

Analisis Prediktif Penyakit Jantung Menggunakan Algoritma Klasifikasi (Random Forest)

Nama Anda

2025-11-13

Contents

1. Judul Proyek	1
2. Latar Belakang Masalah	1
3. Rumusan Masalah dan Tujuan Proyek	1
4. Sumber dan Karakteristik Data	2
5. Metodologi Data Mining (CRISP-DM)	2
6. Teknik dan Algoritma yang Digunakan	5
7. Rencana Implementasi dan Alat yang Digunakan	6
8. Hasil yang Diharapkan dan Dampak	6

1. Judul Proyek

Analisis Prediktif Penyakit Jantung Menggunakan Algoritma Klasifikasi (Random Forest) Berbasis Data Klinis.

2. Latar Belakang Masalah

Penyakit jantung adalah salah satu penyebab utama kematian secara global. Deteksi dini merupakan kunci untuk penanganan yang efektif dan peningkatan prognosis pasien. Data klinis pasien (seperti usia, tekanan darah, kolesterol, dan hasil EKG) menyimpan pola tersembunyi. Dengan menerapkan teknik data mining, kita dapat membangun model prediktif untuk mengidentifikasi individu yang berisiko tinggi terkena penyakit jantung, sehingga memungkinkan intervensi medis lebih awal.

3. Rumusan Masalah dan Tujuan Proyek

Rumusan Masalah

1. Bagaimana membangun model klasifikasi dengan akurasi tinggi untuk memprediksi apakah seorang pasien menderita penyakit jantung (**HeartDisease**) berdasarkan atribut klinis yang ada?
2. Bagaimana menangani masalah kualitas data seperti nilai yang tidak logis (misalnya, Kolesterol '0') dalam dataset?
3. Faktor klinis apa yang menjadi prediktor paling penting dalam menentukan risiko penyakit jantung?

Tujuan Proyek

1. Menerapkan metodologi CRISP-DM untuk memproses dan menganalisis dataset penyakit jantung.
2. Membangun dan mengevaluasi model klasifikasi (fungsi mayor) menggunakan algoritma Random Forest.
3. Menghasilkan model yang jujur dan robust dengan menerapkan teknik validasi silang (cross-validation) dan menghindari kebocoran data (*data leakage*).
4. Mengidentifikasi variabel (fitur) yang paling berpengaruh terhadap prediksi penyakit jantung.

4. Sumber dan Karakteristik Data

- **Sumber Data:** Dataset publik “Heart Failure Prediction Dataset” yang diperoleh dari platform Kaggle.
- **Karakteristik Data:** Dataset awal terdiri dari 918 observasi (baris) dan 12 variabel (kolom).
- **Target Variabel:** HeartDisease (Biner: 1 = Sakit Jantung, 0 = Normal).
- **Variabel Prediktor:** Mencakup data demografis (Age, Sex), data klinis (RestingBP, Cholesterol), dan hasil tes (MaxHR, RestingECG, dll.).
- **Kualitas Data:** Terdapat potensi masalah kualitas data, seperti nilai ‘0’ yang tidak logis pada kolom Cholesterol dan RestingBP, yang harus ditangani pada tahap *Data Preparation*.

5. Metodologi Data Mining (CRISP-DM)

Metodologi proyek mengikuti alur standar CRISP-DM.

Tahap 1 & 2: Business & Data Understanding

Tahap ini mencakup pemahaman tujuan (telah dijelaskan di Latar Belakang) dan pemahaman data awal. Kita memuat data dan memeriksanya.

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

# Tampilkan struktur data (mirip df.info())
glimpse(df)

## Rows: 918
## Columns: 12
## $ Age          <int> 40, 49, 37, 48, 54, 39, 45, 54, 37, 48, 37, 58, 39, 49, ~
## $ Sex          <chr> "M", "F", "M", "F", "M", "M", "F", "M", "M", "F", "F", ~
## $ ChestPainType <chr> "ATA", "NAP", "ATA", "ASY", "NAP", "NAP", "ATA", "ATA", ~
## $ RestingBP    <int> 140, 160, 130, 138, 150, 120, 130, 110, 140, 120, 130, ~
## $ Cholesterol  <int> 289, 180, 283, 214, 195, 339, 237, 208, 207, 284, 211, ~
## $ FastingBS    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ RestingECG   <chr> "Normal", "Normal", "ST", "Normal", "Normal", "Normal", ~
## $ MaxHR        <int> 172, 156, 98, 108, 122, 170, 170, 142, 130, 120, 142, 9~
## $ ExerciseAngina <chr> "N", "N", "N", "Y", "N", "N", "N", "N", "Y", "N", "N", ~
## $ Oldpeak      <dbl> 0.0, 1.0, 0.0, 1.5, 0.0, 0.0, 0.0, 0.0, 1.5, 0.0, 0.0, ~
## $ ST_Slope     <chr> "Up", "Flat", "Up", "Flat", "Up", "Up", "Up", "Up", "Fl~
## $ HeartDisease <int> 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1~
```

