In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("train.csv")
```

Data Understanding

In [3]:
```
df.head()
```

Out[3]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	Unite Stat
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	Unite Stat
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	Unite Stat
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite Stat
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite Stat

In [4]:
```
df.tail()
```

Out[4]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Co
9795	9796	CA-2017-125920	21/05/2017	28/05/2017	Standard Class	SH-19975	Sally Hughsby	Corporate	U
9796	9797	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	U
9797	9798	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	U
9798	9799	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	U
9799	9800	CA-2016-128608	12/01/2016	17/01/2016	Standard Class	CS-12490	Cindy Schnelling	Corporate	U

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Row ID         9800 non-null   int64  
 1   Order ID       9800 non-null   object 
 2   Order Date     9800 non-null   object 
 3   Ship Date      9800 non-null   object 
 4   Ship Mode      9800 non-null   object 
 5   Customer ID    9800 non-null   object 
 6   Customer Name  9800 non-null   object 
 7   Segment        9800 non-null   object 
 8   Country        9800 non-null   object 
 9   City           9800 non-null   object 
 10  State          9800 non-null   object 
 11  Postal Code    9789 non-null   float64
 12  Region         9800 non-null   object 
 13  Product ID     9800 non-null   object 
 14  Category       9800 non-null   object 
 15  Sub-Category   9800 non-null   object 
 16  Product Name   9800 non-null   object 
 17  Sales          9800 non-null   float64
dtypes: float64(2), int64(1), object(15)
memory usage: 1.3+ MB
```

In [6]:
```python
# Cek Missing Value
missing_values = df.isnull().sum()
print("Jumlah nilai hilang per kolom:")
print(missing_values[missing_values > 0])
if missing_values.sum() == 0:
    print("-> Tidak ada nilai hilang dalam dataset.")
```

```
Jumlah nilai hilang per kolom:
Postal Code    11
dtype: int64
```

In [7]:
```python
print(df.describe().round(2))
```

```
         Row ID  Postal Code     Sales
count   9800.00      9789.00   9800.00
mean    4900.50     55273.32    230.77
std     2829.16     32041.22    626.65
min        1.00      1040.00      0.44
25%     2450.75     23223.00     17.25
50%     4900.50     58103.00     54.49
75%     7350.25     90008.00    210.60
max     9800.00     99301.00  22638.48
```

In [8]:
```python
# Eksplorasi Kolom Kategori
# Menghitung frekuensi unik di kolom 'Region'
print("\nDistribusi Data per Wilayah (Region):")
print(df['Region'].value_counts())

# Menghitung frekuensi unik di kolom 'Segment'
print("\nDistribusi Data per Segmen Pelanggan (Segment):")
print(df['Segment'].value_counts())

# Menghitung frekuensi unik di kolom 'Category'
print("\nDistribusi Data per Kategori Produk (Category):")
print(df['Category'].value_counts())

# Menghitung frekuensi unik di kolom 'Ship Mode'
print("\nDistribusi Data per Mode Pengiriman (Ship Class):")
print(df['Ship Mode'].value_counts())
```

```
Distribusi Data per Wilayah (Region):
Region
West       3140
East       2785
Central    2277
South      1598
Name: count, dtype: int64

Distribusi Data per Segmen Pelanggan (Segment):
Segment
Consumer       5101
Corporate      2953
Home Office    1746
Name: count, dtype: int64

Distribusi Data per Kategori Produk (Category):
Category
Office Supplies    5909
Furniture          2078
Technology         1813
Name: count, dtype: int64

Distribusi Data per Mode Pengiriman (Ship Class):
Ship Mode
Standard Class    5859
Second Class      1902
First Class       1501
Same Day           538
Name: count, dtype: int64
```

Ringkasan Temuan dari Tahap Pemahaman Data

Setelah melakukan eksplorasi awal pada dataset penjualan Superstore, beberapa karakteristik dan wawasan kunci berhasil diidentifikasi.

Berikut adalah temuan-temuan utamanya:

1. Kualitas Data Sangat Baik

Dataset berada dalam kondisi bersih dan lengkap. Analisis menunjukkan tidak ada nilai yang hilang pada seluruh 9.800 baris data. Hal ini menyederhanakan proses persiapan data dan mempermudah fokus terhadap fase selanjutnya.

