LAPORAN
MATA KULIAH STATISTIKA MULTIVARIAT
KELAS A

"IMPLEMENTASI PRINCIPAL COMPONENT ANALYSIS PADA DATA METRO INTERSTATE TRAFFIC VOLUME"

DISUSUN OLEH KELOMPOK 6 :

1. CHRYSILLA CITRA WINDYADARI (21083010023)
2. ELLEXIA LEONIE GUNAWAN (21083010027)
3. ANGELA LISANTHONI (21083010032)

DOSEN PENGAMPU:

AVIOLLA TERZA DAMALIANA, S.Si., M.Stat

PRISMAHARDI AJI RIYANTOKO, S. Si, M.Si

PROGRAM STUDI SAINS DATA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

JAWA TIMUR

2023

DAFTAR ISI

BAB I: PENDAHULUAN

1.1 Latar Belakang

Kemacetan lalu lintas adalah salah satu permasalahan yang sering dihadapi oleh masyarakat perkotaan, khususnya di Indonesia. Salah satu faktor penyebab kemacetan adalah kepadatan penduduk yang semakin tinggi, yang berdampak pada meningkatnya permintaan akan transportasi. Indonesia merupakan negara dengan jumlah penduduk terbanyak keempat di dunia, dengan sekitar 274 juta jiwa pada tahun 2022 [1]. Jakarta sebagai ibu kota negara juga memiliki kepadatan penduduk tertinggi di Indonesia, yaitu sekitar 16 ribu jiwa per kilometer persegi. Hal ini menyebabkan Jakarta menjadi salah satu kota dengan tingkat kemacetan yang tinggi di dunia. Berdasarkan *TomTom Traffic Index 2022*, Jakarta berada pada peringkat 29 dari 389 kota di dunia dalam hal kemacetan lalu lintas [2].

Kemacetan lalu lintas dapat menimbulkan banyak dampak negatif, baik untuk kesehatan manusia, lingkungan, maupun perekonomian. Dengan adanya kemacetan, maka konsumsi bahan bakar dan emisi gas buang akan meningkat. Hal ini dapat mencemari udara, menyebabkan pemanasan global, mengganggu pernafasan, dan mengganggu sistem peredaran darah. Selain itu, dengan kemacetan juga dapat menghambat perekonomian. Kemacetan dapat mengakibatkan keterlambatan dan kerugian untuk distribusi barang dan jasa. Hal ini berarti biaya operasional perusahaan menjadi lebih tinggi, dan profit margin menjadi lebih rendah. Dampak lainnya adalah menurunnya produktivitas pekerja. Kemacetan membuat pekerja harus menghabiskan lebih banyak waktu di jalan, sehingga mengurangi waktu kerja dan istirahat. Hal ini dapat menimbulkan stres, kelelahan, dan penurunan kinerja. Kemacetan juga dapat mengurangi mobilitas dan aksesibilitas pekerja, sehingga membatasi peluang kerja dan pendapatan.

Kemacetan lalu lintas menjadi salah satu masalah yang harus diatasi. Salah satu cara awal yang dapat dilakukan yaitu dengan mencari tahu faktor-faktor apa saja yang memiliki hubungan terhadap penyebab kemacetan lalu lintas. Untuk mengetahui faktor-faktor yang memiliki hubungan dengan penyebab kemacetan, kita dapat menggunakan *Principal Component Analysis.*

Principal Component Analysis (PCA) adalah metode yang digunakan untuk mengurangi dimensi data besar dengan mentransformasikan kumpulan variabel menjadi lebih kecil namun masih mengandung sebagian besar informasi dalam data. PCA akan mengidentifikasi seperangkat sumbu ortogonal, disebut komponen utama, yang

menangkap varians maksimum dalam data. Komponen utama adalah kombinasi linear dari variabel asli dalam kumpulan data dan diurutkan menurut urutan pentingnya [3]. Sehingga diharapkan dengan metode ini, dapat mereduksi dimensi data kemacetan agar interpretasi data lebih cepat dan dapat mengetahui faktor-faktor yang memiliki hubungan dengan penyebab kemacetan.

1.2 Permasalahan

Rumusan permasalahan dari penelitian ini adalah sebagai berikut:

1. Bagaimana cara menghitung *Principal Component Analysis* (PCA) pada data kemacetan?
2. Ada berapa faktor yang memiliki hubungan dengan penyebab kemacetan?

1.3 Tujuan

1. Untuk mengetahui cara menghitung *Principal Component Analysis* (PCA) pada data kemacetan.
2. Untuk mengetahui jumlah faktor yang memiliki hubungan dengan penyebab kemacetan.

1.4 Manfaat

1. Bagi Penulis

   Mengembangkan pengetahuan penulis mengenai penerapan perhitungan *Principal Component Analysis* (PCA) pada data.
2. Bagi Pengembangan Ilmu Pengetahuan
   a. Menambah wawasan terkait proses perhitungan *Principal Component Analysis* (PCA) pada data kemacetan
   b. Mempermudah untuk penelitian berikutnya dari data yang sudah direduksi
3. Bagi Pemerintah

   Dapat mengambil kebijakan-kebijakan lalu lintas yang lebih tepat

BAB II: TINJAUAN PUSTAKA

2.1 Teori Penunjang

2.1.1 *Principal Component Analysis* (PCA)

Principal Component Analysis (PCA) merupakan metode yang digunakan untuk mengurangi atau mereduksi dimensi dari dataset yang memiliki banyak variabel (dimensi) sehingga data yang diinterpretasikan hanya memuat informasi penting untuk kebutuhan analisis. PCA akan mempertahankan sebanyak mungkin 'variabilitas' yang berarti menemukan variabel baru yang merupakan fungsi linier dari dataset asli [4]. PCA akan memaksimalkan varians dan secara matematis PCA akan mentransformasikan variabel yang berkorelasi ke dalam bentuk yang tidak berkorelasi satu sama lain.

Secara umum perhitungan model PCA sebagai berikut [5]:

1. Data terlebih dahulu ditransformasikan ke dalam bentuk baku **Z** (*standardized*). Transformasi ini dilakukan terhadap data yang satuan pengamatannya tidak sama.
2. Menghitung matriks kovarian dari data yang telah distandarisasi dengan menggunakan $Cov(X_j, X_k) = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ji} - \bar{X}_j)(X_{ki} - \bar{X}_k)$.
3. Menghitung nilai eigen dari data yang telah distandarisasi dan nilai eigen diurutkan dari terbesar ke terkecil.
   $det(\mathbf{Z} - \lambda\mathbf{I}) = 0$
4. Menghitung vektor eigen dari nilai eigen yang telah didapat sebelumnya.
   $[\mathbf{Z} - \lambda\mathbf{I}][X] = 0$
5. Masukkan ke persamaan model PCA berikut

$$Y_1 = e_1'X = e_{11}'X_1 + e_{21}'X_2 + \cdots + e_{p1}'X_p$$
$$Y_2 = e_2'X = e_{12}'X_1 + e_{22}'X_2 + \cdots + e_{p2}'X_p$$
$$\cdots$$
$$Y_p = e_p'X = e_{1p}'X_1 + e_{2p}'X_2 + \cdots + e_{pi}'X_p \quad [6]$$

   p adalah banyaknya variabel asal dan e_{pi}' adalah eigen vektor ke-i pada variabel ke-p
6. Menentukan variabel baru (komponen utama)

   Banyaknya komponen utama (PC) yang terbentuk sama dengan banyaknya variabel asli. Untuk menentukan jumlah komponen utama yang akan digunakan terdapat tiga cara berikut :

a. Melihat total variansi data yang dapat diperoleh dengan rumus berikut [5].

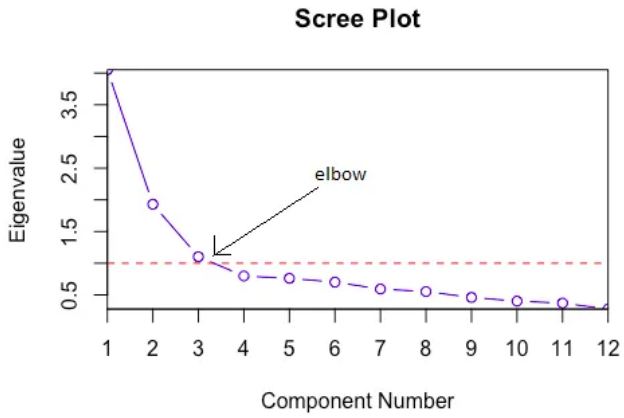
$$\rho I = \frac{\lambda_p}{\sum_{i=1}^{D} \lambda_i} \times 100\%$$

Jumlah komponen utama yang akan digunakan berdasarkan kriteria persentase keragaman data (variansi) yang diterangkan. Kriteria persentase (threshold) disesuaikan dengan kebutuhan analisis, beberapa persentase yang digunakan penganalisis meliputi kriteria persentase $>$70%, $>$75%, $>$ 80%, $>$ 90%, dan $>$ 95%.

b. Melihat nilai eigen (λ_p) yang lebih dari satu

c. Mengamati scree plot dengan melihat patahan siku dari scree plot

Untuk menentukan jumlah komponen utama dapat menggunakan representasi grafis yang dikenal sebagai scree plot. Scree Plot adalah plot segmen garis sederhana yang menunjukkan nilai eigen untuk setiap komponen utama. Sumbu x yang menunjukkan jumlah komponen utama dan sumbu y menunjukkan nilai eigen dari data.