```
# Tampilkan ringkasan statistik (mirip df.describe())
summary(df)
```

```
##      Age          Sex          ChestPainType          RestingBP
## Min.   :28.00    Length:918    Length:918          Min.    :  0.0
## 1st Qu.:47.00    Class :character  Class :character  1st Qu.:120.0
## Median :54.00    Mode  :character  Mode  :character  Median :130.0
## Mean   :53.51                                Mean   :132.4
## 3rd Qu.:60.00                                3rd Qu.:140.0
## Max.   :77.00                                Max.   :200.0
## Cholesterol    FastingBS    RestingECG          MaxHR
## Min.    :  0.0    Min.    :0.0000    Length:918          Min.    : 60.0
## 1st Qu.:173.2    1st Qu.:0.0000    Class :character    1st Qu.:120.0
## Median :223.0    Median :0.0000    Mode  :character    Median :138.0
## Mean    :198.8    Mean    :0.2331                                Mean    :136.8
```

```
## 3rd Qu.:267.0 3rd Qu.:0.0000 3rd Qu.:156.0
## Max. :603.0 Max. :1.0000 Max. :202.0
## ExerciseAngina Oldpeak ST_Slope HeartDisease
## Length:918 Min. :-2.6000 Length:918 Min. :0.0000
## Class :character 1st Qu.: 0.0000 Class :character 1st Qu.:0.0000
## Mode :character Median : 0.6000 Mode :character Median :1.0000
## Mean : 0.8874 Mean :0.5534
## 3rd Qu.: 1.5000 3rd Qu.:1.0000
## Max. : 6.2000 Max. :1.0000
```

Representasi (Temuan Awal): Dari `summary(df)`, kita melihat masalah kualitas data: 1. `RestingBP` memiliki nilai minimum 0. 2. `Cholesterol` memiliki nilai minimum 0. Kedua nilai ini tidak mungkin secara biologis dan akan kita tangani sebagai data hilang.

Tahap 3: Data Preparation

Ini adalah tahap kritis di mana kita membersihkan data, melakukan imputasi, dan membaginya.

```
# 1. Membersihkan 'RestingBP' (Drop baris karena jumlahnya sedikit)
df_clean <- df %>%
  filter(RestingBP > 0)
print(paste("Jumlah baris setelah membersihkan RestingBP:", nrow(df_clean)))

## [1] "Jumlah baris setelah membersihkan RestingBP: 917"

# 2. Split Data (WAJIB dilakukan SEBELUM imputasi untuk mencegah data leakage)
set.seed(26)
trainIndex <- createDataPartition(df_clean$HeartDisease, p = .8,
                                   list = FALSE,
                                   times = 1)
train_data <- df_clean[ trainIndex,]
test_data <- df_clean[-trainIndex,]

# 3. Imputasi 'Cholesterol = 0'
# 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"

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

test_data <- test_data %>%
  mutate(Cholesterol = ifelse(Cholesterol == 0, valid_chol_median, Cholesterol))

# 4. Konversi Variabel Kategorikal menjadi Factor
categorical_cols <- c('Sex', 'ChestPainType', 'RestingECG', 'ExerciseAngina', 'ST_Slope', 'FastingBS',

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

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

# Tampilkan struktur data latih setelah bersih
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~
## $ 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~
## $ 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~
```

Tahap 4: Modeling

Kita melatih model menggunakan data latih (`train_data`) dengan metode 10-Fold Cross-Validation (CV) untuk memastikan model stabil dan *robust*.

```
# Tentukan metode kontrol: 10-Fold CV
set.seed(18)
trControl <- trainControl(method = "cv",
  number = 10)

# Latih model Random Forest
# Kita juga menambahkan 'preProcess' untuk scaling (fungsi minor)
model_rf <- train(HeartDisease ~ .,
  data = train_data,
  method = "rf",
  trControl = trControl,
  preProcess = c("center", "scale")
)

# Representasi (Hasil Cross-Validation):
# Menampilkan akurasi rata-rata dari 10-fold CV
print(model_rf)
```

```
## 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.
```

Tahap 5: Evaluation

Model yang sudah dilatih (`model_rf`) kini dievaluasi kinerjanya menggunakan data uji (`test_data`) yang belum pernah dilihat sebelumnya.

