LAPORAN
ANALISIS 3 FAKTOR PENYEBAB JUMLAH TINDAK PIDANA DI INDONESIA

Disusun oleh :

1. Ivan Cahya Aryasuta (24031554172)
2. Firda Nurkhairani (24031554209)

Kelas : 2024D

Mata Kuliah : Data Wrangling

Dosen Pengampu :

1. Dinda Galuh Guminta, S.Stat.,M.Stat. (NIDN : 0011129602)
2. Belgis Ainatul Iza, S.Si., M.Mat. (NIP : 202509237)

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

PROGRAM STUDI S1-SAINS DATA

TAHUN 2025

KATA PENGANTAR

Puji syukur yang kita panjatkan ke hadirat kepada Tuhan Yang Maha Esa karena atas berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan proyek ini yang berjudul "*Analisis 3 Faktor Terhadap Jumlah Tindak Pidana di Indonesia Tahun 2023"*. Pada proyek ini merupakan bagian dari tugas akhir dari mata kuliah "Data Wrangling" yang bertujuan untuk mengintegrasikan berbagai sumber data resmi dan mengaplikasikannya ke dalam pengolahan data menggunakan bahasa pemrograman Python. Melalui proses ekstraksi, pembersihan, dan penggabungan data dari tiga sumber berbeda penulis berusaha menyajikan data yang lebih komprehensif mengenai hubungan antara pembangunan manusia, kondisi ketenagakerjaan, tingkat kemiskinan dan tingkat kriminalitas di berbagai provinsi yang ada di Indonesia.

Dengan latar belakang dari proyek ini berawalan dari hipotesis penulis bahwa rendahnya kualitas pembangunan manusia dapat mengakibatkan berdampak pada peningkatan angka pengangguran yang mendorong tingginya tingkat kemiskinan untuk berpotensi meningkatkan tindak pidana. Oleh karena itu pendekatan raw driven penulis juga menggunakan teknik wrangling untuk menyatukan data IPM, pengangguran, tingkat kemiskinan dan tindak pidana ke dalam satu kerangka analisis agar terstruktur. Diharapkan dari hasil proyek ini bisa menjadi langkah pijakan awal untuk menganalisis lanjutan seperti korelasi atau pemodelan prediktif serta memberikan kontribusi kecil dalam memahami dinamika sosial ekonomi yang ada di Indonesia secara lebih objektif dan berdasarkan data.

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital pada saat ini ada data yang akurat sehingga menjadi komponen penting dalam pengambilan keputusan di berbagai macam bidang termasuk pemerintahan, ekonomi, dan sosial. Salah satu tantangan utama dalam analisis data adalah memperoleh data yang akurat dan terstruktur dari sumber resmi. Badan Pusat Statistik (BPS) sebagai lembaga penyedia data nasional yang menyediakan berbagai informasi penting namun, sering kali data tersebut tersebar dalam berbagai format dan halaman web sehingga memerlukan teknik khusus untuk mengakses dan mengolahnya secara efisien. Salah satu teknik tersebut adalah scraping yaitu berupa web scraping, pdf scrapping, dan lain sebagainya.

Web scraping merupakan salah satu cara digunakan untuk mengekstraksi data secara otomatis dari suatu situs web. Dalam konteks data wrangling mempunyai kemampuan untuk melakukan scraping, membersihkan, dan menggabungkan data menjadi satu format yang siap dianalisis merupakan keterampilan yang relevan dan aplikatif. Melalui tugas ini penulis melakukan scraping terhadap data Indeks Pembangunan Manusia (IPM), tingkat pengangguran, dan jumlah tindak pidana tahun 2023 dari sumber resmi yang kemudian mengintegrasikannya ke dalam satu file CSV dapat digunakan untuk ditinjau lebih mendalam.

Di laporan ini disusun sebagai bagian dari pemenuhan tugas akhir mata kuliah "Data Wrangling", dengan tujuan untuk menyampaikan runtutan proses pengambilan, pembersihan, dan transformasi data, serta menyajikan hasil analisis awal terhadap indikator sosial ekonomi di Indonesia berdasarkan data yang telah didapat.

1.2 Rumusan Masalah

Dalam menyusun laporan ini penulis dapat merumuskan beberapa poin utama yang menjadi fokus utama dalam pengambilan data ini yaitu:

1. Bagaimana cara memperoleh data dari sumber resmi seperti contoh BPS secara otomatis dan legal menggunakan teknik web scraping?
2. Apa saja tahapan yang diperlukan untuk membersihkan, menggabungkan, dan mentransformasikan data agar siap digunakan dalam format CSV yang terstruktur dan dapat dianalisa?
3. Bagaimana cara menyajikan data hasil scraping dalam bentuk visualisasi dan analisis agar dapat memberikan insight terhadap kondisi sosial ekonomi di Indonesia?
4. Apa tantangan yang dialami dalam proses scraping dan bagaimana solusi yang dapat diterapkan untuk mengatasi masalah tersebut?

1.3 Tujuan

Adapun tujuan utama penulis dalam penyusunan laporan ini yaitu:

1. Mengotomatisasi pengambilan data dari situs resmi menggunakan teknik web scraping dengan bahasa python supaya proses akuisisi data menjadi efisien, bisa digunakan kembali dan bebas dari kesalahan yang terjadi secara manual (*human error*).
2. Membersihkan dan menggabungkan data dalam format CSV
3. Menyediakan dataset final dalam format CSV yang siap digunakan untuk menganalisis lanjutan seperti clustering, korelasi, atau machine learning dasar.
4. Mensimulasikan skenario nyata di mana scraping menjadi satu-satunya cara untuk memperoleh data yang tidak tersedia dalam format unduhan langsung.

1.4 Manfaat

Adapun manfaat utama penulis dalam penyusunan laporan ini yaitu :

1. Meningkatkan keterampilan mahasiswa dalam mengakses dan mengolah data dari sumber resmi.
2. Menyediakan dataset baru yang terintegrasi dan dapat digunakan sebagai analisis lanjutan.
3. Menjadi referensi dan insight bagi mahasiswa lain yang ingin belajar scraping dari situs resmi melalui dokumentasi kode dan laporan yang sudah dilakukan dengan jelas.

BAB II
LANDASAN TEORI

2.1 Data Wrangling

Data wrangling dikenal juga sebagai data mining adalah suatu proses yang mengubah dan mempersiapkan data mentah menjadi format lebih terstruktur dan siap dianalisis lebih lanjut. Dalam proses ini mencakup berbagai tahapan seperti akuisisi data, pembersihan, transformasi, integrasi, dan validasi. Di konteks menganalisis data modern data wrangling menjadi tahap krusial karena sebagian besar data yang tersedia tidak langsung dapat langsung digunakan tanpa melalui proses tersebut.

2.2 Web Scraping

Web scraping adalah proses otomatis untuk mengekstraksi data dan informasi dari situs web yang kemudian dapat disimpan ke dalam berkas lokal atau basis data untuk dianalisis. Teknik scraping ini memanfaatkan dalam bahasa pemrograman di Python seperti Requests, BeautifulSoup, Selenium

- Teknik scraping Requests : Pengiriman permintaan HTTP ke server situs web untuk mengambil data, yang kemudian dianalisis untuk mengekstrak data.
- Teknik scraping BeautifulSoup : Untuk mengekstrak data dari halaman web dengan menggunakan Python BeautifulSoup (bs4) untuk mengurai dan menavigasi dari struktur dokumen ke HTML dan XML
- Teknik scraping Selenium : Digunakan untuk mengotomatisasi pengambilan data dari situs web yang memiliki data dinamis

Etika dan legalitas scraping data publik merupakan metode efektif untuk memperoleh data yang penggunaannya harus diperhatikan etika dan aspek hukum yang berlaku.

- Etika : Tidak membebani dalam server situs web
- Legalitas : Data harus berasal dari sumber yang diizinkan untuk digunakan kembali atau dari sumber terbuka (open data). Situs resmi pemerintah, seperti Badan Pusat Statistik (BPS), menyediakan data publik yang dapat diakses secara legal.

2.3 Sumber Data

Raw data adalah data awal yang diperoleh sebelum mengalami proses pengolahan. Pada raw data ini diperoleh dari beberapa sumber, antara lain:

1. Badan Pusat Statistik (BPS) : data Indeks Pembangunan Manusia (IPM) 2023 dalam format PDF yang kemudian dikonversi menjadi tabel.
2. Kaggle : dataset tingkat pengangguran terbuka dalam bentuk csv
3. Badan Pusat Statistik (BPS) : Jumlah Tindak Pidana
4. Open Data Jabar : data penduduk miskin

Seluruh raw data ini kemudian disatukan untuk membentuk dataset gabungan yang digunakan dalam analisis.

