Tahapan Preprocessing Dataset Rumah Sakit Indonesia Pemerintah vs Swasta per Provinsi

DISUSUN OLEH:

NAMA: FAUZAN MUHAMMAD ZAHRAN

NIM :105841116723

KELAS: 5E

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025

PREPROCESSING DAN FEATURE ENGINEERING

DATASET RUMAH SAKIT INDONESIA

A. Tujuan Laporan

Tujuan laporan ini adalah menjelaskan seluruh tahapan preprocessing dan feature engineering yang dilakukan pada dataset rumah sakit Indonesia. Proses ini mencakup pembersihan struktur tabel, penanganan nilai hilang, standarisasi tipe data, deteksi dan penanganan outlier, normalisasi fitur numerik.

Tahapan ini dilakukan agar dataset menjadi lebih bersih, terstruktur, dan informatif sehingga dapat digunakan untuk analisis lanjutan, seperti pemodelan machine learning, evaluasi kapasitas kesehatan, dan pemetaan pemerataan fasilitas rumah sakit di seluruh provinsi.

B. Tahapan Lengkap Preprocessing dan Feature Engineering

1) Tahap Import Library

Inisialisasi Environment Pada tahap awal, dilakukan import library utama yang dibutuhkan untuk pengolahan data:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

Fungsi:

- **pandas, numpy** → manipulasi data & operasi numerik
- **MinMaxScaler, StandardScaler** → penskalaan fitur
- **matplotlib** → visualisasi data
- **Path** → manajemen file & direktori.

2) Tahap Load Dataset

```
file_path = "Hospital_Indonesia_datasets.csv"
df_raw = pd.read_csv(file_path, engine="python", sep=None)

print("Data Asli (5 baris pertama):")
print(df_raw.head())

print("\nInfo Data:")
print(df_raw.info())

print("\nMissing per kolom:")
print(df_raw.isnull().sum())
```

0.0s

```
Data Asli (5 baris pertama):
        id                                  nama propinsi               kab  \
0  1110053                   RS Arun Lhokseumawe     Aceh   Kota Lhokseumawe
1  1106014                       RS Umum Fandika     Aceh        Aceh Tengah
2  1171110                RS Umum Daerah Meuraxa     Aceh    Kota Banda Aceh
3  1171163  RS Gigi Mulut Universitas Syiah Kuala     Aceh    Kota Banda Aceh
4  1102027       RS Umum Daerah Kota Subulussalam     Aceh  Kota Subulussalam

                                      alamat  \
0  Jl. Plaju Komplek Perumahan PT Arun Batuphat T...
1  Jl. Terminal Simpang Wariji Blangkolak 1 Kec. ...
2  Jl. Soekarno Hatta Km. 2 Desa Mibo Kecamatan B...
3  Jl. Prof A. Madjid Ibrahim I No. 5 Banda Aceh ...
4    Jl. Hamzah Fansyuri (Subulussalam-Rundeng) K...

                          jenis kelas    status_blu       kepemilikan  \
0                Rumah Sakit Umum     C  Non BLU/BLUD    SWASTA/LAINNYA
1                Rumah Sakit Umum     D  Non BLU/BLUD    SWASTA/LAINNYA
2                Rumah Sakit Umum     B          BLUD            Pemkot
3  Rumah Sakit Khusus Gigi dan Mulut     B           BLU  Kementerian Lain
4                Rumah Sakit Umum     C          BLUD            Pemkot

   total_tempat_tidur  total_layanan  total_tenaga_kerja
0                 218             36                 328
1                  45             15                  45
...
total_tempat_tidur    0
total_layanan         0
total_tenaga_kerja    0
dtype: int64
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Dataset rumah sakit dimuat dari file CSV dan ditampilkan menggunakan df.head() untuk melihat lima baris awal sebagai langkah verifikasi awal bahwa file berhasil dibaca dengan benar. Dari hasil tampilan tersebut terlihat bahwa struktur dataset sudah rapi, dengan kolom-kolom seperti *id*, *nama*, *provinsi*, *kabupaten/kota*, *alamat*, *jenis*, *kelas*, *status_blu*, *kepemilikan*, serta fitur numerik seperti *total_tempat_tidur*, *total_layanan*, dan *total_tenaga_kerja* terbaca tanpa adanya header ganda atau kolom “Unnamed”, sehingga tidak memerlukan pembersihan struktur seperti pada dataset peserta didik SD. Informasi tambahan dari df.info() dan pemeriksaan nilai hilang melalui df.isnull().sum() menunjukkan tipe data dan jumlah missing value pada setiap kolom, yang menjadi dasar penting sebelum melanjutkan ke tahap preprocessing berikutnya seperti imputasi, standarisasi, dan feature engineering.

3) Tahap Membersihkan dan Menata Ulang Header

