🦢 Sesi 2

DuckDB & Visualisasi

Hari 1 - 120 menit

Bagian A: DuckDB (60 menit) | Bagian B: Visualisasi (60 menit)

Bagian A: Dasar DuckDB

60 menit

🦆 Apa itu DuckDB?

DuckDB = database SQL OLAP in-process

- 🚀 **Cepat:** Dioptimalkan untuk analytical queries
- 💻 **Embedded:** Tanpa server
- 🐍 **Ramah Python:** Integrasi native
- 📊 **Kolumnar:** Efisien untuk analytics
- 🔧 **SQL:** Sintaks familiar

Anggap saja: SQLite untuk analytics

🎯 Mengapa DuckDB?

Pandas vs DuckDB

Aspek	Pandas	DuckDB
Sintaks	Python (imperatif)	SQL (deklaratif)
Kecepatan	Cepat	**Lebih cepat** (data besar)
Memori	Semua di memori	Dioptimalkan
Keakraban	Dev Python	Analis SQL

Best Practice: Pakai keduanya! Pandas untuk manipulasi, DuckDB untuk analisis

1️⃣ Setup DuckDB

```
import duckdb
import pandas as pd

# Buat koneksi in-memory
conn = duckdb.connect(':memory:')

# Load DataFrame
df = pd.read_parquet('data.parquet')

# Register DataFrame jadi tabel
conn.register('rup', df)

# Sekarang bisa query dengan SQL!
```

2️⃣ Query Dasar

```sql
-- Select semua
SELECT * FROM rup LIMIT 10;

-- Pilih kolom tertentu
SELECT nama_paket, pagu, metode_pengadaan
FROM rup
LIMIT 5;

-- Hitung baris
SELECT COUNT(*) as total FROM rup;
```

Eksekusi di Python:

```python
result = conn.execute("SELECT * FROM rup LIMIT 10").df()
```

3 Filtering dengan WHERE

```sql
-- Kondisi tunggal
SELECT * FROM rup
WHERE pagu > 1000000000;

-- Banyak kondisi (AND)
SELECT * FROM rup
WHERE pagu > 1000000000
  AND metode_pengadaan = 'Tender';

-- Banyak kondisi (OR)
SELECT * FROM rup
WHERE metode_pengadaan = 'Tender'
   OR metode_pengadaan = 'Seleksi';
```

4 Fungsi Agregasi

```sql
-- Count
SELECT COUNT(*) as jumlah_paket FROM rup;

-- Sum
SELECT SUM(pagu) / 1e9 as total_pagu_miliar FROM rup;

-- Rata-rata
SELECT AVG(pagu) / 1e6 as rata_pagu_juta FROM rup;

-- Min & Max
SELECT
    MIN(pagu) as pagu_min,
    MAX(pagu) as pagu_max
FROM rup;
```

5 GROUP BY

```sql
-- Group by satu kolom
SELECT metode_pengadaan,
       COUNT(*) as jumlah_paket,
       SUM(pagu) / 1e9 as total_pagu_miliar
FROM rup
GROUP BY metode_pengadaan
ORDER BY total_pagu_miliar DESC;
```

🔢 GROUP BY Banyak Kolom

```
SELECT metode_pengadaan,
       jenis_pengadaan,
       COUNT(*) as jumlah,
       ROUND(SUM(pagu) / 1e9, 2) as total_miliar
FROM rup
WHERE jenis_pengadaan IS NOT NULL
GROUP BY metode_pengadaan, jenis_pengadaan
ORDER BY total_miliar DESC;
```

6 Klausa HAVING

WHERE = filter SEBELUM grouping

HAVING = filter SESUDAH grouping

```
SELECT nama_satker,
       COUNT(*) as jumlah_paket,
       SUM(pagu) / 1e9 as total_miliar
FROM rup
GROUP BY nama_satker
HAVING SUM(pagu) > 10000000000
ORDER BY total_miliar DESC;
```

7 ORDER BY

```sql
-- Menaik (default)
SELECT * FROM rup
ORDER BY pagu ASC
LIMIT 10;

-- Menurun
SELECT * FROM rup
ORDER BY pagu DESC
LIMIT 10;

-- Banyak kolom
SELECT * FROM rup
ORDER BY metode_pengadaan ASC, pagu DESC;
```