Gambar 1. Scree Plot

Sebagian besar scree plot terlihat sangat mirip bentuknya, mulai dari tinggi di sebelah kiri, jatuh agak cepat, dan kemudian mendatar di beberapa titik. Hal ini karena komponen pertama biasanya menjelaskan sebagian besar variabilitas, beberapa komponen berikutnya menjelaskan jumlah sedang, dan komponen terakhir hanya menjelaskan sebagian kecil dari keseluruhan variabilitas. Untuk kriteria scree plot sendiri adalah mencari "siku" di kurva dan memilih semua komponen sebelum garis rata [7].

2.1.2 Uji Asumsi

Sebelum melakukan pereduksian data terdapat beberapa uji asumsi yang harus dipenuhi. Uji asumsi ini meliputi uji asumsi korelasi antara variabel menggunakan Uji Barlett dan identifikasi kecukupan sampel data menggunakan KMO.

2.1.2.1 Uji asumsi korelasi antara variabel (Multikolinieritas)

Uji asumsi klasik kebebasan antar variabel dapat menggunakan uji Bartlett. Uji Bartlett bertujuan untuk mengetahui ada tidaknya hubungan (korelasi) yang signifikan antar variabel bebas dalam kasus multivariat. Jika variabel $X_1, X_2, \dots, X_p$ merupakan independent (bersifat saling bebas), maka matriks korelasi antar variabel sama dengan matriks identitas (I). Sehingga untuk menguji asumsi ini, uji Bartlett menyatakan hipotesis berikut:

H_0 : $\mathbf{R} = \mathbf{I}$ (tidak ada korelasi antar variabel)

H_1 : $\mathbf{R} \neq \mathbf{I}$ (terdapat korelasi antar variabel**)**

Statistik uji [8] :

$\bar{r_k} = \frac{1}{p-1}\sum_{i=1}^{p} r_{ik}\,, \quad k = 1,2,\dots,p$

$\bar{r} = \frac{2}{p(p-1)}\sum\sum_{i<k} r_{ik}$

$\hat{\gamma} = \frac{(p-1)^2[1-(1-\bar{r})^2]}{p-(p-2)(1-\bar{r})^2}$

Dimana :

$\bar{r_k} =$ rata-rata elemen diagonal pada kolom atau baris ke *k* dari matrik R (matrik korelasi)

$\bar{r} =$ rata-rata keseluruhan dari elemen diagonal

Daerah penolakan : tolak H_0 jika $T = \frac{(n-1)}{(1-\bar{r})^2}\left[\sum\sum_{i<k}(r_{ik}-\bar{r})^2 - \hat{\gamma}\sum_{k=1}^{p}(\bar{r_k}-\bar{r})^2\right] > \chi^2_{(p+1)_{(p-2)/2;\alpha}}$ yang berarti analisis multivariat layak untuk digunakan analisis komponen utama (PCA).

2.1.2.2 Identifikasi kecukupan sampel data

Identifikasi kecukupan sampel data dapat diuji menggunakan nilai *Measure of Sampling Adequacy* (MSA) dan Kaiser-Meyer-Olkin (KMO) [9]. Sekelompok data dikatakan memenuhi syarat uji asumsi klasik ini apabila nilai dari MSA dan KMO lebih besar daripada 0,5 (Widarjono, 2010). Adapun hipotesis yang diujikan sebagai berikut.

H_0 : Jumlah data cukup untuk dilakukan analisis PCA

H_1 : Jumlah data tidak cukup untuk dilakukan analisis PCA

Statistik uji :

$$KMO = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p} r_{ij}^2}{\sum_{i=1}^{p}\sum_{j=1}^{p} r_{ij}^2 + \sum_{i=1}^{p}\sum_{j=1}^{p} a_{ij}^2}$$

$i = 1,2,3, \dots, p$ dan $j = 1,2, \dots, p$, untuk $i \neq j$

Dimana :

$r_{ij} =$ Koefisien korelasi antara variabel i dan j

$a_{ij} =$ Koefisien korelasi parsial antara variabel i dan j

Daerah kritis : gagal tolak H_0apabila nilai KMO lebih besar dari 0,5 sehingga dapat disimpulkan jumlah data telah cukup untuk dilakukan analisis PCA.

Selain nilai KMO juga terdapat nilai MSA yang menunjukkan jumlah kecukupan sampel data untuk tiap variabel. Dengan hipotesis yang diujikan meliputi.

H_0 : Variabel belum memadai untuk dianalisis lebih lanjut

H_1 : Variabel telah memadai untuk dianalisis lebih lanjut

Statistik uji :

$$MSA_i = \frac{\sum_{i=1}^{p} r_{ij}^2}{\sum_{i=1}^{p} r_{ij}^2 + \sum_{i=1}^{p} a_{ij}^2}, \text{ untuk } i \neq j$$

Dimana :

$r_{ij} =$ Koefisien korelasi antara variabel i dan j

$a_{ij} =$ Koefisien korelasi parsial antara variabel i dan j

Daerah kritis : tolak H_0 apabila nilai MSA_i atau diagonal Anti Image Correlation > 0.5.

2.2 Penelitian Terkait

2.2.1 Penelitian 1

(Rozy dkk., 2022) melakukan penelitian untuk menganalisis faktor biomotor yang mempengaruhi performa atlet atletik Sumatra Utara pada pelatihan daerah jangka panjang untuk PON 2024. Penelitian ini menggunakan *Principal Component Analysis* (PCA) dengan software Rstudio. Data dikumpulkan dari atlet nomor lari jarak jauh dan sprint yang berjumlah 18 orang dengan rentan usia 17 – 25 tahun. Data terkumpul sebanyak 12 variabel yang meliputi usia, tinggi badan, berat badan, IMT, kelentukan V sit and reach, kelentukan shoulder & wrist, daya tahan otot perut, daya tahan otot lengan, daya ledak otot tungkai, kecepatan, core test, dan daya tahan aerobik (VO2 Max). Hasil cumulative variant maupun scree plot dan perhitungan

eigen vector didapatkan 4 komponen utama yang mempengaruhi performa atlet meliputi:

1. Komponen utama 1 yang berkontribusi adalah daya ledak otot tungkai, daya ledak otot lengan, tinggi badan
2. Komponen utama 2 yang berkontribusi adalah IMT dan berat badan
3. Komponen utama 3 yang berkontribusi adalah kelentukan
4. Komponen utama 4 yang berkontribusi adalah core tes atau kekuatan

2.2.2 Penelitian 2

(Wangge, 2021) melakukan penelitian untuk mengetahui faktor-faktor dominan yang mempengaruhi penyelesaian skripsi mahasiswa Program Studi Pendidikan Matematika Undana. Data dikumpulkan dengan menyebarkan angket kepada alumni mahasiswa Jurusan/Prodi Pendidikan Matematika berjumlah 50 orang dan dari laboratorium berupa data wisudawan seperti data tanggal penerimaan judul skripsi mahasiswa, data jadwal ujian proposal dan data jadwal ujian skripsi mahasiswa Pendidikan Matematika serta melakukan wawancara pada bagian administrasi jurusan. Data ini kemudian diproses pereduksian dimensi dengan menggunakan metode PCA agar mendapatkan faktor-faktor dominan dari permasalahan. Dengan bantuan software SPSS, dataset awal yang memiliki 12 variabel tereduksi menjadi 10 variabel serta 4 komponen utama yang terbentuk meliputi

1. Komponen utama 1 adalah faktor pendukung penulisan skripsi dengan variabel yang berkontribusi meliputi kualitas bimbingan skripsi, ketersediaan sumber belajar, gaya bimbingan dosen, sikap dan interaksi dengan dosen pembimbing, lingkungan teman sebaya, dan perhatian orang tua.
2. Komponen utama 2 adalah faktor motivasi lulus tepat waktu
3. Komponen utama 3 adalah faktor membagi waktu
4. Komponen utama 4 adalah faktor kemampuan menulis dengan variabel yang berkontribusi meliputi kegiatan kemahasiswaan dan kemampuan menulis karya ilmiah.

2.2.3 Penelitian 3

Penelitian (Masithoh dkk., 2022) ini bertujuan untuk mengetahui unsur cuaca yang paling mempengaruhi hasil produksi bidang pertanian di daerah Malang. Penelitian menggunakan data unsur-unsur cuaca pada periode 2018-2021 yang diambil dari BPS. Dengan metode PCA untuk mendapatkan unsur cuaca yang paling utama

mempengaruhi hasil produksi dan software SPSS terbentuk 2 komponen utama dari 7 variabel sebagai berikut.

