Task Project 4 (Streaming processing)

1. Check Module ML
2. masukan data ke dlm db
3. lempar data new ke kafka, time.sleep(10)
4. Mengambil data dari kafka, procedure dan consumer dijalankan bersamaan
5. ambil data dalam database
6. Join data kafka Dan DB postgres
7. panggil func Predict Detect
8. insert di mongodb
9. Save data CSV
10. Data Visualisation (Bonus)

File Edit Selection View Go ··· DE - STREAM PROCESSING

EXPLORER

DE - STREAM PROCESSING
- DE - STREAM PROCESSING
  - .ipynb_checkpoints
    - consumer-checkpoint.ipynb
    - mongo-checkpoint.ipynb
    - Tes-Consumer-checkpoint.ipynb
    - Tes-Model-checkpoint.ipynb
  - modelling
    - __pycache__
    - .ipynb_checkpoints
    - packages
    - __init__.py
    - FraudModel.py
    - modelling.ipynb
  - producer
    - .ipynb_checkpoints
    - __init__.py
    - datadump.ipynb
    - New_Information.csv
    - Old_Information.csv
    - producer.ipynb
    - producer.py
  - ~$Materi - Stream Processing.pptx
  - consumer.ipynb
  - Materi - Stream Processing.pptx
  - Tes-Consumer.ipynb
  - Tes-Model.ipynb
  - Tes-Mongo.ipynb
- env

OUTLINE
TIMELINE

datadump.ipynb | producer.ipynb | Tes-Model.ipynb | modelling.ipynb | FraudModel.py | Tes-Mongo.ipynb

DE - STREAM PROCESSING > Tes-Model.ipynb > result

+ Code + Markdown | Run All | Restart | Execute Group 1 | Execute Group 2 | Clear All Outputs | Variables | Outline ··· env (Python 3.11.9)

```
from modelling import FraudModel
```
[1] ✓ 0.5s Python

```
#parameter inputan data
new_data = {
    'step': 1,
    'type': 'PAYMENT',
    'amount': 9839.64,
    'oldbalanceOrg': 170136.0,
    'newbalanceOrig': 160296.36,
    'oldbalanceDest': 0.0,
    'newbalanceDest': 0.0
}

new_data
```
[7] ✓ 0.0s Python

```
{'step': 1,
 'type': 'PAYMENT',
 'amount': 9839.64,
 'oldbalanceOrg': 120136.0,
 'newbalanceOrig': 150296.36,
 'oldbalanceDest': 0.0,
 'newbalanceDest': 0.0}
```

```
#parameter inputan path
import os

path = os.getcwd()
path = path + "\\modelling\\"
path
```
[8] ✓ 0.0s Python

```
'd:\\Data-Engineer-Skola\\Task\\project\\New folder\\DE - STREAM PROCESSING\\DE - STREAM PROCESSING\\modelling\\'
```

+ Code + Markdown

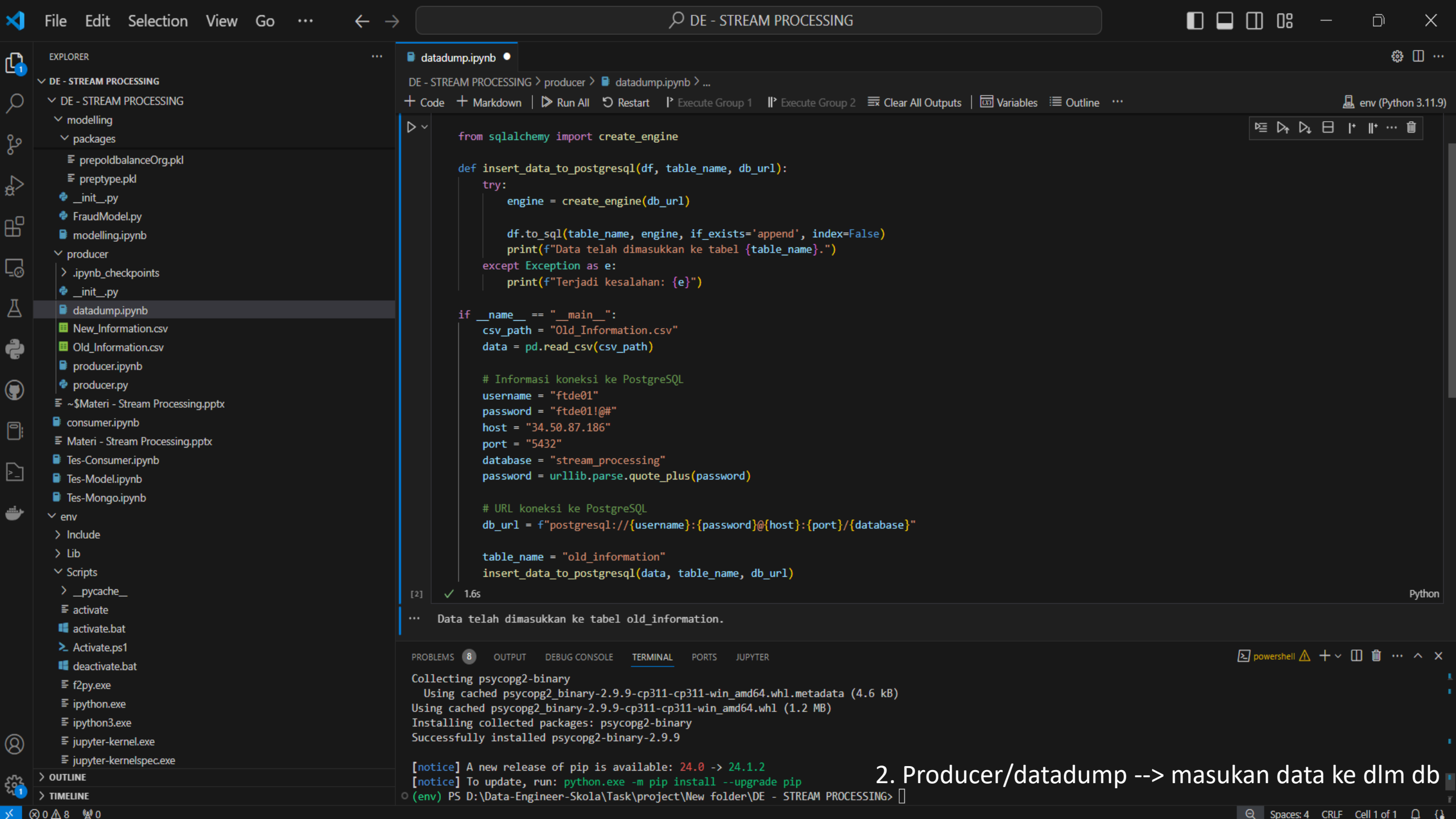
```
result = FraudModel.runModel(new_data, path)
```
[ ] Python

```
result
```
[12] ✓ 0.0s Python

```
'White List'
```

1. Modeling/Fraudmodul.py --> Check Module ML

⊗ 0 ⚠ 8 0 Spaces: 4 CRLF Cell 5 of 5


File Edit Selection View Go
DE - STREAM PROCESSING
EXPLORER
DE - STREAM PROCESSING
DE - STREAM PROCESSING
modelling
packages
prepoldbalanceOrg.pkl
preptype.pkl
__init__.py
FraudModel.py
modelling.ipynb
producer
.ipynb_checkpoints
__init__.py
datadump.ipynb
New_Information.csv
Old_Information.csv
producer.ipynb
producer.py
~$Materi - Stream Processing.pptx
consumer.ipynb
Materi - Stream Processing.pptx
Tes-Consumer.ipynb
Tes-Model.ipynb
Tes-Mongo.ipynb
env
Include
Lib
Scripts
__pycache__
activate
activate.bat
Activate.ps1
deactivate.bat
f2py.exe
ipython.exe
ipython3.exe
jupyter-kernel.exe
jupyter-kernelspec.exe
OUTLINE
TIMELINE
datadump.ipynb
DE - STREAM PROCESSING > producer > datadump.ipynb > ...
Code Markdown Run All Restart Execute Group 1 Execute Group 2 Clear All Outputs Variables Outline
env (Python 3.11.9)
from sqlalchemy import create_engine
def insert_data_to_postgresql(df, table_name, db_url):
try:
engine = create_engine(db_url)
df.to_sql(table_name, engine, if_exists='append', index=False)
print(f"Data telah dimasukkan ke tabel {table_name}.")
except Exception as e:
print(f"Terjadi kesalahan: {e}")
if __name__ == "__main__":
csv_path = "Old_Information.csv"
data = pd.read_csv(csv_path)
# Informasi koneksi ke PostgreSQL
username = "ftde01"
password = "ftde01!@#"
host = "34.50.87.186"
port = "5432"
database = "stream_processing"
password = urllib.parse.quote_plus(password)
# URL koneksi ke PostgreSQL
db_url = f"postgresql://{username}:{password}@{host}:{port}/{database}"
table_name = "old_information"
insert_data_to_postgresql(data, table_name, db_url)
[2] 1.6s
Python
Data telah dimasukkan ke tabel old_information.
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER
powershell
Collecting psycopg2-binary
Using cached psycopg2_binary-2.9.9-cp311-cp311-win_amd64.whl.metadata (4.6 kB)
Using cached psycopg2_binary-2.9.9-cp311-cp311-win_amd64.whl (1.2 MB)
Installing collected packages: psycopg2-binary
Successfully installed psycopg2-binary-2.9.9
[notice] A new release of pip is available: 24.0 -> 24.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
(env) PS D:\Data-Engineer-Skola\Task\project\New folder\DE - STREAM PROCESSING>
2. Producer/datadump --> masukan data ke dlm db
Spaces: 4 CRLF Cell 1 of 1