2.4 Indikator Sosial Ekonomi

Indikator sosial ekonomi adalah variabel yang digunakan untuk mengukur kondisi sosial dan tingkat perkembangan suatu wilayah. Pada penelitian ini digunakan beberapa indikator utama, yaitu:

1. Indeks Pembangunan Manusia (IPM) : mengukur kualitas pembangunan manusia berdasarkan pendidikan, kesehatan, dan standar hidup.
2. Tingkat Pengangguran Terbuka (TPT) : persentase penduduk usia kerja yang aktif mencari pekerjaan namun belum memperoleh pekerjaan.
3. Tingkat Kemiskinan : proporsi penduduk dengan pengeluaran di bawah garis kemiskinan.
4. Tindak kriminal merupakan indikator sosial yang menggambarkan tingkat keamanan suatu wilayah. Data ini biasanya mencakup jumlah yang terjadi dalam kurun waktu tertentu.

BAB III
METODOLOGI

3.1 Alat dan Lingkungan

Penelitian ini dilaksanakan yang menggunakan lingkungan komputasi berbasis *Python* melalui platform Jupyter Notebook. Beberapa pustaka yang digunakan di dalam proses analisis antara lain:

- Pandas : digunakan untuk membaca, membersihkan, menggabungkan, dan melakukan transformasi data.
- NumPy : digunakan untuk manipulasi nilai numerik dan operasi matematis dasar.
- Matplotlib dan Seaborn : digunakan untuk membuat visualisasi grafik, seperti *bar chart*, *scatter plot*, *heatmap*, dan *line chart*.
- Requests dan BeautifulSoup : digunakan dalam proses *web scraping* untuk mengambil data dari situs sumber.
- OpenPyXL : digunakan untuk menyimpan dan membaca data Excel hasil akhir.

Seluruh proses pengolahan dilakukan pada laptop dengan sistem operasi Windows, menggunakan Python versi 3.x. Data akhir hasil *wrangling* disimpan dalam format Excel (gabungan sudah clean.xlsx) untuk analisis visual lanjutan.

3.2 Proses Scraping

Tahapan *web scraping* dilakukan untuk mengumpulkan data dari beberapa laman resmi yang menyediakan informasi sosial ekonomi dan kriminalitas. Proses scraping ini memiliki dengan tahapan-tahapan berikut:

- Mengakses situs target menggunakan pustaka *Requests* untuk mengambil konten HTML.
- Mem-parsing struktur HTML menggunakan *BeautifulSoup* untuk mengekstrak tabel atau elemen halaman yang berisi data penting.
- Mengidentifikasi elemen tabel seperti <table>, <tr>, dan <td> untuk memperoleh setiap baris data.
- Menyusun data ke dalam DataFrame Pandas sehingga data mentah dapat mudah diolah.

- Menyimpan hasil scraping ke dalam file CSV atau langsung ke DataFrame untuk tahap *cleaning* berikutnya.

Beberapa data akan diambil dari sumber situs resmi statistik dan publikasi tahunan yang menyediakan indikator sosial ekonomi seperti TPT, IPM, tingkat kemiskinan, serta data kriminalitas tiap provinsi. Dari seluruh data mentah kemudian digabungkan menjadi satu sumber utama untuk proses wrangling.

3.3 Pembersihan dan Transformasi Data

Tahapan pembersihan (cleaning) dan transformasi dilakukan untuk memastikan data benar-benar siap untuk dianalisis. Berikut ini langkah-langkah yang diterapkan berdasarkan kode ditulis:

1. Menghapus Missing Value

   Menggunakan fungsi seperti df.dropna() atau pengisian nilai kosong apabila fungsi itu diperlukan. Data yang tidak lengkap dihapus untuk menjaga kualitas analisis.
2. Standarisasi Nama Kolom

   Melakukan penyesuaian nama kolom agar konsisten seperti mengganti spasi menjadi underscore, menyamakan huruf kecil, dan memperbaiki penamaan kolom yang tidak seragam antar dataset.
3. Pengecekan dan Konversi Tipe Data

   Beberapa kolom di konversi dari *string* menjadi numerik dengan fungsi astype(float) untuk menghindari error pada visualisasi dan analisis korelasi.
4. Pembersihan Data Non-Numerik

   Hilangkan karakter seperti koma, persen, atau teks tambahan pada data angka agar data dapat dianalisis secara matematis.
5. Normalisasi dan Transformasi

   Beberapa variabel dilakukan untuk transformasi ringan seperti perhitungan ulang persentase atau penyesuaian unit jika diperlukan.
6. Validasi Data

   Data akan melakukan pengecekan seperti:
   - memastikan tidak ada duplikasi data,
   - memastikan jumlah provinsi konsisten,
   - memastikan rentang nilai sesuai tabel publikasi resmi.

Tahapan ini menghasilkan dataset bersih yang kemudian disimpan dalam file gabungan sudah clean.xlsx yang menjadi sumber utama untuk proses visualisasi dan eksplorasi data.

3.4 Integrasi dan Penyimpanan

Proses integrasi dilakukan dengan menggabungkan beberapa dataset hasil scraping dan dataset eksternal ke dalam satu tabel utama. Penggabungan dilakukan menggunakan fungsi *merge* agar Pandas bisa melakukan berdasarkan kolom ‘provinsi’ yang menjadi kunci penyatuan data.

Tahapan integrasi yang akan dilakukan meliputi:

1. Menyatukan data TPT, IPM, tingkat kemiskinan, dan tindak kriminal ke dalam satu DataFrame utama.
2. Mengecek kesesuaian jumlah baris (setiap provinsi harus muncul satu kali).
3. Menghilangkan redundansi kolom yang tidak diperlukan
4. Melakukan final cleaning untuk memastikan seluruh nilai telah dalam format angka yang siap divisualisasikan.

Setelah data terintegrasi dan bersih lalu menyimpannya ke dalam file Excel gabungan sudah clean.xlsx. File ini kemudian digunakan pada Bab 4 untuk eksplorasi serta pembuatan visualisasi seperti grafik per provinsi dan *heatmap* korelasi.

BAB IV
ANALISIS DAN HASIL DATA

4.1 Analisis Semua Visualisasi

Pada bagian ini dijelaskan satu per satu visualisasi yang dibuat beserta interpretasinya. Visualisasi ini umumnya digunakan dalam proyek yang meliputi: grafik batang (bar chart), histogram, boxplot, scatter plot (dengan atau tanpa garis regresi), dan heatmap korelasi. Untuk setiap visualisasi dijabarkan apa yang ditampilkan, bagaimana membacanya, dan temuan penting.

4.1.1 **Grafik Batang** : Perbandingan IPM, Pengangguran, dan Kemiskinan per Provinsi

Grafik batang akan menampilkan nilai indikator (mis. IPM, TPT, persentase kemiskinan) untuk setiap provinsi. Sumbu-x menampilkan provinsi, sedangkan sumbu-y menampilkan nilai indikator. Interpretasi penting:

- Urutan dan perbandingan: Provinsi dengan batang tertinggi atau terendah mudah diidentifikasi misalnya provinsi A memiliki IPM tertinggi sementara provinsi B memiliki tingkat kemiskinan tertinggi.
- Pola regional: Jika provinsi-provinsi di satu pulau menunjukkan nilai yang serupa bisa dapat disimpulkan dengan adanya kecenderungan regional.
- Variabilitas antar provinsi: Perbedaan tinggi batang menunjukkan disparitas antar wilayah.

Temuan contoh (format yang bisa diganti sesuai angka nyata): banyak mayoritas provinsi dengan IPM tinggi menunjukkan tingkat kemiskinan relatif rendah, ada beberapa anomali (provinsi X) yang memiliki IPM moderat namun kemiskinan tinggi sebab indikasi ketimpangan dalam distribusi pendapatan atau akses layanan.

4.1.2 **Histogram** : Distribusi Setiap Indikator

Histogram menampilkan sebaran frekuensi nilai suatu indikator (mis. IPM atau TPT). Interpretasi:

- Bentuk distribusi: Normal, skew kanan atau kiri, atau multimodal.
- Kepadatan nilai: Area yang paling sering muncul (mode).
- Deteksi outlier: Bin yang terpisah jauh mengindikasikan nilai ekstrim.

Temuan contoh: IPM cenderung mendekati distribusi normal dengan sedikit skew ke kiri, tingkat pengangguran menunjukkan skew kanan dalam artian sebagian kecil provinsi memiliki pengangguran sangat tinggi.

4.1.3 Boxplot : Variasi dan Outlier per Indikator

Boxplot memvisualisasikan median, kuartil, dan outlier. Interpretasi:

- Median vs mean (jika digabung): Perbedaan menunjukkan kemiringan distribusi.
- Rentang antar-kuartil (IQR): Menggambarkan variabilitas utama.
- Outlier: Titik di luar batas IQR ±1.5*IQR dapat menandai kesalahan data atau kasus istimewa yang perlu dikaji.

Temuan contoh: beberapa provinsi menunjukkan outlier pada indikator kemiskinan yang harus ditelusuri apakah karena kesalahan input atau kondisi lokal ekstrem.

