
Laporan Kecerdasan Web dan Big Data

Tugas 1: Scraping & Visualisasi Data

Disusun Oleh:

Azaria Putri Fawnia - 5024221038

Dosen Pengampu:

Arta Kusuma Hernanda



COMPUTER ENGINEERING

DEPARTEMEN TEKNIK KOMPUTER
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SEMESTER GENAP 2025

Daftar Isi

Daftar Isi	ii
Daftar Gambar	iii
1 Penjelasan Umum	1
1.1 Latar Belakang	1
1.2 Manfaat Tugas	1
2 Scraping Data	2
2.1 Tujuan	2
2.2 Dasar Teori	2
2.3 Alat dan Bahan	3
2.3.1 Library yang Digunakan	3
2.4 Prosedur Scraping Data	4
2.5 Instruksi Tugas	4
2.6 Implementasi	4
2.6.1 Bagian 1: Scraping Data	4
3 Visualisasi Data	9
3.1 Tujuan	9
3.2 Dasar Teori	9
3.2.1 Currency Pair	9
3.2.2 Data Visualization	9
3.2.3 Database MySQL	9
3.2.4 Docker	10
3.2.5 Metabase	10
3.2.6 Time Rounding dan Agregasi Data	10
3.3 Alat dan Bahan	11
3.4 Prosedur	11
3.4.1 Bagian 1: Visualisasi Data	11
3.5 Skenario	11
3.6 Instruksi Tugas	11
3.7 Implementasi	11
3.7.1 Bagian 1: Visualisasi Data	11

3.8	Analisis	13
-----	--------------------	----

Daftar Gambar

3.1	Dashboard Metabase Market Forex	13
-----	---	----

Big Data 1

Penjelasan Umum

1.1 Latar Belakang

Dalam era globalisasi dan kemajuan teknologi informasi, data nilai tukar mata uang menjadi sangat penting dalam berbagai sektor seperti ekonomi, bisnis, dan investasi. Oleh karena itu, diperlukan metode yang efektif untuk memperoleh data nilai tukar secara real-time. Salah satu caranya adalah dengan melakukan *data crawling* dari sumber data terpercaya seperti Yahoo Finance.

1.2 Manfaat Tugas

1. Memahami konsep dasar web crawling dan pengolahan data secara otomatis.
2. Mengembangkan kemampuan dalam memproses dan menyimpan data hasil crawling.
3. Menambah kemampuan dalam visualisasi data hasil crawling.

Big Data 2

Scraping Data

2.1 Tujuan

Laporan ini bertujuan untuk menjelaskan terkait crawling data nilai tukar mata uang dari Yahoo Finance dan menyimpan data tersebut ke dalam file CSV untuk keperluan visualisasi dan analisis lebih lanjut.

2.2 Dasar Teori

- **Web Crawling:** Web crawling adalah proses otomatisasi untuk mengambil data dari halaman-halaman web secara sistematis. Proses ini dilakukan oleh program yang disebut *crawler* atau *spider*, yang akan mengunjungi situs web, membaca isinya, lalu mengambil data yang dibutuhkan. Dalam konteks tugas ini, web crawling digunakan untuk mengakses data nilai tukar mata uang secara langsung dari Yahoo Finance. Web crawling menjadi teknik penting dalam *data mining* dan *big data analysis* karena membantu mengumpulkan data dalam jumlah besar dari berbagai sumber secara cepat dan terstruktur.
- **Yahooquery:** `yahooquery` adalah sebuah pustaka (library) Python yang digunakan untuk mengambil data keuangan secara langsung dari Yahoo Finance API secara resmi maupun tidak langsung. Library ini mendukung berbagai jenis data seperti harga saham, nilai tukar mata uang, laporan keuangan, hingga data historis. Dalam tugas ini, `yahooquery` digunakan untuk mengambil harga pasar terbaru (*regular market price*), perubahan harga (*regular market change*), dan persentase perubahan harga (*regular market change percent*) dari berbagai pasangan mata uang dunia. Dibandingkan scraping manual, `yahooquery` menawarkan akses yang lebih stabil, terstruktur, dan mudah diolah.
- **Dataframe:** DataFrame adalah struktur data tabular dua dimensi yang disediakan oleh pustaka `pandas` di Python. Ia mirip dengan tabel pada basis data atau lembar kerja Excel, yang terdiri dari baris dan kolom. Setiap kolom dalam DataFrame memiliki nama, dan setiap baris memiliki indeks. Dalam proyek ini, DataFrame

digunakan untuk menyimpan hasil crawling data mata uang dengan format yang rapi, di mana setiap baris mewakili data satu pasangan mata uang pada satu waktu tertentu. `DataFrame` memudahkan manipulasi data seperti filtering, sorting, dan agregasi sebelum data tersebut disimpan ke dalam file CSV.