```python
df_raw.columns = df_raw.columns.str.strip()

COL_ID = "id"
COL_NAMA = "nama"
COL_PROV = "propinsi"
COL_KAB = "kab"
COL_ALAMAT = "alamat"
COL_JENIS = "jenis"
COL_KELAS = "kelas"
COL_BLU = "status_blu"
COL_OWNER = "kepemilikan"
COL_TT = "total_tempat_tidur"
COL_LAYANAN = "total_layanan"
COL_TK = "total_tenaga_kerja"


needed = [COL_JENIS, COL_OWNER, COL_PROV]
missing = [c for c in needed if c not in df_raw.columns]
assert not missing, f"Kolom kunci tidak ditemukan: {missing}. Pastikan header sesuai."

df = df_raw.copy()

✓ 0.0s
```

Kode tersebut melakukan tahap awal pembersihan dataset dengan merapikan nama kolom dan memastikan bahwa kolom-kolom penting tersedia sebelum data diproses lebih lanjut. Pertama, seluruh nama kolom dibersihkan menggunakan .str.strip() untuk menghilangkan spasi berlebih di awal atau akhir sehingga penamaan kolom menjadi konsisten. Setelah itu, beberapa variabel dibuat untuk mendefinisikan nama-nama kolom utama seperti *id*, *nama*, *provinsi*, *kab*, *alamat*, *jenis*, *kelas*, *status_blu*, *kepemilikan*, serta kolom numerik seperti *total_tempat_tidur*, *total_layanan*, dan *total_tenaga_kerja*. Selanjutnya, kode memeriksa apakah kolom kunci seperti *jenis*, *kepemilikan*, dan *provinsi* benar-benar ada di dalam dataset; jika ada yang hilang, program menghentikan proses dan memberikan pesan kesalahan agar pengguna mengetahui bahwa header perlu diperiksa kembali. Setelah semua kolom dipastikan lengkap, dataframe kemudian disalin untuk pemrosesan tahap berikutnya.

4) Tahap Normalisasi Nilai dan Filter

```
HOSPITAL_ALIASES = {
    "RUMAH SAKIT", "RUMAH SAKIT UMUM", "RUMAH SAKIT KHUSUS",
    "RS", "RSU", "RSUD", "HOSPITAL", "GENERAL HOSPITAL", "SPECIALIST HOSPITAL"
}

for c in [COL_JENIS, COL_OWNER, COL_PROV]:
    df[c] = df[c].astype(str).str.strip()

df[COL_JENIS] = df[COL_JENIS].str.upper()
df[COL_OWNER] = df[COL_OWNER].str.title()
df[COL_PROV]  = df[COL_PROV].str.title()

def is_hospital(x: str) -> bool:
    x = str(x).upper().strip()
    return (x in HOSPITAL_ALIASES) or ("RUMAH SAKIT" in x) or (x.startswith("RS")) or ("HOSPITAL" in x)

df = df[df[COL_JENIS].apply(is_hospital)].copy()
print("Ukuran setelah filter RS:", df.shape)
display(df.head(5))

