TUGAS SCIENTIFIC COMPUTATION

KLASIFIKASI DATA HEART DISEASE MENGGUNAKAN METODE DECISION TREE

DESKRIPSI MASALAH

Jantung merupakan organ tubuh yang bekerja dengan memompa darah ke seluruh tubuh melalui sistem peredaran darah menurut Dr. Lawrence Phillips, seorang ahli jantung di NYU Langone Medical Center. Jantung merupakan salah satu organ tubuh yang paling penting. Ada banyak penyakit yang bisa dialami jantung, seperti serangan jantung, penyakit jantung koroner, dan gagal jantung. Pemeriksaan penyakit jantung dapat dilakukan melalui berbagai cara berikut EKG, Peneriksaan darah, dan Angiography. Angiografi adalah pemeriksaan pembuluh darah menggunakan zat kontras khusus dan memanfaatkan Rontgen. Hasil angiografi akan disebut normal jika aliran darah ke jantung normal dan tidak ada penyumbatan. Angiografi yang biasanya perlu waktu sekitar setengah hingga dua jam ini umumnya dijalankan di departemen radiologi rumah sakit dengan menggunakan pencitraan Rontgen. Singkatnya waktu pemeriksaan membuat pasien biasanya tidak perlu menginap dan dapat pulang di hari yang sama setelah selesai. Namun, angiografi tidak disarankan untuk dilakukan pada seseorang yang memiliki riwayat alergi dengan zat kontras, menderita gangguan pembekuan darah, kerusakan ginjal, memiliki tekanan darah tinggi yang sulit terkendali, aritmia, anemia, dan demam. Walau umumnya aman, angiografi berisiko menyebabkan seperti tekanan darah rendah, tamponade jantung, cedera pada arteri jantung, detak jantung tidak teratur, stroke, serangan jantung, kerusakan ginjal, dan walaupun relatif jarang, reaksi alergi. Prediksi Angiographic Disease Status menggunakan data Heart Disease yang berasal dari Kaggle. Sumber Data : https://archive.ics.uci.edu/ml/datasets/Heart+Disease

DESKRIPSI DATA

Data yang digunakan pada laporan ini adalah data sekunder yang diambil dari Kaggle. Data berisi 303 observasi yang menunjukkan jumlah pasien yang memeriksakan kesehatan dan diduga memiliki penyakit jantung. Terdapat 14 variabel, dengan rincian 13 variabel adalah hasil *screening* kesehatan dari masing-masing pasien yang mencakup umur, jenis kelamin, status merokok dan lain-lain. Sedangkan satu variabel terakhir adalah label, yaitu kode yang menunjukkan pasien tersebut menderita penyakit jantung atau tidak.

Selanjutnya variabel-variabel yang digunakan dalam laporan dijelaskan pada tabel berikut.

Tabel 1 Variabel Penelitian

Variabel	Nama Variabel	Tipe data
Y	Target	Diskrit
X_1	Age : umur pasien	Kontinyu
X_2	Sex : jenis kelamin pasien 1 : Laki-laki 0 : Perempuan	Diskrit
X_3	Cp : jenis nyeri dada 0 : *typical angina* 1 : *atypical angina* 2 : *non-anginal pain* 3 : *asymptomatic*	Diskrit
X_4	Trestbps : tekanan darah (dalam mm Hg saat masuk rumahsakit)	Kontinyu
X_5	Chol : kolesterol serum (dalam mg/dl)	Kontinyu
X_6	Fbs : keadaan gula darah (Fbs > 120) 1 : *true* 0 : *false*	Diskrit
X_7	Restecg : hasil elektrokardiografi 0 : normal 1 : memiliki kelainan ST-T 2 : menunjukkan hipertrofi ventrikel kiri	Diskrit
X_8	Thalach : detak jantung maksimum	Kontinyu
X_9	Exang : olahraga yang diinduksi olahraga	Diskrit

	1 : *yes* 0 : *no*	
X_{10}	Oldpeak : ST yang diinduksi oleh olahraga relatif terhadap istirahat	Kontinyu
X_{11}	Slope : kemiringan segmen ST 0 : *upsloping* 1 : *flat* 2 : *downsloping*	Diskrit
X_{12}	Ca : jumlah pembuluh darah besar (0-3) yang diwarnai dengan fluoroskopi	
X_{13}	Thal	Diskrit

Adapun struktur data yang digunakan dalam laporan ini adalah sebagai berikut.

