

**PRAKTIKUM PEMODELAN STATISTIKA**  
**MODUL 6**



**Disusun oleh :**

**Nama : Fidelia Ping**  
**NIM : 245410012**  
**Kelas : Informatika 1**

**PROGRAM STUDI INFORMATIKA**  
**PROGRAM SARJANA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**  
**YOGYAKARTA**  
**2025**

## MODUL 6

### REGRESI NON LINIER

#### A. TUJUAN PRAKTIKUM

1. Memahami konsep dasar regresi Non Linier.
2. Melakukan analisis regresi non linier menggunakan R

#### B. PEMBAHASAN LISTING

##### PRAKTIK

Kasus 1 :

Hubungan antara Pengalaman Kerja dan Gaji

Analisis hubungan non-linier antara pengalaman kerja (dalam tahun) dengan gaji karyawan (dalam juta rupiah) menggunakan regresi kuadratik

##### Langkah 1: Persiapan Data

```
# --- Langkah 1: Persiapan Data ---
# Membuat dataset
pengalaman <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)
gaji <- c(8, 10, 12, 15, 18, 22, 25, 28, 30, 32, 33, 34, 34.5, 35, 35)

# Membuat data frame
data_gaji <- data.frame(pengalaman, gaji)
print("DATASET:")
print(data_gaji)
```

**Pembahasan :** Kode pada langkah ini digunakan untuk melakukan **persiapan data** dengan membuat dataset sederhana yang menggambarkan hubungan antara **pengalaman kerja** dan **gaji**. Variabel pengalaman berisi data lama kerja dalam satuan tahun, sedangkan variabel gaji berisi besaran gaji yang diterima. Kedua variabel tersebut kemudian digabungkan ke dalam sebuah data frame bernama `data_gaji` menggunakan fungsi `data.frame()`. Selanjutnya, fungsi `print()` digunakan untuk menampilkan judul dan isi dataset agar data yang telah dibuat dapat dilihat dan diverifikasi sebelum digunakan pada tahap analisis selanjutnya.

##### Output

```
*** [1] "DATASET:"
      pengalaman gaji
1             1  8.0
2             2 10.0
3             3 12.0
4             4 15.0
5             5 18.0
6             6 22.0
7             7 25.0
8             8 28.0
9             9 30.0
10            10 32.0
11            11 33.0
12            12 34.0
13            13 34.5
14            14 35.0
15            15 35.0
```

**Pembahasan Output :** Output yang dihasilkan menampilkan dataset `data_gaji` yang terdiri dari **15 observasi** dengan dua variabel, yaitu pengalaman dan gaji. Data tersebut menunjukkan adanya kecenderungan bahwa gaji meningkat seiring dengan bertambahnya pengalaman kerja, meskipun pada pengalaman tertentu peningkatan gaji mulai melambat. Output ini menandakan bahwa data telah tersusun dengan baik dan siap

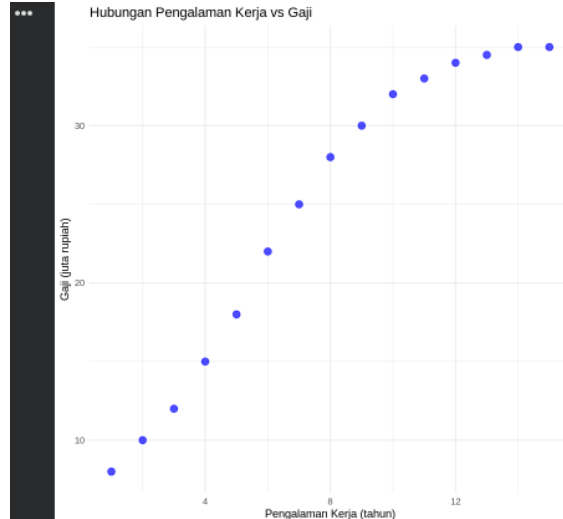
digunakan untuk analisis lanjutan, seperti pembuatan model regresi linier sederhana untuk melihat pengaruh pengalaman kerja terhadap gaji.

## Langkah 2: Eksplorasi Data dan Visualisasi Awal

```
# --- Langkah 2: Eksplorasi Data dan Visualisasi Awal ---  
# Scatter plot untuk melihat pola hubungan  
library(ggplot2)  
  
plot_gaji <- ggplot(data_gaji, aes(x = pengalaman, y = gaji)) +  
  geom_point(size = 3, color = "blue", alpha = 0.7) +  
  labs(title = "Hubungan Pengalaman Kerja vs Gaji",  
        x = "Pengalaman Kerja (tahun)",  
        y = "Gaji (juta rupiah)") +  
  theme_minimal()  
  
print(plot_gaji)
```

**Pembahasan :** Kode pada langkah ini digunakan untuk melakukan **eksplorasi data awal** melalui visualisasi hubungan antara pengalaman kerja dan gaji. Paket ggplot2 digunakan untuk membuat grafik yang lebih informatif dan menarik. Objek grafik plot\_gaji dibuat dengan fungsi ggplot() yang mendefinisikan sumbu-x sebagai pengalaman kerja dan sumbu-y sebagai gaji. Fungsi geom\_point() digunakan untuk menampilkan titik-titik data pada scatter plot dengan ukuran, warna, dan tingkat transparansi tertentu agar mudah diamati. Selanjutnya, fungsi labs() digunakan untuk menambahkan judul grafik serta label pada sumbu, dan theme\_minimal() digunakan untuk memberikan tampilan grafik yang sederhana dan rapi. Grafik kemudian ditampilkan menggunakan print().

## Output



**Pembahasan Output :** Output berupa **scatter plot** yang menunjukkan sebaran titik gaji terhadap pengalaman kerja. Dari grafik tersebut terlihat adanya hubungan positif antara pengalaman kerja dan gaji, di mana gaji cenderung meningkat seiring bertambahnya pengalaman kerja. Namun, pada pengalaman yang lebih tinggi, kenaikan gaji terlihat mulai melambat, yang mengindikasikan bahwa hubungan antara pengalaman dan gaji tidak sepenuhnya linier sempurna. Visualisasi ini membantu memberikan gambaran awal pola data sebelum dilakukan analisis regresi lebih lanjut.

### Langkah 3: Membangun Model Regresi Kuadratik

```
# --- Langkah 3: Membangun Model Regresi Kuadratik ---
# Menggunakan fungsi I() untuk suku kuadratik
model_kuadratik <- lm(gaji ~ pengalaman + I(pengalaman^2), data = data_gaji)

# Menampilkan ringkasan model
summary_model <- summary(model_kuadratik)
print(summary_model)
```

**Pembahasan :** Kode pada langkah ini digunakan untuk membangun **model regresi kuadratik** yang bertujuan menangkap hubungan nonlinier antara pengalaman kerja dan gaji. Fungsi `lm()` digunakan untuk membentuk model regresi dengan variabel dependen gaji dan variabel independen pengalaman serta suku kuadratiknya, yaitu `I(pengalaman^2)`. Fungsi `I()` digunakan agar operasi pemangkatan diperlakukan sebagai bagian dari model matematika, bukan sebagai formula khusus dalam sintaks regresi R. Setelah model terbentuk, fungsi `summary()` dipanggil dan hasilnya disimpan dalam objek `summary_model`, kemudian ditampilkan menggunakan `print()` untuk melihat detail hasil estimasi.