8 Pernyataan CASE

```
SELECT
    nama_paket,
    pagu,
    CASE
        WHEN pagu < 10000000 THEN 'Kecil'
        WHEN pagu < 100000000 THEN 'Menengah'
        WHEN pagu < 1000000000 THEN 'Besar'
        ELSE 'Sangat Besar'
    END as kategori
FROM rup
LIMIT 10;
```

9️⃣ Fungsi String

```sql
-- Pencarian
SELECT * FROM rup
WHERE LOWER(nama_paket) LIKE '%belanja%';

-- Panjang
SELECT nama_paket, LENGTH(nama_paket) as panjang
FROM rup
ORDER BY panjang DESC
LIMIT 10;

-- Upper/Lower
SELECT UPPER(metode_pengadaan) as metode
FROM rup
LIMIT 5;
```

🔟 Ekspor Hasil

```
-- Export ke CSV
COPY (
    SELECT metode_pengadaan,
           COUNT(*) as jumlah,
           SUM(pagu) / 1e9 as total_miliar
    FROM rup
    GROUP BY metode_pengadaan
) TO 'summary.csv' (HEADER, DELIMITER ',');
```

Atau lewat Python:

```
df_result = conn.execute("SELECT ...").df()
df_result.to_csv('output.csv', index=False)
```

🎯 Contoh Lengkap

```sql
-- Top 10 Satker dengan analisis lengkap
SELECT
    nama_satker,
    COUNT(*) as jumlah_paket,
    ROUND(SUM(pagu) / 1e9, 2) as total_miliar,
    ROUND(AVG(pagu) / 1e6, 2) as rata_juta,
    SUM(CASE WHEN status_pdn = 'PDN' THEN 1 ELSE 0 END) as paket_pdn,
    SUM(CASE WHEN status_ukm = 'UKM' THEN 1 ELSE 0 END) as paket_ukm
FROM rup
GROUP BY nama_satker
ORDER BY total_miliar DESC
LIMIT 10;
```

Bagian B: Visualisasi Data

60 menit

📊 Apa itu Plotly?

Plotly = library visualisasi interaktif

- 🎨 Chart modern & menarik
- 🖱️ Interaktif (zoom, hover, pan)
- 🌐 Siap web (output HTML)
- 📱 Mobile-friendly
- 🚀 Mudah digunakan

Sintaks: Mirip Matplotlib tapi interaktif!

🎯 Mengapa Plotly?

Matplotlib vs Plotly

Fitur	Matplotlib	Plotly
Interaktif	❌ Statik	✅ Interaktif
Modern	Klasik	✅ Modern
Web	PNG/PDF	✅ HTML
Belajar	Curam	✅ Mudah
Dashboard	Terbatas	✅ Sangat cocok

Untuk Streamlit: Plotly pilihan terbaik!

1️⃣ Setup Plotly

```
import plotly.express as px
import plotly.graph_objects as go
import pandas as pd

# Sesimpel itu!
```

Dua API:

- `plotly.express` (px) - High-level, cepat
- `plotly.graph_objects` (go) - Low-level, kontrol penuh

Mulai dari px, pakai go untuk kustomisasi

2 Diagram Batang

```
# Siapkan data
metode_count = df['metode_pengadaan'].value_counts().reset_index()
metode_count.columns = ['metode', 'jumlah']

# Buat chart
fig = px.bar(
    metode_count,
    x='metode',
    y='jumlah',
    title='Distribusi Metode Pengadaan',
    labels={'metode': 'Metode', 'jumlah': 'Jumlah Paket'},
    color='jumlah',
    color_continuous_scale='Blues'
)

fig.show()
```

🔄 Diagram Batang Horizontal

```
# Top 10 Satker
top_satker = df.groupby('nama_satker')['pagu'].sum() \
    .sort_values(ascending=False).head(10)

fig = px.bar(
    y=top_satker.index,
    x=top_satker.values / 1e9,
    orientation='h',
    title='Top 10 Satker',
    labels={'x': 'Total Pagu (Miliar)', 'y': 'Satker'}
)

fig.show()
```

3 Diagram Pai

```
# Distribusi Jenis Pengadaan
jenis_count = df['jenis_pengadaan'].value_counts().head(5)

fig = px.pie(
    values=jenis_count.values,
    names=jenis_count.index,
    title='Distribusi Jenis Pengadaan (Top 5)',
    hole=0.3  # Donut chart
)

fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show()
```