Tabel 2. Struktur Data

Pasien ke-i	**Y**	**X_1**	**X_2**	⋯	**X_5**
1	Y_1	X_{11}	X_{21}	⋯	X_{51}
2	Y_2	X_{12}	X_{22}	⋯	X_{52}
⋮	⋮	⋮	⋮	⋯	⋮
303	Y_{303}	X_{303}	X_{303}	⋯	X_{303}

PENGEMBANGAN MODEL

Pengembangan model dilakukan dengan langkah-langkah sebagai berikut.

1. *Import* data

   Data yang digunakan berbentuk file excel dengan formal .csv, sehingga untuk import data menggunakan syntax sebagai berikut :

```
heart<-read.csv("D:/heart.csv", header = TRUE, sep = ",")
```

   Data excel diimport, dan disimpan di dalam data frame yang bernama *heart*.

2. *Cleaning* data

   Sebelum dilakukan analisis, langkah pertama yang dilakukan adalah *cleaning data*. Pertama dilakukan pengecekan apakah terdapat *missing value* di dalam data, menggunakan syntax sebagai berikut :

```
colSums(is.na(heart))
```

   Output dari syntax tersebut adalah 0. Artinya untuk masing-masing variabel tidak terdapat *missing value*. Berdasarkan hasil tersebut, sehingga tidak dilakukan *cleaning* data lanjutan.

   Untuk langkah *preprocessing* dilakukan juga penamaan label. Jadi, untuk setiap data kategorik diberikan label, gunanya untuk memberikan keterangan pada *output-output* yang diberikan. Synntax untuk memberikan label adalah sebagai berikut :

```
heart <- heart %>%
  mutate_if(is.integer, as.factor) %>%
  mutate(sex    =    factor(sex,    levels    =    c(0,1),    labels    =
c("Female","Male")),
         fbs =factor(fbs, levels = c(0,1), labels = c("False","True")),
         exang = factor(exang, levels = c(0,1), labels = c("No","Yes")),
         target = factor(target, levels = c(0,1),
                         labels = c("Health","Not Health")))
glimpse(heart)
```

   Berdasarkan cuplikan syntax tersebut, dapat diketahui bahwa variabel yang akan diberi label adalah *sex,* fbs, exang dan target.

3. Membentuk data *training* dan data *testing*.

   Dalam analisis klasifikasi, data akan dibagi menjadi dua yaitu data *training* dan data *testing*. Pembagian data *training* dan data *testing* pada penelitian ini yaitu 80% untuk *training* dan 20% untuk *testing*. Syntax yang digunakan adalah sebagai berikut :

```
set.seed(250)
intrain <- sample(nrow(heart),nrow(heart)*0.8)
heart_train <- heart[intrain, ]
heart_test <- heart[-intrain, ]
```

   Digunakan set.seed agar random data selalu sama meskipun dijalankan berulang-ulang.

   Setelah dilakukan pembagian data *training* dan data *testing*, selanjutnya dicek proporsi antar keduanya. Jadi ingin dipastikan kalau pada kedua data tersebut terbagi merata atau

balance. Syntax yang digunakan untuk cek data *balance* atau tidak adalah sebagai berikut:

```
prop.table(table(heart$target))
prop.table(table(heart_train$target))
prop.table(table(heart_test$target))
```

Hasil dari proporsi data seluruhnya yaitu 0,455 untuk kategori *health* dan 0,545 untuk kategori *not health*. Sedangkan untuk data *training*, didapatkan proporsi 0,442 untuk kategori *health* dan 0,558 untuk kategori *not health*. Dan untuk data *testing* didapatkan proporsi 0,508 untuk kategori *health* dan 0,492 untuk kategori *not health*. Berdasarkan hasil tersebut dapat diketahui bahwa ketiga data *balance*, dan juga untuk proporsi masing-masing data seluruhnya, data *training* dan data *testing* hampir sama. Data tersebut dapat dikatakan seimbang, sehingga dilanjutkan dengan pembangunan model menggunakan metode *decision tree.*

4. Membuat model

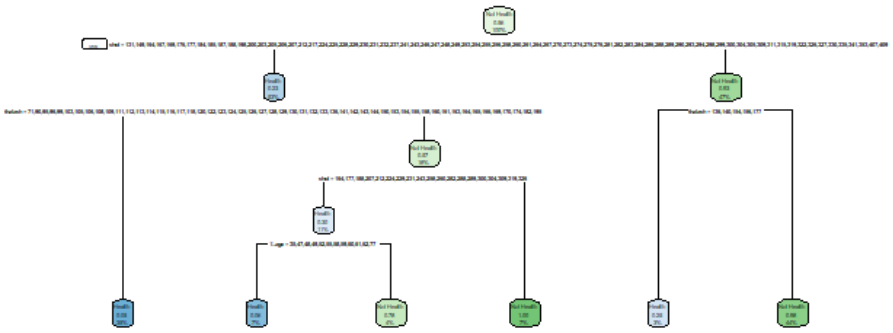
Setelah dipastikan data yang digunakan sudah *balanced*, maka dilakukan dengan pembentukan model. Membuat model dilakukan dengan menggunakan metode *decision tree*. Syntax untuk pembuatan model menggunakan metode *decision tree* adalah sebagai berikut :

```
fit <- rpart(target~., data = heart_train, method = 'class',control = rpart.control(cp = 0.01))
```

Untuk metode *decision tree* dibutuhkan *rpat*. Selanjutnya model tersebut divisualisasikan menjadi bentuk pohonnya. Syntax untuk memvisualisasikan adalah sebagai berikut:

```
rpart.plot(fit,extra=106)
```

Hasil dari visualisasi adalah sebagai berikut :



Gambar 1. Visualisasi Metode *Decision Tree*

Berdasarkan hasil visualisasi tersebut, terlihat bahwa terdapat 6 terminal *node*, 4 internal *node* dan 1 akar *node*. 1 *node* akar berisi kategori *not health*, terdapat kemungkinan eror sebesar 44,2%. Artinya dari 242 data pasien di data *training*, dapat diklasifikasikan 107 pasien *health* dan dan 135 *not health*. Selanjutnya pada *layer* 2 didapatkan 2 internal *node* yaitu *health* dan *not health*. Dengan 128 pasien observasi yang *health* dan 114 pasien yang *not health*. Berdasarkan *summary* fit juga didapatkan bahwa variabel-variabel yang dianggap penting adalah *Chol, Thalach, Age, Thal, Cp, Oldpeak, Trestbpsm* dan *slope*.

5. Membuat Prediksi

Setelah model terbentuk, selanjutnya dilakukan prediksi data *testing*. Akan diprediksi pasien mana yang mungkin *not health* dari 61 pasien yang diperiksa. Syntax yang digunakan adalah sebagai berikut :

```
predict_unseen <-predict(fit, heart_test, type = 'class')
```

Dilanjutkan dengan dibentuk tabel untuk memppermudah melihat berapa pasien yang *health* dan *not health* menggunakan syntax berikut:

```
table_mat <- table(heart_test$target, predict_unseen)
table_mat
```

Output dari syntax tersebut adalah sebagai berikut :

```
            predict_unseen
             Health Not Health
Health           18         13
Not Health       12         18
```

Gambar 2. *Output* Prediksi Model

Berdasarkan *output* tersebut dapat diketahui bahwa model mengklasifikasikan benar sebanyak 36 pasien. Artinya pasien yang *health* benar dikategorikan *health* dan pasien yang *not health* benar dikategorikan sebagai *not health*. Tetapi model tersebut juga salah mengklasifikasikan, yaitu 13 orang *health* dikategorikan sebagai *not health*. Selain itu 12 pasien *not health* dikategorikan sebagai *health*. Kesalahan pengkategorian yang terakhir ini sangat fatal karena akan merugikan pasien, menjadikan pasien yang sakit tidak terseteksi sakit.

6. Evaluasi model

Setelah melakukan prediksi terhadap data *test*, maka selanjutnya dilakukan evaluasi model. Hal ini untuk mengecek apakah model terjadi *underfitting* atau *overfitting*. Digunakan nilai akurasi untuk melihat apakah model sudah baik. Syntax untuk nilai akurasi adalah sebagai berikut:

```
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat))
```

Output dari syntax tersebut didapatkan nilai akurasi sebesar 59,02. Hal tersebut masih tergolong kecil. Untuk melihat parameter evaluasi model yang lain, digunakan juga *Confusion Matrix* dengan syntax sebagai berikut :

```
confusionMatrix(predict_unseen, heart_test$target)
```

Output dari *confusion matrix* adalah sebagai berikut :

```
Confusion Matrix and Statistics

            Reference
Prediction   Health Not Health
  Health         18         12
  Not Health     13         18

               Accuracy : 0.5902
                 95% CI : (0.4568, 0.7145)
    No Information Rate : 0.5082
    P-Value [Acc > NIR] : 0.1244

                  Kappa : 0.1805

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.5806
            Specificity : 0.6000
         Pos Pred Value : 0.6000
         Neg Pred Value : 0.5806
             Prevalence : 0.5082
         Detection Rate : 0.2951
   Detection Prevalence : 0.4918
      Balanced Accuracy : 0.5903

       'Positive' Class : Health
```

Gambar 3. *Confusion Matrix* dari Model

Terdapat tabel kategori prediksi seperti yang sudah dijelaskan sebelumnya. Terdapat juga nilai akurasi sebesar 59,02 dan sensitivity sebesar 0,58.

7. *Tune Hyper-Parameters*

Decision tree memiliki berbagai parameter yang mengontrol aspek kecocokan. Di *library decision tree*, dapat mengontrol parameter menggunakan funngsi *rpart.control ()*. Dalam kode berikut, digunakan untuk mengatur parameter

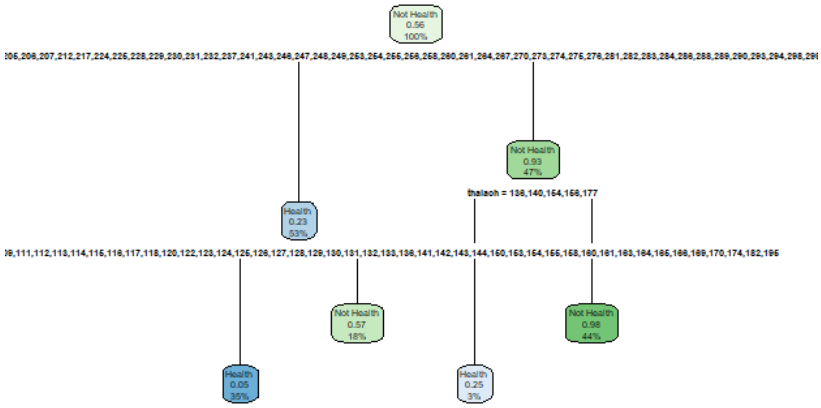
```
control <- rpart.control(minsplit = 3,
                         minbucket = round(5 / 3),
                         maxdepth = 2,
                         cp = 0)
```

Syntax tersebut adalah kontrol yang diberikan pada model yaitu *minsplit* = 3, Cp = 0 dan sebagaiannya. Selanjutnya dilakukan pemodelan ulang dengan tambahan kontrol tersebut sekaligus divisualisasikan hasilnya menggunakan syntax sebagai berikut :

```
tune_fit <- rpart(target~., data =heart_train, method = 'class', control
= control)

rpart.plot(tune_fit,extra=106)
```

Output visualisasinya adalah sebagai berikut:



Gambar 4. Visualisasi Setelah Kontrol Pertama

Berdasarkan hasil visualisasi tersebut, terlihat bahwa terdapat 4 terminal *node*, 2 internal *node* dan 1 akar *node*. 1 *node* akar berisi kategori *not health*, terdapat kemungkinan eror sebesar 44,2%. Artinya dari 242 data pasien di data *training*, dapat diklasifikasikan 107 pasien *health* dan dan 135 *not health*. Selanjutnya pada *layer* 2 didapatkan 2 internal *node* yaitu *health* dan *not health*. Dengan 128 pasien observasi yang *health* dan 114 pasien yang *not health*. Berdasarkan *summary* fit juga didapatkan bahwa variabel-variabel yang dianggap penting adalah *Chol, Thalach, Age, Thal, Cp, Oldpeak, Trestbpsm* dan *slope*.

Selanjutnya dilakukan evaluasi model terhadap model yang dibentuk berdasarkan kontrol pertama tersebut menggunakan syntax sebagai berikut :

```
accuracy_tune <- function(fit) {
  predict_unseen <- predict(fit, heart_test, type = 'class')
  table_mat <- table(heart_test$target, predict_unseen)
  accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
  accuracy_Test
  confusionMatrix(predict_unseen, heart_test$target)
}
accuracy_tune(tune_fit)
```

Hasil dari evaluasi model adalah sebagai berikut :

```
Confusion Matrix and Statistics

            Reference
Prediction   Health Not Health
  Health         17         10
  Not Health     14         20

               Accuracy : 0.6066
                 95% CI : (0.4731, 0.7293)
    No Information Rate : 0.5082
    P-Value [Acc > NIR] : 0.0791

                  Kappa : 0.2146

 Mcnemar's Test P-Value : 0.5403

            Sensitivity : 0.5484
            Specificity : 0.6667
         Pos Pred Value : 0.6296
         Neg Pred Value : 0.5882
             Prevalence : 0.5082
         Detection Rate : 0.2787
   Detection Prevalence : 0.4426
      Balanced Accuracy : 0.6075

       'Positive' Class : Health
```

Gambar 5. Evaluasi Model *Decision Tree* menggunakan Kontrol Pertama

Berdasarkan *confusion matrix* tersebut dapat diketahui bahwa kesalahan prediksi sudah berkurang. Selain itu juga terdapat peningkatan nilai akurasi, yang semula hanya 59,02 sekarang menjadi 60,66. Sehingga dapat disimpulkan bahwa kontrol pertama berhasil. Selanjutnya akan dicoba menggunakan kontrol kedua.

Kontrol kedua terdapat dalam kode berikut :

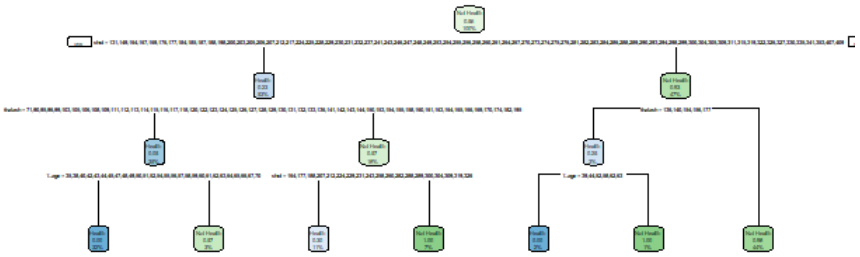
```
control <- rpart.control(minsplit = 4,
                         minbucket = round(4 / 3),
                         maxdepth = 3,
                         cp = 0)
```

Syntax tersebut adalah kontrol yang diberikan pada model yaitu *minsplit* = 4, Cp = 0 dan sebagaiannya. Selanjutnya dilakukan pemodelan ulang dengan tambahan kontrol tersebut sekaligus divisualisasikan hasilnya menggunakan syntax sebagai berikut :

```
tune_fit <- rpart(target~., data =heart_train, method = 'class', control
= control2)

rpart.plot(tune_fit,extra=106)
```

Output visualisasinya adalah sebagai berikut:



Gambar 6. Visualisasi Setelah Kontrol Kedua

Selanjutnya dilakukan evaluasi model terhadap model yang dibentuk berdasarkan kontrol kedua tersebut menggunakan syntax sebagai berikut :

```
accuracy_tune <- function(fit) {
  predict_unseen <- predict(fit, heart_test, type = 'class')
  table_mat <- table(heart_test$target, predict_unseen)
  accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
  accuracy_Test
  confusionMatrix(predict_unseen, heart_test$target)
}
accuracy_tune(tune_fit)
```

Hasil dari evaluasi model adalah sebagai berikut :

```
Confusion Matrix and Statistics

            Reference
Prediction   Health Not Health
  Health         16         11
  Not Health     15         19

               Accuracy : 0.5738
                 95% CI : (0.4406, 0.6996)
    No Information Rate : 0.5082
    P-Value [Acc > NIR] : 0.1851

                  Kappa : 0.1491

 Mcnemar's Test P-Value : 0.5563

            Sensitivity : 0.5161
            Specificity : 0.6333
         Pos Pred Value : 0.5926
         Neg Pred Value : 0.5588
             Prevalence : 0.5082
         Detection Rate : 0.2623
   Detection Prevalence : 0.4426
      Balanced Accuracy : 0.5747

       'Positive' Class : Health
```

Gambar 7. Evaluasi Model *Decision Tree* menggunakan Kontrol Kedua

Berdasarkan *confusion matrix* tersebut dapat diketahui bahwa kesalahan prediksi justru bertambah. Selain itu juga terdapat penurunan nilai akurasi jika dibandingkan dengan kontrol pertama, yang semula hanya 60,66 sekarang menjadi 57,38. Bahkan nilai tersebut masih lebih kecil daripada nilai akurasi model yang tanpa diber kontrol. Sehingga dapat disimpulkan bahwa kontrol kedua tidak berhasil. Selanjutnya dicoba menggunakan kontrol yang ketiga.

Kontrol ketiga terdapat dalam kode berikut :

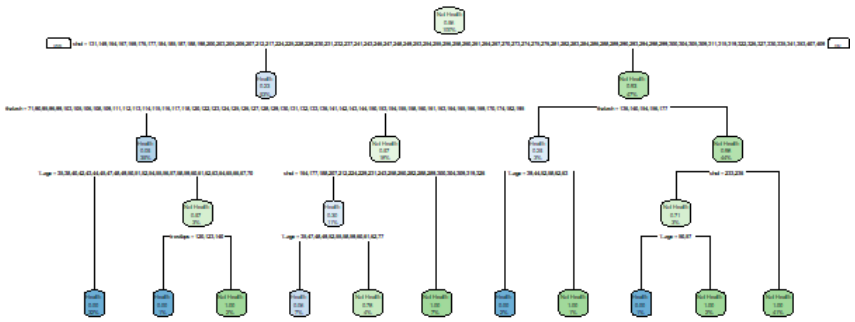
```
control <- rpart.control(minsplit = 5,
                         minbucket = round(4 / 3),
                         maxdepth = 4,
                         cp = 0)
```

Syntax tersebut adalah kontrol yang diberikan pada model yaitu *minsplit* = 5, Cp = 0 dan sebagaiannya. Selanjutnya dilakukan pemodelan ulang dengan tambahan kontrol tersebut sekaligus divisualisasikan hasilnya menggunakan syntax sebagai berikut :

```
tune_fit <- rpart(target~., data =heart_train, method = 'class', control
= control3)

rpart.plot(tune_fit,extra=106)
```

Output visualisasinya adalah sebagai berikut:



Gambar 8. Visualisasi Setelah Kontrol Ketiga

Selanjutnya dilakukan evaluasi model terhadap model yang dibentuk berdasarkan kontrol ketiga tersebut menggunakan syntax sebagai berikut :

```
accuracy_tune <- function(fit) {
  predict_unseen <- predict(fit, heart_test, type = 'class')
  table_mat <- table(heart_test$target, predict_unseen)
  accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
  accuracy_Test
```

```
  confusionMatrix(predict_unseen, heart_test$target)
}
accuracy_tune(tune_fit)
```

Hasil dari evaluasi model adalah sebagai berikut :

```
Confusion Matrix and Statistics

            Reference
Prediction   Health Not Health
  Health         17         11
  Not Health     14         19

               Accuracy : 0.5902
                 95% CI : (0.4568, 0.7145)
    No Information Rate : 0.5082
    P-Value [Acc > NIR] : 0.1244

                  Kappa : 0.1814

 Mcnemar's Test P-Value : 0.6892

            Sensitivity : 0.5484
            Specificity : 0.6333
         Pos Pred Value : 0.6071
         Neg Pred Value : 0.5758
             Prevalence : 0.5082
         Detection Rate : 0.2787
   Detection Prevalence : 0.4590
      Balanced Accuracy : 0.5909

       'Positive' Class : Health
```

Gambar 9. Evaluasi Model *Decision Tree* menggunakan Kontrol Ketiga

Berdasarkan *confusion matrix* tersebut dapat diketahui bahwa kesalahan prediksi berkurang daripada kontrol kedua. Selain itu juga terdapat kenaikan nilai akurasi jika dibandingkan dengan kontrol kedua, yang semula hanya 57,38 sekarang menjadi 59,02. Sehingga dapat disimpulkan bahwa kontrol kedua tidak berhasil.

KESIMPULAN

Berdasarkan paparan di atas dapat disimpulkan sebagai berikut :

1. Ketika melakukan klasifikasi terhadap suatu data set, maka terlebih dahulu harus memastikan bahwa variabel target sudah *balance*.
2. Jika terjadi *imbalance* data maka harus dilakukan modifikasi pada data set salah satunya resampling yang digunakan pada laporan.
3. Model evaluasi untuk data resampling lebih baik daripada data asli.
4. Kontrol yang sesuai untuk data *heart disease* adalah kontrol yang pertama karena memiliki nilai akurasi tertinggi dan kesalahan prediksi terkecil.
5. Model terbaik adalah model dengan menggunakan kontrol pertama yaitu dengan parameter *minsplit* = 3, *minbucket* = *round* (5/3), *maxdepth* = 2 dan *Cp* = 0. Hasil dari model terbaik yaitu didapatkan akurasi sebesar 60,66 dengan total kesalahan akurasi sebanyak 24 pasien.