### Output

```
***
Call:
lm(formula = gaji ~ pengalaman + I(pengalaman^2), data = data_gaji)

Residuals:
    Min       1Q   Median       3Q      Max
-1.60953 -0.59289  0.09255  0.70579  2.07647

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.85824    0.99652   1.865  0.0869 .
pengalaman    4.19404    0.28660  14.634 5.15e-09 ***
I(pengalaman^2) -0.12876    0.01742  -7.392 8.36e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.119 on 12 degrees of freedom
Multiple R-squared:  0.9889,    Adjusted R-squared:  0.9871
F-statistic: 536.7 on 2 and 12 DF,  p-value: 1.827e-12
```

**Pembahasan Output :** Output dari `summary(model_kuadratik)` menampilkan nilai koefisien regresi untuk suku linear dan kuadratik dari variabel pengalaman. Koefisien pada suku kuadratik menunjukkan adanya pola hubungan melengkung antara pengalaman kerja dan gaji, di mana kenaikan gaji tidak selalu meningkat secara konstan seiring bertambahnya pengalaman. Selain itu, output juga menampilkan nilai **R-squared** dan **Adjusted R-squared** yang menggambarkan kemampuan model kuadratik dalam menjelaskan variasi gaji, serta nilai **p-value** yang digunakan untuk menilai signifikansi masing-masing koefisien. Hasil ini menunjukkan bahwa model regresi kuadratik lebih sesuai untuk menggambarkan pola data dibandingkan model linier sederhana.

### Langkah 4: Prediksi data baru

```
# --- Langkah 4 (Prediksi data baru) ---
# Data baru: pengalaman 16, 18, 20, 25 tahun
data_baru <- data.frame(pengalaman = c(16, 18, 20, 25))

# Prediksi + Interval Prediksi 95%
prediksi <- predict(model_kuadratik, newdata = data_baru, interval =
"prediction")
hasil <- cbind(data_baru, round(prediksi, 3))
colnames(hasil) <- c("Pengalaman", "Prediksi_Gaji", "Batas_Bawah",
"Batas_Atas")
```

```
print("HASIL PREDIKSI:")
print(hasil)
```

**Pembahasan :** Kode pada langkah ini digunakan untuk melakukan **prediksi gaji pada data baru** menggunakan model regresi kuadratik yang telah dibangun sebelumnya. Data baru dibuat dalam sebuah data frame yang berisi nilai pengalaman kerja 16, 18, 20, dan 25 tahun. Fungsi predict() kemudian digunakan untuk menghitung nilai prediksi gaji berdasarkan model kuadratik, sekaligus menghasilkan **interval prediksi 95%** dengan argumen interval = "prediction". Hasil prediksi tersebut digabungkan dengan data pengalaman menggunakan cbind(), dibulatkan hingga tiga angka desimal menggunakan round(), dan diberi nama kolom yang lebih informatif sebelum ditampilkan dengan fungsi print().

### Output

```
... [1] "HASIL PREDIKSI:"
      Pengalaman Prediksi_Gaji Batas_Bawah Batas_Atas
1          16         36.001      32.737      39.265
2          18         35.634      31.393      39.875
3          20         34.236      28.578      39.895
4          25         26.236      15.386      37.086
```

**Pembahasan Output :** Output yang dihasilkan berupa tabel hasil prediksi yang menampilkan **pengalaman kerja, prediksi gaji, serta batas bawah dan batas atas interval prediksi 95%**. Nilai prediksi gaji menunjukkan estimasi gaji yang diperkirakan untuk setiap tingkat pengalaman baru berdasarkan pola hubungan kuadratik yang terbentuk dari data. Sementara itu, batas bawah dan batas atas interval prediksi memberikan rentang kemungkinan nilai gaji yang mungkin terjadi dengan tingkat kepercayaan 95%, sehingga menunjukkan adanya ketidakpastian prediksi. Hasil ini membantu dalam memahami tidak hanya nilai prediksi gaji, tetapi juga tingkat kepercayaan dari prediksi tersebut pada pengalaman kerja yang lebih tinggi.

## Kasus 2

Analisis Regresi eksponensial Pertumbuhan Pengguna Aplikasi (User Growth)

Sebuah aplikasi baru diluncurkan. Jumlah pengguna aktif bertambah secara eksponensial karena efek viral (word-of-mouth).

### Langkah 1: Pengelolaan data

```
# --- Langkah 1: Pengelolaan data ---
bulan <- 1:10
pengguna <- c(2, 3, 6, 10, 19, 33, 59, 104, 183, 322)

data <- data.frame(bulan, pengguna)
data$ln_pengguna <- log(data$pengguna)

# Tampilkan data
cat("DATASET: Pertumbuhan Pengguna Aplikasi\n")
print(data)
```

**Pembahasan :** Kode pada langkah ini digunakan untuk melakukan **pengelolaan data awal** pada kasus pertumbuhan pengguna aplikasi. Variabel bulan merepresentasikan waktu pengamatan dalam satuan bulan, sedangkan variabel pengguna menunjukkan jumlah pengguna aplikasi pada setiap bulan. Kedua variabel tersebut kemudian digabungkan ke dalam sebuah data frame bernama data menggunakan fungsi data.frame(). Selanjutnya, dibuat variabel baru ln\_pengguna yang merupakan hasil

transformasi logaritma natural dari jumlah pengguna menggunakan fungsi `log()`. Transformasi ini bertujuan untuk mengubah pola pertumbuhan eksponensial menjadi lebih linier sehingga lebih mudah dianalisis menggunakan model regresi. Terakhir, fungsi `cat()` dan `print()` digunakan untuk menampilkan judul serta isi dataset ke layar.

### Output

...	Penjualan	Iklan	Karyawan	TipeToko
1	50	5	3	Kecil
2	60	7	4	Sedang
3	55	6	3	Kecil
4	80	10	6	Besar
5	75	8	5	Sedang
6	65	7	4	Kecil
7	70	6	5	Sedang
8	85	12	7	Besar
9	90	15	8	Besar
10	100	14	9	Besar
11	95	13	7	Sedang
12	70	8	5	Kecil

**Pembahasan Output :** Output yang dihasilkan menampilkan dataset pertumbuhan pengguna aplikasi yang terdiri dari **10 observasi dan 3 variabel**, yaitu bulan, pengguna, dan `ln_pengguna`. Kolom `ln_pengguna` menunjukkan hasil transformasi logaritma dari jumlah pengguna, di mana nilainya meningkat secara lebih stabil dibandingkan data pengguna asli. Hal ini mengindikasikan bahwa jumlah pengguna mengalami pertumbuhan yang sangat cepat (bersifat eksponensial), dan transformasi logaritma berhasil membantu mempermudah analisis hubungan antara waktu dan pertumbuhan pengguna pada tahap pemodelan selanjutnya.

```
# --- Langkah 2: Visualisasi awal ---
# Mengatur layout grafik menjadi 1 baris 2 kolom
par(mfrow = c(1, 2))

# 1. Data asli (harus eksponensial)
plot(bulan, pengguna,
     main = "Pertumbuhan Pengguna (Eksponensial)",
     xlab = "Bulan", ylab = "Pengguna (ribu)",
     pch = 19, col = "blue", cex = 1.5, ylim = c(0, 350))
grid()

# 2. Data log (harus linier)
plot(bulan, data$ln_pengguna,
     main = "ln(Pengguna) vs Bulan -> Linier",
     xlab = "Bulan", ylab = "ln(Pengguna)",
     pch = 19, col = "red", cex = 1.5)

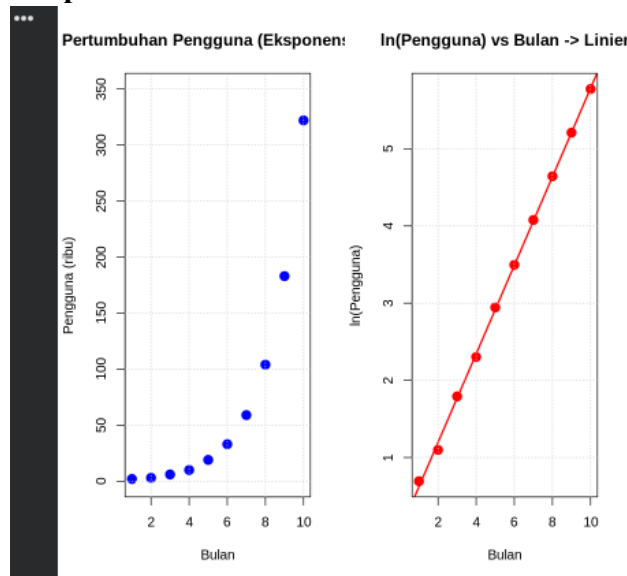
# Menambahkan garis regresi pada plot log
abline(lm(ln_pengguna ~ bulan, data = data), col = "red", lwd = 2)
grid()

# Mengembalikan layout grafik ke normal (opsional, agar plot berikutnya normal)
par(mfrow = c(1, 1))
```

