ArrayOfObject_vs_NestedObject

April 14, 2024

[ ]:
```
import pandas as pd
```

0.1 Dict of Lists

Column Oriented Data

[ ]:
```
student_data = {
    "ids": [1, 2, 3],
    "names": ["Andi", "Budi", "Cindy"],
    "ages": [15, 16, 15],
    "class_ids": [101, 102, 101]
}

# Data Class menggunakan Dict of lists
class_data = {
    "class_ids": [101, 102],
    "class_names": ["Mathematics", "History"],
    "subjects": [
        ["Math", "Science"],
        ["History", "English"]
    ]
}

student_df = pd.DataFrame(student_data)
class_df = pd.DataFrame(class_data)

print(student_df.head())
print(class_df.head())
```

```
   ids  names  ages  class_ids
0    1   Andi    15        101
1    2   Budi    16        102
2    3  Cindy    15        101
   class_ids  class_names            subjects
0        101  Mathematics     [Math, Science]
1        102      History  [History, English]
```

0.2 List of Dicts

Row Oriented Data

[ ]:
```
students_data = [
    {"id": 1, "name": "Andi", "age": 15, "classId": 101},
    {"id": 2, "name": "Budi", "age": 16, "classId": 102},
    {"id": 3, "name": "Cindy", "age": 15, "classId": 101}
]

classes_data = [
    {"classId": 101, "className": "Mathematics", "subjects": ["Math",␣
 ↪"Science"]},
    {"classId": 102, "className": "History", "subjects": ["History", "English"]}
]

students_df = pd.DataFrame(students_data)
classes_df = pd.DataFrame(classes_data)

print(students_df.head())
print(classes_df.head())
```

```
   id   name  age  classId
0   1   Andi   15      101
1   2   Budi   16      102
2   3  Cindy   15      101
   classId    className            subjects
0      101  Mathematics     [Math, Science]
1      102      History  [History, English]
```

0.3 Konversi Data

[ ]:
```
student_df.to_dict(orient='records')
```

[ ]:
```
[{'ids': 1, 'names': 'Andi', 'ages': 15, 'class_ids': 101},
 {'ids': 2, 'names': 'Budi', 'ages': 16, 'class_ids': 102},
 {'ids': 3, 'names': 'Cindy', 'ages': 15, 'class_ids': 101}]
```

[ ]:
```
student_df.to_dict(orient='list')
```

[ ]:
```
{'ids': [1, 2, 3],
 'names': ['Andi', 'Budi', 'Cindy'],
 'ages': [15, 16, 15],
 'class_ids': [101, 102, 101]}
```

0.4 Kesimpulan

jadi baik ditulis dalam bentuk Dict of Lists seperti kode berikut

```
student_data = {
    "ids": [1, 2, 3],
    "names": ["Andi", "Budi", "Cindy"],
    "ages": [15, 16, 15],
    "class_ids": [101, 102, 101]
}
```

maupun dengan List of Dicts

```
student_data = [
    {"id": 1, "name": "Andi", "age": 15, "classId": 101},
    {"id": 2, "name": "Budi", "age": 16, "classId": 102},
    {"id": 3, "name": "Cindy", "age": 15, "classId": 101}
]
```

keduanya bisa ditangani dengan baik dengan cara yang sama dengan pandas

```
students_df = pd.DataFrame(students_data)
```

Selain itu pandas juga dapat mengonversi dari Dataframe ke List of Dicts maupun ke Dict of Lists

```
student_df.to_Dict(orient='records') # List of Dicts
student_df.to_Dict(orient='list') # Dict of Lists
```

referensi lebih lanjut

https://stackoverflow.com/questions/30522982/list-with-many-dictionaries-vs-dictionary-with-fe