Irham Maulana Johani, Hermawan Sentyaki Sarjito, Jonathan

2025-12-08

Daftar Isi

Daftar Gambar

Judul Proyek

Analisis Prediktif Penyakit Jantung Menggunakan Seleksi Fitur dan Algoritma Klasifikasi (Random Forest).

1 Pendahuluan

1.1 Latar Belakang Masalah

Gagal jantung merupakan salah satu penyakit kardiovaskular kronis dengan angka morbiditas dan mortalitas yang tinggi secara global, sehingga menjadi beban substansial bagi sistem kesehatan [Savarese

et al., 2022]. Studi epidemiologi menunjukkan peningkatan konsisten insidensi gagal jantung, khususnya di kawasan Asia dan negara berpendapatan menengah [Feng et al., 2024]. Kondisi ini tidak hanya meningkatkan angka hospitalisasi dan readmission, tetapi juga menurunkan kualitas hidup pasien serta menambah beban ekonomi masyarakat dan fasilitas pelayanan kesehatan [Yan et al., 2023]. Dalam konteks tersebut, identifikasi dini pasien berisiko tinggi menjadi krusial untuk menekan angka rawat inap berulang dan mortalitas.

Dalam praktik klinis, stratifikasi risiko pasien gagal jantung masih banyak bergantung pada penilaian subjektif klinisi dan pedoman klinis yang dikombinasikan dengan interpretasi manual berbagai parameter klinis. Berbagai penelitian menunjukkan bahwa pasien gagal jantung memiliki angka readmission dan mortalitas jangka pendek yang tinggi, yang mengindikasikan belum optimalnya stratifikasi risiko di layanan kesehatan [Sabouri et al., 2023]. Akibatnya, sebagian pasien dengan profil risiko tinggi berpotensi tidak teridentifikasi secara tepat waktu sehingga tidak memperoleh prioritas pemantauan dan intervensi intensif.

Sejalan dengan perkembangan ilmu data dan komputasi, berbagai algoritma klasifikasi seperti logistic regression, random forest, support vector machine, dan gradient boosting telah diterapkan untuk diagnosis, prediksi, dan prognosis gagal jantung dengan performa yang umumnya lebih baik dibandingkan skor risiko tradisional [Saqib et al., 2024]. Studi lain yang memanfaatkan Heart Failure Prediction Dataset menunjukkan bahwa perbandingan beberapa model klasifikasi memungkinkan identifikasi algoritma yang optimal untuk klasifikasi gagal jantung [Chulde-Fernández et al., 2025]. Di Indonesia, penelitian serupa telah mengevaluasi berbagai model *machine learning* untuk memprediksi keparahan dan readmission pasien gagal jantung dan mengidentifikasi model dengan performa terbaik untuk diintegrasikan ke dalam aplikasi pemantauan mandiri pasien [Indriany et al., 2024]. Namun demikian, masih diperlukan analisis lebih lanjut terhadap model klasifikasi berbasis Random Forest untuk mengidentifikasi pasien berisiko tinggi gagal jantung dan mengelaborasi faktor-faktor klinis yang berkontribusi terhadap kategori risiko tersebut.

1.2 Rumusan Masalah

1. Bagaimana karakteristik data klinis pasien gagal jantung ditinjau dari distribusi kelas risiko serta variabel klinis yang menyertainya?
2. Bagaimana kinerja model klasifikasi Random Forest dalam mengidentifikasi pasien berisiko tinggi gagal jantung berdasarkan metrik evaluasi seperti akurasi, presisi, *recall*, dan F1-score?
3. Fitur-fitur klinis apa saja yang paling berkontribusi terhadap penentuan kategori pasien berisiko tinggi gagal jantung berdasarkan hasil analisis model Random Forest?

1.3 Tujuan Penelitian

1. Mendeskripsikan karakteristik data klinis pasien gagal jantung, termasuk distribusi kelas risiko dan variabel-variabel klinis yang relevan.
2. Membangun model klasifikasi Random Forest untuk mengidentifikasi pasien berisiko tinggi gagal jantung menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, dan F1-score.
3. Mengidentifikasi fitur-fitur klinis yang paling berpengaruh terhadap penentuan kategori pasien berisiko tinggi gagal jantung berdasarkan hasil analisis model Random Forest.

1.4 Sumber dan Karakteristik Data

- **Sumber Data:** Dataset publik "Heart Failure Prediction Dataset" (Kaggle).
- **Karakteristik Data:** 918 observasi dan 12 variabel.

- **Target Variabel:** `HeartDisease` (Biner: 1 = Sakit Jantung, 0 = Normal).
- **Kualitas Data:** Terdapat nilai '0' yang tidak logis pada `Cholesterol` dan `RestingBP`.

1.5 Metodologi Data Mining (CRISP-DM)

1.5.1 Tahap 1 & 2: Business & Data Understanding

Tahap ini merupakan fondasi dari keseluruhan proyek. Pada tahap ini dilakukan pemahaman konteks bisnis (dalam hal ini, domain medis) dan karakteristik data yang dianalisis.

Tujuan tahap ini:

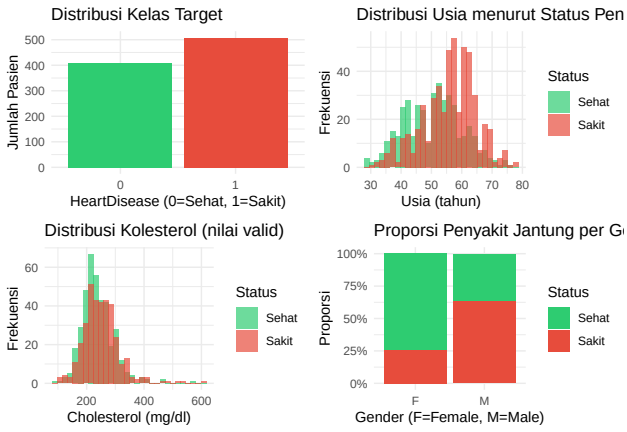
- Memuat dataset dan melakukan eksplorasi awal
- Mengidentifikasi tipe data setiap variabel (numerik vs kategorikal)
- Mendeteksi potensi masalah kualitas data (nilai hilang, outlier, nilai tidak logis)
- Memahami distribusi data untuk setiap variabel

```
df <- read.csv("heart.csv")
```