2.3 Alat dan Bahan

- Bahasa Pemrograman: Python
- Library: `yahooquery`, `pandas`, `datetime`
- Sumber Data: Yahoo Finance
- Software Tambahan: Visual Studio Code

2.3.1 Library yang Digunakan

Dalam implementasi program scraping data mata uang ini, beberapa library eksternal digunakan untuk mempermudah proses pengambilan, pengolahan, dan penyimpanan data. Berikut adalah penjelasan masing-masing library:

1. `yahooquery`

`yahooquery` adalah library Python yang digunakan untuk mengambil data keuangan secara langsung dari Yahoo Finance melalui API yang disediakan. Dibandingkan metode scraping biasa, `yahooquery` jauh lebih cepat, andal, dan efisien karena bekerja dengan pendekatan berbasis API.

Pada program ini, `yahooquery` digunakan untuk:

- Membuat objek `Ticker` untuk daftar pasangan mata uang.
- Mengambil informasi harga pasar saat ini (*`regularMarketPrice`*), perubahan harga (*`regularMarketChange`*), dan persentase perubahan harga (*`regularMarketChangePercent`*).

Keunggulan `yahooquery` adalah mampu mengambil banyak instrumen keuangan dalam satu waktu tanpa perlu melakukan scraping manual halaman web.

2. `pandas`

`pandas` adalah library Python yang sangat populer untuk manipulasi dan analisis data dalam bentuk tabel (struktur data `DataFrame`). `Pandas` menyediakan fungsi-fungsi yang mudah digunakan untuk membaca, menulis, dan memproses data dalam berbagai format seperti CSV, Excel, SQL, dan lain-lain.

Dalam program ini, `pandas` berperan untuk:

- Membuat `DataFrame` dari hasil data scraping agar data lebih terstruktur.
- Menyimpan `DataFrame` ke dalam file CSV dengan fungsi `to_csv()`.

- Mengecek apakah file CSV sudah ada untuk menentukan apakah header perlu ditulis atau tidak.

Penggunaan `pandas` membuat proses penyimpanan dan pengolahan data menjadi jauh lebih sederhana dan terorganisir.

3. `datetime`

`datetime` adalah library bawaan Python yang digunakan untuk memanipulasi dan merepresentasikan tanggal serta waktu. Library ini sangat berguna ketika program membutuhkan penandaan waktu (timestamp) dalam data yang dikumpulkan.

Pada program ini, `datetime` digunakan untuk:

- Mengambil waktu saat scraping berlangsung menggunakan fungsi `datetime.now()`.
- Memformat waktu dalam bentuk YYYY-MM-DD HH:MM:SS dengan fungsi `strftime()`.

Penambahan timestamp sangat penting untuk keperluan analisis waktu-seri di masa depan.

2.4 Prosedur Scraping Data

Proses scraping dilakukan dengan:

1. Menentukan daftar pasangan mata uang yang akan diambil.
2. Menggunakan library `yahooquery` untuk mengakses data harga terbaru.
3. Menyimpan hasil scraping ke dalam struktur `DataFrame`.
4. Menyimpan `DataFrame` ke dalam file CSV untuk keperluan analisis selanjutnya.

2.5 Instruksi Tugas

- Melakukan crawling data nilai tukar dari Yahoo Finance.
- Menyimpan hasil crawling ke dalam file CSV.

