PENGAMBILAN DATA ASET WEBSITE PROPERTI DAN PEMBUATAN DATABASE PENYIMPANAN DATA WEBSITE INI VIE HOSPITALITY

I.G.N.B.Ferry Mahayudha¹, IB Gede Dwidasmara², dan Luh Gede Astuti³

ABSTRAK

Divisi Web Development di perusahaan Ini Vie Hospitality memiliki peran kunci dalam pengembangan dan pengelolaan website properti. Divisi ini memiliki tanggung jawab utama dalam mengambil data yang berkaitan dengan aset properti dan merancang database yang efisien untuk penyimpanan serta manajemen data tersebut. Tujuan utama dari Divisi Web Development ini adalah untuk menyediakan platform yang efektif dan user-friendly bagi pengguna untuk menjelajahi, mencari, dan mendapatkan informasi terkait properti yang dikelola oleh perusahaan Ini Vie Hospitality. Tim ini juga bertanggung jawab untuk mengumpulkan data properti, termasuk gambar, deskripsi, detail teknis, dan informasi lainnya yang relevan. Mereka dapat mengintegrasikan data dari seluruh website properti yang dikelola. Pengambilan data dilakukan bantuan script program yang dapat melakukan web scraping.

Kata kunci: Ini Vie, Database, Script, Web Scraping.

ABSTRACT

The Web Development Division at Ini Vie Hospitality plays a key role in the development and management of property websites. This division has the primary responsibility of gathering data related to property assets and designing an efficient database for the storage and management of this data. The main objective of this Web Development Division is to provide an effective and user-friendly platform for users to explore, search, and obtain information about properties managed by Ini Vie Hospitality. This team is also responsible for collecting property data, including images, descriptions, technical details, and other relevant information. They can integrate data from all managed property websites. Data retrieval is facilitated through a program script capable of web scraping.

Keywords: Ini Vie, Database, Script, Web Scraping

Submitted: 10 April 2025 Revised: 29 April 2025 Accepted: 30 April 2025

¹ Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, Bukit Jimbaran, 80361, Denpasar-Indonesia, tedi.dharma84@gmaill.com

² Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, Bukit Jimbaran, 80361, Denpasar-Indonesia, dwidasmara@unud.ac.id

³ Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana, Bukit Jimbaran, 80361, Denpasar-Indonesia, hendra.suputra@unud.ac.id

1. PENDAHULUAN

Perusahaan Ini Vie Hospitality merupakan Perusahaan yang bergerak di bidang pelayanan hotel dan villa. Perusahaan ini mengurus beberapa properti dimulai dari pelayanan atau hospitality, pembuatan dan pengembangan website resmi setiap properti, serta pembuatan sosial media setiap properti. Dalam Perusahaan ini, divisi yang membuat dan mengurus website adalah divisi Website Development.

Data scraping, juga dikenal sebagai web scraping, adalah proses ekstraksi data dari situs web atau sumber lainnya. Web scraping merupakan salah satu bentuk dari web automation. Menurut (Saaya, Z. 2021), Otomatisasi web digunakan untuk pengujian perangkat lunak sebagai alat otomatisasi untuk mengurangi intervensi manusia dan tugas yang berulang. Ini melibatkan penggunaan perangkat lunak atau teknik pemrograman untuk mengambil informasi tertentu dari halaman web dan menyimpannya dalam bentuk yang dapat digunakan, seperti file .csv, spreadsheet, atau database. Metode ini dapat dilakukan menggunakan selenium. Selenium bekerja dengan menggunakan perangkat lunak tiruan berupa chrome untuk mencari website dan mencari data secara otomatis sesuai dengan kode yang ditulis. Dengan selenium, pengguna dapat menguji cara kerja situs web dengan mengeklik tautan, mengisi formulir, mengonfirmasi konten, dan banyak lagi (Rusdiansyah, R., dkk., 2024). Beautifulsoup4 juga digunakan untuk web scraping berdampingan dengan penggunaan selenium. Tugas utama Beautiful Soup adalah mengambil konten HTML dan mengubahnya menjadi berbasis pohon perwakilan (Simona V. O., Adela B. 2022).