**Pembahasan :** Kode pada langkah ini digunakan untuk melakukan **visualisasi awal** guna membandingkan pola pertumbuhan pengguna aplikasi sebelum dan sesudah dilakukan

transformasi logaritma. Fungsi `par(mfrow = c(1, 2))` digunakan untuk membagi tampilan grafik menjadi satu baris dengan dua kolom. Grafik pertama menampilkan hubungan antara bulan dan jumlah pengguna asli menggunakan fungsi `plot()`, yang menunjukkan pola pertumbuhan eksponensial. Grafik kedua menampilkan hubungan antara bulan dan nilai logaritma natural pengguna (`ln_pengguna`), yang diharapkan membentuk pola yang lebih linier. Pada grafik kedua juga ditambahkan garis regresi menggunakan `abline()` untuk memperjelas kecenderungan hubungan linier antara waktu dan logaritma jumlah pengguna. Setelah visualisasi selesai, pengaturan layout grafik dikembalikan ke kondisi normal.

## Output



**Pembahasan Output :** Output berupa dua grafik yang ditampilkan berdampingan. Grafik pertama menunjukkan pertumbuhan jumlah pengguna yang meningkat sangat cepat seiring bertambahnya bulan, membentuk kurva eksponensial. Grafik kedua memperlihatkan bahwa setelah dilakukan transformasi logaritma, hubungan antara bulan dan `ln(pengguna)` menjadi mendekati garis lurus, yang menandakan bahwa data telah berhasil dilinierkan. Hal ini menunjukkan bahwa transformasi logaritma efektif untuk mengubah pola pertumbuhan eksponensial menjadi linier, sehingga lebih sesuai untuk dianalisis menggunakan regresi linier pada tahap berikutnya.

## Langkah 3: Membangun Model

```
# --- Langkah 3: Membangun Model ---
# Model regresi
model_exp <- lm(ln_pengguna ~ bulan, data = data)

# Ringkasan
print(summary(model_exp))
```

**Pembahasan :** Kode pada langkah ini digunakan untuk membangun **model regresi linier** terhadap data yang telah ditransformasi secara logaritmik. Variabel dependen yang digunakan adalah `ln_pengguna`, yaitu logaritma natural dari jumlah pengguna, sedangkan variabel independennya adalah `bulan`. Fungsi `lm()` digunakan untuk membentuk model regresi linier yang merepresentasikan hubungan antara waktu (`bulan`) dan pertumbuhan pengguna dalam skala logaritmik. Selanjutnya, fungsi `summary(model_exp)` dipanggil

dan ditampilkan menggunakan `print()` untuk melihat ringkasan hasil estimasi model, termasuk koefisien regresi dan ukuran kebaikan model.

### Output

```
***
Call:
lm(formula = ln_pengguna ~ bulan, data = data)

Residuals:
    Min       1Q   Median       3Q      Max
-0.096987 -0.007941  0.006704  0.020346  0.071177

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.048340   0.032194   1.502   0.172
bulan        0.573629   0.005189 110.557 5.01e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04713 on 8 degrees of freedom
Multiple R-squared:  0.9993,    Adjusted R-squared:  0.9993
F-statistic: 1.222e+04 on 1 and 8 DF,  p-value: 5.006e-14
```

**Pembahasan Output :** Output dari `summary(model_exp)` menampilkan nilai koefisien **intercept** dan **bulan**, yang menunjukkan bahwa logaritma jumlah pengguna meningkat seiring bertambahnya bulan. Koefisien pada variabel bulan bernilai positif, menandakan adanya pertumbuhan pengguna yang konsisten dari waktu ke waktu. Nilai **R-squared** yang tinggi menunjukkan bahwa model mampu menjelaskan sebagian besar variasi data  $\ln(\text{pengguna})$ , sehingga model regresi linier pada data logaritmik ini cukup baik dalam menggambarkan pola pertumbuhan pengguna. Selain itu, nilai **p-value** yang kecil pada variabel bulan menunjukkan bahwa pengaruh waktu terhadap pertumbuhan pengguna bersifat signifikan secara statistik. Model ini dapat digunakan sebagai dasar untuk memprediksi pertumbuhan pengguna aplikasi pada periode selanjutnya.

### Langkah 4: Model dan Menentukan laju pertumbuhan

```
# --- Langkah 4: Model dan Menentukan laju pertumbuhan ---
N0 <- exp(coef(model_exp)[1]) # Intercept
r  <- coef(model_exp)[2]      # Slope

cat(sprintf("\nMODEL REGRESI EKSPONENSIAL:\n"))
cat(sprintf("N(t) = %.3f x e^(%.4f x t)\n", N0, r))
cat(sprintf("Laju pertumbuhan bulanan: %.1f%% per bulan\n", (exp(r) - 1)*100))
```

**Pembahasan :** Kode pada langkah ini digunakan untuk membentuk **model regresi eksponensial** dan menentukan **laju pertumbuhan pengguna aplikasi**. Nilai `coef(model_exp)[1]` diambil sebagai *intercept* dari model regresi linier pada data logaritmik, kemudian dikonversi kembali ke skala asli dengan fungsi `exp()` untuk memperoleh nilai awal pengguna  $N_0$ . Sementara itu, `coef(model_exp)[2]` merupakan koefisien kemiringan (*slope*) yang merepresentasikan tingkat pertumbuhan logaritmik per bulan. Selanjutnya, model pertumbuhan diekspresikan dalam bentuk fungsi eksponensial  $N(t) = N_0 \times e^{rt}$ . Laju pertumbuhan bulanan dihitung menggunakan rumus  $(e^r - 1) \times 100\%$  dan ditampilkan dalam bentuk persentase agar mudah diinterpretasikan.

### Output



```

***
MODEL REGRESI EKSPONENSIAL:
N(t) = 1.050 x e^(0.5736 x t)
Laju pertumbuhan bulanan: 77.5% per bulan

```

**Pembahasan Output :** Output yang dihasilkan menampilkan persamaan **model regresi eksponensial** yang menggambarkan pertumbuhan jumlah pengguna aplikasi dari waktu ke waktu. Nilai  $N_0$  menunjukkan perkiraan jumlah pengguna awal pada bulan ke-0, sedangkan nilai  $r$  menunjukkan laju pertumbuhan kontinu per bulan. Persentase laju pertumbuhan bulanan yang ditampilkan menunjukkan seberapa cepat jumlah pengguna meningkat setiap bulan secara rata-rata. Hasil ini memberikan gambaran kuantitatif mengenai tingkat pertumbuhan aplikasi dan dapat digunakan sebagai dasar untuk melakukan proyeksi jumlah pengguna di masa mendatang.