1. Komponen utama 1 meliputi kelembaban udara, curah hujan, arah angin, penyinaran matahari, dan tekanan udara dengan variabel yang paling dominan berpengaruh adalah curah hujan sebesar 0.931
2. Komponen utama 2 meliputi suhu udara dan kecepatan angin dengan variabel yang paling dominan adalah kecepatan angin sebesar 0.833

2.2.4 Penelitian 4

(Ritonga & Muhandhis., 2021) melakukan penelitian yang bertujuan untuk mengetahui tingkat kepuasan wisatawan terhadap destinasi wisata dengan menggabungkan metode *Principal Component Analysis* (PCA) dengan algoritma klasifikasi *Support Vector Machine* (SVM), *Artificial Neural Networks* (ANN), dan *Decision Trees*. Dataset yang digunakan diperoleh dari website UCI (*University of California*, Irvine) yang terdiri dari 11 variabel. *Principal component* yang digunakan adalah yang memenuhi 95% dari total variance data. Hasilnya dataset yang terlebih dahulu dilakukan analisis PCA mendapatkan akurasi terbesar dibandingkan jika tidak mereduksi dataset. Adapun rincian hasil akurasi sebelum dan sesudah dilakukan PCA sebagai berikut.

Tabel 1. Perbandingan Persentase Akurasi Model

Akurasi Sebelum PCA (%)	Akurasi Setelah PCA (%)
SVM 87,76	SVM-PCA 91,50
ANN 88,44	ANN-PCA 89,46
Decision Tree 98,30	Decision Tree-PCA 88,78

2.2.5 Penelitian 5

(Rokhanah dkk., 2023) melakukan penelitian yang bertujuan untuk memprediksi dini Diabetes Melitus dengan memanfaatkan PCA sebagi metode penemuan fitur optimal dalam klasifikasi prediksi ini. Penelitian ini menggabungkan metode PCA dengan algoritma *Naïve Bayes* dan *k-Nearest Neighbors* menggunakan software Rapidminer. Dataset yang digunakan adalah Dataset Prediksi Risiko Diabetes Tahap Awal LearningRepository dari *open source* Kaggle yang memiliki 7 variabel. Dari hasil analisis PCA dan klasifikasi didapatkan akurasi yang baik pada algoritma *Naïve Bayes-PCA* sebesar 90.19% dan *k-Nearest Neighbors-PCA* sebesar 93.27% dengan komponen utama yang terbentuk adalah 5. Peneliti juga membandingkan hasil

akurasi tanpa analisis komponen utama pada algoritma *k-Nearest Neighbors* sebesar 90.70% yang berarti PCA mampu meningkatkan keakuratan data sebesar 2.57%.

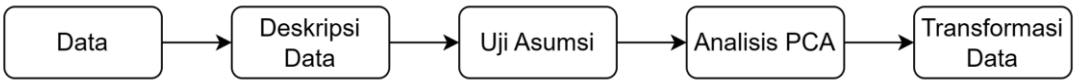
BAB III: METODOLOGI PENELITIAN

3.1 Dataset

Pada penelitian ini, kami menggunakan dataset Metro Interstate Traffic Volume, yang disediakan oleh repositori data UCI. Dataset ini terdiri 48204 baris dan 9 kolom, dimana data ini berisi informasi volume lalu lintas Interstate 94 Westbound untuk MN DoT ATR station 301, berkisar antara Minneapolis dan St. Paul, MN dari tahun 2012-2018 dengan berbagai faktor sebagai berikut:

- holiday yaitu hari libur nasional Amerika Serikat kategorikal ditambah hari libur regional, Minnesota State Fair
- temp yaitu numerik rata-rata suhu dalam kelvin
- rain_1h yaitu numeric jumlah dalam mm hujan yang terjadi dalam satu jam
- snow_1h yaitu jumlah numerik dalam mm dari salju yang terjadi dalam satu jam
- cloud_all yaitu persentase numerik tutupan awan
- weather_main yaitu kategori deskripsi tekstual singkat tentang cuaca saat ini
- weather_description yaitu kategorikal deskripsi tekstual yang lebih panjang tentang cuaca saat ini
- date_time yaitu tanggal, waktu, dan jam dari data yang dikumpulkan dalam waktu CST lokal
- traffic_volume yaitu numeric hourly I-94 ATR 301 melaporkan volume lalu lintas arah barat

3.2 Rencana Analisis



Gambar 2. Rencana Analisis

Pada penelitian ini, kami akan melakukan deskripsi data yang terdiri dari statistika deskriptif, melihat penyebaran data, membandingkan antar variabel, dan melihat korelasi antar variabel. Setelah mengetahui kondisi-kondisi di dalam data, kami akan melakukan uji asumsi. Kami akan melakukan dua uji asumsi yaitu Uji Asumsi Bartlett dan Uji Asumsi KMO. Melalui uji asumsi ini kami akan mengetahui layak atau tidaknya data ini untuk dianalisis dengan metode *Principal Component Analysis* (PCA). Apabila hasil menunjukkan layak, maka kami akan melakukan PCA. Pada tahap PCA, kami akan melakukan standarisasi data, menghitung nilai eigen dan vektor eigen, membuat

persamaan PCA, dan menghitung kumulatif varians. Setelah mengetahui jumlah dimensi data baru dari hasil menghitung kumulatif varians, maka akan dilakukan transformasi data. Hasil transformasi data ini yang dapat digunakan analisis data berikutnya.

BAB IV: HASIL DAN PEMBAHASAN

4.1 Deskripsi Data

4.1.1 Statistika Deskriptif

Statistika deskriptif yang meliputi rata - rata, median, dan lainnya digunakan untuk melihat penyebaran data. Tabel 2 menampilkan statistika deskriptif untuk tiap kolom numerik dan Tabel 3 menampilkan statistika deskriptif untuk tiap kolom object.

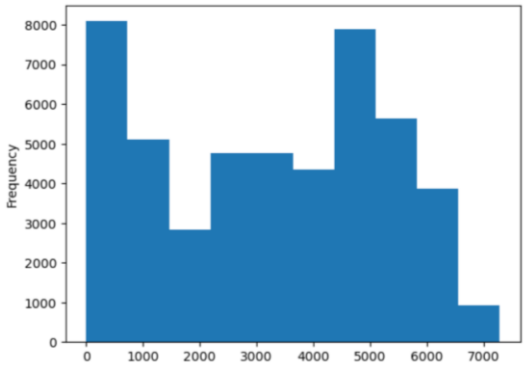
Tabel 2. statistika deskriptif untuk tiap kolom numerik

	temp	rain_1h	snow_1h	clouds_all	traffic_ volume
count	48204.0000	48204.00000	48204.00000	48204.00000	48204.00000
mean	281.205870	0.334264	0.000222	49.362231	3259.818355
std	13.338232	44.789133	0.008168	39.015750	1986.860670
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	272.160000	0.000000	0.000000	1.000000	1193.000000
50%	282.450000	0.000000	0.000000	64.000000	3380.000000
75%	291.806000	0.000000	0.000000	90.000000	4933.000000
max	310.070000	9831.300000	0.510000	100.000000	7280.000000

Tabel 3. statistika deskriptif untuk tiap kolom Object

	holiday	weather_main	weather_ description
count	48204	48204	48204
unique	12	11	38
top	None	Clouds	sky is clear
freq	48143	15164	11665

4.1.2 Penyebaran Volume Kemacetan dari Dataset

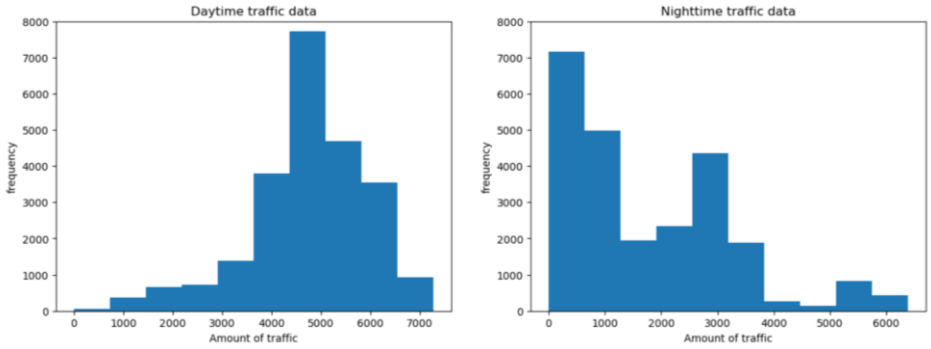


Gambar 3. Penyebaran Volume Kemacetan

Berdasarkan gambar 3, maka beberapa informasi yang bisa didapatkan, diantaranya:

- Dari 2012-10-02 09:00:00 ke 2018-09-30 23:00:00, jumlah kendaraan yang melintas bervariasi dengan range 0 hingga 7280.
- Di beberapa hari, tidak terjadi kemacetan sama sekali atau dalam data adalah 0
- 25% dari keseluruhan waktu, terdapat 4933 atau lebih jumlah kendaraan; secara kontras, terdapat 25% juga dari keseluruhan waktu, terdapat 1193 atau kurang jumlah kendaraan
- Ketika jumlah kendaraan sampai 4933 atau lebih, maka ini merupakan puncak kemacetan dalam sehari yang bisa disebabkan oleh banyaknya mobilitas untuk bekerja dan sebagainya dan ketika jumlah kendaraan hanya 1193 atau kurang, maka kemungkinan terjadi ketika hari mendekati malam.