4 Diagram Garis

```
# Tren per bulan
df['bulan'] = df['tgl_pengumuman_paket'].dt.to_period('M').astype(str)
monthly = df.groupby('bulan').size().reset_index(name='jumlah')

fig = px.line(
    monthly,
    x='bulan',
    y='jumlah',
    title='Tren Pengumuman Paket per Bulan',
    markers=True
)

fig.show()
```

5 Kustomisasi

```
fig = px.bar(...)

# Update layout
fig.update_layout(
    title='Judul Kustom',
    title_font_size=20,
    height=500,
    width=800,
    showlegend=False,
    template='plotly_white'  # Tema bersih
)

# Update axis
fig.update_xaxes(title='Label X', tickangle=45)
fig.update_yaxes(title='Label Y')

fig.show()
```

🎨 Skala Warna

```
# Skala warna kontinu
color_continuous_scale='Blues'      # Satu hue
color_continuous_scale='Viridis'    # Multi-hue
color_continuous_scale='RdYlGn'     # Merah-Kuning-Hijau

# Warna diskret
color_discrete_sequence=px.colors.qualitative.Plotly
color_discrete_sequence=px.colors.qualitative.Set3
```

Lihat semua: `px.colors.named_colorscales()`

6 Ekspor Grafik

```
# Simpan sebagai HTML (interaktif)
fig.write_html('chart.html')

# Simpan sebagai PNG (butuh kaleido)
fig.write_image('chart.png')

# Simpan sebagai PDF
fig.write_image('chart.pdf')
```

Untuk Streamlit: Gunakan `st.plotly_chart(fig)`

🎯 Contoh Praktis

```
# Multi-group bar chart
grouped = df.groupby(['metode_pengadaan', 'jenis_pengadaan']) \
    .size().reset_index(name='jumlah')
grouped = grouped.nlargest(15, 'jumlah')

fig = px.bar(
    grouped,
    x='metode_pengadaan',
    y='jumlah',
    color='jenis_pengadaan',
    title='Distribusi Jenis per Metode (Top 15)',
    barmode='group',
    height=500
)

fig.show()
```

💡 Praktik Terbaik

✅ **Gunakan judul & label yang jelas**
✅ **Pilih tipe chart yang tepat**
✅ **Batasi kategori** (maks 10-15 agar jelas)
✅ **Gunakan warna dengan makna** (bukan sekadar dekorasi)
✅ **Buat interaktif** (hover info!)
✅ **Export untuk dibagikan** (HTML untuk web, PNG untuk dokumen)

❌ Jangan overload dengan terlalu banyak data
❌ Hindari 3D jika tidak perlu

📊 Panduan Pemilihan Chart

Jenis Data	Tipe Chart	Kasus Penggunaan
Kategorikal	Bar, Pie	Distribusi
Time Series	Line	Tren
Perbandingan	Bar	Peringkat
Bagian-dari-keseluruhan	Pie, Donut	Proporsi
Distribusi	Histogram	Frekuensi
Relationship	Scatter	Korelasi

🎓 Latihan Praktik

Bagian A - DuckDB (30 menit):

1. Setup koneksi DuckDB
2. Query: total pagu per jenis pengadaan
3. Query: Top 10 satker by count
4. Export hasil ke CSV

Bagian B - Visualisasi (30 menit):

1. Bar chart: Metode pengadaan
2. Pie chart: Status PDN/UKM
3. Line chart: Tren bulanan
4. Export charts

📚 Ringkasan Inti

✅ **DuckDB** = SQL analytics cepat di Python
✅ **SQL** = Deklaratif, powerful untuk analisis data
✅ **GROUP BY + HAVING** = Inti analisis agregasi
✅ **Plotly** = Visualisasi interaktif & modern
✅ **px** = Cepat bikin chart, **go** = Kontrol penuh
✅ Integrasi: DuckDB → Pandas → Plotly

🔗 Referensi

DuckDB:

- Docs: https://duckdb.org/docs/
- SQL Reference: https://duckdb.org/docs/sql/introduction

Plotly:

- Docs: https://plotly.com/python/
- Gallery: https://plotly.com/python/plotly-express/
- Cheat Sheet: https://images.plot.ly/plotly-documentation/images/python_cheat_sheet.pdf

🍱 Waktunya Makan Siang!

Selanjutnya (Besok): Sesi 3 - Dasar Streamlit

Hari 1 selesai! 🎉