



KERJA PRAKTIK - EF234603

Pengolahan Data Global pada Sektor Perdagangan, Pariwisata, Investasi, dan Jasa melalui Proses ETL (Extract, Transform, Load) serta Pengembangan Dashboard Sistem Informasi Diplomasi Ekonomi pada Sektor Pariwisata Indonesia

Kementerian Luar Negeri - Badan Strategi Kebijakan Luar Negeri

Jl. Taman Pejambon No. 6 Jakarta Pusat, 10110

Oleh:

Laurivasya Gadhing Syahafidh 5025211136

**Pembimbing Departemen
Dr. Ahmad Saikhu, S.Si., M.T.**

**Pembimbing Lapangan
Hanavi, S.Kom., M.T.**

**DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024**



KERJA PRAKTIK - EF234603

Pengolahan Data Global pada Sektor Perdagangan, Pariwisata, Investasi, dan Jasa melalui Proses ETL (Extract, Transform, Load) serta Pengembangan Dashboard Sistem Informasi Diplomasi Ekonomi pada Sektor Pariwisata Indonesia

Kementerian Luar Negeri - Badan Strategi Kebijakan Luar Negeri
Jl. Taman Pejambon No. 6 Jakarta Pusat, 10110

Oleh:

Laurivasya Gadhing Syahafidh

5025211136

Pembimbing Departemen

Dr. Ahmad Saikhu, S.Si., M.T.

Pembimbing Lapangan

Hanavi, S.Kom., M.T.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2024

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	iv
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
DAFTAR KODE SUMBER	xiii
LEMBAR PENGESAHAN	xv
KATA PENGANTAR	xx
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Manfaat	2
1.4. Rumusan Masalah	2
1.5. Lokasi dan Waktu Kerja Praktik	3
1.6. Metodologi Kerja Praktik	3
1.6.1. Perumusan Masalah	3
1.6.2. Studi Literatur	4
1.6.3. Implementasi Sistem	5
1.6.4. Pengujian dan Evaluasi	5
1.6.5. Kesimpulan dan Saran	5
1.7. Sistematika Laporan	5
1.7.1. Bab I Pendahuluan	5
1.7.2. Bab II Profil Perusahaan	6
1.7.3. Bab III Tinjauan Pustaka	6

1.7.4.	Bab IV Implementasi Sistem	6
1.7.5.	Bab V Pengujian dan Evaluasi	6
1.7.6.	Bab VI Kesimpulan dan Saran	6
BAB II	PROFIL PERUSAHAAN	8
2.1.	Profil Kementerian Luar Negeri	8
2.1.1	Profil Badan Strategi Kebijakan Luar Negeri	9
2.2.	Lokasi	10
2.3.	Logo Perusahaan	10
2.4.	Visi Misi Perusahaan	10
2.4.1	Visi	10
2.4.2	Misi	10
BAB III	TINJAUAN PUSTAKA	13
3.1.	<i>Extract, Transform, Load (ETL)</i>	13
3.2.	Python	14
3.3.	<i>Web Scraping</i>	15
3.3.1	Selenium	16
3.3.2	Beautiful Soup	16
3.4.	<i>API Data Extraction</i>	16
3.5.	<i>Preprocessing Data</i>	17
3.5.1	Pandas	19
3.5.2	Data Frame	20
3.5.3	NumPy	22
3.6.	<i>Processing Data</i>	23
3.6.1	Scikit-learn	24

3.6.2	Matplotlib	24
3.7.	Manajemen Basis Data	25
3.7.1	Physical Data Model	26
3.7.2	<i>Relational Data Base Management System (RDBMS)</i>	27
3.7.3	MySQL	27
3.7.3.1	<i>Data Definition Language (DDL)</i>	28
3.7.3.2	<i>Data Manipulation Language (DML)</i>	28
BAB IV	IMPLEMENTASI SISTEM	31
4.1	Implementasi Ekstraksi Data (<i>Extract</i>)	31
4.1.1	Web Scraping	31
4.1.2	API Data Extraction	33
4.2	Implementasi Transformasi Data (<i>Transform</i>)	36
4.2.1	Preprocessing Data	36
4.2.2	Processing Data	39
4.3	Implementasi Pemuatan Data (<i>Load</i>)	45
4.3.1	Physical Data Model	45
4.3.2	Data Definition Language (DDL)	45
4.3.3	Data Manipulation Language (DML)	50
4.4	Implementasi Visualisasi Data	53
BAB V	PENGUJIAN DAN EVALUASI	56
5.1.	Tujuan Pengujian	56
5.2.	Kriteria Pengujian	56

5.3. Skenario Pengujian	57
5.4. Evaluasi Pengujian	58
BAB VI KESIMPULAN DAN SARAN	61
6.1. Kesimpulan	61
6.2. Saran	61
DAFTAR PUSTAKA	64
LAMPIRAN	67
BIODATA PENULIS	76

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 5.4. Hasil Evaluasi Pengujian

58

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1.1 Fungsi Bskln Kementerian Luar Negeri	9
Gambar 2.3 Logo Kementerian Luar Negeri	10
Gambar 3.5.2.2 Implementasi .Shape Dan Info()	21
Gambar 3.5.2.3 Implementasi Head()Dan Tail()	22
Gambar 3.6.2 Pentunjuk Visualisasi Data Matplotlib	25

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1.1 Web Scraping Trademap.Org	33
Kode Sumber 4.1.2 Konfigurasi Api Data Extraction	36
Kode Sumber 4.2.1 Preprocessing Data Jumlah Wisatawan Mancanegara Indonesia	39
Kode Sumber 4.2.2.1 Forecasting Data Analisis Daya Saing Pariwisata Indonesia	42
Kode Sumber 4.2.2.2 Clustering Data Segmentasi Frekuensi-Moneter	44
Kode Sumber 4.2.2.1 Create Table Data Wisatawan	47
Kode Sumber 4.3.2.2 Create View Pertumbuhan Nilai Daya Saing Pariwisata	50
Kode Sumber 4.3.3.1 Query Untuk Menampilkan Jumlah Wisatawan Mancanegara Setiap Bulannya.	52
Kode Sumber 4.3.3.2 Query Untuk Menampilkan Ranking Tahunan Berdasarkan Nilai Pangsa Jumlah Wisatawan Mancanegara	52

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Pengolahan Data Global pada Sektor Perdagangan,
Pariwisata, Investasi, dan Jasa melalui Proses ETL
(Extract, Transform, Load) serta Pengembangan
Dashboard Sistem Informasi Diplomasi Ekonomi pada
Sektor Pariwisata Indonesia**

Oleh:

Laurivasya Gadhing Syahafidh

5025211136

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Ahmad Saikhu, S.Si.,
M.T.
NIP.
197107182006041001



(Pembimbing
Departemen)

2. Hanavi, S.Kom., M.T.



(Pembimbing Lapangan)



[Halaman ini sengaja dikosongkan]

**Pengolahan Data Global pada Sektor Perdagangan,
Pariwisata, Investasi, dan Jasa melalui Proses ETL
(Extract, Transform, Load) serta Pengembangan
Dashboard Sistem Informasi Diplomasi Ekonomi pada
Sektor Pariwisata Indonesia**

Nama Mahasiswa : Laurivasya Gadhing Syahafidh
NRP : 5025211136
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dr. Ahmad Saikhu, S.Si., M.T.
Pembimbing Lapangan : Hanavi, S.Kom., M.T.

ABSTRAK

Sistem Informasi Diplomasi Ekonomi (SIDE) merupakan aplikasi dashboard yang dikembangkan untuk mendukung analisis dan pengambilan keputusan terkait diplomasi ekonomi Indonesia, terutama dalam bidang pariwisata. Tujuan dari sistem ini adalah untuk menyediakan visualisasi data yang komprehensif mengenai wisatawan mancanegara, analisis daya saing pariwisata, serta pemantauan distribusi pengeluaran wisatawan. Dashboard ini mengintegrasikan berbagai proses pengolahan data, termasuk ETL (Extract, Transform, Load), preprocessing data, dan processing data, yang memungkinkan sistem untuk memproses dan menyajikan informasi yang relevan secara efektif. Selain itu, aplikasi ini juga memanfaatkan teknik web scraping untuk mengumpulkan data dari sumber eksternal dan API data extraction untuk mengakses informasi yang diperlukan. Dalam pengelolaan data, sistem ini menggunakan Data Definition Language dan Data Manipulation).

Pengujian sistem dilakukan untuk memastikan fungsionalitasnya sesuai dengan dokumen persyaratan perangkat lunak (SRS) dan untuk memastikan bahwa data yang ditampilkan valid, konsisten, serta dapat diakses dengan baik di berbagai

perangkat. Hasil pengujian menunjukkan bahwa SIDE dapat memberikan wawasan yang berguna dan efektif dalam analisis data diplomasi ekonomi, khususnya di sektor pariwisata.

Kata Kunci : Web Scrapping, API Data Extraction, Dashboard, ETL, Preprocessing Data, Processing Data

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan program magang MSIB di Badan Strategi Kebijakan Luar Negeri (BSKLN) Kementerian Luar Negeri dengan baik.

Hal ini tentunya tidak terlepas dari dukungan, bantuan, dan motivasi dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih yang mendalam kepada:

1. Keluarga yang senantiasa memberikan semangat, dukungan, dan doa selama kegiatan magang berlangsung.
2. Bapak Dr. Ahmad Saikhu, S.Si., M.T. selaku dosen pembimbing kerja praktik departemen Teknik Informatika.
3. Bapak Hanavi, S.Kom., M.T. selaku dosen pembimbing lapangan di Badan Strategi Kebijakan Luar Negeri.
4. Rekan-rekan tim penulis yang telah bekerja sama dengan baik, saling mendukung, dan membantu dalam proses pengerjaan hingga proyek dapat diselesaikan dengan baik.
5. Semua pihak yang telah membantu dan tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, penulis menerima segala bentuk kritik dan saran yang membangun untuk perbaikan di masa depan. Akhir kata, semoga laporan ini dapat memberikan manfaat bagi para pembaca.

Surabaya, 10 Januari 2024
Laurivasya Gadhing Syahafidh

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam era globalisasi dan digitalisasi, diplomasi ekonomi memegang peranan penting dalam memperkuat hubungan internasional dan mencapai tujuan pembangunan nasional. Sektor pariwisata, perdagangan, investasi, dan jasa merupakan elemen penggerak utama dalam perekonomian Indonesia. Namun, para diplomat Indonesia menghadapi tantangan dalam mengakses data yang terfragmentasi, tidak terstruktur, dan heterogen dari berbagai sumber global. Data mentah ini sering kali sulit diolah menjadi informasi yang relevan dan dapat digunakan secara langsung untuk mendukung pengambilan keputusan strategis berbasis bukti (*evidence-based decision making*) dalam menjalin kerja sama bilateral antar negara.

Sebagai jawaban atas tantangan ini, Pengembangan dashboard sistem informasi diplomasi ekonomi dirancang sebagai platform terintegrasi yang mendukung pengambilan keputusan strategis pada sektor pariwisata. Dengan memanfaatkan teknik ekstraksi data seperti web *Scraping*, integrasi API, dan transformasi data, dashboard ini menyederhanakan proses pengolahan data mentah menjadi visualisasi yang interaktif dan informatif. Hal ini memungkinkan diplomat untuk memahami pola, tren, dan peluang ekonomi, sehingga dapat merumuskan strategi yang lebih responsif terhadap dinamika global.

1.2. Tujuan

Tujuan kerja praktik ini adalah untuk memenuhi kewajiban akademik dalam mata kuliah Kerja Praktik di Institut Teknologi Sepuluh Nopember dengan bobot dua SKS. Selain itu, kerja praktik ini bertujuan untuk mendukung Badan Strategi Kebijakan Luar Negeri (BSKLN) dalam pengembangan website dashboard sistem informasi diplomasi ekonomi yang berfokus pada sektor pariwisata. Proyek ini juga mencakup pengumpulan dan pengolahan data global periode 2000–2024 dari sektor perdagangan, pariwisata, investasi, dan jasa sebagai bagian dari proses penyediaan data yang terintegrasi.

1.3. Manfaat

Terciptanya sistem yang terintegrasi untuk mendukung diplomasi ekonomi, khususnya pada sektor pariwisata. Dashboard ini memberikan visualisasi data yang informatif dan interaktif, memungkinkan diplomat lebih cepat dalam memahami pola, tren, dan peluang ekonomi. Selain itu, BSKLN juga memiliki data dari keempat sektor yang sudah terintegrasi dari berbagai sumber data ke dalam satu *database* sehingga akan memudahkan proses pengembangan lebih lanjut ke depannya.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana mengintegrasikan data yang terfragmentasi, tidak terstruktur, dan heterogen dari keempat sektor (perdagangan, pariwisata, investasi, dan jasa) ke dalam satu database yang terorganisasi?
2. Bagaimana merancang dan mengembangkan dashboard diplomasi ekonomi yang mampu memvisualisasikan setiap komponen data dari sektor pariwisata secara interaktif, informatif, dan berbasis hasil pengolahan data?

1.5. Lokasi dan Waktu Kerja Praktik

Pelaksanaan kerja praktik ini dilaksanakan secara *full-offline* di kantor Badan Strategi Kebijakan Luar Negeri yang beralamat di Jl. Taman Pejambon No. 6 Jakarta Pusat, 10110.

