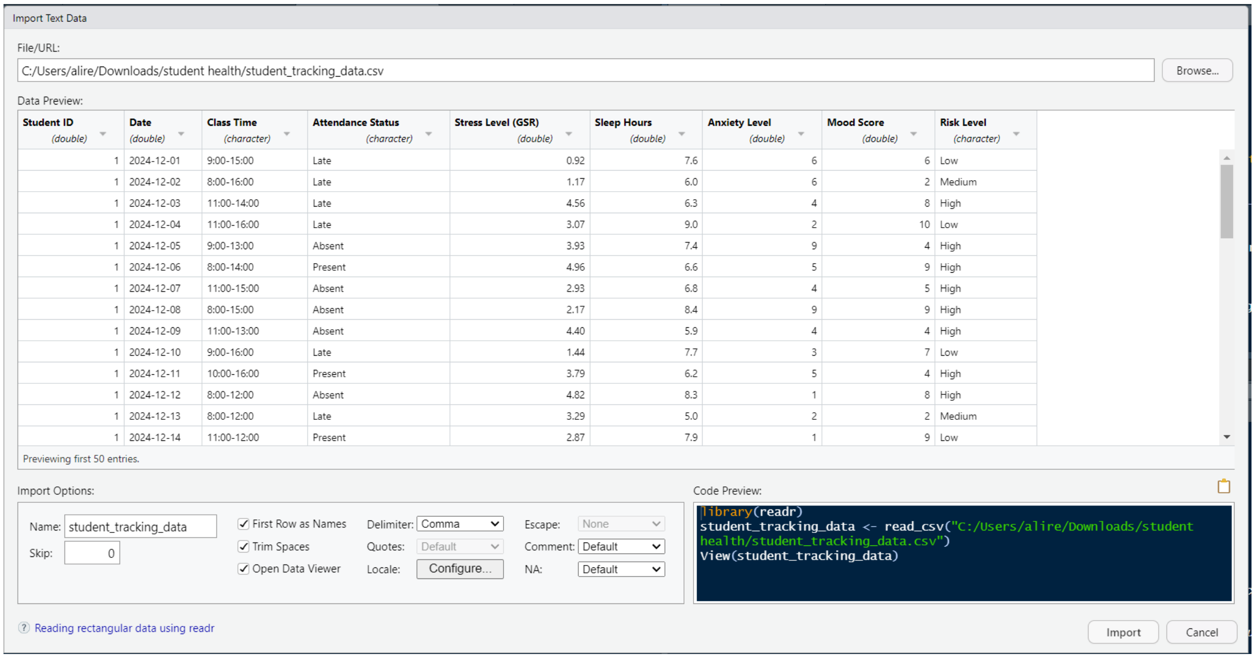
Final Project

Data Analyst Camp 2025

Nama: Ali Reza Bahtiar

Berikut langkah-langkah dan hasil dari Final Project:

Impor data student_tracking_data.csv



Import Text Data

File/URL: C:/Users/alire/Downloads/student health/student_tracking_data.csv

Data Preview:

Student ID (double)	Date (double)	Class Time (character)	Attendance Status (character)	Stress Level (GSR) (double)	Sleep Hours (double)	Anxiety Level (double)	Mood Score (double)	Risk Level (character)
1	2024-12-01	9:00-15:00	Late	0.92	7.6	6	6	Low
1	2024-12-02	8:00-16:00	Late	1.17	6.0	6	2	Medium
1	2024-12-03	11:00-14:00	Late	4.56	6.3	4	8	High
1	2024-12-04	11:00-16:00	Late	3.07	9.0	2	10	Low
1	2024-12-05	9:00-13:00	Absent	3.93	7.4	9	4	High
1	2024-12-06	8:00-14:00	Present	4.96	6.6	5	9	High
1	2024-12-07	11:00-15:00	Absent	2.93	6.8	4	5	High
1	2024-12-08	8:00-15:00	Absent	2.17	8.4	9	9	High
1	2024-12-09	11:00-13:00	Absent	4.40	5.9	4	4	High
1	2024-12-10	9:00-16:00	Late	1.44	7.7	3	7	Low
1	2024-12-11	10:00-16:00	Present	3.79	6.2	5	4	High
1	2024-12-12	8:00-12:00	Absent	4.82	8.3	1	8	High
1	2024-12-13	8:00-12:00	Late	3.29	5.0	2	2	Medium
1	2024-12-14	11:00-12:00	Present	2.87	7.9	1	9	Low

Previewing first 50 entries.