4.1.3 Perbedaan Volume Kemacetan Berdasarkan Siang dan Malam



(a) (b)

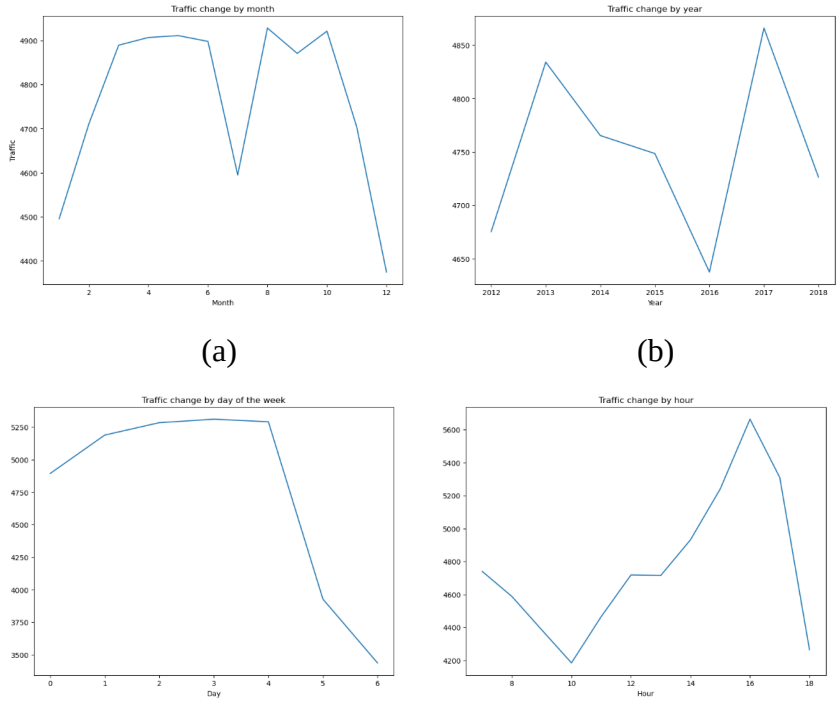
Gambar 4. Penyebaran Volume Kemacetan berdasarkan (a) siang dan (b) malam

Berdasarkan gambar 4, maka beberapa informasi yang bisa didapatkan, diantaranya:

- Kemacetan bertambah ketika di siang hari karena memiliki rata - rata jumlah kendaraan lebih tinggi dibanding ketika sudah malam.

- 75% dari keseluruhan waktu, kemacetan timbul ketika jumlah kendaraan mencapai 4252.
- Kemacetan di siang hari lebih condong ke kanan sehingga disimpulkan bahwa sebagian besar jumlah kendaraan lebih banyak.
- Kemacetan di malam hari lebih condong ke kiri sehingga disimpulkan sebagian besar jumlah kendaraan lebih sedikit.

4.1.4 Perbedaan Volume Kemacetan Berdasarkan Tahun, Bulan, Hari, dan Jam



(a) (b)

(c) (d)

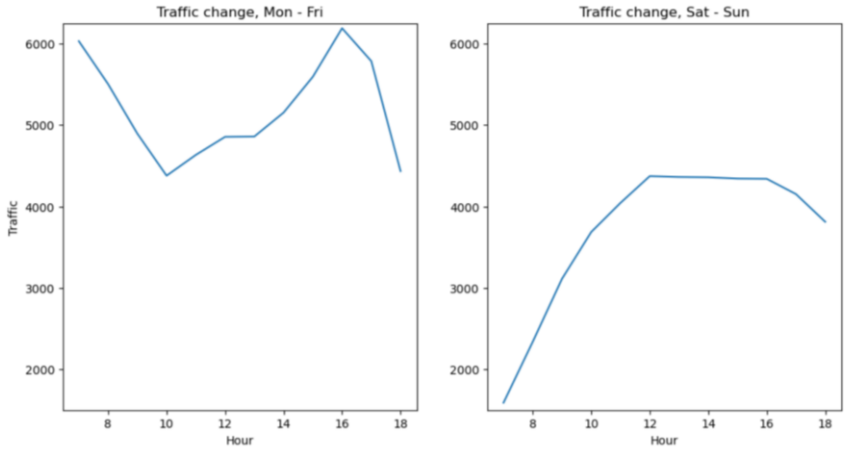
Gambar 5. Penyebaran Volume Kemacetan berdasarkan
(a) bulan, (b) tahun, (c) hari dan (d) jam

Berdasarkan gambar 5, maka beberapa informasi yang bisa didapatkan, diantaranya:

- Berdasarkan bulan: kemacetan berkurang ketika cuaca lebih dingin karena terlihat pada bulan musim salju, tingkat kemacetan berkurang.
- Berdasarkan tahun: terdapat penurunan tingkat kemacetan di tahun 2016 yang cukup signifikan namun, itu tak berlangsung lama karena pada tahun 2017, terdapat kenaikan signifikan sehingga memiliki tingkat kemacetan tertinggi.
- Berdasarkan hari: Pada hari kerja (Senin - Jumat), menjadi hari tersibuk yang terbukti dari tingkat kemacetannya selalu tinggi dan akan mengalami pengurangan di hari Sabtu dan Minggu.

- Berdasarkan jam: Pada jam 10 pagi dan jam 6 malam, jumlah kendaraan lebih sedikit, tingkat kemacetan tinggi terjadi pada jam 4 sore hingga jam 5 sore dan tingkat kemacetan mengalami kenaikan yang stabil di antara jam 2 siang hingga jam 4 sore.

4.1.5 Perbedaan Volume Kemacetan Berdasarkan Hari Kerja dan Akhir Pekan



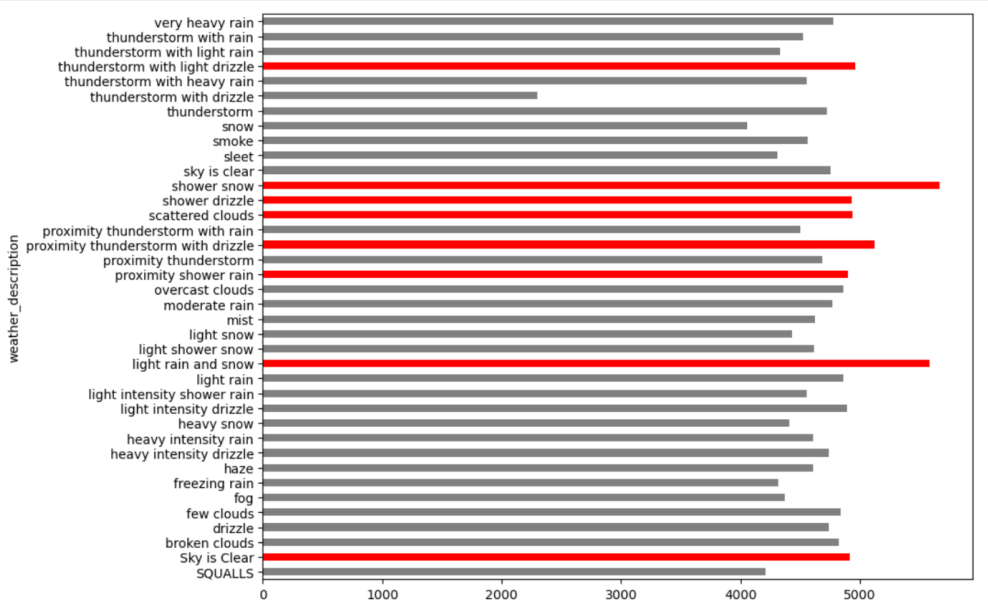
(a) (b)

Gambar 6. Penyebaran Volume Kemacetan berdasarkan (a) hari kerja dan (b) Akhir pekan

Berdasarkan gambar 6, maka beberapa informasi yang bisa didapatkan, diantaranya:

- Pada hari kerja (Senin - Jumat): tingkat kemacetan paling tinggi terjadi jam 7 pagi dan jam 4 sore. Hal ini kemungkinan disebabkan karena biasanya merupakan jam pergi kerja dan jam pulang kerja. Tingkat kemacetan mengalami penurunan di sekitar jam 10 pagi hingga jam 6 sore. Hal ini kemungkinan disebabkan karena masih banyak orang sedang bekerja atau sudah pulang kerja dan tingkat kemacetan paling tinggi dicapai ketika >6000 jumlah kendaraan.
- Pada akhir pekan (Sabtu dan Minggu): tingkat kemacetan cukup rendah hingga jam 10 pagi dan baru mengalami kenaikan serta tingkat kemacetan paling tinggi di jam 12 siang dengan jumlah kendaraan perkiraan adalah 4500

4.1.6 Korelasi Antara Kemacetan dan Cuaca



Gambar 7. Korelasi antara kemacetan dan cuaca

Berdasarkan gambar 7, maka beberapa informasi yang bisa didapatkan, diantaranya:

- Berdasarkan ‘weather_description’, salju memiliki dampak yang besar terhadap tingkat kemacetan. Hal ini didasari 2 cuaca yang berkaitan dengan salju mengalami kenaikan kemacetan dengan jumlah kendaraan lebih dari 5000.
- Cuaca badai petir ternyata juga memberikan dampak pada kenaikan kemacetan dengan jumlah kendaraan lebih dari 5000
- Ketika hari terang atau dalam cuaca yang baik, jumlah kendaraan hampir mendekati 5000 juga.