Kerja praktik ini berlangsung mulai tanggal 6 September 2024 hingga 31 Desember 2024.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Untuk menentukan fitur serta data yang dibutuhkan dalam pembuatan dashboard, tim penulis terlebih dahulu mengidentifikasi *user needs* melalui diskusi dengan para pemangku kepentingan, yaitu Pak Jepri dan Pak Hanavi selaku penanggung jawab proyek. Diskusi ini bertujuan untuk memahami kebutuhan spesifik pengguna terkait fungsi dan informasi yang harus disediakan oleh dashboard, sehingga dapat

mendukung tujuan diplomasi ekonomi secara optimal. Dalam rapat tersebut, tim penulis juga mengajukan beberapa usulan fitur berdasarkan analisis awal terhadap kebutuhan proyek.

Setelah kebutuhan pengguna teridentifikasi, tim penulis menyusun dokumen standar kebutuhan sistem (*Software Requirements Specifications/SRS*), yang mencakup spesifikasi teknis, fitur utama, alur kerja dashboard, serta jenis data yang akan diintegrasikan. Dokumen ini dirancang sebagai pedoman selama proses pengembangan dashboard, memastikan setiap tahapan pekerjaan berjalan sesuai dengan kebutuhan pengguna dan standar teknis yang telah ditentukan. Pendekatan ini membantu tim penulis mengurangi risiko kesalahan atau kekurangan fitur selama proses implementasi, sehingga hasil akhir pengembangan dapat memenuhi kebutuhan pengguna secara optimal.

1.6.2. Studi Literatur

Setelah memperoleh gambaran umum mengenai pengembangan dashboard, tim penulis melakukan riset melalui berbagai literatur yang relevan dengan tujuan untuk membantu tim merumuskan fitur-fitur yang dibutuhkan serta menentukan informasi yang akan ditampilkan pada dashboard. Selain itu, pada tahap ini akan membahas mengenai tinjauan yang digunakan dalam proyek, termasuk framework pengembangan, manajemen basis data, dan library yang dipilih berdasarkan relevansi dan kesepakatan tim.

1.6.3. Implementasi Sistem

Implementasi merupakan realisasi dari dokumen *software requirement specifications*. Pada tahap ini akan dijelaskan bagaimana proses ETL dilakukan hingga akhirnya menampilkan hasil analisis dan visualisasi data.

1.6.4. Pengujian dan Evaluasi

Setelah dashboard yang dirancang telah selesai dibuat, diperlukan evaluasi untuk memastikan bahwa dashboard yang dibuat telah memenuhi ekspektasi *client*. Apabila terdapat ketidaksesuaian atau kebutuhan akan penambahan fitur, tim akan mengadakan rapat kembali untuk mendiskusikan dan menentukan fitur-fitur yang perlu diperbaiki atau ditambahkan. diperbaiki atau ditambah.

1.6.5. Kesimpulan dan Saran

Memaparkan kesimpulan yang diperoleh berdasarkan hasil pengembangan serta pengujian dan juga terdapat beberapa saran untuk ke depannya selama proses pengerjaan proyek.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari enam bab dengan rincian sebagai berikut:

1.7.1. Bab I Pendahuluan

Bab ini memaparkan latar belakang permasalahan, tujuan, manfaat, rumusan masalah, lokasi dan jadwal pelaksanaan kerja praktik, metodologi pengerjaan kerja praktik, dan sistematika penulisan laporan kerja praktik.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum terkait profil Kementerian Luar Negeri – Badan Strategi, tempat penulis melaksanakan kegiatan kerja praktik.

1.7.3. Bab III Tinjauan Pustaka

Bab ini menjelaskan mengenai tinjauan pustaka, literatur, dan teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Implementasi Sistem

Bab ini berisi uraian penjelasan mengenai tahapan-tahapan yang dilakukan untuk proses pembuatan dashboard dan ETL data.

1.7.5. Bab V Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari dashboard yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.6. Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan yang dapat diambil berdasarkan hasil pengembangan dan juga terdapat beberapa saran selama proses pengerjaan proyek.

[Halaman ini sengaja dikosongkan]

BAB II

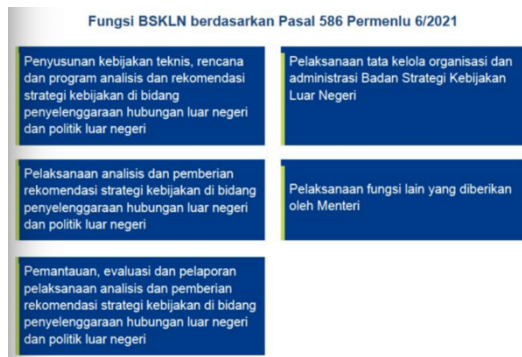
PROFIL PERUSAHAAN

2.1. Profil Kementerian Luar Negeri

Kementerian Luar Negeri merupakan kementerian yang bertugas untuk membantu Presiden menyelenggarakan pemerintahan di bidang luar negeri. Tugas pokok Kementerian Luar Negeri adalah perumusan, penetapan, dan pelaksanaan kebijakan, pelaksanaan pengkajian, dan pengembangan di bidang hubungan luar negeri dan politik luar negeri. Selain itu, Kementerian Luar Negeri juga melaksanakan dukungan yang bersifat substantif, administratif, dan pengawasan kepada seluruh unsur organisasi di lingkungan Kementerian Luar Negeri dan Perwakilan Republik Indonesia [1].

Di dalam Kementerian Luar Negeri ini sendiri kemudian dibagi lagi menjadi beberapa direktorat dan juga badan yang masing-masing mengemban tugas yang berbeda. Salah satu badan tersebut merupakan Badan Strategi Kebijakan Luar Negeri (BSKLN).

2.1.1 Profil Badan Strategi Kebijakan Luar Negeri



Gambar 2.1.1 Fungsi BSKLN Kementerian Luar Negeri

Kemudian untuk dapat menjalankan tugas dan fungsi yang diberikan dengan baik maka BSKLN kemudian dibagi lagi menjadi empat pusat wilayah kerja masing-masing, yaitu Pusat Strategi Kebijakan Multilateral (Pusat SK Muliterateral), Pusat Strategi Kebijakan Kawasan Asia Pasifik dan Afrika (Pusat SKK Aspasaf), Pusat Strategi Kebijakan Isu Khusus dan Analisis Data (Pusat SK IKAD). Selain empat pusat tersebut terdapat pula satu badan kesekretariatan yang mengurus seluruh kegiatan administrasi di dalam BSKLN [2].

Pelaksanaan kerja praktik dilakukan pada Pusat SK IKAD yang memiliki tugas untuk melaksanakan penyusunan kebijakan teknis, rencana, dan program analisis data, serta sinkronisasi perumusan, penetapan, dan pemberian rekomendasi strategis kebijakan terhadap isu khusus di bidang penyelenggaraan hubungan luar negeri dan politik luar negeri. Pusat SK IKAD kemudian

dibagi lagi menjadi tujuh kelompok jabatan fungsional, yaitu tiga bagian Isu Khusus dan empat bagian data dengan fokus yang berbeda-beda [3].

2.2. Lokasi

Kementerian Luar Negeri – Badan Strategi Kebijakan Luar Negeri, Jl. Taman Pejambon No. 6 Jakarta Pusat, 10110

2.3. Logo Perusahaan



KEMENTERIAN LUAR NEGERI

Gambar 2.3 Logo Kementerian Luar Negeri

2.4. Visi Misi Perusahaan

2.4.1. Visi

Terwujudnya wibawa diplomasi guna memperkuat jati diri sebagai negara maritim untuk kepentingan rakyat.

2.4.2. Misi

1. Memperkuat peran dan kepemimpinan Indonesia sebagai negara maritim dalam kerja sama

internasional untuk memajukan kepentingan nasional.

2. Memantapkan peran Kementerian Luar Negeri sebagai penjurur pelaksana hubungan luar negeri dengan dukungan peran aktif pemangku kepentingan nasional.
3. Mewujudkan kapasitas Kementerian Luar Negeri dan perwakilan Republik Indonesia yang mumpuni.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. *Extract, Transform, Load (ETL)*

Extract, Transform, Load (ETL) merupakan proses fundamental dalam sistem *data warehouse* yang bertujuan untuk [4] mengintegrasikan data dari berbagai sumber ke dalam sebuah *database* yang terorganisasi. Proses ETL berfungsi sebagai jembatan antara data mentah yang tersebar dari berbagai sumber data dan data yang telah diolah sehingga dapat dianalisis untuk mendukung pengambilan keputusan strategis. Dalam dunia komputasi modern, keberhasilan *data warehouse* bergantung pada kualitas proses ETL, yang mencakup tiga tahap utama: *extract* (mengekstraksi data dari berbagai sumber), *transform* (mengolah data agar sesuai dengan format atau struktur yang diinginkan), dan *load* (memuat data ke dalam sistem data warehouse) [4].

Ekstraksi (*Extract*) merupakan langkah awal dalam proses ETL, di mana data diambil dari sumber heterogen yang memiliki karakteristik dan format yang berbeda-beda. Sumber data ini dapat berupa basis data relasional seperti MySQL atau PostgreSQL, layanan berbasis API (misalnya, REST atau GraphQL), *file flat* seperti CSV atau Excel, hingga data yang diambil melalui teknik *web scraping* dari situs web. Hasil akhir dari tahap ekstraksi adalah sekumpulan data mentah yang telah dikumpulkan menjadi satu [5].

Transformasi (*Transform*) adalah tahap paling krusial dalam proses ETL, di mana data mentah diolah agar dapat digunakan secara optimal dalam

analisis atau dimasukkan ke dalam *data warehouse*. Pada tahap ini, berbagai proses dilakukan, seperti pembersihan data untuk menghapus anomali atau kesalahan, normalisasi untuk menyeragamkan struktur dan format data, serta penggabungan data dari berbagai sumber menjadi satu kesatuan. Proses ini juga mencakup penghitungan ulang atau agregasi data, seperti menghitung rata-rata atau total, untuk menyederhanakan analisis di tahap berikutnya [5].

Tahap pemuatan (*Load*) merupakan langkah akhir yang esensial untuk memindahkan data yang telah diproses ke dalam data warehouse. Pada tahap ini, data yang telah diekstraksi dan ditransformasikan dimasukkan ke dalam struktur penyimpanan yang terorganisir, seperti tabel relasional atau skema multidimensional, sesuai dengan desain *data warehouse* yang telah ditetapkan. Proses pemuatan dapat dilakukan secara *full load*, di mana seluruh dataset dimasukkan sekaligus, atau *incremental load*, yang hanya memuat data baru atau yang diperbarui sejak pemuatan terakhir [5].

3.2. Python

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum dan pertama kali diperkenalkan pada tahun 1991. Bahasa ini dirancang dengan sintaks yang sederhana, intuitif, dan berfokus pada keterbacaan, sejalan dengan filosofi Python yang disebut *The Zen of Python*. Filosofi ini mencakup prinsip-prinsip seperti "*Simple is better than complex*," dan "*Readability counts*," yang menjadi dasar dalam pengembangan dan penggunaan Python.

Seiring perkembangannya, Python telah menjadi bahasa yang sangat serbaguna, digunakan dalam berbagai

bidang mulai dari pengembangan web hingga otomatisasi proses kompleks dan analisis data, menjadikannya salah satu bahasa pemrograman terpopuler di dunia teknologi saat ini. Dalam konteks pengolahan data, Python memungkinkan proses pengumpulan, pembersihan, dan transformasi data secara efisien dari berbagai sumber tanpa memerlukan sintaks yang rumit. Selain itu, Python juga mendukung analisis data yang mendalam dengan mentransformasikannya ke dalam bentuk visualisasi yang informatif dan mudah dipahami [6].

3.3. *Web Scraping*

Web scraping adalah metode yang digunakan untuk dapat secara otomatis mengumpulkan data dari berbagai situs web menggunakan perangkat lunak atau skrip automasi khusus. Proses ini melibatkan pengambilan konten HTML dari halaman web, mengekstraksi informasi yang relevan, dan menyimpan data tersebut dalam format yang terstruktur untuk analisis lebih lanjut. Namun, teknik ini memiliki beberapa tantangan seperti perubahan struktur situs web yang dapat mengganggu proses pengumpulan data dan penyesuaian skrip automasi, pengelolaan permintaan yang tinggi yang berpotensi menyebabkan pemblokiran oleh server, serta kepatuhan terhadap regulasi hukum dan etika yang berlaku dalam proses ekstraksi data [7]. Berikut beberapa python library yang penulis gunakan dalam proses pengerjaan kerja praktik :

3.3.1. Selenium

Selenium adalah pustaka *open-source* yang dirancang untuk mengotomasi interaksi dengan *browser* web, memungkinkan pengguna untuk mengontrol dan menjalankan berbagai fitur yang tersedia pada sebuah situs web dengan meniru perilaku manusia saat menjelajahi halaman tersebut. Dengan kemampuannya menginterpretasikan perilaku pengguna, seperti melakukan klik tombol, *scrolling* pada halaman situs, dll. Selenium dapat mengakses, menavigasi, dan mengekstraksi data secara efektif, bahkan dari situs yang menggunakan teknologi dinamis seperti JavaScript. [8]

3.3.2. BeautifulSoup

Beautiful Soup adalah *library* Python yang dirancang khusus untuk mempermudah proses *parsing* dokumen HTML dan XML, yang menjadi langkah penting dalam *web Scraping*. Dengan menggunakan BeautifulSoup, pengguna dapat memahami dan menavigasi struktur *website*, seperti tabel, paragraf, atau tag HTML, lalu mengekstrak informasi yang diperlukan secara efisien [9].