Import Options:

Name: student_tracking_data
Skip: 0

☑ First Row as Names
☑ Trim Spaces
☑ Open Data Viewer

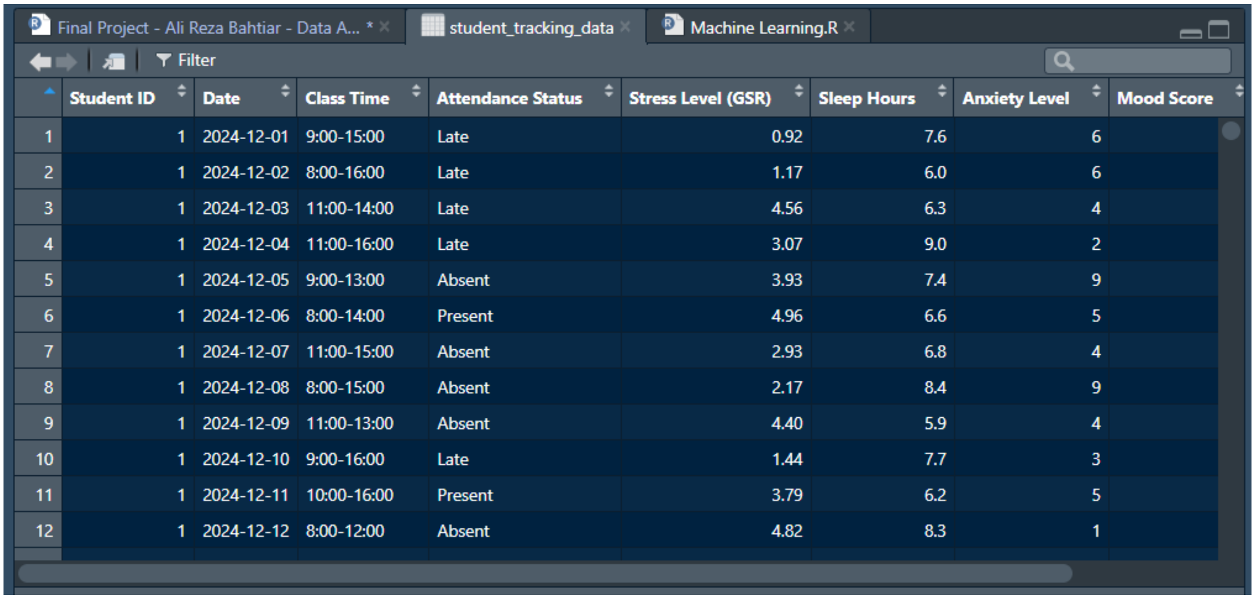
Delimiter: Comma
Quotes: Default
Locale: Configure...

Escape: None
Comment: Default
NA: Default

Code Preview:

```
library(readr)
student_tracking_data <- read_csv("C:/Users/alire/Downloads/student
health/student_tracking_data.csv")
View(student_tracking_data)
```

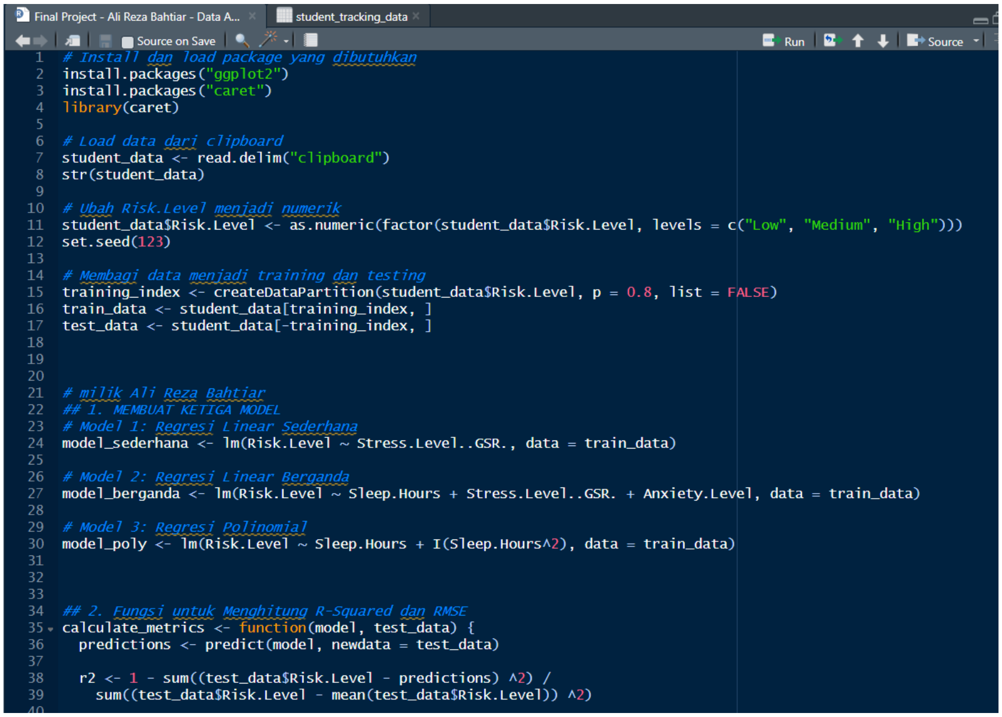
Browse...
Reading rectangular data using readr
Import
Cancel