4.1.4 Scatter Plot & Regresi : Hubungan Antar Variabel

Scatter plot menggambarkan hubungan antar dua variabel (mis. IPM versus TPT, IPM versus kemiskinan). Menambahkan garis regresi linear untuk membantu melihat tren umum. Interpretasi:

- Arah hubungan: Positif (keduanya naik bersama) atau negatif (salah satu naik yang lain turun).
- Kekuatan hubungan: Kerapatan titik di sekitar garis regresi menunjukkan kekuatan hubungan.
- Cluster / kelompok: Titik yang membentuk kelompok menandakan kelas provinsi yang serupa.

Temuan contoh: IPM dan kemiskinan menunjukkan korelasi negatif di provinsi dengan IPM tinggi cenderung memiliki kemiskinan rendah. Namun dengan adanya titik-titik yang menyebar cukup lebar menandakan korelasi tidak sempurna dan faktor lain ikut berperan.

4.1.5 Heatmap Korelasi

Heatmap menampilkan matriks korelasi antar semua indikator. Warna menunjukkan arah dan besaran korelasi. Interpretasi:

- Korelasi kuat atau lemah: Nilai mendekati ±1 menandai hubungan kuat bahwa mendekati 0 menandai hubungan lemah.
- Pola keseluruhan: Misalnya IPM berkorelasi negatif kuat dengan kemiskinan tetapi korelasi lemah dengan variabel X.

Temuan contoh: heatmap menunjukkan korelasi negatif moderat antara IPM dan kemiskinan ($r \approx -0.6$) dalam korelasi negatif lemah antara IPM dan TPT ($r \approx -0.3$), serta korelasi positif antara kemiskinan dan TPT ($r \approx 0.4$).

4.1.6 Visualisasi Tambahan (Peta/Choropleth bila tersedia)

Jika tersedia peta choropleth per provinsi, visualisasi ini membantu menilai sebaran spasial indikator. Hal ini berguna untuk mengidentifikasi klaster geografis (mis. pulau yang umum memiliki IPM rendah). Interpretasi:

- Klaster spasial: Kawasan yang berkumpul menandakan faktor regional (infrastruktur, kebijakan lokal).
- Kaitan ruang waktu: ketika digabung dengan seri waktu dapat dilihat tren peningkatan/penurunan.

Temuan contoh: pulau Y memiliki banyak provinsi dengan IPM rendah dan tingkat kemiskinan tinggi akan berpotensi intervensi kebijakan kelompok wilayah.

4.2 Korelasi dan Eksplorasi Data Lebih Dalam

Bagian ini juga memadukan analisis korelasi formal dengan eksplorasi statistik yang lebih mendalam karena bukan hanya angka korelasi tetapi juga pemahaman distribusi, kelompok, dan signifikansi.

4.2.1 Statistik Ringkasan

Untuk setiap variabel (IPM, TPT, kemiskinan, dsb.) disediakan: count, mean, std (standar deviasi) min, 25% kuantil, median (50%), 75% quantile, dan max. Interpretasi:

- Mean vs median: Perbedaan menunjukkan skew jika mean lebih dari median maka hasilnya skew kanan.
- Std besar: Menandakan variasi antar provinsi tinggi.
- Rentang (min-max): Menunjukkan disparitas ekstrem antar wilayah.

Contoh temuan: rata-rata IPM nasional X dengan standar deviasi Y menunjukkan adanya perbedaan antar provinsi yang perlu perhatian.

4.2.2 Matriks Korelasi (Pearson dan Spearman)

Hitung korelasi Pearson (mengukur hubungan linier) dan Spearman (mengukur hubungan monoton). Interpretasi:

- Pearson besar (positif atau negatif): Hubungan linier kuat.
- Spearman berbeda dari Pearson: Menandakan hubungan non-linier/monoton.
- Uji signifikansi: Gunakan p-value untuk menilai apakah korelasi signifikan secara statistik.

Contoh temuan: Pearson(IPM, Kemiskinan) = -0.6 ($p < 0.01$) hasilnya korelasi negatif moderat signifikan sedangkan Spearman sebanding menunjukkan tren robust terhadap peringkat.

4.2.3 Eksplorasi Distribusi dan Kelompok (Grouping)

Lakukan grouping untuk melihat pola per cluster contohnya:

- Kelompokkan beberapa provinsi berdasarkan kuartil IPM dan bandingkan dengan rata-rata pengangguran dan kemiskinan di tiap kuartil.
- Bandingkan rata-rata antar wilayah geografis (misal di Sumatera, Jawa, Kalimantan, dsb.) untuk mendeteksi ketimpangan yang regional.

Interpretasi:

- Jika kuartil IPM tertinggi menunjukkan TPT dan kemiskinan lebih rendah secara konsisten sebagai bukti hubungan terstruktur.
- Jika satu wilayah memiliki performa buruk di semua indikator maka rekomendasi kebijakan ada di wilayah.

Contoh temuan: Provinsi dalam kuartil IPM terendah memiliki rata-rata kemiskinan dua kali lipat dibanding kuartil tertinggi.

4.2.4 Deteksi Outlier dan Validasi Data

Identifikasi titik yang menyimpang (menggunakan boxplot atau IQR atau Z-score). Untuk setiap outlier:

- Periksa sumber data apakah nilai benar atau error input.
- Jika valid, analisis kontekstual (mis. bencana alam tahun tertentu, kebijakan lokal).

Contoh temuan: Provinsi X menunjukkan pengangguran ekstrim lalu setelah di cek, menyebabkan data survei berbeda definisi maka perlu harmonisasi definisi.

4.2.5 Analisis Pairwise & Regresi Sederhana

Lakukan regresi linier sederhana untuk pasangan penting (mis. IPM sebagai dependent, kemiskinan dan pengangguran sebagai independent variables). Evaluasi:

- Koefisien regresi: Arah pengaruh (positif atau negatif) dan besaran.
- R-squared: Persentase variasi IPM yang dijelaskan oleh variabel independen.
  P-values: Signifikansi koefisien.

Contoh temuan: Kemiskinan berpengaruh negatif signifikan terhadap IPM (koef -0.4, $p < 0.01$) sedangkan pengangguran berpengaruh negatif tetapi kurang signifikan ($p \approx 0.08$).

4.2.6 Eksplorasi Multivariat / Klaster

Jika ingin melakukan clustering (k-means atau hierarchical) pada setiap provinsi yang berdasarkan vektor indikator untuk menemukan kelompok provinsi serupa. Langkah ini yang dilakukan untuk mencakup beberapa hal yaitu:

- Mengidentifikasi prototipe provinsi (maju, menengah, tertinggal).
- Menyusun rekomendasi kebijakan yang tersegmentasi.

Temuan contoh: Tiga klaster muncul yaitu klaster A (IPM tinggi, kemiskinan rendah), klaster B (menengah), klaster C (IPM rendah & kemiskinan tinggi).

4.2.7 Eksplorasi Tren Temporal (Jika Data Time-Series Tersedia)

Jika dataset berisi beberapa tahun untuk analisis tren perubahan IPM, kemiskinan, dan TPT yaitu:

- Grafik garis per provinsi atau rata-rata nasional.
- Cek percepatan/perlambatan perubahan.

Interpretasi: Tren peningkatan IPM nasional X% per tahun dengan percepatan di provinsi tertentu.

4.3 Insight

Bagian insight ini berisi ringkasan singkat dari temuan-temuan yang bisa diterapkan berdasarkan visualisasi dan eksplorasi sebelumnya ada batasan-batasan dalam menganalisis serta saran tindakan selanjutnya.

4.3.1 Ringkasan Temuan Utama

1. Hubungan IPM dan Kemiskinan: Terdapat korelasi negatif moderat antara IPM dan tingkat kemiskinan yaitu provinsi dengan IPM lebih tinggi umumnya menunjukkan bahwa kemiskinan lebih rendah.
2. Hubungan IPM dan Pengangguran: Hubungan negatif lebih lemah sedangkan pengangguran berpengaruh tetapi tidak sekuat kemiskinan terhadap IPM.
3. Disparitas Antar Provinsi: Variasi antar provinsi cukup besar yakni beberapa provinsi menunjukkan kondisi sosial ekonomi jauh lebih buruk dibandingkan dengan rata-rata nasional.
4. Outlier dan Data Issues: Terdapat beberapa nilai ekstrim yang perlu diverifikasi sumbernya dalam hal ini menandakan perlunya validasi definisi dan harmonisasi sumber data.

4.3.2 Implikasi Kebijakan / Praktis

- Intervensi terfokus: Daerah provinsi yang memiliki IPM rendah dan tingkat kemiskinan tinggi memerlukan program yang terpadu seperti pendidikan, kesehatan, dan peluang kerja.
- Perbaikan definisi dan pengumpulan data: Menyamakan definisi penilaian indikator antar sumber (misalnya pengukuran pengangguran) akan meningkatkan kualitas dalam menganalisis kebijakan.
- Pemantauan dan evaluasi: Direkomendasikan pembuatan dashboard pemantauan secara berkala untuk mengetahui dampak kebijakan terhadap berbagai indikator sosial dan ekonomi.

