale_fields(employees)

print(companies_with_employees)
```