4.2 Uji Asumsi

Dalam melakukan uji asumsi, kolom ‘traffic_volume’ akan dihapus karena merupakan kolom variabel target yang tidak perlu dilakukan analisis PCA maupun uji asumsi dan kolom ‘date_time’ juga akan dihapus karena hanya mendeskripsikan waktu real-time. Untuk kolom ‘holiday’, ‘weather_main’, dan ‘weather_description’ akan diubah menjadi numerik berdasarkan nilai uniknya. Hasil tabel yang digunakan dalam melakukan uji asumsi ditampilkan pada tabel 4.

Tabel 4. Penggalan Hasil Dataset untuk melakukan uji asumsi

holiday	temp	rain_1h	snow_1h	clouds_all	weather_ main	weather_ description
7	288.28	0.0	0.0	40	1	24
7	289.36	0.0	0.0	75	1	2
7	289.58	0.0	0.0	90	1	19
7	290.13	0.0	0.0	90	1	19
7	291.14	0.0	0.0	75	1	2
…	…	…	…	…	…	…
7	283.45	0.0	0.0	75	1	2
7	282.76	0.0	0.0	90	1	19
7	282.73	0.0	0.0	90	10	21
7	282.09	0.0	0.0	90	1	19
7	282.12	0.0	0.0	90	1	19

4.2.1 Pengujian Asumsi Bartlett

Variabel $X_1, X_2 \ldots X_i$ dikatakan saling bebas jika korelasi antar variabel membentuk matriks identitas. Sehingga pengujian asumsi Bartlett dilakukan untuk menguji kebebasan antar variabel dengan H0: R = I dan H1: R ≠ I serta α: 0.05. Dengan menggunakan library factor_analyzer yang merupakan bawaan dari python, maka nilai sig yang didapatkan sebesar 0.0. Karena nilai sig < α, maka tolak H0 sehingga variabel $X_1, X_2 \ldots X$ saling berkorelasi.

4.2.2 Pengujian Asumsi KMO

Pengujian asumsi KMO dilakukan untuk melihat kelayakan data untuk digunakan dalam analisis PCA. Asumsi KMO terpenuhi apabila nilai dari KMO-Measure sampling of Adequacy (MSA) minimal 0.5. Dengan menggunakan library factor_analyzer yang merupakan bawaan dari python, maka nilai KMO-MSA adalah 0.5146 sehingga dataset yang digunakan layak untuk dilakukan analisis PCA.

4.3 Analisis PCA

4.3.1 Standarisasi Data

Dalam melakukan standarisasi data, akan digunakan modul bawaan dari python yaitu StandardScaler. Sehingga hasil dari standarisasi data ditampilkan pada tabel 5.

Tabel 5. Penggalan Hasil Dataset setelah Melakukan Standarisasi Data

holiday	temp	rain_1h	snow_1h	clouds_all	weather_ main	weather_ description
0.015856	0.530370	-0.007463	-0.027228	-0.239963	-0.566905	0.831600
0.015856	0.611341	-0.007463	-0.027228	0.657120	-0.566905	-1.621016
0.015856	0.627836	-0.007463	-0.027228	1.041584	-0.566905	0.274187
0.015856	0.669071	-0.007463	-0.027228	1.041584	-0.566905	0.274187
0.015856	0.744794	-0.007463	-0.027228	0.657120	-0.566905	-1.621016
…	…	…	…	…	…	…
0.015856	0.168250	-0.007463	-0.027228	0.657120	-0.566905	-1.621016
0.015856	0.116518	-0.007463	-0.027228	1.041584	-0.566905	0.274187
0.015856	0.114269	-0.007463	-0.027228	1.041584	2.665627	0.497152
0.015856	0.066286	-0.007463	-0.027228	1.041584	-0.566905	0.274187
0.015856	0.068535	-0.007463	-0.027228	1.041584	-0.566905	0.274187

4.3.2 Nilai Eigen dan Vektor Eigen

Proses kedua dalam melakukan analisis PCA adalah menghitung nilai eigen dan vektor eigen. Nilai eigen akan diurutkan dari yang terbesar hingga yang terkecil untuk mempermudahkan proses kumulatif total varians. Hasil dari nilai eigen (λ) beserta vektor eigen ditampilkan pada tabel 6.

Tabel 6. Hasil Nilai Eigen dan Vektor Eigen

i	λ_i	E_i
1	1.6789940137414496	[0.00512839 -0.09921167 0.00466836 0.05877816 0.6708817 0.58555702 -0.44010772]
2	1.0461274685706068	[-0.01661115 -0.77522602 -0.03535758 0.47060224 0.02700629 0.09702418 0.40729477]
3	1.0024123782255512	[-0.1753174 0.10044394 0.96260717 0.15021183 -0.00346126 0.05989296 0.07999704]
4	1.0001210087195187	[-0.9838341 -0.00099574 -0.17215491 -0.04682317 -0.00335753 0.01389438 -0.00595108]
5	0.9857985004653943	[-0.0103408 0.46656645 -0.17406183 0.85805192 -0.05559166 -0.02418512 -0.10946622]
6	0.8480965343073124	[-0.02794379 -0.38053296 0.1102543 0.11859085 -0.04933539 -0.55897764

		-0.71645239]
7	0.4385953151476951	[0.01201038 -0.12884916 0.00496416 -0.00240269 -0.73731997 0.57522877 -0.32968826]

Berdasarkan hasil perhitungan nilai eigen, nilai yang lebih dari 1 didapatkan ketika i = 4 dan ada satu nilai eigen ketika i=5 mendekati nilai 1 artinya bahwa akan ada 5 dimensi baru yang dihasilkan dari proses PCA.

4.3.3 Persamaan PCA

Berdasarkan nilai vektor eigen yang telah ditemukan, maka terdapat 7 persamaan PCA yang didapatkan. Persamaan dibawah adalah hasil persamaannya.

Y_1 = 0.00512839 (clouds_all)- 0.09921167(weather_main) + 0.00466836(weather_description) + 0.05877816(temp) + 0.6708817(snow_1h) + 0.58555702(holiday) - 0.44010772(rain_1h)

Y_2 = -0.01661115(temp)-0.77522602(snow_1h) -0.03535758(weather_description) + 0.47060224(weather_main) + 0.02700629(rain_1h) + 0.09702418(clouds_all) - 0.40729477(holiday)

Y_3 = -0.1753174(rain_1h)+ 0.10044394(holiday)+ 0.96260717(snow_1h) + 0.15021183(temp) -0.00346126(weather_description) + 0.05989296(weather_main) + 0.07999704(clouds_all)

Y_4 = -0.9838341(holiday) -0.00099574(rain_1h) -0.17215491(snow_1h)- 0.04682317(weather_main) -0.00335753 (weather_description)+ 0.01389438(clouds_all) -0.00595108(temp)

Y_5 = -0.0103408(snow_1h) +0.46656645(temp) -0.17406183(rain_1h) +0.85805192(weather_description) -0.05559166(clouds_all) - 0.02418512(weather_main) -0.10946622(holiday)

Y_6 = -0.02794379(weather_description) -0.38053296(weather_main)+ 0.1102543(temp) + 0.11859085(snow_1h) -0.04933539(rain_1h) - 0.55897764(clouds_all) -0.71645239(holiday)

Y_7 = 0.01201038 (clouds_all) -0.12884916(weather_main)+ 0.00496416(weather_description) -0.00240269 (temp) -0.73731997(holiday) + 0.57522877 (rain_1h) -0.32968826 (snow_1h)

4.3.4 Kumulatif Total Varians

Proses selanjutnya adalah menghitung total varians dan proporsi kumulatifnya. Hasil perhitungan ditampilkan pada tabel 7.

Tabel 7. Hasil Total Varians dan Proposi Kumulatif

i	Total Varians ke-i	Proporsi kumulatif dari total varians
1	0.2399	0.2399
2	0.1494	0.3893
3	0.1432	0.5325
4	0.1429	0.6754
5	0.1408	0.8162
6	0.1212	0.9373
7	0.0627	1.000

Berdasarkan hasil perhitungan proporsi kumulatif dari total varians, nilai yang lebih dari 0.7 didapatkan ketika i = 5 artinya bahwa akan ada 5 dimensi baru yang dihasilkan dari proses PCA.

4.4 Transformasi Data

Transformasi data adalah proses perubahan dataset awal menjadi dataset baru berdasarkan m-dimensi yang didapatkan. hasil dataset setelah melakukan proses PCA ditampilkan pada tabel 8.