0.0s
```

```
Ukuran setelah filter RS: (3151, 12)
```

	id	nama	propinsi	kab	alamat	jenis	kelas	status_blu	kepemilikan	total_tempat_tidur	total_layanan	total_tenaga_kerja
0	1110053	RS Arun Lhokseumawe	Aceh	Kota Lhokseumawe	Jl. Plaju Komplek Perumahan PT Arun Batuphat T...	RUMAH SAKIT UMUM	C	Non BLU/BLUD	Swasta/Lainnya	218	36	328
1	1106014	RS Umum Fandika	Aceh	Aceh Tengah	Jl. Terminal Simpang Warjii Blangkolak 1 Kec. ...	RUMAH SAKIT UMUM	D	Non BLU/BLUD	Swasta/Lainnya	45	15	45
2	1171110	RS Umum Daerah Meuraxa	Aceh	Kota Banda Aceh	Jl. Soekarno Hatta Km. 2 Desa Mibo Kecamatan B...	RUMAH SAKIT UMUM	B	BLUD	Pemkot	310	77	487
3	1171163	RS Gigi Mulut Universitas Syiah Kuala	Aceh	Kota Banda Aceh	Jl. Prof A. Madjid Ibrahim I No. 5 Banda Aceh ...	RUMAH SAKIT KHUSUS GIGI DAN MULUT	B	BLU	Kementerian Lain	11	24	0
4	1102027	RS Umum Daerah Kota Subulussalam	Aceh	Kota Subulussalam	Jl. Hamzah Fansyuri (Subulussalam-Rundeng) K...	RUMAH SAKIT UMUM	C	BLUD	Pemkot	189	34	537

Kode ini membersihkan nilai pada kolom penting dengan mengubahnya menjadi string, menghapus spasi, lalu menormalkan format teks—kolom *jenis* dibuat huruf besar, sedangkan *kepemilikan* dan *provinsi* dibuat dalam format Title Case. Setelah itu, didefinisikan daftar kata kunci (HOSPITAL_ALIASES) untuk mendeteksi entitas yang benar-benar merupakan rumah sakit. Fungsi is_hospital() memeriksa apakah nilai kolom *jenis* mengandung pola terkait rumah sakit seperti “RS”, “RUMAH SAKIT”, atau “HOSPITAL”. Terakhir, dataset difilter sehingga hanya baris yang teridentifikasi sebagai rumah sakit yang disimpan.

5) Tahap Cek Missing Value

```
numeric_cols = [c for c in df.columns if pd.api.types.is_numeric_dtype(df[c])]
categorical_cols = [c for c in df.columns if c not in numeric_cols]

print("Numerik:", numeric_cols)
print("Kategorikal:", categorical_cols)

for c in numeric_cols:
    if df[c].isnull().any():
        df[c] = df[c].fillna(df[c].median())

NEW_CATEGORY_LABEL = "Tidak Diketahui"
for c in categorical_cols:
    if df[c].isnull().any():
        mode_val = df[c].mode(dropna=True)
        mode_val = mode_val.iloc[0] if not mode_val.empty else NEW_CATEGORY_LABEL
        df[c] = df[c].fillna(mode_val)

print("Cek missing setelah imputasi:")
display(df.isnull().sum())

0.0s
```

```
Numerik: ['id', 'total_tempat_tidur', 'total_layanan', 'total_tenaga_kerja']
Kategorikal: ['nama', 'propinsi', 'kab', 'alamat', 'jenis', 'kelas', 'status_blu', 'kepemilikan']
Cek missing setelah imputasi:

id                  0
nama                0
propinsi            0
kab                 0
alamat              0
jenis               0
kelas               0
status_blu          0
kepemilikan         0
total_tempat_tidur  0
total_layanan       0
total_tenaga_kerja  0
dtype: int64
```

Pemeriksaan missing value dilakukan untuk memastikan apakah ada nilai kosong pada kolom numerik maupun kategori. Tahap ini penting karena keberadaan nilai kosong dapat menyebabkan error ketika dilakukan proses analisis, scaling, atau feature engineering. Pada dataset ini, sebagian nilai sudah lengkap, namun proses pengecekan tetap wajib dilakukan.

6) Tahap One Hot Encoding