2. Distribusi Penjualan yang Miring (Skewed)

Data penjualan menunjukkan adanya beberapa transaksi dengan nilai yang sangat tinggi. Ini terbukti dari nilai rata-rata penjualan ($$230.77) yang jauh lebih tinggi dibandingkan nilai mediannya ($54.49).

Apa dampaknya?

- Sebagian besar transaksi sebenarnya bernilai relatif kecil. Adanya transaksi bernilai besar ini menjadi sinyal untuk melakukan analisa lebih lanjut: Produk, pelanggan, atau wilayah mana yang menjadi pendorong transaksi bernilai tinggi tersebut?

3. Komposisi Bisnis Teridentifikasi

Segmen Pelanggan: Didominasi oleh segmen Consumer, yang mencakup lebih dari 50% total transaksi.

Wilayah Geografis: Wilayah West tercatat sebagai wilayah dengan jumlah transaksi terbanyak.

Kategori Produk: Office Supplies merupakan kategori dengan frekuensi penjualan tertinggi.

Implikasi: Ini memberikan gambaran awal tentang di mana fokus bisnis saat ini. Pertanyaan selanjutnya adalah apakah volume transaksi yang tinggi ini sejalan dengan pendapatan dan profitabilitas yang tinggi.

4. Kebutuhan Persiapan Data untuk Analisis Waktu

Kolom Order Date dan Ship Date saat ini masih terdeteksi sebagai tipe data teks (object).

Apa dampaknya?

- Untuk menjawab pertanyaan bisnis mengenai tren penjualan dari waktu ke waktu (misalnya, pertumbuhan tahunan atau pola musiman bulanan), kedua kolom ini wajib diubah menjadi format tanggal.

Kesimpulan pada Tahap Ini:

Secara keseluruhan, tahap pemahaman data ini menegaskan bahwa dataset ini memiliki informasi dan kualitas yang baik untuk dianalisis. Langkah selanjutnya adalah Persiapan Data, dengan fokus utama pada konversi tipe data tanggal untuk membuka potensi analisis tren.

Data Preparation

In [9]:
```
# Mengubah kolom tanggal menjadi format datetime
# Format '%d/%m/%Y' disesuaikan dengan format tanggal di CSV (contoh: 08/11/2017)
df['Order Date'] = pd.to_datetime(df['Order Date'], format='%d/%m/%Y')
df['Ship Date'] = pd.to_datetime(df['Ship Date'], format='%d/%m/%Y')

# Menampilkan tipe data setelah konversi untuk verifikasi
print("\nTipe Data Setelah Konversi")
print(df[['Order Date', 'Ship Date']].dtypes)
```

```
Tipe Data Setelah Konversi
Order Date    datetime64[ns]
Ship Date     datetime64[ns]
dtype: object
```

In [10]:
```
# Membuat kolom baru (Feature Engineering) dari 'Order Date'
df['Order_Year'] = df['Order Date'].dt.year
df['Order_Month'] = df['Order Date'].dt.month
df['Order_Month_Name'] = df['Order Date'].dt.month_name()

print("\nKolom baru ('Order_Year', 'Order_Month', 'Order_Month_Name') berhasil dibu
print("Menampilkan 5 baris pertama dari kolom baru:")
print(df[['Order Date', 'Order_Year', 'Order_Month', 'Order_Month_Name']].head())
```

```
Kolom baru ('Order_Year', 'Order_Month', 'Order_Month_Name') berhasil dibuat.
Menampilkan 5 baris pertama dari kolom baru:
  Order Date  Order_Year  Order_Month Order_Month_Name
0 2017-11-08        2017           11         November
1 2017-11-08        2017           11         November
2 2017-06-12        2017            6             June
3 2016-10-11        2016           10          October
4 2016-10-11        2016           10          October
```