Tabel 8. Penggalan Hasil Dataset setelah Melakukan Proses PCA

clouds_all	temp	rain_1h	holiday	snow_1h
-0.913109	-0.146748	0.072621	-0.025588	0.161243
0.760110	-1.184229	-0.118553	-0.014085	0.417630
0.182310	-0.414727	0.033384	-0.026670	0.196492
0.178219	-0.446694	0.033384	-0.026711	0.215731
0.746870	-1.287685	-0.105148	-0.014217	0.479894
…	…	…	…	…
0.804070	-0.840733	-0.163059	-0.013643	0.210898
0.233038	-0.018340	-0.017975	-0.026161	-0.042072
2.027965	0.387850	0.193242	0.017428	-0.145707
0.238022	0.020601	-0.023020	-0.026111	-0.065508
0.237799	0.018857	-0.022795	-0.026113	-0.064459

BAB V: KESIMPULAN

Berdasarkan Bartlett test mendapatkan nilai signifikansi 0.0 yang artinya kurang dari nilai signifikan 0.05 sehingga H0 dapat ditolak yang menyebabkan adanya korelasi antar variabel independen. Selain itu, Berdasarkan KMO test mendapatkan nilai 0.5146 yang artinya lebih dari 0.5 sehingga data layak dilakukan PCA. Berdasarkan nilai eigen, terdapat 5 fitur yang memiliki nilai eigen lebih dari 1 maupun mendekati 1 sehingga disimpulkan data dapat tereduksi hingga 5 fitur. Berdasarkan kumulatif total varians, nilai komponen atau fitur yang memiliki lebih dari 70% adalah 5. Sehingga disimpulkan data dapat tereduksi hingga 5 fitur saja. Karena syarat nilai eigen dan syarat kumulatif total varians mendapatkan hasil yang sama maka m dimensi data baru setelah melakukan PCA adalah sebanyak 5 dimensi data yaitu clouds_all, temp, rain_1h, holiday, snow_1h.

DAFTAR PUSTAKA

[1] *Daftar Negara Menurut Jumlah penduduk* (2023) *Wikipedia*. Available at: https://id.wikipedia.org/wiki/Daftar_negara_menurut_jumlah_penduduk (Accessed: 16 May 2023).

[2] *Jakarta Traffic Report: Tomtom traffic index* (2023) *Jakarta traffic report | TomTom Traffic Index*. Available at: https://www.tomtom.com/traffic-index/jakarta-traffic/ (Accessed: 16 May 2023).

[3] *Principal component analysis(pca)* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/ml-principal-component-analysispca/ (Accessed: 16 May 2023).

[4] Jolliffe, I.T. and Cadima, J. (2016) 'Principal component analysis: A review and recent developments', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), p. 20150202. doi:10.1098/rsta.2015.0202.

[5] Hendro, G., Adji, T.B. and Setiawan, N.A., 2012. Penggunaan metodologi analisa komponen utama (PCA) untuk mereduksi faktor-faktor yang mempengaruhi penyakit jantung koroner. *Semin. Nas. ScrETec*, pp.1-5.

[6] R-Stats (2019) *Analisis Komponen Utama (principal component analysis)*, *Rumus Statistik*. Available at: https://www.rumusstatistik.com/2015/03/analisis-komponen-utama-principal.html (Accessed: 16 May 2023).

[7] Mangale, S. (2020) *Scree plot*, *Medium*. Available at: https://sanchitamangale12.medium.com/scree-plot-733ed72c8608 (Accessed: 16 May 2023).

[8] Rahardjo, B., 2013. Analisis faktor untuk mengetahui pengaruh personal selling dan word of mouth tehadap keputusan pembelian suatu studi kasus pada PT. Starmas inti alumunium industry. *Jurnal Ekonomika dan Manajemen*, *2*(1).

[9] Puspitasari, E., Mukid, M.A. and Sudarno, S., 2014. Perbandingan Analisis Faktor Klasik Dan Analisis Faktor Robust Untuk Data Inflasi Kelompok Bahan Makanan Di Jawa Tengah. *Jurnal Gaussian*, *3*(3), pp.343-352.

[10] Ritonga, A.S. and Muhandhis, I., 2021. Teknik Data Mining Untuk Mengklasifikasikan Data Ulasan Destinasi Wisata Menggunakan Reduksi Data Principal Component Analysis (Pca). *Jurnal Ilmiah Edutic: Pendidikan dan Informatika*, *7*(2), pp.124-133.

[11] Rokhanah, S., Hermawan, A. and Avianto, D., 2023. Pengaruh Principal Component Analysis Pada Naïve Bayes dan K-Nearest Neighbor Untuk Prediksi Dini Diabetes Melitus Menggunakan Rapidminer. *EVOLUSI: Jurnal Sains dan Manajemen*, *11*(1).

[12] Rozy, F., Daulay, D.A.A. and Vigriawan, G.E., 2022. Penerapan Metode Principal Component Analysis Terhadap Faktor Biomotor Yang Mempengaruhi Performa Atlet Atletik Sumatra Utara. *PENJAGA: Pendidikan Jasmani dan Olahraga*, *3*(1), pp.1-7.

[13] Wangge, M., 2021. Penerapan Metode Principal Component Analysis (PCA) Terhadap Faktor-faktor yang Mempengaruhi Lamanya Penyelesaian Skripsi Mahasiswa Program Studi Pendidikan Matematika FKIP UNDANA. *Jurnal Cendekia: Jurnal Pendidikan Matematika*, *5*(2), pp.974-988.

[14] Masithoh, H., Putri, H.M.R., Pratiwi, J.R., Fahira, J.N., Nafi'Uddin, M.L., Sunarmi, N. and Sifa, W.A., 2022, July. Analisis Unsur Cuaca Dibidang Pertanian Malang 2018-2021 Menggunakan Metode Principal Component Analysis. In *SNPF (Seminar Nasional Pendidikan Fisika)*.

LAMPIRAN

Lampiran 1: Potongan Script Pemrograman Python dan Output dalam Melakukan Deskripsi Data

```
data.describe()
```

[44]:

	temp	rain_1h	snow_1h	clouds_all	traffic_volume
count	48204.000000	48204.000000	48204.000000	48204.000000	48204.000000
mean	281.205870	0.334264	0.000222	49.362231	3259.818355
std	13.338232	44.789133	0.008168	39.015750	1986.860670
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	272.160000	0.000000	0.000000	1.000000	1193.000000
50%	282.450000	0.000000	0.000000	64.000000	3380.000000
75%	291.806000	0.000000	0.000000	90.000000	4933.000000
max	310.070000	9831.300000	0.510000	100.000000	7280.000000

```
data.describe(include='object')
```

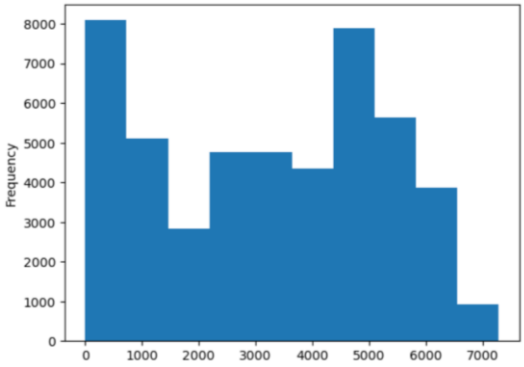
[45]:

	holiday	weather_main	weather_description
count	48204	48204	48204
unique	12	11	38
top	None	Clouds	sky is clear
freq	48143	15164	11665

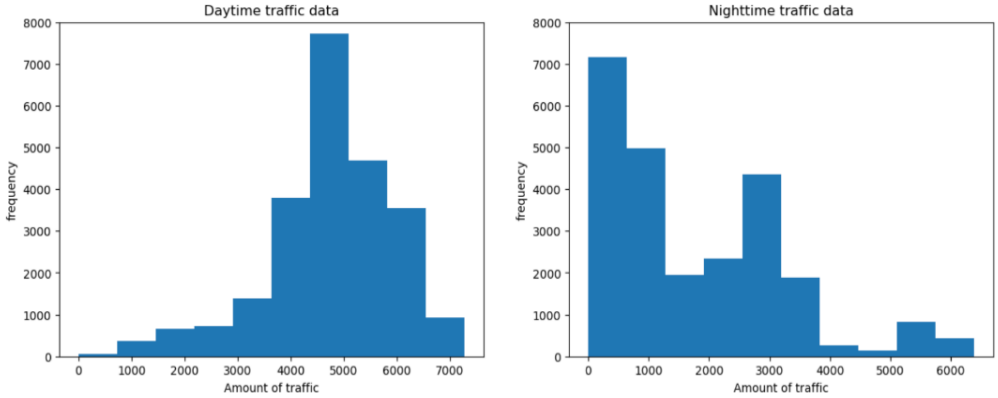
```
data['date_time'] = pd.to_datetime(data['date_time'])
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   holiday              48204 non-null  object
 1   temp                 48204 non-null  float64
 2   rain_1h              48204 non-null  float64
 3   snow_1h              48204 non-null  float64
 4   clouds_all           48204 non-null  int64
 5   weather_main         48204 non-null  object
 6   weather_description  48204 non-null  object
 7   date_time            48204 non-null  datetime64[ns]
 8   traffic_volume       48204 non-null  int64
dtypes: datetime64[ns](1), float64(3), int64(2), object(3)
memory usage: 3.3+ MB
```