```python
LOW_FREQ_THRESHOLD = 0.01

def collapse_low_freq(series: pd.Series, threshold=LOW_FREQ_THRESHOLD, label="Lain-Lain"):
    freq = series.value_counts(dropna=False, normalize=True)
    low = freq[freq < threshold].index
    return series.apply(lambda x: label if x in low else x)

categorical_processed = df[categorical_cols].copy()
for c in categorical_processed.columns:
    if categorical_processed[c].nunique(dropna=False) > 10:
        categorical_processed[c] = collapse_low_freq(categorical_processed[c])

df_ohe = pd.get_dummies(categorical_processed, drop_first=False, dtype=int)
df_num = df[numeric_cols].copy()
df_prepared = pd.concat([df_num.reset_index(drop=True), df_ohe.reset_index(drop=True)], axis=1)

print("Dimensi setelah OHE:", df_prepared.shape)
display(df_prepared.head(5))
```

✓ 0.0s

Dimensi setelah OHE: (3151, 75)

	id	total_tempat_tidur	total_layanan	total_tenaga_kerja	nama_Lain-Lain	propinsi_Aceh	propinsi_Bali	propinsi_Banten	propinsi_Dki Jakarta	propinsi_Jambi	...	kepemilikan_Organisasi Katholik	kepemilikan_Organisasi Sosial	kepemilikan_Pemkab
0	1110053	218	36	328	1	1	0	0	0	0	...	0	0	0
1	1106014	45	15	45	1	1	0	0	0	0	...	0	0	0
2	1171110	310	77	487	1	1	0	0	0	0	...	0	0	0
3	1171163	11	24	0	1	1	0	0	0	0	...	0	0	0
4	1102027	189	34	537	1	1	0	0	0	0	...	0	0	0

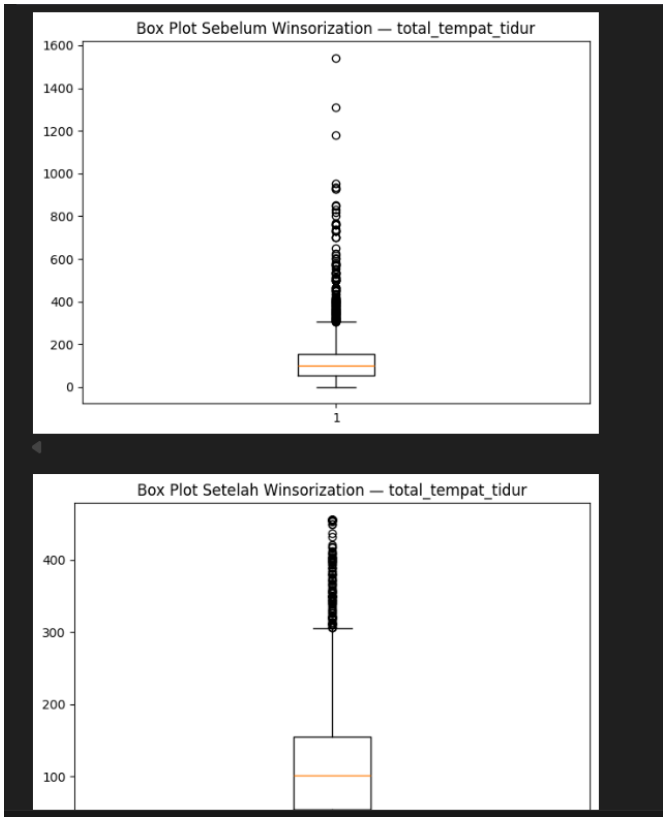
5 rows × 75 columns

Kode ini menangani preprocessing fitur kategorikal dan numerik sebelum model digunakan. Pertama, kategori yang muncul sangat sedikit (kurang dari 1% frekuensi) digabungkan ke dalam label “Lain-Lain” menggunakan fungsi collapse_low_freq(). Lalu seluruh kolom kategorikal diproses: jika jumlah

kategorinya terlalu banyak (>10), kategori frekuensi rendah disederhanakan. Setelah itu, kolom kategorikal diubah menjadi variabel numerik menggunakan One-Hot Encoding (pd.get_dummies). Kolom numerik disalin apa adanya, kemudian kedua jenis fitur numerik dan hasil OHE digabung menjadi satu dataframe yang siap dipakai untuk modelling.

7) Tahap Menangani Outlier (IQR Winsorization)