Dalam praktik data scraping, menurut penelitian (Andreea-Maria T., dkk. 2023), pengambilan data dilakukan dengan berinteraksi dengan struktur HTML dan untuk website statis, library beautifulsoup4 sudah cukup untuk melakukan web scraping. Menurut (Thota, P., & Ramez, E. 2021), selenium dapat digunakan untuk menjalankan kode javascript sehingga dapat mengakses website yang dibuat menggunakan javascript yang memiliki struktur yang lebih dinamis. Selenium terdiri dari driver web untuk rendering halaman web, yang dapat digunakan untuk melakukan klik, sorotan, gulir, dll secara otomatis. Untuk membantu mengambil data yang belum tersimpan ke dalam database, diperlukan script, yaitu berupa program dalam jupyter notebook (.ipynb) yang dapat mengambil data dalam setiap entitas yang memuat tentang data aset pada struktur HTML, memproses data mentah yang sudah diambil dari website properti, serta dapat menginput data ke database. Dalam membuat database untuk penyimpanan data berupa aset website setiap properti, dibutuhkan website yang dapat mendesain dan mengekspor hasil desain dalam bentuk .sql, yaitu dbdiagrams.net.

2. METODE PELAKSANAAN

Webdriver dipasang terlebih dahulu untuk memudahkan pengguna mengambil data dari sebuah entity pada HTML suatu website agar dapat mengambil data yang diinginkan dari attribut dalam suatu entitas. Pengambilan data dilakukan dengan cara yang berurutan seperti berikut:

2.1 Mencari website yang akan dituju

Untuk website yang dituju untuk pencarian data, ada total 43 website yang sudah dituju. Dimana, seluruh website merupakan website di bawah pengelolaan Perusahaan Ini Vie Hospitality.

2.2 Mencari elemen berdasarkan nama tag, class, dan berdasarkan struktur HTML secara lebih lengkap (xpath)

Untuk mencari seluruh elemen berdasarkan nama tag, class, dan berdasarkan struktur HTML lebih lengkap (xpath) secara spesifik, digunakan method driver.find_elements() pada variabel 'driver'. Parameter yang digunakan hanyalah 2, yaitu nama dari elemen dan value dari elemen yang akan dicari pada struktur HTML.

```
# berdasarkan nama tag
all data = [data for data in driver.find elements(By.TAG NAME, 'tag-name')]
# berdasarkan nama class
all data = [data for data in driver.find elements(By.CLASS NAME, 'class-name')]
# berdasarkan xpath
all data = [data for data in driver.find elements(By.XPATH, '//entity[@attribute="attribute-name"]')]
```

Tabel 2.1 Kode untuk mencari elemen berdasarkan HTML

2.3 Mencari nilai atribut

Untuk mencari nilai atribut setelah mendapatkan seluruh elemen yang dicari, digunakan method data.get_attributes() pada variabel 'data' dalam seluruh data yang ditemukan oleh driver. Parameter yang digunakan hanyalah satu, yaitu nama dari atribut yang dicari.

```
# berdasarkan nama tag
all data = [data.get attribute('attribute-name') for data in driver.find elements(By.TAG NAME, 'tag-
name')]
# berdasarkan nama class
all data = [data.get_attribute('attribute-name') for data in driver.find_elements(By.CLASS_NAME,
'class-name')]
# berdasarkan xpath
all data = [data.get attribute('attribute-name') for data in driver.find elements(By.XPATH,
'//entity[@attribute="attribute-name"]')]
```