4.3.3 Keterbatasan Analisis

- Keterbatasan data cross-sectional: Sebagian besar analisis hanya berupa gambaran sebagian (satu tahun) tetapi untuk sulit membuat kesimpulan sebab-akibat tanpa adanya data jangka waktu.
- Kemungkinan adanya bias dari sumber data: Beberapa dataset berasal dari sumber yang berbeda dengan cara pengukuran yang tidak sama.
- Dibutuhkan variabel tambahan: Faktor-faktor yang mempengaruhi IPM mungkin lebih rumit (seperti kualitas pelayanan publik, infrastruktur, dan investasi) tidak semua sudah termasuk.

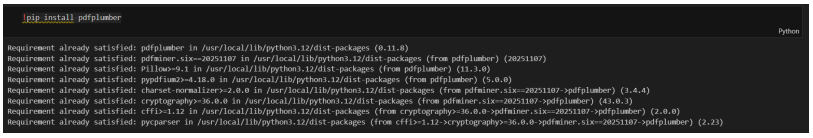
4.3.4 Rekomendasi Analisis Lanjutan

1. Analisis kausalitas (mis. regresi panel atau metode quasi-experimental) jika tersedia data yang berurutan dengan waktu.
2. Model multivariat untuk memasukkan variabel kontrol (pendapatan per kapita, pendidikan, kesehatan).
3. Eksplorasi spasial lebih lanjut dengan pemetaan dan analisis klaster spasial (Moran's I) untuk mendeteksi autocorrelation wilayah.
4. Validasi data dengan kembali ke sumber raw data untuk outlier yang terdeteksi.

4.4 Hasil output

Untuk link collab atau code : project data wrangling.ipynb

1. Menginstall pdfplumber



Untuk awalan disini pakai menginstall library pdfplumber agar mengekstrak teks dan tabel dari file pdf.

2. Ekstraksi pdf menjadi csv dari sumber data ke 1

```
import pdfplumber
import pandas as pd

pdf_path = "/content/statistik-indonesia-2025.pdf"
output_csv = "ipm sudah ekstrak.csv"

target_page_pdf = 364

extracted_data = []

print(f"Mengekstrak Data Tabel dari halaman file PDF ke-{target_page_pdf}...")

try:
    with pdfplumber.open(pdf_path) as pdf:
        if target_page_pdf <= len(pdf.pages):
            page = pdf.pages[target_page_pdf - 1]
            table = page.extract_table()

            if table:
                for row in table:
                    if row and len(row) >= 4:
                        provinsi = row[0]
                        nilai_2023 = row[3]

                        if not provinsi:
                            continue

                        if "Provinsi" in provinsi or "Province" in provinsi:
                            continue

                        print(f"Mengambil: {provinsi} | {nilai_2023}")
                        extracted_data.append([provinsi, nilai_2023])
            else:
                print("Tabel tidak terdeteksi secara otomatis pada halaman ini.")
        else:
            print("Nomor halaman melebihi jumlah halaman PDF.")

    if extracted_data:
        df = pd.DataFrame(extracted_data, columns=["Provinsi", "2023"])
        df.to_csv(output_csv, index=False)
        print(f"\nSukses! {len(extracted_data)} baris data tersimpan di '{output_csv}'.")
    else:
        print("\nTidak ada data yang berhasil diekstrak. Cek kembali nomor halaman.")

except Exception as e:
    print(f"Terjadi Error: {e}")
```

Pada proses ini membuka file dari pdf menggunakan pdfplumber lalu mengekstrak tabel teks atau data. Kemudian mengubah hasil ekstraksi menjadi DataFrame pandas habis itu menyimpan data provinsi dan nilai ipm 2023 ke dalam csv. Hasilnya pada output ini :

```
Mengekstrak Data Tabel dari halaman file PDF ke-364...
Mengambil: (1) | (4)
Mengambil: Aceh | 74,70
Mengambil: Sumatera Utara | 75,13
Mengambil: Sumatera Barat | 75,64
Mengambil: Riau | 74,95
Mengambil: Jambi | 73,73
Mengambil: Sumatera Selatan | 73,18
Mengambil: Bengkulu | 74,30
Mengambil: Lampung | 72,48
Mengambil: Kepulauan Bangka Belitung | 74,09
Mengambil: Kepulauan Riau | 79,08
Mengambil: DKI Jakarta | 83,55
Mengambil: Jawa Barat | 74,24
Mengambil: Jawa Tengah | 73,39
Mengambil: D.I. Yogyakarta | 81,09
Mengambil: Jawa Timur | 74,65
Mengambil: Banten | 75,77
Mengambil: Bali | 78,01
Mengambil: Nusa Tenggara Barat | 72,37
Mengambil: Nusa Tenggara Timur | 68,40
Mengambil: Kalimantan Barat | 70,47
Mengambil: w
/
Kalimantan Tengah | 73,73
...
Mengambil: Papua Pegunungan | 53,45
Mengambil: Indonesia | 74,39

Sukses! 40 baris data tersimpan di 'ipm sudah ekstrak.csv'.
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

3. Scrap pdf

```python
import pandas as pd
import re

input_csv = "ipm sudah ekstrak.csv"
output_csv = "ipm sudah scrap.csv"

try:
    df = pd.read_csv(input_csv)

    def bersihkan_nama(text):
        if not isinstance(text, str):
            return text

        text = text.replace('\n', ' ')
        match = re.search(r'([A-Z].*)', text)

        if match:
            clean_text = match.group(1)
            clean_text = re.sub(r'[^\w\s.]', '', clean_text)

            return clean_text.strip()
        else:
            return text

    nama_kolom_provinsi = df.columns[0]
    print("Memproses pembersihan data...")
    df[nama_kolom_provinsi] = df[nama_kolom_provinsi].apply(bersihkan_nama)
    df = df[df[nama_kolom_provinsi].str.contains(r'[a-zA-Z]', na=False)]

    kolom_angka = df.columns[1]
    if df[kolom_angka].dtype == object:
        df[kolom_angka] = df[kolom_angka].astype(str).str.replace(',', '.', regex=False)

    df.to_csv(output_csv, index=False)
    print("-" * 30)
    print(f"Data berhasil dibersihkan! Disimpan ke: {output_csv}")
    print("-" * 30)

    print(df.head(50))

except Exception as e:
    print(f"Terjadi Error: {e}")
```

Prosesnya : membaca csv dari hasil ekstraksi ipm lalu membersihkan data seperti menghapus karakter aneh tidak perlu (seperti \n, simbol), mengubah koma menjadi titik untuk memperbaiki format angka, menghapus simbol non-digit dengan regex (re) setelah itu menyimpan hasilnya sebagai csv bersih. Hasilnya outputnya:

```
Memproses pembersihan data...
------------------------------
Data berhasil dibersihkan! Disimpan ke: ipm sudah scrap.csv
------------------------------
                     Provinsi   2023
1                        Aceh  74.70
2              Sumatera Utara  75.13
3              Sumatera Barat  75.64
4                        Riau  74.95
5                       Jambi  73.73
6             Sumatera Selatan  73.18
7                    Bengkulu  74.30
8                     Lampung  72.48
9    Kepulauan Bangka Belitung  74.09
10             Kepulauan Riau  79.08
11                DKI Jakarta  83.55
12                Jawa Barat  74.24
13                Jawa Tengah  73.39
14           D.I. Yogyakarta  81.09
15                Jawa Timur  74.65
16                    Banten  75.77
17                      Bali  78.01
18       Nusa Tenggara Barat  72.37
19       Nusa Tenggara Timur  68.40
20          Kalimantan Barat  70.47
...
36              Papua Selatan  68.24
37              Papua Tengah  59.44
38           Papua Pegunungan  53.45
39                  Indonesia  74.39
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

4. Preprocessing IPM

```python
import pandas as pd

# load data
df = pd.read_csv("ipm sudah scrap.csv")

# rename provinsi sesuai aturan
df["Provinsi"] = df["Provinsi"].str.upper()

df["Provinsi"] = df["Provinsi"].replace({
    "D.I. YOGYAKARTA": "DI YOGYAKARTA",
    "KEPULAUAN RIAU": "KEP. RIAU",
    "KEPULAUAN BANGKA BELITUNG": "KEP. BANGKA BELITUNG"
})

# daftar provinsi yang harus dihapus
hapus_prov = [
    "INDONESIA",
    "PAPUA TENGAH",
    "PAPUA SELATAN",
    "PAPUA PEGUNUNGAN",
    "PAPUA BARAT DAYA"
]

df = df[~df["Provinsi"].isin(hapus_prov)]

# ubah nama kolom 2023 menjadi IPM_2023
df = df.rename(columns={"2023": "IPM_2023"})

# simpan output
df.to_csv("IPM sudah cleaning.csv", index=False)

print("Selesai! File tersimpan sebagai 'IPM sudah cleaning.csv'")
```