```
def winsorize_iqr(series: pd.Series, k=3.0):
    q1 = series.quantile(0.25); q3 = series.quantile(0.75)
    iqr = q3 - q1
    lower = q1 - k * iqr
    upper = q3 + k * iqr
    return series.clip(lower, upper)

key_numeric = [c for c in [COL_TT, COL_LAYANAN, COL_TK] if c in df_prepared.columns] or df_prepared.select_dtypes(include=[np.number]).columns.tolist()[:1]

for c in key_numeric:
    plt.figure()
    plt.boxplot(df_prepared[c].dropna().values)
    plt.title(f"Box Plot Sebelum Winsorization — {c}")
    plt.tight_layout()
    plt.show()

    df_prepared[c] = winsorize_iqr(df_prepared[c])

    plt.figure()
    plt.boxplot(df_prepared[c].dropna().values)
    plt.title(f"Box Plot Setelah Winsorization — {c}")
    plt.tight_layout()
    plt.show()
```



Kode ini melakukan penanganan outlier pada kolom numerik menggunakan metode Winsorization berbasis IQR. Fungsi winsorize_iqr() menghitung batas

bawah dan atas dari data berdasarkan kuartil serta meng-clip nilai yang berada di luar rentang tersebut agar tidak terlalu ekstrem. Daftar key_numeric berisi kolom numerik penting yang akan diproses. Untuk setiap kolom, kode menampilkan boxplot sebelum Winsorization sebagai visualisasi outlier, kemudian menerapkan Winsorization, dan menampilkan boxplot setelah Winsorization untuk memastikan bahwa nilai ekstrem telah terkontrol. Proses ini membantu menstabilkan distribusi data numerik sebelum digunakan dalam analisis atau pemodelan.

8) Penskalaan Fitur

```
SCALING = "minmax"
train_df, test_df = train_test_split(df_prepared, test_size=0.2, random_state=42)

scaler = MinMaxScaler() if SCALING == "minmax" else StandardScaler()
num_cols_all = train_df.select_dtypes(include=[np.number]).columns.tolist()

train_scaled = train_df.copy()
test_scaled = test_df.copy()
train_scaled[num_cols_all] = scaler.fit_transform(train_df[num_cols_all])
test_scaled[num_cols_all]  = scaler.transform(test_df[num_cols_all])

print("Train/Test:", train_scaled.shape, test_scaled.shape)
display(train_scaled.head(3))
display(test_scaled.head(3))
```

✓ 0.0s

