Laporan Praktikum 1: Rule-based Expert System

Mata Kuliah: Penalaran dan Representasi Pengetahuan

Nama : Agung Rashif Madani

NIM : 20106050083

METODE

Dalam praktikum ini, Saya menggunakan metode Decision Tree (Pohon Keputusan) sebagai dasar sistem pakar berbasis aturan. Decision Tree adalah model prediktif yang menggunakan struktur pohon untuk menggambarkan aturan-aturan keputusan berdasarkan fitur-fitur data input.

EKSPERIMEN

Dataset Saya bagi menjadi dua bagian: 80% untuk training dan 20% untuk testing. Berikut adalah kode program yang Saya gunakan:

```
1  library(rpart)
2  library(rpart.plot)
3  library(ggplot2)
4  library(caTools)
5  library(caret)
6
7  dataset <- read.csv("Advertisement.csv")
8
9  set.seed(123)
10 split <- sample.split(dataset$Purchased, SplitRatio = 0.8)
11 train_data <- subset(dataset, split == TRUE)
12 test_data <- subset(dataset, split == FALSE)
```

Gambar 1 – Screenshot Coding Library & Sampling

Penjelasan:

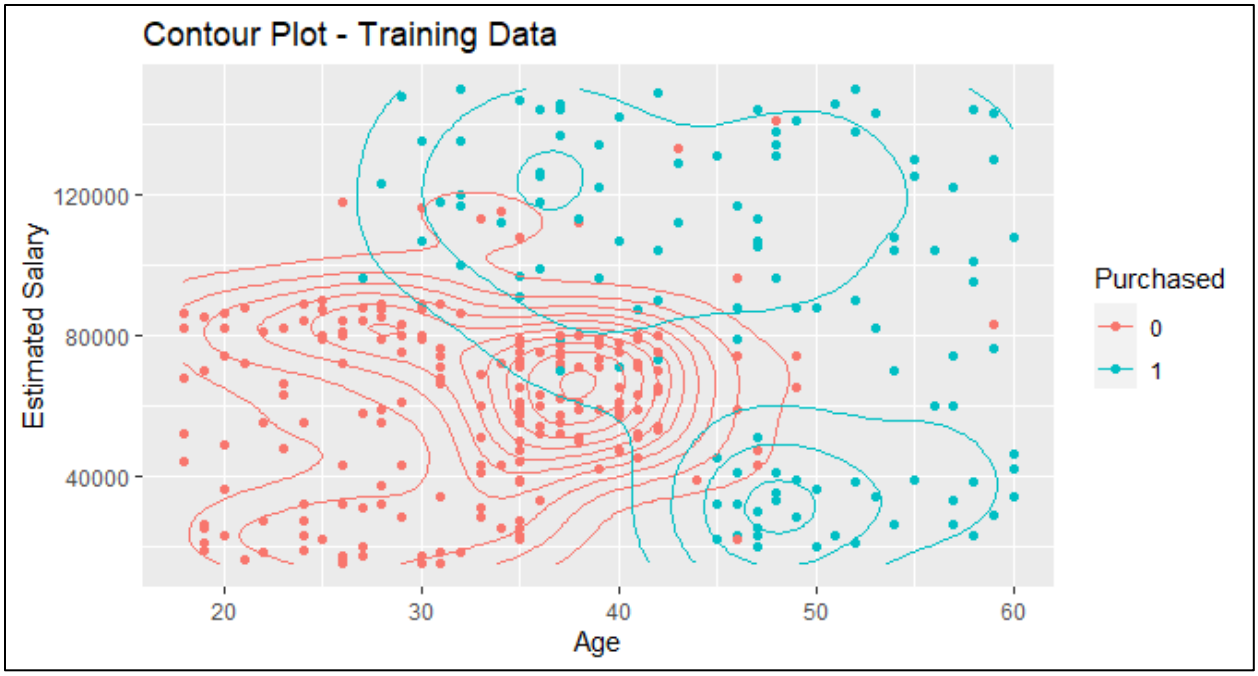
- **library()**: Memanggil libraries / packages yang dibutuhkan.
- **dataset <- read.csv("advertisement.csv")**: Membaca file CSV dengan nama "advertisement.csv" dan menyimpannya dalam variabel **dataset**. File ini berisi data yang akan digunakan dalam eksperimen.
- **set.seed(123)**: Random dataset, 123 digunakan agar hasil random dataset akan selalu sama setiap kali script run.
- **split <- sample.split(dataset$Purchased, SplitRatio = 0.8)**: Menggunakan fungsi **sample.split** dari paket **caTools** untuk membagi data menjadi *training data* dan *test data*. Pembagian dilakukan berdasarkan variabel "Purchased" dalam data frame **dataset**, dan argumen **SplitRatio** menentukan proporsi data training data (80%).
- **train_data <- subset(dataset, split == TRUE)**: menyimpan training data kedalam variabel **train_data**
- **test_data <- subset(dataset, split == FALSE)**: menyimpan selain training data ke dalam variabel **test_data**

HASIL dan ANALISIS

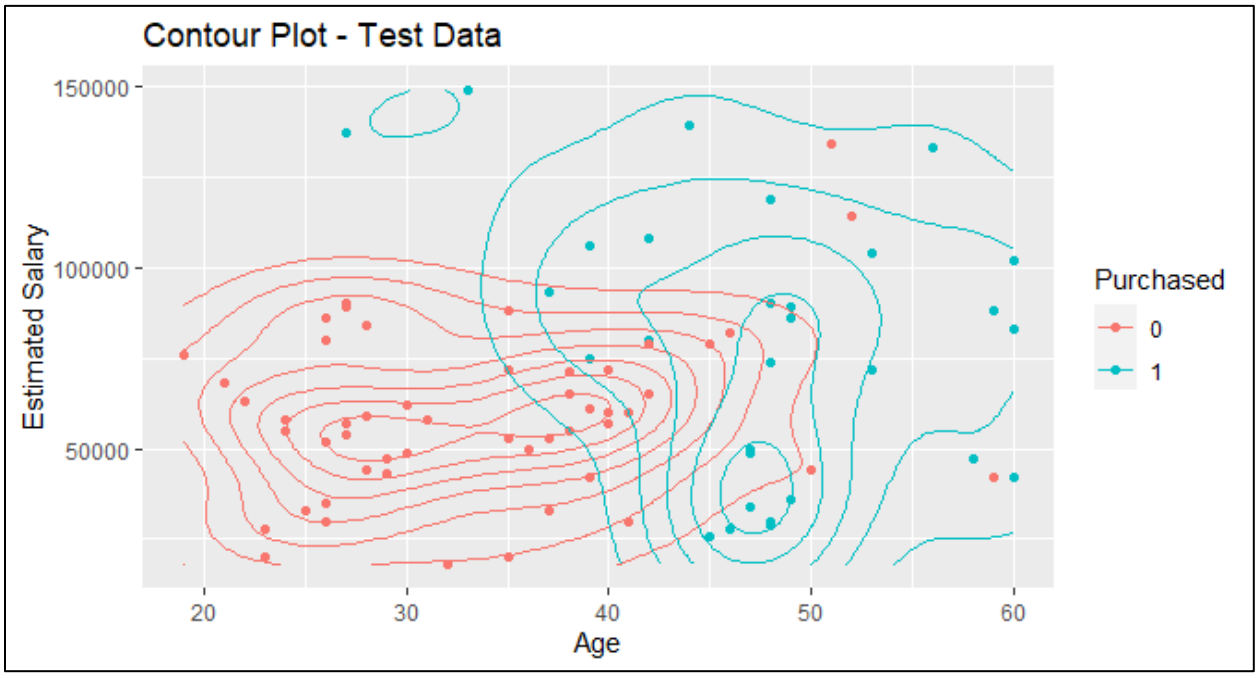
Berikut hasil contour plot dari *training data* dan *test data*:

```
14  ggplot(train_data, aes(x = Age, y = EstimatedSalary, color = factor(train_data$Purchased))) +
15    geom_point() +
16    geom_density_2d() +
17    labs(title = "Contour Plot - Training Data", x = "Age", y = "Estimated Salary", color = "Purchased")
18
19  ggplot(test_data, aes(x = Age, y = EstimatedSalary, color = factor(test_data$Purchased))) +
20    geom_point() +
21    geom_density_2d() +
22    labs(title = "Contour Plot - Test Data", x = "Age", y = "Estimated Salary", color = "Purchased")
```

Gambar 2 – Screenshot Coding Contour Plot Training & Test Data



Gambar 3 Contour Plot – Training Data



Gambar 4 Contour Plot – Test Data

Kontur plot digunakan untuk memvisualisasikan hubungan antara usia(Age) dan estimasi gaji(EstimatedSalary) dalam dataset latihan(train_data) dan dataset uji(test_data) terhadap variabel

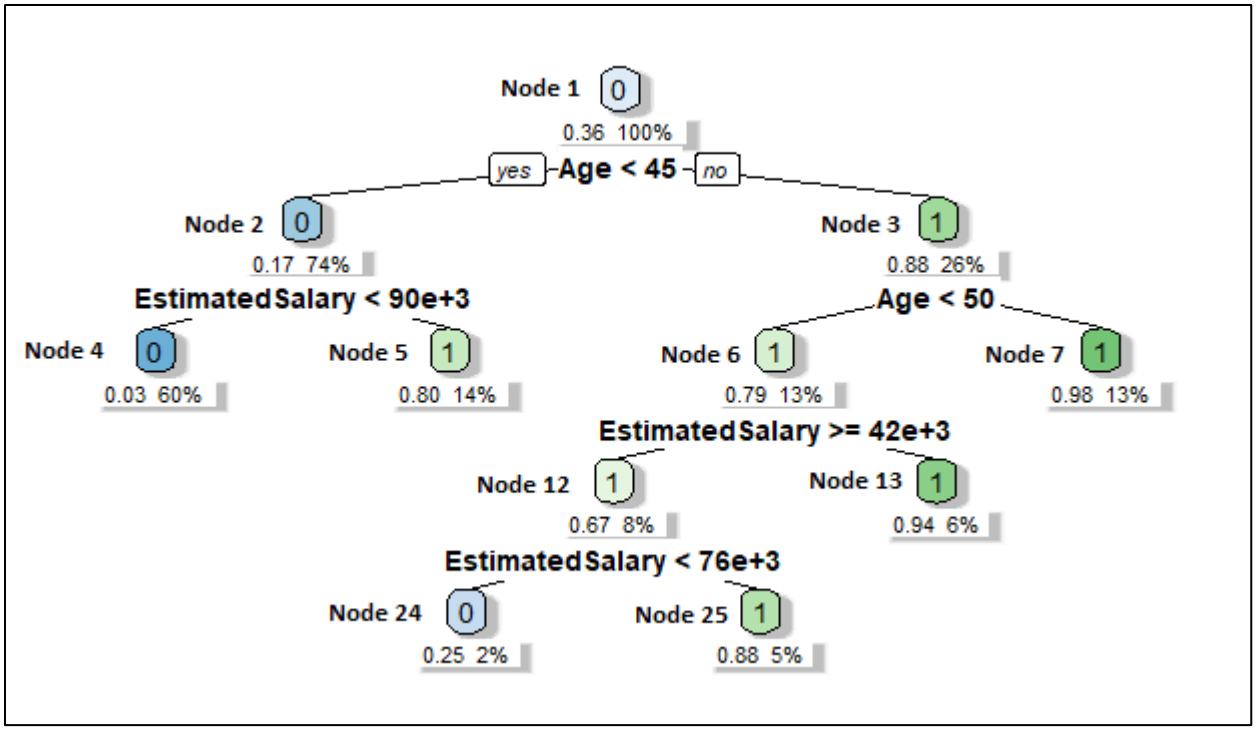
target(Purchased). Dalam kontur plot data latih, dapat dilihat pada Gambar 3, terlihat pola penyebaran data titik yang mewakili kombinasi usia dan estimasi gaji. Area dengan kepadatan tinggi menunjukkan banyaknya titik data dalam kombinasi tersebut. Plot tersebut dapat membantu untuk melihat apakah ada hubungan yang jelas antara usia, estimasi gaji, dan keputusan pembelian (Purchased). Dalam kontur plot data uji Gambar 4, dapat dilihat pola penyebaran cukup serupa tapi berbeda dengan data latih. Hal ini menunjukkan bahwa pembagian data menjadi data latihan dan data uji dilakukan secara acak dan representatif.

Selain itu, pada kontur plot terdapat dua kontur berwarna merah dan biru yang saling tumpeng tindih atau berpisah. Area dengan kontur merah menujukkan area di mana titik data memiliki nilai target (Purchased) yang sama dengan 0, dan sebaliknya warna biru untuk target (Purchased) yang sama dengan 1. Jika melihat pada kontor maka kemungkinan besar keputusan akan membeli atau 1 ketika seorang memiliki umur di atas 40 dan gaji di atas 100000.

Berikut hasil dari Decision Tree(pohon keputusan):

```
24  dt_model <- rpart(Purchased ~ Gender + Age + EstimatedSalary,
25                    data = train_data, method = "class")
26  rpart.plot(dt_model, fallen.leaves = FALSE, uniform = TRUE,
27             box.palette = "auto", shadow.col = "gray", under = TRUE,
28             varlen = 0, clip.right.labs = TRUE)
```

Gambar 5 Screenshot Coding Decision Tree Plot



Gambar 6 - Decision Tree Plot

Decision tree (pohon keputusan) adalah model yang digunakan untuk melakukan klasifikasi pada dataset. Setiap simpul (node) pada pohon keputusan merepresentasikan aturan (rules) atau keputusan berdasarkan nilai variabel input. Pohon dimulai dari akar (root) dan berakhir pada simpul-simpul daun (leaf nodes) yang mewakili hasil prediksi.

Berikut penjelasan Gambar 6:

Node 1: terdapat 320 data. Node ini memiliki probabilitas 0.36 untuk kelas 0 dan 0.64 untuk kelas 1. Terdapat dua node anak, yaitu Node 2 dengan 237 data(74% Node 1) dan Node 3 dengan 83 data(26% Node 1). Terpilih dibagi berdasarkan fitur Age < 43 yang memiliki nilai Improve terbesar.

- **Node 2**: terdapat 237 data. Node ini memiliki probabilitas 0.17 untuk kelas 0 dan 0.83 untuk kelas 1. Terdapat dua node anak, yaitu Node 4 dengan 193 data dan Node 5 dengan 44 data. Pembagian utama dilakukan berdasarkan fitur EstimatedSalary, Age, dan Gender. Terpilih fitur EstimatedSalary < 89500 yang memiliki nilai Improve terbesar.
    - **Node 4**: Terdapat 193 data. Node ini dengan probabilitas 0.03 untuk kelas 0 dan 0.97 untuk kelas 1.

    - **Node 5**: Terdapat 44 data. Node ini memiliki probabilitas 0.20 untuk kelas 0 dan 0.80 untuk kelas 1.
- **Node 3**: terdapat 83 data. Node ini memiliki probabilitas 0.12 untuk kelas 0 dan 0.88 untuk kelas 1. Terdapat dua node anak, yaitu Node 6 dengan 42 data dan Node 7 dengan 41 data. Terpilih fitur Age < 49.5 yang memiliki nilai Improve terbesar.
    - **Node 6**: Terdapat 42 data. Node ini memiliki probabilitas 0.21 untuk kelas 0 dan 0.79 untuk kelas 1. Terdapat dua node anak, yaitu Node 12 dengan 24 data dan Node 13 dengan 18 data. Terpilih fitur EstimatedSalary < 42000 yang memiliki nilai Improve terbesar.
        - **Node 12**: Terdapat 24 data. Node ini memiliki probabilitas 0.33 untuk kelas 0 dan 0.67 untuk kelas 1. Terdapat dua node anak, yaitu Node 24 dengan 8 data dan Node 25 dengan 16 data. Terpilih fitur EstimatedSalary < 76500 yang memiliki nilai Improve terbesar.
            - **Node 24**: Terdapat 8Node ini dengan probabilitas 0.25 untuk kelas 0 dan 0.75 untuk kelas 1.
            - **Node 25**: Terdapat 16 data dengan prediksi kelas 1 dan loss sebesar 0.13. Node ini dengan probabilitas 0.13 untuk kelas 0 dan 0.87 untuk kelas 1.
        - **Node 13**: Terdapat 18 data dengan prediksi kelas 1 dan loss sebesar 0.06. Node ini dengan probabilitas 0.06 untuk kelas 0 dan 0.94 untuk kelas 1.
    - **Node 7**: Terdapat 41 data. Prediksi kelas adalah 1 dengan loss sebesar 0.02. Node ini memiliki probabilitas 0.02 untuk kelas 0 dan 0.98 untuk kelas 1.

Dalam analisis ini, terdapat tiga variabel input, yaitu Age, EstimatedSalary, dan Gender untuk memprediksi apakah seseorang akan membeli produk atau tidak. Dari pohon keputusan, dapat disimpulkan bahwa variabel input Gender tidak memberikan kontribusi signifikan terhadap prediksi variabel target Purchased. Oleh karena itu, Gender tidak digunakan dalam pembagian simpul-simpul pada pohon keputusan ini. Pada tingkat Node tertentu, fitur-fitur tersebut membagi data berdasarkan nilai-nilai tertentu untuk menghasilkan prediksi yang lebih akurat.

AKURASI

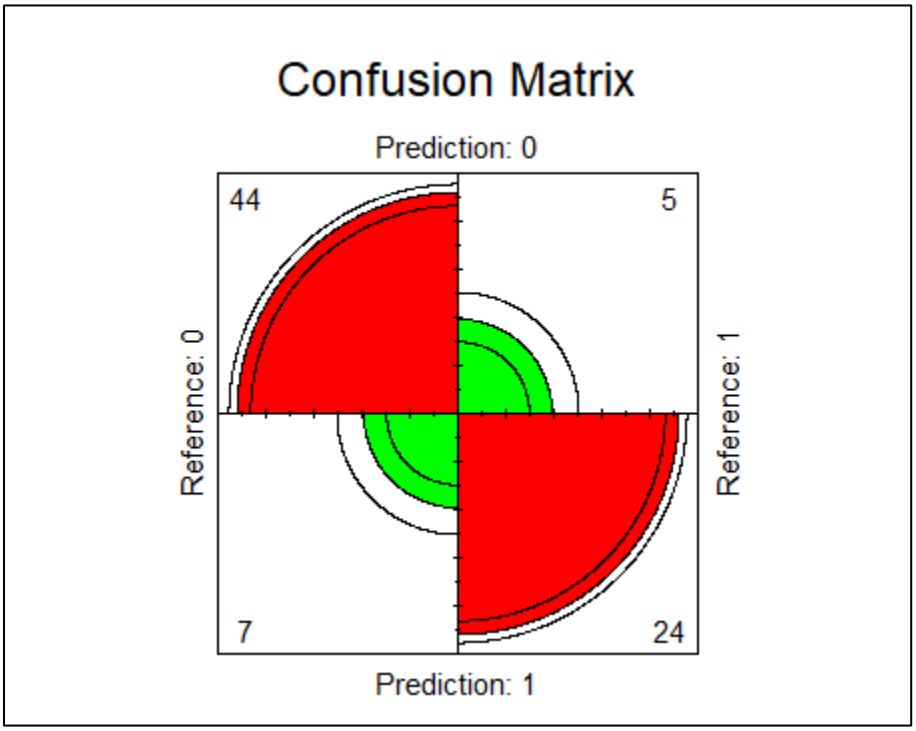
```
summary(dt_model)
predictions <- predict(dt_model, test_data, type = "class")

test_data$Purchased <- factor(test_data$Purchased, levels = levels(predictions))

confusion_matrix <- caret::confusionMatrix(data=predictions,
                                           reference=test_data$Purchased)
fourfoldplot(as.table(confusion_matrix),color=c("green","red"),main = "Confusion Matrix")

accuracy <- confusion_matrix$overall['Accuracy']
print(paste("Accuracy:", accuracy))
```

Gambar 7 – Screenshot Coding Menghitung Confusion Matrix dan Akurasi



Gambar 8 – Tabel Confusion Matrix

Nilai Akurasi didapat dengan:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

$$\frac{44 + 24}{44 + 24 + 5 + 7}$$

$$\frac{68}{80} = 0.85x100\% = \mathbf{85}\%$$

Dapat disimpulkan bahwa nilai akurasi dari model pohon keputusan yang dianalisis adalah 85%. Nilai akurasi menggambarkan sejauh mana model dapat memprediksi dengan benar kelas target (membeli atau tidak membeli) berdasarkan variabel input yang diberikan.

Dalam konteks ini, nilai akurasi sebesar 85% menunjukkan bahwa model pohon keputusan tersebut dapat memprediksi dengan benar sekitar 85% dari total data yang digunakan untuk evaluasi. Artinya, dari 80 data yang dievaluasi, sebanyak 68 data diprediksi dengan benar (membeli atau tidak membeli) oleh model. Nilai akurasi yang tinggi ini menunjukkan bahwa model pohon keputusan memiliki performa yang baik dalam memprediksi variabel target Purchased berdasarkan variabel input pada dataset yang digunakan.