3.4. API Data Extraction

Ekstraksi data melalui API (*Application Programming Interface*) adalah proses ekstraksi data dari suatu aplikasi atau situs web secara terstruktur dan otomatis tanpa adanya intervensi manual. Proses ini memerlukan akses ke API yang disediakan oleh penyedia layanan, dokumentasi API untuk memahami metode dan parameter yang digunakan,

serta kredensial otentikasi untuk menjamin keamanan akses data. Dalam implementasinya, *user* akan mengirim *request* ke *endpoint* API yang diperlukan, kemudian menerima *response* yang biasanya berformat JSON atau XML. Data yang diperoleh dari *response* tersebut dapat diolah lebih lanjut untuk memperoleh hasil analisis, visualisasi, atau mengintegrasikannya dengan sistem lain, sehingga memberikan fleksibilitas dan efisiensi dalam pengelolaan informasi [10].

3.5. *Preprocessing Data*

Preprocessing Data adalah langkah fundamental dalam proses pengolahan dan analisis data. Tahap ini bertujuan untuk mengubah data mentah menjadi format yang siap digunakan oleh model machine learning atau algoritma analisis lainnya. Data yang telah diekstraksi sering kali memiliki kualitas yang kurang optimal, seperti keberadaan *null values*, data duplikat, atau ketidakkonsistenan antar sumber. Oleh karena itu, preprocessing data menjadi langkah penting untuk memastikan integritas dan kualitas data. Berikut adalah beberapa proses utama dalam preprocessing data:

1. *Explanatory Data Analysis* (EDA)
EDA merupakan langkah eksplorasi awal untuk memahami struktur dan pola data. Tahap ini mencakup identifikasi tren, distribusi data, serta potensi masalah seperti *outlier* atau ketidakseimbangan data (*data imbalance*). Langkah ini memberikan

wawasan yang berguna untuk menentukan strategi preprocessing selanjutnya.

2. *Data Cleaning*

Proses ini berfokus pada pembersihan dataset dari masalah seperti *null values*, *outlier*, dan entri duplikat. Sebagai contoh, nilai yang hilang dapat diatasi dengan imputasi, seperti mengganti dengan rata-rata atau median, sementara *outlier* dapat ditangani dengan menghapusnya jika data tersebut tidak relevan atau mengganggu analisis.

3. *Data Integration*

Langkah ini bertujuan menggabungkan data dari berbagai sumber menjadi satu dataset yang kohesif. Tantangan utama pada tahap ini adalah mengatasi perbedaan format, skema, atau unit antar sumber data. Dengan data integration yang baik, dataset menjadi lebih konsisten dan siap digunakan untuk analisis yang lebih mendalam.

4. *Data Transformation*

Transformasi data melibatkan proses seperti normalisasi dan standarisasi, yang bertujuan untuk mengubah data mentah menjadi format yang sesuai dengan kebutuhan analisis. Sebagai contoh, mengonversi data ke dalam satuan yang seragam (seperti USD ke EUR) atau menyesuaikan nilai data agar berada dalam rentang tertentu untuk mempermudah pemrosesan oleh model [11].

3.5.1. Pandas

Library Pandas merupakan komponen utama yang digunakan dalam tahapan *preprocessing data*. Pandas adalah sebuah *library* berbasis Python yang dirancang untuk mempermudah manipulasi dan analisis data dengan struktur tabel yang dikenal sebagai *DataFrame*. Dalam proses *preprocessing*, Pandas memungkinkan pengguna untuk memuat *dataset* dari berbagai format, seperti CSV, Excel, JSON, dll, dengan menggunakan fungsi seperti `read_csv` atau `read_excel`. Setelah data dimuat, Pandas menyediakan beragam fungsi untuk melakukan pengolahan data, termasuk menangani *missing data*. Misalnya, fungsi `fillna()` yang dapat digunakan untuk menggantikan nilai yang hilang dengan rata-rata, median, atau nilai lain yang ditentukan. Selain itu, transformasi data menjadi lebih sederhana dengan fungsi seperti `apply()` yang memungkinkan penerapan operasi khusus pada kolom tertentu, atau `astype()` untuk mengubah tipe data agar sesuai dengan kebutuhan analisis.

Pandas juga memberikan kemampuan kepada pengguna untuk melakukan *filtering* data sesuai kebutuhan analisis. *Data Selection* dapat dilakukan dengan menggunakan indeks atau kondisi tertentu melalui metode seperti `loc[]` atau `iloc[]`, yang sering digunakan untuk memilih atau menghapus kolom maupun baris pada indeks tertentu. Selain itu, fungsi seperti `merge()` atau `concat()` sangat berguna untuk menggabungkan beberapa *dataset* menjadi satu, sehingga mempermudah integrasi data dari

berbagai sumber. Dengan fitur-fitur ini, Pandas menjadi salah satu *library* yang bersifat esensial dalam mempersiapkan dataset untuk analisis lebih lanjut [12].

3.5.2. Data Frame

DataFrame adalah struktur data dua dimensi yang menyerupai tabel, digunakan secara luas dalam pemrosesan data, terutama dengan *library* Pandas di Python. *DataFrame* terdiri dari baris dan kolom, di mana setiap kolom dapat menyimpan data dengan tipe yang berbeda, seperti angka, teks, atau nilai *boolean*. Struktur ini dirancang untuk mempermudah pengelolaan data, mulai dari membaca, mengolah, hingga menampilkan *dataset* dalam bentuk yang lebih terorganisir. Cara pertama untuk membuat *DataFrame* adalah dengan menggunakan struktur data *dictionary*. Dalam metode ini, *key* pada *dictionary* berfungsi sebagai nama kolom, sementara nilai-nilai yang terkait akan menjadi isi dari kolom tersebut dalam bentuk baris.

```

import pandas as pd

data = {'Nama': ['Andi', 'Budi', 'Citra'], 'Usia': [25, 30, 22]}
df = pd.DataFrame(data)
print(df)

```

[2] ✓ 0.0s

	Nama	Usia
0	Andi	25
1	Budi	30
2	Citra	22

Gambar 3.5.2.1 Implementasi DataFrame dengan Dictionary

Atribut `.shape` pada `DataFrame` digunakan untuk mengetahui dimensi atau ukuran data yang terkandung dalam `DataFrame`. Hasil dari `.shape` berupa tuple yang berisi dua elemen: jumlah baris dan jumlah kolom dalam `DataFrame`. Selain itu, penggunaan `info()` berguna untuk mengetahui index data, jumlah kolom beserta informasi tiap kolom untuk null data dan tipe data, serta jumlah total penggunaan memory pada setiap kolom dalam satuan bytes.

```

import pandas as pd

data = {'Nama': ['Andi', 'Budi', 'Citra'], 'Usia': [25, 30, 22]}
df = pd.DataFrame(data)
df.info()

```

[4] ✓ 0.0s

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Nama    3 non-null      object
1   Usia    3 non-null      int64
dtypes: int64(1), object(1)
memory usage: 180.0+ bytes

```

```

df.shape

```

[5] ✓ 0.0s

... (3, 2)

Gambar 3.5.2.2 Implementasi `.shape` dan `info()`

Lalu, misalkan kita memiliki objek DataFrame yang memiliki baris hingga jutaan, namun kita tidak ingin menampilkan data secara keseluruhan karena akan menghabiskan memory. Kita dapat menggunakan fungsi `head()` dan `tail()` untuk menampilkan data secara default untuk 5 data teratas dan 5 data terbawah.

```
import pandas as pd
pd.set_option('display.max_columns', 10)

df = pd.read_csv('rfm.csv')
```

[6] ✓ 0.0s

```
df.head()
```

[7] ✓ 0.0s

	Kode_Alpha3_Negara	Tahun	Jumlah_Kunjungan	Jumlah_Pengeluaran	RLM
0	AUS	2022	655370	1407.59	10.64
1	AUS	2023	1431177	1513.66	10.08
2	AUT	2022	17708	2767.62	19.28
3	AUT	2023	34984	4274.22	6.73
4	BEL	2022	21120	2458.84	18.57