```
Train/Test: (2520, 75) (631, 75)
```

	id	total_tempat_tidur	total_layanan	total_tenaga_kerja	nama_Lain-Lain	propinsi_Aceh	propinsi_Bali	propinsi_Banten	propinsi_Dki Jakarta	propinsi_Jambi	...	kepemilikan_Organisasi Katholik	kepemilikan_Organisasi Sosial
1295	0.266339	0.028509	0.082707	0.073567	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
2737	0.649321	0.368421	0.338346	0.141146	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
764	0.253425	0.173246	0.413534	0.277160	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0

3 rows × 75 columns

Kode ini membagi dataset terolah (df_prepared) menjadi data train dan test dengan proporsi 80:20, kemudian menerapkan normalisasi pada seluruh kolom numerik. Pemilihan metode skala ditentukan oleh variabel SCALING — jika bernilai "minmax" maka digunakan MinMaxScaler, jika tidak maka digunakan StandardScaler. Setelah itu, semua kolom numerik diambil dan dinormalkan pada data train menggunakan fit_transform(), sedangkan data test hanya menggunakan transform() agar skala konsisten. Hasil akhirnya adalah dua dataframe — train_scaled dan test_scaled — yang memiliki kolom numerik sudah terskala.

9) Tahapan Agregasi Kepemilikan dan provinsi

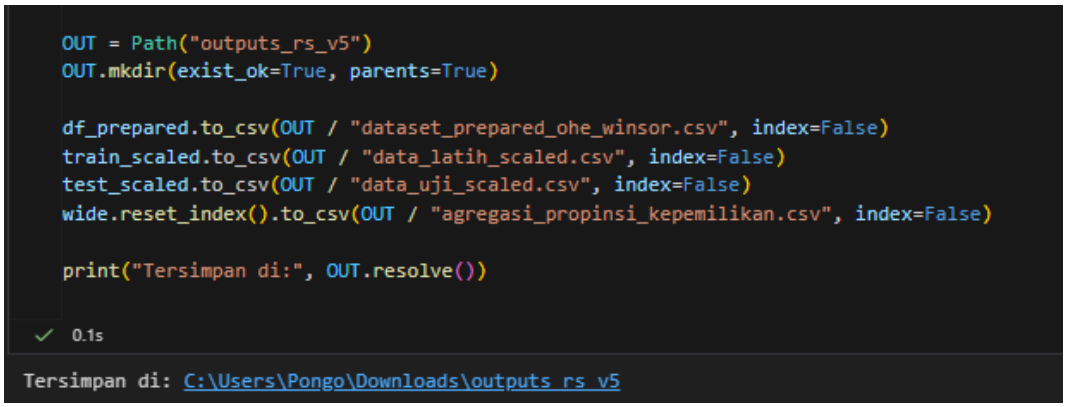
```python
def norm_owner(x: str) -> str:
    up = str(x).upper()
    if any(k in up for k in ["PEMERINTAH", "KEMENTERIAN", "PEMDA", "RSUD", "TNI", "POLRI", "NEG", "STATE", "GOV"]):
        return "Pemerintah"
    if any(k in up for k in ["SWASTA", "PRIVATE", "YAYASAN", "COMPANY", "PT "]):
        return "Swasta"
    return str(x).title()

df_agg = df.copy()
df_agg[COL_OWNER] = df_agg[COL_OWNER].map(norm_owner)

agg = df_agg.groupby([COL_PROV, COL_OWNER]).size().reset_index(name="jumlah")
wide = (agg.pivot(index=COL_PROV, columns=COL_OWNER, values="jumlah").fillna(0).rename_axis(None, axis=1))
for col in ["Pemerintah", "Swasta"]:
    if col not in wide.columns:
        wide[col] = 0
wide["Total"] = wide["Pemerintah"] + wide["Swasta"]
wide = wide.sort_values("Total", ascending=False)

