Heart Failure Prediction (R) - Revisi Kritis & Robust

1. Muat Data & Split Data (Mencegah Kebocoran)

Kita harus membagi data **sebelum** melakukan pra-pemrosesan apa pun.

```
# Muat data.
df <- read.csv("heart.csv")

# Handling RestingBP = 0 (Drop rows) - Ini bisa dilakukan di awal
df <- df %>%
  filter(RestingBP > 0)

# Pisahkan data latih dan data uji (80/20)
set.seed(26)
trainIndex <- createDataPartition(df$HeartDisease, p = .8,
                                  list = FALSE,
                                  times = 1)

train_data <- df[ trainIndex,]
test_data  <- df[-trainIndex,]

dim(train_data)
```

```
## [1] 734  12
```

```
dim(test_data)
```

```
## [1] 183  12
```

2. Pra-pemrosesan (Tanpa Kebocoran)

Kita akan membuat preprocessing secara terpisah untuk data latih dan uji.

```
# --- PRA-PEMROSESAN DATA LATIH ---

# 1. Hitung median HANYA dari data latih
valid_chol_median <- train_data %>%
  filter(Cholesterol > 0) %>%
  summarise(MedianChol = median(Cholesterol, na.rm = TRUE)) %>%
  pull(MedianChol)

print(paste("Median kolesterol DARI DATA LATIH:", valid_chol_median))
```

```
## [1] "Median kolesterol DARI DATA LATIH: 237"
```

```
# 2. Terapkan median ke data latih
train_data <- train_data %>%
  mutate(Cholesterol = ifelse(Cholesterol == 0, valid_chol_median, Cholesterol))

# 3. Ubah tipe data latih menjadi factor
categorical_cols <- c('Sex', 'ChestPainType', 'RestingECG', 'ExerciseAngina', 'ST_Slope', 'FastingBS',
```

```r
train_data <- train_data %>%
  mutate(across(all_of(categorical_cols), as.factor))


# --- PRA-PEMROSESAN DATA UJI ---

# 1. Terapkan median DARI DATA LATIH ke data uji (PENTING!)
test_data <- test_data %>%
  mutate(Cholesterol = ifelse(Cholesterol == 0, valid_chol_median, Cholesterol))

# 2. Ubah tipe data uji menjadi factor
test_data <- test_data %>%
  mutate(across(all_of(categorical_cols), as.factor))

glimpse(train_data)
```

```
## Rows: 734
## Columns: 12
## $ Age            <int> 40, 37, 48, 54, 39, 45, 54, 37, 48, 58, 39, 49, 38, 43,~
## $ Sex            <fct> M, M, F, M, M, F, M, M, F, M, M, M, M, F, M, M, F, M, F~
## $ ChestPainType  <fct> ATA, ATA, ASY, NAP, NAP, ATA, ATA, ASY, ATA, ATA, ATA, ~
## $ RestingBP      <int> 140, 130, 138, 150, 120, 130, 110, 140, 120, 136, 120, ~
## $ Cholesterol    <int> 289, 283, 214, 195, 339, 237, 208, 207, 284, 164, 204, ~
## $ FastingBS      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ RestingECG     <fct> Normal, ST, Normal, Normal, Normal, Normal, Normal, Nor~
## $ MaxHR          <int> 172, 98, 108, 122, 170, 170, 142, 130, 120, 99, 145, 14~
## $ ExerciseAngina <fct> N, N, Y, N, N, N, N, Y, N, Y, N, Y, N, N, N, N, N, N, N~
## $ Oldpeak        <dbl> 0.0, 0.0, 1.5, 0.0, 0.0, 0.0, 0.0, 1.5, 0.0, 2.0, 0.0, ~
## $ ST_Slope       <fct> Up, Up, Flat, Up, Up, Up, Up, Flat, Up, Flat, Up, Flat,~
## $ HeartDisease   <fct> 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0~
```

```r
glimpse(test_data)
```

```
## Rows: 183
## Columns: 12
## $ Age            <int> 49, 37, 42, 54, 51, 53, 56, 32, 54, 35, 59, 36, 41, 41,~
## $ Sex            <fct> F, F, F, F, M, M, M, M, F, M, M, M, M, F, M, M, M, M, M~
## $ ChestPainType  <fct> NAP, NAP, NAP, ATA, ATA, NAP, NAP, ATA, NAP, ATA, NAP, ~
## $ RestingBP      <int> 160, 130, 115, 120, 125, 145, 130, 125, 130, 150, 130, ~
## $ Cholesterol    <int> 180, 211, 211, 273, 188, 518, 167, 254, 294, 264, 318, ~
## $ FastingBS      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0~
## $ RestingECG     <fct> Normal, Normal, ST, Normal, Normal, Normal, Normal, Nor~
## $ MaxHR          <int> 156, 142, 137, 150, 145, 130, 114, 155, 100, 168, 120, ~
## $ ExerciseAngina <fct> N, N, N, N, N, N, N, N, Y, N, Y, N, N, N, N, Y, N, Y, Y~
## $ Oldpeak        <dbl> 1.0, 0.0, 0.0, 1.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, ~
## $ ST_Slope       <fct> Flat, Up, Up, Flat, Up, Flat, Up, Up, Flat, Up, Flat, F~
## $ HeartDisease   <fct> 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1~
```