```
# Buat prediksi pada data uji
y_pred <- predict(model_rf, newdata = test_data)
y_test <- test_data$HeartDisease

# Representasi (Confusion Matrix dan Statistik Lengkap):
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
##
```

6. Teknik dan Algoritma yang Digunakan

- **Algoritma Utama:** Random Forest
- **Fungsi Mayor:** Klasifikasi. Tujuan model adalah mengklasifikasikan pasien ke dalam dua kelas: '0' (Sehat) atau '1' (Sakit Jantung).
- **Fungsi Minor:**

- **Pembersihan Data:** Menghapus baris dengan `RestingBP = 0`.
- **Imputasi:** Mengganti nilai `Cholesterol = 0` dengan **Median** dari data latih (bukan *mean*, untuk menghindari bias dari *outlier*).
- **Transformasi Tipe Data:** Mengubah variabel objek (`char`) menjadi *factor* agar dapat diproses oleh R.
- **Normalisasi (Scaling):** Menggunakan `center` dan `scale` (standarisasi Z-score) pada fitur numerik selama proses *training* untuk membantu beberapa algoritma (meskipun dampaknya minimal pada Random Forest, ini adalah *best practice*).

7. Rencana Implementasi dan Alat yang Digunakan

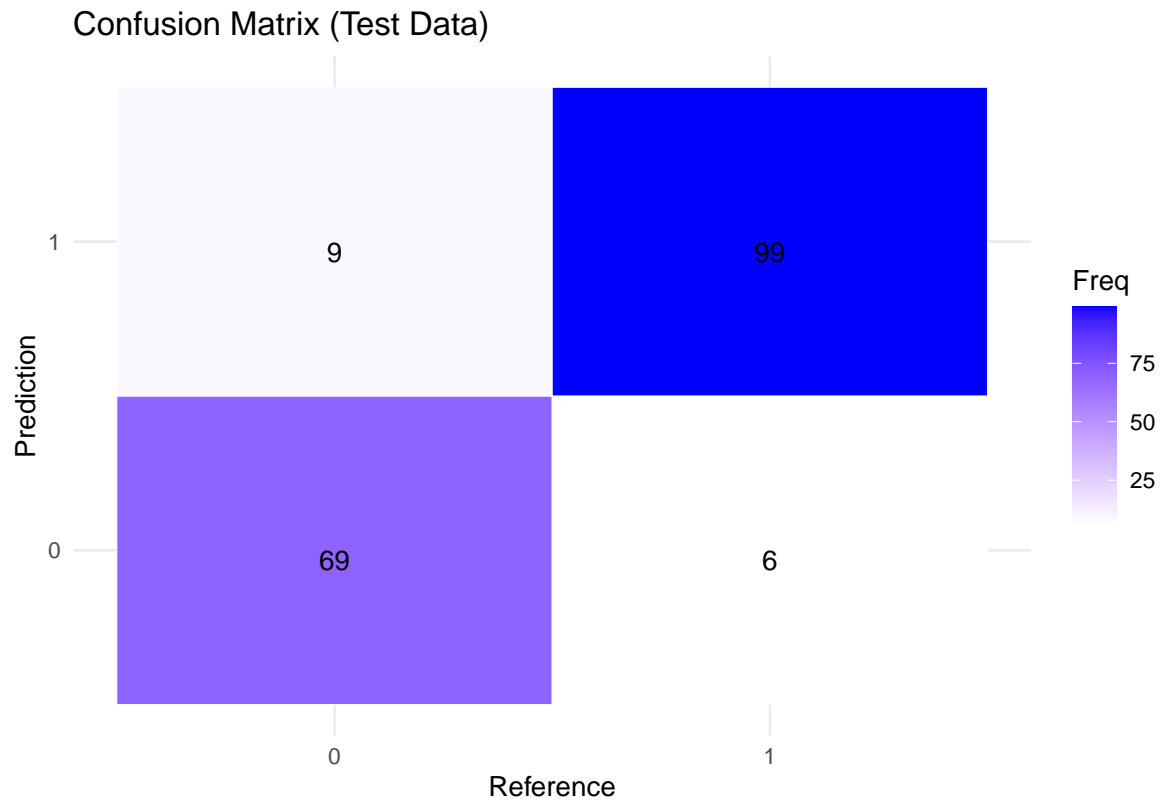
- **Alat Utama:** RStudio
- **Bahasa:** R
- **Paket (Library) Kunci:**
 - `tidyverse` (untuk manipulasi data/Data Preparation).
 - `caret` (untuk proses *split data*, *modeling*, *preprocessing*, dan *evaluation*).
 - `randomForest` (sebagai *engine* algoritma utama).