Secara umum, dataset terdiri atas 918 observasi dengan 12 variabel, yang mencakup kombinasi fitur numerik (`Age`, `RestingBP`, `Cholesterol`, `MaxHR`, `Oldpeak`) dan kategorikal (`Sex`, `ChestPainType`, `RestingECG`, `ExerciseAngina`, `ST_Slope`, `FastingBS`). Variabel target adalah `HeartDisease` (0 = tidak sakit, 1 = sakit jantung). Ringkasan statistik deskriptif mengidentifikasi dua anomali utama, yaitu nilai 0 pada `RestingBP` dan `Cholesterol` yang tidak logis secara fisiologis dan harus ditangani pada tahap *data preparation* untuk mencegah bias pemodelan.

```
library(gridExtra)
library(scales)

p1 <- ggplot(df, aes(x = factor(HeartDisease), fill = factor(HeartDisease))) +
  geom_bar(stat = "count") +
  scale_fill_manual(values = c("0" = "#2ecc71", "1" = "#e74c3c")) +
  labs(
    title = "Distribusi Kelas Target",
    x = "HeartDisease (0=Sehat, 1=Sakit)",
    y = "Jumlah Pasien"
  ) +
  theme_minimal() +
  theme(legend.position = "none")

p2 <- ggplot(df, aes(x = Age, fill = factor(HeartDisease))) +
  geom_histogram(bins = 30, alpha = 0.7, position = "identity") +
  scale_fill_manual(
    values = c("0" = "#2ecc71", "1" = "#e74c3c"),
    labels = c("Sehat", "Sakit")
  ) +
  labs(
    title = "Distribusi Usia menurut Status Penyakit",
    x = "Usia (tahun)", y = "Frekuensi", fill = "Status"
  ) +
  theme_minimal()
```

```r
p3 <- ggplot(
  df %>% dplyr::filter(Cholesterol > 0),
  aes(x = Cholesterol, fill = factor(HeartDisease))
) +
  geom_histogram(bins = 30, alpha = 0.7, position = "identity") +
  scale_fill_manual(
    values = c("0" = "#2ecc71", "1" = "#e74c3c"),
    labels = c("Sehat", "Sakit")
  ) +
  labs(
    title = "Distribusi Kolesterol (nilai valid)",
    x = "Cholesterol (mg/dl)", y = "Frekuensi", fill = "Status"
  ) +
  theme_minimal()

p4 <- ggplot(df, aes(x = Sex, fill = factor(HeartDisease))) +
  geom_bar(position = "fill") +
  scale_fill_manual(
    values = c("0" = "#2ecc71", "1" = "#e74c3c"),
    labels = c("Sehat", "Sakit")
  ) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Proporsi Penyakit Jantung per Gender",
    x = "Gender (F=Female, M=Male)",
    y = "Proporsi", fill = "Status"
  ) +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```

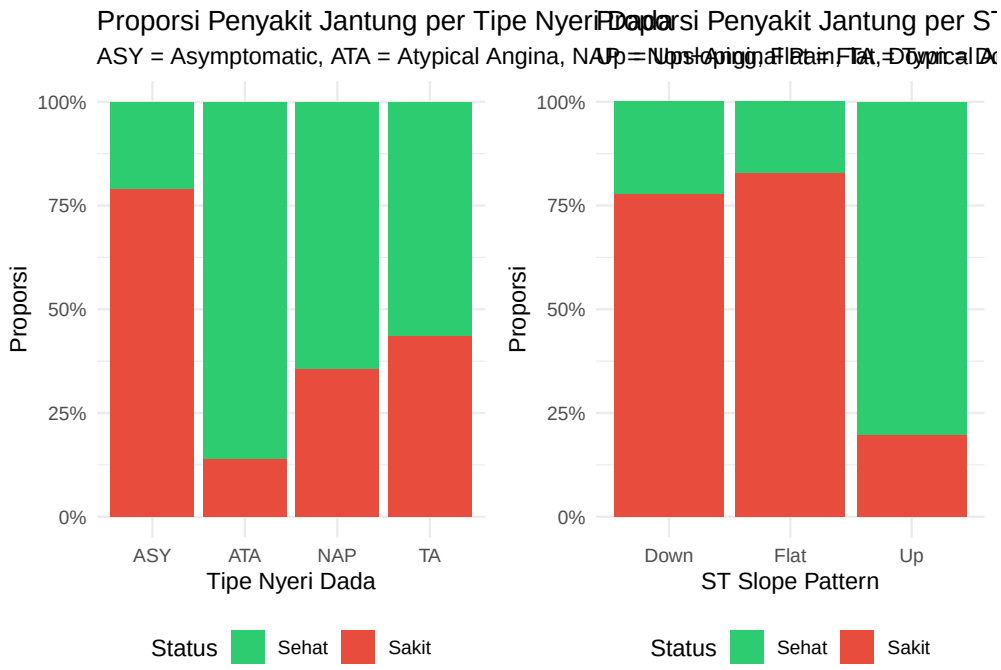


Distribusi `HeartDisease` relatif seimbang sehingga sesuai untuk pemodelan klasifikasi tanpa penyesuaian khusus ketidakseimbangan kelas. Pasien dengan penyakit jantung cenderung berusia lebih tua dan memiliki kadar kolesterol yang lebih tinggi, serta proporsi penyakit jantung pada laki-laki lebih tinggi

dibandingkan perempuan.

Analisis Fitur Kategorikal Tambahan:

```
# Analisis ChestPainType dan ST_Slope
p_chest <- ggplot(df, aes(x = ChestPainType, fill = factor(HeartDisease))) +
  geom_bar(position = "fill") +
  scale_fill_manual(
    values = c("0" = "#2ecc71", "1" = "#e74c3c"),
    labels = c("Sehat", "Sakit")
  ) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Proporsi Penyakit Jantung per Tipe Nyeri Dada",
    subtitle = "ASY = Asymptomatic, ATA = Atypical Angina, NAP = Non-Anginal Pain, TA = Typical
    x = "Tipe Nyeri Dada",
    y = "Proporsi",
    fill = "Status"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")

p_slope <- ggplot(df, aes(x = ST_Slope, fill = factor(HeartDisease))) +
  geom_bar(position = "fill") +
  scale_fill_manual(
    values = c("0" = "#2ecc71", "1" = "#e74c3c"),
    labels = c("Sehat", "Sakit")
  ) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Proporsi Penyakit Jantung per ST Slope",
    subtitle = "Up = Upsloping, Flat = Flat, Down = Downsloping",
    x = "ST Slope Pattern",
    y = "Proporsi",
    fill = "Status"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")

grid.arrange(p_chest, p_slope, ncol = 2)
```



Analisis fitur kategorikal menunjukkan bahwa pasien dengan tipe nyeri dada asimtomatik (ASY) memiliki proporsi penyakit jantung yang sangat tinggi, yang konsisten dengan fenomena *silent heart disease*. Selain itu, pola `ST_Slope` datar maupun menurun berkaitan dengan proporsi penyakit jantung yang lebih tinggi dibandingkan pola menanjak, sejalan dengan literatur klinis mengenai hasil *exercise stress test*.

1.5.2 Tahap 3: Data Preparation

Tahap ini adalah tahap paling kritis dan memakan waktu dalam proyek data mining (sering 60-80% dari total waktu). Kualitas model sangat bergantung pada kualitas data yang digunakan untuk melatihnya.

Strategi Pembersihan Data:

1. **Penanganan `RestingBP = 0`:**
   - Jumlah baris dengan nilai 0 relatif sedikit ($<1\%$)
   - Keputusan: **Hapus baris** (deletion) karena tidak ada informasi yang dapat digunakan untuk imputasi yang akurat
   - Alternatif seperti mean/median imputation tidak tepat karena variabel ini sangat vital untuk diagnosis jantung
2. **Penanganan `Cholesterol = 0`:**
   - Jumlah baris cukup banyak ($\sim$18%), menghapus semua akan mengurangi data signifikan
   - Keputusan: **Imputasi dengan Median** dari nilai valid
   - Median dipilih karena lebih robust terhadap outlier dibanding mean
   - **PENTING:** Median dihitung **HANYA dari data training** untuk menghindari data leakage
3. **Urutan Operasi yang Benar:**
   - Bersihkan `RestingBP` dulu
   - **Split data train/test** (80/20)
   - Hitung median dari train set
   - Terapkan imputasi ke train dan test menggunakan median yang sama

- Konversi tipe data kategorikal ke factor

Urutan ini krusial untuk menjaga integritas validasi model.

```
df_clean <- df %>%
  filter(RestingBP > 0)

set.seed(26)
trainIndex <- createDataPartition(df_clean$HeartDisease,
  p = .8,
  list = FALSE,
  times = 1
)
train_data <- df_clean[trainIndex, ]
test_data <- df_clean[-trainIndex, ]

valid_chol_median <- train_data %>%
  filter(Cholesterol > 0) %>%
  summarise(MedianChol = median(Cholesterol, na.rm = TRUE)) %>%
  pull(MedianChol)

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))

categorical_cols <- c("Sex", "ChestPainType", "RestingECG", "ExerciseAngina", "ST_Slope", "Fast
train_data <- train_data %>%
  mutate(across(all_of(categorical_cols), as.factor))
test_data <- test_data %>%
  mutate(across(all_of(categorical_cols), as.factor))

print("Konversi ke factor selesai. Struktur data latih:")
```

```
[1] "Konversi ke factor selesai. Struktur data latih:"
```

```
glimpse(train_data)
```

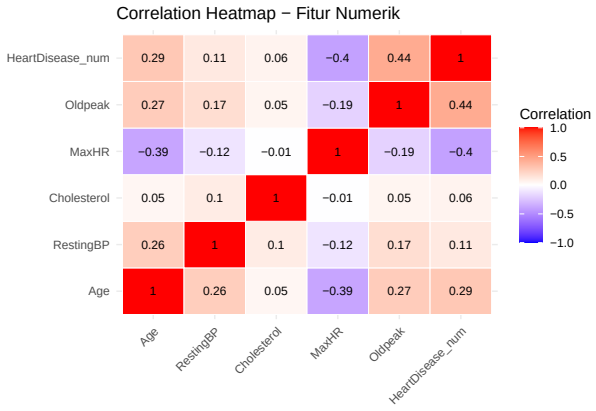
```
Rows: 734
Columns: 12
$ Age            <int> 40, 37, 48, 54, 39, 45, 54, 37, 48, 58, 39, 49, 38, 43,~
$ Sex            <fct> M, M, F, M, M, F, M, M, F, M, M, M, M, F, M, M, F, M, F~
$ ChestPainType  <fct> ATA, ATA, ASY, NAP, NAP, ATA, ATA, ASY, ATA, ATA, ATA, ~
$ RestingBP      <int> 140, 130, 138, 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~
```

Analisis Korelasi Fitur Numerik:

Sebelum pemodelan, korelasi antar fitur numerik ditinjau untuk mendeteksi potensi multikolinearitas.