Proses : mengubah nama provinsi menjadi huruf kapital melakukan cleaning tambahan yaitu menghapus baris kosong, memperbaiki nama kolom, mengkonversi angka teks menjadi numerik. Mengganti nama kolom 2023 menjadi IPM_2023 setelah itu menyimpan file ke IPM sudah cleaning.csv. Hasil outputnya :

```
Selesai! File tersimpan sebagai 'IPM sudah cleaning.csv'
```

5. Untuk sumber data ke 2

```python
import pandas as pd
from google.colab import files
uploaded = files.upload()

#Baca file melewati header non-data
df = pd.read_csv('data 2 tingkat pengangguran.csv', skiprows=3)

#Atur nama kolom biar rapi
df.columns = ['Provinsi', 'Februari_2023', 'Agustus_2023', 'Tahunan']

#Bersihkan nama provinsi dari spasi yg berlebih
df['Provinsi'] = df['Provinsi'].str.strip()

#Pastikan semua provinsi tetap ada, termasuk ACEH dan yang datanya kosong
df = df[df['Provinsi'].notna()]

#Hitung rata-rata tahunan dari Februari dan Agustus
def hitung_rata_rata(row):
    try:
        feb = float(row['Februari_2023'])
        agu = float(row['Agustus_2023'])
        return round((feb + agu) / 2, 2)
    except:
        return None

df['Rata_rata_Tahunan'] = df.apply(hitung_rata_rata, axis=1)

#Simpan ke CSV baru hanya dengan kolom Provinsi dan Rata-rata
output_df = df[['Provinsi', 'Rata_rata_Tahunan']]
output_filename = 'tingkat pengangguran sudah pre processing.csv'
output_df.to_csv(output_filename, index=False)

#Unduh file hasil dan auto download
files.download(output_filename)
```

Proses : membaca file csv dan melewati baris header kemudian menghitung rata-rata pengangguran dari bulan februari dan agustus 2023 setelah itu menyimpan file ke tingkat pengangguran sudah pre processing.csv. Hasil outputnya:

```
Saving data 2 tingkat pengangguran.csv to data 2 tingkat pengangguran.csv
```

6. Preprocessing Tingkat Pengangguran dan rename nama

```python
import pandas as pd

# Load data
df = pd.read_csv("tingkat pengangguran sudah pre processing.csv")

# daftar provinsi yang ingin dihapus
hapus_prov = ["PAPUA BARAT DAYA", "PAPUA SELATAN", "PAPUA TENGAH"]

# cleaning
df_clean = df[
    (df["Provinsi"].str.upper() != "INDONESIA") &
    (~df["Provinsi"].str.upper().isin(hapus_prov))
].dropna()

# save output
df_clean.to_csv("tingkat pengangguran sudah pre processing part 2.csv", index=False)

print("Cleaning selesai, file tersimpan.")
```

```python
import pandas as pd

# load data
df = pd.read_csv("tingkat pengangguran sudah pre processing part 2.csv")

# rename kolom
df = df.rename(columns={"Rata_rata_Tahunan": "Tingkat_Pengangguran_2023"})

# simpan output
df.to_csv("tingkat pengangguran sudah cleaning.csv", index=False)

print("Selesai! File tersimpan sebagai 'tingkat pengangguran sudah cleaning.csv'")
```

Proses : Menghapus provinsi tertentu yang tidak diperlukan lalu menggantikan nama kolom menjadi Tingkat_Pengangguran_2023 yang menyimpan ke tingkat pengangguran sudah cleaning.csv. Hasil Output:

```
Cleaning selesai, file tersimpan.
```

```
Selesai! File tersimpan sebagai 'tingkat pengangguran sudah cleaning.csv'
```

7. Sumber data ke 3

```python
# Unggah file CSV dari bps
from google.colab import files
uploaded = files.upload()

# Baca file tanpa menghilangkan baris ACEH
df = pd.read_csv('data 3 tindak pidana.csv', skiprows=3, header=None)

# Atur nama kolom secara manual
df.columns = ['Provinsi', 'Jumlah_Tindak_Pidana_2023']

# Bersihkan nama provinsi dari spasi ekstra
df['Provinsi'] = df['Provinsi'].astype(str).str.strip()

# Pastikan semua provinsi tetap ada, termasuk ACEH
df = df[df['Provinsi'].notna() & (df['Provinsi'] != '')]

# Simpan ke CSV baru
output_filename = 'tindak pidana sudah scrap.csv'
df.to_csv(output_filename, index=False)

# Unduh file hasil
files.download(output_filename)
```

Proses : load csv tindak pidana dari bps dan merapikan barisnya. Hasil outputnya:

```
Saving data 3 tindak pidana.csv to data 3 tindak pidana.csv
```

```python
import pandas as pd

# load data
df = pd.read_csv("tindak pidana sudah scrap.csv")

# ubah METRO JAYA menjadi DKI JAKARTA
df["Provinsi"] = df["Provinsi"].replace("METRO JAYA", "DKI JAKARTA")

# hapus Indonesia
df = df[df["Provinsi"].str.upper() != "INDONESIA"]

# format ribuan pada kolom yang benar
df["Jumlah_Tindak_Pidana_2023"] = (
    df["Jumlah_Tindak_Pidana_2023"]
    .astype(str)
    .str.replace(r"[^0-9]", "", regex=True)
    .astype(int)
    .map(lambda x: f"{x:,}".replace(",", "."))
)

# simpan output
df.to_csv("tindak pidana sudah cleaning.csv", index=False)

print("Selesai! File tersimpan sebagai 'tindak pidana sudah cleaning.csv'")
```

Proses : hapus kata indonesia dan beri aturan penulisan ribuan lalu ganti dengan metro jaya menjadi DKI JAKARTA. Hasil outputnya :

```
Selesai! File tersimpan sebagai 'tindak pidana sudah cleaning.csv'
```

8. Sumber data ke 4

```python
# uload CSV open data jabar
from google.colab import files
uploaded = files.upload()

#Baca file CSV
import pandas as pd

filename = list(uploaded.keys())[0]  # otomatis ambil nama file yang di upload
df = pd.read_csv('data 4 penduduk miskin.csv')

#Filter data untuk tahun 2023
df_2023 = df[df['tahun'] == 2023].reset_index(drop=True)

#Simpan hasil ke file baru
output_filename = "penduduk miskin sudah scrap.csv"
df_2023.to_csv(output_filename, index=False)

#Unduh file hasil
files.download(output_filename)
```

```python
import pandas as pd

# load data
df = pd.read_csv("penduduk miskin sudah scrap.csv")

# daftar provinsi yang ingin dihapus
hapus_prov = [
    "PAPUA TENGAH",
    "PAPUA BARAT DAYA",
    "PAPUA SELATAN",
    "PAPUA PEGUNUNGAN"
]

# hapus provinsi miss value
df = df[~df["nama_provinsi"].str.upper().isin(hapus_prov)]

# buat kolom penduduk_miskin dengan persen
df["penduduk_miskin"] = df["persentase_penduduk_miskin"].astype(str) + "%"

# hapus kolom yang diminta
df = df.drop(columns=["id", "tahun", "kode_provinsi", "persentase_penduduk_miskin", "satuan"])

# simpan output
df.to_csv("penduduk miskin pre processing.csv", index=False)

print("Cleaning selesai, file tersimpan.")
```

```python
import pandas as pd

# load data
df = pd.read_csv("penduduk miskin pre processing.csv")

# rename kolom
df = df.rename(columns={"penduduk_miskin": "Penduduk_Miskin_2023"})

# simpan output
df.to_csv("penduduk miskin pre processing part 2.csv", index=False)

print("Selesai! File tersimpan sebagai 'penduduk miskin pre processing part 2.csv'")
```

```python
import pandas as pd

# load data
df = pd.read_csv("penduduk miskin pre processing part 2.csv")

# Rename 'nama_provinsi' to 'Provinsi' for consistency
df = df.rename(columns={'nama_provinsi': 'Provinsi'})

# ubah nama provinsi
df["Provinsi"] = df["Provinsi"].str.replace("KEPULAUAN RIAU", "KEP. RIAU", regex=False)
df["Provinsi"] = df["Provinsi"].str.replace("KEPULAUAN BANGKA BELITUNG", "KEP. BANGKA BELITUNG", regex=False)

# simpan output
df.to_csv("penduduk miskin sudah cleaning.csv", index=False)

print("Selesai! File tersimpan sebagai 'penduduk miskin sudah cleaning.csv'")
```

Proses : load csv penduduk miskin dari open data jabar lalu mengambil hanya pada tahun 2023 kemudian merubah satuan persen menjadi satuan dengan angka dan menghapus provinsi menggunakan miss value. Ubah kolom b1 penduduk_miskin menjadi Penduduk_Miskin_2023. Setelah itu menyimpan file ke penduduk miskin sudah cleaning.csv. Hasil outputnya:

```
Saving data 4 penduduk miskin.csv to data 4 penduduk miskin.csv
```

```
Cleaning selesai, file tersimpan.
```

```
Selesai! File tersimpan sebagai 'penduduk miskin pre processing part 2.csv'
```

```
Selesai! File tersimpan sebagai 'penduduk miskin sudah cleaning.csv'
```