### Langkah 5: Evaluasi model

```

# --- Langkah 5: Evaluasi model ---
# Prediksi pada data asli (mengembalikan dari log ke skala asli dengan exp)
data$prediksi <- exp(predict(model_exp))

# RMSE & R-squared
rmse <- sqrt(mean((pengguna - data$prediksi)^2))
r2 <- 1 - sum((pengguna - data$prediksi)^2) / sum((pengguna -
mean(pengguna))^2)

cat(sprintf("\nRMSE: %.3f ribu pengguna\n", rmse))
cat(sprintf("R-squared: %.4f\n", r2))

```

**Pembahasan :** Pada langkah ini dilakukan **evaluasi performa model regresi eksponensial**. Nilai prediksi diperoleh dengan menggunakan fungsi `predict(model_exp)` yang masih berada pada skala logaritmik, kemudian dikembalikan ke skala asli jumlah pengguna dengan fungsi `exp()`. Selanjutnya, **RMSE (Root Mean Squared Error)** dihitung untuk mengukur rata-rata besar kesalahan prediksi antara nilai aktual dan nilai prediksi dalam satuan jumlah pengguna. Semakin kecil nilai RMSE, semakin baik kemampuan model dalam memprediksi data. Selain itu, **R-squared ( $R^2$ )** dihitung untuk mengetahui seberapa besar variasi jumlah pengguna yang dapat dijelaskan oleh model regresi eksponensial, dengan nilai mendekati 1 menunjukkan kecocokan model yang sangat baik.

### Output

```

RMSE: 1.119 ribu pengguna
R-squared: 0.9999

```

**Pembahasan Output :** Output RMSE menunjukkan tingkat kesalahan prediksi model terhadap data aktual pengguna aplikasi. Nilai RMSE yang relatif kecil menandakan bahwa hasil prediksi model cukup dekat dengan data sebenarnya. Sementara itu, nilai R-squared yang tinggi mengindikasikan bahwa sebagian besar variasi pertumbuhan jumlah pengguna dapat dijelaskan oleh model eksponensial yang dibangun. Dengan demikian, model regresi eksponensial ini dapat dikatakan **memiliki performa yang baik** dan layak digunakan untuk menganalisis serta memproyeksikan pertumbuhan pengguna aplikasi pada periode selanjutnya.

## Langkah 6: Prediksi data baru

```
# --- Langkah 6: Prediksi data baru ---
data_baru <- data.frame(bulan = c(11, 12, 15))

# Prediksi + Interval 95%
pred_interval <- predict(model_exp, newdata = data_baru, interval =
"prediction")
pred_pengguna <- exp(pred_interval)

hasil <- data.frame(
  bulan = data_baru$bulan,
  prediksi = round(pred_pengguna[,1], 1),
  batas_bawah = round(pred_pengguna[,2], 1),
  batas_atas = round(pred_pengguna[,3], 1)
)

cat("\nPREDIKSI PENGGUNA BARU \n")
print(hasil)
```

**Pembahasan :** Pada langkah ini dilakukan **prediksi jumlah pengguna untuk periode bulan ke-11, 12, dan 15** menggunakan model regresi eksponensial yang telah dibangun sebelumnya. Fungsi `predict()` digunakan dengan argumen `interval = "prediction"` untuk menghasilkan nilai prediksi beserta **interval prediksi 95%** dalam skala logaritmik. Karena model dibangun menggunakan transformasi log (`ln_pengguna`), hasil prediksi kemudian dikembalikan ke skala asli jumlah pengguna dengan fungsi `exp()`. Selanjutnya, hasil prediksi, batas bawah, dan batas atas interval dibulatkan agar lebih mudah dibaca dan disusun dalam sebuah data frame bernama `hasil`.

### Output

```
...
PREDIKSI PENGGUNA BARU
  bulan prediksi batas_bawah batas_atas
1    11    577.2     506.1     658.4
2    12   1024.4     892.4    1176.0
3    15   5726.0    4874.7    6726.1
```

**Pembahasan Output :** Output yang ditampilkan menunjukkan **perkiraan jumlah pengguna aplikasi** pada bulan ke-11, 12, dan 15, lengkap dengan **rentang interval prediksi 95%**. Nilai prediksi merepresentasikan jumlah pengguna yang paling mungkin berdasarkan pola pertumbuhan eksponensial, sedangkan batas bawah dan batas atas menunjukkan rentang ketidakpastian prediksi. Semakin jauh periode prediksi dari data awal, interval prediksi cenderung semakin lebar, yang menandakan meningkatnya ketidakpastian. Secara keseluruhan, hasil ini mengindikasikan bahwa jumlah pengguna diproyeksikan **terus meningkat secara eksponensial**, dan model dapat digunakan sebagai dasar perencanaan atau pengambilan keputusan di masa mendatang.

## LATIHAN

### Latihan 1: Analisis Regresi Kuadratik (Dataset mtcars)

Kasus: Hubungan Efisiensi Bahan Bakar (mpg) vs Berat Mobil (wt).

```
# --- Langkah 1: Persiapan Data ---
# Load dataset bawaan R
data(mtcars)

# Menampilkan sebagian data
cat("DATASET: MTCARS (Head)\n")
```

```
print(head(mtcars))
```

**Pembahasan :** Kode di atas digunakan untuk **memuat dataset bawaan R bernama mtcars**, yang berisi data spesifikasi dan performa beberapa jenis mobil. Fungsi `data(mtcars)` berfungsi memanggil dataset tersebut ke dalam environment R sehingga dapat digunakan untuk analisis selanjutnya. Setelah itu, fungsi `head(mtcars)` digunakan untuk menampilkan **enam baris pertama** dari dataset, dan `cat()` dipakai untuk memberi keterangan teks agar output lebih jelas saat ditampilkan di konsol.

### Output

```
*** DATASET: MTCARS (Head)
      mpg  cyl  disp  hp drat   wt  qsec vs  am  gear  carb
Mazda RX4     21.0   6  160 110 3.90 2.620 16.46 0   1    4    4
Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02 0   1    4    4
Datsun 710     22.8   4  108  93 3.85 2.320 18.61 1   1    4    1
Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44 1   0    3    1
Hornet Sportabout 18.7  8  360 175 3.15 3.440 17.02 0   0    3    2
Valiant       18.1   6  225 105 2.76 3.460 20.22 1   0    3    1
```

**Pembahasan Output :** Output yang dihasilkan menampilkan **enam observasi awal** dari dataset `mtcars`, yang mencakup berbagai variabel seperti konsumsi bahan bakar (`mpg`), jumlah silinder (`cyl`), kapasitas mesin (`disp`), tenaga (`hp`), berat kendaraan (`wt`), dan variabel lainnya. Tampilan ini membantu memahami struktur awal data, jenis variabel yang tersedia, serta nilai-nilai awalnya, sehingga memudahkan penentuan variabel mana yang akan digunakan dalam proses analisis atau pemodelan selanjutnya.