DBeaver 24.0.4 - old_information

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto stream_processing public@stream_processing

Projects

- postgres - *localhost:5432*
- postgres 2 - *postgresql://poke:p0k3!!123@postgres/pokebase*
- postgres 3 - *aws-0-ap-southeast-1.pooler.supabase.com:5432*
- stream_processing - *34.50.87.186:5432*
  - Databases
    - **stream_processing**
      - Schemas
        - **public**
          - Tables
            - old_information 3.3M
            - predicted_data 16K
          - Foreign Tables
          - Views
          - Materialized Views
          - Indexes
          - Functions
          - Sequences
          - Data types
          - Aggregate functions
        - Event Triggers
        - Extensions

Project - General

Name	DataSource
Bookmarks	
Dashboards	
Diagrams	
Scripts	

old_information predicted_data

Properties Data ER Diagram

stream_processing Databases stream_processing Schemas public Tables old_information

old_information Enter a SQL expression to filter results (use Ctrl+Space)

Grid Text Record

	nameOrig	oldbalanceOrg	nameDest	oldbalanceDest
1	C1231006815	170,136	M1979787155	0
2	C1666544295	21,249	M2044282225	0
3	C1305486145	181	C553264065	0
4	C840083671	181	C38997010	21,182
5	C2048537720	41,554	M1230701703	0
6	C90045638	53,860	M573487274	0
7	C154988899	183,195	M408069119	0
8	C1912850431	176,087.23	M633326333	0
9	C1265012928	2,671	M1176932104	0
10	C712410124	41,720	C195600860	41,898
11	C1900366749	4,465	C997608398	10,845
12	C249177573	20,771	M2096539129	0
13	C1648232591	5,070	M972865270	0
14	C1716932897	10,127	M801569151	0
15	C1026483832	503,264	M1635378213	0
16	C905080434	15,325	C476402209	5,083
17	C761750706	450	M1731217984	0
18	C1237762639	21,156	M1877062907	0
19	C2033524545	15,123	M473053293	0
20	C1670993182	705	C1100439041	22,425
21	C20804602	13,854	M1344519051	0
22	C1566511282	11,299	C1973528135	29,832

