

The background is a light blue gradient with several realistic water droplets of various sizes scattered across the surface. The droplets have highlights and shadows, giving them a three-dimensional appearance.

FINAL PROJECT

MUH SANVI FAHRUDIN

LATAR BELAKANG

DALAM PROYEK KALI INI, ANDA DIHARAPKAN MAMPU UNTUK MENDESIGN SEBUAH DATAWAREHOUSE

MENGGUNAKAN STAR SCHEMA DARI DATA YANG DISIAPKAN. DARI DATA YANG ADA, ANALISA PENGARUH CURAH

HUJAN DAN SUHU TERHADAP REVIEW / RATING DI RESTORAN.

SUMBER DATA

1. DATA CLIMATE BERUPA PRECIPITATION DALAM BENTUK CSV
2. DATA CLIMATE BERUPA TEMPERATURE DALAM BENTUK CSV
3. DATA BUSINESS RESTORAN DALAM BENTUK JSON
4. DATA CHECKIN RESTORAN DALAM BENTUK JSON
5. DATA REVIEW RESTORAN DALAM BENTUK JSON
6. DATA TIP RESTORAN DALAM BENTUK JSON
7. DATA USER RESTORAN DALAM BENTUK JSON

LANGKAH Pengerjaan

- 1) BUATLAH DOCKER COMPOSE (AIRFLOW, POSTGRESQL) DI LOCAL COMPUTER ANDA.
- 2) BUATLAH DATA ARCHITECTURE DIAGRAM YANG DAPAT MENGGAMBARAKAN BAGAIMANA CARA ANDA MENGEKTRACT RAW DATA JSON & CSV KEDALAM DATABASE POSTGRESQL.
- 3) CASE DIATAS, BUATKAN DATAFLOW DIAGRAM YANG MENGGAMBARAKAN DATAWAREHOUSE LAYER.
- 4) GAMBARAKAN / BUATKAN ER DIAGRAM (ENTITY RELATIONSHIP) YANG MENGGAMBARAKAN HUBUNGAN ANTARA TABLE – TABLE YANG DI INGEST KE MYSQL (ODS LAYER)
- 5) DARI RAW DATA TERSEBUT, BUAT DATA MODELING MENGGUNAKAN KIMBAL STAR SCHEMA. TENTUKAN MANA YANG MERUPAKAN FACT TABLE, MANA YANG MERUPAKAN DIM TABLE DAN GAMBARAKAN DIAGRAMNYA.
- 6) BUATLAH SATU DAG DI AIRFLOW DENGAN TASK :
 - A. EXTRACT LOAD DATA DARI RAW DATA KE ODS LAYER
 - B. EXTRACT DAN LOAD DATA DARI ODS LAYER KE DWH LAYER (DIM & FACT)
 - C. TRANSFORM DATA DARI DWH LAYER KE SERVING LAYER
- 7) PADA SERVING LAYER BUAT SATU TABLE AGREGASI YANG MELAKUKAN JOIN FACT DAN DIM TABLE YELP DATA DAN CLIMATE DATA
- 8) (QUERY BISA DIJALANKAN SEBAGAI TASK AIRFLOW PADA POIN 6C)
- 9) BUATLAH SEBUAH ANALISA DARI TABLE AGREGASI PADA POIN 7, UNTUK MENCARI KORELASI ANTARA REVIEW, TIPS DAN CUACA.
- 10) NILAI TAMBAH JIKA MAMPU MENAMPILKAN HASIL ANALISA KEDALAM BENTUK VISUAL
- 11) COMMIT CODE YANG DIBUAT KE GITHUB ANDA MASING