2.6 Implementasi

2.6.1 Bagian 1: Scraping Data

Implementasi program dilakukan dengan langkah-langkah berikut:

1. Menentukan Daftar Pasangan Mata Uang

Code

```
1 from yahooquery import Ticker
2 import pandas as pd
3 from datetime import datetime
4
5 # Daftar pasangan mata uang
6 currency_pairs = [
7     "EURUSD=X", "JPY=X", "GBPUSD=X", "AUDUSD=X", "NZDUSD=X",
8     "EURJPY=X", "GBPJPY=X", "EURGBP=X", "EURCAD=X", "EURSEK=X",
9     "EURCHF=X", "EURHUF=X", "CNY=X", "HKD=X", "SGD=X", "INR=X",
10    "MXN=X", "PHP=X", "IDR=X", "THB=X", "MYR=X", "ZAR=X", "RUB=X"
11 ]
```

Langkah pertama adalah menentukan pasangan mata uang (currency pairs) yang ingin diambil datanya. Pasangan mata uang seperti EUR/USD, GBP/JPY, hingga IDR/USD ditentukan di awal dan dimasukkan ke dalam sebuah daftar (*list*) di dalam program. Penentuan pasangan ini penting untuk memperjelas ruang lingkup scraping agar hanya data yang diperlukan yang diambil dari sumber.

2. Menginisialisasi Objek Ticker dari Library yahooquery

Code

```
1 tickers = Ticker(currency_pairs)
2 prices = tickers.price
```

Library *yahooquery* digunakan untuk berinteraksi dengan Yahoo Finance secara langsung. Dengan menggunakan objek *Ticker* dari *yahooquery*, seluruh pasangan mata uang yang telah ditentukan dapat diakses informasinya secara bersamaan. Hal ini memudahkan pengambilan data karena hanya membutuhkan satu kali inisialisasi untuk banyak instrumen sekaligus.

3. Mengambil Data Harga Pasar

Code

```
1 # Proses data
2 currency_data = []
3 timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S") # Waktu
4     ↪ scraping
```

```

5     for pair in currency_pairs:
6         if pair in prices:
7             price = prices[pair].get("regularMarketPrice", "N/A")
8             change = prices[pair].get("regularMarketChange", "N/A")
9             change_percent =
    ↪ prices[pair].get("regularMarketChangePercent", "N/A")
10            currency_data.append([timestamp, pair, price, change,
    ↪ change_percent])

```

Setelah objek `Ticker` dibuat, data harga pasar diperoleh dengan memanggil atribut `price`. Data yang diambil untuk masing-masing pasangan mata uang meliputi:

- Harga pasar saat ini (*regularMarketPrice*)
- Nilai perubahan harga (*regularMarketChange*)
- Persentase perubahan harga (*regularMarketChangePercent*)

Data ini dikumpulkan dalam format dictionary, dengan masing-masing pasangan mata uang sebagai kunci.

4. Menyusun Data dalam Struktur DataFrame

Code

```

1     # Konversi ke DataFrame
2     df = pd.DataFrame(currency_data, columns=["timestamp",
    ↪ "currency_pair", "price", "change_value", "change_percent"])

```

Setiap data hasil scraping diproses menjadi sebuah list yang berisi timestamp pengambilan data, pasangan mata uang, harga, perubahan harga, dan persentase perubahan harga. Semua list kemudian dikumpulkan ke dalam sebuah `pandas DataFrame` agar data lebih terstruktur. `DataFrame` sangat membantu dalam menyusun data tabular sehingga dapat langsung diolah lebih lanjut atau disimpan ke dalam file.

5. Menyimpan Data ke dalam File CSV

Code

```

1     # Simpan ke CSV (append tanpa header jika sudah ada file)
2
    ↪ df.to_csv("C:/zafaa/kuliah/SEMESTER6/BIGDATA/test2/currency_rates.csv",
    ↪ mode='a',

```

```

3         header=not
    ↪ pd.io.common.file_exists("C:/zafaa/kuliah/SEMESTER6/BIGDATA/test2/
4         currency_rates.csv"), index=False)
5
6     print("Scraping selesai! Data berhasil ditambahkan ke
    ↪ 'currency_rates.csv'.")