9. Integritas data

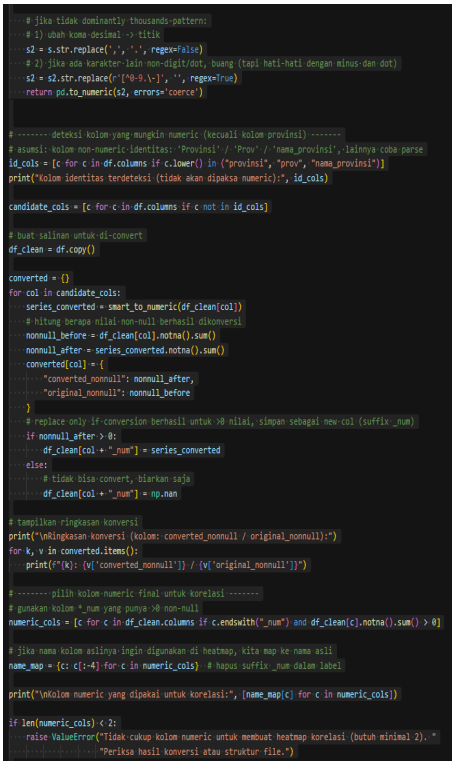
```python
import pandas as pd
from openpyxl import load_workbook
from openpyxl.styles import PatternFill, Font

# ====== LOAD DATA ======
df1 = pd.read_csv("penduduk miskin sudah cleaning.csv")
df2 = pd.read_csv("tindak pidana sudah cleaning.csv")
df3 = pd.read_csv("tingkat pengangguran sudah cleaning.csv")
df4 = pd.read_csv("IPM sudah cleaning.csv")

# Rename 'nama_provinsi' column in df1 to 'Provinsi' for consistent merging
df1 = df1.rename(columns={'nama_provinsi': 'Provinsi'})

# ====== GABUNG DATA ======
df = df1.merge(df2, on="Provinsi", how="outer") \
        .merge(df3, on="Provinsi", how="outer") \
        .merge(df4, on="Provinsi", how="outer")

# ====== SIMPAN KE EXCEL ======
output = "gabungan_belum_fix.xlsx"
df.to_excel(output, index=False)

# ====== FORMAT WARNA ======
wb = load_workbook(output)
ws = wb.active

# header styling
header_fill = PatternFill(start_color="4F81BD", end_color="4F81BD", fill_type="solid")
header_font = Font(color="FFFFFF", bold=True)

for cell in ws[1]:
    cell.fill = header_fill
    cell.font = header_font

# alternating row color
fill1 = PatternFill(start_color="DCE6F1", end_color="DCE6F1", fill_type="solid")

for row in ws.iter_rows(min_row=2):
    if row[0].row % 2 == 0:
        for cell in row:
            cell.fill = fill1

wb.save(output)

print("Integrasi selesai, file tersimpan:", output)
```

Proses : menggabungkan data berdasarkan kolom Provinsi lalu menyimpan hasil ke file excel gabungan_belum_fix.xlsx setelah itu Melakukan styling pada header dan baris untuk tampilan yang lebih baik dan ini hasil output :

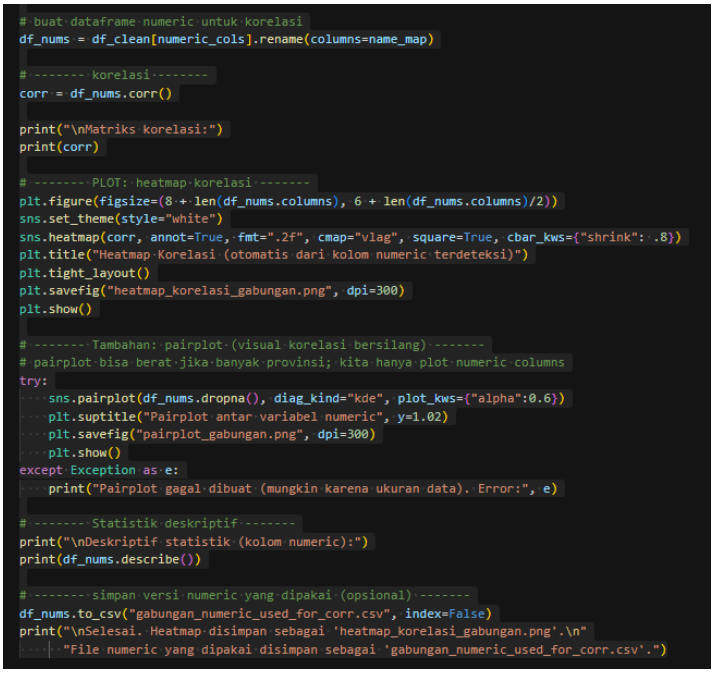
```
Integrasi selesai, file tersimpan: gabungan_belum_fix.xlsx
```

10. Analisis data

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re

# -------- path file (pakai path lokal yang ada) -------
file_path = "gabungan_belum_fix.xlsx" # Corrected file path

# -------- load data -------
df = pd.read_excel(file_path)

# tampilkan kolom awal untuk pengecekan cepat
print("Kolom dalam file:", list(df.columns))

# -------- helper: ubah series ke numeric dengan cleaning -------

def smart_to_numeric(s):
    """
    Bersihkan string-number menjadi numeric:
    - hilangkan '%' jika ada
    - deteksi apakah string menggunakan titik sebagai pemisah ribuan:
        pola seperti '11.916' atau '1.234.567' -> hilangkan semua titik
    - jika memakai koma sebagai desimal -> ubah ',' jadi '.'
    - fallback: coba to_numeric dengan errors='coerce'
    """
    s_orig = s.astype(str).str.strip()
    # kosongkan nilai yang benar-benar 'nan' str
    s_orig = s_orig.replace({'nan': np.nan, 'None': np.nan, 'none': np.nan})

    # hapus persen
    s = s_orig.str.replace('%', '', regex=False)

    # definisi fungsi pengecekan pola ribuan seperti 1.234 atau 12.345.678
    def is_thousands_pattern(x):
        if pd.isna(x):
            return False
        # hanya angka dan titik, tidak ada koma
        if ',' in x:
            return False
        return bool(re.match(r'^\d{1,3}(\.\d{3})+$', x))

    # cek apakah mayoritas non-null cocok pola ribuan
    non_null = s.dropna().head(500).astype(str)  # cek sample up to 500
    if len(non_null) > 0:
        matches = non_null.apply(is_thousands_pattern).sum()
        if matches / len(non_null) >= 0.6:  # ambang 60%
            # hapus semua titik (pemisah ribuan)
            s_clean = s.str.replace('.', '', regex=False)
            # setelah hapus titik, ganti koma jadi titik kalau ada
            s_clean = s_clean.str.replace(',', '.', regex=False)
            return pd.to_numeric(s_clean, errors='coerce')
```



```python
    # jika tidak dominantly thousands-pattern:
    # 1) ubah koma-desimal -> titik
    s2 = s.str.replace(',', '.', regex=False)
    # 2) jika ada karakter-lain non-digit/dot, buang (tapi hati-hati dengan minus dan dot)
    s2 = s2.str.replace(r'[^0-9.\-]', '', regex=True)
    return pd.to_numeric(s2, errors='coerce')


# -------- deteksi kolom yang mungkin numeric (kecuali kolom provinsi) -------
# asumsi: kolom non-numeric identitas: 'Provinsi' / 'Prov' / 'nama_provinsi', lainnya coba parse
id_cols = [c for c in df.columns if c.lower() in ("provinsi", "prov", "nama_provinsi")]
print("Kolom identitas terdeteksi (tidak akan dipaksa numeric):", id_cols)

candidate_cols = [c for c in df.columns if c not in id_cols]

# buat salinan untuk di-convert
df_clean = df.copy()

converted = {}
for col in candidate_cols:
    series_converted = smart_to_numeric(df_clean[col])
    # hitung berapa nilai non-null berhasil dikonversi
    nonnull_before = df_clean[col].notna().sum()
    nonnull_after = series_converted.notna().sum()
    converted[col] = {
        "converted_nonnull": nonnull_after,
        "original_nonnull": nonnull_before
    }
    # replace only if conversion berhasil untuk >0 nilai, simpan sebagai new-col (suffix _num)
    if nonnull_after > 0:
        df_clean[col + "_num"] = series_converted
    else:
        # tidak bisa convert, biarkan saja
        df_clean[col + "_num"] = np.nan

# tampilkan ringkasan konversi
print("\nRingkasan konversi (kolom: converted_nonnull / original_nonnull):")
for k, v in converted.items():
    print(f"{k}: {v['converted_nonnull']} / {v['original_nonnull']}")