1. DOCKER COMPOSE

```
v-fp > docker-compose.yaml
# for other purpose (development, test and especially production usage) build/
_PIP_ADDITIONAL_REQUIREMENTS: ${_PIP_ADDITIONAL_REQUIREMENTS:-}
volumes:
  - ${AIRFLOW_PROJ_DIR:-.}/dags:/opt/airflow/dags
  - ${AIRFLOW_PROJ_DIR:-.}/logs:/opt/airflow/logs
  - ${AIRFLOW_PROJ_DIR:-.}/plugins:/opt/airflow/plugins
  # - ${AIRFLOW_PROJ_DIR:-.}/sql:/opt/airflow/sql
user: "${AIRFLOW_UID:-50000}:0"
depends_on:
  &airflow-common-depends-on
  redis:
    condition: service_healthy
  postgres:
    condition: service_healthy

services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow
      POSTGRES_DB: airflow
    volumes:
      - postgres-db-volume:/var/lib/postgresql/data
    ports:
      - "5432:5432"
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "airflow"]
      interval: 10s
      retries: 5
      start_period: 5s
      restart: always
```

Containers [Give feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

☐ Only show running containers

Search

<input type="checkbox"/>	Name	Image	Status	Port(s)	Last started	Actions
<input checked="" type="checkbox"/>	airflow-fp	-	Running (6/7)		35 minutes ago	
<input type="checkbox"/>	postgres-1 837907ac5069	postgres:13	Running	5432:5432	10 hours ago	
<input type="checkbox"/>	airflow-triggerer-1 df7a00166206	apache/airflow:2.5.2	Running		10 hours ago	
<input type="checkbox"/>	airflow-scheduler-1 9992bccd8f5b	apache/airflow:2.5.2	Running		10 hours ago	
<input type="checkbox"/>	airflow-webserver-1 c29e55242ce7	apache/airflow:2.5.2	Running	8080:8080	40 minutes ago	
<input type="checkbox"/>	airflow-worker-1 070f7bf9e1a1	apache/airflow:2.5.2	Running		35 minutes ago	
<input type="checkbox"/>	airflow-init-1 e74118e93b58	apache/airflow:2.5.2	Exited (255)		10 hours ago	
<input type="checkbox"/>	redis-1 f07aff6cd09c	redis:latest	Running		10 hours ago	