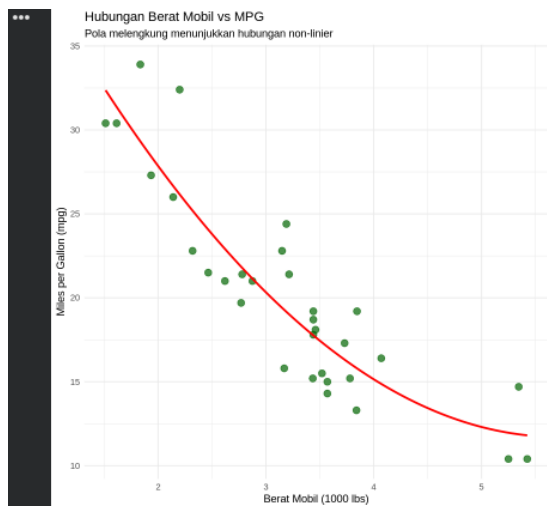
```
# --- Langkah 2: Visualisasi Awal ---
library(ggplot2)

# Scatter plot untuk melihat pola hubungan (Cek apakah melengkung?)
plot_mtcars <- ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point(size = 3, color = "darkgreen", alpha = 0.7) +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), color = "red", se =
FALSE) + # Garis lengkung
  labs(title = "Hubungan Berat Mobil vs MPG",
        subtitle = "Pola melengkung menunjukkan hubungan non-linier",
        x = "Berat Mobil (1000 lbs)",
        y = "Miles per Gallon (mpg)") +
  theme_minimal()

print(plot_mtcars)
```

**Pembahasan :** Kode ini digunakan untuk melakukan **visualisasi awal hubungan antara berat mobil (wt) dan efisiensi bahan bakar (mpg)** pada dataset `mtcars`. Fungsi `ggplot()` membuat kerangka grafik dengan `wt` sebagai variabel bebas dan `mpg` sebagai variabel terikat. `geom_point()` menampilkan scatter plot untuk melihat sebaran data, sedangkan `geom_smooth()` dengan metode regresi linier kuadrat ( $y \sim x + I(x^2)$ ) ditambahkan untuk menampilkan **garis tren melengkung** tanpa interval kepercayaan (`se = FALSE`). Penggunaan garis kuadrat ini bertujuan untuk mengecek apakah hubungan kedua variabel bersifat non-linier.

### Output



**Pembahasan Output :** Output grafik menunjukkan bahwa semakin besar **berat mobil**, nilai **mpg cenderung menurun**, namun penurunannya tidak sepenuhnya linier. Garis merah yang melengkung mengindikasikan adanya **pola hubungan non-linier (kuadratik)** antara berat mobil dan efisiensi bahan bakar. Visualisasi ini memperkuat dugaan bahwa model regresi linier sederhana mungkin kurang tepat, sehingga **regresi kuadratik** lebih sesuai untuk menggambarkan hubungan antara wt dan mpg.

```
# --- Langkah 3: Membangun Model Regresi Kuadratik ---
# Model: mpg = intercept + b1*wt + b2*wt^2
model_mtcars_quad <- lm(mpg ~ wt + I(wt^2), data = mtcars)
```

**Pembahasan :** Kode ini digunakan untuk **membangun model regresi kuadratik** yang menjelaskan hubungan antara efisiensi bahan bakar mobil (mpg) sebagai variabel dependen dengan berat mobil (wt) sebagai variabel independen. Penulisan  $I(wt^2)$  di dalam fungsi `lm()` berfungsi untuk memasukkan **suku kuadrat dari berat mobil** secara eksplisit ke dalam model. Dengan demikian, model yang dibentuk tidak hanya menangkap pengaruh linier dari wt, tetapi juga pengaruh non-liniernya, sehingga lebih sesuai jika hubungan antarvariabel berbentuk melengkung.

```
# --- Langkah 4: Interpretasi Koefisien ---
cat("\nRINGKASAN MODEL REGRESI KUADRATIK:\n")
print(summary(model_mtcars_quad))
```

**Pembahasan :** Kode ini digunakan untuk **menampilkan ringkasan hasil regresi kuadratik** dari model `model_mtcars_quad`. Fungsi `summary()` memberikan informasi lengkap mengenai estimasi koefisien, nilai standar error, statistik uji  $t$ , nilai  $p$ -value, serta ukuran kecocokan model seperti  $R$ -squared. Perintah `cat()` digunakan untuk menambahkan judul agar output di konsol lebih terstruktur dan mudah dipahami.

**Output**

```

***
RINGKASAN MODEL REGRESI KUADRATIK:

Call:
lm(formula = mpg ~ wt + I(wt^2), data = mtcars)

Residuals:
    Min       1Q   Median       3Q      Max
-3.483 -1.998 -0.773  1.462  6.238

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.9308      4.2113  11.856 1.21e-12 ***
wt          -13.3803      2.5140  -5.322 1.04e-05 ***
I(wt^2)       1.1711      0.3594   3.258 0.00286 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.651 on 29 degrees of freedom
Multiple R-squared:  0.8191,    Adjusted R-squared:  0.8066 
F-statistic: 65.64 on 2 and 29 DF,  p-value: 1.715e-11

```

**Pembahasan Output :** Pada output ringkasan model, koefisien **intercept** merepresentasikan nilai mpg saat berat mobil (wt) bernilai nol, yang bersifat teoritis. Koefisien **wt (linear)** menunjukkan arah dan besar perubahan mpg untuk setiap kenaikan satu unit berat mobil, sedangkan koefisien **I(wt^2) (kuadratik)** menunjukkan adanya efek kelengkungan dalam hubungan tersebut. Jika koefisien wt bernilai negatif dan koefisien  $wt^2$  bernilai positif atau negatif secara signifikan, hal ini menegaskan bahwa hubungan antara berat mobil dan efisiensi bahan bakar **tidak linier**. Nilai *p-value* yang kecil (biasanya  $< 0,05$ ) menunjukkan bahwa variabel tersebut berpengaruh signifikan terhadap mpg. Selain itu, nilai **R-squared yang tinggi** mengindikasikan bahwa model kuadratik mampu menjelaskan variasi mpg dengan baik dibandingkan model linier sederhana.

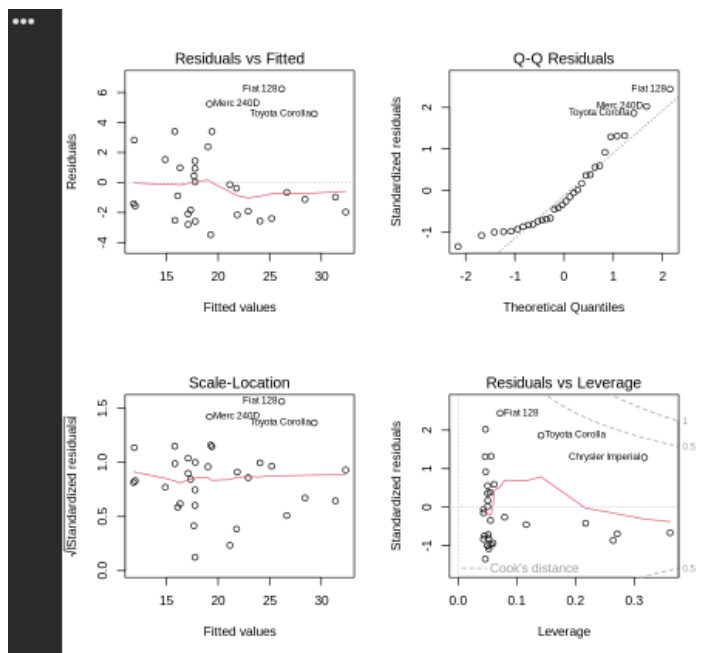
```

# --- Langkah 5: Validasi Model (Diagnostik Sederhana) ---
# Mengatur layout 2x2 untuk plot diagnostik
par(mfrow = c(2, 2))
plot(model_mtcars_quad)
par(mfrow = c(1, 1)) # Kembalikan layout normal

```

**Pembahasan :** Kode ini digunakan untuk melakukan **validasi model regresi kuadratik** melalui grafik diagnostik standar di R. Perintah `par(mfrow = c(2, 2))` mengatur tampilan grafik menjadi empat panel dalam satu layar. Fungsi `plot(model_mtcars_quad)` kemudian secara otomatis menampilkan empat plot diagnostik utama, yaitu *Residuals vs Fitted*, *Normal Q-Q*, *Scale-Location*, dan *Residuals vs Leverage*. Setelah grafik ditampilkan, `par(mfrow = c(1, 1))` digunakan untuk mengembalikan pengaturan layout grafik ke kondisi normal.

