In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
import matplotlib.pyplot as plt
from pandas import read_csv
```

In [2]:
```
# Load dataset
df =read_csv('house_price_regression_dataset.csv')
# Tampilkan 2 kolom dari dataset
df[['Square_Footage', 'Lot_Size', 'House_Price']]
```

Out[2]:

	Square_Footage	Lot_Size	House_Price
0	1360	0.599637	2.623829e+05
1	4272	4.753014	9.852609e+05
2	3592	3.634823	7.779774e+05
3	966	2.730667	2.296989e+05
4	4926	4.699073	1.041741e+06
...	...	...	...
995	3261	2.165110	7.014940e+05
996	3179	2.977123	6.837232e+05
997	2606	4.055067	5.720240e+05
998	4723	1.930921	9.648653e+05
999	3268	3.108790	7.425993e+05

1000 rows × 3 columns

In [3]:
```
# Memisahkan variabel bebas dan varibel terikat
x = df.iloc[:,[0,3]]
y = df['House_Price']
print(x)
```

```
     Square_Footage  Year_Built
0              1360        1981
1              4272        2016
2              3592        2016
3               966        1977
4              4926        1993
..              ...         ...
995            3261        1978
996            3179        1999
997            2606        1962
998            4723        1950
999            3268        1983

[1000 rows x 2 columns]
```

In [4]:
```
# Bagi data mennjadi training dan testing (80% training dan 220% testing)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

In [5]:
```
# Model Regression Linear
linear_regression = LinearRegression()
linear_regression.fit(x_train, y_train)
y_pred_regression= linear_regression.predict(x_test)
```

In [6]:
```
# Model Decision Tree
decision_tree = DecisionTreeRegressor(random_state=42)
decision_tree.fit(x_train, y_train)
y_pred_tree = decision_tree.predict(x_test)
```

In [12]:
```
# Model Random Forest
random_forest = RandomForestRegressor(random_state=42, n_estimators=100)
random_forest.fit(x_train, y_train)
y_pred_forest = random_forest.predict(x_test)
```

In [9]:
```
# Evaluasi menggunakan MSE dan R-squared
mse_regression = mean_squared_error(y_test, y_pred_regression)
mse_tree = mean_squared_error(y_test, y_pred_tree)
mse_forest = mean_squared_error(y_test, y_pred_forest)

r2_regression = r2_score(y_test, y_pred_regression)
r2_tree = r2_score(y_test, y_pred_tree)
r2_forest = r2_score(y_test, y_pred_forest)
```

In [10]:
```
# Print Hasil Evaluasi
print(f'Linear Regression MSE: {mse_regression}, R^2: {r2_regression}')
print(f'Decision Tree MSE: {mse_tree}, R^2: {r2_tree}')
print(f'Random Forest MSE: {mse_forest}, R^2: {r2_forest}')
```

```
Linear Regression MSE: 807292374.0718881, R^2: 0.9874758503243062
Decision Tree MSE: 1496904797.664347, R^2: 0.9767773605470191
Random Forest MSE: 1076887000.0195258, R^2: 0.9832934208160222
```

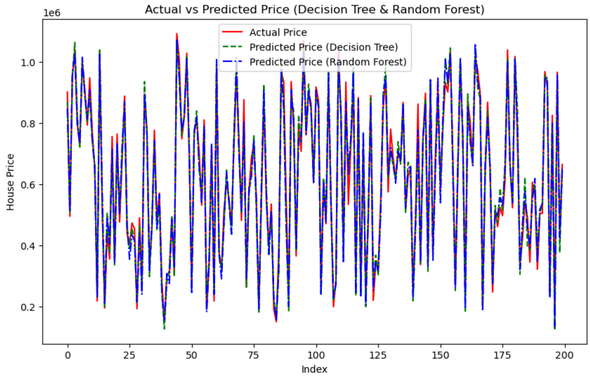
In [16]:
```
# Visualisasi hasil prediksi
plt.figure(figsize=(10, 6))

# Plot data asli (Actual)
plt.plot(range(len(y_test)), y_test, color='red', label='Actual Price', linestyle='-')

# Plot prediksi dari Decision Tree
plt.plot(range(len(y_pred_tree)), y_pred_tree, color='green', label='Predicted Price (Decision Tree)', linestyle='--')

# Plot prediksi dari Random Forest
plt.plot(range(len(y_pred_forest)), y_pred_forest, color='blue', label='Predicted Price (Random Forest)', linestyle='-.')

plt.title('Actual vs Predicted Price (Decision Tree & Random Forest)')
plt.xlabel('Index')
plt.ylabel('House Price')
plt.legend()
plt.show()
```