In [11]:
```
# Membuat kolom baru untuk durasi pengiriman
df['Lama_Pengiriman'] = (df['Ship Date'] - df['Order Date']).dt.days

print("\nKolom baru 'Lama_Pengiriman' (dalam hari) berhasil dibuat.")
print("Menampilkan 5 baris pertama dari kolom terkait pengiriman:")
print(df[['Order Date', 'Ship Date', 'Lama_Pengiriman']].head())
```

```
Kolom baru 'Lama_Pengiriman' (dalam hari) berhasil dibuat.
Menampilkan 5 baris pertama dari kolom terkait pengiriman:
  Order Date  Ship Date  Lama_Pengiriman
0 2017-11-08 2017-11-11                3
1 2017-11-08 2017-11-11                3
2 2017-06-12 2017-06-16                4
3 2016-10-11 2016-10-18                7
4 2016-10-11 2016-10-18                7
```

In [12]:
```
# Menyimpan dataframe yang sudah bersih ke file CSV baru
df.to_csv('data_bersih.csv', index=False)
print("\nProses persiapan data selesai. Hasil disimpan di 'Train_Cleaned.csv'")
```

```
Proses persiapan data selesai. Hasil disimpan di 'Train_Cleaned.csv'
```

Ringkasan Temuan Dari Tahap Data Preparation

Berdasarkan temuan dari tahap pemahaman data, langkah-langkah persiapan telah dilakukan untuk membersihkan dan memperkaya dataset. Proses ini memastikan data siap untuk dianalisis dan divisualisasikan, terutama untuk menjawab pertanyaan terkait tren dan efisiensi operasional.

Berikut adalah langkah-langkah persiapan yang telah diselesaikan:

1. Konversi Tipe Data Kolom Tanggal

- **Tindakan:** Kolom `Order Date` dan `Ship Date` yang semula berformat teks (`object`) telah berhasil dikonversi menjadi format tanggal (`datetime`).

- **Tujuan:** Ini adalah langkah fundamental yang memungkinkan dilakukannya analisis berbasis waktu, seperti tren penjualan bulanan atau tahunan.

2. Rekayasa Fitur (Feature Engineering) untuk Analisis Waktu

- **Tindakan:** Tiga kolom baru telah dibuat dari `Order Date`:
  - `Order_Year`: Menyimpan informasi tahun transaksi.
  - `Order_Month`: Menyimpan nomor bulan (1-12).
  - `Order_Month_Name`: Menyimpan nama bulan (e.g., "January").
- **Tujuan:** Kolom-kolom ini akan sangat memudahkan proses agregasi dan visualisasi data penjualan berdasarkan periode waktu tertentu.

3. Rekayasa Fitur untuk Analisis Operasional

- **Tindakan:** Sebuah kolom baru bernama `Lama_Pengiriman` telah ditambahkan. Kolom ini menghitung selisih hari antara `Ship Date` dan `Order Date`.
- **Tujuan:** Menambahkan dimensi baru untuk analisis efisiensi logistik, hal ini bertujuan agar datasetnya dapat digunakan untuk menganalisis rata-rata waktu pengiriman per wilayah, segmen, atau mode pengiriman.

4. Penyimpanan Data Bersih

- **Tindakan:** Seluruh hasil dari proses di atas telah disimpan ke dalam sebuah file baru bernama `data_bersih.csv`.
- **Tujuan:** Memisahkan data yang sudah diproses dari data mentah asli (`train.csv`).

Kesimpulan Tahap Ini:

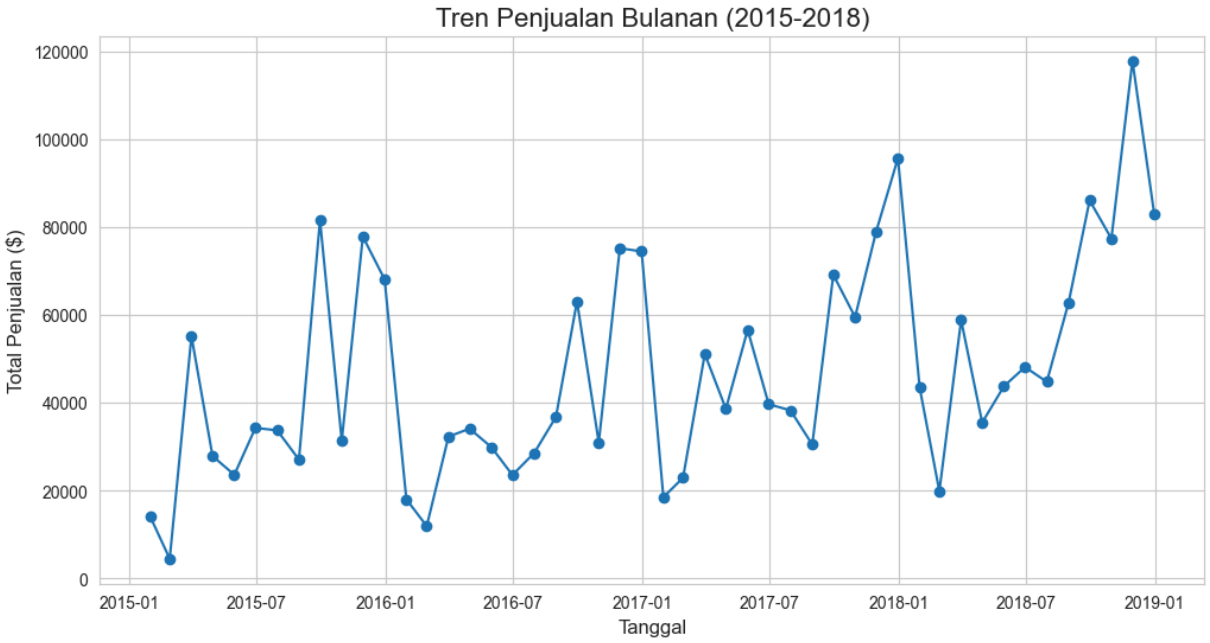
Dataset kini telah sepenuhnya siap untuk tahap **Analisis Data Eksploratif (EDA)**. Semua kolom memiliki tipe data yang sesuai, dan penambahan kolom-kolom baru telah memperkaya dataset dengan informasi yang akan mempermudah penggalian wawasan.

VISUALISASI

In [19]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (12, 6)

# Tren Penjualan Keseluruhan dari Waktu ke Waktu
print("\nMembuat visualisasi tren penjualan bulanan...")

df_time_series = df.set_index('Order Date')
monthly_sales = df_time_series['Sales'].resample('M').sum()

# Membuat plot
plt.plot(monthly_sales.index, monthly_sales.values, marker='o', linestyle='-')
```

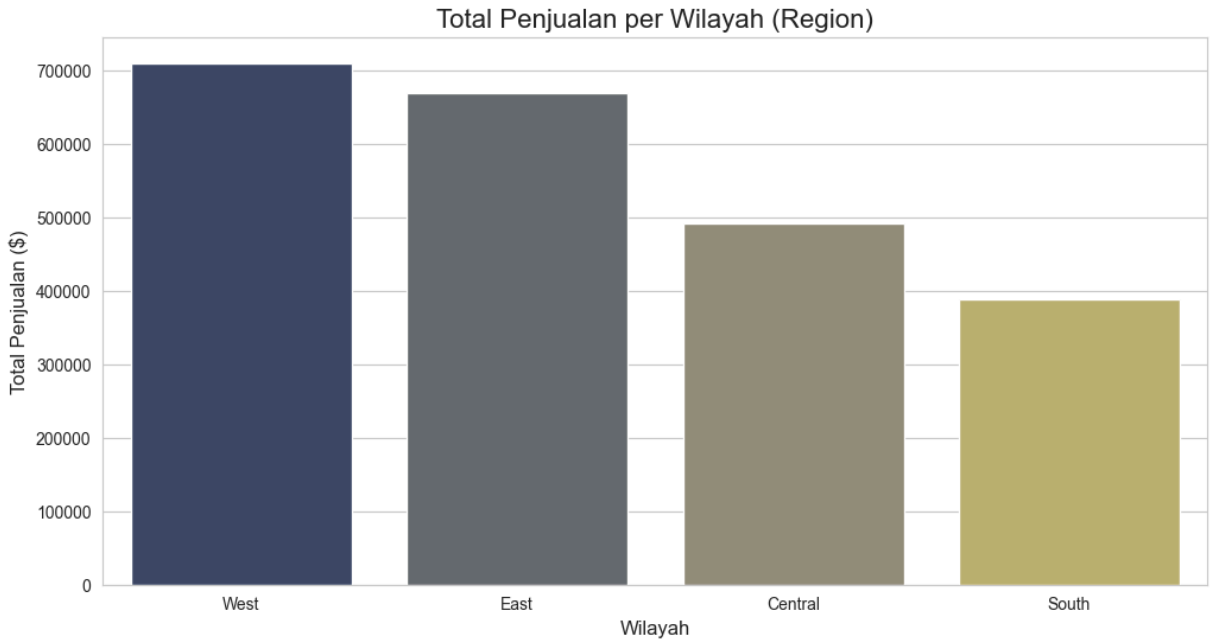
```
plt.title('Tren Penjualan Bulanan (2015-2018)', fontsize=16)
plt.xlabel('Tanggal', fontsize=12)
plt.ylabel('Total Penjualan ($)', fontsize=12)
plt.grid(True)