Value

C1231006815

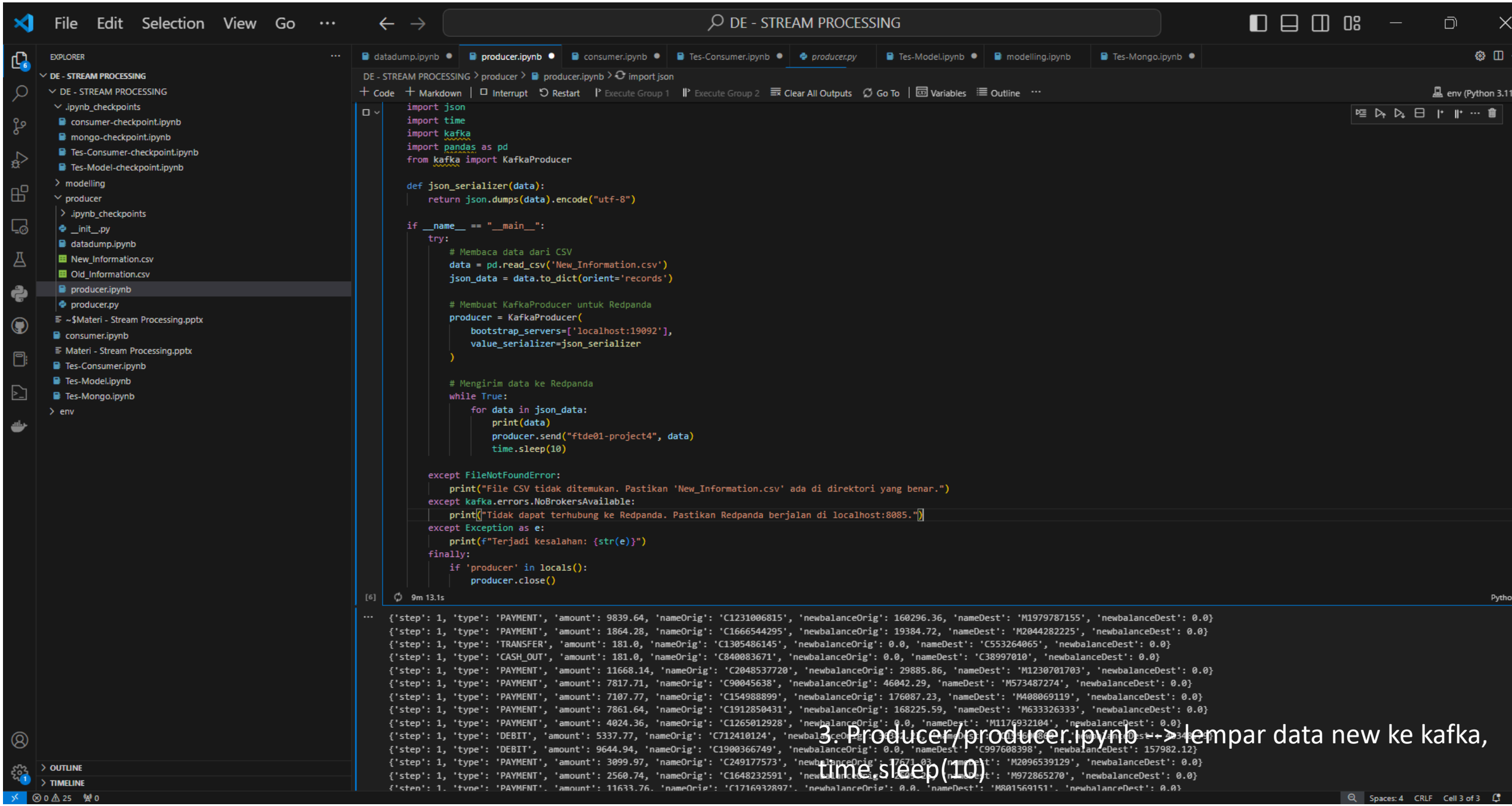
Panels

Refresh Save Cancel Export data 200 200+

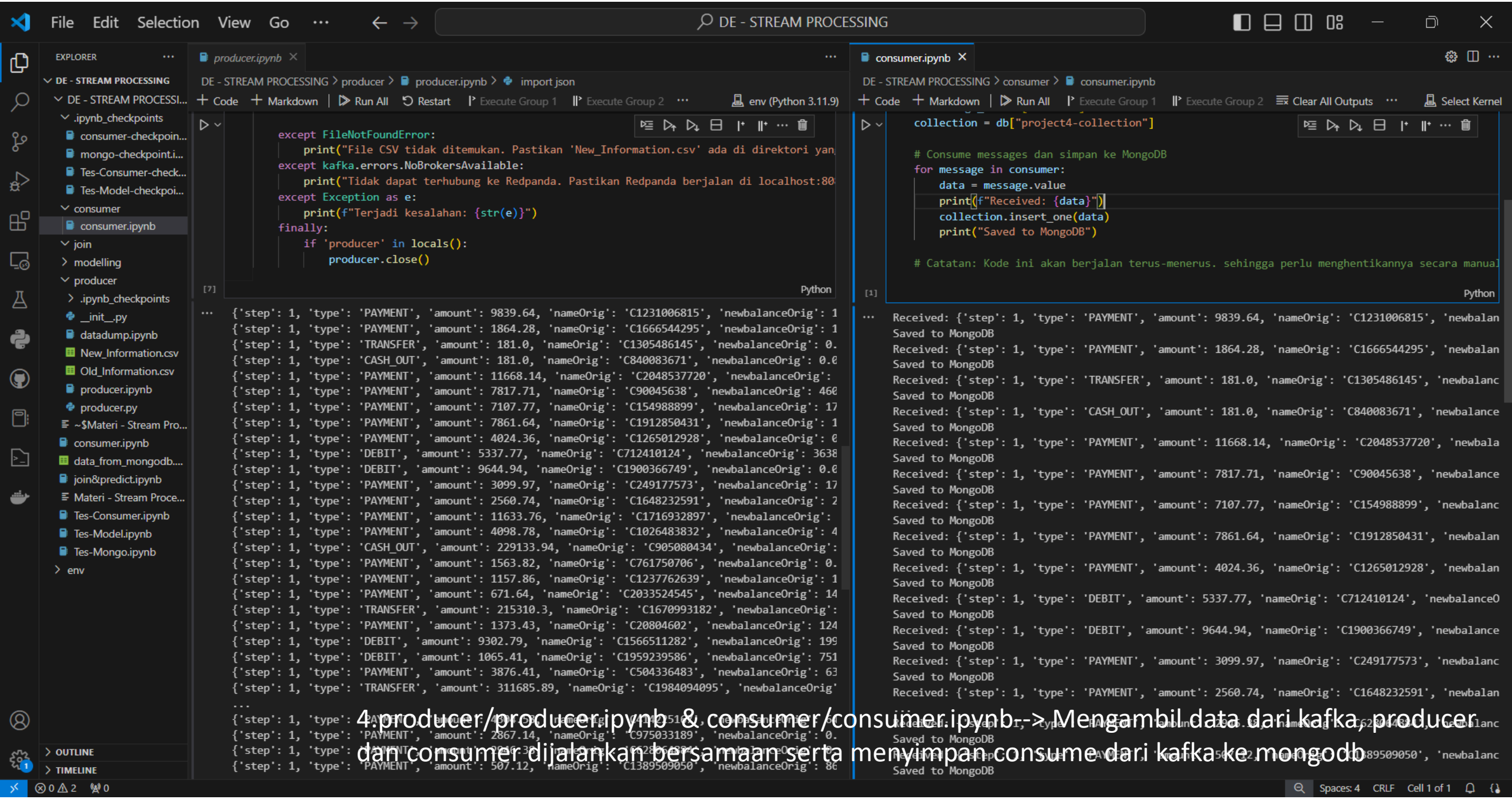
200 row(s) fetched - 0.015s (0.006s fetch), on 2024-07-11 at 21:25:48