# -------- pilih kolom-numeric final untuk korelasi -------
# gunakan kolom *_num yang punya >0 non-null
numeric_cols = [c for c in df_clean.columns if c.endswith("_num") and df_clean[c].notna().sum() > 0]

# jika nama kolom aslinya ingin digunakan di heatmap, kita map ke nama asli
name_map = {c: c[:-4] for c in numeric_cols}  # hapus suffix _num dalam label

print("\nKolom numeric yang dipakai untuk korelasi:", [name_map[c] for c in numeric_cols])

if len(numeric_cols) < 2:
    raise ValueError("Tidak cukup kolom numeric untuk membuat heatmap korelasi (butuh minimal 2). "
                     "Periksa hasil konversi atau struktur file.")
```



```python
# buat dataframe numeric untuk korelasi
df_nums = df_clean[numeric_cols].rename(columns=name_map)

# -------- korelasi -------
corr = df_nums.corr()

print("\nMatriks korelasi:")
print(corr)

# -------- PLOT: heatmap korelasi -------
plt.figure(figsize=(8 + len(df_nums.columns), 6 + len(df_nums.columns)/2))
sns.set_theme(style="white")
sns.heatmap(corr, annot=True, fmt=".2f", cmap="vlag", square=True, cbar_kws={"shrink": .8})
plt.title("Heatmap Korelasi (otomatis dari kolom numeric terdeteksi)")
plt.tight_layout()
plt.savefig("heatmap_korelasi_gabungan.png", dpi=300)
plt.show()

# -------- Tambahan: pairplot (visual korelasi bersilang) -------
# pairplot bisa berat jika banyak provinsi; kita hanya plot numeric columns
try:
    sns.pairplot(df_nums.dropna(), diag_kind="kde", plot_kws={"alpha":0.6})
    plt.suptitle("Pairplot antar variabel numeric", y=1.02)
    plt.savefig("pairplot_gabungan.png", dpi=300)
    plt.show()
except Exception as e:
    print("Pairplot gagal dibuat (mungkin karena ukuran data). Error:", e)

# -------- Statistik deskriptif -------
print("\nDeskriptif statistik (kolom numeric):")
print(df_nums.describe())

# -------- simpan versi numeric yang dipakai (opsional) -------
df_nums.to_csv("gabungan_numeric_used_for_corr.csv", index=False)
print("\nSelesai. Heatmap disimpan sebagai 'heatmap_korelasi_gabungan.png'.\n"
      "File numeric yang dipakai disimpan sebagai 'gabungan_numeric_used_for_corr.csv'.")
```

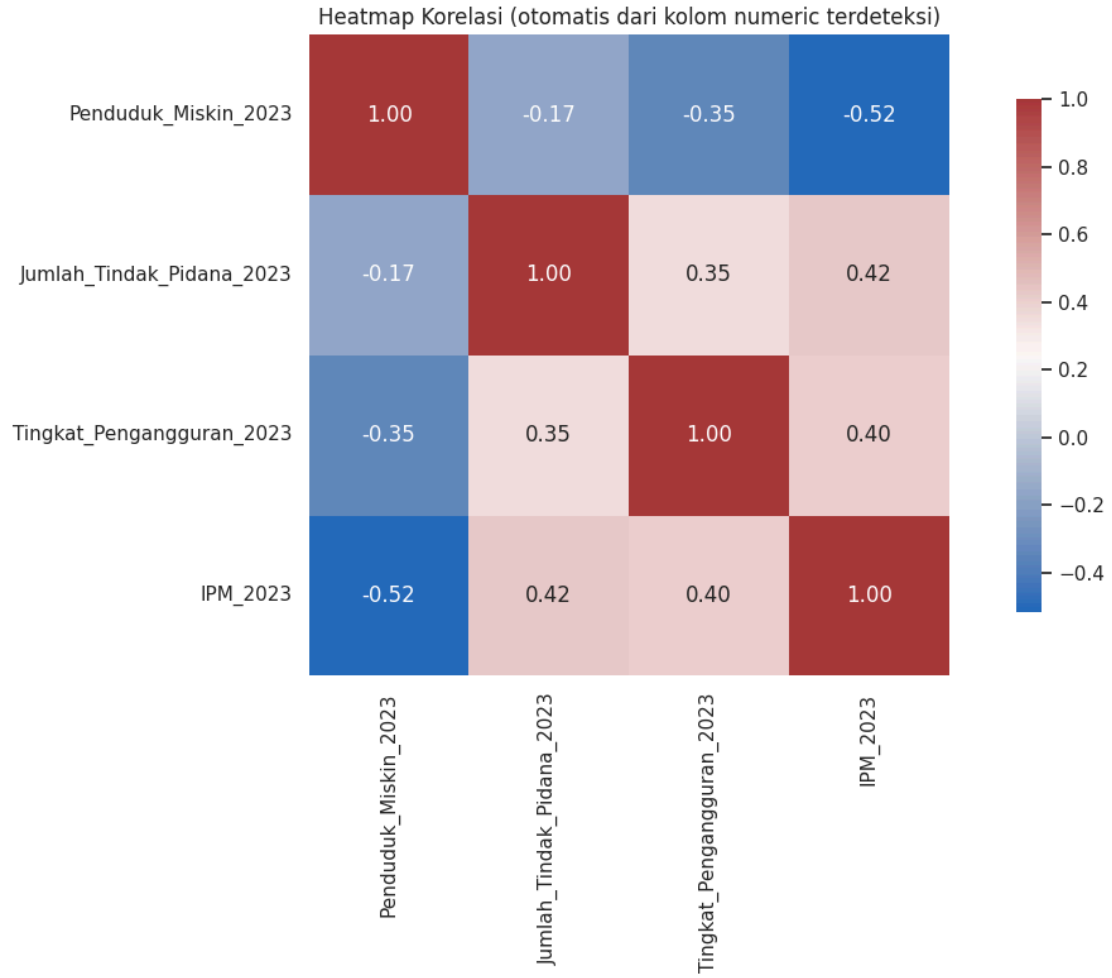
Proses:

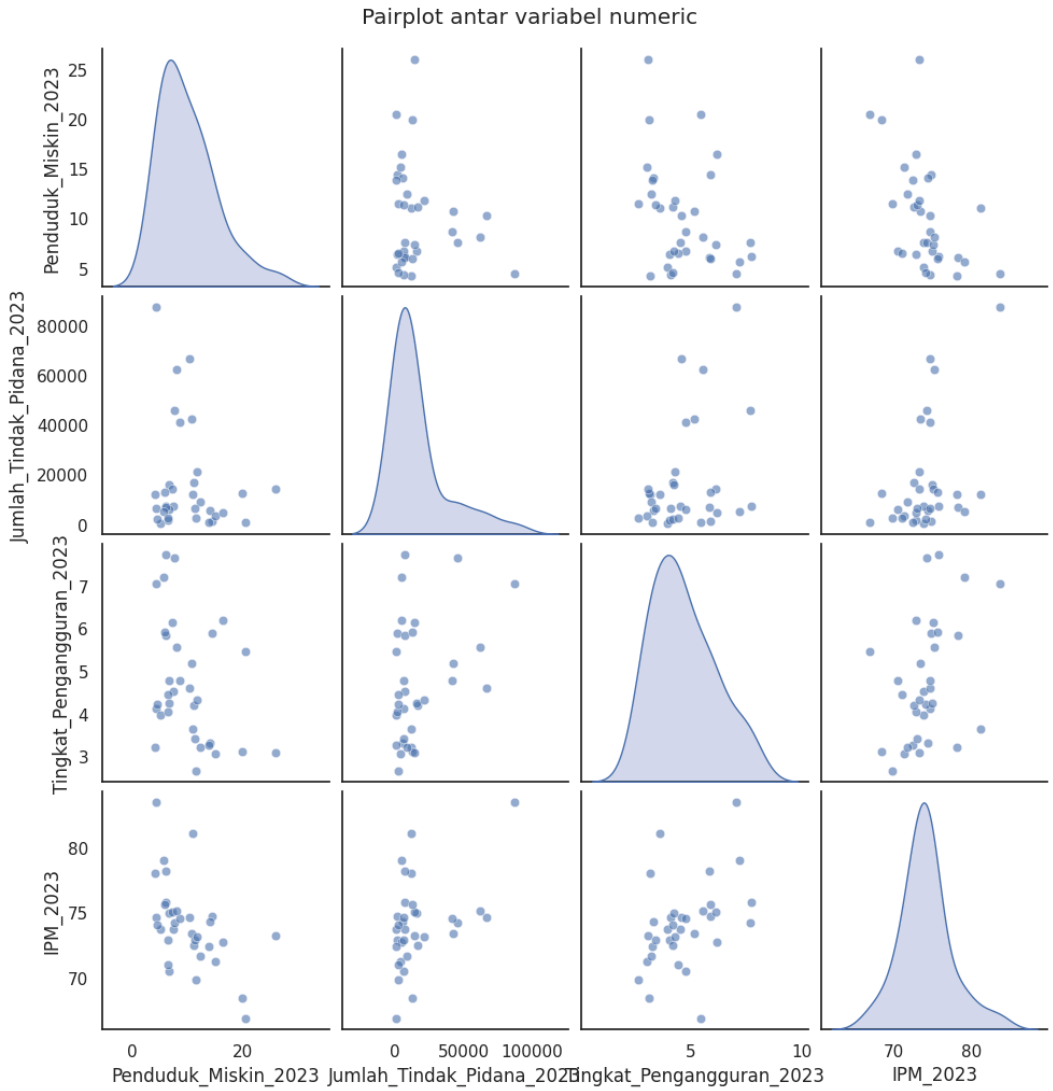
- Konversikan ke dalam Numerik untuk mengubah data string (dengan %, pemisah ribuan) menjadi numerik.
- Korelasikan dalam menghitung matriks korelasi antara variabel.
- Heatmap yaitu visualisasikan korelasi antar variabel.

- Pairplot untuk visualisasi distribusi dan hubungan antar variabel.
- Statistik Deskriptif menunjukkan tampilkan ringkasan statistik.

Hasil output:

```
Kolom dalam file: ['Provinsi', 'Penduduk_Miskin_2023', 'Jumlah_Tindak_Pidana_2023', 'Tingkat_Pengangguran_2023', 'IPM_2023']
Kolom identitas terdeteksi (tidak akan dipaksa numeric): ['Provinsi']

Ringkasan konversi (kolom: converted_nonnull / original_nonnull):
Penduduk_Miskin_2023: 34 / 34
Jumlah_Tindak_Pidana_2023: 34 / 34
Tingkat_Pengangguran_2023: 34 / 34
IPM_2023: 34 / 34

Kolom numeric yang dipakai untuk korelasi: ['Penduduk_Miskin_2023', 'Jumlah_Tindak_Pidana_2023', 'Tingkat_Pengangguran_2023', 'IPM_2023']

Matriks korelasi:
                        Penduduk_Miskin_2023  Jumlah_Tindak_Pidana_2023  \
Penduduk_Miskin_2023                1.000000                  -0.166412
Jumlah_Tindak_Pidana_2023          -0.166412                   1.000000
Tingkat_Pengangguran_2023          -0.346740                   0.354514
IPM_2023                           -0.515294                   0.420728

                        Tingkat_Pengangguran_2023  IPM_2023
Penduduk_Miskin_2023                    -0.346740 -0.515294
Jumlah_Tindak_Pidana_2023                0.354514  0.420728
Tingkat_Pengangguran_2023                1.000000  0.403562
IPM_2023                                 0.403562  1.000000
```






```
Deskriptif statistik (kolom numeric):
       Penduduk_Miskin_2023  Jumlah_Tindak_Pidana_2023  \
count             34.000000                  34.000000
mean              10.089118               16390.323529
std                5.183509               21231.863454
min                4.250000                 442.000000
25%                6.240000                3865.750000
50%                8.425000                7412.000000
75%               12.252500               15399.000000
max               26.030000               87426.000000

       Tingkat_Pengangguran_2023   IPM_2023
count                  34.000000  34.000000
mean                    4.710000  74.052353
std                     1.407609   3.291068
min                     2.660000  66.840000
25%                     3.465000  72.547500
50%                     4.390000  73.910000
75%                     5.770000  75.017500
max                     7.740000  83.550000

Selesai. Heatmap disimpan sebagai 'heatmap_korelasi_gabungan.png'.
File numeric yang dipakai disimpan sebagai 'gabungan_numeric_used_for_corr.csv'.
```

BAB V
PENUTUP

5.1 Kesimpulan

Setelah melewati dari seluruh proses mulai dari mengumpulkan data dengan cara web scraping, membersihkan data, mengubah bentuk data, menggabungkan data, hingga membuat visualisasi bisa dapat disimpulkan bahwa data mengenai kondisi sosial ekonomi dan kriminalitas yang dikumpulkan telah berhasil diubah menjadi dataset yang teratur. Dataset ini mampu mencerminkan situasi di setiap provinsi di Indonesia. Proses penyusunan data ini berhasil meningkatkan kualitas data secara signifikan sehingga hasil visualisasi dan analisis yang diperoleh lebih tepat dan mampu menunjukkan bahwa hubungan antara berbagai variabel seperti tingkat pengangguran, Indeks Pembangunan Manusia (IPM), tingkat kemiskinan, dan tingkat tindak kriminal. Hasil eksplorasi juga menunjukkan dengan adanya pola-pola yang bisa diamati lebih lanjut selama memastikan hasilnya baik untuk keperluan penelitian maupun pembuatan kebijakan yang didasarkan pada data.

5.2 Saran

- Penelitian berikutnya menggunakan rentang tahun yang lebih lama agar bisa melihat trend dengan jangka panjang dan mengurangi pengaruh bias dari data tahunan.
- Variabel sosial ekonomi bisa ditambahkan lebih banyak lagi seperti indeks ketimpangan, pendapatan rata-rata, atau tingkat pendidikan agar bisa menganalisisnya lebih lengkap dan mendalam.
- Proses pengumpulan data ini bisa diperbaiki dengan menggunakan automasi berkala sehingga data selalu diperbarui secara otomatis.
- Analisis bisa ditingkatkan dengan menggunakan metode statistik yang lebih canggih atau teknik machine learning untuk membuat prediksi atau memodelkan hubungan antar variabel.

DAFTAR PUSTAKA

Badan Pusat Statistik. 2024. *Indeks Pembangunan Manusia (IPM) Indonesia Tahun 2023*. Jakarta: BPS RI.

Kementerian Komunikasi dan Informatika. 2022. *Pemanfaatan Data dan Teknologi untuk Analisis Sosial Ekonomi di Indonesia*. Jakarta: Kementerian Kominfo RI.

Nugroho, Andi dan Dwi Setiawan. 2021. *Pengolahan Data Besar (Big Data) untuk Analisis Sosial di Indonesia*. Yogyakarta: Deepublish.

McKinney, Wes. 2018. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. Sebastopol, California: O'Reilly Media.

Mitchell, Ryan. 2015. *Web Scraping with Python: Collecting More Data from the Modern Web*. Sebastopol, California: O'Reilly Media.

LAMPIRAN

Lampiran 1. Data Hasil Wrangling

A	B	C	D	E
Provinsi	Penduduk_Miskin_2023	Jumlah_Tindak_Pidana_2023	Tingkat_Pengangguran_2023	IPM_2023
ACEH	14.45%	12.42	5.89	74.7
BALI	4.25%	11.916	3.21	78.01
BANTEN	6.17%	7.392	7.74	75.77
BENGKULU	14.04%	5.579	3.31	74.3
DI YOGYAKARTA	11.04%	12.061	3.63	81.09
DKI JAKARTA	4.44%	87.426	7.05	83.55
GORONTALO	15.15%	3.574	3.06	71.25
JAMBI	7.58%	7.432	4.52	73.73
JAWA BARAT	7.62%	45.694	7.67	74.24
JAWA TENGAH	10.77%	42.304	5.19	73.39
JAWA TIMUR	10.35%	66.741	4.61	74.65
KALIMANTAN BARAT	6.71%	6.028	4.79	70.47
KALIMANTAN SELATAN	4.29%	6.375	4.13	74.66
KALIMANTAN TENGAH	5.11%	4.42	3.97	73.73
KALIMANTAN TIMUR	6.11%	6.762	5.84	78.2
KALIMANTAN UTARA	6.45%	1.701	4.05	72.88
KEP. BANGKA BELITUNG	4.52%	2.211	4.22	74.09
KEP. RIAU	5.69%	5.074	7.21	79.08
LAMPUNG	11.11%	16.608	4.21	72.48
MALUKU	16.42%	4.741	6.2	72.75
MALUKU UTARA	6.46%	2.334	4.46	70.98
NUSA TENGGARA BARAT	13.85%	7.55	3.26	72.37
NUSA TENGGARA TIMUR	19.96%	12.692	3.12	68.4
PAPUA	26.03%	14.074	3.08	73.23
PAPUA BARAT	20.49%	6.41	5.46	66.84
RIAU	6.68%	15.777	4.24	74.95
SULAWESI BARAT	11.49%	2.679	2.66	69.8
SULAWESI SELATAN	8.7%	41.196	4.79	74.6
SULAWESI TENGAH	12.41%	8.944	3.22	71.66
SULAWESI TENGGARA	11.43%	6.276	3.41	72.94
SULAWESI UTARA	7.38%	14.265	6.14	75.04
SUMATERA BARAT	5.95%	12.722	5.92	75.64
SUMATERA SELATAN	11.78%	21.335	4.32	73.18
SUMATERA UTARA	8.15%	62.278	5.56	75.13

Lampiran 2. Dokumentasi Pipeline