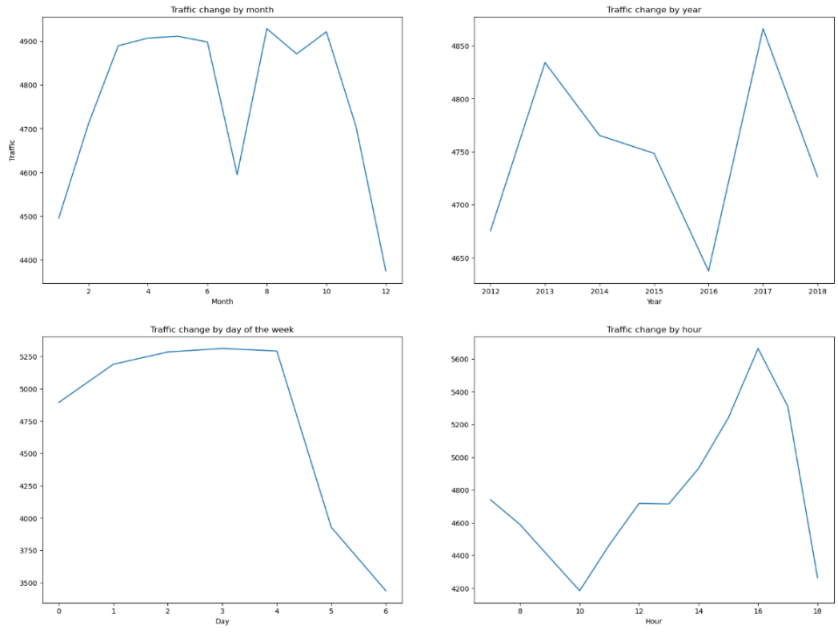
```
data['traffic_volume'].plot.hist()
plt.show()
```



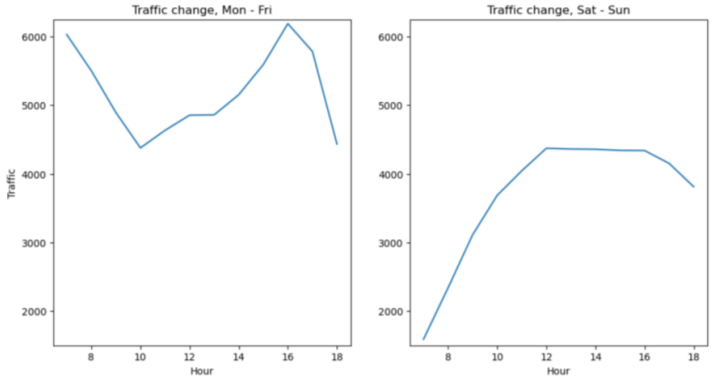

```
daytime_data = data.copy()[(data['date_time'].dt.hour >= 7) &
(data['date_time'].dt.hour < 19)]
nighttime_data = data.copy()[(data['date_time'].dt.hour >= 19) |
(data['date_time'].dt.hour < 7)]

plt.figure(figsize=(15, 5))
plt.subplot(1, 2, 1)
plt.hist(daytime_data['traffic_volume'])
plt.title('Daytime traffic data')
plt.xlabel('Amount of traffic')
plt.ylabel('frequency')
plt.ylim([0, 8000])

plt.subplot(1, 2, 2)
plt.hist(nighttime_data['traffic_volume'])
plt.title('Nighttime traffic data')
plt.xlabel('Amount of traffic')
plt.ylabel('frequency')
plt.ylim([0, 8000])

plt.show()
```



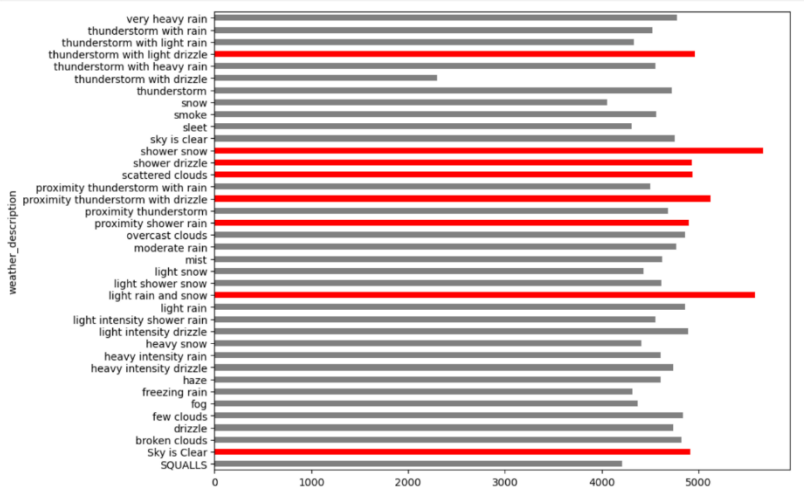
```
daytime_data['month']=daytime_data['date_time'].dt.month
by_month_day = daytime_data.groupby('month').mean()

daytime_data['year'] = daytime_data['date_time'].dt.year
by_year_day = daytime_data.groupby('year').mean()

daytime_data['hour'] = daytime_data['date_time'].dt.hour
by_hour_day = daytime_data.groupby('hour').mean()

daytime_data['dayofweek']=
daytime_data['date_time'].dt.dayofweek
by_dayofweek = daytime_data.groupby('dayofweek').mean()

plt.figure()
plt.figure(figsize=(20, 15))
plt.subplot(2, 2, 1)
plt.plot(by_month_day['traffic_volume'])
plt.title("Traffic change by month")
plt.xlabel("Month")
plt.ylabel("Traffic")

plt.subplot(2, 2, 2)
plt.plot(by_year_day['traffic_volume'])
plt.title("Traffic change by year")
plt.xlabel("Year")

plt.subplot(2, 2, 3)
plt.plot(by_dayofweek['traffic_volume'])
plt.title("Traffic change by day of the week")
plt.xlabel("Day")

plt.subplot(2, 2, 4)
plt.plot(by_hour_day['traffic_volume'])
plt.title("Traffic change by hour")
plt.xlabel("Hour")

plt.show()
```



```
daytime_data['hour'] = daytime_data['date_time'].dt.hour
bussiness_days = daytime_data.copy()[daytime_data['dayofweek']
<= 4]
weekend=daytime_data.copy()[daytime_data['dayofweek']>= 5]
by_hour_business = bussiness_days.groupby('hour').mean()
by_hour_weekend = weekend.groupby('hour').mean()

plt.figure()
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(by_hour_business['traffic_volume'])
plt.title("Traffic change, Mon - Fri")
plt.xlabel("Hour")
plt.ylim([1500, 6250])
plt.ylabel("Traffic")

plt.subplot(1, 2, 2)
plt.plot(by_hour_weekend['traffic_volume'])
plt.title("Traffic change, Sat - Sun")
plt.xlabel("Hour")
plt.ylim([1500, 6250])
```




```
by_weather_description =
daytime_data.groupby('weather_description').mean()

colors = ["red" if i > 4900 else "grey" for i in
by_weather_description['traffic_volume']]
by_weather_description['traffic_volume'].plot.barh(figsize=(10,8
), color=colors)
plt.show()
```



Lampiran 2: Potongan Script Pemrograman Python dan Output dalam Melakukan Uji Asumsi

```
del data['date_time']
del data['traffic_volume']
data
```

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description
0	None	288.28	0.0	0.0	40	Clouds	scattered clouds
1	None	289.36	0.0	0.0	75	Clouds	broken clouds
2	None	289.58	0.0	0.0	90	Clouds	overcast clouds
3	None	290.13	0.0	0.0	90	Clouds	overcast clouds
4	None	291.14	0.0	0.0	75	Clouds	broken clouds
...	...	...	...	...	...	...	...
48199	None	283.45	0.0	0.0	75	Clouds	broken clouds
48200	None	282.76	0.0	0.0	90	Clouds	overcast clouds
48201	None	282.73	0.0	0.0	90	Thunderstorm	proximity thunderstorm
48202	None	282.09	0.0	0.0	90	Clouds	overcast clouds
48203	None	282.12	0.0	0.0	90	Clouds	overcast clouds

48204 rows × 7 columns

```
# mengambil nilai unik dari setiap kolom
unique_holiday = sorted(data['holiday'].unique())
unique_weather_main = sorted(data['weather_main'].unique())
unique_weather_description =
sorted(data['weather_description'].unique())

# membuat kamus untuk mengubah nilai kategori menjadi nilai
numerik
holiday_dict = dict(zip(unique_holiday,
range(len(unique_holiday))))
weather_main_dict = dict(zip(unique_weather_main,
range(len(unique_weather_main))))
weather_description_dict = dict(zip(unique_weather_description,
range(len(unique_weather_description))))

# melakukan pengkodean label
df_encoded = data.copy()
df_encoded['holiday'] = df_encoded['holiday'].apply(lambda x:
holiday_dict[x])
df_encoded['weather_main'] =
df_encoded['weather_main'].apply(lambda x: weather_main_dict[x])
df_encoded['weather_description'] =
df_encoded['weather_description'].apply(lambda x:
weather_description_dict[x])

# tampilkan hasil encoding
df_encoded
```