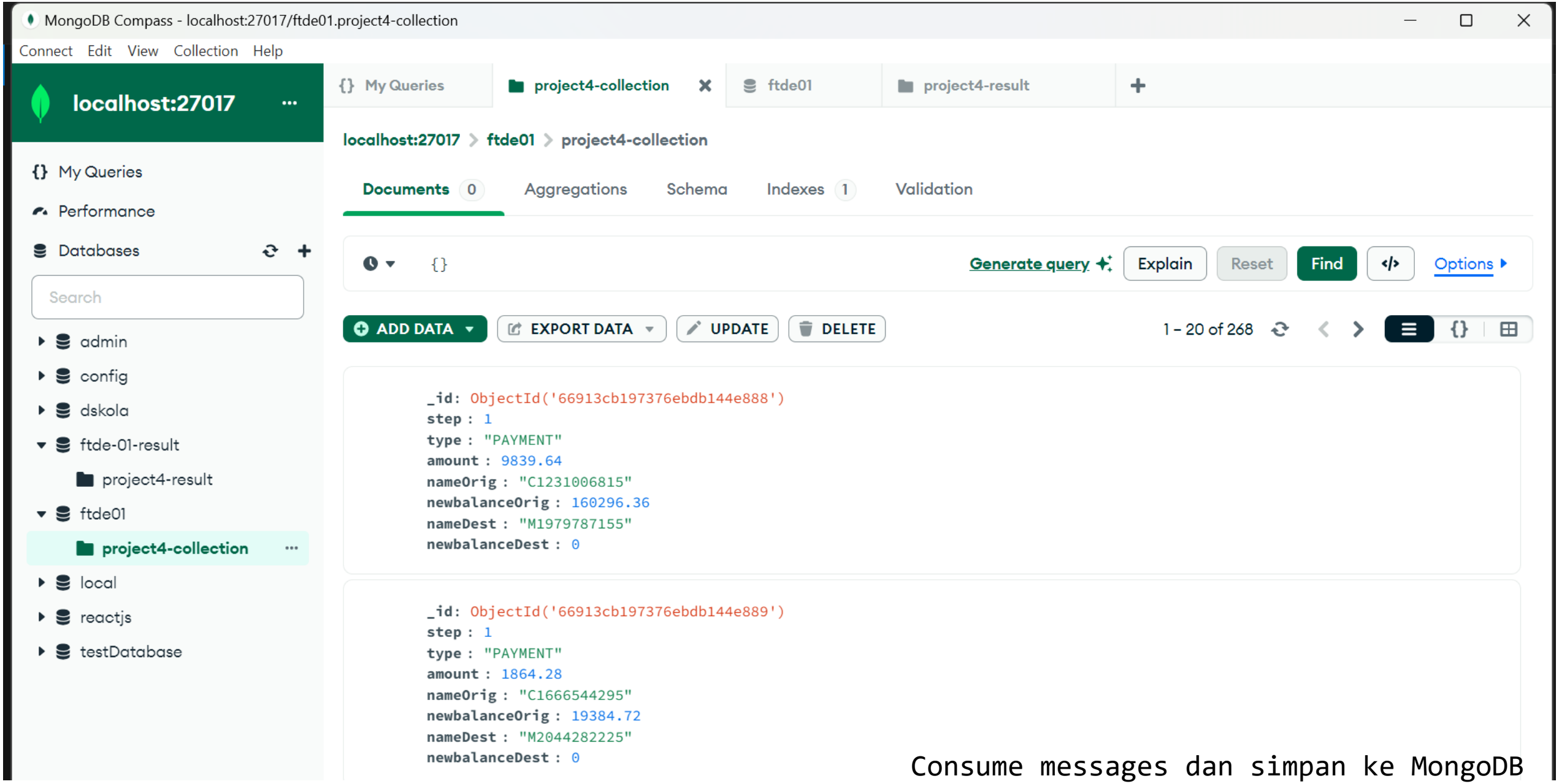
Tables - Tables

WIB en

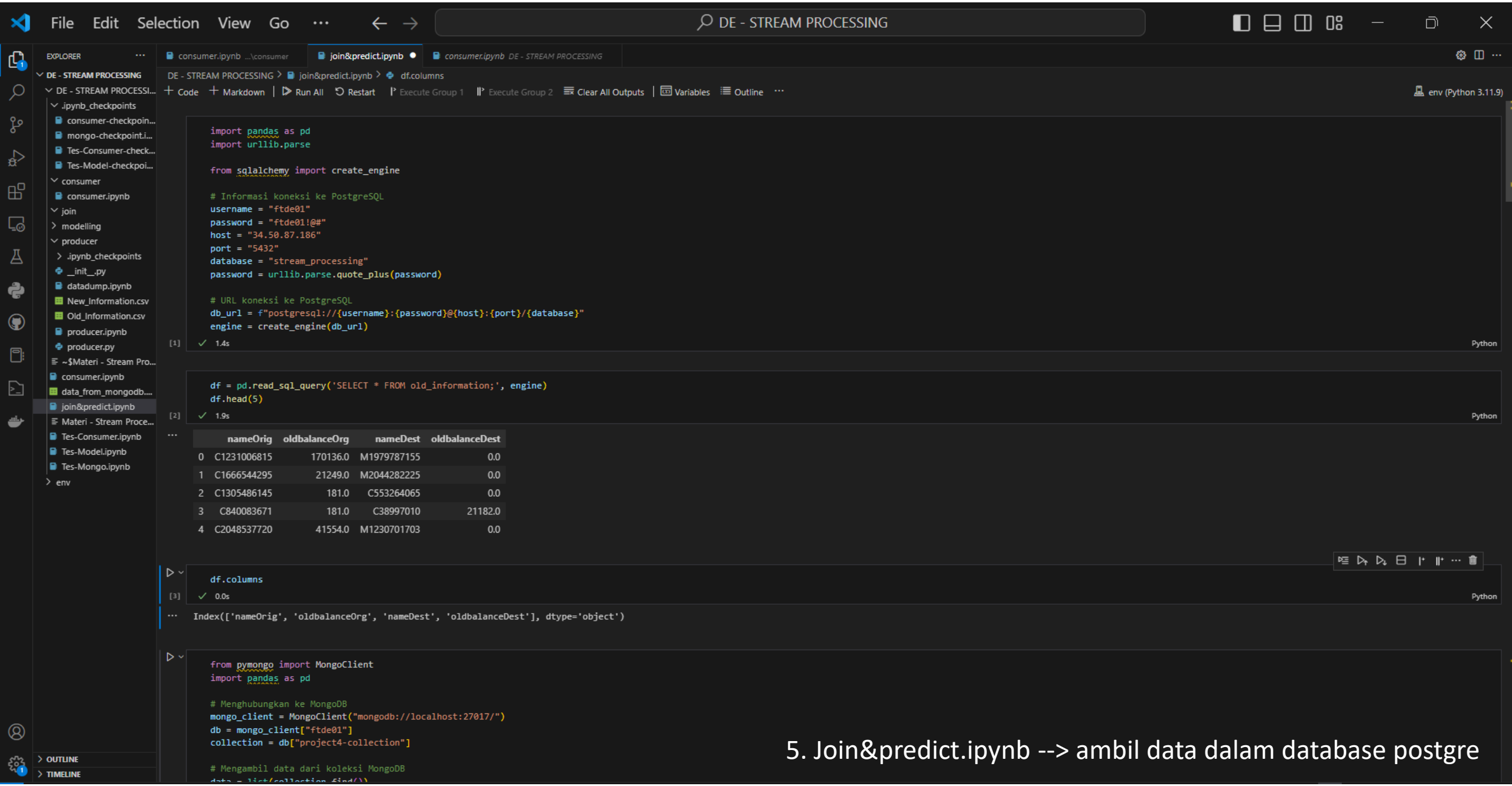


3. Producer/producer.ipynb --> kempar data new ke kafka, time.sleep(10)

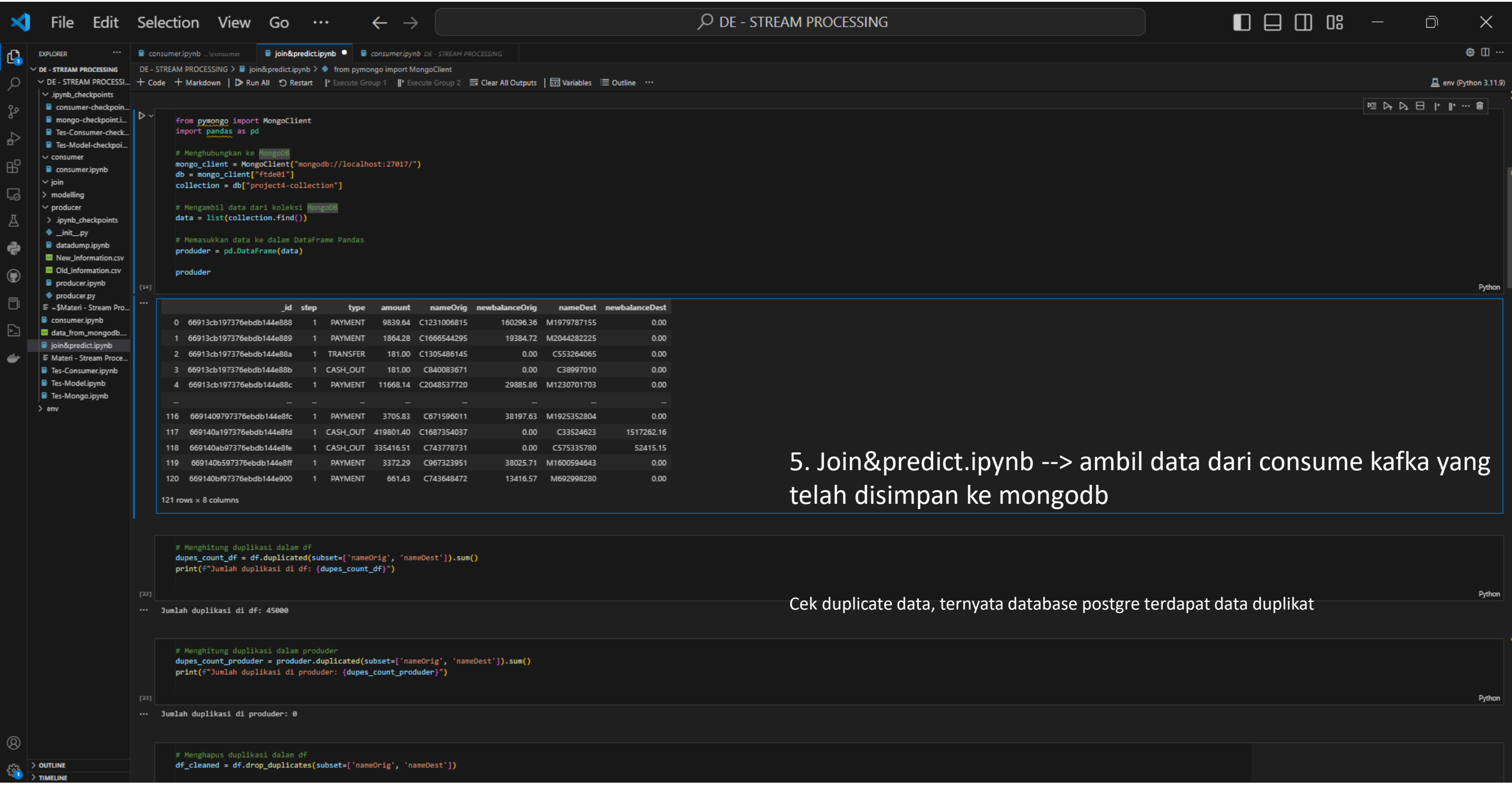

File Edit Selection View Go
DE - STREAM PROCESSING
EXPLORER
DE - STREAM PROCESSING
producer.ipynb
consumer.ipynb
DE - STREAM PROCESSING > producer > producer.ipynb > import json
DE - STREAM PROCESSING > consumer > consumer.ipynb
+ Code + Markdown | Run All Restart Execute Group 1 Execute Group 2 env (Python 3.11.9)
+ Code + Markdown | Run All Execute Group 1 Execute Group 2 Clear All Outputs Select Kernel
except FileNotFoundError:
print("File CSV tidak ditemukan. Pastikan 'New_Information.csv' ada di direktori yan
except kafka.errors.NoBrokersAvailable:
print("Tidak dapat terhubung ke Redpanda. Pastikan Redpanda berjalan di localhost:80
except Exception as e:
print(f"Terjadi kesalahan: {str(e)}")
finally:
if 'producer' in locals():
producer.close()
Python
collection = db["project4-collection"]
# Consume messages dan simpan ke MongoDB
for message in consumer:
data = message.value
print(f"Received: {data}")
collection.insert_one(data)
print("Saved to MongoDB")
# Catatan: Kode ini akan berjalan terus-menerus. sehingga perlu menghentikannya secara manual
Python
{'step': 1, 'type': 'PAYMENT', 'amount': 9839.64, 'nameOrig': 'C1231006815', 'newbalanceOrig': 1
{'step': 1, 'type': 'PAYMENT', 'amount': 1864.28, 'nameOrig': 'C1666544295', 'newbalanceOrig': 1
{'step': 1, 'type': 'TRANSFER', 'amount': 181.0, 'nameOrig': 'C1305486145', 'newbalanceOrig': 0.
{'step': 1, 'type': 'CASH_OUT', 'amount': 181.0, 'nameOrig': 'C840083671', 'newbalanceOrig': 0.0
{'step': 1, 'type': 'PAYMENT', 'amount': 11668.14, 'nameOrig': 'C2048537720', 'newbalanceOrig':
{'step': 1, 'type': 'PAYMENT', 'amount': 7817.71, 'nameOrig': 'C90045638', 'newbalanceOrig': 460
{'step': 1, 'type': 'PAYMENT', 'amount': 7107.77, 'nameOrig': 'C154988899', 'newbalanceOrig': 17
{'step': 1, 'type': 'PAYMENT', 'amount': 7861.64, 'nameOrig': 'C1912850431', 'newbalanceOrig': 1
{'step': 1, 'type': 'PAYMENT', 'amount': 4024.36, 'nameOrig': 'C1265012928', 'newbalanceOrig': 0
{'step': 1, 'type': 'DEBIT', 'amount': 5337.77, 'nameOrig': 'C712410124', 'newbalanceOrig': 3638
{'step': 1, 'type': 'DEBIT', 'amount': 9644.94, 'nameOrig': 'C1900366749', 'newbalanceOrig': 0.0
{'step': 1, 'type': 'PAYMENT', 'amount': 3099.97, 'nameOrig': 'C249177573', 'newbalanceOrig': 17
{'step': 1, 'type': 'PAYMENT', 'amount': 2560.74, 'nameOrig': 'C1648232591', 'newbalanceOrig': 2
{'step': 1, 'type': 'PAYMENT', 'amount': 11633.76, 'nameOrig': 'C1716932897', 'newbalanceOrig':
{'step': 1, 'type': 'PAYMENT', 'amount': 4098.78, 'nameOrig': 'C1026483832', 'newbalanceOrig': 4
{'step': 1, 'type': 'CASH_OUT', 'amount': 229133.94, 'nameOrig': 'C905080434', 'newbalanceOrig':