plt.savefig('tren_penjualan_bulanan.png', dpi=300)
print("Plot 'tren_penjualan_bulanan.png' berhasil disimpan.")
plt.show()
```

```
Membuat visualisasi tren penjualan bulanan...
Plot 'tren_penjualan_bulanan.png' berhasil disimpan.
```



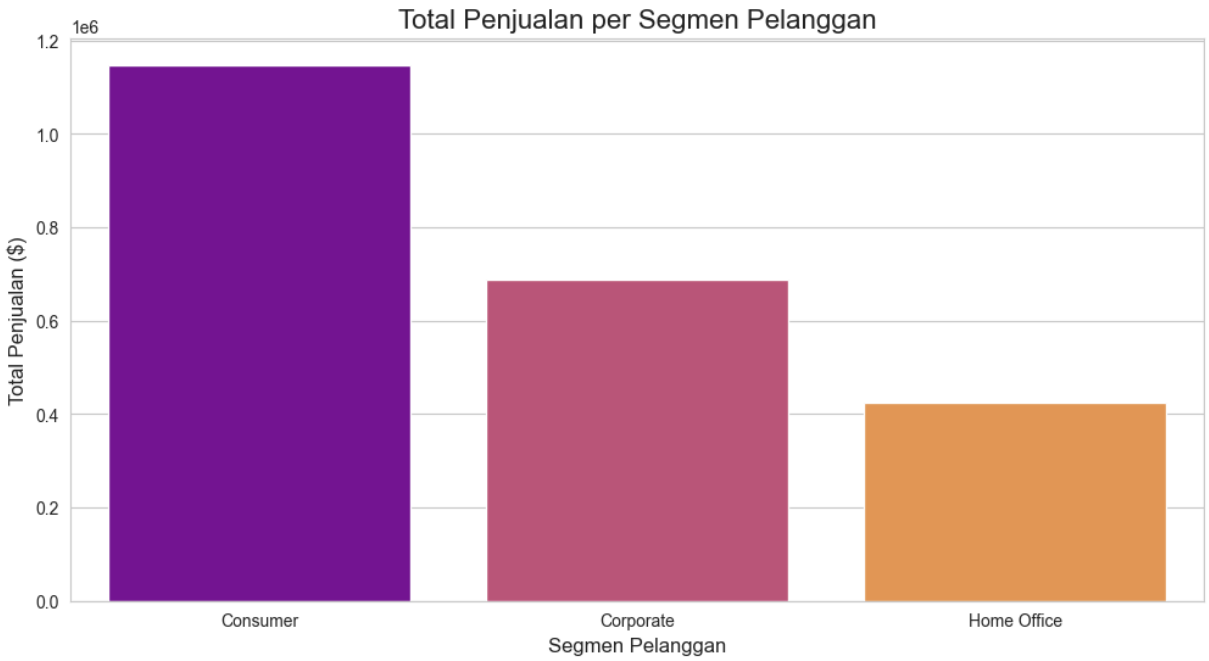
In [20]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

# Analisis Penjualan Geografis Berdasarkan Wilayah
print("\nMembuat visualisasi penjualan per wilayah...")

# Mengelompokkan data berdasarkan Wilayah dan menjumlahkan penjualan
region_sales = df.groupby('Region')['Sales'].sum().sort_values(ascending=False)

# Membuat bar plot
sns.barplot(x=region_sales.index, y=region_sales.values, palette='cividis')
plt.title('Total Penjualan per Wilayah (Region)', fontsize=16)
plt.xlabel('Wilayah', fontsize=12)
plt.ylabel('Total Penjualan ($)', fontsize=12)

plt.savefig('penjualan_per_wilayah.png', dpi=300)
print("Plot 'penjualan_per_wilayah.png' berhasil disimpan.")
plt.show()
```

```
Membuat visualisasi penjualan per wilayah...
Plot 'penjualan_per_wilayah.png' berhasil disimpan.
```