In [15]:
```
# Model Random Forest
random_forest = RandomForestRegressor(random_state=42, n_estimators=70)
random_forest.fit(x_train, y_train)
y_pred_forest = random_forest.predict(x_test)

# Evaluasi menggunakan MSE dan R-squared
mse_forest = mean_squared_error(y_test, y_pred_forest)
r2_forest = r2_score(y_test, y_pred_forest)

# Print Hasil Evaluasi
print(f'Random Forest MSE: {mse_forest}, R^2: {r2_forest}')
```

```
Random Forest MSE: 1086132634.9873526, R^2: 0.9831499861448884
```

1.buatkan model prediksi harga dengan memilih dua fitur,dan lakukan prediksi dengan menggunakan dataset house_price_regression.csv dan bandingkan performa dari kedua model tersebut. jawaban:

- Decision Tree MSE: 1496904797.664347, R^2: 0.9767773605470191
- Random Forest MSE: 1076887000.0195258, R^2: 0.9832934208160222

> perbandingan:

- MSE: Model Random Forest memiliki MSE yang lebih rendah, yang menunjukkan bahwa prediksi yang dihasilkan lebih dekat dengan nilai sebenarnya dibandingkan dengan Decision Tree.
- R^2: Random Forest juga memiliki nilai R^2 yang lebih tinggi, menunjukkan bahwa model ini menjelaskan proporsi varians dalam variabel dependen lebih baik dibandingkan Decision Tree.

> Kesimpulanya: Model Random Forest menunjukkan kinerja yang lebih baik dibandingkan dengan Decision Tree dalam hal akurasi prediktif (MSE yang lebih rendah) dan kekuatan penjelasan (R^2 yang lebih tinggi).

2.ubah jumlah tree pada random forest dan bandingkan performa modelnya. jawaban:

- Random Forest MSE: 1076887000.0195258, R^2: 0.9832934208160222
- Random Forest MSE: 1086132634.9873526, R^2: 0.9831499861448884

> perbandingan:

- MSE:Model pertama memiliki MSE yang lebih rendah dibandingkan model kedua, yang menunjukkan bahwa prediksi model pertama lebih akurat.
- R^2:R^2 dari model pertama juga sedikit lebih tinggi dibandingkan model kedua. Ini menunjukkan bahwa model pertama dapat menjelaskan variasi dalam data lebih baik dibandingkan model kedua.

> Kesimpulannya: Meningkatkan jumlah pohon dalam Random Forest tidak selalu menjamin peningkatan performa. Dalam hal ini, meskipun kedua model memiliki R^2 yang sangat mendekati satu (yang menunjukkan model yang baik), model pertama menunjukkan performa yang sedikit lebih baik dalam hal MSE dan R^2. Ini menunjukkan bahwa setelah titik tertentu, menambah jumlah pohon tidak selalu memberikan manfaat yang signifikan dalam meningkatkan akurasi model.

3.bandingkan dengan regresi tugas pertama jawaban:

- Decision Tree MSE: 1496904797.664347, R^2: 0.9767773605470191
- Random Forest MSE: 1076887000.0195258, R^2: 0.9832934208160222
- Random Forest MSE: 1086132634.9873526, R^2: 0.9831499861448884

1. Linear Regression MSE: 807,292,374.07 R^2: 0.9875
2. Decision Tree MSE: 1,496,904,797.66 R^2: 0.9768
3. Random Forest MSE: 1,076,887,000.02 R^2: 0.9833

> Perbandingan:

- MSE: Linear Regression memiliki MSE terendah (807,292,374.07), menunjukkan akurasi prediksi terbaik di antara ketiga model. Random Forest memiliki MSE yang lebih rendah (1,076,887,000.02) dibandingkan Decision Tree (1,496,904,797.66), menunjukkan bahwa Random Forest lebih baik dalam akurasi daripada Decision Tree.
- R^2: Linear Regression memiliki R^2 tertinggi (0.9875), menunjukkan bahwa model ini menjelaskan variabilitas data dengan sangat baik. Random Forest (0.9833) juga menunjukkan kinerja yang baik, tetapi sedikit di bawah Linear Regression. Decision Tree memiliki R^2 terendah (0.9768), yang berarti kurang efektif dalam menjelaskan variabilitas data dibandingkan dengan dua model lainnya.

> Kesimpulannya: Linear Regression adalah model yang paling efektif dalam hal akurasi prediksi, baik dari segi MSE maupun R^2. Random Forest lebih baik dibandingkan Decision Tree, tetapi tidak sebaik Linear Regression. Secara keseluruhan, Linear Regression memberikan performa terbaik untuk dataset ini.