```
df.tail()
```

[8] ✓ 0.0s

	Kode_Alpha3_Negara	Tahun	Jumlah_Kunjungan	Jumlah_Pengeluaran	RLM
57	THA	2023	111786	683.27	4.50
58	GBR	2022	170881	1909.22	14.85
59	GBR	2023	335209	3394.13	12.13
60	USA	2022	188764	2104.59	14.87
61	USA	2023	392310	2454.57	9.46

Gambar 3.5.2.3 Implementasi `head()` dan `tail()`

3.5.3. NumPy

NumPy, atau *Numerical Python*, adalah *library* yang dirancang untuk menyederhanakan pengolahan data numerik dan ilmiah. NumPy menjadi dasar bagi banyak *library* python lainnya seperti SciPy, Pandas, dan Matplotlib. Keunggulan utamanya terletak pada array N-

multidimensinya, yang menyerupai struktur data bawaan Python yaitu *list*, namun jauh lebih efisien dan cepat. Array ini memungkinkan berbagai operasi matematika kompleks dilakukan dengan optimal, karena implementasinya menggunakan bahasa C untuk meningkatkan efisiensi komputasi.

Selain mempercepat proses komputasi, NumPy juga mempermudah analisis data dengan mendukung operasi matematis yang lebih cepat melalui pendekatan vektorisasi, sehingga menghilangkan kebutuhan akan penggunaan *loop* yang biasanya memperlambat proses [13].

3.6. *Processing Data*

Processing Data adalah tahapan pengolahan *dataset* untuk diubah menjadi bentuk yang lebih terstruktur dan bermakna, sehingga dapat digunakan dalam pengambilan keputusan atau analisis lebih lanjut. Proses ini melibatkan berbagai langkah, seperti *merge data*, membuat model seperti *machine learning* (misalnya untuk *forecasting* atau *clustering*), menganalisis hubungan antar variabel, melakukan perhitungan statistik, hingga menyajikan hasil dalam bentuk visualisasi. Tujuan utamanya adalah menemukan pola, tren, atau wawasan penting yang relevan dengan kebutuhan atau masalah yang sedang dikaji.

3.6.1. Scikit-learn

Scikit-learn adalah *library* Python yang dirancang untuk menyederhanakan pengembangan dan implementasi model machine learning secara efisien. *Library* ini menyediakan berbagai algoritma, termasuk *regression*, *classification*, *clustering*, dan *dimensionality reduction*, yang dapat digunakan untuk membangun dan mengevaluasi model.

Dalam *workflow machine learning*, scikit-learn mempermudah berbagai langkah penting, seperti membagi dataset menjadi dua bagian, yaitu *training* dan *testing*, memilih model yang paling sesuai, serta melakukan optimasi parameter menggunakan teknik seperti *grid search* dan *cross-validation*. Pendekatan ini memungkinkan pengguna untuk mengembangkan *machine learning* model dengan lebih terstruktur dan efektif [14].

3.6.2. Matplotlib

Matplotlib merupakan *library* Python yang dirancang khusus untuk melakukan proses visualisasi data, memungkinkan pengguna membuat berbagai jenis grafik seperti garis, batang, *scatter*, dan histogram. Dengan kemampuan ini, Matplotlib membantu menganalisis data secara lebih mendalam melalui representasi visual yang intuitif, sehingga pola, tren, atau anomali

dalam data dapat dikenali dengan lebih mudah.

Keunggulan utama Matplotlib terletak pada fleksibilitasnya. Pengguna dapat

	<i>Points</i>	<i>Lines</i>	<i>Areas</i>	<i>Best to show</i>
<i>Shape</i>		<i>possible, but too weird to show</i>	<i>cartogram</i>	<i>qualitative differences</i>
<i>Size</i>			<i>cartogram</i>	<i>quantitative differences</i>
<i>Color Hue</i>				<i>qualitative differences</i>
<i>Color Value</i>				<i>quantitative differences</i>
<i>Color Intensity</i>				<i>qualitative differences</i>
<i>Texture</i>				<i>qualitative & quantitative differences</i>

menyesuaikan elemen-elemen grafik, seperti warna, label, dan sumbu, sesuai kebutuhan

Gambar 3.6.2 Pentunjuk Visualisasi Data Matplotlib

spesifik. Selain itu, pustaka ini mendukung pembuatan grafik statis, animasi, hingga grafik interaktif, menjadikannya sangat relevan untuk melakukan berbagai analisis jenis data [6].

3.7. Manajemen Basis Data

Manajemen basis data adalah proses sistematis yang bertujuan untuk mengelola data dalam suatu sistem terpusat, memungkinkan penyimpanan, pengorganisasian,

dan akses data secara efisien. Sistem ini dirancang agar data dapat diakses dengan mudah oleh pengguna atau aplikasi, sekaligus menjaga integritas dan keamanannya. Fungsi utama dari manajemen basis data meliputi penyediaan mekanisme untuk menyimpan data secara terstruktur, mempermudah pencarian informasi, serta mendukung pengolahan data untuk menghasilkan informasi yang bermanfaat. Dengan memanfaatkan sistem manajemen basis data, redundansi data dapat diminimalkan, konsistensi dapat ditingkatkan, dan analisis data yang diperlukan untuk pengambilan keputusan strategis dapat difasilitasi secara optimal.

3.7.1. Physical Data Model

Physical Data Model (PDM) adalah model yang menggambarkan bagaimana data disimpan secara fisik di dalam database. Model ini berfungsi dengan mengubah desain logis yang bersifat konseptual menjadi struktur konkret yang dapat diterapkan dalam sistem manajemen basis data (DBMS) tertentu. Dalam PDM, setiap elemen dirancang secara spesifik untuk memastikan data dapat disimpan dan diakses dengan efisien. Proses ini mencakup pengaturan tipe data untuk setiap kolom, seperti *integer*, *varchar*, atau *date*, yang disesuaikan dengan kebutuhan penyimpanan. Selain itu, panjang atribut juga diatur secara rinci, misalnya dengan menentukan batas maksimum karakter pada kolom teks.

Selanjutnya, PDM menetapkan *primary key* sebagai identifikasi unik untuk setiap baris

data dalam tabel, yang juga berfungsi menjaga konsistensi dan keunikan data. Untuk memastikan integritas antar tabel, *foreign key* digunakan guna menghubungkan data antara tabel yang berbeda. Di sisi lain, desain indeks menjadi salah satu elemen penting dalam PDM karena memungkinkan proses pencarian dan pengambilan data berlangsung lebih cepat dan efisien.

3.7.2. ***Relational Data Base Management System (RDBMS)***

Relational Database Management System (RDBMS) adalah sistem perangkat lunak yang dirancang untuk mengelola, menyimpan, dan mengakses data yang terstruktur dalam bentuk tabel relasional. RDBMS memanfaatkan *Structured Query Language (SQL)* sebagai bahasa standar untuk melakukan manipulasi dan pengelolaan data, memungkinkan pengguna melakukan operasi seperti pencarian, penyortiran, pembaruan, serta analisis data dengan efisien. Dalam pengerjaan proyek ini, penulis dan tim menyepakati untuk menggunakan *MySQL Workbench* sebagai alat untuk mendukung implementasi RDBMS.

3.7.3. **MySQL**

MySQL adalah sistem manajemen basis data relasional (RDBMS) *open-source* yang dikembangkan pertama kali oleh MySQL AB pada tahun 1995. MySQL dirancang untuk mengelola dan mengatur data dalam berbagai aplikasi, mulai

dari skala kecil hingga besar. Sistem ini sering digunakan dalam pengembangan aplikasi web, *e-commerce*, dan platform media sosial. Contoh populer penerapannya adalah pada Facebook untuk pengelolaan data pengguna, WordPress sebagai *backend* pengelolaan konten, dan Shopify untuk mendukung sistem *e-commerce*.

3.7.3.1. *Data Definition Language (DDL)*

Data Definition Language (DDL) adalah kumpulan perintah dalam MySQL yang digunakan untuk mendefinisikan dan mengatur struktur *database*. DDL berfokus pada bagaimana data disusun secara fisik di dalam basis data. Perintah seperti :

- CREATE TABLE: Digunakan untuk membuat tabel baru, lengkap dengan definisi kolom, tipe data, dan batasan.
- ALTER TABLE: Memungkinkan penambahan, penghapusan, atau perubahan kolom dalam tabel yang sudah ada.
- DROP TABLE: Menghapus tabel dari *database* secara permanen.

DDL sering digunakan pada tahap awal pembuatan *database* untuk memastikan struktur yang dirancang sesuai dengan kebutuhan sistem.

3.7.3.2. *Data Manipulation Language (DML)*

Data Manipulation Language (DML) adalah kumpulan perintah yang digunakan

untuk mengelola data yang ada dalam tabel *database*. DML memungkinkan pengguna untuk memanipulasi data tanpa mengubah struktur tabel itu sendiri. Perintah penting dalam DML meliputi:

- SELECT: Mengambil data dari tabel berdasarkan kriteria tertentu. Contohnya, menampilkan data pelanggan yang membeli produk tertentu.
- INSERT: Menambahkan data baru ke tabel. Misalnya, menambahkan data transaksi baru ke tabel penjualan.
- UPDATE: Mengubah data yang sudah ada di tabel. Contoh: memperbarui alamat pelanggan.
- DELETE: Menghapus data dari tabel berdasarkan kondisi tertentu. Misalnya, menghapus data produk yang tidak lagi tersedia.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari dashboard yang kami buat. Implementasi ini akan dibagi ke dalam beberapa bagian, yaitu bagian implementasi ekstraksi data (*extract*), transformasi data (*transform*), pemuatan data (*load*), dan visualisasi data ke dalam dashboard.

4.1. Implementasi Ekstraksi Data (*Extract*)

Untuk mengekstraksi data perdagangan, pariwisata, investasi, dan jasa penulis menggunakan dua teknik ekstraksi yaitu sebagai berikut :

4.1.1. Web Scraping

Berikut adalah salah satu implementasi proses *web scraping* yang digunakan untuk mengumpulkan data perdagangan ekspor-impor dari situs web trademap.org. Proses ini mencakup scraping data dari 199 negara dengan menggunakan HSCode level 2 dan 4.

```
# * Logging and Initialization
#
# Configure logging to track the scraping process.
#
# Logs are saved using the `setlog` module, capturing start,
# success, and error events.

# Initialize the TradeSpider class to manage scraping
# operations.

[tradespider]
logging                = setlog()
driver_options         = chrome_options
download_dir           =
"/Users/laurivasy/Desktop/current/kemlu/databank/trademap/data"
```

```

safebrowsing_enabled = true

# * Selenium WebDriver Configuration
#
# Setup the Chrome WebDriver with options for downloading files
directly.
#
# Chrome preferences include disabling download prompts and
enabling safe browsing.

[webdriver]
driver_path          = chromedriver
prefs_download_path =
"/Users/laurivasy/Desktop/current/kemlu/databank/trademap/data"
prompt_for_download = false
directory_upgrade   = true
safebrowsing_enabled = true

# * Login to TradeMap
#
# Login credentials are entered via Selenium WebDriver.
#
# The script waits for elements to be interactable before
proceeding.

[login]
url =
"https://www.trademap.org/Country_SelProduct_TS.aspx"
username_xpath = '//*[@id="Username"]'
password_xpath = '//*[@id="Password"]'
login_button_xpath =
'/html/body/div[3]/div/div[2]/div/div/div/form/fieldset/div[4]/d
iv/button'
iframe_id = 'iframe_login'
verify_title_contains = "Trade Map"

# * Dropdown and Form Interaction
#
# Configure interaction with form elements such as dropdowns for
HS codes and countries.

[dropdown]
hs_code_dropdown_id =
'ctl00_NavigationControl_DropDownList_ProductClusterLevel'
years_dropdown_xpath =
'/html/body/form/div[3]/table/tbody/tr[2]/td/div[1]/table/tbody/
tr[2]/td[6]/div/select/option[3]'

```

```

select_by_value      = "2"

# * Country Code Extraction
#
# Extract country codes and names using regex from an HTML
`<option>` string.

[countries]
pattern              = r"<option value=\"(\d+)\"
Title=\"(.*)\">(.*?)</option>"
output_format        = dictionary

# * Main Scraping Process
#
# Iterate through the list of countries, constructing URLs
dynamically.
#
# Skip specified countries if necessary.

[scraping]
target_url_template =
"https://www.trademap.org/Bilateral_TS.aspx?nvpm=1%7c{source_cou
ntry}%7c%7c{target_country}%7c%7cTOTAL%7c%7c%7c2%7c1%7c1%7c1%7c2
%7c1%7c2%7c1%7c1%7c1"
excluded_countries  = ["Chad"]
data_download_button =
'//*[@id="ctl00_PageContent_GridViewPanelControl_ImageButton_Tex
t"]'
wait_time_seconds   = 2

# * Cleanup
#
# Close the browser instance after completing scraping for each
country.

[cleanup]
driver_close        = true
log_cleanup_event   = true

```

Kode Sumber 4.1.1 Web Scraping trademap.org

4.1.2. API Data Extraction

API data extraction dilakukan pada situs web UN Comtrade untuk mengunduh data secara massal dari 199 negara dalam rentang waktu tahun 2000 hingga 2024.

Konfigurasi dari *API data extraction* dapat dilihat pada kode sumber berikut.

```
# * API Data Extraction
#
# This script performs bulk data extraction from the UN Comtrade
# API.
# It downloads trade data for 199 countries within a specified
# time period (2000-2024).

# The data can include FINALCLASSIC, TARIFFLINE, or other types
# of trade data.
# Files are downloaded in `.gz` format and optionally
# decompressed to `.txt`.

[api_configuration]
base_url           = "https://comtradeapi.un.org/bulk/v1/"
subscription_key   = "<Your Subscription Key>"
request_timeout_seconds = 120

# * Parameters for API Request
#
# Configure the API parameters, such as:
# - Trade data type (e.g., FINALCLASSIC, TARIFFLINE)
# - Data frequency (annual, monthly)
# - Classification code (e.g., HS)
# - Reporting country code
# - Time period for data extraction

[api_parameters]
trade_data_type     = {TARIFFLINE | FINALCLASSIC | FINAL}
type_code           = <Trade Type Code>
frequency_code      = <Frequency Code>
classification_code = <Classification Code>
reporter_code       = <Reporter Country Code>
time_period         = {2000-2024}
published_date_from = <Optional Start Date>
published_date_to   = <Optional End Date>

# * Data Download and Processing
#
# The following steps are performed:
# 1. Send an API request with the specified parameters.
# 2. Parse the JSON response to extract available file URLs.
# 3. For each file:
```

```

# - Download the file in `.gz` format.
# - Rename the file for clarity.
# - Decompress the file to `.txt` (if requested).

[data_download]
download_directory      = <Target Directory>
decompress_files        = {true | false}
file_format              = {gzip (.gz) | plain text (.txt)}

[process_steps]
1. Construct API URL based on parameters.
2. Send HTTP GET request with API key and parameters.
3. Parse JSON response:
   - Check if data is available.
   - Extract file URLs from the response.
4. For each file URL:
   - Download the file to the target directory.
   - Rename the file for clear identification.
   - If `decompress_files` is true:
     - Decompress `.gz` file to `.txt`.
     - Remove the original `.gz` file.
5. Log the status of each file download and processing step.

# * Error Handling and Logging
#
# Log errors for request failures, data unavailability, or file
download issues.
# Provide clear status messages for successful and failed
operations.

[error_handling]
request_errors          = "Log and retry if timeout or network
failure occurs."
data_availability_check = "Log if no data matches the selected
criteria."
file_download_errors    = "Log HTTP status and response text for
failed downloads."

# * Cleanup
#
# Ensure proper cleanup after the process:
# - Remove temporary files (if any).
# - Summarize the total number of files downloaded and
processed.

[cleanup]
remove_temp_files      = true

```



```
log_summary = "Total files downloaded: {count}"
```

Kode Sumber 4.1.2 Konfigurasi API Data Extraction

4.2. Implementasi Transformasi Data (*Transform*)

4.2.1. Preprocessing Data

Salah satu implementasi proses *preprocessing data* berikut digunakan untuk mengolah data kunjungan wisatawan internasional yang berasal dari beberapa file CSV. Data ini mencatat jumlah kunjungan wisatawan dari berbagai negara antara tahun 2019 hingga 2024. Proses dimulai dengan memuat file-file CSV, lalu membersihkan data dengan menghapus baris dan kolom yang tidak diperlukan. Setelah itu, nama-nama negara dipetakan ke kode Alpha-3 menggunakan library `pycountry`, data diubah formatnya, dan kolom-kolom baru seperti tahun, bulan, dan jumlah wisatawan ditambahkan. Data yang sudah dibersihkan dan diubah kemudian digabungkan menjadi satu `DataFrame`, nilai-nilai yang hilang dibersihkan, dan beberapa kolom ditambah untuk menyesuaikan dengan format tabel database. Di akhir proses, data yang sudah siap disimpan dalam file CSV untuk kemudian dimasukkan ke dalam database.

```
# * Data Processing for Tourist Visits Data
#
# This script processes data from multiple CSV files containing
# tourist visit data.
# It performs cleaning, transformation, and adds additional
# columns for further analysis.
# The final dataset is saved as a cleaned CSV.

[data_processing]
file_paths = [
    'jumlah-wisatawan-internasional/Jumlah-Kunjungan-Wisatawan-
```

```

2019.csv',
    'jumlah-wisatawan-internasional/Jumlah-Kunjungan-Wisatawan-
2020.csv',
    'jumlah-wisatawan-internasional/Jumlah-Kunjungan-Wisatawan-
2021.csv',
    'jumlah-wisatawan-internasional/Jumlah-Kunjungan-Wisatawan-
2022.csv',
    'jumlah-wisatawan-internasional/Jumlah-Kunjungan-Wisatawan-
2023.csv',
    'jumlah-wisatawan-internasional/Jumlah-Kunjungan-Wisatawan-
2024.csv'
]

# * Mapping Countries to Alpha-3 Code
#
# Use a manual mapping for countries with non-standard names or
those not recognized by pycountry.

manual_mapping = { ... } # Mapping dictionary

# Function to convert country names to Alpha-3 code
function country_to_alpha3(country_name):
    if country_name in manual_mapping:
        return manual_mapping[country_name]
    try:
        country = pycountry.countries.get(name=country_name)
        if country:
            return country.alpha_3
        else:
            return country_name
    except Exception as e:
        return country_name

# * Data Loading and Cleaning
#
# Process each CSV file, cleaning and restructuring the data.
# Steps include dropping unnecessary rows, renaming columns, and
applying the country mapping.