{'step': 1, 'type': 'PAYMENT', 'amount': 1563.82, 'nameOrig': 'C761750706', 'newbalanceOrig': 0.
{'step': 1, 'type': 'PAYMENT', 'amount': 1157.86, 'nameOrig': 'C1237762639', 'newbalanceOrig': 1
{'step': 1, 'type': 'PAYMENT', 'amount': 671.64, 'nameOrig': 'C2033524545', 'newbalanceOrig': 14
{'step': 1, 'type': 'TRANSFER', 'amount': 215310.3, 'nameOrig': 'C1670993182', 'newbalanceOrig':
{'step': 1, 'type': 'PAYMENT', 'amount': 1373.43, 'nameOrig': 'C20804602', 'newbalanceOrig': 124
{'step': 1, 'type': 'DEBIT', 'amount': 9302.79, 'nameOrig': 'C1566511282', 'newbalanceOrig': 199
{'step': 1, 'type': 'DEBIT', 'amount': 1065.41, 'nameOrig': 'C1959239586', 'newbalanceOrig': 751
{'step': 1, 'type': 'PAYMENT', 'amount': 3876.41, 'nameOrig': 'C504336483', 'newbalanceOrig': 63
{'step': 1, 'type': 'TRANSFER', 'amount': 311685.89, 'nameOrig': 'C1984094095', 'newbalanceOrig'
{'step': 1, 'type': 'PAYMENT', 'amount': 2867.14, 'nameOrig': 'C975033189', 'newbalanceOrig': 0.
{'step': 1, 'type': 'PAYMENT', 'amount': 507.12, 'nameOrig': 'C1389509050', 'newbalanceOrig': 80
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 9839.64, 'nameOrig': 'C1231006815', 'newbalan
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 1864.28, 'nameOrig': 'C1666544295', 'newbalan
Saved to MongoDB
Received: {'step': 1, 'type': 'TRANSFER', 'amount': 181.0, 'nameOrig': 'C1305486145', 'newbalanc
Saved to MongoDB
Received: {'step': 1, 'type': 'CASH_OUT', 'amount': 181.0, 'nameOrig': 'C840083671', 'newbalance
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 11668.14, 'nameOrig': 'C2048537720', 'newbala
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 7817.71, 'nameOrig': 'C90045638', 'newbalance
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 7107.77, 'nameOrig': 'C154988899', 'newbalanc
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 7861.64, 'nameOrig': 'C1912850431', 'newbalan
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 4024.36, 'nameOrig': 'C1265012928', 'newbalan
Saved to MongoDB
Received: {'step': 1, 'type': 'DEBIT', 'amount': 5337.77, 'nameOrig': 'C712410124', 'newbalanceO
Saved to MongoDB
Received: {'step': 1, 'type': 'DEBIT', 'amount': 9644.94, 'nameOrig': 'C1900366749', 'newbalance
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 3099.97, 'nameOrig': 'C249177573', 'newbalanc
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 2560.74, 'nameOrig': 'C1648232591', 'newbalan
Saved to MongoDB
Received: {'step': 1, 'type': 'PAYMENT', 'amount': 507.12, 'nameOrig': 'C1389509050', 'newbalanc
Saved to MongoDB
4.producer/producer.ipynb & consumer/consumer.ipynb -> Mengambil data dari kafka, producer
dan consumer dijalankan bersamaan serta menyimpan consume dari kafka ke mongodb
OUTLINE
TIMELINE
Spaces: 4 CRLF Cell 1 of 1