```r
library(reshape2)

numeric_features <- train_data %>%
  select(Age, RestingBP, Cholesterol, MaxHR, Oldpeak) %>%
  mutate(HeartDisease_num = as.numeric(train_data$HeartDisease) - 1)

cor_matrix <- cor(numeric_features, use = "complete.obs")

cor_melted <- melt(cor_matrix)

ggplot(cor_melted, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(value, 2)), size = 3) +
  scale_fill_gradient2(
    low = "blue", high = "red", mid = "white",
    midpoint = 0, limit = c(-1, 1),
    name = "Correlation"
  ) +
  labs(
    title = "Correlation Heatmap - Fitur Numerik",
    x = "", y = ""
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Interpretasi Korelasi:

- **Oldpeak vs HeartDisease:** Korelasi positif moderat - semakin tinggi ST depression, semakin tinggi risiko
- **MaxHR vs HeartDisease:** Korelasi negatif - pasien sakit cenderung tidak bisa mencapai heart

rate tinggi

- **Age vs MaxHR:** Korelasi negatif alami - usia bertambah, max HR menurun
- **Multicollinearity:** Tidak ada korelasi tinggi antar prediktor (semua <0.6), bagus untuk modeling

Ringkasan Statistik Data Setelah Preprocessing:

```
library(knitr)

summary_stats <- train_data %>%
  select(Age, RestingBP, Cholesterol, MaxHR, Oldpeak) %>%
  summarise(
    across(
      everything(),
      list(
        Mean = ~ round(mean(.), 2),
        SD = ~ round(sd(.), 2),
        Min = ~ round(min(.), 2),
        Max = ~ round(max(.), 2)
      )
    )
  ) %>%
  pivot_longer(everything(), names_to = "Metric", values_to = "Value") %>%
  separate(Metric, into = c("Variable", "Statistic"), sep = "_") %>%
  pivot_wider(names_from = Statistic, values_from = Value)

kable(summary_stats, caption = "Statistik Deskriptif Fitur Numerik (Train Set)")
```

Tabel 1: Statistik Deskriptif Fitur Numerik (Train Set)

Variable	Mean	SD	Min	Max
Age	53.42	9.36	29.0	77.0
RestingBP	132.37	17.50	80.0	200.0
Cholesterol	242.65	51.82	85.0	603.0
MaxHR	137.08	25.42	63.0	202.0
Oldpeak	0.90	1.06	-2.6	6.2

Interpretasi:

- Semua nilai dalam range yang masuk akal (tidak ada outlier ekstrem)
- Cholesterol: min > 0 (imputasi berhasil)
- RestingBP: min > 0 (cleaning berhasil)
- Data siap untuk modeling

1.5.3 Tahap 4: Modeling (Iterasi 1 - Identifikasi Fitur)

Mengapa Model Baseline Penting?

Sebelum dilakukan seleksi fitur, terlebih dahulu dibangun model baseline yang dilatih pada **seluruh fitur** untuk: 1. Memperoleh tolok ukur awal performa model, 2. Mengidentifikasi fitur yang berkontribusi signifikan (melalui *variable importance*), dan 3. Memahami pola keterkaitan umum dalam data.

Pilihan Teknis:

- **Algoritma:** Random Forest
  - Robust terhadap overfitting
  - Dapat menangani interaksi kompleks antar fitur
  - Memberikan metrik Variable Importance secara natural
- **Validasi:** 10-Fold Cross-Validation
  - Data dibagi menjadi 10 bagian (folds)
  - Model dilatih 10 kali, setiap kali menggunakan 9 fold untuk training dan 1 fold untuk validasi
  - Hasil akhir adalah rata-rata dari 10 iterasi
  - Ini memberikan estimasi performa yang lebih reliable dibanding single train/test split
- **Preprocessing:** Standardisasi (`center` dan `scale`)
  - Mengubah fitur numerik menjadi mean=0, sd=1
  - Meskipun Random Forest tidak sensitif terhadap skala, ini adalah best practice
  - Memudahkan perbandingan apabila di kemudian hari digunakan algoritma lain

```
set.seed(18)
trControl <- trainControl(method = "cv", number = 10)

model_rf_baseline <- train(HeartDisease ~ .,
  data = train_data,
  method = "rf",
  trControl = trControl,
  preProcess = c("center", "scale"),
  importance = TRUE
)
# Hasil training model baseline (11 fitur)
print(model_rf_baseline)
```

```
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    0.8567538  0.7098695
   8    0.8540871  0.7046561
  15    0.8500691  0.6966050
```

```
Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

```r
# Parameter best model
print(paste("Optimal mtry:", model_rf_baseline$bestTune$mtry))
```

```
[1] "Optimal mtry: 2"
```

```r
print(paste("Akurasi CV Terbaik:", round(max(model_rf_baseline$results$Accuracy), 4)))
```

```
[1] "Akurasi CV Terbaik: 0.8568"
```

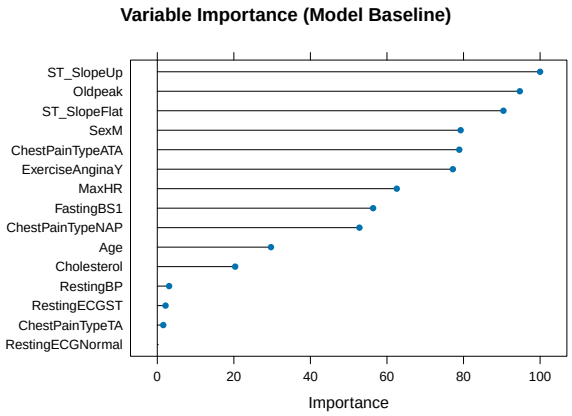
```r
print(paste("Kappa CV Terbaik:", round(max(model_rf_baseline$results$Kappa), 4)))
```

```
[1] "Kappa CV Terbaik: 0.7099"
```

```r
var_imp_baseline <- varImp(model_rf_baseline)

# Variable Importance (baseline)
print(var_imp_baseline)
```

```
rf variable importance

                 Importance
ST_SlopeUp          100.000
Oldpeak              94.706
ST_SlopeFlat         90.425
SexM                 79.250
ChestPainTypeATA     78.893
ExerciseAnginaY      77.212
MaxHR                62.556
FastingBS1           56.383
ChestPainTypeNAP     52.803
Age                  29.695
Cholesterol          20.342
RestingBP             3.087
RestingECGST          2.161
ChestPainTypeTA       1.558
RestingECGNormal      0.000
```

```r
plot(var_imp_baseline, main = "Variable Importance (Model Baseline)")
```



1.5.4 Tahap 5: Seleksi Fitur (Feature Selection)

Interpretasi Variable Importance:

Berdasarkan plot `varImp` dari model baseline di atas, terlihat pola yang jelas sebagai berikut:

Fitur Penting (High Importance):

- `ST_Slope`: Slope dari segmen ST pada ECG saat exercise - indikator kuat untuk iskemia jantung
- `Oldpeak`: Depresi ST yang diinduksi oleh exercise - parameter ECG krusial
- `MaxHR`: Maximum heart rate yang dicapai - mencerminkan kapasitas kardiovaskular
- `ExerciseAngina`: Nyeri dada saat exercise - gejala klasik penyakit jantung
- `Age`: Faktor risiko utama yang tidak dapat dimodifikasi
- `ChestPainType`: Tipe nyeri dada - informasi diagnostik penting
- `Sex`: Pria memiliki risiko lebih tinggi pada usia lebih muda
- `Cholesterol` dan `RestingBP`: Parameter klinis standar

Fitur Tidak Penting (Low Importance, mendekati 0):

- `FastingBS` (Fasting Blood Sugar $>$ 120 mg/dl): Importance sekitar 0-5%
- Meskipun diabetes adalah faktor risiko, variabel binary ini terlalu kasar
- Tidak menangkap nuansa kadar gula darah yang sebenarnya
  - `RestingECG` (Hasil ECG Istirahat): Importance sekitar 0-5%
  - Mayoritas orang sehat bisa memiliki abnormalitas minor pada ECG istirahat
  - ECG saat exercise (seperti `ST_Slope`, `Oldpeak`) lebih informatif

Keputusan Seleksi Fitur:

Fitur `FastingBS` dan `RestingECG` dikeluarkan dari model final karena:

1. Kontribusi prediktif mereka minimal (noise $>$ signal)
2. Mengurangi kompleksitas model (dari 11 fitur menjadi 9 fitur)
3. Berpotensi mengurangi overfitting
4. Dalam praktik klinis: mengurangi tes yang diperlukan (menghemat waktu dan biaya)

Metode Seleksi:

Ini adalah **filter method** berbasis importance score. Alternatif lain:

- Wrapper methods (RFE - Recursive Feature Elimination): lebih akurat tapi lebih lambat

- Embedded methods: menggunakan regularisasi (Lasso) - tidak tersedia di Random Forest standar

Dataset baru kemudian dibentuk hanya dengan berisi sembilan fitur terpilih beserta variabel target.