[data_loading_and_cleaning]
for file in file_paths:
    year = extract_year_from_filename(file)
    df = load_csv(file)

    # Drop first three rows and last column
    df = drop_rows(df, 0, 3)
    df = remove_last_column(df)

```

```

df = reset_index(df)

# Rename columns with new headers
df.columns = ['Negara'] +
generate_column_names(len(df.columns) - 1)

# Drop specific rows by index
df = drop_specific_rows(df, [10, 11, 20, 50, ...])

# Apply manual country-to-alpha3 mapping and remove
duplicates
df['Negara'] = apply_country_to_alpha3_mapping(df['Negara'])
df = drop_duplicates(df, 'Negara')

# Reshape data to long format
df_melted = melt_data(df, id_vars=['Negara'],
var_name='Bulan', value_name='Jumlah')

# Add year to the dataset
df_melted['Tahun'] = year

# Store cleaned data in a dictionary by year
dataframes[year] = df_melted

# * Combine All Years into One DataFrame
#
# Concatenate all cleaned data into a single DataFrame.

[combine_data]
clean_df = concatenate_data(dataframes)

# * Additional Data Transformation
#
# Transform 'Jumlah' to integer, replace '-' with 0, and apply
renaming.

transform_column(clean_df, 'Jumlah', '-')
rename_columns(clean_df, {'Negara': 'Kode_Alpha3_Asal',
'Jumlah': 'Jumlah_Wisatawan'})

# * Add New Columns for Further Analysis
#
# Add necessary columns for analysis, e.g., province, city,
travel purpose.

[new_columns]
new_columns = [

```

```

    'ID', 'Provinsi_Asal', 'Kota_Asal',
    'Provinsi_Tujuan', 'Kode_Alpha3_Tujuan', 'Kota_Tujuan',
    'Tujuan_Perjalanan', 'Lama_Tinggal', 'Jenis_Tempat_Tinggal',
    'Jumlah_Pengeluaran', 'Jenis_Pengeluaran',
    'Pintu_Masuk', 'Jenis_Pintu_Masuk', 'Kode_Sumber', 'Status'
]

# Initialize new columns with empty values
initialize_columns(clean_df, new_columns)

# * Filter and Adjust Data
#
# Set values for 'Status' and 'Kode_Sumber', drop missing
'Jumlah_Wisatawan'.

set_column_values(clean_df, 'Status', 'INBOUND')
set_column_values(clean_df, 'Kode_Sumber', '02')

# Convert 'Jumlah_Wisatawan' to numeric and drop missing values
convert_to_numeric(clean_df, 'Jumlah_Wisatawan')
drop_na(clean_df, 'Jumlah_Wisatawan')

# * Final Cleanup
#
# Reset index, save the final cleaned data to CSV.

final_cleanup(clean_df)

# Save to CSV file
save_to_csv(clean_df, 'clean/jumlah_kunjungan_wisatawan.csv')

# Output final cleaned data
print(clean_df)

```

Kode Sumber 4.2.1 Preprocessing Data Jumlah Wisatawan

Mancanegara Indonesia

4.2.2. Processing Data

Untuk melakukan analisis pemantauan daya saing pariwisata di setiap provinsi Indonesia setiap tahunnya, langkah pertama yang perlu dilakukan adalah perhitungan terhadap nilai indeks komposit. Nilai indeks komposit ini dihitung berdasarkan sejumlah indikator, seperti Human

Tourism, Human Resource, Price Competitiveness, Environment, Social Development, Technology Advancement, dan Infrastructure Development. Setelah memperoleh kedelapan nilai indeks yang terkait, nilai indeks komposit dapat dihitung sebagai representasi gabungan dari semua faktor tersebut. Namun, dalam studi ini, penulis berfokus pada prediksi nilai indeks komposit untuk tahun 2025 dengan hasil *preprocessing* yang telah dilakukan pada data 2018-2024. Untuk itu, penulis akan melakukan *forecasting* terhadap masing-masing nilai indeks yang ada menggunakan metode *linear regression* guna memperkirakan tren atau pola yang akan terjadi pada tahun yang akan datang.

```
# * Data Forecasting Using Linear Regression
# This script performs forecasting for specified indicators
using linear regression models.
# The goal is to predict the values for the year 2025 based on
historical data.

[setup_environment]
import pandas as pd
from sklearn.linear_model import LinearRegression
import numpy as np

# Load dataset
df = indeks

# Required columns validation
required_columns = ['Provinsi', 'Tahun', 'PCI', 'HRI', 'IDI',
'EI', 'TAI', 'HTI', 'SDI', 'Komposit']
assert all(col in df.columns for col in required_columns),
"Dataset is missing required columns."

# Define features and target columns
features = ['Tahun']
predict_columns = ['PCI', 'HRI', 'IDI', 'EI', 'TAI', 'HTI',
'SDI', 'Komposit']

[prepare_data]
```

```

# Shift year data to start from 0
df['Tahun'] = df['Tahun'] - 2018

[predict_future]
# Initialize list to store predictions for 2025
predictions_2025 = []

# Group data by 'Provinsi' and forecast for each group
for provinsi, group in df.groupby('Provinsi'):
    if len(group) < 2:
        continue

    # Forecast data for the year 2025
    future_year = 2025 - 2018
    last_row = group.iloc[-1]

    # Create a DataFrame for future data
    future_data = {'Tahun': [future_year]}
    future_df = pd.DataFrame(future_data)

    prediction = {'Provinsi': provinsi, 'Tahun': 2025}

    # Apply linear regression for each indicator to forecast
    future values
    for column in predict_columns:
        X = group[['Tahun']]
        y = group[column]

        model = LinearRegression()
        model.fit(X, y)

        predicted_value = model.predict(future_df)[0]
        prediction[column] = predicted_value

    predictions_2025.append(prediction)

[combine_predictions]
# Create a DataFrame for forecasted data
predictions_df = pd.DataFrame(predictions_2025)

# Restore the original year in the dataset
df['Tahun'] = df['Tahun'] + 2018

# Combine original data with forecasted data
final_df = pd.concat([df, predictions_df], ignore_index=True)

# Save the final DataFrame to CSV

```

```
final_df.to_csv('forecasted_dataset_2025.csv', index=False)

# Display final DataFrame
print(final_df)
```

Kode Sumber 4.2.2.1 Forecasting Data Analisis Daya Saing Pariwisata
Indonesia

Selain melakukan forecasting, penulis juga melakukan clustering terhadap data segmentasi frekuensi-monometer pangsa pariwisata Indonesia dengan mengkategorikan data ke dalam 3 kelas, yaitu: jumlah wisatawan rendah-jumlah pengeluaran rendah, jumlah wisatawan rendah-jumlah pengeluaran tinggi, dan jumlah wisatawan tinggi-jumlah pengeluaran rendah.

```
# * Clustering Data for Tourist Visits
#
# This script performs clustering on tourist visit data for 2022
and 2023 based on two features:
# 'Jumlah_Kunjungan' (Number of Visits) and 'Jumlah_Pengeluaran'
(Expenditure).
# It uses KMeans clustering and evaluates the clusters using the
Elbow method and Silhouette scores.

[data_processing]
# Load the cleaned dataset
df = load_csv('rfm_clean.csv')

# Split the data into 2022 and 2023 subsets
df_2022 = filter_data_by_year(df, 2022)
df_2023 = filter_data_by_year(df, 2023)

# Define features to use for clustering
features = ['Jumlah_Kunjungan', 'Jumlah_Pengeluaran']

# Remove rows with missing values for selected features
drop_na(df_2022, features)
drop_na(df_2023, features)

# Scale the features using StandardScaler
```

```

scaler = StandardScaler()
df_2022_scaled = scale_data(df_2022, features, scaler)
df_2023_scaled = scale_data(df_2023, features, scaler)

# * Elbow Method to Find Optimal Number of Clusters
#
# The Elbow method helps determine the best number of clusters
by plotting WCSS (within-cluster sum of squares).
function elbow_method(data, max_clusters=10, year=2022):
    wcss = []
    for i in range(1, max_clusters + 1):
        kmeans = KMeans(n_clusters=i, random_state=42)
        fit_kmeans(kmeans, data)
        wcss.append(kmeans.inertia_)

    plot_elbow_method(wcss, year)

# * Silhouette Score to Validate Clusters
#
# Silhouette scores are used to evaluate how well-separated the
clusters are. A higher score indicates better separation.
function silhouette_scores(data, max_clusters=10, year=2022):
    scores = []
    for i in range(2, max_clusters + 1):
        kmeans = KMeans(n_clusters=i, random_state=42)
        labels = fit_predict_kmeans(kmeans, data)
        score = compute_silhouette_score(data, labels)
        scores.append(score)
        print_silhouette_score(i, score)

    plot_silhouette_scores(scores, year)

# Apply Elbow Method for both 2022 and 2023
elbow_method(df_2022_scaled, year=2022)
elbow_method(df_2023_scaled, year=2023)

# Apply Silhouette Score for both 2022 and 2023
silhouette_scores(df_2022_scaled, year=2022)
silhouette_scores(df_2023_scaled, year=2023)

# Set the number of clusters
n_clusters = 3

# Apply KMeans clustering with the chosen number of clusters
kmeans_2022 = KMeans(n_clusters=n_clusters, random_state=42)
kmeans_2023 = KMeans(n_clusters=n_clusters, random_state=42)

```



```

# Assign cluster labels to the data
df_2022['Cluster'] = fit_predict_kmeans(kmeans_2022,
df_2022_scaled)
df_2023['Cluster'] = fit_predict_kmeans(kmeans_2023,
df_2023_scaled)

# Display the first few rows with assigned clusters
print_cluster_assignments(df_2022)
print_cluster_assignments(df_2023)

# * Cluster Analysis for 2022 and 2023
#
# Analyze the characteristics of each cluster by calculating
summary statistics.
function cluster_analysis(df, year):
    print_cluster_analysis_header(year)
    cluster_summary = group_by_cluster(df)
    print_cluster_summary(cluster_summary)
    plot_cluster_scatter(df, year)

# Perform cluster analysis for 2022 and 2023
cluster_analysis(df_2022, 2022)
cluster_analysis(df_2023, 2023)

# Combine the 2022 and 2023 data for further analysis
df_combined = concatenate_data([df_2022, df_2023])
sort_data(df_combined, ['Tahun', 'Kode_Alpha3_Negara'])
reset_index(df_combined)

# Display the combined data and save to CSV
print_combined_data(df_combined)
save_to_csv(df_combined, 'clustered_tourism_data.csv')

# Output final results
print(df_combined)

```

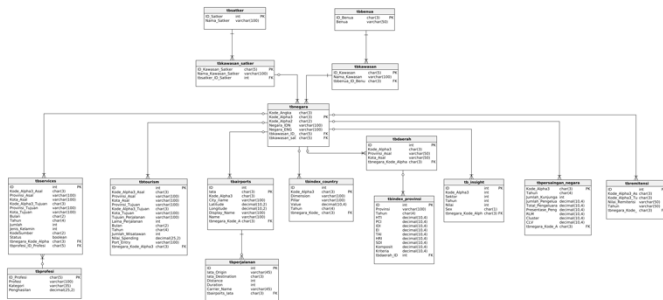
Kode Sumber 4.2.2.2 Clustering Data Segmentasi Frekuensi-Moneter

4.3. Implementasi Pemuatan Data (*Load*)

4.3.1. Physical Data Model

Sebelum membangun database, penulis terlebih dahulu merancang physical data model untuk mempermudah pembuatan struktur database yang efisien. Dengan model ini, penulis dapat merencanakan tabel-tabel yang dibutuhkan beserta kolom-kolom yang sesuai. Model ini juga memudahkan penentuan tipe data yang tepat untuk setiap kolom, serta bagaimana hubungan antar tabel dapat diatur secara optimal. Secara keseluruhan, physical data model ini berfungsi sebagai panduan yang jelas dalam proses pembuatan dan pengelolaan database, sehingga dapat meningkatkan performa sistem dan memudahkan pemeliharaan data kedepannya.

4.3.2. Data Definition Language (DDL)



Gambar 4.3.1 Physical Data Model Database Side

Pada bagian ini, akan dijelaskan beberapa penerapan DDL (*Data Definition Language*) yang digunakan untuk mendefinisikan struktur dan tampilan data dalam database, seperti pembuatan tabel dan view. Beberapa contoh implementasinya termasuk pembuatan

tabel untuk menyimpan data wisatawan dan view untuk menyederhanakan query dalam menghitung pertumbuhan nilai setiap tahunnya terhadap indeks pemantauan daya saing pariwisata di provinsi-provinsi Indonesia.