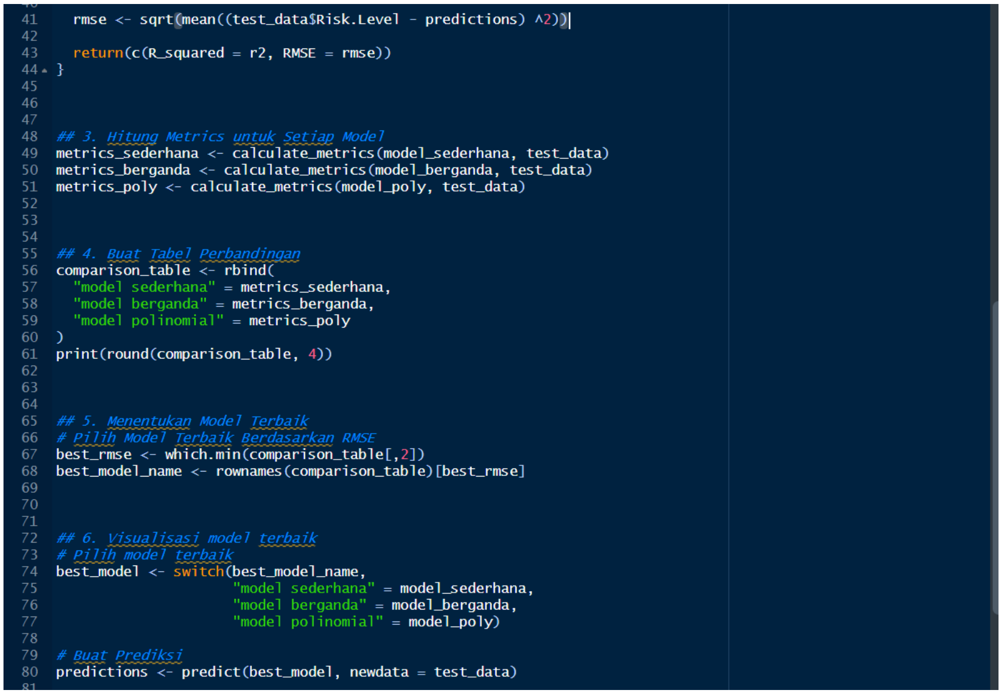
Final Project - Ali Reza Bahtiar - Data A... *
student_tracking_data
Machine Learning.R
Filter

	Student ID	Date	Class Time	Attendance Status	Stress Level (GSR)	Sleep Hours	Anxiety Level	Mood Score
1	1	2024-12-01	9:00-15:00	Late	0.92	7.6	6	
2	1	2024-12-02	8:00-16:00	Late	1.17	6.0	6	
3	1	2024-12-03	11:00-14:00	Late	4.56	6.3	4	
4	1	2024-12-04	11:00-16:00	Late	3.07	9.0	2	
5	1	2024-12-05	9:00-13:00	Absent	3.93	7.4	9	
6	1	2024-12-06	8:00-14:00	Present	4.96	6.6	5	
7	1	2024-12-07	11:00-15:00	Absent	2.93	6.8	4	
8	1	2024-12-08	8:00-15:00	Absent	2.17	8.4	9	
9	1	2024-12-09	11:00-13:00	Absent	4.40	5.9	4	
10	1	2024-12-10	9:00-16:00	Late	1.44	7.7	3	
11	1	2024-12-11	10:00-16:00	Present	3.79	6.2	5	
12	1	2024-12-12	8:00-12:00	Absent	4.82	8.3	1	

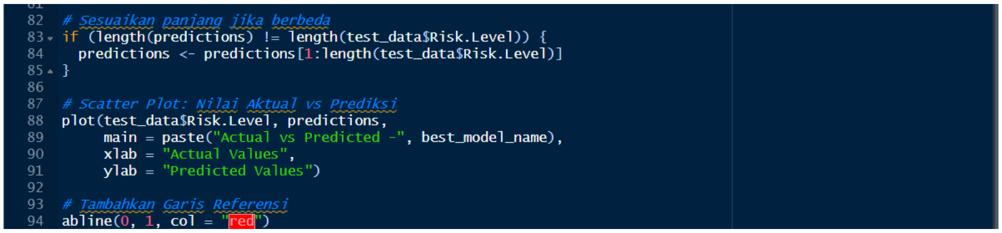
Coding:



```r
# Install dan load package yang dibutuhkan
install.packages("ggplot2")
install.packages("caret")
library(caret)

# Load data dari clipboard
student_data <- read.delim("clipboard")
str(student_data)

# Ubah Risk.Level menjadi numerik
student_data$Risk.Level <- as.numeric(factor(student_data$Risk.Level, levels = c("Low", "Medium", "High")))
set.seed(123)

# Membagi data menjadi training dan testing
training_index <- createDataPartition(student_data$Risk.Level, p = 0.8, list = FALSE)
train_data <- student_data[training_index, ]
test_data <- student_data[-training_index, ]



# milik Ali Reza Bahtiar
## 1. MEMBUAT KETIGA MODEL
# Model 1: Regresi Linear Sederhana
model_sederhana <- lm(Risk.Level ~ Stress.Level..GSR., data = train_data)

# Model 2: Regresi Linear Berganda
model_berganda <- lm(Risk.Level ~ Sleep.Hours + Stress.Level..GSR. + Anxiety.Level, data = train_data)

# Model 3: Regresi Polinomial
model_poly <- lm(Risk.Level ~ Sleep.Hours + I(Sleep.Hours^2), data = train_data)



## 2. Fungsi untuk Menghitung R-Squared dan RMSE
calculate_metrics <- function(model, test_data) {
  predictions <- predict(model, newdata = test_data)

  r2 <- 1 - sum((test_data$Risk.Level - predictions) ^2) /
    sum((test_data$Risk.Level - mean(test_data$Risk.Level)) ^2)
```