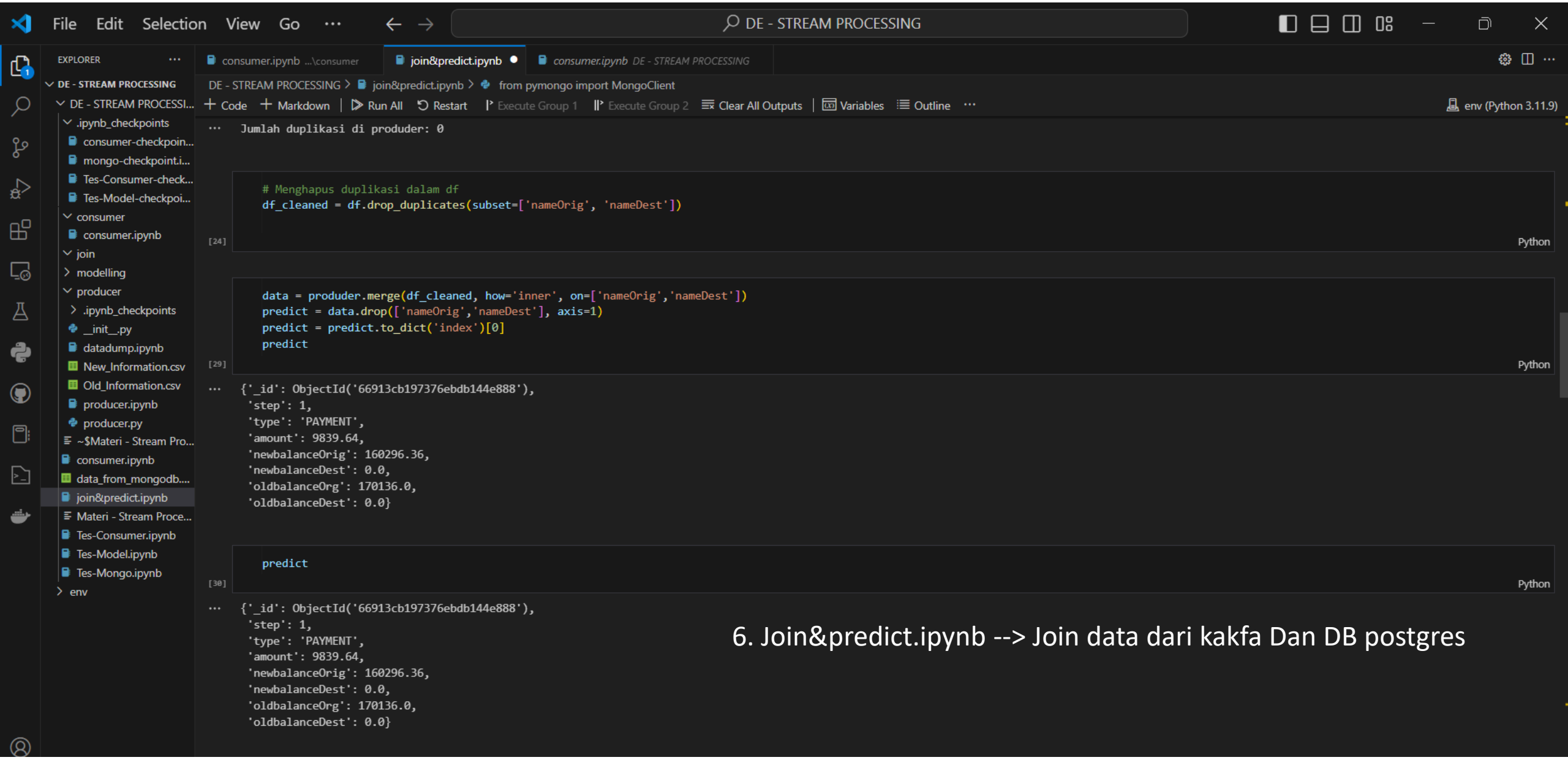


Consume messages dan simpan ke MongoDB

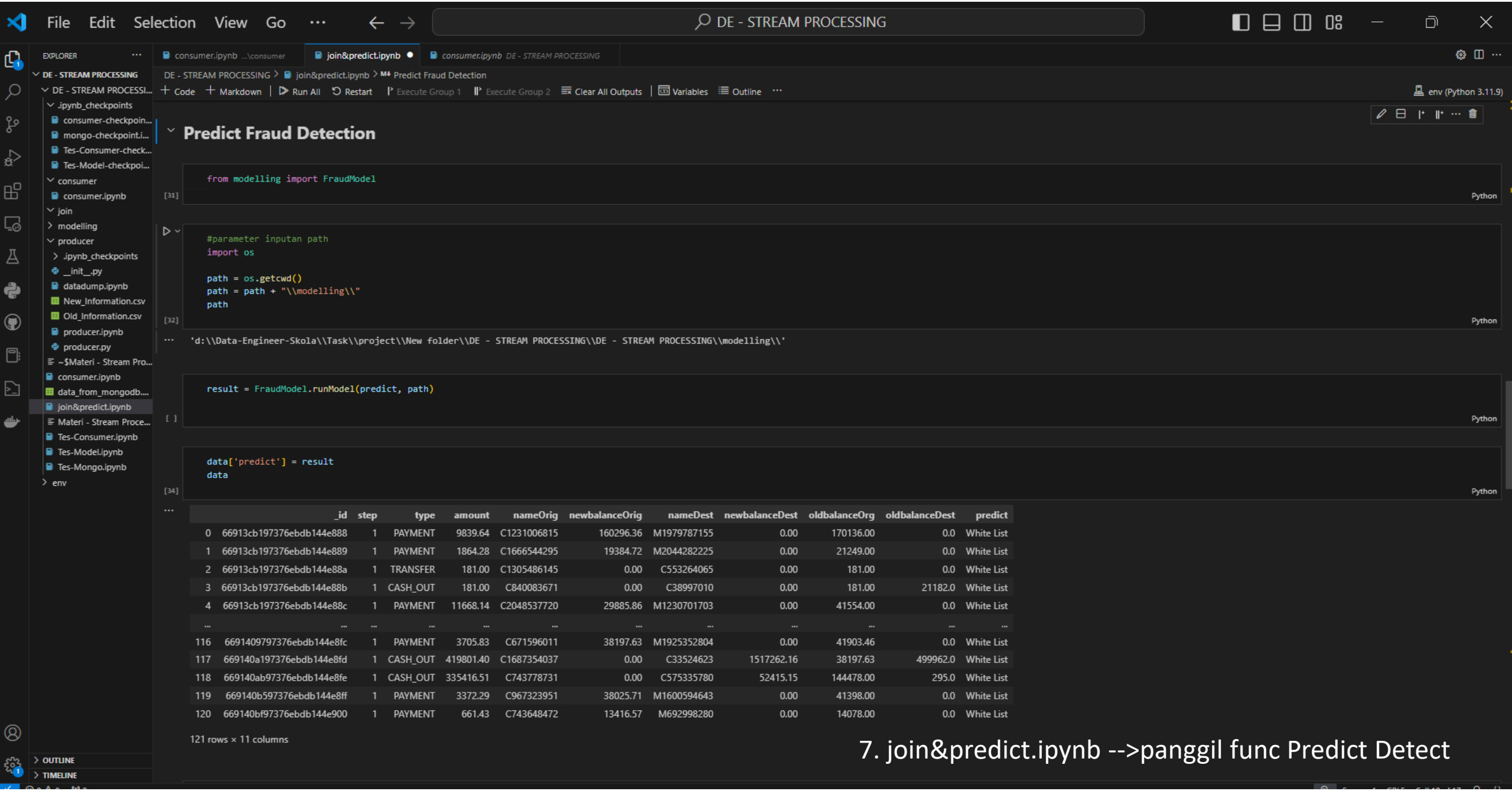


5. Join&predict.ipynb --> ambil data dalam database postgre


File Edit Selection View Go
DE - STREAM PROCESSING
consumer.ipynb
join&predict.ipynb
consumer.ipynb DE - STREAM PROCESSING
DE - STREAM PROCESSING > join&predict.ipynb > from pymongo import MongoClient
+ Code + Markdown | Run All Restart Execute Group 1 Execute Group 2 Clear All Outputs | Variables Outline
env (Python 3.11.9)
EXPLORER
DE - STREAM PROCESSING
.ipynb_checkpoints
consumer-checkpoin...
mongo-checkpoint.i...
Tes-Consumer-check...
Tes-Model-checkpoi...
consumer
consumer.ipynb
join
modelling
producer
.ipynb_checkpoints
__init__.py
datadump.ipynb
New_Information.csv
Old_Information.csv
producer.ipynb
producer.py
~$Materi - Stream Pro...
consumer.ipynb
data_from_mongodb....
join&predict.ipynb
Materi - Stream Proce...
Tes-Consumer.ipynb
Tes-Model.ipynb
Tes-Mongo.ipynb
env
OUTLINE
TIMELINE
from pymongo import MongoClient
import pandas as pd
# Menghubungkan ke MongoDB
mongo_client = MongoClient("mongodb://localhost:27017/")
db = mongo_client["ftde01"]
collection = db["project4-collection"]
# Mengambil data dari koleksi MongoDB
data = list(collection.find())
# Memasukkan data ke dalam DataFrame Pandas
produder = pd.DataFrame(data)
produder
[34]
Python
_id step type amount nameOrig newbalanceOrig nameDest newbalanceDest
0 66913cb197376ebdb144e888 1 PAYMENT 9839.64 C1231006815 160296.36 M1979787155 0.00
1 66913cb197376ebdb144e889 1 PAYMENT 1864.28 C1666544295 19384.72 M2044282225 0.00
2 66913cb197376ebdb144e88a 1 TRANSFER 181.00 C1305486145 0.00 C553264065 0.00
3 66913cb197376ebdb144e88b 1 CASH_OUT 181.00 C840083671 0.00 C38997010 0.00
4 66913cb197376ebdb144e88c 1 PAYMENT 11668.14 C2048537720 29885.86 M1230701703 0.00
116 6691409797376ebdb144e8fc 1 PAYMENT 3705.83 C671596011 38197.63 M1925352804 0.00
117 669140a197376ebdb144e8fd 1 CASH_OUT 419801.40 C1687354037 0.00 C33524623 1517262.16
118 669140ab97376ebdb144e8fe 1 CASH_OUT 335416.51 C743778731 0.00 C575335780 52415.15
119 669140b597376ebdb144e8ff 1 PAYMENT 3372.29 C967323951 38025.71 M1600594643 0.00
120 669140bf97376ebdb144e900 1 PAYMENT 661.43 C743648472 13416.57 M692998280 0.00
121 rows × 8 columns
5. Join&predict.ipynb --> ambil data dari consume kafka yang telah disimpan ke mongodb
# Menghitung duplikasi dalam df
dupes_count_df = df.duplicated(subset=['nameOrig', 'nameDest']).sum()
print(f"Jumlah duplikasi di df: {dupes_count_df}")
[22]
Python
Jumlah duplikasi di df: 45000
Cek duplicate data, ternyata database postgre terdapat data duplikat
# Menghitung duplikasi dalam produder
dupes_count_produder = produder.duplicated(subset=['nameOrig', 'nameDest']).sum()
print(f"Jumlah duplikasi di produder: {dupes_count_produder}")
[23]
Python
Jumlah duplikasi di produder: 0
# Menghapus duplikasi dalam df
df_cleaned = df.drop_duplicates(subset=['nameOrig', 'nameDest'])



6. Join&predict.ipynb --> Join data dari kakfa Dan DB postgres



	_id	step	type	amount	nameOrig	newbalanceOrig	nameDest	newbalanceDest	oldbalanceOrg	oldbalanceDest	predict
0	66913cb197376ebdb144e888	1	PAYMENT	9839.64	C1231006815	160296.36	M1979787155	0.00	170136.00	0.0	White List
1	66913cb197376ebdb144e889	1	PAYMENT	1864.28	C1666544295	19384.72	M2044282225	0.00	21249.00	0.0	White List
2	66913cb197376ebdb144e88a	1	TRANSFER	181.00	C1305486145	0.00	C553264065	0.00	181.00	0.0	White List
3	66913cb197376ebdb144e88b	1	CASH_OUT	181.00	C840083671	0.00	C38997010	0.00	181.00	21182.0	White List
4	66913cb197376ebdb144e88c	1	PAYMENT	11668.14	C2048537720	29885.86	M1230701703	0.00	41554.00	0.0	White List
...	...	...	...	...	...	...	...	...	...	...	...
116	6691409797376ebdb144e8fc	1	PAYMENT	3705.83	C671596011	38197.63	M1925352804	0.00	41903.46	0.0	White List
117	669140a197376ebdb144e8fd	1	CASH_OUT	419801.40	C1687354037	0.00	C33524623	1517262.16	38197.63	499962.0	White List
118	669140ab97376ebdb144e8fe	1	CASH_OUT	335416.51	C743778731	0.00	C575335780	52415.15	144478.00	295.0	White List
119	669140b597376ebdb144e8ff	1	PAYMENT	3372.29	C967323951	38025.71	M1600594643	0.00	41398.00	0.0	White List
120	669140bf97376ebdb144e900	1	PAYMENT	661.43	C743648472	13416.57	M692998280	0.00	14078.00	0.0	White List