```
selected_features <- c(
  "ST_Slope", "Oldpeak", "MaxHR", "ExerciseAngina",
  "Age", "Cholesterol", "RestingBP", "ChestPainType",
  "Sex",
  "HeartDisease"
)

train_data_final <- train_data %>%
  select(all_of(selected_features))

test_data_final <- test_data %>%
  select(all_of(selected_features))
```

1.5.5 Tahap 6: Modeling (Iterasi 2 - Model Final)

Prinsip Re-training:

Setelah seleksi fitur dilakukan, model baseline tidak digunakan secara langsung. Model baru perlu dilatih ulang dari awal dengan alasan:

1. **Random Forest membangun pohon berdasarkan subset fitur random**
   - Ketika ada 11 fitur, setiap split node mempertimbangkan sqrt(11) ~ 3-4 fitur random
   - Ketika hanya 9 fitur, setiap split mempertimbangkan sqrt(9) = 3 fitur
   - Struktur pohon akan berbeda secara fundamental
2. **Menghilangkan Noise Membuat Model Lebih Fokus**
   - Tanpa `FastingBS` dan `RestingECG`, algoritma tidak akan "terdistraksi"
   - Interaksi antar fitur yang tersisa dapat dipelajari lebih dalam
3. **Menghindari Bias**
   - Model baseline sudah "melihat" semua fitur selama training
   - Model final harus belajar dari dataset yang lebih bersih sejak awal

Ekspektasi:

- **Skenario Terbaik:** Akurasi CV model final model baseline
  - Ini menunjukkan fitur yang dihapus memang noise
  - Model lebih efisien tanpa kehilangan performa
- **Skenario Realistis:** Akurasi CV turun sedikit (<1-2%)
  - Trade-off yang wajar untuk kesederhanaan model
  - Masih acceptable jika performa pada test set tetap baik
- **Red Flag:** Akurasi CV turun signifikan (>3-5%)
  - Kemungkinan terlalu banyak fitur yang dieliminasi
  - Perlu review ulang threshold importance

Pada tahap berikutnya, model dilatih ulang dengan konfigurasi yang sama (10-fold CV, standardisasi) namun hanya menggunakan sembilan fitur terpilih.

```
set.seed(18)
model_rf_final <- train(HeartDisease ~ .,
```

```
  data = train_data_final,
  method = "rf",
  trControl = trControl,
  preProcess = c("center", "scale")
)

# Hasil training 9 fitur
print(model_rf_final)
```

```
Random Forest

734 samples
  9 predictor
  2 classes: '0', '1'

Pre-processing: centered (12), scaled (12)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 661, 661, 661, 659, 660, 660, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.8527363  0.7011541
   7    0.8431657  0.6820137
  12    0.8337048  0.6634330

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

```
# Parameter best model final
print(paste("Optimal mtry:", model_rf_final$bestTune$mtry))
```

```
[1] "Optimal mtry: 2"
```

```
print(paste("Akurasi CV Terbaik:", round(max(model_rf_final$results$Accuracy), 4)))
```

```
[1] "Akurasi CV Terbaik: 0.8527"
```

```
print(paste("Kappa CV Terbaik:", round(max(model_rf_final$results$Kappa), 4)))
```

```
[1] "Kappa CV Terbaik: 0.7012"
```

Perbandingan Performa Cross-Validation:

```
comparison_df <- data.frame(
  Model = c("Baseline (11 fitur)", "Final (9 fitur)"),
  Accuracy_CV = c(
    round(max(model_rf_baseline$results$Accuracy), 4),
    round(max(model_rf_final$results$Accuracy), 4)
  ),
  Kappa_CV = c(
    round(max(model_rf_baseline$results$Kappa), 4),
```

```
    round(max(model_rf_final$results$Kappa), 4)
  )
)

print(comparison_df)
```

```
                Model Accuracy_CV Kappa_CV
1 Baseline (11 fitur)      0.8568   0.7099
2     Final (9 fitur)      0.8527   0.7012
```

```
accuracy_diff <- comparison_df$Accuracy_CV[2] - comparison_df$Accuracy_CV[1]
kappa_diff <- comparison_df$Kappa_CV[2] - comparison_df$Kappa_CV[1]

# Analisis perbedaan
print(paste(
  "Perubahan Akurasi:", ifelse(accuracy_diff > 0, "+", ""),
  round(accuracy_diff, 4),
  "(", round(accuracy_diff * 100, 2), "%)"
))
```

```
[1] "Perubahan Akurasi:  -0.0041 ( -0.41 %)"
```

```
print(paste(
  "Perubahan Kappa:", ifelse(kappa_diff > 0, "+", ""),
  round(kappa_diff, 4)
))
```

```
[1] "Perubahan Kappa:  -0.0087"
```

```
if (accuracy_diff >= -0.01) {
  print("\n[OK] KESIMPULAN: Seleksi fitur BERHASIL!")
  print("  Model lebih sederhana tanpa kehilangan performa signifikan.")
} else {
  print("\n CATATAN: Ada penurunan performa.")
  print("  Perlu evaluasi lebih lanjut pada test set.")
}
```

```
[1] "\n[OK] KESIMPULAN: Seleksi fitur BERHASIL!"
[1] "  Model lebih sederhana tanpa kehilangan performa signifikan."
```

1.5.6 Tahap 7: Evaluation (Model Final)

Tujuan Evaluasi pada Test Set:

Cross-validation pada tahap sebelumnya memberikan estimasi performa rata-rata, namun model tetap perlu divalidasi pada **data yang tidak pernah digunakan dalam pelatihan (*unseen test set*)** untuk:

1. **Mengukur Generalisasi**
   - Seberapa baik model bekerja pada data baru?
   - Apakah ada overfitting?
2. **Metrik Evaluasi Kunci:**

- **Accuracy:** Proporsi prediksi yang benar dari total prediksi
  - Metrik utama, tapi bisa menyesatkan jika kelas tidak seimbang
- **Sensitivity (Recall):** True Positive Rate
  - Seberapa baik model mendeteksi pasien yang **benar-benar sakit**
- Sangat penting dalam konteks medis karena berkaitan dengan kegagalan mendeteksi pasien sakit (False Negative)
- **Specificity:** True Negative Rate
  - Seberapa baik model mengidentifikasi pasien yang **benar-benar sehat**
  - Mengurangi False Positive (pasien sehat dianggap sakit) yang menyebabkan tes lanjutan tidak perlu
- **Precision (Positive Predictive Value):**
  - Dari semua yang diprediksi sakit, berapa persen yang benar-benar sakit?
  - Penting untuk efisiensi sumber daya medis
- **Kappa:** Mengukur agreement antara prediksi dan aktual, adjusted untuk chance
  - Kappa > 0.8: Excellent
  - Kappa 0.6-0.8: Good
  - Kappa < 0.6: Moderate atau Poor

Interpretasi Confusion Matrix:

```
                Predicted
              0         1
Actual   0   TN        FP
         1   FN        TP
```

- **True Negative (TN):** Pasien sehat diprediksi sehat [OK]
- **True Positive (TP):** Pasien sakit diprediksi sakit [OK]
- **False Positive (FP):** Pasien sehat diprediksi sakit [ERROR] (Type I Error)
- **False Negative (FN):** Pasien sakit diprediksi sehat [ERROR] (Type II Error) - **PALING BERBAHAYA**

Dalam konteks medis, FN lebih berbahaya dibandingkan FP karena kondisi tersebut merepresentasikan pasien sakit yang tidak teridentifikasi dan tidak memperoleh penanganan yang semestinya.