Tabel 2.2 Kode untuk mencari atribut berdasarkan HTML

2.4 Memproses data

Pada tahap pemrosesan data, seluruh data yang telah terkumpul akan dijadikan satu dictionary yang bertujuan untuk mempermudah pengolahan data menjadi file .csv. Penggunaan for loop digunakan agar mempermudah pembacaan kode dalam pemisahan atau pembersihan data.

```
\overline{\text{dict}} = \{\}
dict['key1'] = []
dict['key2'] = []
# berdasarkan nama tag
for data in driver.find elements(By.TAG NAME, 'tag-name'):
   data = data.get attribute('attribute-name')
   if value1 in data:
      dict['key1'].append(data.remove(value1, ''))
   if value2 in data:
      dict['key2'].append(data.remove(value2, ''))
# berdasarkan nama class
for data in driver.find elements(By.CLASS NAME, 'tag-name'):
   data = data.get attribute('attribute-name')
   if value1 in data:
      dict['key1'].append(data.remove(value1, ''))
   if value2 in data:
      dict['key2'].append(data.remove(value2, ''))
# berdasarkan xpath
for data in driver.find elements(By.XPATH, '//entity[@attribute="attribute-name"]'):
   data = data.get attribute('attribute-name')
   if value1 in data:
      dict['key1'].append(data.remove(value1, ''))
   if value2 in data:
      dict['key2'].append(data.remove(value2, ''))
```

Tabel 2.3 Kode untuk memproses data

2.5 Membuat file .csv

Dalam tahap ini, *file .csv* dibuat menggunakan bantuan *library* pandas. Dengan method pandas.DataFrame().to_*csv*(), *file* tersebut dapat dibuat.

```
pandas.DataFrame(dict).to_csv(FILE_PATH + '/dict.csv')
```

Tabel 2.4 Kode untuk membuat file .csv

2.6 Membuat rancangan database

Dalam tahap ini, database dirancang pada *website* dbdiagrams.io dengan suatu bahasa yang sudah disediakan sehingga pembuatan desain dapat dilakukan dengan lebih mudah. Tipe data dan status dalam bahasa tersebut disesuaikan dengan SQL sehingga tidak perlu mempelajari bahasa tersebut lebih jauh. Relasi antar tabel disimbolkan dengan < (untuk relasi one to many), > (untuk relasi many to one), <> (untuk relasi many to many), dan – (untuk relasi one to one).

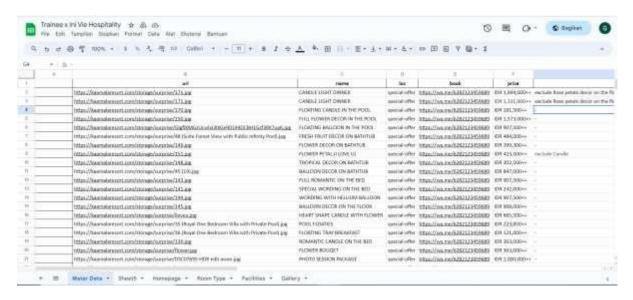
I.G.N.B.Ferry Mahayudha, IB Gede Dwidasmara, dan Luh Gede Astuti

```
# membuat tabel dengan atribut, tipe data, dan status yang diinginkan
Table nama tabel {
 nama_atribut1 tipe_data [status],
 nama atribut2 tipe data [status],
# membuat relasi antar tabel
Ref: "nama tabel1". "nama atribut" < " nama tabel1". "nama atribut "
```

Tabel 2.5 Sintaks untuk membuat database dan relasinya

3. HASIL DAN PEMBAHASAN

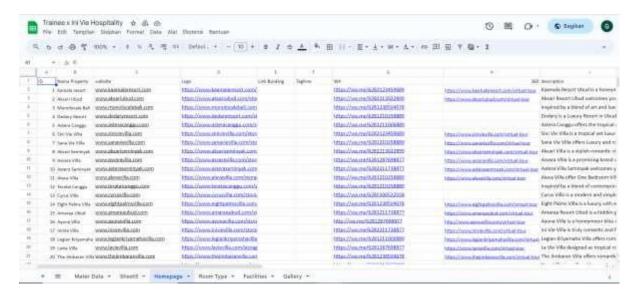
Notebook web scraping yang dibuat dapat membantu dalam pengambilan data secara semi-otomatis dibandingkan dengan metode manual dengan library selenium dan beautifulsoup. Notebook ini membaca struktur HTML dan mengambil data yang ada di dalam entitas berdasarkan nama atribut dari entitas tersbut. Dengan adanya notebook ini, pengambilan data dapat dilakukan denagan jauh lebih cepat.



Gambar 3.1 Data pada sheet 'Mater Data'

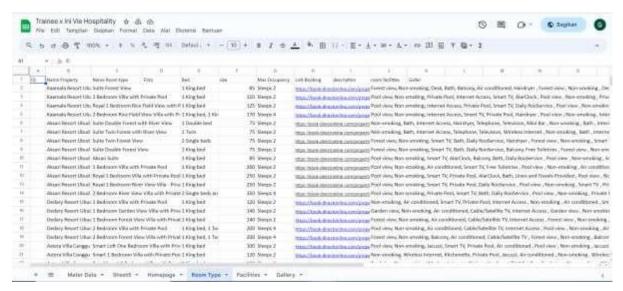
Pada gambar 3.1, terdapat data berupa fasilitas dari tiap resort yang dikelola perusahaan. Data disimpan pada google sheet. Pencarian data dilakukan dengan mencari elemen berdasarkan nama tag, class, dan berdasarkan struktur HTML secara lebih lengkap (xpath) serta mencari nilai atribut menggunakan library beautifulsoup4 dan selenium. Pemrosesan data dan pembuatan .csv dilakukan menggunakan pandas. Setelah file .csv dibuat, file diupload ke google drive dan dijadikan google sheet.

Pengambilan Data Aset Website Properti dan Pembuatan Database Penyimpanan Data Website



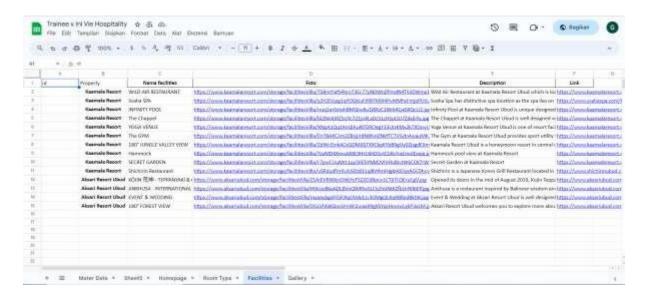
Gambar 3.2 Data pada sheet 'Homepage'

Pada gambar 3.2, terdapat kolom 'url' yang berisi data url seluruh situs yang dikelola perusahaan yang diakses untuk mencari dan menyimpan data pada google sheet. Pencarian data dilakukan dengan mencari elemen berdasarkan struktur HTML serta mencari nilai atribut menggunakan library beautifulsoup4 dan selenium. Pemrosesan data dan pembuatan .csv dilakukan menggunakan pandas.



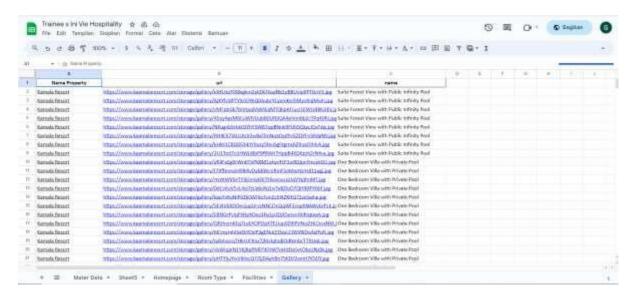
Gambar 3.3 Data pada sheet 'Room Type'

Pada gambar 3.3, terdapat data seluruh ruangan secara detail pada setiap *resort* yang dikelola perusahaan dan disimpan pada *google sheet*. Pencarian data dilakukan dengan mencari elemen berdasarkan struktur *HTML* serta mencari nilai atribut menggunakan *library beautifulsoup4* dan *selenium*. Pemrosesan data dan pembuatan *.csv* dilakukan menggunakan pandas.