121 rows × 11 columns

7. join&predict.ipynb -->panggil func Predict Detect

File Edit Selection View Go ··· DE - STREAM PROCESSING

EXPLORER

- DE - STREAM PROCESSING
  - DE - STREAM PROCESSING
    - .ipynb_checkpoints
      - consumer-checkpoint.ipynb
      - mongo-checkpoint.ipynb
      - Tes-Consumer-checkpoint.ipynb
      - Tes-Model-checkpoint.ipynb
    - modelling
    - producer
      - .ipynb_checkpoints
      - __init__.py
      - datadump.ipynb
      - New_Information.csv
      - Old_Information.csv
      - producer.ipynb
      - producer.py
    - ~$Materi - Stream Processing.pp...
    - consumer.ipynb
    - Materi - Stream Processing.pptx
    - Tes-Consumer.ipynb
    - Tes-Model.ipynb
    - Tes-Mongo.ipynb
  - env

datadump.ipynb | consumer.ipynb | producer.ipynb | Tes-Consumer.ipynb | producer.py | Tes-Model.ipynb | modelling.ipynb | Tes-Mongo.ipynb

DE - STREAM PROCESSING > Tes-Consumer.ipynb > Precit Fraud Detection > result

+ Code + Markdown | Run All | Restart | Execute Group 1 | Execute Group 2 | Clear All Outputs | Variables | Outline ··· env (Python 3.11.9)

3	1	PAYMENT	3671.32	C361380654	96422.68	M631673932	0.0	100094.0	0.0	White List

Inser MongoDB

```python
from pymongo import MongoClient

# Mengatur koneksi ke MongoDB lokal tanpa autentikasi
mongo_client = MongoClient("mongodb://localhost:27017/")

db = mongo_client["ftde01"]
collection = db["project4-collection"]

# Asumsikan 'data' adalah DataFrame pandas yang sudah didefinisikan sebelumnya

for _, row in data.iterrows():
    document = row.to_dict()
    collection.insert_one(document)

print("Data telah disimpan ke MongoDB")
```

[19] ✓ 0.0s Python

```
Data telah disimpan ke MongoDB
```

```python
mongo_client = MongoClient("mongodb://localhost:27017/")
db = mongo_client["ftde01"]
collection = db["project4-collection"]

# Membaca data dari MongoDB ke dalam list of dictionaries
data_from_mongo = list(collection.find())

# Membaca data ke dalam DataFrame
df = pd.DataFrame(data_from_mongo)
df
```

[21] ✓ 0.0s Python

	_id	step	type	amount	nameOrig	newbalanceOrig	nameDest	newbalanceDest	oldbalanceOrg	oldbalanceDest	predict
0	669128897460fec20029f857	1	PAYMENT	3671.32	C361380654	96422.68	M631673932	0.0	100094.0	0.0	White List
1	669128897460fec20029f858	1	PAYMENT	3671.32	C361380654	96422.68	M631673932	0.0	100094.0	0.0	White List
2	669128897460fec20029f859	1	PAYMENT	3671.32	C361380654	96422.68	M631673932	0.0	100094.0	0.0	White List
3	669128897460fec20029f85a	1	PAYMENT	3671.32	C361380654	96422.68	M631673932	0.0	100094.0	0.0	White List

8. Join&predict.ipynb --> insert hasil prediksi di mongodb

MongoDB Compass - localhost:27017/ftde-01-result.project4-result

Connect Edit View Collection Help

localhost:27017

My Queries
Performance
Databases

Search

admin
config
dskola
ftde-01-result
project4-result
ftde01
project4-collection
local
reactjs
testDatabase

My Queries | project4-collection | ftde01 | project4-result

localhost:27017 > ftde-01-result > project4-result

Documents 0 | Aggregations | Schema | Indexes 1 | Validation

Type a query: { field: 'value' } or Generate query

Explain | Reset | Find | </> | Options

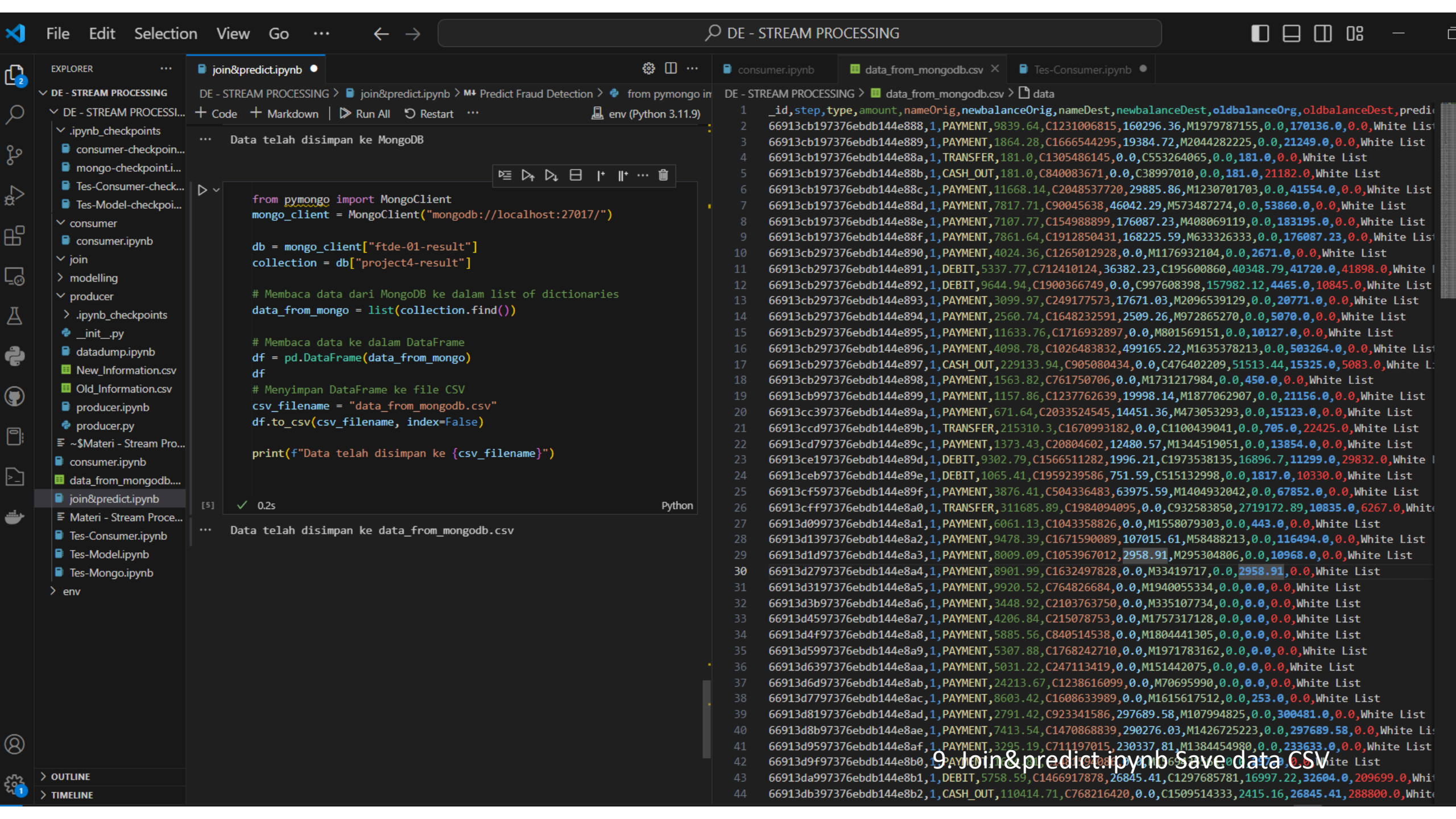
ADD DATA | EXPORT DATA | UPDATE | DELETE

1 – 20 of 121

```
_id: ObjectId('66913cb197376ebdb144e888')
step : 1
type : "PAYMENT"
amount : 9839.64
nameOrig : "C1231006815"
newbalanceOrig : 160296.36
nameDest : "M1979787155"
newbalanceDest : 0
oldbalanceOrg : 170136
oldbalanceDest : 0
predict : "White List"
```

```
_id: ObjectId('66913cb197376ebdb144e889')
step : 1
type : "PAYMENT"
amount : 1864.28
nameOrig : "C1666544295"
newbalanceOrig : 19384.72
nameDest : "M2044282225"
```