display(wide.head(10))
```

✓ 0.0s

propinsi	Bumn	Kemkes	Organisasi Budha	Organisasi Hindu	Organisasi Islam	Organisasi Katholik	Organisasi Protestan	Organisasi Sosial	Pemerintah	Pemkab	Pemkot	Pemprop	Perorangan	Perusahaan	Swasta	Total
Jawa Timur	2.0	2.0	0.0	0.0	21.0	2.0	4.0	23.0	30.0	62.0	10.0	14.0	7.0	43.0	212.0	242.0
Jawa Barat	3.0	5.0	0.0	0.0	6.0	3.0	2.0	35.0	19.0	44.0	19.0	6.0	5.0	118.0	156.0	175.0
Jawa Tengah	1.0	5.0	0.0	0.0	58.0	6.0	4.0	51.0	17.0	54.0	9.0	7.0	7.0	51.0	84.0	101.0
Dki Jakarta	4.0	10.0	1.0	0.0	4.0	3.0	2.0	28.0	17.0	0.0	0.0	30.0	2.0	27.0	59.0	76.0
Banten	1.0	1.0	0.0	0.0	2.0	1.0	0.0	3.0	4.0	8.0	7.0	2.0	1.0	38.0	61.0	65.0
Sumatera Utara	7.0	1.0	0.0	0.0	0.0	4.0	2.0	39.0	11.0	33.0	8.0	5.0	6.0	52.0	34.0	45.0
Sulawesi Selatan	1.0	4.0	0.0	0.0	3.0	1.0	0.0	13.0	10.0	36.0	5.0	8.0	8.0	9.0	25.0	35.0
Bali	0.0	1.0	0.0	2.0	0.0	0.0	0.0	9.0	4.0	14.0	1.0	3.0	2.0	13.0	30.0	34.0
Riau	2.0	0.0	0.0	0.0	1.0	0.0	0.0	2.0	5.0	15.0	2.0	3.0	3.0	24.0	24.0	29.0
Aceh	2.0	0.0	0.0	0.0	0.0	0.0	0.0	11.0	9.0	22.0	4.0	3.0	2.0	7.0	19.0	28.0

Kode ini menstandarkan kategori kepemilikan rumah sakit dan kemudian membuat tabel ringkasan jumlah rumah sakit per provinsi. Fungsi norm_owner() digunakan untuk mengubah berbagai variasi penulisan kepemilikan menjadi dua kategori utama, yaitu “Pemerintah” dan “Swasta”, berdasarkan kata kunci seperti PEMERINTAH, RSUD, TNI, atau PRIVATE, YAYASAN, COMPANY, dan sebagainya. Setelah kategori dibersihkan, dataset digabungkan kembali dan dilakukan groupby berdasarkan provinsi dan kepemilikan untuk menghitung jumlah rumah sakit pada tiap kombinasi. Hasilnya dipivot sehingga setiap provinsi memiliki dua kolom: jumlah rumah sakit Pemerintah dan Swasta. Nilai kosong diisi nol, lalu dibuat kolom “Total” sebagai penjumlahan keduanya. Terakhir, tabel diurutkan berdasarkan total rumah sakit dari yang tertinggi untuk memudahkan analisis sebaran fasilitas kesehatan antar provinsi.

10) Tahapan Simpan Output



```
OUT = Path("outputs_rs_v5")
OUT.mkdir(exist_ok=True, parents=True)

df_prepared.to_csv(OUT / "dataset_prepared_ohe_winsor.csv", index=False)
train_scaled.to_csv(OUT / "data_latih_scaled.csv", index=False)
test_scaled.to_csv(OUT / "data_uji_scaled.csv", index=False)
wide.reset_index().to_csv(OUT / "agregasi_propinsi_kepemilikan.csv", index=False)

print("Tersimpan di:", OUT.resolve())
```

```
✓ 0.1s
Tersimpan di: C:\Users\Pongo\Downloads\outputs_rs_v5
```

Kode ini membuat folder output bernama outputs_rs_v5 (jika belum ada) dan kemudian menyimpan beberapa hasil preprocessing ke dalam file CSV. Dataset yang sudah melalui proses OHE dan Winsorization disimpan sebagai *dataset_prepared_ohe_winsor.csv*, sedangkan data latih dan data uji yang sudah dinormalisasi masing-masing disimpan sebagai *data_latih_scaled.csv* dan *data_uji_scaled.csv*. Selain itu, tabel agregasi jumlah rumah sakit berdasarkan provinsi dan kepemilikan juga diekspor sebagai *agregasi_propinsi_kepemilikan.csv.*