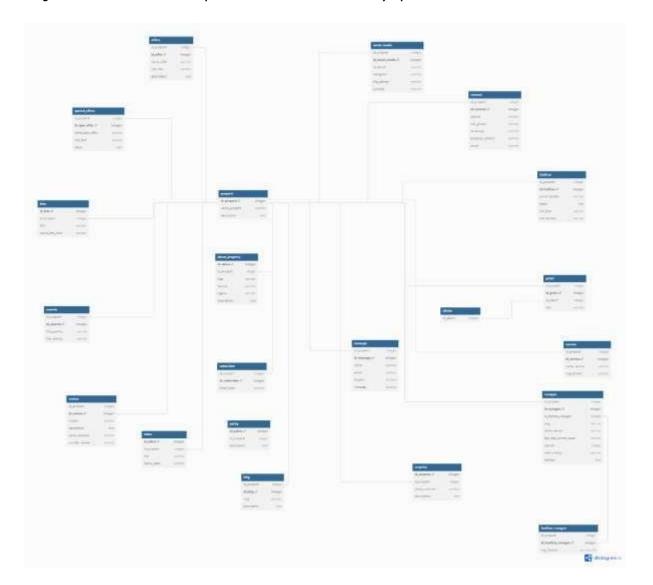
Gambar 3.4 Data pada sheet 'Facilities'

Pada gambar 3.4, terdapat data seluruh fasilitas secara detail pada setiap resort yang dikelola perusahaan dan disimpan pada google sheet. Pencarian data dilakukan dengan mencari elemen berdasarkan struktur HTML serta mencari nilai atribut menggunakan library beautifulsoup4 dan selenium. Pemrosesan data dan pembuatan .csv dilakukan menggunakan pandas



Gambar 3.5 Data pada sheet 'Gallery'

Pada gambar 3.5, terdapat data galeri pada setiap resort yang dikelola perusahaan dan disimpan pada google sheet. Pencarian data dilakukan dengan mencari elemen berdasarkan struktur HTML serta mencari nilai atribut menggunakan library beautifulsoup4 dan selenium. Pemrosesan data dan pembuatan .csv dilakukan menggunakan pandas.



Gambar 3.6 Desain database

Pada gambar 3.6, database yang diperlukan didesain menggunakan script yang terdapat pada website dbdiagrams.net. Setelah rancangan atau desain database telah jadi, script dapat diimport menjadi file .sql sehingga mempermudah pembuatan website.

4. KESIMPULAN

Seluruh data yang diperlukan dapat dicari pada struktur HTML website yang ingin dicari menggunakan library beautifulsoup4 dan selenium dan hasil pencarian data disimpan pada file .csv menggunakan library pandas dan sql-connector. Database yang diperlukan dapat dibuat menggunakan bahasa yang disediakan pada website dbdiagrams.io dan hasil rancangan dapat diekspor dalam bentul .sql sehingga database yang diisi dengan data dapat diimplementasikan ke dalam website perusahaan.

5. UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Ini Vie Hospitality atas kesempatan yang diberikan bagi penulis untuk menjalani PKL. Bahkan, penulis juga dibimbing dan diberi kesempatan untuk bekerja sesuai dengan kemampuan dan kurikulum bagi mahasiswa magang yang sudah disediakan oleh instansi.

DAFTAR PUSTAKA

Saaya, Z. (2021). Extracting Academic Publication Data Using Web Automation Tools. Journal of Advanced Computing Technology and Application (JACTA), vol 3:1, pp. 17-24.

Rusdiansyah, R., Suharyanti, N., Supendar, H., & Tuslaela, T. (2024). Web Program Testing Using Selenium Python: Best Practices and Effective Approaches . Sinkron: Jurnal Dan Penelitian Teknik Informatika, vol. 8:2, pp. 994-1000.

Simona V. O., Adela B. (2022). Why Is More Efficient to Combine Beautiful Soup and Selenium in Scraping For Data Under Energy Crisis. "Ovidius" University Annals, Economic Sciences Series, vol. 22:2, pp. 146-152.

Andreea-Maria T., Simona-Vasilica O., Adela B. (2023). Web Scraping and Review Analytics. Extracting Insights from Commercial Data. "Ovidius" University Annals, Economic Sciences Series, vol. 23:2, pp. 370-

Thota, P., & Ramez, E. (2021). Web Scraping of COVID-19 News Stories to Create Datasets for Sentiment and Emotion Analysis. PETRA '21: The 14th PErvasive Technologies Related to Assistive Environments *Conference*, pp. 306-314.