```

DataFrame yang telah berisi hasil scraping kemudian disimpan ke dalam file CSV (Comma Separated Values) menggunakan fungsi `to_csv()` dari pandas. File CSV disimpan di direktori lokal yang telah ditentukan. Untuk menghindari penulisan ulang file setiap kali scraping dilakukan, program dirancang untuk menambahkan data (*append*) ke file CSV yang sudah ada, serta hanya menulis header jika file tersebut belum tersedia.

6. Menambahkan Timestamp

Setiap record data scraping dilengkapi dengan kolom `timestamp` yang berisi waktu pengambilan data dalam format `YYYY-MM-DD HH:MM:SS`. Timestamp ini sangat penting untuk menandai kapan data tersebut diambil, sehingga memungkinkan analisis tren atau perubahan harga seiring waktu di masa depan.

Berikut adalah listing program lengkap untuk scraping data:

Code

```

1 from yahooquery import Ticker
2 import pandas as pd
3 from datetime import datetime
4
5 # Daftar pasangan mata uang
6 currency_pairs = [
7     "EURUSD=X", "JPY=X", "GBPUSD=X", "AUDUSD=X", "NZDUSD=X",
8     "EURJPY=X", "GBPJPY=X", "EURGBP=X", "EURCAD=X", "EURSEK=X",
9     "EURCHF=X", "EURHUF=X", "CNY=X", "HKD=X", "SGD=X", "INR=X",
10    "MXN=X", "PHP=X", "IDR=X", "THB=X", "MYR=X", "ZAR=X", "RUB=X"
11 ]
12
13 # Ambil data dari Yahoo Finance
14 tickers = Ticker(currency_pairs)
15 prices = tickers.price
16
17 # Proses data
18 currency_data = []
19 timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S") # Waktu scraping
20
21 for pair in currency_pairs:

```

```

22     if pair in prices:
23         price = prices[pair].get("regularMarketPrice", "N/A")
24         change = prices[pair].get("regularMarketChange", "N/A")
25         change_percent = prices[pair].get("regularMarketChangePercent", "N/A")
26         currency_data.append([timestamp, pair, price, change, change_percent])
27
28 # Konversi ke DataFrame
29 df = pd.DataFrame(currency_data, columns=["timestamp", "currency_pair", "price",
    ↪ "change_value", "change_percent"])
30
31 # Simpan ke CSV (append tanpa header jika sudah ada file)
32 df.to_csv("C:/zafaa/kuliah/SEMESTER6/BIGDATA/test2/currency_rates.csv", mode='a',
33         header=not
    ↪ pd.io.common.file_exists("C:/zafaa/kuliah/SEMESTER6/BIGDATA/test2/
34         currency_rates.csv"), index=False)
35
36 print("Scraping selesai! Data berhasil ditambahkan ke 'currency_rates.csv'.")

```

Big Data 3

Visualisasi Data

3.1 Tujuan

Laporan ini bertujuan untuk menjelaskan terkait pengumpulan, penyimpanan, dan visualisasi data harga pasangan mata uang (currency pair) secara periodik, serta menganalisis stabilitas dan volatilitas masing-masing pasangan berdasarkan data tersebut. Visualisasi dilakukan menggunakan Metabase terintegrasi dengan database MySQL.

3.2 Dasar Teori

3.2.1 Currency Pair

Currency pair adalah harga relatif dari satu mata uang terhadap mata uang lainnya. Dalam dunia trading forex, sebuah currency pair, seperti EUR/USD, menunjukkan berapa banyak dolar AS (USD) yang diperlukan untuk membeli satu euro (EUR). Setiap pasangan memiliki nilai tukar yang terus bergerak berdasarkan dinamika pasar seperti suku bunga, tingkat inflasi, pertumbuhan ekonomi, dan kondisi geopolitik.

3.2.2 Data Visualization

Visualisasi data adalah teknik untuk merepresentasikan data secara grafis sehingga pola, tren, dan insight dapat dengan mudah dipahami oleh pengguna. Menurut *Tufte* (1983), visualisasi yang efektif memaksimalkan rasio data-ink, yakni memaksimalkan jumlah data yang ditampilkan relatif terhadap jumlah elemen grafis non-data. Dalam konteks proyek ini, visualisasi data membantu pengguna untuk memahami pergerakan harga currency pair secara lebih intuitif dan efisien.

3.2.3 Database MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) open-source yang berbasis pada Structured Query Language (SQL). MySQL digunakan untuk menyimpan data currency pair dalam format tabel yang terstruktur, sehingga data dapat dikelola, diquery,

dan dianalisis dengan efisien. Beberapa fitur penting MySQL yang digunakan dalam proyek ini antara lain:

- **Auto Increment** untuk kolom `id`.
- **Primary Key** untuk menjaga keunikan data.
- **Datetime field** untuk penyimpanan waktu data dengan presisi.

3.2.4 Docker

Docker adalah platform berbasis container yang memungkinkan developer untuk mengemas aplikasi beserta semua dependensinya ke dalam sebuah container yang portabel dan ringan. Penggunaan Docker dalam proyek ini bertujuan untuk menjalankan Metabase tanpa harus melakukan instalasi manual di sistem operasi, meningkatkan portabilitas dan mengurangi kompleksitas setup.

3.2.5 Metabase

Metabase adalah aplikasi open-source untuk business intelligence (BI) yang memudahkan analisis data dan pembuatan dashboard tanpa perlu kemampuan teknis mendalam dalam SQL. Metabase dapat terhubung ke berbagai jenis database termasuk MySQL, dan menyediakan fitur seperti:

- Query Builder visual
- Filter interaktif
- Grafik dan chart otomatis
- Dashboard yang bisa dibagikan

Dalam proyek ini, Metabase digunakan untuk membuat visualisasi harga currency pair, menganalisis volatilitas pasar, dan menampilkan pergerakan harga secara real-time atau historis.

3.2.6 Time Rounding dan Agregasi Data

Dalam analisis data berkala, penting untuk melakukan normalisasi waktu agar data dapat digabungkan dan dibandingkan dengan adil. Teknik *rounding time* digunakan untuk membulatkan waktu ke interval tertentu, seperti kelipatan 5 menit, sehingga memudahkan pengelompokan data. Selanjutnya, fungsi `AVG()` dalam SQL digunakan untuk menghitung harga rata-rata pada setiap interval waktu tersebut, mengurangi noise akibat fluktuasi harga mikro dan memperjelas tren utama.

3.3 Alat dan Bahan

- XAMPP (MySQL dan phpMyAdmin)
- Docker Desktop
- File CSV hasil scraping data currency pair
- Aplikasi Metabase (open-source, dijalankan lewat Docker)

3.4 Prosedur

3.4.1 Bagian 1: Visualisasi Data

1. Setup MySQL server menggunakan XAMPP.
2. Membuat database `currency_db` dan tabel `currency_rates`.
3. Import data CSV ke dalam tabel, dengan memastikan tidak ada duplikasi berdasarkan kolom `timestamp`.
4. Deploy Metabase menggunakan Docker:
 - Pull image Metabase: `docker pull metabase/metabase`
 - Jalankan container: `docker run -d -p 3000:3000 --name metabase metabase/metabase`
5. Hubungkan Metabase ke database MySQL `currency_db`.
6. Buat dashboard dan visualisasi dengan berbagai macam filter dan chart.

3.5 Skenario

Data scraping dilakukan secara otomatis setiap 5 menit, menghasilkan data harga, perubahan nilai (change value), dan perubahan persentase (change percent) untuk berbagai currency pair. Data yang dikumpulkan digunakan untuk menganalisis stabilitas, volatilitas, dan tren harga.

3.6 Instruksi Tugas

- Melakukan visualisasi data dengan dashboard Metabase dari data yang sudah di crawling sebelumnya.

3.7 Implementasi

3.7.1 Bagian 1: Visualisasi Data

1. Setup Database dan Import Data

1. Membuat database `currency_db`.

2. Membuat tabel `currency_rates` dengan kolom:

- `id` (Primary Key, Auto Increment)
- `timestamp` (Datetime)
- `currency_pair` (Varchar)
- `price` (Double)
- `change_value` (Double)
- `change_percent` (Double)
- `rounded_time` (Datetime, hasil pembulatan timestamp ke 5 menit terdekat)

3. Import file CSV menggunakan phpMyAdmin dengan fitur *Import* dan memastikan tidak ada duplikasi menggunakan query SQL untuk membandingkan `timestamp`.

2. Setup Metabase via Docker

- Jalankan perintah: `docker run -d -p 3000:3000 --name metabase metabase/metabase`
- Akses Metabase di browser melalui `localhost:3000`.
- Integrasikan Metabase dengan database MySQL `currency_db`.

3. Pembuatan Dashboard dan Visualisasi

- Membuat **New Dashboard** di Metabase.
- Membuat berbagai **New Question**:

1. Grafik Harga per Pair:

- Visualisasi: Line Chart.
- Summarize: Average of price.
- Group By: `rounded_time` (per 5 menit atau per jam).
- Filter: `currency_pair` dan tanggal.

2. Pair Paling Volatil:

- Summarize: Standard deviation of price.
- Group By: `currency_pair`.
- Sort: dari nilai standar deviasi tertinggi.

3. Pair Paling Stabil:

- Summarize: Standard deviation of price.
- Group By: `currency_pair`.
- Sort: dari nilai standar deviasi terendah.

4. Tren Perubahan Nilai:

- Visualisasi `change_value` dan `change_percent` dalam Line Chart.

5. Insight Summary:

- Menggunakan fungsi aggregation: AVG(price), MAX(price) per hari.
- Menambahkan **Global Dashboard Filter** untuk memilih currency_pair dan rounded_time.

3.8 Analisis

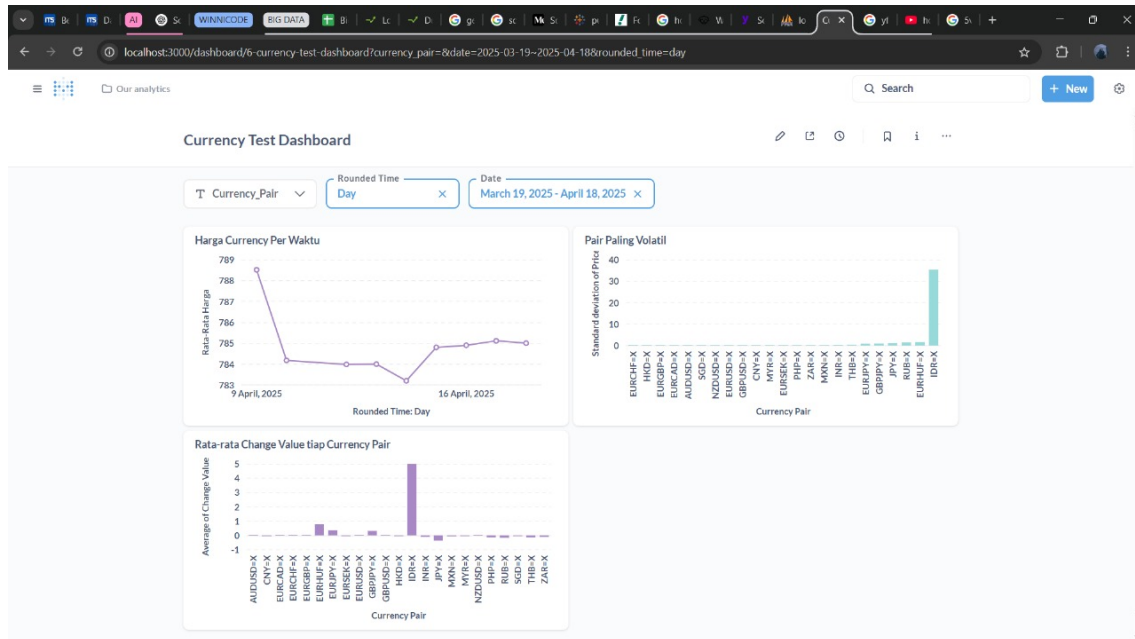


Figure 3.1: Dashboard Metabase Market Forex

Berdasarkan visualisasi yang dihasilkan:

- Pair dengan volatilitas tertinggi menunjukkan potensi peluang trading yang lebih besar namun juga risiko lebih tinggi.
- Pair yang stabil cocok untuk konservasi nilai dan analisis tren jangka panjang.
- Grafik harga per waktu membantu mengidentifikasi pola pergerakan harga.
- Filter tanggal dan pair memungkinkan analisis spesifik per hari atau per event tertentu.