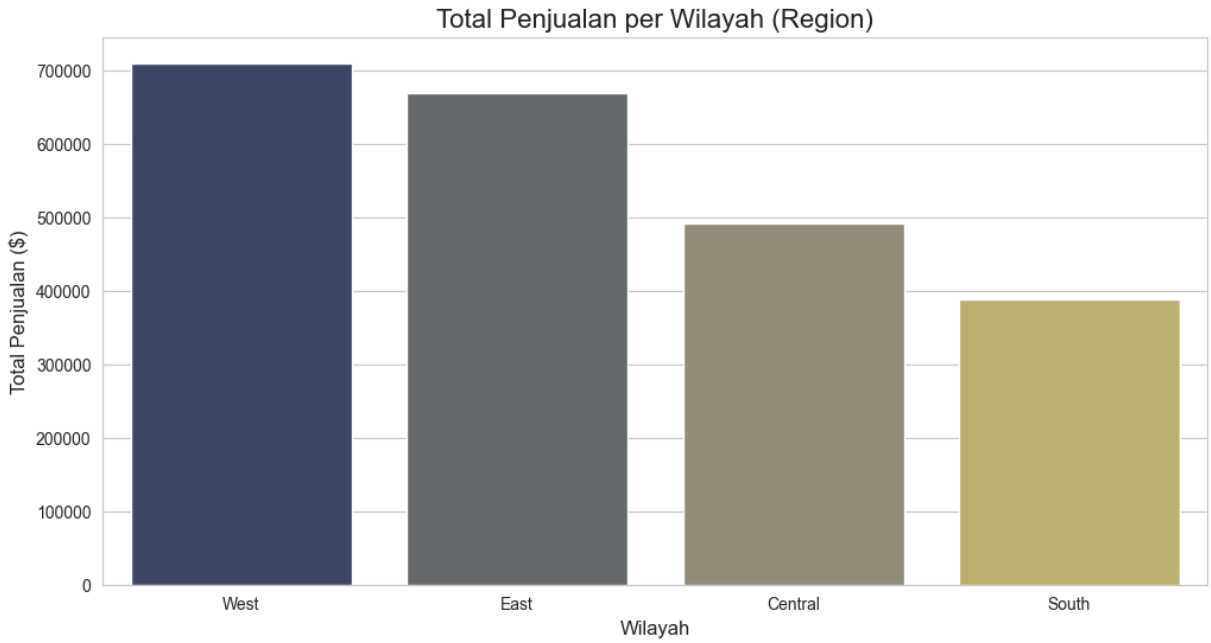
In [22]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)

# Penjualan Berdasarkan Segmen Pelanggan
print("\nMembuat visualisasi penjualan per segmen pelanggan...")

# Mengelompokkan data berdasarkan Segmen dan menjumlahkan penjualan
segment_sales = df.groupby('Segment')['Sales'].sum().sort_values(ascending=False)

# Membuat bar plot
sns.barplot(x=segment_sales.index, y=segment_sales.values, palette='plasma')
plt.title('Total Penjualan per Segmen Pelanggan', fontsize=16)
plt.xlabel('Segmen Pelanggan', fontsize=12)
plt.ylabel('Total Penjualan ($)', fontsize=12)

plt.savefig('penjualan_per_segmen.png', dpi=300)
print("Plot 'penjualan_per_segmen.png' berhasil disimpan.")
plt.show()
```

```
Membuat visualisasi penjualan per segmen pelanggan...
Plot 'penjualan_per_segmen.png' berhasil disimpan.
```



In [23]:
```python
# Analisis Penjualan Geografis Berdasarkan Wilayah
print("\nMembuat visualisasi penjualan per wilayah...")