```
-- -----  
-- Table structure for tbtourism  
-- -----  
DROP TABLE IF EXISTS `tbtourism`;  
CREATE TABLE `tbtourism` (  
  `ID_Tourism` INT NOT NULL AUTO_INCREMENT,  
  `Kode_Alpha3_Asal` CHAR(3) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Provinsi_Asal` VARCHAR(100) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Kota_Asal` VARCHAR(100) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Kode_Alpha3_Tujuan` CHAR(3) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Provinsi_Tujuan` VARCHAR(100) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Kota_Tujuan` VARCHAR(100) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Tujuan_Perjalanan` VARCHAR(100) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Lama_Tinggal` DECIMAL(10, 2) NULL DEFAULT NULL,  
  `Jenis_Tempat_Tinggal` VARCHAR(50) CHARACTER SET utf8mb4  
COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Bulan` INT NULL DEFAULT NULL,  
  `Tahun` YEAR NULL DEFAULT NULL,  
  `Jumlah_Wisatawan` INT NULL DEFAULT NULL,  
  `Jumlah_Pengeluaran` DECIMAL(10, 2) NULL DEFAULT NULL,  
  `Jenis_Pengeluaran` VARCHAR(50) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Pintu_Masuk` VARCHAR(50) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Kode_Sumber` CHAR(2) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  `Status` enum('INBOUND','OUTBOUND') CHARACTER SET latin1  
COLLATE latin1_swedish_ci NULL DEFAULT NULL,  
  `Jenis_Pintu_Masuk` VARCHAR(50) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL,  
  PRIMARY KEY (`ID_Tourism`) USING BTREE,  
  FOREIGN KEY (`Kode_Alpha3_Asal`) REFERENCES `tbnegara`  
(`Kode_Alpha3`) ON DELETE SET NULL ON UPDATE CASCADE
```

```
) ENGINE = MyISAM CHARACTER SET = utf8mb4 COLLATE =  
utf8mb4_unicode_ci ROW_FORMAT = DYNAMIC;
```

Kode Sumber 4.2.2.1 Create Table Data Wisatawan

```
DROP VIEW IF EXISTS `05_provinsi_pertumbuhan_general`;  
CREATE VIEW `05_provinsi_pertumbuhan_general` AS  
SELECT  
    'HRI' AS Index_Name,  
    p1.Provinsi,  
    p1.HRI AS Jumlah_Awal,  
    p2.HRI AS Jumlah_Akhir,  
    ROUND(((p2.HRI - p1.HRI) / p1.HRI) * 100, 2) AS  
Pertumbuhan,  
    ROW_NUMBER() OVER (ORDER BY  
    (ROUND(((p2.HRI - p1.HRI) / p1.HRI) * 100, 2)) DESC) AS  
Ranking  
FROM  
    tbindex_provinsi p1  
JOIN  
    tbindex_provinsi p2 ON p1.Provinsi =  
p2.Provinsi  
WHERE  
    p1.Tahun = '2019'  
    AND p2.Tahun = '2025'  
GROUP BY  
    p1.Provinsi, p1.HRI, p2.HRI  
UNION ALL  
SELECT  
    'HTI' AS Index_Name,  
    p1.Provinsi,  
    p1.HTI AS Jumlah_Awal,  
    p2.HTI AS Jumlah_Akhir,  
    ROUND(((p2.HTI - p1.HTI) / p1.HTI) * 100, 2) AS  
Pertumbuhan,  
    ROW_NUMBER() OVER (ORDER BY  
    (ROUND(((p2.HTI - p1.HTI) / p1.HTI) * 100, 2)) DESC) AS  
Ranking  
FROM  
    tbindex_provinsi p1  
JOIN  
    tbindex_provinsi p2 ON p1.Provinsi =  
p2.Provinsi  
WHERE  
    p1.Tahun = '2019'  
    AND p2.Tahun = '2025'
```

```

GROUP BY
    p1.Provinsi, p1.HTI, p2.HTI
UNION ALL
SELECT
    'EI' AS Index_Name,
    p1.Provinsi,
    p1.EI AS Jumlah_Awal,
    p2.EI AS Jumlah_Akhir,
    ROUND(((p2.EI - p1.EI) / p1.EI) * 100, 2) AS
Pertumbuhan,
    ROW_NUMBER() OVER (ORDER BY
    (ROUND(((p2.EI - p1.EI) / p1.EI) * 100, 2)) DESC) AS
Ranking
FROM
    tbindex_provinsi p1
JOIN
    tbindex_provinsi p2 ON p1.Provinsi =
p2.Provinsi
WHERE
    p1.Tahun = '2019'
    AND p2.Tahun = '2025'
GROUP BY
    p1.Provinsi, p1.EI, p2.EI
UNION ALL
SELECT
    'IDI' AS Index_Name,
    p1.Provinsi,
    p1.IDI AS Jumlah_Awal,
    p2.IDI AS Jumlah_Akhir,
    ROUND(((p2.IDI - p1.IDI) / p1.IDI) * 100, 2) AS
Pertumbuhan,
    ROW_NUMBER() OVER (ORDER BY
    (ROUND(((p2.IDI - p1.IDI) / p1.IDI) * 100, 2)) DESC) AS
Ranking
FROM
    tbindex_provinsi p1
JOIN
    tbindex_provinsi p2 ON p1.Provinsi =
p2.Provinsi
WHERE
    p1.Tahun = '2019'
    AND p2.Tahun = '2025'
GROUP BY
    p1.Provinsi, p1.IDI, p2.IDI
UNION ALL
SELECT
    'KOMPOSIT' AS Index_Name,

```

```

        p1.Provinsi,
        p1.Komposit AS Jumlah_Awal,
        p2.Komposit AS Jumlah_Akhir,
        ROUND(((p2.Komposit - p1.Komposit) / p1.Komposit) *
100, 2) AS Pertumbuhan,
        ROW_NUMBER() OVER (ORDER BY
        (ROUND(((p2.Komposit - p1.Komposit) / p1.Komposit) *
100, 2)) DESC) AS Ranking
    FROM
        tbindex_provinsi p1
    JOIN
        tbindex_provinsi p2 ON p1.Provinsi =
p2.Provinsi
    WHERE
        p1.Tahun = '2019'
        AND p2.Tahun = '2025'
    GROUP BY
        p1.Provinsi, p1.Komposit, p2.Komposit
    UNION ALL
    SELECT
        'PCI' AS Index_Name,
        p1.Provinsi,
        p1.PCI AS Jumlah_Awal,
        p2.PCI AS Jumlah_Akhir,
        ROUND(((p2.PCI - p1.PCI) / p1.PCI) * 100, 2) AS
Pertumbuhan,
        ROW_NUMBER() OVER (ORDER BY
        (ROUND(((p2.PCI - p1.PCI) / p1.PCI) * 100, 2)) DESC) AS
Ranking
    FROM
        tbindex_provinsi p1
    JOIN
        tbindex_provinsi p2 ON p1.Provinsi =
p2.Provinsi
    WHERE
        p1.Tahun = '2019'
        AND p2.Tahun = '2025'
    GROUP BY
        p1.Provinsi, p1.PCI, p2.PCI
    UNION ALL
    SELECT
        'SDI' AS Index_Name,
        p1.Provinsi,
        p1.SDI AS Jumlah_Awal,
        p2.SDI AS Jumlah_Akhir,
        ROUND(((p2.SDI - p1.SDI) / p1.SDI) * 100, 2) AS
Pertumbuhan,

```

```

                ROW_NUMBER() OVER (ORDER BY
                (ROUND(((p2.SDI - p1.SDI) / p1.SDI) * 100, 2)) DESC) AS
Ranking
FROM
                tbindex_provinsi p1
JOIN
                tbindex_provinsi p2 ON p1.Provinsi =
p2.Provinsi
WHERE
                p1.Tahun = '2019'
                AND p2.Tahun = '2025'
GROUP BY
                p1.Provinsi, p1.SDI, p2.SDI
UNION ALL
SELECT
                'TAI' AS Index_Name,
                p1.Provinsi,
                p1.TAI AS Jumlah_Awal,
                p2.TAI AS Jumlah_Akhir,
                ROUND(((p2.TAI - p1.TAI) / p1.TAI) * 100, 2) AS
Pertumbuhan,
                ROW_NUMBER() OVER (ORDER BY
                (ROUND(((p2.TAI - p1.TAI) / p1.TAI) * 100, 2)) DESC) AS
Ranking
FROM
                tbindex_provinsi p1
JOIN
                tbindex_provinsi p2 ON p1.Provinsi =
p2.Provinsi
WHERE
                p1.Tahun = '2019'
                AND p2.Tahun = '2025'
GROUP BY
                p1.Provinsi, p1.TAI, p2.TAI
ORDER BY Index_Name, Ranking, Provinsi;
SELECT * FROM `05_provinsi_pertumbuhan_general`;

```

Kode Sumber 4.3.2.2 Create View Pertumbuhan Nilai Daya Saing
Pariwisata

4.3.3. Data Manipulation Language (DML)

Pada bagian ini, akan dibahas penerapan DML (*Data Manipulation Language*) yang digunakan untuk mengelola dan memanipulasi data dalam database, seperti

melakukan query untuk menampilkan informasi yang diperlukan. Contoh penerapan DML dalam konteks ini termasuk menampilkan jumlah wisatawan mancanegara yang datang ke Indonesia setiap bulannya serta menampilkan ranking tahunan berdasarkan nilai pangsa jumlah wisatawan mancanegara.