Showing 8 items

RAM 1.73 GB CPU 31.17% Disk 73.21 GB avail. of 87.45 GB Connected to Hub v4.17.0

2. ARCHITECTURE EXTRACT RAW DATA TO DATABASE

CLIMATE DATA

–
PRECIPITATION
–
TEMPERATURE

RESTAURANT DATA

–BUSINESS
–CHECKIN
–REVIEW
–TIP
–USER

RAW DATA



BUILD AIRFLOW
FROM
DOCKER AND
CREATE DAG
TO ETL
PROCESS

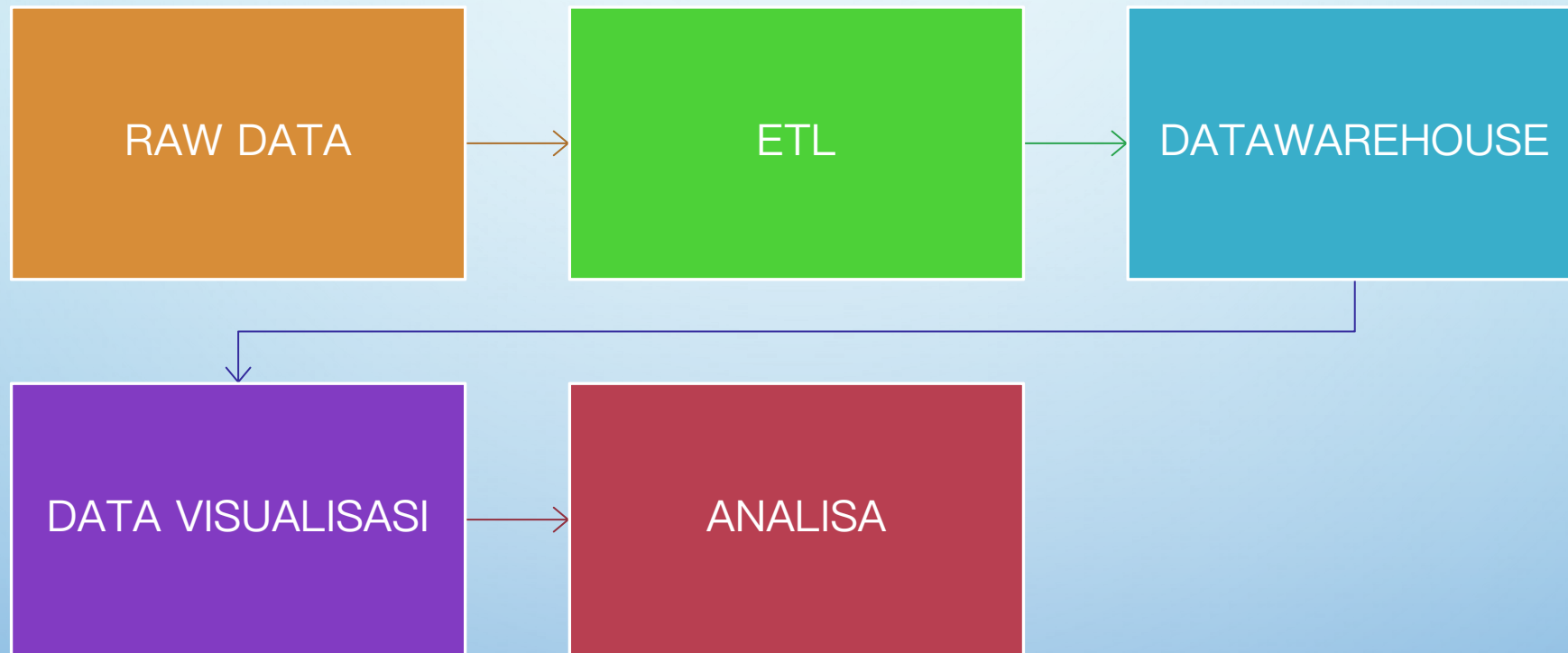


IMPORT PYTHON
OPERATOR
TO USING PANDAS,
SQLALCHEMY,
AND PSYCOPG2

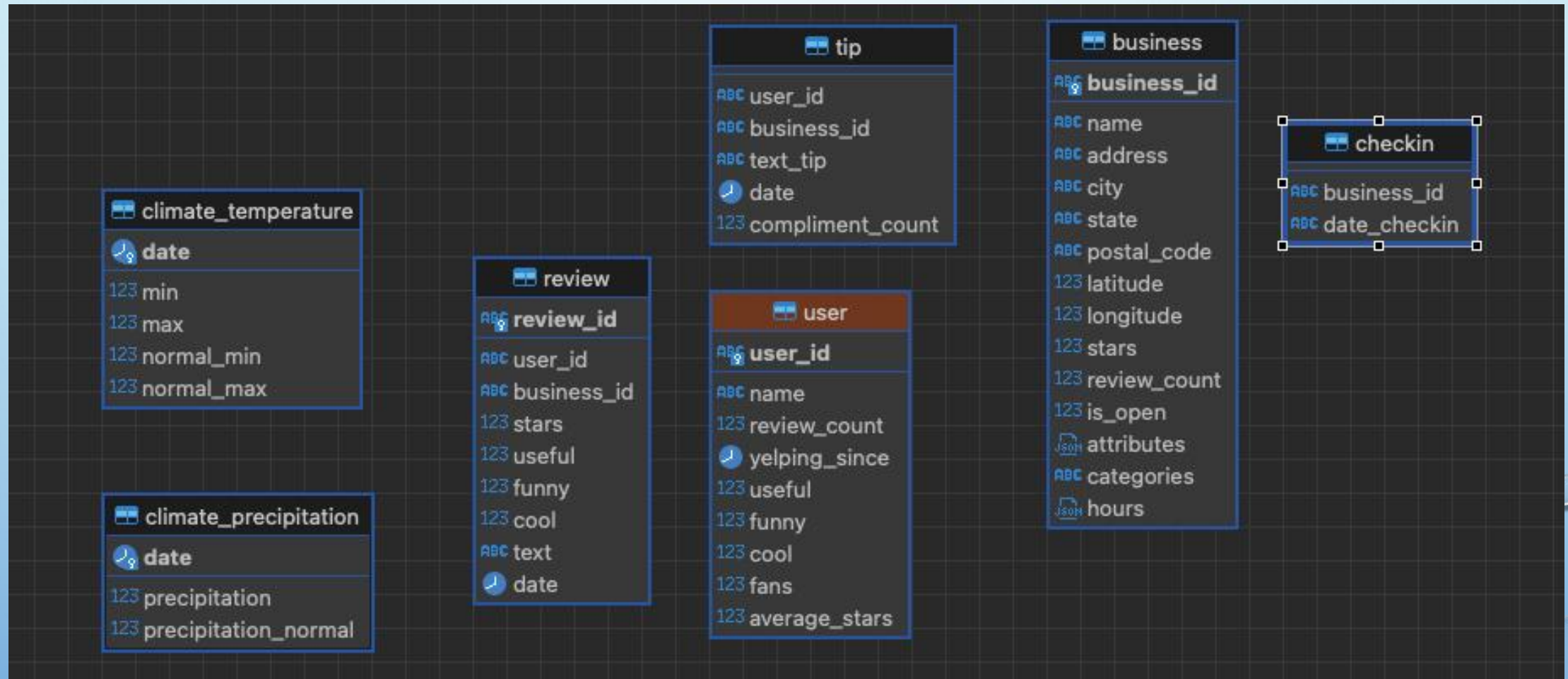


LOAD DATA TO
DATABASE
POSTGRESQL AND
OPEN USING
DBEAVER

3. DATAWAREHOUSE LAYER



4. ER DIAGRAM (ENTITY RELATIONSHIP)



Dim

5. DIAGRAM STAR SCHEMA

Dim

Fact

Dim