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description
0	7	288.28	0.0	0.0	40	1	24
1	7	289.36	0.0	0.0	75	1	2
2	7	289.58	0.0	0.0	90	1	19
3	7	290.13	0.0	0.0	90	1	19
4	7	291.14	0.0	0.0	75	1	2
...	...	...	...	...	...	...	...
48199	7	283.45	0.0	0.0	75	1	2
48200	7	282.76	0.0	0.0	90	1	19
48201	7	282.73	0.0	0.0	90	10	21
48202	7	282.09	0.0	0.0	90	1	19
48203	7	282.12	0.0	0.0	90	1	19

48204 rows × 7 columns

```
df_encoded.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 7 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   holiday              48204 non-null  int64  
 1   temp                 48204 non-null  float64
 2   rain_1h              48204 non-null  float64
 3   snow_1h              48204 non-null  float64
 4   clouds_all           48204 non-null  int64  
 5   weather_main         48204 non-null  int64  
 6   weather_description  48204 non-null  int64  
dtypes: float64(3), int64(4)
memory usage: 2.6 MB
```

```
from factor_analyzer import FactorAnalyzer
from factor_analyzer.factor_analyzer import
calculate_bartlett_sphericity
from factor_analyzer.factor_analyzer import calculate_kmo

#CHECK ADEQUACY
#Bartlett
#p-value should be <0.05
chi_square_value,p_value=calculate_bartlett_sphericity(df_encoded)
print('significant value of bartlett test:', p_value)

#KMO
#Value should be >0.5
kmo_all,kmo_model=calculate_kmo(df_encoded)
print("result of KMO test:", kmo_model)
```

```
significant value of bartlett test: 0.0
result of KMO test: 0.514604781176736
```

Lampiran 3: Potongan Script Pemrograman Python dan Output dalam Melakukan Analisis PCA

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_standardize=df_encoded.copy()
df_standardize=pd.DataFrame(scaler.fit_transform(df_standardize)
, columns=df_standardize.columns)
df_standardize
```

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description
0	0.015856	0.530370	-0.007463	-0.027228	-0.239963	-0.566905	0.831600
1	0.015856	0.611341	-0.007463	-0.027228	0.657120	-0.566905	-1.621016
2	0.015856	0.627836	-0.007463	-0.027228	1.041584	-0.566905	0.274187
3	0.015856	0.669071	-0.007463	-0.027228	1.041584	-0.566905	0.274187
4	0.015856	0.744794	-0.007463	-0.027228	0.657120	-0.566905	-1.621016
...	...	...	...	...	...	...	...
48199	0.015856	0.168250	-0.007463	-0.027228	0.657120	-0.566905	-1.621016
48200	0.015856	0.116518	-0.007463	-0.027228	1.041584	-0.566905	0.274187
48201	0.015856	0.114269	-0.007463	-0.027228	1.041584	2.665627	0.497152
48202	0.015856	0.066286	-0.007463	-0.027228	1.041584	-0.566905	0.274187
48203	0.015856	0.068535	-0.007463	-0.027228	1.041584	-0.566905	0.274187

48204 rows × 7 columns

```
# Get the eigenvalues and eigenvectors
eigenvalues = pca.explained_variance_
eigenvectors = pca.components_

# Print the eigenvalues and corresponding eigenvectors
for i, (eigenvalue, eigenvector) in enumerate(zip(eigenvalues,
eigenvectors)):
    print(f"Eigenvalue {i+1}: {eigenvalue}")
    print(f"Eigenvector {i+1}: {eigenvector}")
    print("Corresponding Features:")

    # Get the indices of top absolute values in the eigenvector
    component_indices = np.argsort(np.abs(eigenvector))[::-1]

    # Print the corresponding column names
    for component_index in component_indices:
        print(column_names[component_index])

    print()

# Get the explained variance ratios and cumulative explained
variance
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_explained_variance =
np.cumsum(explained_variance_ratio)

# Print the explained variance ratios
print("\nExplained Variance Ratios:")
for i, ratio in enumerate(explained_variance_ratio):
    print(f"PC{i+1}: {ratio:.4f}")
```

```
# Print the cumulative explained variance
print("\nCumulative Explained Variance:")
for i, variance in enumerate(cumulative_explained_variance):
    print(f"PC{i+1}: {variance:.4f}")

# Select the number of components to retain based on the desired
variance threshold
variance_threshold = 0.7
n_components = np.argmax(cumulative_explained_variance >=
variance_threshold) + 1
print(f"\nNumber of components to retain for
{variance_threshold:.0%} variance: {n_components}")

# Perform PCA with the selected number of components
pca_selected = PCA(n_components=n_components)
df_transformed = pca_selected.fit_transform(df_standardize)

# Get the most contributing feature for each principal component
most_contributing_features = []

# Iterate over the principal components
for i, eigenvector in enumerate(eigenvectors[:n_components]):
    component_indices = np.argsort(np.abs(eigenvector))[::-1]
    most_contributing_feature = column_names[component_indices[0]]
    most_contributing_features.append(most_contributing_feature)

# Print the most contributing features for each principal
component
for i, feature in enumerate(most_contributing_features):
    print(f"Principal Component {i+1}: {feature}")
```

```
Eigenvalue 1: 1.6789940137414496
Eigenvector 1: [ 0.00512839 -0.09921167  0.00466836  0.05877816  0.6708817   0.58555702
 -0.44010772]
Corresponding Features:
clouds_all
weather_main
weather_description
temp
snow_1h
holiday
rain_1h

Eigenvalue 2: 1.0461274685706068
Eigenvector 2: [-0.01661115 -0.77522602 -0.03535758  0.47060224  0.02700629  0.09702418
  0.40729477]
Corresponding Features:
temp
snow_1h
weather_description
weather_main
rain_1h
clouds_all
holiday

Eigenvalue 3: 1.0024123782255512
Eigenvector 3: [-0.1753174   0.10044394  0.96260717  0.15021183 -0.00346126  0.05989296
  0.07999704]
Corresponding Features:
rain_1h
holiday
snow_1h
temp
weather_description
weather_main
clouds_all
```

```
Eigenvalue 4: 1.0001210087195187
Eigenvector 4: [-0.9838341  -0.00099574 -0.17215491 -0.04682317 -0.00335753  0.01389438
 -0.00595108]
Corresponding Features:
holiday
rain_1h
snow_1h
weather_main
weather_description
clouds_all
temp

Eigenvalue 5: 0.9857985004653943
Eigenvector 5: [-0.0103408   0.46656645 -0.17406183  0.85805192 -0.05559166 -0.02418512
 -0.10946622]
Corresponding Features:
snow_1h
temp
rain_1h
weather_description
clouds_all
weather_main
holiday

Eigenvalue 6: 0.8480965343073124
Eigenvector 6: [-0.02794379 -0.38053296  0.1102543   0.11859085 -0.04933539 -0.55897764
 -0.71645239]
Corresponding Features:
weather_description
weather_main
temp
snow_1h
rain_1h
clouds_all
holiday
Eigenvalue 7: 0.4385953151476951
Eigenvector 7: [ 0.01201038 -0.12884916  0.00496416 -0.00240269 -0.73731997  0.57522877
 -0.32968826]
Corresponding Features:
clouds_all
weather_main
weather_description
temp
holiday
rain_1h
snow_1h


Explained Variance Ratios:
PC1: 0.2399
PC2: 0.1494
PC3: 0.1432
PC4: 0.1429
PC5: 0.1408
PC6: 0.1212
PC7: 0.0627

Cumulative Explained Variance:
PC1: 0.2399
PC2: 0.3893
PC3: 0.5325
PC4: 0.6754
PC5: 0.8162
PC6: 0.9373
PC7: 1.0000

Number of components to retain for 70% variance: 5
Principal Component 1: clouds_all
Principal Component 2: temp
Principal Component 3: rain_1h
Principal Component 4: holiday
Principal Component 5: snow_1h
```

Lampiran 4: Potongan Script Pemrograman Python dan Output dalam Melakukan Transformasi Data Berdasarkan Hasil PCA

```
# Perform PCA with the selected number of components
pca_selected = PCA(n_components=n_components)
df_transformed = pca_selected.fit_transform(df_standardize)

# Create a new DataFrame with the transformed data
df_transformed_original = pd.DataFrame(df_transformed,
columns=[most_contributing_columns[i] for i in
range(n_components)])

# Print the transformed DataFrame
df_transformed_original
```

	clouds_all	temp	rain_1h	holiday	snow_1h
0	-0.913109	-0.146748	0.072621	-0.025588	0.161243
1	0.760110	-1.184229	-0.118553	-0.014085	0.417630
2	0.182310	-0.414727	0.033384	-0.026670	0.196492
3	0.178219	-0.446694	0.037526	-0.026711	0.215731
4	0.746870	-1.287685	-0.105148	-0.014217	0.479894
...	...	...	...	...	...
48199	0.804070	-0.840733	-0.163059	-0.013643	0.210898
48200	0.233038	-0.018340	-0.017975	-0.026161	-0.042072
48201	2.027965	0.387850	0.193242	0.017428	-0.145707
48202	0.238022	0.020601	-0.023020	-0.026111	-0.065508
48203	0.237799	0.018857	-0.022795	-0.026113	-0.064459

48204 rows × 5 columns