 - `ggplot2` (untuk visualisasi).

8. Hasil yang Diharapkan dan Dampak

Hasil Proyek (Representasi Output)

1. **Model Prediktif:** Sebuah model R (`model_rf`) yang siap digunakan, yang terbukti memiliki akurasi **0.918** dan Kappa **0.832** pada data uji.
2. **Visualisasi Hasil (Confusion Matrix):**

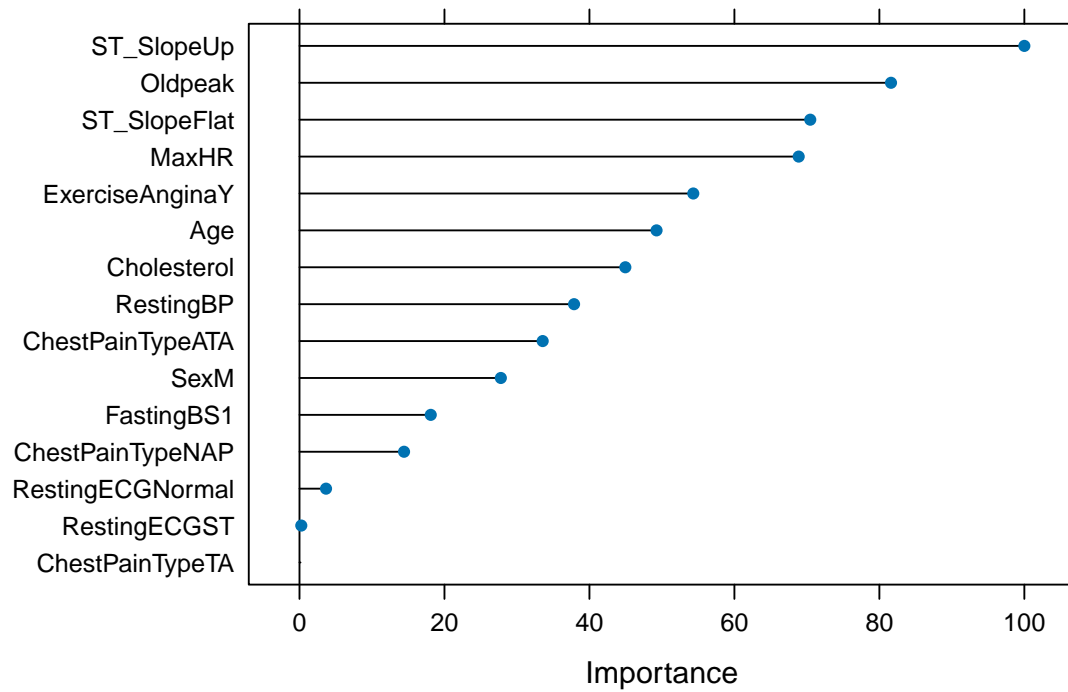
```
# Plot heatmap confusion matrix
cm_table <- as.data.frame(cm$table)
ggplot(cm_table, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), vjust = 1) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Confusion Matrix (Test Data)") +
  theme_minimal()
```



3. **Insight Kunci (Variable Importance):** Kita dapat mengekstrak fitur apa yang dianggap paling penting oleh model.

```
# Plot variable importance  
var_imp <- varImp(model_rf)  
plot(var_imp, main = "Variable Importance (Faktor Paling Berpengaruh)")
```

Variable Importance (Faktor Paling Berpengaruh)



Dampak dan Potensi Penerapan

Berdasarkan hasil di atas (terutama *Variable Importance*), kita dapat memberikan *insight* bagi praktisi medis. Model ini dapat digunakan sebagai **Sistem Pendukung Keputusan (Decision Support System)** untuk *screening* awal pasien. Pasien yang diprediksi memiliki risiko tinggi dapat segera dirujuk untuk pemeriksaan lebih lanjut, sehingga mempercepat proses diagnosis dan penanganan.