3. Pembangunan Model (Robust dengan Cross-Validation & Scaling)

Di sinilah letak perubahan terbesarnya.

```r
# 1. Tentukan metode validasi: 10-Fold Cross-Validation (CV)
# Ini akan menggantikan validasi 80/20 tunggal yang tidak stabil
set.seed(18)
```

```r
trControl <- trainControl(method = "cv",
                          number = 10) # 10-Fold CV

# 2. Latih model
model <- train(HeartDisease ~ .,
               data = train_data,
               method = "rf",             # RandomForest
               trControl = trControl,     # Gunakan 10-Fold CV
               preProcess = c("center", "scale") # Terapkan scaling (Best Practice)
               )

# Hasil 'model' sekarang jauh lebih bisa diandalkan.
# Ini menunjukkan rata-rata akurasi & kappa dari 10-fold CV
print(model)
```

```
## Random Forest
##
## 734 samples
##  11 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (15), scaled (15)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 661, 661, 661, 659, 660, 660, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8512929  0.6986005
##    8    0.8417954  0.6793796
##   15    0.8459230  0.6880457
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

4. Evaluasi Akhir (pada Data Uji)

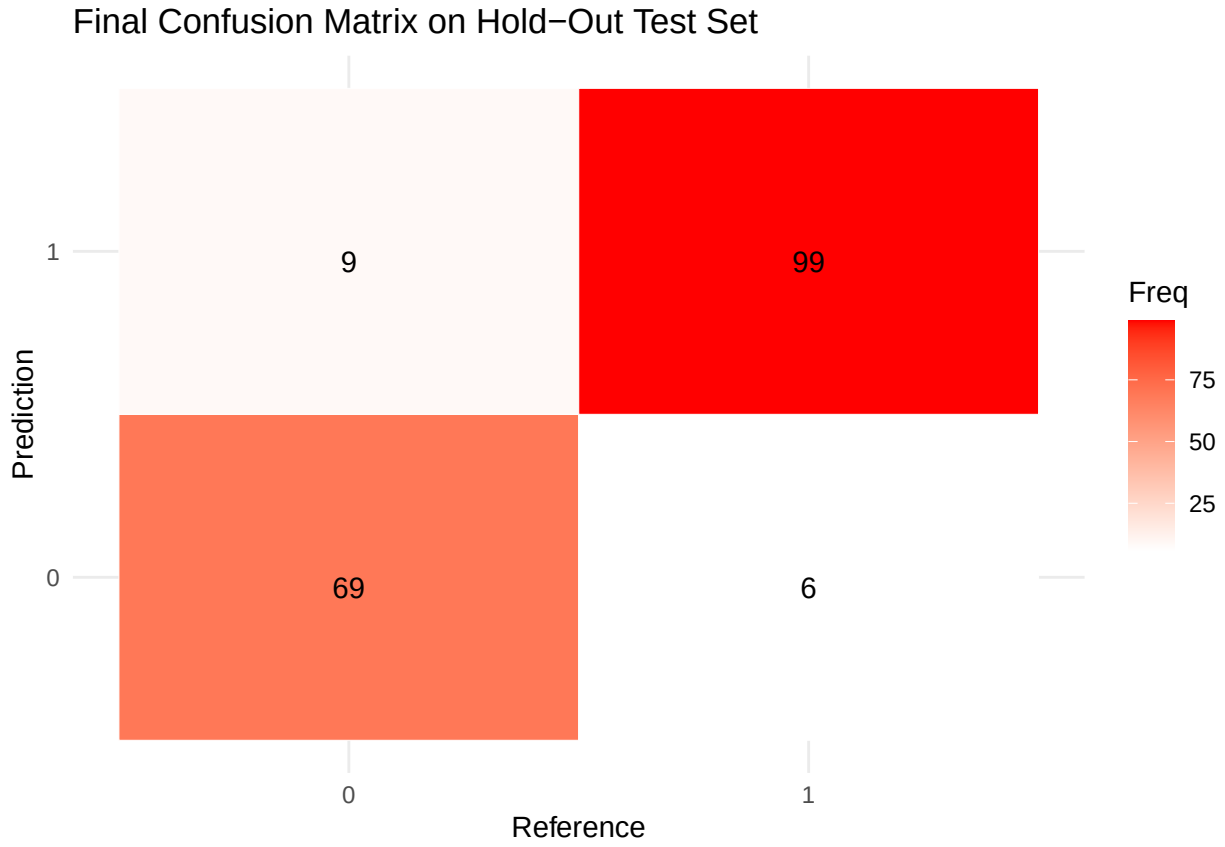
Kita tetap menggunakan data uji pada akhirnya, tetapi sekarang kita tahu bahwa model kita telah divalidasi dengan kuat.

```r
# Buat prediksi pada data uji (test_data)
y_pred <- predict(model, newdata = test_data)

# Definisikan y_test (variabel target dari test_data)
y_test <- test_data$HeartDisease

# Evaluasi model (Confusion Matrix, Accuracy, Classification Report)
cm <- confusionMatrix(y_pred, y_test)

print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 69  6
```

```
##          1  9 99
##
##                Accuracy : 0.918
##                  95% CI : (0.8684, 0.9534)
##     No Information Rate : 0.5738
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8316
##
##  Mcnemar's Test P-Value : 0.6056
##
##             Sensitivity : 0.8846
##             Specificity : 0.9429
##          Pos Pred Value : 0.9200
##          Neg Pred Value : 0.9167
##              Prevalence : 0.4262
##          Detection Rate : 0.3770
##    Detection Prevalence : 0.4098
##       Balanced Accuracy : 0.9137
##
##        'Positive' Class : 0
##
```

```r
# Visualisasi Confusion Matrix
cm_table <- as.data.frame(cm$table)

ggplot(cm_table, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Final Confusion Matrix on Hold-Out Test Set") +
  theme_minimal()
```


Final Confusion Matrix on Hold−Out Test Set
Prediction
1
0
9
99
69
6
0
1
Reference
Freq
75
50
25