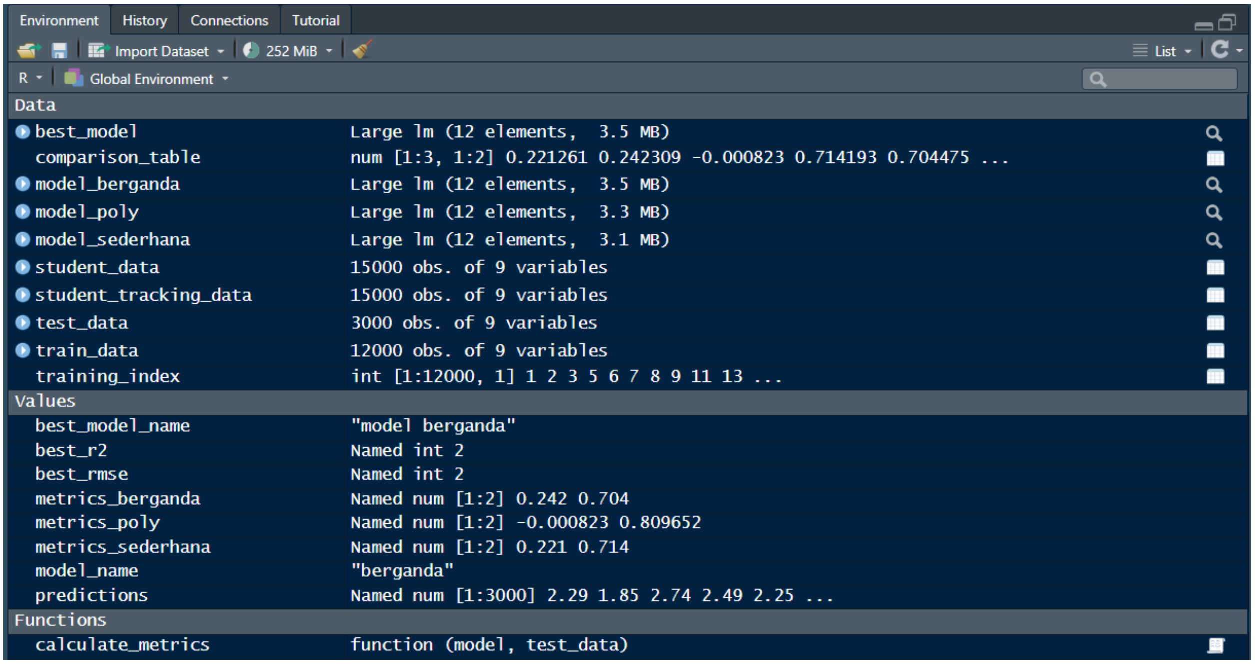


```r
  rmse <- sqrt(mean((test_data$Risk.Level - predictions) ^2))

  return(c(R_squared = r2, RMSE = rmse))
}



## 3. Hitung Metrics untuk Setiap Model
metrics_sederhana <- calculate_metrics(model_sederhana, test_data)
metrics_berganda <- calculate_metrics(model_berganda, test_data)
metrics_poly <- calculate_metrics(model_poly, test_data)



## 4. Buat Tabel Perbandingan
comparison_table <- rbind(
  "model sederhana" = metrics_sederhana,
  "model berganda" = metrics_berganda,
  "model polinomial" = metrics_poly
)
print(round(comparison_table, 4))



## 5. Menentukan Model Terbaik
# Pilih Model Terbaik Berdasarkan RMSE
best_rmse <- which.min(comparison_table[,2])
best_model_name <- rownames(comparison_table)[best_rmse]



## 6. Visualisasi model terbaik
# Pilih model terbaik
best_model <- switch(best_model_name,
                     "model sederhana" = model_sederhana,
                     "model berganda" = model_berganda,
                     "model polinomial" = model_poly)

# Buat Prediksi
predictions <- predict(best_model, newdata = test_data)
```



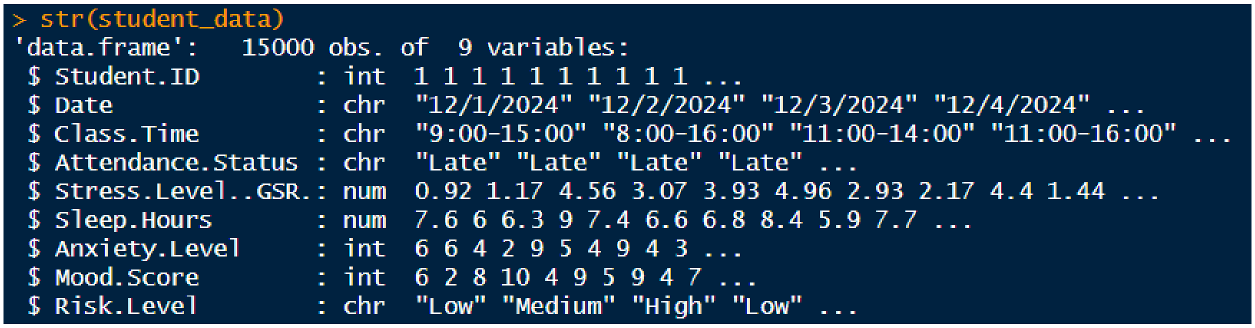
```r
# Sesuaikan panjang jika berbeda
if (length(predictions) != length(test_data$Risk.Level)) {
  predictions <- predictions[1:length(test_data$Risk.Level)]
}

# Scatter Plot: Nilai Aktual vs Prediksi
plot(test_data$Risk.Level, predictions,
     main = paste("Actual vs Predicted -", best_model_name),
     xlab = "Actual Values",
     ylab = "Predicted Values")

# Tambahkan Garis Referensi
abline(0, 1, col = "red")
```

Environment



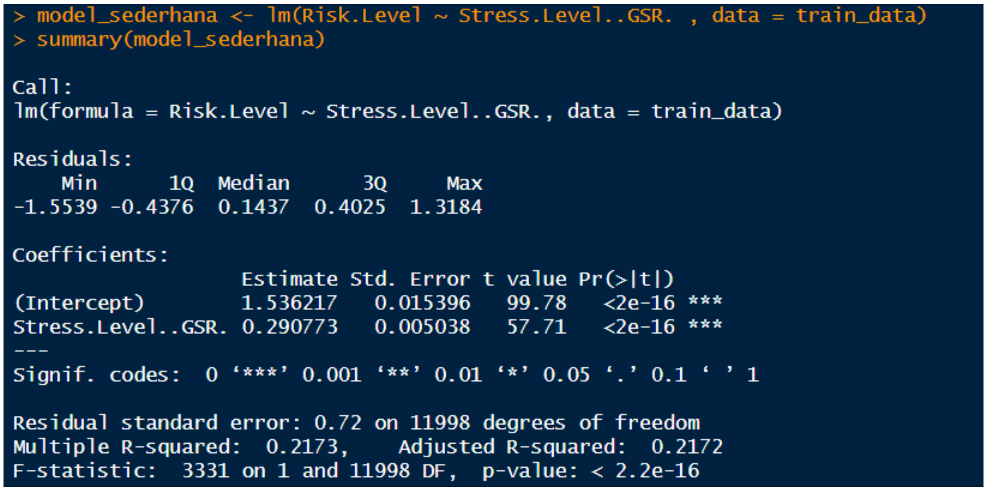
```
Data
best_model                 Large lm (12 elements,  3.5 MB)
comparison_table           num [1:3, 1:2] 0.221261 0.242309 -0.000823 0.714193 0.704475 ...
model_berganda             Large lm (12 elements,  3.5 MB)
model_poly                 Large lm (12 elements,  3.3 MB)
model_sederhana            Large lm (12 elements,  3.1 MB)
student_data               15000 obs. of 9 variables
student_tracking_data      15000 obs. of 9 variables
test_data                  3000 obs. of 9 variables
train_data                 12000 obs. of 9 variables
training_index             int [1:12000, 1] 1 2 3 5 6 7 8 9 11 13 ...
Values
best_model_name            "model berganda"
best_r2                    Named int 2
best_rmse                  Named int 2
metrics_berganda           Named num [1:2] 0.242 0.704
metrics_poly               Named num [1:2] -0.000823 0.809652
metrics_sederhana          Named num [1:2] 0.221 0.714
model_name                 "berganda"
predictions                Named num [1:3000] 2.29 1.85 2.74 2.49 2.25 ...
Functions
calculate_metrics          function (model, test_data)
```

Keterangan variabel setelah import data



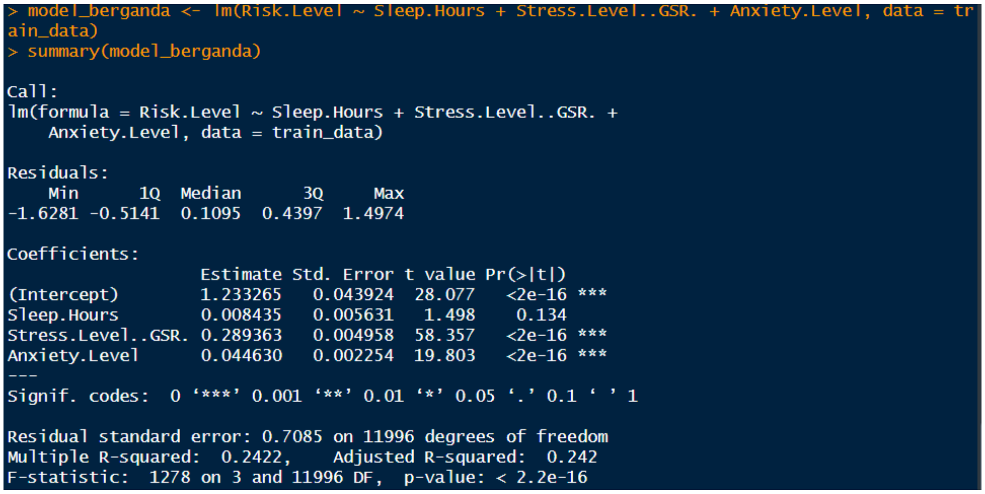
```
> str(student_data)
'data.frame':	15000 obs. of  9 variables:
 $ Student.ID         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Date               : chr  "12/1/2024" "12/2/2024" "12/3/2024" "12/4/2024" ...
 $ Class.Time         : chr  "9:00-15:00" "8:00-16:00" "11:00-14:00" "11:00-16:00" ...
 $ Attendance.Status  : chr  "Late" "Late" "Late" "Late" ...
 $ Stress.Level..GSR.: num  0.92 1.17 4.56 3.07 3.93 4.96 2.93 2.17 4.4 1.44 ...
 $ Sleep.Hours        : num  7.6 6 6.3 9 7.4 6.6 6.8 8.4 5.9 7.7 ...
 $ Anxiety.Level      : int  6 6 4 2 9 5 4 9 4 3 ...
 $ Mood.Score         : int  6 2 8 10 4 9 5 9 4 7 ...
 $ Risk.Level         : chr  "Low" "Medium" "High" "Low" ...
```

Model 1: Regresi Linear Sederhana



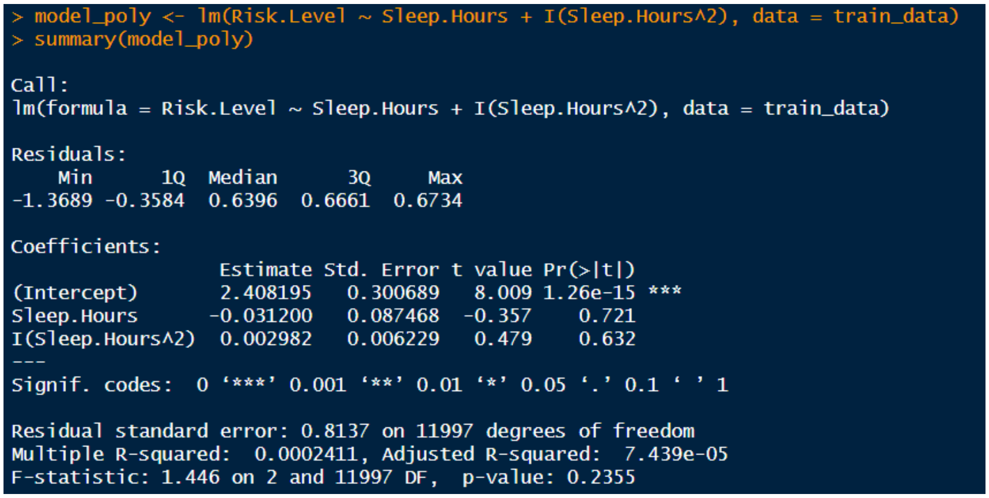
```
> model_sederhana <- lm(Risk.Level ~ Stress.Level..GSR. , data = train_data)
> summary(model_sederhana)

Call:
lm(formula = Risk.Level ~ Stress.Level..GSR., data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.5539 -0.4376  0.1437  0.4025  1.3184 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)        1.536217   0.015396   99.78   <2e-16 ***
Stress.Level..GSR. 0.290773   0.005038   57.71   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.72 on 11998 degrees of freedom
Multiple R-squared:  0.2173,	Adjusted R-squared:  0.2172 
F-statistic:  3331 on 1 and 11998 DF,  p-value: < 2.2e-16
```

Model 2: Regresi Linear Berganda



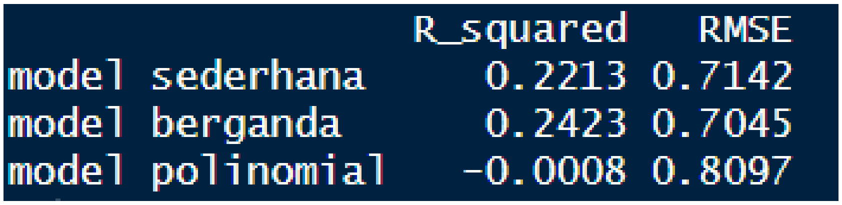
```
> model_berganda <- lm(Risk.Level ~ Sleep.Hours + Stress.Level..GSR. + Anxiety.Level, data = tr
ain_data)
> summary(model_berganda)

Call:
lm(formula = Risk.Level ~ Sleep.Hours + Stress.Level..GSR. + 
    Anxiety.Level, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.6281 -0.5141  0.1095  0.4397  1.4974 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)        1.233265   0.043924  28.077   <2e-16 ***
Sleep.Hours        0.008435   0.005631   1.498    0.134    
Stress.Level..GSR. 0.289363   0.004958  58.357   <2e-16 ***
Anxiety.Level      0.044630   0.002254  19.803   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7085 on 11996 degrees of freedom
Multiple R-squared:  0.2422,	Adjusted R-squared:  0.242 
F-statistic:  1278 on 3 and 11996 DF,  p-value: < 2.2e-16
```

Model 3: Regresi Polinomial



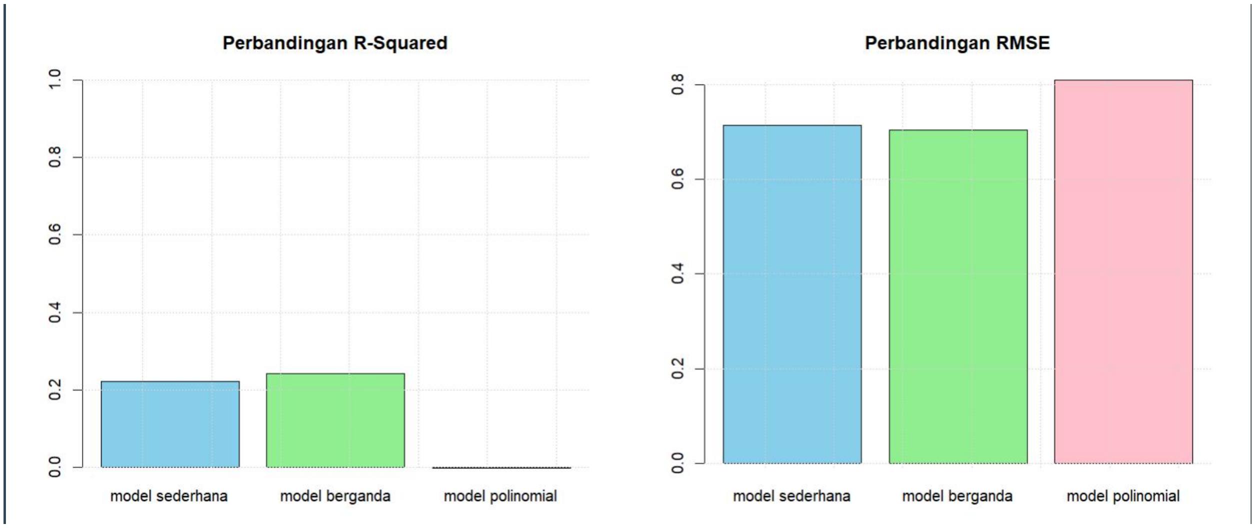
```
> model_poly <- lm(Risk.Level ~ Sleep.Hours + I(Sleep.Hours^2), data = train_data)
> summary(model_poly)

Call:
lm(formula = Risk.Level ~ Sleep.Hours + I(Sleep.Hours^2), data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.3689 -0.3584  0.6396  0.6661  0.6734 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2.408195   0.300689   8.009 1.26e-15 ***
Sleep.Hours      -0.031200   0.087468  -0.357    0.721    
I(Sleep.Hours^2)  0.002982   0.006229   0.479    0.632    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8137 on 11997 degrees of freedom
Multiple R-squared:  0.0002411,	Adjusted R-squared:  7.439e-05 
F-statistic: 1.446 on 2 and 11997 DF,  p-value: 0.2355
```

Tabel perbandingan

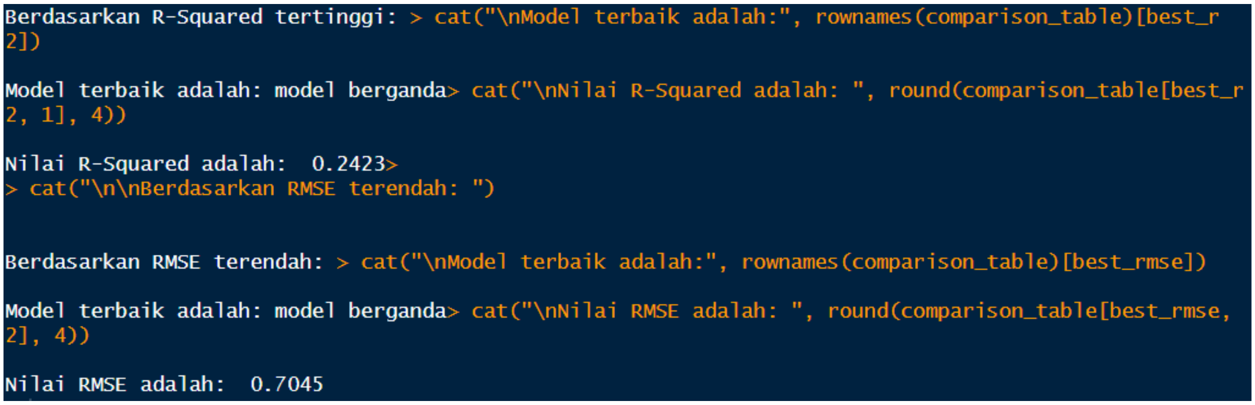


```
                 R_squared   RMSE
model sederhana     0.2213 0.7142
model berganda      0.2423 0.7045
model polinomial   -0.0008 0.8097
```

Visualisasi perbandingan metrics

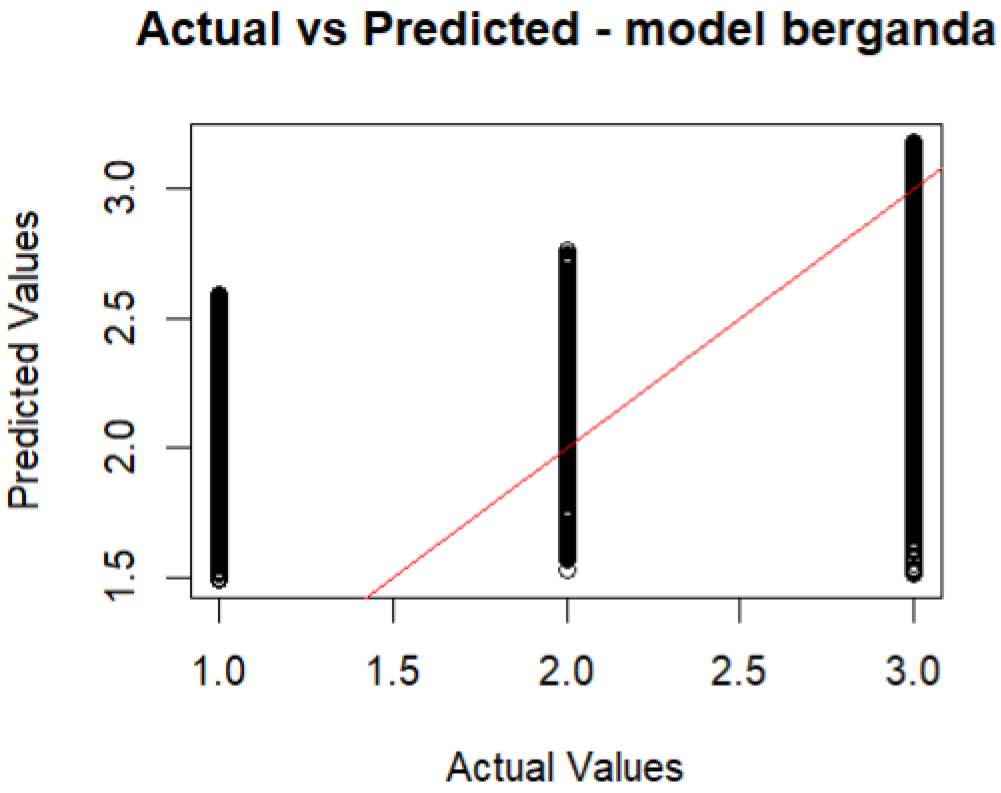


Menentukan model terbaik



```
Berdasarkan R-Squared tertinggi: > cat("\nModel terbaik adalah:", rownames(comparison_table)[best_r
2])

Model terbaik adalah: model berganda> cat("\nNilai R-Squared adalah: ", round(comparison_table[best_r
2, 1], 4))

Nilai R-Squared adalah:  0.2423>
> cat("\n\nBerdasarkan RMSE terendah: ")


Berdasarkan RMSE terendah: > cat("\nModel terbaik adalah:", rownames(comparison_table)[best_rmse])

Model terbaik adalah: model berganda> cat("\nNilai RMSE adalah: ", round(comparison_table[best_rmse,
2], 4))

Nilai RMSE adalah:  0.7045
```

Visualisasi model terbaik


Actual vs Predicted - model berganda
Predicted Values
Actual Values