# Mengelompokkan data berdasarkan Wilayah dan menjumlahkan penjualan
region_sales = df.groupby('Region')['Sales'].sum().sort_values(ascending=False)

# Membuat bar plot
sns.barplot(x=region_sales.index, y=region_sales.values, palette='cividis')
plt.title('Total Penjualan per Wilayah (Region)', fontsize=16)
plt.xlabel('Wilayah', fontsize=12)
plt.ylabel('Total Penjualan ($)', fontsize=12)

# Menyimpan plot sebagai file gambar
plt.savefig('penjualan_per_wilayah.png', dpi=300)
print("Plot 'penjualan_per_wilayah.png' berhasil disimpan.")
plt.show()


import warnings

# Menonaktifkan FutureWarning
warnings.simplefilter(action='ignore', category=FutureWarning)
```

```
Membuat visualisasi penjualan per wilayah...
Plot 'penjualan_per_wilayah.png' berhasil disimpan.
```



Ringkasan Tahap Analisis Data Eksploratif (EDA)

Berikut adalah temuan utama dari setiap analisis yang dilakukan:

1. Analisis Tren Penjualan Berdasarkan Waktu

- **Wawasan:**
  - **Pertumbuhan Positif:** Terdapat tren pertumbuhan penjualan yang konsisten dari tahun 2015 hingga 2018, menandakan bisnis berada dalam kondisi sehat.
  - **Pola Musiman yang Jelas:** Penjualan secara rutin mencapai puncaknya di akhir tahun (November-Desember) dan cenderung melambat di awal kuartal pertama (Januari-Februari).
- **Implikasi Bisnis:**
  - Perusahaan dapat merancang strategi pemasaran yang lebih agresif dan memastikan ketersediaan stok yang optimal menjelang akhir tahun.
  - Periode awal tahun dapat dimanfaatkan untuk program retensi pelanggan atau promosi khusus guna menstabilkan pendapatan.

2. Analisis Kinerja Berdasarkan Kategori Produk

- **Wawasan:**
  - **Pendorong Pendapatan:** Kategori `Technology` merupakan penyumbang pendapatan terbesar, diikuti oleh `Furniture`.
  - **Paradoks Volume vs. Nilai:** Meskipun `Office Supplies` memiliki frekuensi transaksi tertinggi, total nilai penjualannya adalah yang terendah di antara ketiganya.
- **Implikasi Bisnis:**

- Fokus strategis harus diberikan pada produk `Technology` karena nilai jualnya yang tinggi.
- Untuk `Office Supplies`, peluang terletak pada peningkatan volume per transaksi (penjualan B2B, penawaran paket) atau strategi *cross-selling* dengan produk bernilai lebih tinggi.

3. Analisis Kontribusi Berdasarkan Segmen Pelanggan

- **Wawasan:**
  - **Pasar Utama:** Segmen `Consumer` (perorangan) mendominasi penjualan dan merupakan pasar terbesar bagi perusahaan.
  - **Kontributor Sekunder:** Segmen `Corporate` menempati posisi kedua dengan kontribusi yang kuat, diikuti oleh `Home Office`.
- **Implikasi Bisnis:**
  - Strategi pemasaran dan loyalitas pelanggan harus memprioritaskan segmen `Consumer`.
  - Segmen `Corporate` menyimpan potensi pertumbuhan yang signifikan dan dapat ditargetkan dengan program penjualan B2B atau kontrak bisnis.

4. Analisis Kinerja Geografis Berdasarkan Wilayah

- **Wawasan:**
  - **Wilayah Unggulan:** Wilayah `West` dan `East` adalah dua pasar terkuat yang menjadi penopang utama pendapatan perusahaan.
  - **Area Potensial:** Wilayah `South` menunjukkan kinerja penjualan yang paling rendah, menjadikannya area potensial untuk ekspansi dan pertumbuhan di masa depan.
- **Implikasi Bisnis:**
  - Alokasi sumber daya, logistik, dan kampanye pemasaran regional harus difokuskan pada wilayah `West` dan `East`.
  - Perlu dilakukan analisis lebih lanjut untuk memahami tantangan di wilayah `South` dan merancang strategi penetrasi pasar yang sesuai.

Kesimpulan Tahap Ini:

Analisis data eksploratif ini telah memberikan gambaran tentang kinerja penjualan Superstore. Wawasan yang didapat akan menjadi fondasi yang kuat untuk menyusun rekomendasi bisnis yang strategis dan berbasis data pada tahap akhir proyek.

In [ ]: