TUGAS KELOMPOK

Disusun Guna Memenuhi Tugas Mata Kuliah Data Mining

Disusun oleh:

Kelompok 5

Najwa Rachel Kharizta	(24050121120027)
Zahra Ulaya Sifa	(24050121130097)
Hilda Ayu Meylia	(24050121140119)
Sekar Fajrina	(24050121140121)

Datmin Mining – D

DEPARTEMEN STATISTIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

2024

A. PEMBAHASAN

Syntax

Syntax yang digunakan dalam melakukan uji akurasi dengan metode *cart* dan *randomforest*

```
bankloan <- read.csv("D:/1. KULIAH PER SEMESTER/Kuliah Semester
6/Data Mining/SETELAH UTS/3. bankloan.csv", header=TRUE)
head(bankloan)
#set.seed(20)
acak <- sample(1:nrow(bankloan), 450, replace=FALSE)
bankloan.training <- bankloan[acak,]
bankloan.testing <- bankloan[-acak,]
library(rpart)
model.pohon <- rpart(as.factor(default) ~ age + ed + employ +
address
                     + income + debtinc + creddebt + othdebt,
                     data=bankloan.training)
prob.prediksi <- predict(model.pohon, bankloan.testing)
prediksi <- ifelse(prob.prediksi[,2] > 0.5, 1, 0)
tabel <- table(bankloan.testing$default, prediksi)
akurasi <- (tabel[1,1] + tabel[2,2])/sum(tabel)
akurasi

library(randomForest)
set.seed(100)
model.forest <- randomForest(as.factor(default) ~ age + ed + employ
+ address
                             + income + debtinc + creddebt +
othdebt,
                             data=bankloan.training,
importance=TRUE, ntree=2000, mtry=3)
prediksi.rf <- predict(model.forest, bankloan.testing)
tabel.rf <- table(bankloan.testing$default, prediksi.rf)
akurasi.rf <- (tabel.rf[1,1] + tabel.rf[2,2])/sum(tabel.rf)
akurasi.rf
#importance(model.forest)
varImpPlot(model.forest)
#getTree(model.forest, labelVar=TRUE, k=2)
for(i in 1:100){
  acak <- sample(1:nrow(bankloan), 450, replace=FALSE)
  bankloan.training <- bankloan[acak,]
  bankloan.testing <- bankloan[-acak,]
  model.pohon<- rpart(as.factor(default) ~ age + ed + employ +
address
                      + income + debtinc + creddebt + othdebt,
```

```
                        data=bankloan.training)
  prob.prediksi <- predict(model.pohon, bankloan.testing)
  prediksi <- ifelse(prob.prediksi[,2] > 0.5, 1, 0)
  tabel <- table(bankloan.testing$default, prediksi)
  akurasi[i] <- (tabel[1,1] + tabel[2,2])/sum(tabel)
  model.forest <- randomForest(as.factor(default) ~ age + ed +
employ + address
                               + income+debtinc + creddebt +
othdebt,
                               data=bankloan.training,
importance=TRUE, ntree=2000, mtry=3)
  prediksi.rf <- predict(model.forest, bankloan.testing)
  tabel.rf <- table(bankloan.testing$default, prediksi.rf)
  akurasi.rf[i] <- (tabel.rf[1,1] + tabel.rf[2,2])/sum(tabel.rf)
}
boxplot(cbind(akurasi, akurasi.rf))
plot(akurasi, akurasi.rf)
points(akurasi, akurasi, type="l")
```

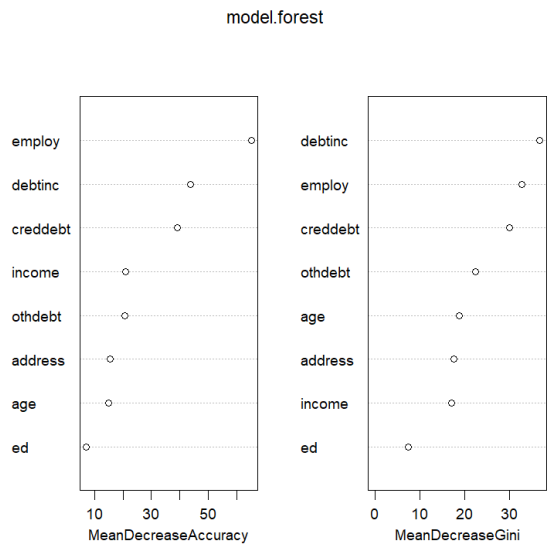
Syntax yang digunakan dalam melakukan penanganan *missing-values* pada data "bankloan" adalah sebagai berikut:

```
bankloan <- read.csv("D:/1. KULIAH PER SEMESTER/Kuliah Semester
6/Data Mining/SETELAH UTS/3. bankloan.csv", header=TRUE)
head(bankloan)
#PRE PROCESSING (ADRESS = MEAN & EMPLOY = MEAN)
# Menghitung mean untuk kolom employ dan address
mean_employ <- mean(bankloan$employ[bankloan$employ != 0], na.rm =
TRUE)
mean_address <- mean(bankloan$address[bankloan$address != 0], na.rm
= TRUE)

# Mengganti nilai 0 dengan mean
bankloan$employ[bankloan$employ == 0] <- mean_employ
bankloan$address[bankloan$address == 0] <- mean_address

# Menampilkan hasil untuk memverifikasi perubahan
head(bankloan)
```

Output

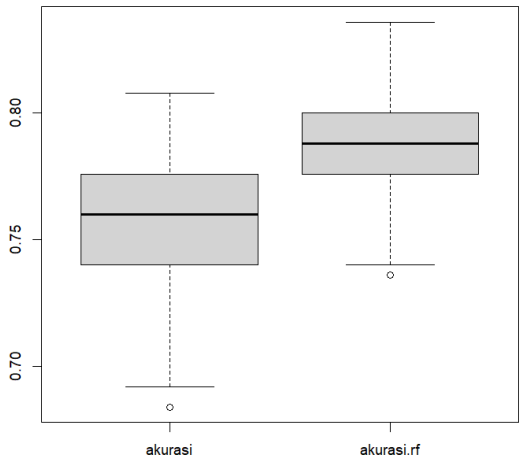
Output hasil uji akurasi dengan metode *cart* dan *randomforest* (sebelum pre-processing) adalah sebagai berikut:

```
> bankloan <- read.csv("D:/1. KULIAH PER SEMESTER/Kuliah Semester 6
/Data Mining/SETELAH UTS/3. bankloan.csv", header=TRUE)
> head(bankloan)
  age ed employ address income debtinc  creddebt  othdebt default
1  41  3     17      12    176     9.3 11.359392 5.008608       1
2  27  1     10       6     31    17.3  1.362202 4.000798       0
3  40  1     15      14     55     5.5  0.856075 2.168925       0
4  41  1     15      14    120     2.9  2.658720 0.821280       0
5  24  2      2       0     28    17.3  1.787436 3.056564       1
6  41  2      5       5     25    10.2  0.392700 2.157300       0
> set.seed(20)
> acak <- sample(1:nrow(bankloan), 450, replace=FALSE)
> bankloan.training <- bankloan[acak,]
> bankloan.testing <- bankloan[-acak,]
> library(rpart)
> model.pohon <- rpart(as.factor(default) ~ age + ed + employ + add
ress
+                      + income + debtinc + creddebt + othdebt,
+                      data=bankloan.training)
> prob.prediksi <- predict(model.pohon, bankloan.testing)
> prediksi <- ifelse(prob.prediksi[,2] > 0.5, 1, 0)
> tabel <- table(bankloan.testing$default, prediksi)
> akurasi <- (tabel[1,1] + tabel[2,2])/sum(tabel)
> akurasi
[1] 0.736
>
> library(randomForest)
> # Menghapus baris dengan missing values
> bankloan.training <- na.omit(bankloan.training)
> # Tetapkan seed untuk reproduktibilitas
> set.seed(100)
> # Model random forest
> model.forest <- randomForest(as.factor(default) ~ age + ed + empl
oy + address + income + debtinc + creddebt + othdebt,
+                              data = bankloan.training, importance
= TRUE, ntree = 2000, mtry = 3)
> prediksi.rf <- predict(model.forest, bankloan.testing)
> tabel.rf <- table(bankloan.testing$default, prediksi.rf)
> akurasi.rf <- (tabel.rf[1,1] + tabel.rf[2,2])/sum(tabel.rf)
> akurasi.rf
[1] 0.796
> #importance(model.forest)
> varImpPlot(model.forest)
```

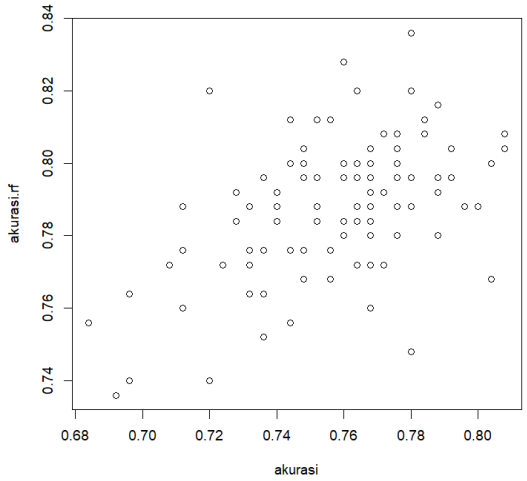



```
> #getTree(model.forest, labelVar=TRUE, k=2)
> # Inisialisasi vector akurasi
> akurasi <- numeric(100)
> akurasi.rf <- numeric(100)
```

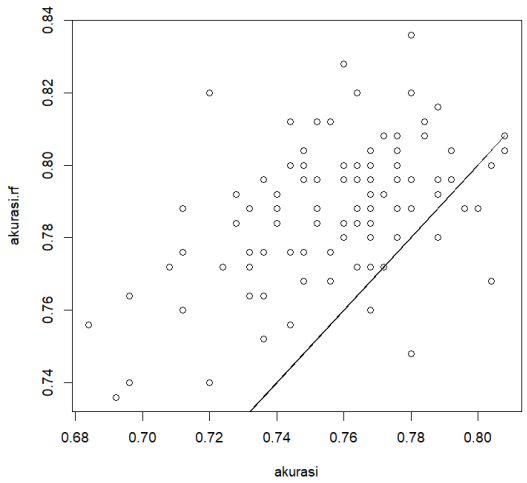
```
> # Tetapkan seed untuk reproduktibilitas
> set.seed(100)
>
> for (i in 1:100) {
+   acak <- sample(1:nrow(bankloan), 450, replace = FALSE)
+   bankloan.training <- bankloan[acak, ]
+   bankloan.testing <- bankloan[-acak, ]
+
+   # Uji model decision tree
+   model.pohon <- rpart(as.factor(default) ~ age + ed + employ + a
ddress + income + debtinc + creddebt + othdebt,
+                        data = bankloan.training)
+
+   # Prediksi dengan model decision tree
+   prob.prediksi <- predict(model.pohon, bankloan.testing)
+   prediksi <- ifelse(prob.prediksi[, 2] > 0.5, 1, 0)
+
+   # Menghitung akurasi untuk model decision tree
+   tabel <- table(bankloan.testing$default, prediksi)
+   akurasi[i] <- (tabel[1, 1] + tabel[2, 2]) / sum(tabel)
+
+   # Uji model random forest
+   model.forest <- randomForest(as.factor(default) ~ age + ed + em
ploy + address + income + debtinc + creddebt + othdebt,
+                                data = bankloan.training, importan
ce = TRUE, ntree = 2000, mtry = 3)
+
+   # Prediksi dengan model random forest
+   prediksi.rf <- predict(model.forest, bankloan.testing)
+
+   # Menghitung akurasi untuk model random forest
+   tabel.rf <- table(bankloan.testing$default, prediksi.rf)
+   akurasi.rf[i] <- (tabel.rf[1, 1] + tabel.rf[2, 2]) / sum(tabel.
rf)
+ }
>
> boxplot(cbind(akurasi, akurasi.rf))
```



```
> plot(akurasi, akurasi.rf)
```



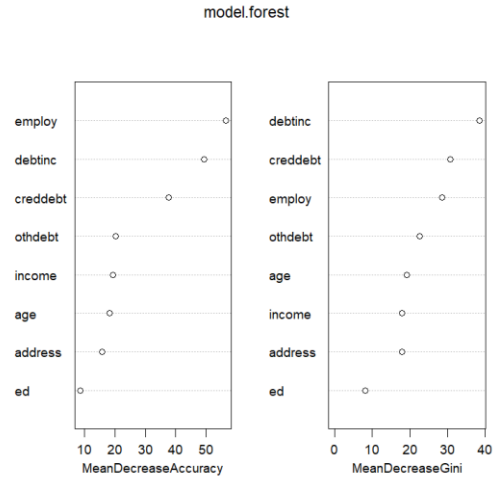

```
> points(akurasi, akurasi, type="l")
```



Output hasil uji akurasi dengan data "bankloan" yang telah dilakukan pre-processing *(missing-values)* adalah sebagai berikut:

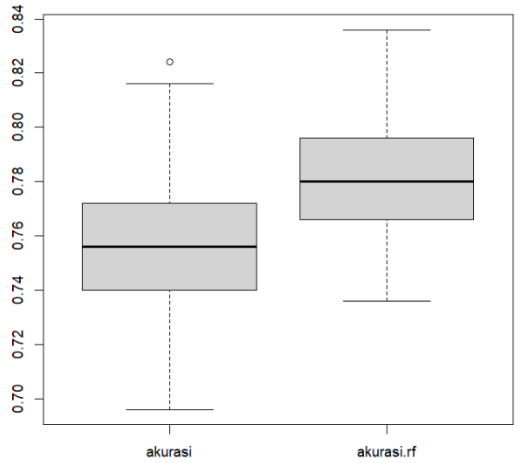
```
> bankloan <- read.csv("D:/1. KULIAH PER SEMESTER/Kuliah Semester 6
/Data Mining/SETELAH UTS/3. bankloan.csv", header=TRUE)
> head(bankloan)
  age ed employ address income debtinc  creddebt  othdebt default
1  41  3     17      12    176     9.3 11.359392 5.008608       1
2  27  1     10       6     31    17.3  1.362202 4.000798       0
3  40  1     15      14     55     5.5  0.856075 2.168925       0
4  41  1     15      14    120     2.9  2.658720 0.821280       0
5  24  2      2       0     28    17.3  1.787436 3.056564       1
6  41  2      5       5     25    10.2  0.392700 2.157300       0
>
> # Membuat histogram untuk address sebelum penanganan missing valu
e
> hist(bankloan$address, main="Histogram Address sebelum", xlab="Ad
dress")
>
> # Membuat histogram untuk employ sebelum penanganan missing value
> hist(bankloan$employ, main="Histogram Employ sebelum", xlab="Empl
oy")
```

```
>
> #PRE PROCESSING (ADRESS = MEAN & EMPLOY = MEDIAN)
> # Menghitung mean untuk kolom employ dan address
> mean_employ <- mean(bankloan$employ[bankloan$employ != 0], na.rm
= TRUE)
> mean_address <- mean(bankloan$address[bankloan$address != 0], na.
rm = TRUE)
>
> # Mengganti nilai 0 dengan mean
> bankloan$employ[bankloan$employ == 0] <- mean_employ
> bankloan$address[bankloan$address == 0] <- mean_address
>
> # Menampilkan hasil untuk memverifikasi perubahan
> head(bankloan)
  age ed employ   address income debtinc  creddebt  othdebt default
1  41  3     17 12.000000    176     9.3 11.359392 5.008608       1
2  27  1     10  6.000000     31    17.3  1.362202 4.000798       0
3  40  1     15 14.000000     55     5.5  0.856075 2.168925       0
4  41  1     15 14.000000    120     2.9  2.658720 0.821280       0
5  24  2      2  8.915385     28    17.3  1.787436 3.056564       1
6  41  2      5  5.000000     25    10.2  0.392700 2.157300       0
> set.seed(20)
> acak <- sample(1:nrow(bankloan), 450, replace=FALSE)
> bankloan.training <- bankloan[acak,]
> bankloan.testing <- bankloan[-acak,]
> library(rpart)
> model.pohon <- rpart(as.factor(default) ~ age + ed + employ + add
ress
+                      + income + debtinc + creddebt + othdebt,
+                      data=bankloan.training)
> prob.prediksi <- predict(model.pohon, bankloan.testing)
> prediksi <- ifelse(prob.prediksi[,2] > 0.5, 1, 0)
> tabel <- table(bankloan.testing$default, prediksi)
> akurasi <- (tabel[1,1] + tabel[2,2])/sum(tabel)
> akurasi
[1] 0.768
>
> library(randomForest)
> # Menghapus baris dengan missing values
> bankloan.training <- na.omit(bankloan.training)
> # Tetapkan seed untuk reproduktibilitas
> set.seed(100)
> # Model random forest
> model.forest <- randomForest(as.factor(default) ~ age + ed + empl
oy + address + income + debtinc + creddebt + othdebt,
+                              data = bankloan.training, importance
= TRUE, ntree = 2000, mtry = 3)
> prediksi.rf <- predict(model.forest, bankloan.testing)
> tabel.rf <- table(bankloan.testing$default, prediksi.rf)
> akurasi.rf <- (tabel.rf[1,1] + tabel.rf[2,2])/sum(tabel.rf)
> akurasi.rf
[1] 0.796
> #importance(model.forest)
> varImpPlot(model.forest)
```

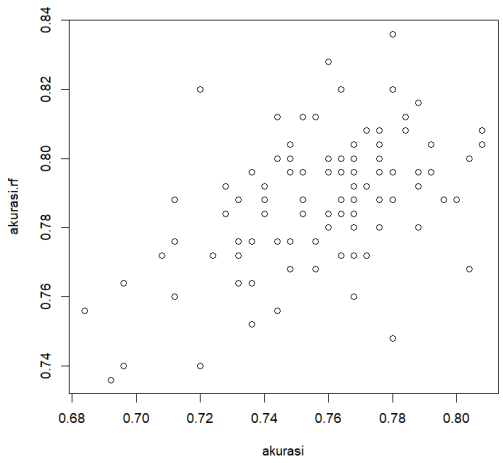



```
> #getTree(model.forest, labelVar=TRUE, k=2)
> # Initialize accuracy vectors
> akurasi <- numeric(100)
> akurasi.rf <- numeric(100)
>
> # Tetapkan seed untuk reproduktibilitas
> set.seed(100)
> for (i in 1:100) {
+   acak <- sample(1:nrow(bankloan), 450, replace = FALSE)
+   bankloan.training <- bankloan[acak, ]
+   bankloan.testing <- bankloan[-acak, ]
+
+   # Uji model decision tree
+   model.pohon <- rpart(as.factor(default) ~ age + ed + employ + a
ddress + income + debtinc + creddebt + othdebt,
+                        data = bankloan.training)
+
+   # Prediksi dengan model decision tree
+   prob.prediksi <- predict(model.pohon, bankloan.testing)
+   prediksi <- ifelse(prob.prediksi[, 2] > 0.5, 1, 0)
+
+   # Menghitung akurasi untuk model decision tree
+   tabel <- table(bankloan.testing$default, prediksi)
+   akurasi[i] <- (tabel[1, 1] + tabel[2, 2]) / sum(tabel)
+
+   # Uji model random forest
+   model.forest <- randomForest(as.factor(default) ~ age + ed + em
ploy + address + income + debtinc + creddebt + othdebt,
+                                data = bankloan.training, importan
ce = TRUE, ntree = 2000, mtry = 3)
+
+   # Prediksi dengan model random forest
+   prediksi.rf <- predict(model.forest, bankloan.testing)
+
+   # Menghitung akurasi untuk model random forest
+   tabel.rf <- table(bankloan.testing$default, prediksi.rf)
+   akurasi.rf[i] <- (tabel.rf[1, 1] + tabel.rf[2, 2]) / sum(tabel.
rf)
+ }
>
```

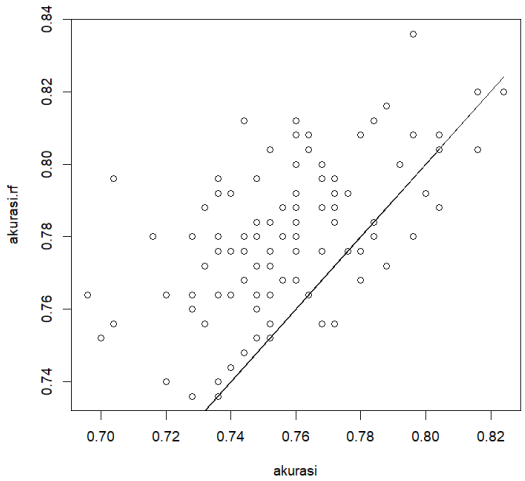
```
> boxplot(cbind(akurasi, akurasi.rf))
```




```
> plot(akurasi, akurasi.rf)
```




```
> points(akurasi, akurasi, type="l")
```



B. KESIMPULAN

Berdasarkan output program R diatas, menghasilkan kesimpulan sebagai berikut:

1. Penanganan *Missing Value* (data 0) pada *Feature Variables*

   Terdapat 2 variabel yang terdapat *missing values*, yaitu *employ* dan *address*. Pada 2 variabel tersebut, dilakukan beberapa cara dengan menghasilkan beberapa akurasi terbaik menggunakan mean dan median (modus tidak direkomendasikan karena tidak direkomendasikan pada variabel yang bukan berupa data kategorik), sbb:

No	Penanganan Missing Values		Nilai Akurasi	
	Employ	Address	CART	Random Forest
1	Mean	Mean	0.768	0.796
2	Median	Median	0.76	0.796
3	Mean	Median	0.768	0.792
4	Median	Mean	0.76	0.784

   Dari hasil tersebut didapat bahwa penanganan *missing value* terbaik terdapat pada penggunaan mean untuk variabel *employ* dan variabel *address* dikarenakan memiliki tingkat akurasi paling tinggi diantara yang lain, yaitu 0.768 untuk CART dan 0.796 untuk Random Forest.

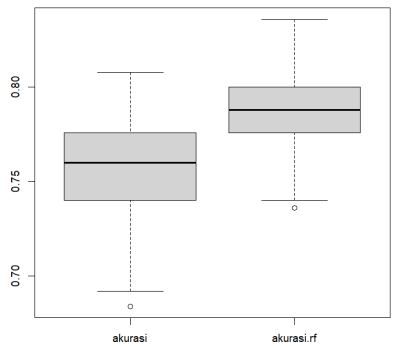
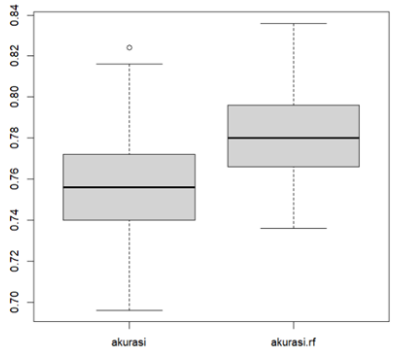
2. Perbandingan akurasi antara data asli dengan data setelah dilakukan penganganan *missing values*

Data Asli		Setelah Dilakukan Penanganan *Missing Values*	
CART	Random Forest	CART	Random Forest
0.736	0.796	0.768	0.796

   Berdasarkan tabel tersebut, dapat dilihat bahwa akurasi model CART mengalami peningkatan dari 0.736 menjadi 0.768 setelah dilakukan penanganan missing values, sedangkan akurasi model Random Forest tetap sama pada nilai 0.796 sebelum dan sesudah penanganan missing values. Hal ini menunjukkan bahwa penganganan missing values tidak memiliki dampak pada akurasi model Random Forest. Sehingga, dalam hal peningkatan akurasi setelah penanganan

missing values, model CART menunjukkan perbaikan yang signifikan dibandingkan dengan model Random Forest.

3. Perbandingan akurasi antara data asli dengan data setelah dilakukan penanganan missing-values berdasarkan boxplot

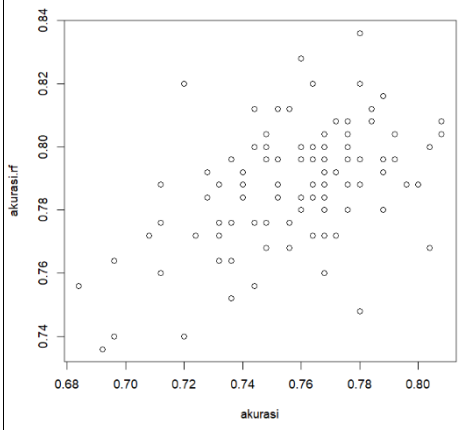
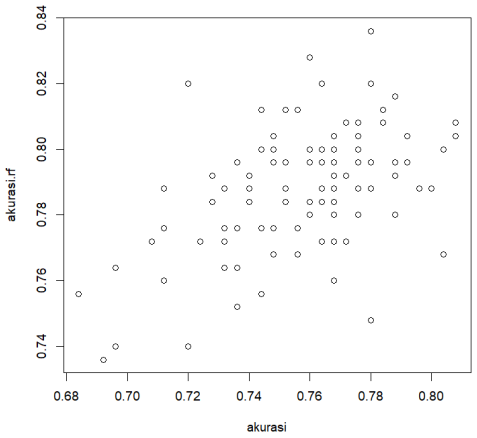
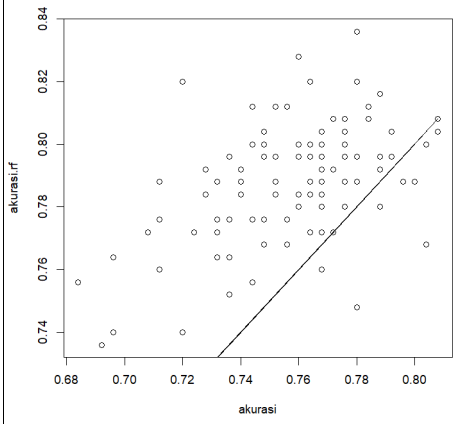
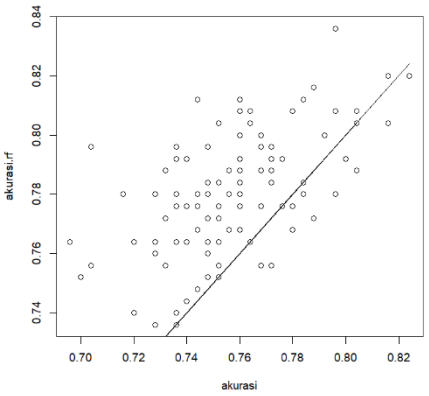
Data Asli	Setelah Dilakukan Penanganan *Missing Values*
	

Berdasarkan output boxplot diatas dapat disimpulkan bahwa

- Variasi akurasi lebih kecil di boxplot setelah penanganan *missing values* dimana akurasinya memiliki rentang antar kuartil yang lebih sempit dan lebih sedikit outliers dibandingkan dengan di boxplot data asli yang menunjukkan variasi hasil yang lebih kecil.
- Akurasi.rf lebih baik di kedua boxplot yang ditunjukkan pada median akurasi.rf lebih tinggi daripada akurasi di kedua boxplot. Hal tersebut menunjukkan bahwa model akurasi.rf secara umum memiliki akurasi yang lebih tinggi.
- Konsistensi akurasi.rf lebih baik di boxplot setelah penanganan *missing values* yang ditunjukkan dengan tidak adanya outlier pada akurasi.rf dimana hal tersebut menunjukkan konsistensi hasil yang lebih baik dibandingkan dengan boxplot data asli akurasi.rf yang memiliki satu outlier.
- Rentang akurasi.rf lebih besar di boxplot setelah penanganan *missing values* dimana hal tersebut menunjukkan rentang antar kuartil yang lebih

lebar untuk akurasi.rf, mengindikasikan variasi yang lebih besar dalam hasil akurasinya.

4. Perbandingan hasil akurasi antara data asli dengan data yang telah dilakukan penanganan *missing values* berdasarkan *plot* dan *points*

Data Asli	Setelah Dilakukan Penanganan *Missing Values*
	
	

Berdasarkan output *plot* dan *points* diatas, maka dapat disimpulkan sebagai berikut:

- Scatterplot pada data asli dan data setelah dilakukan penanganan *missing-values* sama-sama menunjukkan korelasi yang positif antara “akurasi” dan “akurasi.rf”. Namun, scatterplot data asli memiliki penyebaran titik data yang sedikit lebih terpusat atau lebih tersebar ditengah-tengah grafik dibandingkan dengan scatterplot pada data yang

telah dilakukan penanganan missing-values yang dimana tampak lebih tersebar ke bagian atas grafik.

- Pada grafik points data asli dan data setelah dilakukan penanganan *missing-values*, garis regresi nya sama-sama menunjukkan tren/korelasi positif. Akan tetapi pada data asli, lebih banyak titik yang menyimpang dari garis tersebut sehingga menunjukkan variabilitas yang lebih tinggi sedangkan pada data setelah dilakukan penanganan *missing-values* sebaran data lebih konsisten di sekitar garis sehingga menunjukkan hubungan yang lebih kuat dan konsisten antara “akurasi” dan “akurasi.rf”.