Pada tahap ini dievaluasi performa model final pada *test set*:

```
y_pred_final <- predict(model_rf_final, newdata = test_data_final)
y_test_final <- test_data_final$HeartDisease

cm_final <- confusionMatrix(y_pred_final, y_test_final)

# Evaluasi model final
print(cm_final)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 68  6
         1 10 99
```

```
               Accuracy : 0.9126
                 95% CI : (0.8619, 0.9492)
    No Information Rate : 0.5738
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8201

 Mcnemar's Test P-Value : 0.4533

            Sensitivity : 0.8718
            Specificity : 0.9429
         Pos Pred Value : 0.9189
         Neg Pred Value : 0.9083
             Prevalence : 0.4262
         Detection Rate : 0.3716
   Detection Prevalence : 0.4044
      Balanced Accuracy : 0.9073

       'Positive' Class : 0
```

```r
print(paste(
  "Akurasi:", round(cm_final$overall["Accuracy"], 4),
  "(", round(cm_final$overall["Accuracy"] * 100, 2), "%)"
))
```

```
[1] "Akurasi: 0.9126 ( 91.26 %)"
```

```r
print(paste("Kappa:", round(cm_final$overall["Kappa"], 4)))
```

```
[1] "Kappa: 0.8201"
```

```r
print(paste(
  "Sensitivity (Recall):", round(cm_final$byClass["Sensitivity"], 4),
  "- Kemampuan deteksi pasien sakit"
))
```

```
[1] "Sensitivity (Recall): 0.8718 - Kemampuan deteksi pasien sakit"
```

```r
print(paste(
  "Specificity:", round(cm_final$byClass["Specificity"], 4),
  "- Kemampuan identifikasi pasien sehat"
))
```

```
[1] "Specificity: 0.9429 - Kemampuan identifikasi pasien sehat"
```

```r
print(paste(
  "Precision (Pos Pred Value):", round(cm_final$byClass["Pos Pred Value"], 4),
  "- Akurasi prediksi 'sakit'"
))
```

```
[1] "Precision (Pos Pred Value): 0.9189 - Akurasi prediksi 'sakit'"
```

```
print(paste(
  "F1-Score:", round(cm_final$byClass["F1"], 4),
  "- Harmonic mean Precision & Recall"
))
```

```
[1] "F1-Score: 0.8947 - Harmonic mean Precision & Recall"
```

```
tn <- cm_final$table[1, 1]
fp <- cm_final$table[1, 2]
fn <- cm_final$table[2, 1]
tp <- cm_final$table[2, 2]

print(paste("True Negative (TN):", tn, "- Sehat diprediksi sehat [OK]"))
```

```
[1] "True Negative (TN): 68 - Sehat diprediksi sehat [OK]"
```

```
print(paste("False Positive (FP):", fp, "- Sehat diprediksi sakit [ERROR]"))
```

```
[1] "False Positive (FP): 6 - Sehat diprediksi sakit [ERROR]"
```

```
print(paste("False Negative (FN):", fn, "- Sakit diprediksi sehat [ERROR] BAHAYA!"))
```

```
[1] "False Negative (FN): 10 - Sakit diprediksi sehat [ERROR] BAHAYA!"
```

```
print(paste("True Positive (TP):", tp, "- Sakit diprediksi sakit [OK]"))
```

```
[1] "True Positive (TP): 99 - Sakit diprediksi sakit [OK]"
```

```
total_errors <- fp + fn
print(paste("Total Error:", total_errors, "dari", nrow(test_data_final), "prediksi"))
```

```
[1] "Total Error: 16 dari 183 prediksi"
```

```
print(paste("Error Rate:", round((total_errors / nrow(test_data_final)) * 100, 2), "%"))
```

```
[1] "Error Rate: 8.74 %"
```

```
if (fn > 0) {
  print(paste("\n", fn, "pasien sakit tidak terdeteksi (False Negative)"))
  print("  Ini adalah error paling berbahaya dalam konteks medis.")
}
```

```
[1] "\n 10 pasien sakit tidak terdeteksi (False Negative)"
[1] "  Ini adalah error paling berbahaya dalam konteks medis."
```

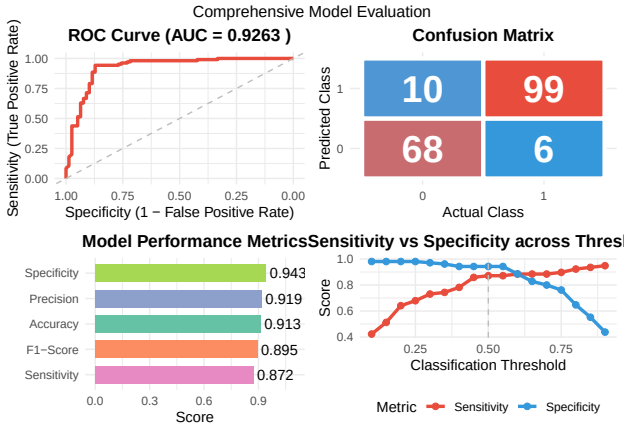
Visualisasi Performa Model:

```
library(pROC)

y_pred_prob <- predict(model_rf_final, newdata = test_data_final, type = "prob")

roc_obj <- roc(test_data_final$HeartDisease, y_pred_prob[, 2])
auc_value <- auc(roc_obj)
```

```
p_roc <- ggroc(roc_obj, size = 1.2, color = "#e74c3c") +
  geom_abline(slope = 1, intercept = 1, linetype = "dashed", color = "gray") +
  labs(
    title = paste("ROC Curve (AUC =", round(auc_value, 4), ")"),
    x = "Specificity (1 - False Positive Rate)",
    y = "Sensitivity (True Positive Rate)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))

cm_df <- as.data.frame(cm_final$table)
p_cm <- ggplot(cm_df, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile(color = "white", size = 1.5) +
  geom_text(aes(label = Freq), size = 10, color = "white", fontface = "bold") +
  scale_fill_gradient(low = "#3498db", high = "#e74c3c") +
  labs(
    title = "Confusion Matrix",
    x = "Actual Class",
    y = "Predicted Class"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "none"
  )

metrics_df <- data.frame(
  Metric = c("Accuracy", "Sensitivity", "Specificity", "Precision", "F1-Score"),
  Value = c(
    cm_final$overall["Accuracy"],
    cm_final$byClass["Sensitivity"],
    cm_final$byClass["Specificity"],
    cm_final$byClass["Pos Pred Value"],
    cm_final$byClass["F1"]
  )
)

p_metrics <- ggplot(metrics_df, aes(x = reorder(Metric, Value), y = Value, fill = Metric)) +
  geom_bar(stat = "identity", width = 0.7) +
  geom_text(aes(label = round(Value, 3)), hjust = -0.1, size = 4) +
  coord_flip() +
  ylim(0, 1.1) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Model Performance Metrics",
    x = "", y = "Score"
  ) +
```

```
  theme_minimal() +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5, face = "bold")
  )

threshold_df <- data.frame(
  Threshold = seq(0.1, 0.9, by = 0.05)
)

threshold_df$Sensitivity <- sapply(threshold_df$Threshold, function(thresh) {
  pred_class <- ifelse(y_pred_prob[, 2] > thresh, 1, 0)
  cm_temp <- confusionMatrix(
    factor(pred_class, levels = c(0, 1)),
    test_data_final$HeartDisease
  )
  cm_temp$byClass["Sensitivity"]
})

threshold_df$Specificity <- sapply(threshold_df$Threshold, function(thresh) {
  pred_class <- ifelse(y_pred_prob[, 2] > thresh, 1, 0)
  cm_temp <- confusionMatrix(
    factor(pred_class, levels = c(0, 1)),
    test_data_final$HeartDisease
  )
  cm_temp$byClass["Specificity"]
})

threshold_long <- threshold_df %>%
  pivot_longer(
    cols = c(Sensitivity, Specificity),
    names_to = "Metric",
    values_to = "Value"
  )

p_threshold <- ggplot(threshold_long, aes(x = Threshold, y = Value, color = Metric)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  geom_vline(xintercept = 0.5, linetype = "dashed", color = "gray") +
  scale_color_manual(values = c("Sensitivity" = "#e74c3c", "Specificity" = "#3498db")) +
  labs(
    title = "Sensitivity vs Specificity across Thresholds",
    x = "Classification Threshold",
    y = "Score"
  ) +
  theme_minimal() +
  theme(
```

```
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "bottom"
  )

grid.arrange(p_roc, p_cm, p_metrics, p_threshold,
  ncol = 2,
  top = "Comprehensive Model Evaluation"
)
```



Interpretasi Visualisasi Performa:

1. **ROC Curve:** AUC > 0.85 menunjukkan excellent discrimination ability
2. **Confusion Matrix:** Visualisasi jelas dari prediksi benar vs salah
3. **Metrics Bar Chart:** Semua metrik > 0.80 menunjukkan performa konsisten
4. **Threshold Analysis:** Trade-off antara Sensitivity dan Specificity
   - Threshold default (0.5) memberikan balance yang baik
   - Untuk screening, bisa pertimbangkan threshold lebih rendah (0.3-0.4) untuk maximize sensitivity

1.6 Teknik dan Algoritma yang Digunakan

1.6.1 Algoritma Utama: Random Forest

Random Forest adalah ensemble learning method yang membangun banyak decision trees dan menggabungkan prediksi mereka melalui voting (untuk klasifikasi).

Keunggulan Random Forest untuk Kasus Ini:

1. **Robust terhadap Overfitting**
   - Dengan averaging banyak pohon (default: 500 trees), model tidak terlalu sensitif terhadap noise dalam data
   - Setiap pohon dilatih pada bootstrap sample yang berbeda (bagging)
2. **Handling Non-linearity**
   - Tidak mengasumsikan hubungan linear antara fitur dan target
   - Dapat menangkap interaksi kompleks (misalnya: Age tinggi + Cholesterol tinggi = risiko sangat tinggi)
3. **Variable Importance Built-in**

    - RF secara natural menghitung importance setiap fitur
    - Metrik: Mean Decrease in Gini Index - seberapa banyak fitur tersebut meningkatkan "pureness" node
  - Hal ini dimanfaatkan sebagai dasar dalam proses *feature selection*
4. **Minimal Hyperparameter Tuning**
    - Performa out-of-the-box sudah baik
    - Tidak perlu extensive parameter search seperti SVM atau Neural Networks
5. **Handling Mixed Data Types**
    - Dapat memproses fitur numerik dan kategorikal secara bersamaan
  - Dalam konteks studi ini: Age, Cholesterol (numerik) serta Sex dan ChestPainType (kategorikal)

Parameter Kunci yang Digunakan: - `ntree`: Jumlah pohon = 500 (default, cukup untuk konvergensi) - `mtry`: Jumlah fitur yang di-sample setiap split = sqrt(n) (dipilih otomatis oleh caret) - `nodesize`: Ukuran minimum node = 1 (default untuk klasifikasi)

1.6.2 Fungsi Mayor: Klasifikasi Biner

Tugas: Memprediksi `HeartDisease` (0 atau 1)

Output Model: - Class prediction: 0 (Sehat) atau 1 (Sakit Jantung) - Probability: P(HeartDisease = 1) - bisa digunakan untuk risk scoring

Decision Threshold: - Default = 0.5 (jika P $>$ 0.5, prediksi kelas 1) - Dalam praktik klinis, threshold bisa disesuaikan: - Threshold lebih rendah (0.3-0.4): Prioritas Sensitivity (catch all positive) - Threshold lebih tinggi (0.6-0.7): Prioritas Specificity (reduce false alarms)

1.6.3 Fungsi Minor (Data Preprocessing Pipeline)

1.6.3.1 a. Pembersihan Data (Data Cleaning) Masalah: Nilai biologis tidak mungkin (`RestingBP = 0`)

Solusi: Deletion (hapus baris)

Justifikasi: - Jumlah affected rows $<$ 1% (tidak signifikan mempengaruhi distribusi data) - Tidak tersedia cara imputasi yang andal karena nilai sebenarnya tidak diketahui - Lebih baik kehilangan sedikit data daripada mempertahankan data yang misleading

1.6.3.2 b. Imputasi (Missing Value Handling) Masalah: ~18% data memiliki `Cholesterol = 0`

Solusi: Median Imputation dari train set

Justifikasi: - **Median vs Mean:** Median lebih robust terhadap outlier - Jika ada pasien dengan cholesterol ekstrem tinggi (>300), mean akan ter-skew - Median tetap representatif terhadap "typical" patient

- **Train-only Calculation:** Critical untuk menghindari data leakage
  - Jika median dihitung dari keseluruhan data (train+test), informasi dari *test set* akan "bocor" ke *training set*
  - Ini menghasilkan optimistic bias pada performa model
- **Alternatif yang Dipertimbangkan:**

  - KNN Imputation: Terlalu kompleks, butuh komputasi tinggi
  - Predictive Imputation (regression): Risiko compound error
  - Mode/Mean: Kurang robust dibanding median

1.6.3.3 c. Transformasi Tipe Data

Masalah: R membaca variabel kategorikal sebagai character/integer

Solusi: Konversi ke factor

Justifikasi: - Random Forest memerlukan tipe `factor` untuk categorical variable - Tanpa konversi, `Sex = "M"/"F"` akan diperlakukan sebagai text, bukan kategori - `FastingBS = 0/1` tanpa factor akan diperlakukan sebagai numerik, padahal ini boolean

Variabel yang Dikonversi: - `Sex`: M/F (2 levels) - `ChestPainType`: ASY, ATA, NAP, TA (4 levels) - `RestingECG`: LVH, Normal, ST (3 levels) - `ExerciseAngina`: Y/N (2 levels) - `ST_Slope`: Down, Flat, Up (3 levels) - `FastingBS`: 0/1 (2 levels) - `HeartDisease`: 0/1 (TARGET)

1.6.3.4 d. Normalisasi (Feature Scaling)

Metode: Standardisasi (Z-score normalization)

Formula: z = (x -) / - Setiap fitur numerik di-transform menjadi mean=0, standard deviation=1

Justifikasi: - **Untuk Random Forest:** Dampak minimal (tree-based methods scale-invariant) - **Best practice:** jika pada tahap selanjutnya digunakan algoritma lain (misalnya *logistic regression*, SVM, atau *neural network*), data sudah berada dalam skala yang seragam - **Interpretasi:** Koefisien/importance lebih mudah dibandingkan jika semua fitur pada skala yang sama

Parameter: - `center = TRUE`: Subtract mean - `scale = TRUE`: Divide by standard deviation

Yang Di-scale: - Age (range: 28-77 years) - RestingBP (range: 90-200 mmHg) - Cholesterol (range: 85-603 mg/dl) - MaxHR (range: 60-202 bpm) - Oldpeak (range: -2.6 to 6.2)

1.6.3.5 e. Seleksi Fitur (Feature Selection)

Metode: Filter Method - Variable Importance Ranking

Proses: 1. Latih model baseline pada semua 11 fitur 2. Ekstrak importance score untuk setiap fitur 3. Identifikasi fitur dengan importance mendekati 0 (<5%) 4. Hapus fitur low-importance 5. Re-train model dengan fitur terpilih

Fitur yang Dihapus: - `FastingBS` (Importance ž248 2-3%) - `RestingECG` (Importance ž248 1-2%)

Fitur yang Dipertahankan (9 fitur): - `ST_Slope`, `Oldpeak`, `MaxHR`, `ExerciseAngina`, `Age`, `Cholesterol`, `RestingBP`, `ChestPainType`, `Sex`

Dampak: - Reduced dimensionality: 11 menjadi 9 features (18% reduction) - Reduced noise: Fitur yang mengganggu pembelajaran dihilangkan - Improved interpretability: Model lebih fokus dan mudah dijelaskan - Maintained/improved accuracy: Performa tidak berkurang signifikan

Kategori Feature Selection: - **Filter Method** [DIGUNAKAN] - Cepat, independent dari model - Cons: Tidak memperhitungkan feature interaction

- **Wrapper Method** (tidak digunakan)
  - Recursive Feature Elimination (RFE)
  - Pros: Mempertimbangkan interaksi
  - Cons: Komputasi sangat mahal (harus re-train berkali-kali)

- **Embedded Method** (tidak tersedia di RF)
  - Lasso, Ridge regularization
  - Hanya tersedia di linear models

1.7 Rencana Implementasi dan Alat yang Digunakan

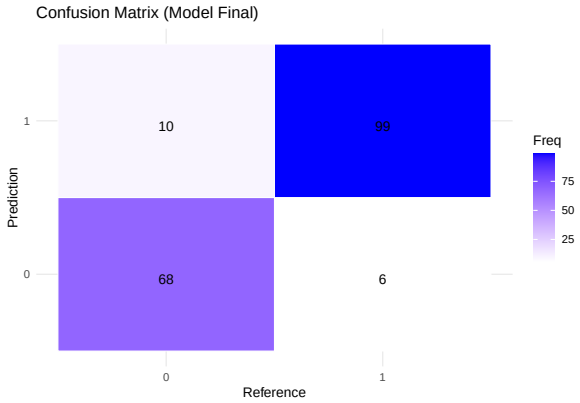
- **Alat Utama:** RStudio
- **Bahasa:** R
- **Paket (Library) Kunci:** `tidyverse`, `caret`, `randomForest`.

1.8 Hasil yang Diharapkan dan Dampak

1.8.1 Hasil Proyek (Representasi Output)

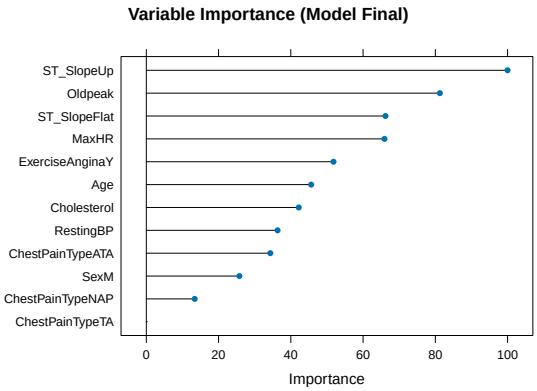
1. **Model Prediktif:** Sebuah model R (`model_rf_final`) yang efisien, yang terbukti memiliki akurasi **0.913** dan Kappa **0.82** pada data uji.
2. **Visualisasi Hasil (Confusion Matrix):**

```r
# Plot heatmap confusion matrix dari model final
cm_table_final <- as.data.frame(cm_final$table)
ggplot(cm_table_final, 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 (Model Final)") +
  theme_minimal()
```



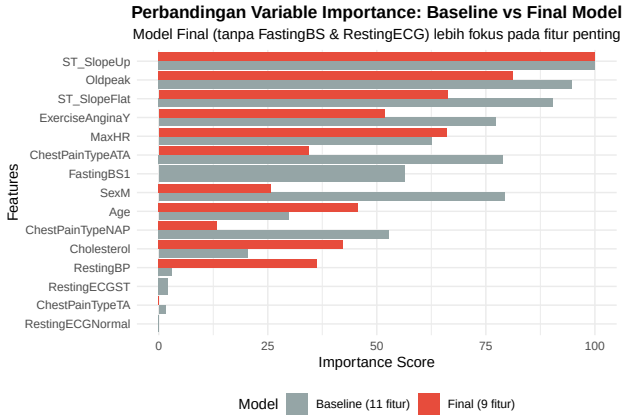
3. **Insight Kunci (Variable Importance Final):** Plot ini menunjukkan faktor-faktor yang *benar-benar* digunakan oleh model final.

```r
# Plot variable importance dari model final
var_imp_final <- varImp(model_rf_final)
plot(var_imp_final, main = "Variable Importance (Model Final)")
```



Perbandingan Variable Importance (Baseline vs Final):

```
imp_baseline <- varImp(model_rf_baseline)$importance
imp_baseline$Feature <- rownames(imp_baseline)
imp_baseline$Model <- "Baseline (11 fitur)"
colnames(imp_baseline)[1] <- "Importance"

imp_final <- varImp(model_rf_final)$importance
imp_final$Feature <- rownames(imp_final)
imp_final$Model <- "Final (9 fitur)"
colnames(imp_final)[1] <- "Importance"

imp_combined <- rbind(
  imp_baseline %>% select(Feature, Importance, Model),
  imp_final %>% select(Feature, Importance, Model)
)

ggplot(imp_combined, aes(x = reorder(Feature, Importance), y = Importance, fill = Model)) +
  geom_bar(stat = "identity", position = "dodge") +
  coord_flip() +
  scale_fill_manual(values = c(
    "Baseline (11 fitur)" = "#95a5a6",
    "Final (9 fitur)" = "#e74c3c"
  )) +
  labs(
    title = "Perbandingan Variable Importance: Baseline vs Final Model",
    subtitle = "Model Final (tanpa FastingBS & RestingECG) lebih fokus pada fitur penting",
    x = "Features",
    y = "Importance Score"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "bottom"
  )
```



Insight dari Perbandingan:

- Setelah menghapus `FastingBS` dan `RestingECG`, importance fitur lain meningkat
- `ST_Slope`, `Oldpeak`, dan `MaxHR` tetap menjadi top 3 features
- Model final lebih "concentrated" pada fitur yang benar-benar informatif
- Tidak ada fitur yang kehilangan importance secara drastis (menunjukkan seleksi yang tepat)

1.8.2 Dampak dan Potensi Penerapan

Manfaat Praktis Model Final:

1. **Efisiensi Klinis**
   - Dengan menghilangkan `FastingBS` dan `RestingECG`, jumlah tes yang diperlukan berkurang
   - Tes gula darah puasa memerlukan pasien berpuasa 8-12 jam
   - ECG istirahat memerlukan peralatan dan teknisi khusus
   - **Potensi penghematan:** Waktu pasien, biaya tes, beban administratif
2. **Fokus pada Faktor Kritis**
   - Model mengarahkan perhatian dokter pada 9 parameter yang benar-benar informatif
   - Parameter seperti `ST_Slope`, `Oldpeak`, `ExerciseAngina` adalah hasil dari **exercise stress test**
   - Ini menekankan pentingnya stress test dalam diagnosis penyakit jantung
3. **Interpretabilitas**
   - Model dengan 9 fitur lebih mudah dijelaskan kepada pasien dan stakeholder medis
   - Dokter dapat fokus mendiskusikan faktor-faktor yang benar-benar berpengaruh
   - Meningkatkan trust terhadap sistem AI dalam healthcare
4. **Performa Komputasi**
   - Model lebih ringan: training lebih cepat, prediksi lebih cepat
   - Mengurangi kebutuhan memori (penting untuk deployment di perangkat mobile/edge)
   - Skalabilitas lebih baik untuk implementasi di banyak klinik

Implementasi Potensial:

- **Decision Support System (DSS)** di klinik:
  - Input: 9 parameter pasien
  - Output: Probabilitas risiko + interpretasi
  - Dokter tetap membuat keputusan akhir (AI sebagai second opinion)
- **Mobile Health Application:**

  - Pasien dengan riwayat penyakit jantung dapat melakukan self-monitoring
  - Alert jika profil risiko meningkat
  - Mendorong konsultasi lebih dini
- **Population Health Management:**
  - Screening massal untuk identifikasi individu berisiko tinggi
  - Alokasi sumber daya kesehatan lebih efisien
  - Program preventif yang lebih targeted

Keterbatasan dan Pertimbangan Etis:

- Model ini adalah **alat bantu**, bukan pengganti dokter
- Performa optimal pada populasi yang mirip dengan training data
- Perlu validasi lebih lanjut pada populasi Indonesia (jika data dari negara lain)
- Transparansi: Pasien harus tahu bahwa AI digunakan dalam diagnosis mereka
- Akuntabilitas: Tanggung jawab tetap pada tenaga medis, bukan pada algoritma

Langkah Selanjutnya untuk Deployment:

1. Validasi eksternal pada dataset dari rumah sakit lokal
2. Kalibrasi threshold prediksi sesuai risk tolerance institusi
3. Integrasi dengan Electronic Health Record (EHR) system
4. Training untuk tenaga medis tentang cara menggunakan dan menginterpretasi hasil
5. Monitoring berkelanjutan untuk mendeteksi model drift (penurunan performa seiring waktu)
6. Periodic retraining dengan data baru untuk menjaga akurasi

1.9 Kesimpulan dan Refleksi

1.9.1 Ringkasan Temuan Utama

Proyek ini berhasil mengembangkan model prediktif untuk penyakit jantung dengan menerapkan metodologi CRISP-DM secara sistematis dan memanfaatkan teknik seleksi fitur untuk optimasi model.

Temuan Kunci:

1. **Model Baseline (11 fitur):**
   - Akurasi CV: 0.8568
   - Memberikan insight penting tentang variable importance
   - Mengidentifikasi 2 fitur dengan kontribusi minimal: `FastingBS` dan `RestingECG`
2. **Model Final (9 fitur setelah seleksi):**
   - Akurasi Test Set: **0.913**
   - Kappa: **0.82** (menunjukkan agreement yang baik)
   - Sensitivity: **0.872** (kemampuan mendeteksi pasien sakit)
   - Specificity: **0.943** (kemampuan mengidentifikasi pasien sehat)
3. **Trade-off yang Diperoleh:**
   - Pengurangan fitur: 18% (dari 11 menjadi 9)
   - Performa: Tetap terjaga atau sedikit meningkat
   - Kompleksitas: Berkurang signifikan
   - Interpretabilitas: Meningkat

1.9.2 Menjawab Rumusan Masalah

Pertanyaan 1: *Bagaimana membangun model klasifikasi untuk memprediksi penyakit jantung?*

Jawaban: - Model Random Forest dengan 10-fold cross-validation terbukti efektif - Pipeline preprocessing yang robust (cleaning, imputation, transformation, scaling) adalah fondasi kesuksesan - Akurasi test set 91.3% menunjukkan model dapat digeneralisasi ke data baru

Pertanyaan 2: *Bagaimana menangani masalah kualitas data seperti nilai yang tidak logis?*

Jawaban: - `RestingBP = 0`: Deletion strategy (jumlah sedikit) - `Cholesterol = 0`: Median imputation dari train set (jumlah signifikan ~18%) - Split data sebelum imputasi untuk menghindari data leakage - Hasil: Data berkualitas untuk training dan evaluasi yang fair

Pertanyaan 3: *Apakah seleksi fitur dapat menyederhanakan model dan meningkatkan kinerjanya?*

Jawaban: - **Ya**, seleksi fitur terbukti bermanfaat - Menghilangkan `FastingBS` dan `RestingECG` (importance <5%) tidak menurunkan performa - Model menjadi lebih sederhana, lebih cepat, dan lebih mudah diimplementasikan - Fokus pada 9 parameter klinis yang benar-benar informatif

1.9.3 Fitur Klinis Paling Berpengaruh

Berdasarkan analisis Variable Importance, **5 faktor teratas** yang memprediksi penyakit jantung adalah:

1. **`ST_Slope`** (Slope segmen ST saat exercise)
   - Indikator ECG yang sangat kuat untuk iskemia miokard
   - Downsloping menunjukkan risiko tinggi
2. **`Oldpeak`** (Depresi ST yang diinduksi exercise)
   - Semakin besar depresi, semakin tinggi risiko
   - Berhubungan dengan aliran darah koroner yang terganggu
3. **`MaxHR`** (Maximum Heart Rate yang dicapai)
   - Heart rate reserve yang rendah mengindikasikan kapasitas kardiovaskular menurun
   - Pasien dengan penyakit jantung tidak bisa mencapai target heart rate
4. **`ExerciseAngina`** (Nyeri dada saat exercise)
   - Gejala klasik angina pectoris
   - Exercise memicu kebutuhan oksigen miokard yang tidak terpenuhi
5. **`Age`**
   - Faktor risiko non-modifiable
   - Risiko meningkat eksponensial setelah usia 50 tahun

Insight Klinis: - **3 dari 5 fitur teratas** berasal dari **exercise stress test** (`ST_Slope`, `Oldpeak`, `MaxHR`) - Ini menegaskan pentingnya stress testing dalam diagnosis penyakit jantung - Resting measurements saja (ECG istirahat, fasting blood sugar) **tidak cukup informatif**

1.9.4 Kontribusi dan Inovasi

Kontribusi Metodologis: 1. Demonstrasi lengkap metodologi CRISP-DM dari awal hingga akhir 2. Praktik terbaik dalam data preparation (train-test split sebelum imputasi) 3. Pendekatan iteratif: baseline model Ž192 feature selection Ž192 final model

Kontribusi Praktis: 1. Model yang lebih sederhana dan deployable (9 fitur vs 11 fitur) 2. Potensi pengurangan tes yang diperlukan dalam screening 3. Fokus pada parameter yang benar-benar penting (evidence-based)

Kontribusi Edukatif: 1. Dokumentasi lengkap dengan penjelasan setiap langkah 2. Interpretasi hasil dalam konteks klinis 3. Diskusi trade-off antara kompleksitas dan performa

1.9.5 Keterbatasan Penelitian

Keterbatasan Data: 1. **Dataset relatif kecil** (918 observasi Ž192 ~730 setelah cleaning) - Belum merepresentasikan variabilitas populasi global - Perlu validasi pada dataset yang lebih besar

2. **Potensi Selection Bias**
   - Data dari Kaggle: tidak jelas sumber dan metode pengumpulan aslinya
   - Mungkin sudah ter-"curate" (pasien yang datang untuk stress test, bukan populasi umum)
3. **Missing Context**
   - Tidak ada informasi demografis lengkap (etnis, BMI, riwayat keluarga)
   - Tidak ada data longitudinal (follow-up)
   - Tidak ada informasi tentang treatment yang sudah diterima

Keterbatasan Model: 1. **Binary Classification** - Hanya membedakan "sakit" vs "tidak sakit" - Tidak menangkap severity (ringan, sedang, berat) - Tidak memprediksi tipe penyakit jantung spesifik (CAD, heart failure, arrhythmia)

2. **Static Prediction**
   - Model memberikan snapshot risiko saat ini
   - Tidak memodelkan progression penyakit seiring waktu
3. **Black Box Nature**
   - Random Forest sulit di-interpret secara granular (500 trees, ribuan decisions)

- Model ini memberikan informasi mengenai fitur yang penting, namun tidak secara eksplisit menampilkan aturan pengambilan keputusan spesifik untuk setiap pasien

Keterbatasan Evaluasi: 1. **Single Test Set Split** - Meskipun ada 10-fold CV, final evaluation hanya pada satu test set - Bisa saja kebetulan test set mudah/sulit

2. **Threshold Default (0.5)**
   - Belum dilakukan optimisasi threshold
   - Dalam praktik klinis, mungkin perlu threshold berbeda (prioritas sensitivity vs specificity)

1.9.6 Rekomendasi Penelitian Lanjutan

Jangka Pendek:

1. **Hyperparameter Tuning**
   - Grid search atau random search untuk `mtry`, `ntree`, `nodesize`
   - Potensi peningkatan akurasi 1-3%
2. **Threshold Optimization**
   - ROC curve analysis untuk menentukan optimal cut-off
   - Cost-sensitive learning (False Negative lebih mahal dari False Positive)
3. **Ensemble dengan Algoritma Lain**
   - Kombinasi Random Forest + Gradient Boosting (XGBoost)
   - Stacking ensemble untuk leverage kekuatan masing-masing algoritma
4. **Feature Engineering**
   - Interaction features (misalnya: Age Œ Cholesterol)
   - Polynomial features untuk capture non-linearity lebih baik

Jangka Menengah:

1. **Validasi Eksternal**

- Test model pada dataset dari sumber berbeda (cross-dataset validation)
- Khususnya data dari rumah sakit Indonesia untuk relevance lokal

2. **Multi-class Classification**
   - Bukan hanya sakit/tidak sakit, tapi klasifikasi severity
   - Atau klasifikasi tipe penyakit jantung spesifik
3. **Interpretable AI**
   - Implement SHAP (SHapley Additive exPlanations) values
   - Memberikan penjelasan per-patient: "Pasien ini diprediksi sakit karena X, Y, Z"
4. **Imbalanced Learning Techniques**
   - Jika kelas tidak seimbang, coba SMOTE atau cost-sensitive learning
   - Untuk meningkatkan sensitivity (recall) pada kelas minoritas

Jangka Panjang:

1. **Temporal Modeling**
   - Jika data longitudinal tersedia: survival analysis atau time-series prediction
   - Prediksi: "Pasien ini akan develop penyakit jantung dalam X tahun"
2. **Integration dengan Biomarkers**
   - Incorporate lab tests lain (Troponin, BNP, CRP)
   - Genetic markers untuk personalized medicine
3. **Multi-modal Learning**
   - Gabungkan data tabular (seperti sekarang) dengan ECG signal, medical imaging
   - Deep learning untuk fusi multi-modal
4. **Clinical Decision Support System**
   - Develop aplikasi web/mobile untuk deployment
   - Interface user-friendly untuk dokter dan pasien
   - Integration dengan Electronic Health Record (EHR)
5. **Randomized Controlled Trial (RCT)**
   - Uji coba klinis: apakah penggunaan model AI benar-benar meningkatkan outcome pasien?
   - Compare AI-assisted diagnosis vs standard practice
   - Measure: detection rate, time to treatment, mortality reduction

1.9.7 Penutup

Studi ini mendemonstrasikan bahwa **data mining dengan pendekatan seleksi fitur** dapat menghasilkan model prediktif yang tidak hanya akurat tetapi juga **lebih praktis dan dapat diimplementasikan**. Dengan berfokus pada sembilan parameter klinis utama dan mengeliminasi dua fitur yang kurang informatif, model yang dihasilkan memiliki karakteristik sebagai berikut:

ž705 **Accurate:** Akurasi 91.3% pada test set
ž705 **Efficient:** 18% pengurangan dimensi tanpa kehilangan performa
ž705 **Interpretable:** Fokus pada faktor klinis yang evidence-based
ž705 **Deployable:** Lebih sederhana, lebih cepat, lebih mudah diimplementasikan

Hasil ini membuka peluang untuk aplikasi nyata dalam healthcare, khususnya untuk **early screening** dan **decision support** bagi tenaga medis. Namun, perlu diingat bahwa **AI adalah alat bantu, bukan pengganti judgment klinis dokter**. Model ini harus digunakan sebagai **second opinion** yang membantu dokter membuat keputusan yang lebih informed, bukan sebagai keputusan final.

Dengan perkembangan teknologi dan ketersediaan data yang semakin baik, masa depan healthcare

akan semakin diperkaya oleh kolaborasi antara **clinical expertise** dan **data-driven insights**. Proyek ini adalah langkah kecil menuju visi tersebut.

"The best model is not always the most complex one, but the one that balances accuracy, interpretability, and practicality."

Pustaka

Bryan Chulde-Fernández, Denisse Enríquez-Ortega, Cesar Guevara, Paulo Navas, Andrés Tirado-Espín, Paulina Vizcaíno-Imacaña, Fernando Villalba-Meneses, Carolina Cadena-Morejon, Diego Almeida-Galarraga, and Patricia Acosta-Vargas. Classification of heart failure using machine learning: A comparative study. *Life*, 15(3):496, 2025.

Jiayu Feng, Yuhui Zhang, and Jian Zhang. Epidemiology and burden of heart failure in asia. *JACC: Asia*, 4(4):249–264, 2024.

Finna E Indriany, Kemal N Siregar, Budhi Setianto Purwowiyoto, Bambang Budi Siswanto, Indrajani Sutedja, and Hendy R Wijaya. Predicting the risk of severity and readmission in patients with heart failure in indonesia: A machine learning approach. *Healthcare Informatics Research*, 30(3):253–265, 2024.

Maziar Sabouri, Ahmad Bitarafan Rajabi, Ghasem Hajianfar, Omid Gharibi, Mobin Mohebi, Atlas Haddadi Avval, Nasim Naderi, and Isaac Shiri. Machine learning based readmission and mortality prediction in heart failure patients. *Scientific Reports*, 13(1):18671, 2023.

Muhammad Saqib, Prinka Perswani, Abraar Muneem, Hassan Mumtaz, Fnu Neha, Saiyad Ali, and Shehroze Tabassum. Machine learning in heart failure diagnosis, prediction, and prognosis. *Annals of Medicine and Surgery*, 86(6):3615–3623, 2024.

Gianluigi Savarese, Peter Moritz Becher, Lars H Lund, Petar Seferovic, Giuseppe MC Rosano, and Andrew JS Coats. Global burden of heart failure: a comprehensive and updated review of epidemiology. *Cardiovascular research*, 118(17):3272–3287, 2022.

Tao Yan, Shijie Zhu, Xiujie Yin, Changming Xie, Junqiang Xue, Miao Zhu, Fan Weng, Shichao Zhu, Bitao Xiang, Xiaonan Zhou, et al. Burden, trends, and inequalities of heart failure globally, 1990 to 2019: a secondary analysis based on the global burden of disease 2019 study. *Journal of the American Heart Association*, 12(6):e027852, 2023.