```

WITH Pangsa AS (
    SELECT
        n.Kode_Alpha2 AS Alpha2,
        t.Kode_Alpha3_Asal AS Alpha3,
        n.Negara_IDN AS Negara,
        t.Tahun,
        SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun,
t.Kode_Alpha3_Asal) AS Jumlah_Per_Tahun,
        SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun) AS
Jumlah_Total,
        (SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun,
t.Kode_Alpha3_Asal)
        / SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun))
* 100 AS Pangsa_Pasar
    FROM
        tbtourism t
    JOIN
        tbnegara n ON t.Kode_Alpha3_Asal = n.Kode_Alpha3
    WHERE
        t.Kode_Sumber = '2'
    GROUP BY
        t.Tahun, t.Kode_Alpha3_Asal
    ORDER BY
        n.Negara_IDN
),
RankedNegara AS (
    SELECT
        Alpha2,
        Alpha3,
        Negara,
        Tahun,
        Pangsa_Pasar,
        RANK() OVER (PARTITION BY Tahun ORDER BY Pangsa_Pasar
DESC) AS Ranking
    FROM
        Pangsa
)
SELECT * FROM RankedNegara;

```


Kode Sumber 4.3.3.1 Query untuk menampilkan jumlah wisatawan mancanegara setiap bulannya.

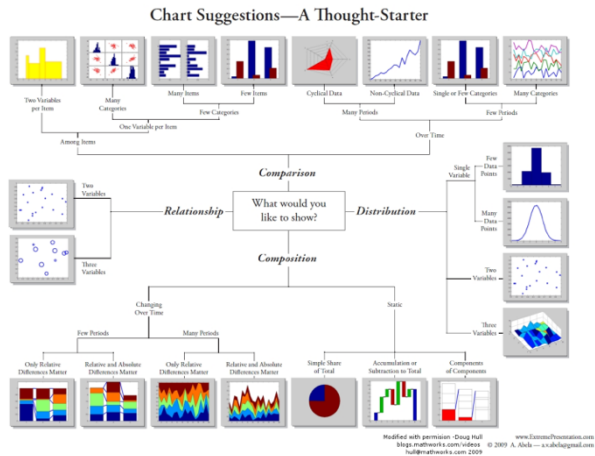
```
WITH Pangsa AS (  
    SELECT  
        n.Kode_Alpha2 AS Alpha2,  
        t.Kode_Alpha3_Asal AS Alpha3,  
        n.Negara_IDN AS Negara,  
        t.Tahun,  
        SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun,  
t.Kode_Alpha3_Asal) AS Jumlah_Per_Tahun,  
        SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun) AS  
Jumlah_Total,  
        (SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun,  
t.Kode_Alpha3_Asal)  
        / SUM(t.Jumlah_Wisatawan) OVER (PARTITION BY t.Tahun))  
* 100 AS Pangsa_Pasar  
    FROM  
        tbtourism t  
    JOIN  
        tbnegara n ON t.Kode_Alpha3_Asal = n.Kode_Alpha3  
    WHERE  
        t.Kode_Sumber = '2'  
    GROUP BY  
        t.Tahun, t.Kode_Alpha3_Asal  
    ORDER BY  
        n.Negara_IDN  
),  
RankedNegara AS (  
    SELECT  
        Alpha2,  
        Alpha3,  
        Negara,  
        Tahun,  
        Pangsa_Pasar,  
        RANK() OVER (PARTITION BY Tahun ORDER BY Pangsa_Pasar  
DESC) AS Ranking  
    FROM  
        Pangsa  
)  
SELECT * FROM RankedNegara;
```

Kode Sumber 4.3.3.2 Query untuk menampilkan ranking tahunan berdasarkan nilai pangsa jumlah wisatawan mancanegara

4.4. Implementasi Visualisasi Data

Tampilan seluruh visualisasi data pada dashboard dapat diakses melalui link [berikut](#). Setiap data yang ditampilkan pada masing-masing grafik diambil dari setiap *view* yang berisi *query* yang telah dibuat oleh penulis. Di sisi *backend*, sistem akan memanggil *view* yang ada di database untuk memuat data sesuai dengan kriteria visualisasi yang ditampilkan. Proses ini memungkinkan data ditarik secara efisien berdasarkan kebutuhan visualisasi yang spesifik, memastikan konsistensi dan relevansi informasi yang ditampilkan pada dashboard.

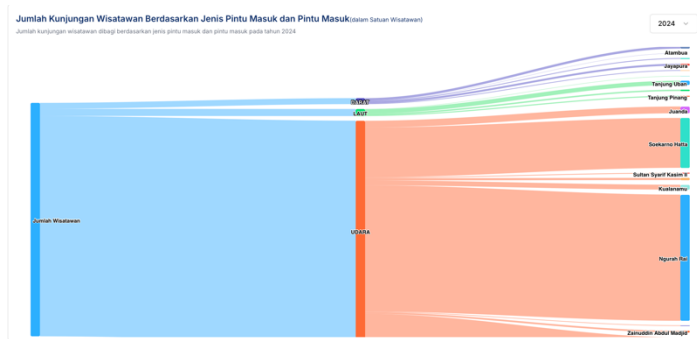
Untuk menentukan setiap visualisasi yang ada di dalam dashboard SIDE, penulis menggunakan *Andrew Abella Chart* sebagai alat bantu untuk memilih bentuk visualisasi data yang paling tepat. Metode ini membantu dalam menentukan representasi grafis yang paling sesuai dengan karakteristik dan tujuan analisis data, memastikan



Gambar 4.4 Andrew Abella Chart

pemahaman yang lebih jelas dan efektif bagi pengguna dashboard.

Berikut adalah salah satu contoh grafik yang bernama *sankey chart* untuk menampilkan proporsi jumlah kedatangan wisatawan mancanegara melalui setiap jenis pintu masuk yang ada di Indonesia setiap tahunnya.



Gambar 4.4 Visualisasi proporsi jumlah kedatangan wisatawan mancanegara melalui setiap jenis pintu masuk

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap dashboard Sistem Informasi Diplomasi Ekonomi (SIDE). Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi dashboard dengan *Software Requirements Specifications* (SRS) yang telah disusun sebelumnya.

5.1. Tujuan Pengujian

Pengujian dilakukan terhadap dashboard Sistem Informasi Diplomasi Ekonomi untuk menguji kesesuaian hasil analisis data dengan melakukan validasi berdasarkan sumber data yang digunakan.

5.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

- a. Hasil analisis dan visualisasi pada dashboard harus mencerminkan data yang valid dan konsisten dengan sumber data yang digunakan. Pengujian akan memastikan bahwa data yang ditampilkan tidak ada kesalahan atau inkonsistensi antara data mentah dan hasil visualisasi.
- b. Semua fitur yang tertera dalam dokumen SRS harus terimplementasi dengan baik di dashboard.
- c. Dashboard harus mampu memuat data dengan cepat dan tidak mengalami lag atau masalah kinerja, baik saat mengambil data dari

- database maupun saat menampilkan visualisasi.
- d. Visualisasi yang ada di dashboard harus mudah dipahami oleh pengguna dan menyajikan informasi secara jelas dan efektif.
 - e. Pengujian harus memverifikasi bahwa dashboard memungkinkan interaksi pengguna yang mudah, seperti filter data, perubahan tampilan visualisasi, dan penggunaan fitur kontrol lain yang relevan dengan analisis.
 - f. Dashboard harus dapat diakses dengan baik di berbagai perangkat dan ukuran layar, serta dapat menyesuaikan tampilan berdasarkan perangkat yang digunakan.

5.3. Skenario Pengujian

Skenario pengujian dilakukan dengan menjalankan berbagai fitur di dalam dashboard untuk memastikan fungsionalitas yang diharapkan. Beberapa halaman utama yang diuji meliputi:

1. Tampilan data jumlah wisatawan mancanegara
2. Tampilan lama tinggal wisatawan mancanegara
3. Distribusi pengeluaran wisatawan mancanegara berdasarkan tujuan kedatangan dan komoditas
4. Pencarian jadwal penerbangan internasional menuju indonesia
5. Distribusi kedatangan wisatawan mancanegara berdasarkan setiap jenis pintu masuk yang ada di indonesia
6. Analisis pemantauan daya saing pariwisata per provinsi
7. Indeks daya saing pariwisata negara

8. Segmentasi frekuensi-moneter

Masing-masing halaman diuji untuk memastikan bahwa data yang ditampilkan sesuai dengan tujuan fungsionalitas dan dapat diakses dengan mudah oleh pengguna.

5.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai kriteria pengujian terhadap kasus skenario uji coba pada dashboard SIDE. Tabel 5.4. di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

Tabel 5.4. Hasil Evaluasi Pengujian

Kriteria Pengujian	Hasil Pengujian
Analisis dan visualisasi di dashboard harus sesuai dengan data yang valid dan konsisten, tanpa kesalahan atau inkonsistensi.	Terpenuhi
Semua fitur dalam dokumen SRS harus terimplementasi dengan baik di dashboard.	Terpenuhi
Dashboard harus memuat data dengan cepat dan tanpa masalah kinerja.	Terpenuhi
Visualisasi harus jelas, mudah	Terpenuhi

dipahami, dan menyajikan informasi secara efektif.	
Pengujian harus memastikan kemudahan interaksi pengguna, termasuk penggunaan fitur filter data dan perubahan tampilan.	Terpenuhi
Dashboard harus dapat diakses dengan baik di berbagai perangkat dan menyesuaikan tampilan sesuai ukuran layar.	Terpenuhi

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Kesimpulan yang didapat setelah berhasil melakukan pembuatan dashboard SIDE (Sistem Informasi Diplomasi Ekonomi) yang dapat menyajikan data dan hasil analisis yang penting bagi pengambilan keputusan dalam bidang diplomasi ekonomi dan pariwisata. Melalui pengujian yang dilakukan, dashboard ini terbukti dapat menyajikan data secara valid dan konsisten dengan sumber yang digunakan, serta mampu memvisualisasikan data dengan cara yang jelas dan mudah dipahami. Semua fitur yang tertera dalam dokumen SRS telah terimplementasi dengan baik, dan dashboard mampu memberikan pengalaman pengguna yang interaktif, responsif, serta dapat diakses di berbagai perangkat.

6.2. Saran

Beberapa saran yang dapat penulis berikan demi perbaikan program di Pusat Strategi Kebijakan Isu Khusus dan Analisis Data (SK IKAD) ke depannya, antara lain:

- 1. Timeline Pengerjaan dan Alokasi Tugas yang Lebih Jelas**

Selama program, pembagian tugas untuk proyek tambahan sering kali mengalami perubahan, sehingga peserta kesulitan menyesuaikan prioritas dan rencana kerja. Selain itu, timeline pengerjaan proyek tidak didefinisikan secara spesifik, yang menyebabkan kurangnya kepastian terkait tenggat waktu dan target pencapaian. Untuk meningkatkan pelaksanaan di masa mendatang, kejelasan dalam pembagian tugas

dan penentuan timeline perlu diperhatikan. Dengan timeline yang terperinci, seperti penetapan tenggat waktu untuk setiap tahapan proyek—mulai dari perencanaan, pengembangan, hingga *deployment*—peserta akan memiliki panduan yang lebih baik dan dapat merencanakan waktu secara efektif untuk menyelesaikan tugas sesuai prioritas.

2. Perincian Kebutuhan *End-User* dengan Lebih Detail
Kebutuhan *end-user* yang dirinci secara mendalam sangat penting dalam pengembangan proyek teknologi informasi. Brief yang lebih detail, termasuk deskripsi kebutuhan fungsional, preferensi desain, dan ekspektasi akhir, akan memberikan arah yang lebih jelas kepada para pengembang. Hal ini akan mengurangi interpretasi yang salah atau kebutuhan untuk revisi yang berulang kali. Dengan perincian yang lebih detail, proses pengembangan dapat dilakukan dengan lebih terarah, menghasilkan solusi yang tepat guna dan memenuhi harapan stakeholders.
3. Penyediaan Infrastruktur yang Lebih Memadai
Penyediaan infrastruktur yang lebih memadai, seperti data internal yang relevan dan *server* dengan spesifikasi tinggi, sangat penting untuk mendukung kelancaran pengembangan proyek. Data internal yang lengkap dapat mempercepat proses kerja para pengembang dengan mengurangi ketergantungan pada data eksternal yang memakan waktu. Di sisi lain, server yang andal memungkinkan pengujian aplikasi dan kolaborasi tim dengan lebih mudah. Dengan memenuhi kebutuhan ini, program dapat memberikan pengalaman belajar yang lebih optimal, mendukung keberhasilan proyek yang dihasilkan, dan berkontribusi pada pengembangan diplomasi ekonomi Indonesia.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Kementerian Luar Negeri, “Rencana Kerja dan Anggaran Kementerian Luar Negeri 2023,” 2023.
- [2] Badan Strategi Kebijakan Luar Negeri (BSKLN), “Laporan Kinerja BSKLN TA 2023,” Kementerian Luar Negeri, 2023.
- [3] Pusat Strategi Kebijakan Isu Khusus dan Analisis Data (Pusat SK IKAD), “Laporan Kinerja Pusat Strategi Kebijakan Isu Khusus dan Analisis Data 2023,” 2023.
- [4] A. Walha, F. Ghozzi dan F. Gargouri, “Data integration from traditional to big data: main features and comparisons of ETL approaches,” *The Journal of Supercomputing*, vol. 80, 2024.
- [5] G. Devi, “An efficient hybrid optimization of ETL process in data warehouse of big data,” *Journal of Cloud Computing*, vol. 13, 2024.
- [6] A. Nagpal dan G. Gabrani, “Python for Data Analytics, Scientific and Technical Applications,” dalam *Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, United Arab Emirates, 2019.
- [7] C. Lotfi, S. Srinivasan, M. Ertz dan I. Latrous, “Web Scraping Techniques and Applications: A Literature Review,” 2021.
- [8] S. Fatima, S. Luqmaan dan N. Rasheed, “Web Scraping with Python and Selenium,” *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 23, no. 3, 2021.

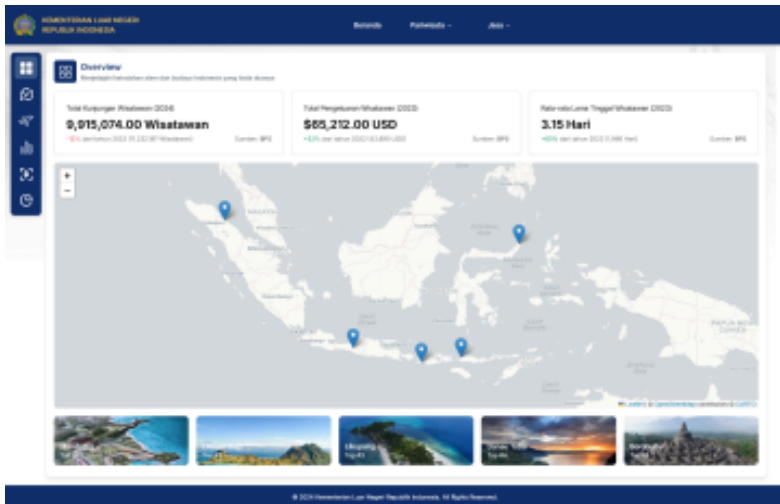
- [9] A. Abodayeh, R. Hejazi, W. Najjar, L. Shihadeh dan R. Latif, “Web Scraping for Data Analytics: A BeautifulSoup Implementation,” dalam *Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, Riyadh, Saudi Arabia, 2023.
- [10] A. Ehsan, M. Abuhaliqa, C. Catal dan D. Mishra, “RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions,” *Applied Sciences*, 2022.
- [11] S. García, S. Ramírez-Gallego dan J. e. a. Luengo, “Big data preprocessing: methods and prospects,” *Big Data Analytics*, vol. 1, no. 9, 2016.
- [12] W. Mckinney, “Pandas: a Foundational Python Library for Data Analysis and Statistics,” *Python High Performance Science Computer*, 2011.
- [13] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, ... dan T. E. Oliphant, “Array programming with NumPy,” *Nature*, 2020.
- [14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. .. Grisel dan E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, 2011.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

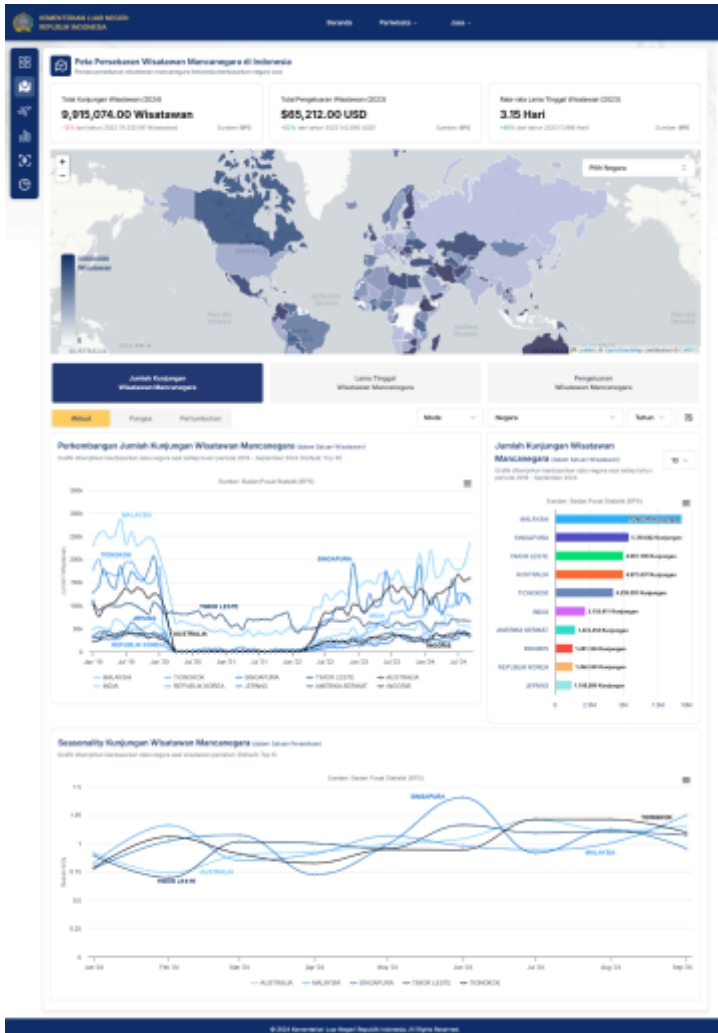


Gambar I: Tampilan Beranda SIDE Parekraf



Gambar II: Tampilan Overview Pariwisata

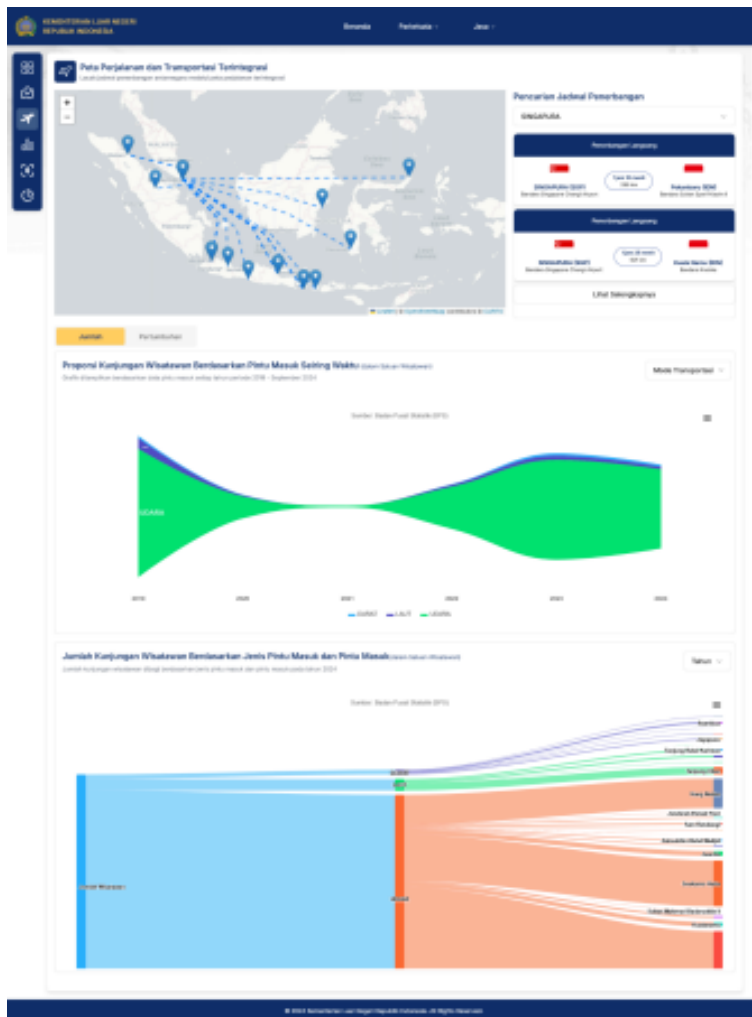
Fitur pariwisata terdiri dari beberapa halaman. Halaman pertama merupakan halaman *overview* yang menampilkan data-data umum seperti total kunjungan wisatawan, total pengeluaran wisatawan, rata-rata lama tinggal wisatawan, dan peta persebaran top destinasi wisata di Indonesia.



Gambar III: Tampilan Peta Persebaran Wisatawan

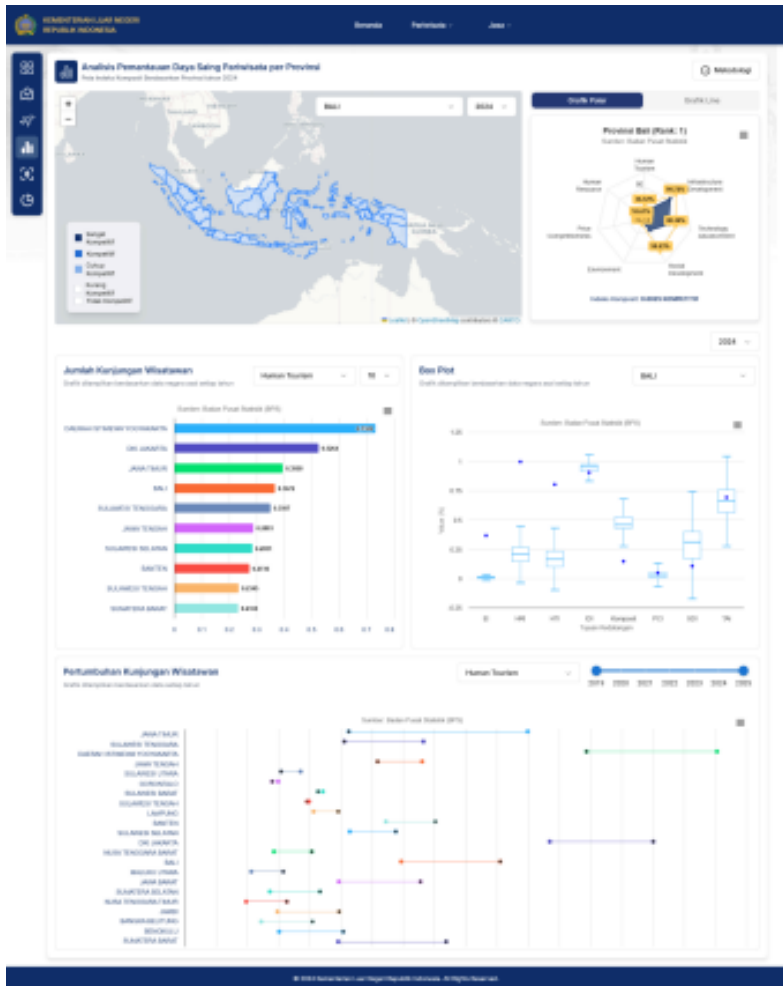
Halaman peta persebaran wisatawan menampilkan berbagai macam grafik untuk visualisasi jumlah kunjungan wisatawan,

lama tinggal wisatawan, dan pengeluaran wisatawan berdasarkan data aktual, pangsa, dan pertumbuhan.



Gambar IV: Tampilan Peta Perjalanan dan Transportasi

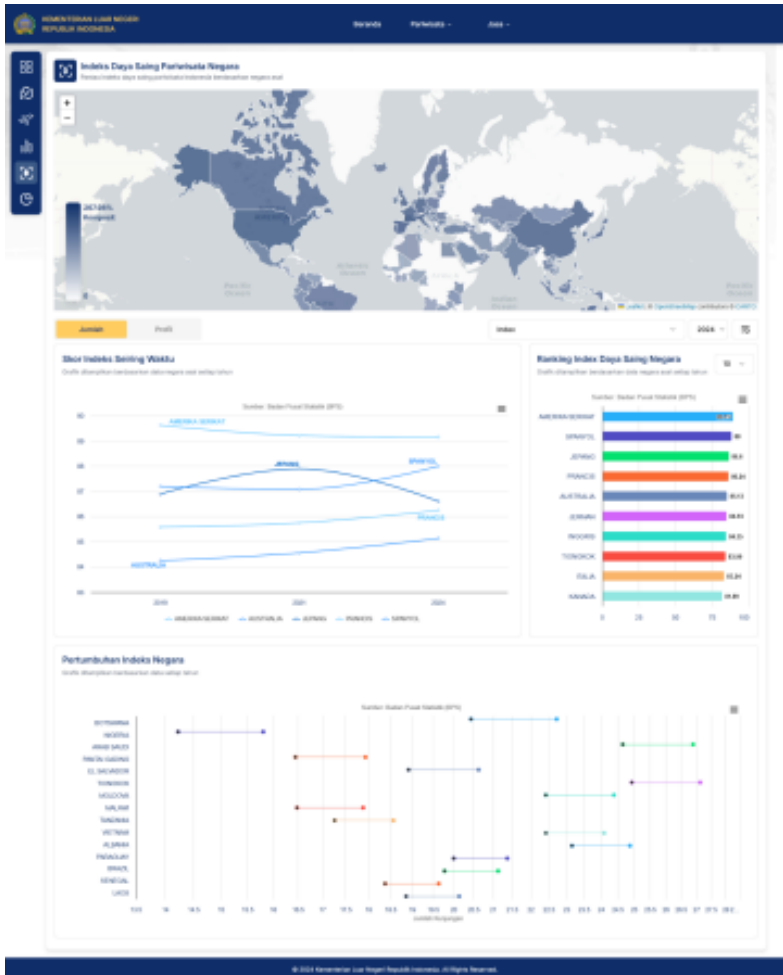
Halaman peta persebaran dan transportasi memberikan informasi terkait rute dan jadwal penerbangan yang juga divisualisasikan ke dalam peta, grafik proporsi kunjungan wisatawan berdasarkan pintu masuk yang dapat difilter berdasarkan moda transportasi, grafik jumlah kunjungan wisatawan per tahun, *ranking* pertumbuhan jumlah kunjungan wisatawan, dan grafik pertumbuhan *year-on-year* jumlah kunjungan wisatawan.



Gambar V: Tampilan Analisis Pemantauan Daya Saing Pariwisata per Provinsi

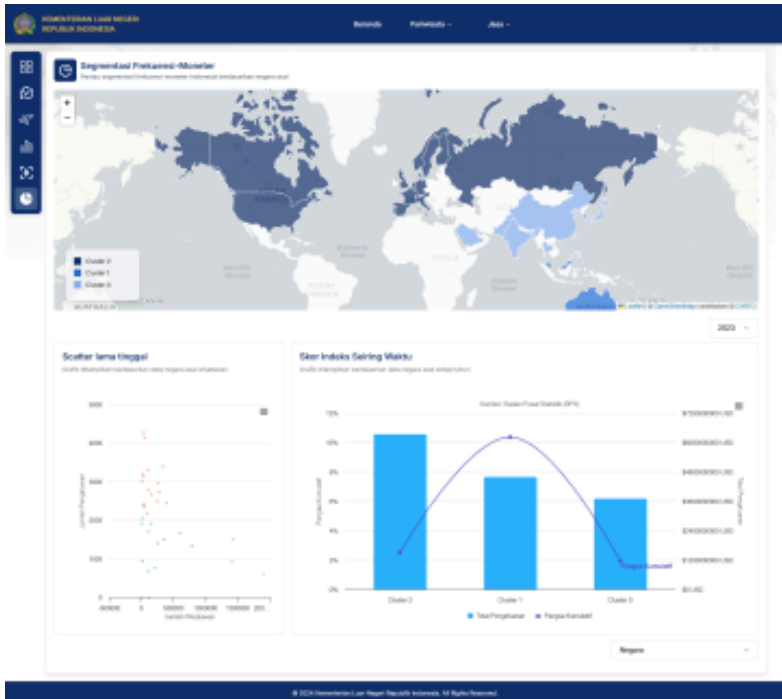
Halaman pemantauan daya saing pariwisata per provinsi menampilkan grafik polar dan *line chart* untuk tujuh indeks yang mempengaruhi daya saing per provinsi. Selain itu, ditampilkan juga statistik jumlah kunjungan wisatawan, *box plot*, dan

pertumbuhan jumlah kunjungan wisatawan dalam *range* tahun tertentu untuk suatu indeks yang dipilih.



Gambar VI: Tampilan Indeks Daya Saing Pariwisata Negara

Halaman indeks daya saing pariwisata negara menampilkan *heatmap* negara berdasarkan *ranking* dan nilai indeks kompositnya. Selain itu, ditampilkan juga *chart* skor indeks seiring waktu, *ranking* indeks daya saing negara, dan pertumbuhan indeks negara yang dapat difilter berdasarkan indeks dan tahun.



Gambar VII: Tampilan Segmentasi Frekuensi-Moneter

Halaman segmentasi frekuensi-moneter menampilkan *heatmap* dan *chart* segmentasi frekuensi-moneter Indonesia berdasarkan negara asal yang sudah di-*cluster*.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : Laurivasya Gadhing Syahafidh
Tempat, Tanggal Lahir : Kediri, 7 Mei 2003
Jenis Kelamin : Perempuan
No. Handphone : (+62)81249112211
Email : laurivasya.gadhing@gmail.com

AKADEMIK

Kuliah : Departemen Teknik Informatika
– Fakultas Teknologi Elektro dan
Informatika Cerdas, Institut
Teknologi Sepuluh Nopember

Angkatan : 2021
Semester : 7 (Tujuh)