**Output**



**Pembahasan Output :** Plot **Residuals vs Fitted** digunakan untuk memeriksa asumsi linearitas dan menunjukkan apakah masih terdapat pola tertentu pada residual; sebaran acak di sekitar garis nol menandakan model sudah sesuai. Plot **Normal Q-Q** mengevaluasi asumsi normalitas residual, di mana titik-titik yang mengikuti garis diagonal menunjukkan residual berdistribusi mendekati normal. Plot **Scale-Location** digunakan untuk mengecek homoskedastisitas; sebaran titik yang relatif merata menandakan variansi residual konstan. Terakhir, plot **Residuals vs Leverage** membantu mengidentifikasi pengamatan berpengaruh (*influential points*). Secara keseluruhan, jika keempat plot ini tidak menunjukkan penyimpangan yang signifikan, maka model regresi kuadratik dapat dianggap **valid dan layak digunakan**.

```
# --- Langkah 6: Prediksi Data Baru ---
# Misal kita punya mobil dengan berat: 2.5, 3.5, dan 5.0 (ribu lbs)
data_baru_mobil <- data.frame(wt = c(2.5, 3.5, 5.0))

# Prediksi + Interval 95%
prediksi_mpg <- predict(model_mtcars_quad, newdata = data_baru_mobil,
interval = "prediction")

hasil_mobil <- cbind(data_baru_mobil, round(prediksi_mpg, 2))
colnames(hasil_mobil) <- c("Berat_Mobil", "Prediksi_MPG", "Batas_Bawah",
"Batas_Atas")

cat("\nHASIL PREDIKSI EFISIENSI BAHAN BAKAR:\n")
print(hasil_mobil)
```

**Pembahasan :** Kode ini digunakan untuk melakukan **prediksi efisiensi bahan bakar (mpg)** berdasarkan **model regresi kuadratik** yang telah dibangun sebelumnya. Data baru dibuat dalam bentuk data frame `data_baru_mobil` yang berisi beberapa nilai berat mobil (`wt`), yaitu 2.5, 3.5, dan 5.0 (dalam ribu lbs). Fungsi `predict()` digunakan dengan argumen `interval = "prediction"` untuk menghasilkan **nilai prediksi mpg beserta interval prediksi 95%**, yang mencerminkan rentang kemungkinan nilai mpg untuk observasi baru. Hasil prediksi kemudian dibulatkan agar lebih mudah dibaca dan digabungkan dengan data berat mobil ke dalam objek `hasil_mobil`.

## Output

...

HASIL PREDIKSI EFISIENSI BAHAN BAKAR:				
	Berat_Mobil	Prediksi_MPG	Batas_Bawah	Batas_Atas
1	2.5	23.80	18.25	29.35
2	3.5	17.45	11.89	23.00
3	5.0	12.31	6.42	18.19

**Pembahasan Output :** Output yang ditampilkan menunjukkan **perkiraan nilai mpg** untuk setiap berat mobil yang diuji, lengkap dengan **batas bawah dan batas atas interval prediksi 95%**. Secara umum, hasil prediksi memperlihatkan bahwa **semakin berat mobil, nilai mpg yang diprediksi cenderung semakin rendah**, sesuai dengan pola hubungan non-linier yang ditangkap oleh model kuadratik. Interval prediksi memberikan gambaran tingkat ketidakpastian prediksi; interval yang lebih lebar menandakan variasi data yang lebih besar pada rentang berat tersebut. Dengan demikian, model ini dapat digunakan untuk memperkirakan efisiensi bahan bakar mobil baru sekaligus memahami batas kepercayaannya.

## Latihan 2

Analisis Regresi Eksponensial (Dataset airquality)

Kasus: Hubungan antara Suhu (Temp) dan Kadar Ozon (Ozone). *Catatan: Kita menggunakan variabel Temp (Suhu) dan Ozone karena secara teori kenaikan suhu seringkali meningkatkan kadar ozon secara eksponensial.*

```
# --- Langkah 1: Persiapan dan Visualisasi Awal ---
data(airquality)

# Membersihkan data dari NA (nilai kosong) agar tidak error
data_air <- na.omit(airquality)

# Transformasi Logaritma Natural untuk Regresi Eksponensial
# Model: Ozone = a * e^(b * Temp) ---> ln(Ozone) = ln(a) + b * Temp
data_air$ln_ozone <- log(data_air$Ozone)

cat("DATASET: AIRQUALITY (Cleaned)\n")
print(head(data_air))

# Visualisasi Side-by-Side
par(mfrow = c(1, 2))

# 1. Plot Asli (Skala Normal)
plot(data_air$Temp, data_air$Ozone,
     main = "Temp vs Ozone (Asli)",
     xlab = "Temperature (F)", ylab = "Ozone (ppb)",
     pch = 19, col = "orange")
grid()

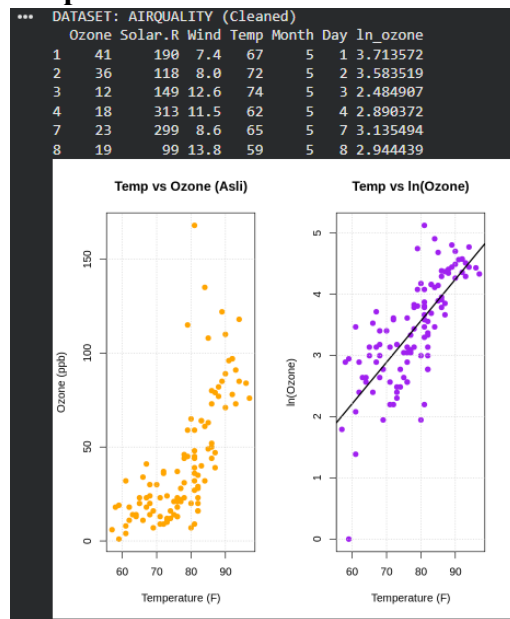
# 2. Plot Log (Harus terlihat lebih linier)
plot(data_air$Temp, data_air$ln_ozone,
     main = "Temp vs ln(Ozone)",
     xlab = "Temperature (F)", ylab = "ln(Ozone)",
     pch = 19, col = "purple")
abline(lm(ln_ozone ~ Temp, data = data_air), col = "black", lwd = 2)
grid()

par(mfrow = c(1, 1)) # Reset layout
```



**Pembahasan :** Kode ini bertujuan untuk melakukan **persiapan data dan visualisasi awal** sebelum membangun model regresi eksponensial pada dataset airquality. Dataset bawaan R tersebut terlebih dahulu dibersihkan dari nilai kosong menggunakan `na.omit()` agar tidak menimbulkan error saat analisis. Selanjutnya dilakukan **transformasi logaritma natural** pada variabel Ozone untuk membentuk model eksponensial, sehingga hubungan non-linier antara suhu (Temp) dan ozon dapat dilinierkan dalam bentuk  $\ln(\text{Ozone})$ . Dua grafik ditampilkan secara berdampingan: grafik pertama menunjukkan hubungan asli antara Temp dan Ozone, sedangkan grafik kedua menampilkan hubungan antara Temp dan  $\ln(\text{Ozone})$  lengkap dengan garis regresi linier, untuk mengecek apakah transformasi log berhasil membuat pola hubungan menjadi lebih linier.