>_MONGOSH


File Edit Selection View Go
DE - STREAM PROCESSING
EXPLORER
DE - STREAM PROCESSING
DE - STREAM PROCESSI...
.ipynb_checkpoints
consumer-checkpoin...
mongo-checkpoint.i...
Tes-Consumer-check...
Tes-Model-checkpoi...
consumer
consumer.ipynb
join
modelling
producer
.ipynb_checkpoints
__init__.py
datadump.ipynb
New_Information.csv
Old_Information.csv
producer.ipynb
producer.py
~$Materi - Stream Pro...
consumer.ipynb
data_from_mongodb....
join&predict.ipynb
Materi - Stream Proce...
Tes-Consumer.ipynb
Tes-Model.ipynb
Tes-Mongo.ipynb
env
OUTLINE
TIMELINE
join&predict.ipynb
DE - STREAM PROCESSING > join&predict.ipynb > Predict Fraud Detection > from pymongo im
+ Code + Markdown | Run All Restart
env (Python 3.11.9)
Data telah disimpan ke MongoDB
from pymongo import MongoClient
mongo_client = MongoClient("mongodb://localhost:27017/")
db = mongo_client["ftde-01-result"]
collection = db["project4-result"]
# Membaca data dari MongoDB ke dalam list of dictionaries
data_from_mongo = list(collection.find())
# Membaca data ke dalam DataFrame
df = pd.DataFrame(data_from_mongo)
df
# Menyimpan DataFrame ke file CSV
csv_filename = "data_from_mongodb.csv"
df.to_csv(csv_filename, index=False)
print(f"Data telah disimpan ke {csv_filename}")
[5] 0.2s Python
Data telah disimpan ke data_from_mongodb.csv
consumer.ipynb
data_from_mongodb.csv
Tes-Consumer.ipynb
DE - STREAM PROCESSING > data_from_mongodb.csv > data
_id,step,type,amount,nameOrig,newbalanceOrig,nameDest,newbalanceDest,oldbalanceOrg,oldbalanceDest,predi
66913cb197376ebdb144e888,1,PAYMENT,9839.64,C1231006815,160296.36,M1979787155,0.0,170136.0,0.0,White Lis
66913cb197376ebdb144e889,1,PAYMENT,1864.28,C1666544295,19384.72,M2044282225,0.0,21249.0,0.0,White List
66913cb197376ebdb144e88a,1,TRANSFER,181.0,C1305486145,0.0,C553264065,0.0,181.0,0.0,White List
66913cb197376ebdb144e88b,1,CASH_OUT,181.0,C840083671,0.0,C38997010,0.0,181.0,21182.0,White List
66913cb197376ebdb144e88c,1,PAYMENT,11668.14,C2048537720,29885.86,M1230701703,0.0,41554.0,0.0,White List
66913cb197376ebdb144e88d,1,PAYMENT,7817.71,C90045638,46042.29,M573487274,0.0,53860.0,0.0,White List
66913cb197376ebdb144e88e,1,PAYMENT,7107.77,C154988899,176087.23,M408069119,0.0,183195.0,0.0,White List
66913cb197376ebdb144e88f,1,PAYMENT,7861.64,C1912850431,168225.59,M633326333,0.0,176087.23,0.0,White Lis
66913cb297376ebdb144e890,1,PAYMENT,4024.36,C1265012928,0.0,M1176932104,0.0,2671.0,0.0,White List
66913cb297376ebdb144e891,1,DEBIT,5337.77,C712410124,36382.23,C195600860,40348.79,41720.0,41898.0,White
66913cb297376ebdb144e892,1,DEBIT,9644.94,C1900366749,0.0,C997608398,157982.12,4465.0,10845.0,White List
66913cb297376ebdb144e893,1,PAYMENT,3099.97,C249177573,17671.03,M2096539129,0.0,20771.0,0.0,White List
66913cb297376ebdb144e894,1,PAYMENT,2560.74,C1648232591,2509.26,M972865270,0.0,5070.0,0.0,White List
66913cb297376ebdb144e895,1,PAYMENT,11633.76,C1716932897,0.0,M801569151,0.0,10127.0,0.0,White List
66913cb297376ebdb144e896,1,PAYMENT,4098.78,C1026483832,499165.22,M1635378213,0.0,503264.0,0.0,White Lis
66913cb297376ebdb144e897,1,CASH_OUT,229133.94,C905080434,0.0,C476402209,51513.44,15325.0,5083.0,White L
66913cb297376ebdb144e898,1,PAYMENT,1563.82,C761750706,0.0,M1731217984,0.0,450.0,0.0,White List
66913cb997376ebdb144e899,1,PAYMENT,1157.86,C1237762639,19998.14,M1877062907,0.0,21156.0,0.0,White List
66913cc397376ebdb144e89a,1,PAYMENT,671.64,C2033524545,14451.36,M473053293,0.0,15123.0,0.0,White List
66913ccd97376ebdb144e89b,1,TRANSFER,215310.3,C1670993182,0.0,C1100439041,0.0,705.0,22425.0,White List
66913cd797376ebdb144e89c,1,PAYMENT,1373.43,C20804602,12480.57,M1344519051,0.0,13854.0,0.0,White List
66913ce197376ebdb144e89d,1,DEBIT,9302.79,C1566511282,1996.21,C1973538135,16896.7,11299.0,29832.0,White
66913ceb97376ebdb144e89e,1,DEBIT,1065.41,C1959239586,751.59,C515132998,0.0,1817.0,10330.0,White List
66913cf597376ebdb144e89f,1,PAYMENT,3876.41,C504336483,63975.59,M1404932042,0.0,67852.0,0.0,White List
66913cff97376ebdb144e8a0,1,TRANSFER,311685.89,C1984094095,0.0,C932583850,2719172.89,10835.0,6267.0,Whit
66913d0997376ebdb144e8a1,1,PAYMENT,6061.13,C1043358826,0.0,M1558079303,0.0,443.0,0.0,White List
66913d1397376ebdb144e8a2,1,PAYMENT,9478.39,C1671590089,107015.61,M58488213,0.0,116494.0,0.0,White List
66913d1d97376ebdb144e8a3,1,PAYMENT,8009.09,C1053967012,2958.91,M295304806,0.0,10968.0,0.0,White List
66913d2797376ebdb144e8a4,1,PAYMENT,8901.99,C1632497828,0.0,M33419717,0.0,2958.91,0.0,White List
66913d3197376ebdb144e8a5,1,PAYMENT,9920.52,C764826684,0.0,M1940055334,0.0,0.0,0.0,White List
66913d3b97376ebdb144e8a6,1,PAYMENT,3448.92,C2103763750,0.0,M335107734,0.0,0.0,0.0,White List
66913d4597376ebdb144e8a7,1,PAYMENT,4206.84,C215078753,0.0,M1757317128,0.0,0.0,0.0,White List
66913d4f97376ebdb144e8a8,1,PAYMENT,5885.56,C840514538,0.0,M1804441305,0.0,0.0,0.0,White List
66913d5997376ebdb144e8a9,1,PAYMENT,5307.88,C1768242710,0.0,M1971783162,0.0,0.0,0.0,White List
66913d6397376ebdb144e8aa,1,PAYMENT,5031.22,C247113419,0.0,M151442075,0.0,0.0,0.0,White List
66913d6d97376ebdb144e8ab,1,PAYMENT,24213.67,C1238616099,0.0,M70695990,0.0,0.0,0.0,White List
66913d7797376ebdb144e8ac,1,PAYMENT,8603.42,C1608633989,0.0,M1615617512,0.0,253.0,0.0,White List
66913d8197376ebdb144e8ad,1,PAYMENT,2791.42,C923341586,297689.58,M107994825,0.0,300481.0,0.0,White List
66913d8b97376ebdb144e8ae,1,PAYMENT,7413.54,C1470868839,290276.03,M1426725223,0.0,297689.58,0.0,White Li
66913d9597376ebdb144e8af,1,PAYMENT,3295.19,C711197015,230337.81,M1384454980,0.0,233633.0,0.0,White List
66913d9f97376ebdb144e8b0,1,PAYMENT,[illegible],0.0,White List
66913da997376ebdb144e8b1,1,DEBIT,5758.59,C1466917878,26845.41,C1297685781,16997.22,32604.0,209699.0,Whi
66913db397376ebdb144e8b2,1,CASH_OUT,110414.71,C768216420,0.0,C1509514333,2415.16,26845.41,288800.0,Whit
9.Join&predict.ipynb Save data CSV