Dim

climate_temperature
date
123 min
123 max
123 normal_min
123 normal_max

climate_precipitation
date
123 precipitation
123 precipitation_normal

user
user_id
name
123 review_count
yelping_since
123 useful
123 funny
123 cool
123 fans
123 average_stars

checkin
business_id
date_checkin

review
review_id
user_id
business_id
123 stars
123 useful
123 funny
123 cool
text
date

tip
user_id
business_id
text_tip
date
123 compliment_count

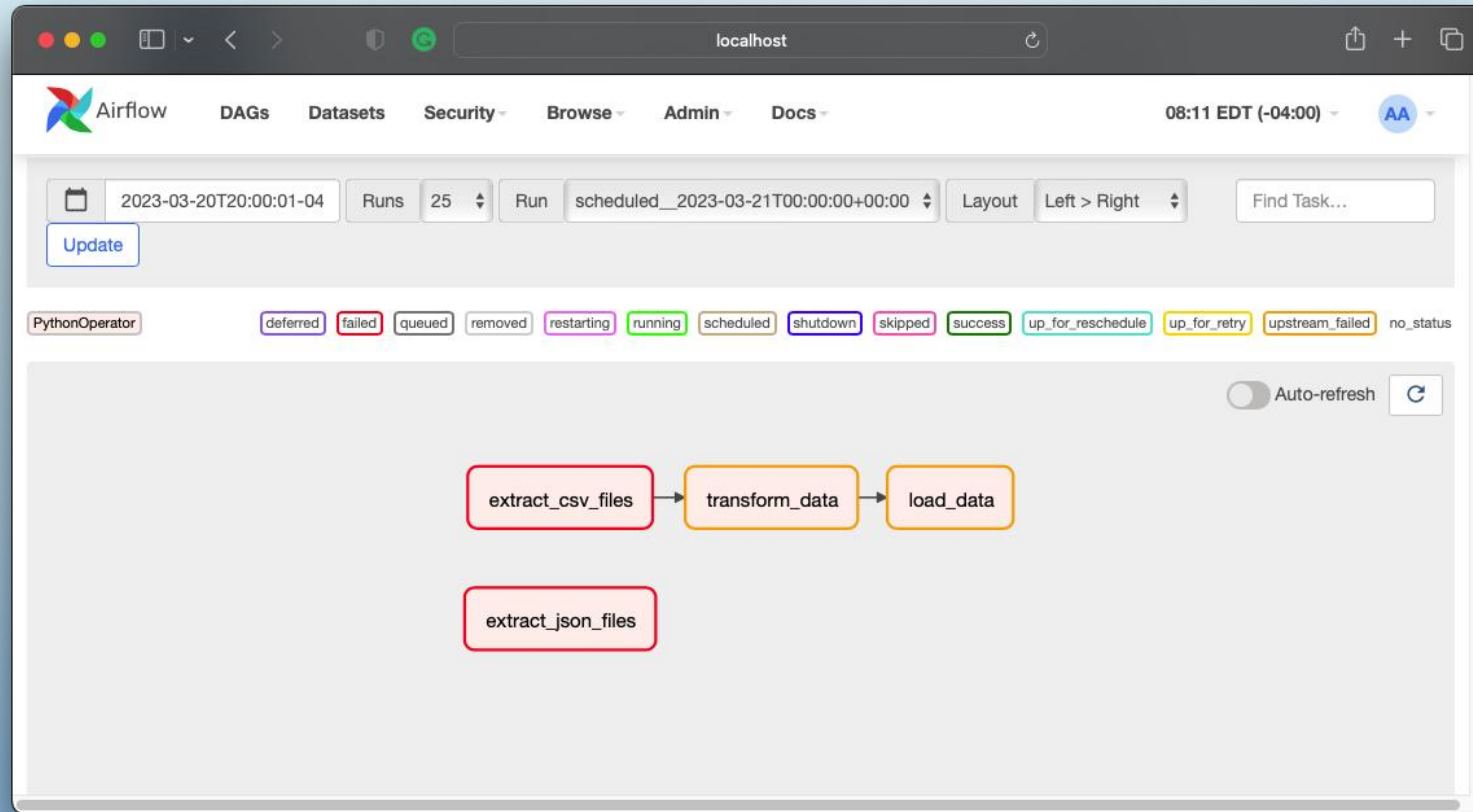
business
business_id
name
address
city
state
postal_code
123 latitude
123 longitude
123 stars
123 review_count
123 is_open
attributes
categories
hours

6. DAG AIRFLOW ETL PROCESS

```
1 # Import airflow modules
2 from datetime import datetime, timedelta
3 from airflow import DAG
4 from airflow.operators.python_operator import PythonOperator
5 from airflow.operators.postgres_operator import PostgresOperator
6
7 # Import modules
8 import pandas as pd
9 import json
10 import psycpg2
11 import os
12
13 # Arguments
14 default_args = {
15     'owner': 'Sanvi',
16     'depends_on_past': False,
17     'start_date': datetime(2023, 3, 21),
18     'retries': 1,
19     'retry_delay': timedelta(minutes=5)
20 }
21
22 dag = DAG(
23     'csv_json_to_postgres',
24     default_args=default_args,
25     description='Extract CSV and JSON files, transform data, and load into Postgres',
26     schedule_interval='@once'
27 )
28
29 # Extract CSV files
30 cwd = os.getcwd() # Get the current working directory (cwd)
31 files = os.listdir(cwd) # Get all the files in that directory
32 def extract_csv_files():
33     df1 = pd.read_csv('/Users/sanvi/Documents/final project/airflow-fp/dags/data/precipitation.csv')
34     df2 = pd.read_csv('/Users/sanvi/Documents/final project/airflow-fp/dags/data/temperature.csv')
```

```
85     , item['date'], item['value1'], item['value2']
86
87     # Commit changes and close connection
88     conn.commit()
89     cur.close()
90     conn.close()
91
92 # Define DAG tasks
93 extract_csv_task = PythonOperator(
94     task_id='extract_csv_files',
95     python_callable=extract_csv_files,
96     dag=dag
97 )
98
99 extract_json_task = PythonOperator(
100     task_id='extract_json_files',
101     python_callable=extract_json_files,
102     dag=dag
103 )
104
105 transform_task = PythonOperator(
106     task_id='transform_data',
107     python_callable=transform_data,
108     dag=dag
109 )
110
111 load_task = PythonOperator(
112     task_id='load_data',
113     python_callable=load_data,
114     dag=dag
115 )
116
117 # Set task dependencies
118 extract_csv_task >> transform_task >> load_task
```

6. DAG AIRFLOW ETL PROCESS

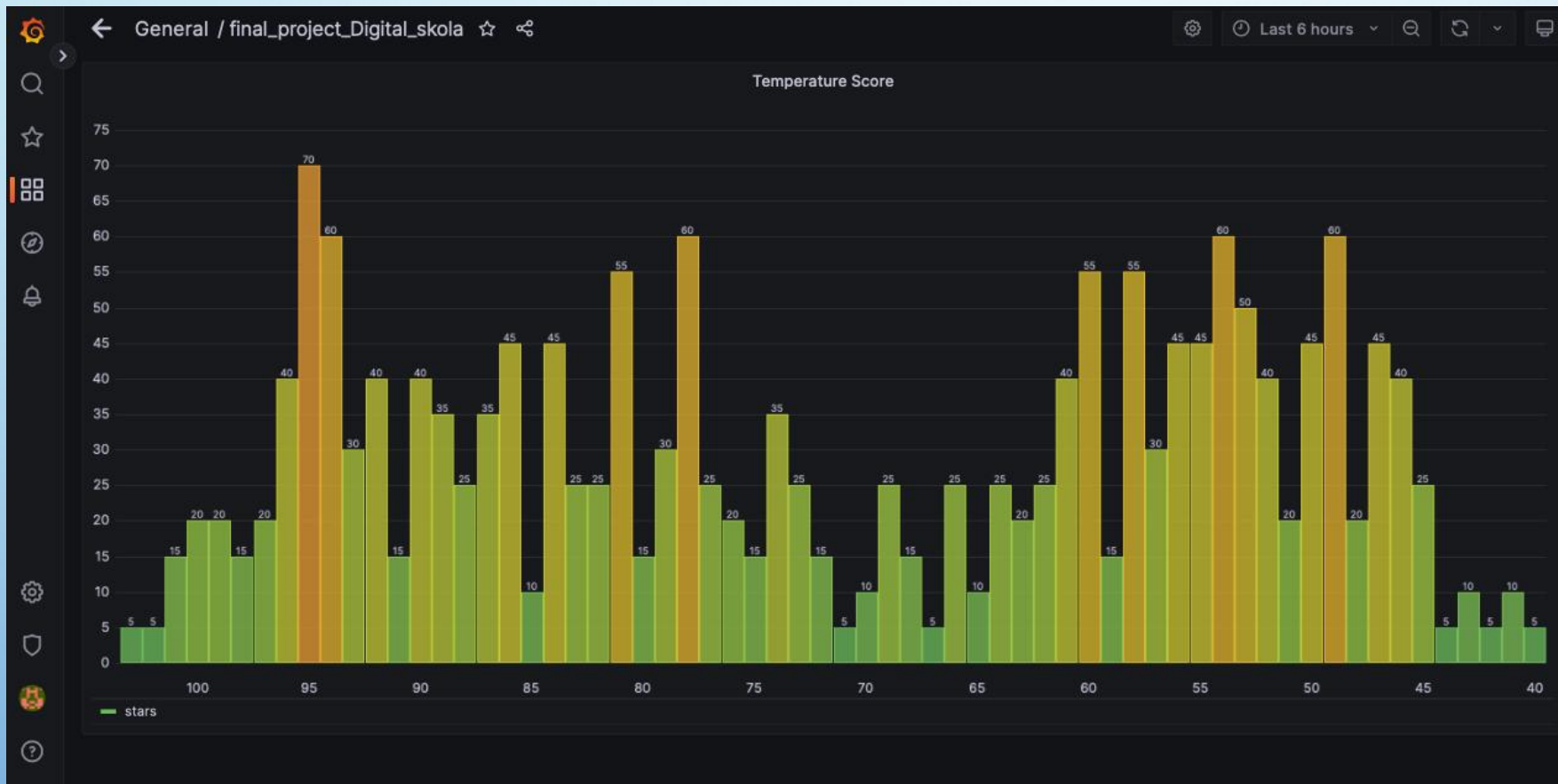


7. SATU TABLE AGREGASI YANG MELAKUKAN JOIN FACT DAN DIM TABLE YELP DATA DAN CLIMATE DATA

The screenshot shows a database management tool interface. On the left, the 'Database Navigator' pane displays a tree structure of a PostgreSQL database. The 'public' schema is expanded, showing tables including 'aggregation' (2.1M rows), 'business' (108M), 'checkin' (146M), 'climate_precipitation' (2.1M), 'climate_temperature' (2.4M), 'fact_review' (4.9G), 'tip' (130M), and 'user' (276M). The main window displays the 'Data' tab for the 'aggregation' table. The table has 10 columns: 'date', 'stars', 'count', 'min', 'max', 'normal_min', 'normal', and 'precipitation'. The data is presented in a grid view with 25 rows. The status bar at the bottom indicates '200 row(s) fetched - 26ms (2ms fetch), on 2023-03-24 at 20:10:27'.

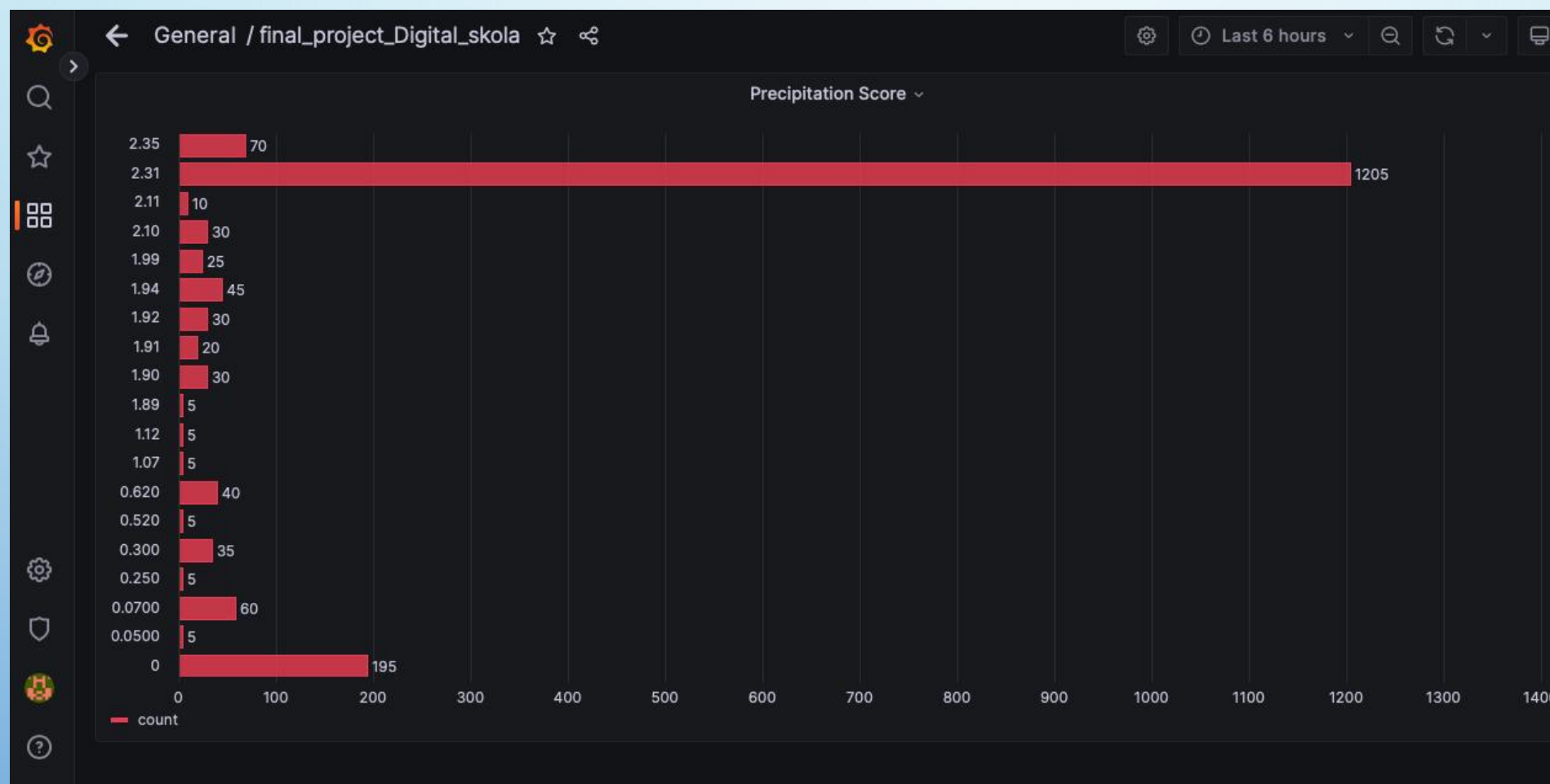
	date	stars	count	min	max	normal_min	normal	precipitation
1	2005-02-16	5	1	64	49	63	44	2.9
2	2005-02-16	4	2	64	49	63	44	2.9
3	2005-03-01	5	3	65	46	67	47	4.5
4	2005-03-01	2	1	65	46	67	47	4.5
5	2005-03-01	4	2	65	46	67	47	4.5
6	2005-03-01	3	1	65	46	67	47	4.5
7	2005-03-02	2	1	68	46	67	47	4.5
8	2005-03-02	4	1	68	46	67	47	4.5
9	2005-03-04	3	1	63	50	68	48	4.6
10	2005-03-04	5	2	63	50	68	48	4.6
11	2005-03-04	2	1	63	50	68	48	4.6
12	2005-03-08	5	2	78	51	69	49	4.6
13	2005-03-08	4	1	78	51	69	49	4.6
14	2005-03-08	3	1	78	51	69	49	4.6
15	2005-03-08	2	1	78	51	69	49	4.6
16	2005-03-09	4	4	81	54	69	49	4.6
17	2005-03-09	3	1	81	54	69	49	4.6
18	2005-03-09	1	1	81	54	69	49	4.6
19	2005-03-09	2	2	81	54	69	49	4.6
20	2005-03-09	5	4	81	54	69	49	4.6
21	2005-03-10	3	3	82	55	70	49	4.6
22	2005-03-10	2	1	82	55	70	49	4.6
23	2005-03-10	5	3	82	55	70	49	4.6
24	2005-03-10	4	1	82	55	70	49	4.6
25	2005-03-10	1	1	82	55	70	49	4.6

8. KORELASI ANTARA REVIEW, TIPS DAN CUACA



VISUALISASI DATA
MENGUNAKAN TOOLS
GRAFANA. BAR CAHRT
TERSEBUT
MENUNJUKKAN
KORELASI TOTAL
BINTANG(REVIEW) YANG
DIDAPAT PADA Y AXIS
DAN SUHU PADA X AXIS

8. KORELASI ANTARA REVIEW, TIPS DAN CUACA



VISUALISASI DATA MENGGUNAKAN TOOLS *GRAFANA*. BAR CHART TERSEBUT MENUNJUKKAN KORELASI TOTAL BINTANG(REVIEW) YANG DIDAPAT PADA X AXIS DAN PERCIPITATION PADA Y AXIS

KESIMPULAN

HUBUNGAN ANTARA REVIEW BERUPA TOTAL BINTANG PADA RESTORAN DENGAN SUHU ADALAH PADA SUHU 95° MENDAPATKAN TOTAL BINTANG PALING BANYAK SEBANYAK 70 SEHINGGA BANYAK PENGUNJUNG YANG LEBIH MENYUKAI RESTORAN DENGAN SUHU PANAS

HUBUNGAN ANTARA REVIEW BERUPA TOTAL BINTANG PADA RESTORAN DENGAN PRECIPITATION ADALAH SEMAKIN TINGGI NILAI RECIPITATION, MAKA SEMAKIN DISUKAI OLEH PENGUNJUNG, HAL TERSEBUT DAPAT DILIHAT PADA NILAI PRECIPITATION 2.31 MENDAPATKAN TOTAL BINTANG TERTINGGI YAITU 1205

9. COMMIT CODE YANG DIBUAT KE GITHUB

[HTTPS://GITHUB.COM/SANVIF-DOT/FINAL_PROJECT_AIRFLOW_DE](https://github.com/SANVIF-DOT/FINAL_PROJECT_AIRFLOW_DE)