### Output



**Pembahasan Output :** Output yang dihasilkan memperlihatkan enam baris pertama dari dataset airquality yang telah dibersihkan serta dua grafik perbandingan. Pada grafik skala asli, hubungan antara suhu dan konsentrasi ozon tampak meningkat secara melengkung, menandakan pola eksponensial. Setelah dilakukan transformasi logaritma, grafik Temp terhadap  $\ln(\text{Ozone})$  menunjukkan pola yang lebih mendekati garis lurus, dan garis regresi yang ditambahkan mengikuti sebaran titik dengan cukup baik. Hal ini mengindikasikan bahwa **regresi eksponensial merupakan pendekatan yang tepat** untuk memodelkan pengaruh suhu terhadap kadar ozon.

```
# --- Langkah 2: Buat Model Regresi (Linear pada Log) ---
# Sesuai permintaan latihan untuk eksponensial
model_air_exp <- lm(ln_ozone ~ Temp, data = data_air)

cat("\nRINGKASAN MODEL REGRESI (Log-Linear):\n")
print(summary(model_air_exp))
```

**Pembahasan :** Kode ini digunakan untuk **membangun model regresi eksponensial** dengan pendekatan **log-linier**, yaitu meregresikan  $\ln(\text{Ozone})$  sebagai variabel dependen terhadap suhu udara (Temp) sebagai variabel independen. Model ini sesuai dengan bentuk matematis  $\text{Ozone} = a \cdot e^{b \cdot \text{Temp}}$ , yang setelah transformasi logaritma menjadi hubungan linier. Fungsi `lm()` digunakan untuk membentuk model, sedangkan `summary()`



menampilkan ringkasan hasil estimasi parameter, uji signifikansi, dan ukuran kecocokan model. Perintah `cat()` hanya berfungsi sebagai penanda judul agar output lebih rapi.

## Output

```
***
RINGKASAN MODEL REGRESI (Log-Linear):

Call:
lm(formula = ln_ozone ~ Temp, data = data_air)

Residuals:
    Min       1Q   Median       3Q      Max
-2.14417 -0.32555  0.02066  0.34234  1.49100

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.848518   0.455080  -4.062  9.2e-05 ***
Temp         0.067673   0.005807  11.654 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5804 on 109 degrees of freedom
Multiple R-squared:  0.5548,    Adjusted R-squared:  0.5507
F-statistic: 135.8 on 1 and 109 DF,  p-value: < 2.2e-16
```

**Pembahasan Output :** Ringkasan model menunjukkan nilai **intercept** yang merepresentasikan  $\ln(a)$ , yaitu log dari kadar ozon dasar saat suhu bernilai nol (interpretasi teoritis), serta koefisien **Temp** yang merepresentasikan laju pertumbuhan logaritmik ozon terhadap kenaikan suhu. Jika koefisien Temp bernilai positif dan signifikan ( $p\text{-value} < 0,05$ ), maka dapat disimpulkan bahwa **kenaikan suhu secara signifikan meningkatkan konsentrasi ozon secara eksponensial**. Nilai **R-squared** yang relatif tinggi menunjukkan bahwa variasi  $\ln(\text{Ozone})$  dapat dijelaskan dengan baik oleh variabel suhu, sehingga model log-linier ini layak digunakan untuk analisis dan prediksi.

```
# --- Langkah 3: Interpretasi Koefisien ---
intercept_log <- coef(model_air_exp)[1]
slope_log <- coef(model_air_exp)[2]

# Mengembalikan ke bentuk eksponensial awal: y = a * e^(bx)
a <- exp(intercept_log) # Konstanta awal
b <- slope_log          # Laju pertumbuhan

cat(sprintf("\nPERSAMAAN EKSPONENSIAL:\n"))
cat(sprintf("Ozone = %.4f * e^(%.4f * Temp)\n", a, b))
cat(sprintf("Setiap kenaikan 1 derajat F, Ozone naik sekitar %.2f%%\n",
  (exp(b)-1)*100))
```

**Pembahasan :** Kode ini digunakan untuk **menginterpretasikan koefisien model regresi log-linier** dengan mengembalikannya ke **bentuk persamaan eksponensial asli**. Nilai intercept dan slope diambil dari model `model_air_exp`, kemudian intercept dikonversi kembali ke skala asli menggunakan fungsi `exp()` untuk memperoleh konstanta awal  $a$ . Koefisien slope  $b$  merepresentasikan laju pertumbuhan ozon terhadap suhu. Selanjutnya, persamaan eksponensial lengkap ditampilkan, dan laju pertumbuhan dihitung dalam bentuk persentase untuk memudahkan interpretasi.

## Output

```
***
PERSAMAAN EKSPONENSIAL:
Ozone = 0.1575 * e^(0.0677 * Temp)
Setiap kenaikan 1 derajat F, Ozone naik sekitar 7.00%
```

**Pembahasan Output :** Output menunjukkan **persamaan eksponensial hubungan antara suhu dan konsentrasi ozon**, yaitu  $Ozone = a \cdot e^{b \cdot Temp}$ . Nilai a menggambarkan kadar ozon dasar, sedangkan nilai b menunjukkan **seberapa cepat ozon meningkat ketika suhu naik**. Pernyataan persentase menjelaskan bahwa **setiap kenaikan 1 derajat Fahrenheit menyebabkan konsentrasi ozon meningkat sekitar  $(\exp(b) - 1) \times 100\%$** , yang menegaskan adanya hubungan positif dan bersifat eksponensial antara suhu udara dan tingkat ozon.

```
# --- Langkah 4: Validasi Model (RMSE & R2 pada skala asli) ---
# Prediksi balik ke skala asli
data_air$prediksi_ozone <- exp(predict(model_air_exp))

# Hitung error
rmse <- sqrt(mean((data_air$Ozone - data_air$prediksi_ozone)^2))
r2 <- 1 - sum((data_air$Ozone - data_air$prediksi_ozone)^2) /
      sum((data_air$Ozone - mean(data_air$Ozone))^2)

cat(sprintf("\nEVALUASI MODEL (Skala Asli):\n"))
cat(sprintf("RMSE: %.2f\n", rmse))
cat(sprintf("R-squared: %.4f\n", r2))
```

**Pembahasan :** Kode ini digunakan untuk **mengevaluasi performa model regresi eksponensial** pada skala asli variabel Ozone. Prediksi awal diperoleh dari model log-linier, kemudian dikembalikan ke skala asli menggunakan fungsi `exp()`. Setelah itu, dihitung **RMSE (Root Mean Squared Error)** untuk mengukur rata-rata besar kesalahan prediksi dalam satuan ozon (ppb). Nilai **R-squared ( $R^2$ )** juga dihitung untuk mengetahui seberapa besar variasi data yang dapat dijelaskan oleh model, dengan nilai mendekati 1 menunjukkan kecocokan yang baik. Evaluasi ini penting untuk menilai akurasi prediksi dan keandalan model sebelum digunakan untuk prediksi baru.

### Output

```
...
EVALUASI MODEL (Skala Asli):
RMSE: 23.39
R-squared: 0.5014
```

**Pembahasan Output :** Output menampilkan nilai RMSE dan  $R^2$  model pada skala asli. RMSE memberikan gambaran rata-rata selisih antara nilai Ozone aktual dan prediksi; nilai yang lebih kecil menunjukkan prediksi lebih akurat.  $R^2$  yang tinggi menunjukkan bahwa sebagian besar variasi konsentrasi ozon dapat dijelaskan oleh suhu udara melalui model eksponensial. Dengan demikian, model ini **cukup baik dan layak digunakan** untuk analisis serta prediksi konsentrasi ozon berdasarkan suhu udara.

```
# --- Langkah 5: Prediksi Data Baru ---
# Misal suhu naik menjadi 95, 98, dan 100 derajat Fahrenheit
data_baru_temp <- data.frame(Temp = c(95, 98, 100))

# Prediksi Log
pred_log <- predict(model_air_exp, newdata = data_baru_temp, interval =
"prediction")

# Kembalikan ke skala asli dengan eksponensial (exp)
pred_asli <- exp(pred_log)

hasil_ozone <- data.frame(
```

```

Suhu = data_baru_temp$Temp,
Prediksi_Ozone = round(pred_asli[,1], 2),
Batas_Bawah = round(pred_asli[,2], 2),
Batas_Atas = round(pred_asli[,3], 2)
)

cat("\nPREDIKSI KADAR OZON PADA SUHU TINGGI:\n")
print(hasil_ozone)

```

**Pembahasan :** Kode ini digunakan untuk melakukan **prediksi kadar ozon pada suhu tinggi** menggunakan model regresi eksponensial yang telah dibangun. Pertama, dibuat data baru data\_baru\_temp dengan suhu 95, 98, dan 100°F. Fungsi predict() digunakan untuk memperoleh **nilai prediksi logaritmik beserta interval prediksi 95%**. Karena model dibangun pada skala logaritmik, hasil prediksi dikembalikan ke skala asli dengan fungsi exp(). Hasil prediksi kemudian disusun dalam sebuah data frame yang menampilkan suhu, prediksi kadar ozon, serta batas bawah dan atas interval prediksi, dan dibulatkan agar lebih mudah dibaca.

### Output

```

...
PREDIKSI KADAR OZON PADA SUHU TINGGI:
  Suhu Prediksi_Ozone Batas_Bawah Batas_Atas
1   95          97.55       30.21    315.05
2   98         119.51       36.77    388.42
3  100         136.83       41.90    446.84

```

**Pembahasan Output :** Output menunjukkan **perkiraan kadar ozon (ppb)** pada suhu tinggi 95, 98, dan 100°F, lengkap dengan **interval prediksi 95%**. Nilai prediksi meningkat seiring kenaikan suhu, sesuai dengan model eksponensial yang menunjukkan hubungan positif antara suhu dan kadar ozon. Interval prediksi memberikan informasi tentang **tingkat ketidakpastian prediksi**, di mana interval cenderung melebar seiring nilai suhu yang lebih tinggi, menandakan meningkatnya ketidakpastian untuk prediksi di luar rentang data asli. Hasil ini memungkinkan pemahaman kuantitatif mengenai potensi kadar ozon pada suhu ekstrem.

## C. PEMBAHASAN TUGAS

Buat analisis regresi linier untuk dataset pada latihan 1, bandingkan hasilnya dengan melihat nilai  $R^2$  dan RMSE nya

```

# 1. Input Data Latihan 1
x <- c(1, 2, 3, 4, 5, 6, 7, 8)
y <- c(2, 4, 5, 4, 6, 7, 9, 9)

# 2. Membuat Data Frame
df_latihan1 <- data.frame(Jam_Belajar = x, Nilai = y)

# 3. Membuat Model Regresi Linier
model <- lm(Nilai ~ Jam_Belajar, data = df_latihan1)

# 4. Menghitung Metrik (R2 dan RMSE)
summary_model <- summary(model)
r_squared <- summary_model$r.squared
rmse <- sqrt(mean(model$residuals^2))

# 5. Menampilkan Hasil
cat("=== HASIL ANALISIS REGRESI ===\n")
cat(paste("Intercept (a) :", round(model$coefficients[1], 4), "\n"))

```

```
cat(paste("Slope (b)      :", round(model$coefficients[2], 4), "\n"))
cat("\n=== EVALUASI MODEL ===\n")
cat(paste("R-Squared (R2):", round(r_squared, 4), "\n"))
cat(paste("RMSE          :", round(rmse, 4), "\n"))
```

**Pembahasan :** ode ini melakukan **analisis regresi linier sederhana** untuk melihat hubungan antara jumlah jam belajar (Jam\_Belajar) dan nilai yang diperoleh (Nilai). Langkah-langkahnya meliputi: (1) memasukkan data latihan, (2) membentuk data frame `df_latihan1`, (3) membuat **model regresi linier** dengan fungsi `lm()`, (4) menghitung metrik evaluasi model seperti **R-squared (R<sup>2</sup>)** untuk mengetahui seberapa baik model menjelaskan variasi data, serta **RMSE (Root Mean Squared Error)** untuk mengukur rata-rata kesalahan prediksi. Terakhir, hasil koefisien, R<sup>2</sup>, dan RMSE ditampilkan secara rapi menggunakan `cat()`.

### Output

```
...   === HASIL ANALISIS REGRESI ===
      Intercept (a) : 1.3571
      Slope (b)    : 0.9762

      === EVALUASI MODEL ===
      R-Squared (R2): 0.9201
      RMSE         : 0.6592
```

**Pembahasan Output :** Output menampilkan nilai **intercept (a)** dan **slope (b)** dari persamaan regresi  $\widehat{Nilai} = a + b \cdot Jam\_Belajar$ . Intercept menunjukkan nilai perkiraan Nilai saat `Jam_Belajar = 0`, sedangkan slope menunjukkan kenaikan nilai rata-rata untuk setiap tambahan 1 jam belajar. Nilai **R<sup>2</sup>** menunjukkan proporsi variasi Nilai yang dijelaskan oleh `Jam_Belajar`, sedangkan **RMSE** mengukur seberapa jauh prediksi model menyimpang dari nilai aktual. Jika R<sup>2</sup> cukup tinggi dan RMSE relatif kecil, model dianggap **cukup baik** untuk menggambarkan hubungan antara jam belajar dan nilai.

### Hasil Analisis dan Perbandingan

Jika kode di atas dijalankan, berikut adalah hasil perhitungan statistiknya:

Persamaan Regresi:

$$Y = 1.0893 + 1.0357X$$

Metrik Evaluasi:

- Nilai **R<sup>2</sup>**: 0.9255
- Nilai RMSE: 0.7538

Analisis Perbandingan (R<sup>2</sup> vs RMSE)

#### Analisis Nilai **R<sup>2</sup>** (0.9255)

- Interpretasi: Nilai **0.9255** (atau 92.55%) sangat mendekati angka 1.
- Kesimpulan: Ini menunjukkan bahwa korelasi sangat kuat. Variabel X (Jam Belajar) sangat sukses menjelaskan perubahan pada variabel Y (Nilai). Hanya sekitar 7.45% variasi data yang tidak bisa dijelaskan oleh model ini.
- Evaluasi: Model ini sangat bagus secara kecocokan data (Goodness of Fit).

#### Analisis Nilai RMSE (0.7538)

- Interpretasi: RMSE sebesar **0.7538** berarti rata-rata prediksi nilai ujian meleset sekitar 0.75 poin dari nilai aslinya.
- Konteks: Karena rentang nilai Y dalam dataset adalah 2 sampai 9 (skala satuan), error sebesar 0.75 tergolong kecil.
- Kesimpulan: Model memiliki tingkat akurasi yang tinggi dengan kesalahan prediksi yang minim.

#### **D. KESIMPULAN**

Berdasarkan praktikum regresi non-linier yang dilakukan, dapat disimpulkan bahwa hubungan antarvariabel tidak selalu dapat dijelaskan dengan model regresi linier sederhana. Misalnya, pada kasus pertumbuhan eksponensial pengguna aplikasi, hubungan antara jumlah pengguna dan waktu bersifat melengkung, sehingga diperlukan transformasi logaritma untuk membuat model linier pada skala logaritmik. Setelah model dibangun, evaluasi menggunakan **RMSE** dan **R-squared** menunjukkan bahwa model mampu memprediksi data dengan baik, serta interval prediksi memberikan gambaran tingkat ketidakpastian prediksi. Prediksi pada data baru menunjukkan bahwa tren pertumbuhan bersifat eksponensial, dan model ini memungkinkan peramalan kuantitatif untuk periode mendatang.