DAFTAR PUSTAKA

Adekitan, A. I., & Salau, O. (2019). The impact of engineering students' performance in the first three years on their graduation result using educational data mining. *Heliyon*, *5*(2), Article 2. https://doi.org/10.1016/j.heliyon.2019.e01250

Alias, N. A., & Zainuddin, A. M. (2005). *Innovation for Better Teaching and Learning: Adopting the Learning Management System*.

Aqsha, M., Thamrin, S., & Lawi, A. (2021). Combination of ADASYN-N and Random Forest in Predicting of Obesity Status in Indonesia: A Case Study of Indonesian Basic Health Research 2013. *Journal of Physics: Conference Series*, *2123*(1), 012039. https://doi.org/10.1088/1742-6596/2123/1/012039

Berry, M. W., & Browne, M. (2006). *Lecture Notes in Data Mining*. World Scientific.

Bilal Zorić, A. (2020). Benefits of Educational Data Mining. *Journal of International Business Research and Marketing*, *6*(1), Article 1. https://doi.org/10.18775/jibrm.1849-8558.2015.61.3002

Brefeld, U., & Scheffer, T. (t.t.). *AUC Maximizing Support Vector Learning*.

Breiman, L. (2001). Random Forests. *2001 Kluwer Academic Publishers. Manufactured in The Netherlands*.

Bulut, O., & Yavuz, H. C. (2019b). Educational data mining: A tutorial for the rattle package in R. *International Journal of Assessment Tools in Education*, *6*(5), Article 5. https://doi.org/10.21449/ijate.627361

Hanamian, H. (2021). *Random Forest adalah algoritma Supervised Learning yang digunakan untuk klasifikasi maupun regresi dengan cara membangun Multiple Decision Tree pada saat proses training data, kemudian mendapatkan beberapa prediksi kemudian dipilih satu solusi terbaik*.

Hidayat, M. M., Yani, J. A., & Telp, S. (2015). Educational Data Mining Untuk Meningkatkan Pendukung Keputusan Dalam Sistem Informasi Sekolah.

Huda, A. S., Awangga, R. M., & Fathonah, R. N. S. (2020). *Prediksi Penerimaan Pegawai Baru Dengan Metode Naive Bayes*. Kreatif.

Jovanovic, M., Vukicevic, M., Milovanovic, M., & Minovic, M. (2012a). Using data mining on student behavior and cognitive style data for improving e-learning systems: A case study: *International Journal of Computational Intelligence Systems*, *5*(3), 597. https://doi.org/10.1080/18756891.2012.696923

Jovanovic, M., Vukicevic, M., Milovanovic, M., & Minovic, M. (2012b). Using data mining on student behavior and cognitive style data for improving e-learning systems: A case study: *International Journal of Computational Intelligence Systems*, *5*(3), Article 3. https://doi.org/10.1080/18756891.2012.696923

Kamath, R. S., Kamath, R. S., & Kamat, R. K. (2016). *Educational data mining with R and Rattle*. River Publishing.

). Aplikasi Klasifikasi Dan Pohon Regresi (Classification And 'rees) Pada Data Tidak Seimbang (Studi Kasus : Data Obesitas Di

Naive Bayesian Classifier.

ely, E. (2015). Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan isifikasi Data Mining. 6.

Medium. 2018. https://medium.com/iykra/mengenal-software-r-dan-manfaatnya-bagi-data-scientist-bac79e72a785 (diakses tanggal 12 juli 2021)

Medium. 2020. https://medium.com/@grohith327 (diakses tanggal 15 juli 2021)

Mohd Kasim, N. N., & Khalid, F. (2016). Choosing the Right Learning Management System (LMS) for the Higher Education Institution Context: A Systematic Review. *International Journal of Emerging Technologies in Learning (IJET)*, *11*(06), 55. https://doi.org/10.3991/ijet.v11i06.5644

Mueen, A., Zafar, B., & Manzoor, U. (2016). Modeling and Predicting Students' Academic Performance Using Data Mining Techniques. *International Journal of Modern Education and Computer Science*, *8*(11), Article 11. https://doi.org/10.5815/ijmecs.2016.11.05

Muhamad, H., Prasojo, C. A., Sugianto, N. A., Surtiningsih, L., & Cholissodin, I. (2017). Optimasi Naïve Bayes Classifier Dengan Menggunakan Particle Swarm Optimization Pada Data Iris. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *4*(3), Article 3. https://doi.org/10.25126/jtiik.201743251

Purba, R. (2012). Data Mining: Masa Lalu, Sekarang, dan Masa Mendatang. *Jurnal SIFO Mikroskil*, *13*(1), Article 1. https://doi.org/10.55601/jsm.v13i1.44

Rasila, I., & Ristian, U. (2019). Implementasi Metode Naive Bayes Classifier Pada Sistem Pengklasifikasi Berita Otomatis Berbasis Website (Studi Kasus: Berita Lokal Dari Mediamassa Online Kalimantan Barat) (No. 2). *07*(2), Article 2.

Sastrawan, A. S., Gunadi, I. G. A., & Sukajaya, I. N. (2019). Perbandingan Kinerja Algoritma Dempster Shafer Dan Fuzzy-Naive Bayes Dalam Klasifikasi Penyakit Demam Berdarah Dan Tifus (No. 2). *2*, Article 2.

Siburian, V. W., & Mulyana, I. E. (2018). *Prediksi Harga Ponsel Menggunakan Metode Random Forest*.

Sutoyo, E., & Almaarif, A. (2020). Educational Data Mining for Predicting Student Graduation Using the Naïve Bayes Classifier Algorithm. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *4*(1), Article 1. https://doi.org/10.29207/resti.v4i1.1502

Tibshirani, S., & Friedman, H. (2008). *Valerie and Patrick Hastie*.

Widyahastuti, F., & Tjhin, V. U. (2018). Performance Prediction in Online Discussion Forum: State-of-the-art and comparative analysis. *Procedia Computer Science*, *135*, 302–314. https://doi.org/10.1016/j.procs.2018.08.178

Yaacob, W. F. W., Nasir, S. A. M., Yaacob, W. F. W., & Sobri, N. M. (2019a). Supervised data mining approach for predicting student performance. *Indonesian Journal of Electrical Engineering and Computer Science*, *16*(3), Article 3. https://doi.org/10.11591/ijeecs.v16.i3.pp1584-1592

Yahya, Y. (2018). *Klasifikasi Ketepatan Lama Studi Mahasiswa Menggunakan Metode [illegible] tor Machine Dan Random Forest (Studi Kasus: Data Lama Studi [illegible] rsitas Islam Indonesia Tahun Kelulusan 2000-2017*.

LAMPIRAN

Lampiran 1. Memuat *Packages*, Import Data, Pembagian Data Dan Pembentukan Model *Random Forest*

```
#panggil_packages_menggunakan_library
install.packages("e1071")
install.packages("haven")
install.packages("randomForest")
install.packages("memisc")
install.packages("rpart")
install.packages("rpart.plot")
install.packages("DMwR2")
install.packages("caret")
install.packages("pROC")
install.packages("ROCR")
install.packages("dplyr")
library(randomForest)
library(memisc)
library(rpart)
library(rpart.plot)
library(DMwR2)
library(caret)
library(pROC)
library(ROCR)
library(dplyr)
library(e1071)
oard")
```

```
summary(dat)
dat$y=as.factor(dat$y)
data_set_size=floor(nrow(dat)*0.80)
index=sample(1:nrow(dat),size = data_set_size)
training=dat[index,]
testing=dat[-index,]
#model random forest
#mencari model random forest menggunakan data training
r251=randomForest(y~.,data = training,mtry=1,ntree=25,importance=TRUE)
r252=randomForest(y~.,data = training,mtry=2,ntree=25,importance=TRUE)
r254=randomForest(y~.,data = training,mtry=4,ntree=25,importance=TRUE)
r501=randomForest(y~.,data = training,mtry=1,ntree=50,importance=TRUE)
r502=randomForest(y~.,data = training,mtry=2,ntree=50,importance=TRUE)
r504=randomForest(y~.,data = training,mtry=4,ntree=50,importance=TRUE)
r1001=randomForest(y~.,data = training,mtry=1,ntree=100,importance=TRUE)
r1002=randomForest(y~.,data = training,mtry=2,ntree=100,importance=TRUE)
r1004=randomForest(y~.,data = training,mtry=4,ntree=100,importance=TRUE)
r2001=randomForest(y~.,data = training,mtry=1,ntree=200,importance=TRUE)
r2002=randomForest(y~.,data = training,mtry=2,ntree=200,importance=TRUE)
r2004=randomForest(y~.,data = training,mtry=4,ntree=200,importance=TRUE)
#prediksi menggunakan data testing
pred251=data.frame(testing,predict(r251,testing))
pred252=data.frame(testing,predict(r252,testing))
pred253=data.frame(testing,predict(r253,testing))
testing,predict(r254,testing))
testing,predict(r255,testing))
testing,predict(r256,testing))
testing,predict(r501,testing))
```

```
pred502=data.frame(testing,predict(r502,testing))
pred503=data.frame(testing,predict(r503,testing))
pred504=data.frame(testing,predict(r504,testing))
pred505=data.frame(testing,predict(r505,testing))
pred506=data.frame(testing,predict(r506,testing))
pred1001=data.frame(testing,predict(r251,testing))
pred1002=data.frame(testing,predict(r252,testing))
pred1003=data.frame(testing,predict(r253,testing))
pred1004=data.frame(testing,predict(r254,testing))
pred1005=data.frame(testing,predict(r255,testing))
pred1006=data.frame(testing,predict(r256,testing))
pred2001=data.frame(testing,predict(r251,testing))
pred2002=data.frame(testing,predict(r252,testing))
pred2003=data.frame(testing,predict(r253,testing))
pred2004=data.frame(testing,predict(r254,testing))
pred2005=data.frame(testing,predict(r255,testing))
pred2006=data.frame(testing,predict(r256,testing))
#confusion matrix pada data hasil prediksi dengan data desting
cm=confusionMatrix(pred251$predict.r251..testing.,testing$y)
```

Output Out of Bag (OOB)

```
> rf251 (B = 25 dan m = 1)

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 1, ntree = 25,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 25
No. of variables tried at each split: 1

        OOB estimate of  error rate: 13.52%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive     24651   11515 0.318392966
Sensing         360   51317 0.006966349

> rf252

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 2, ntree = 25,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 25
No. of variables tried at each split: 2

        OOB estimate of  error rate: 14.4%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive     26129   10037  0.27752585
Sensing        2612   49067  0.05054277

> rf254

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 4, ntree = 25,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 25
No. of variables tried at each split: 4

                              f  error rate: 15.13%

                              ıg class.error
                              460  0.26157164
                              7851  0.07407264
```

```
          Intuitive Sensing class.error
Intuitive     26829    9337  0.25817066
Sensing        3993   47686  0.07726543
```

> rf501

```
Call:
 randomForest(formula = y ~ ., data = dl, mtry = 1, ntree = 50,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 50
No. of variables tried at each split: 1

        OOB estimate of  error rate: 13.49%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive     24567   11599 0.320715589
Sensing         252   51427 0.004876255
```

> rf502

```
Call:
 randomForest(formula = y ~ ., data = dl, mtry = 2, ntree = 50,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 50
No. of variables tried at each split: 2

        OOB estimate of  error rate: 13.62%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive     25824   10342  0.28595919
Sensing        1625   50054  0.03144411
```

> rf504

```
Call:
 randomForest(formula = y ~ ., data = dl, mtry = 4, ntree = 50,      importance = TRUE)
                    dom forest: classification
                     of trees: 50
                    at each split: 4

                    f  error rate: 14.12%

                    g class.error
```

```
Intuitive    26441    9725  0.26889897
Sensing       2676   49003  0.05178119

> rf506

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 6, ntree = 50,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 50
No. of variables tried at each split: 6

        OOB estimate of  error rate: 14.32%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive    26490    9676  0.26754410
Sensing       2901   48778  0.05613499

> rf1001

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 1, ntree = 100,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 1

        OOB estimate of  error rate: 13.51%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive    24522   11644  0.32195985
Sensing        226   51453  0.00437315

> rf1002

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 2, ntree = 100,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 2

                          f  error rate: 13.34%

                          ıg class.error
                          )552  0.29176575
                          0515  0.02252366
```

```
> rf1004

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 4, ntree = 100,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 4

        OOB estimate of  error rate: 13.74%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive     26216    9950  0.27512028
Sensing        2119   49560  0.04100312

> rf2001

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 1, ntree = 200,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 200
No. of variables tried at each split: 1

        OOB estimate of  error rate: 13.52%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive     24471   11695 0.323370016
Sensing         179   51500 0.003463689

> rf2002

Call:
 randomForest(formula = y ~ ., data = dl, mtry = 2, ntree = 200,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 200
No. of variables tried at each split: 2

        OOB estimate of  error rate: 13.24%
Confusion matrix:
                     ig class.error
                    670  0.29502848
                    718  0.01859556
```

```
 randomForest(formula = y ~ ., data = dl, mtry = 4, ntree = 200,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 200
No. of variables tried at each split: 4

        OOB estimate of  error rate: 13.58%
Confusion matrix:
          Intuitive Sensing class.error
Intuitive     26138   10028  0.27727700
Sensing        1901   49778  0.03678477
```

Output Confusion Matrix pada Data Hasil Prediksi dengan Data Testing

```
> cm251
Confusion Matrix and Statistics

          Reference
Prediction  Intuitive Sensing
  Intuitive     14730     162
  Sensing         858    6212

               Accuracy : 0.9536
                 95% CI : (0.9507, 0.9563)
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8908

 Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9450
Specificity : 0.9746
Pos Pred Value : 0.9891
Neg Pred Value : 0.8786
Prevalence : 0.7098
Detection Rate : 0.6707
Detection Prevalence : 0.6781
Balanced Accuracy : 0.9598

       'Positive' Class : Intuitive

> cm252
Confusion Matrix and Statistics

          Reference
Prediction  Intuitive Sensing
  Intuitive     14771     267
  Sensing         817    6107

               Accuracy : 0.9506
                          0.9477, 0.9535)
                          e : 0.7098
                          ] : < 2.2e-16

                          .8832

                          alue : < 2.2e-16
```

```
Sensitivity : 0.9476
Specificity : 0.9581
Pos Pred Value : 0.9822
Neg Pred Value : 0.8820
Prevalence : 0.7098
Detection Rate : 0.6726
Detection Prevalence : 0.6847
Balanced Accuracy : 0.9528

     'Positive' Class : Intuitive

> cm254
Confusion Matrix and Statistics

          Reference
Prediction  Intuitive Sensing
  Intuitive     14802     437
  Sensing         786    5937

             Accuracy : 0.9443
               95% CI : (0.9412, 0.9473)
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                Kappa : 0.867

 Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9496
Specificity : 0.9314
Pos Pred Value : 0.9713
Neg Pred Value : 0.8831
Prevalence : 0.7098
Detection Rate : 0.6740
Detection Prevalence : 0.6939
Balanced Accuracy : 0.9405

     'Positive' Class : Intuitive

                          Statistics
```

```
Prediction  Intuitive Sensing
  Intuitive     14723     149
  Sensing         865    6225

               Accuracy : 0.9538
                 95% CI : (0.951, 0.9566)
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8915

 Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9445
Specificity : 0.9766
Pos Pred Value : 0.9900
Neg Pred Value : 0.8780
Prevalence : 0.7098
Detection Rate : 0.6704
Detection Prevalence : 0.6772
Balanced Accuracy : 0.9606

       'Positive' Class : Intuitive

> cm502
Confusion Matrix and Statistics

           Reference
Prediction  Intuitive Sensing
  Intuitive     14772     261
  Sensing         816    6113

               Accuracy : 0.951
                 95% CI : (0.948, 0.9538)
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.884

                         alue : < 2.2e-16

                         326
                         822
```

Prevalence : 0.7098
Detection Rate : 0.6726
Detection Prevalence : 0.6845
Balanced Accuracy : 0.9534

'Positive' Class : Intuitive

> cm504
Confusion Matrix and Statistics

Reference

Prediction	Intuitive	Sensing
Intuitive	14796	395
Sensing	792	5979

Accuracy : 0.946
95% CI : (0.9429, 0.9489)
No Information Rate : 0.7098
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.8712

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9492
Specificity : 0.9380
Pos Pred Value : 0.9740
Neg Pred Value : 0.8830
Prevalence : 0.7098
Detection Rate : 0.6737
Detection Prevalence : 0.6917
Balanced Accuracy : 0.9436

'Positive' Class : Intuitive

> cm1001
Confusion Matrix and Statistics

ensing
150
6224

).9536
).9508, 0.9564)

```
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8911

 Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9443
Specificity : 0.9765
Pos Pred Value : 0.9899
Neg Pred Value : 0.8776
Prevalence : 0.7098
Detection Rate : 0.6702
Detection Prevalence : 0.6771
Balanced Accuracy : 0.9604

       'Positive' Class : Intuitive

> cm1002
Confusion Matrix and Statistics

            Reference
Prediction  Intuitive Sensing
  Intuitive     14760     225
  Sensing         828    6149

               Accuracy : 0.9521
                 95% CI : (0.9491, 0.9548)
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8868

 Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9469
Specificity : 0.9647
Pos Pred Value : 0.9850
Neg Pred Value : 0.8813

                                 21
                               : 0.6823
                                 0.9558

                                 Intuitive
```

```
> cm1004
Confusion Matrix and Statistics

          Reference
Prediction  Intuitive Sensing
  Intuitive     14800     400
  Sensing         788    5974

               Accuracy : 0.9459
                 95% CI : (0.9428, 0.9489)
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.871

 Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9494
Specificity : 0.9372
Pos Pred Value : 0.9737
Neg Pred Value : 0.8835
Prevalence : 0.7098
Detection Rate : 0.6739
Detection Prevalence : 0.6921
Balanced Accuracy : 0.9433

       'Positive' Class : Intuitive

> cm2001
Confusion Matrix and Statistics

          Reference
Prediction  Intuitive Sensing
  Intuitive     14717     147
  Sensing         871    6227

                        ).9536
                        ).9508, 0.9564)
                        :e : 0.7098
                        ₹] : < 2.2e-16

                        .8911
```

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9441
Specificity : 0.9769
Pos Pred Value : 0.9901
Neg Pred Value : 0.8773
Prevalence : 0.7098
Detection Rate : 0.6701
Detection Prevalence : 0.6768
Balanced Accuracy : 0.9605

'Positive' Class : Intuitive

> cm2002

Confusion Matrix and Statistics

	Reference	
Prediction	Intuitive	Sensing
Intuitive	14758	227
Sensing	830	6147

Accuracy : 0.9519
95% CI : (0.949, 0.9547)
No Information Rate : 0.7098
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.8864

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9468
Specificity : 0.9644
Pos Pred Value : 0.9849
Neg Pred Value : 0.8810
Prevalence : 0.7098
Detection Rate : 0.6720
Detection Prevalence : 0.6823
0.9556

Intuitive

Statistics

```
          Reference
Prediction  Intuitive Sensing
  Intuitive     14800     377
  Sensing         788    5997

               Accuracy : 0.947
                 95% CI : (0.9439, 0.9499)
    No Information Rate : 0.7098
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8737

 Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 0.9494
Specificity : 0.9409
Pos Pred Value : 0.9752
Neg Pred Value : 0.8839
Prevalence : 0.7098
Detection Rate : 0.6739
Detection Prevalence : 0.6911
Balanced Accuracy : 0.9452

       'Positive' Class : Intuitive
```

Jumlah Tree	Mtry		
	1	2	4
25 trees	13,52%	14,40%	15,13%
50 trees	13,49%	13,62%	14,12%
100 trees	13,51%	13,34%	13,74%
200 trees	13,52%	13,24%	13,58%

Berdasarkan nilai OOB optimal diperoleh nilai $B = 200$ dan $m = 2$, sehingga diperoleh confusion matrix sebagai berikut:

Confusion Matrix and Statistics

```
ensing
 227
6147
```

Diperoleh Accuracy, Precision, dan Recall berdasarkan persamaan 2.14, persamaan 2.15, dan persamaan 2.16 sebagai berikut:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} = \frac{14758+6147}{14758+227+830+6147} = \frac{20905}{21962} = 0{,}95$$

$$Precision = \frac{TP}{(TP+FP)} = \frac{14758}{14758+227} = \frac{14758}{14985} = 0{,}98$$

$$Recall = \frac{TP}{(TP+FN)} = \frac{14758}{(14758+830)} = \frac{14758}{15588} = 0{,}94$$

Lampiran 2. Memuat *Packages*, Import Data, Pembagian Data dan Pembentukan Model *Naive Bayes*

```
#instal_packages
install.packages("naivebayes")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("psych")
install.packages("caret")
install.packages("caretEnsemble")
install.packages("Amelia")
install.packages("mice")
install.packages("GGally")
install.packages("rpart")


#panggil_library
library(naivebayes)
library(dplyr)
library(ggplot2)
library(psych)
e)
```

```
library(GGally)
library(rpart)
library(e1071)
#panggil_data
data=read.delim("clipboard")
View(data)
summary(data)
str(data)
datay=as.factor(data$y)
#membagi_data_menjadi_train_dan_test
set.seed(1234)
ind=sample(2,nrow(data),replace=T,prob = c(0.8,0.2))
train=data[ind==1,]
View(train)
test=data[ind==2,]
View(test)
#model
model=naive_bayes(y~.,data=train)
#prediction
p=predict(model,train,type = 'prob')
p
head(cbind(p,train))
p1=predict(model,train)
p1
)
n(tab1)
t)
```

```
p2
tab2=table(p2,test$y)
tab2
1-sum(diag(tab2))/sum(tab2)
#cara lain1
sampel2=sample(1:nrow(data),0.90*nrow(data),replace=TRUE)
training=data.frame(data)[sampel2,]
testing=data.frame(data)[-sampel2,]
modelnb=naiveBayes(y~.,data = training)
modelnb
prediksi2=predict(modelnb,testing)
prediksi2
hasil2=confusionMatrix(table(prediksi2,testing$y))
```

Lampiran 3. Contoh perhitungan penurunan nilai impuritas menqggunakan indeks gini dalam pemilihan pemilah

pemilihan pemilah utama dalam pembentukan pohon klasifikasi menggunakan penurunan nilai impuritas. Variable predictor yang akan dihitung adalah tingkat lama waktu dalam forum (X6). Langkah awal yaitu menghitung indeks gini untuk simpul utama menggunakan persamaan (2.2):

Y (status dimensi)	Jumlah Amatan
Intuitive	36166
Sensing	51679
Jumlah	87845

$$i(t) = 1 - \sum_{j=1}^{k} p^2(j|t)$$

$$\left(\frac{36166}{87845}\right)^2 - \left(\frac{51679}{87845}\right)^2 = 1 - (0{,}169) - (0{,}346) = 0{,}484$$

gini simpul kiri $i(t_l)$ dan simpul kanan $i(t_r)$ dari penyekat X6 naan (2.11):

atus Dimensi	Tingkat lama waktu dalam forum

	0	1	2	3
Intuitive	23124	8186	4852	4
Sensing	20088	17943	13629	19
jumlah	43212	26129	18481	23

$$i(t_0) = 1 - \left(\frac{23124}{43212}\right)^2 - \left(\frac{20088}{43212}\right)^2 = 1 - (0{,}286) - (0{,}216) = 0{,}498$$

$$i(t_1) = 1 - \left(\frac{8186}{26129}\right)^2 - \left(\frac{17943}{26129}\right)^2 = 1 - (0{,}098) - (0{,}471) = 0{,}431$$

$$i(t_2) = 1 - \left(\frac{4852}{18481}\right)^2 - \left(\frac{13629}{18481}\right)^2 = 1 - (0{,}068) - (0{,}543) = 0{,}389$$

$$i(t_3) = 1 - \left(\frac{4}{23}\right)^2 - \left(\frac{19}{23}\right)^2 = 1 - (0{,}030) - (0{,}682) = 0{,}288$$

Hitung nilai impuritas menggunakan persamaan (2.3):

$$\varphi(s,t) = i(t) - p_l i(t_l) - p_r i(t_r)$$

$$\varphi(s,t) = i(t) - p_0 i(t_0) - p_1 i(t_1) - p_2 i(t_2) - p_3 i(t_3)$$

$$\varphi(s,t) = 0{,}484 - \left(\frac{43212}{87845}\right)(0{,}498) - \left(\frac{26129}{87845}\right)(0{,}431) - \left(\frac{18481}{87845}\right)(0{,}389) - \left(\frac{23}{87845}\right)(0{,}288)$$

$$= 0{,}484 - 0{,}244 - 0{,}127 - 0{,}081 - 7{,}5231e05$$

$$= 3{,}0140e02$$

Lakukan perhitungan pada setiap kemungkjnan pemilah dari semua prediktor. Prediktor dengan nilai $\varphi(s,t)$ tertinggi akan dijadikan pemilah utama menggunakan persamaan (2.4).

Lampiran 4. Contoh pembentukan model Naive Bayes

Pembentukan model Naive Bayes menggunakan pendekatan Densitas gausian dengan menentukan nilai mean dan standar deviasi terlebih dahulu

ngan mean standar deviasi kelas intuitive dan sensing

Intuitive		*Sensing*	
lean	Standar Deviasi	Mean	Standar Deviasi

X_1	12,657	18,164	37,352	9,350
X_2	16,363	17,064	14,074	30,661
X_3	78,883	91,217	183,805	83,098
X_4	64747,290	86898,670	153311,120	105447,090
X_5	297,836	705,167	832,064	1139,112

Mean Kelas Intuitive $\left(\boldsymbol{\mu} = \frac{\sum_{i=1}^{n} X_i}{n}\right)$

$$\mu_{x1} = \frac{457781}{36166} = 12{,}65777249$$

$$\mu_{x2} = \frac{591800}{36166} = 16{,}36343527$$

$$\mu_{x3} = \frac{2852915}{36166} = 78{,}88389648$$

$$\mu_{x4} = \frac{2341650669}{36166} = 64747{,}29495$$

$$\mu_{x5} = \frac{10771565}{36166} = 297{,}8367804$$

Standar Deviasi kelas Intuitive $\boldsymbol{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}}$

$$\sigma_{x1} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 18{,}16420434$$

$$\sigma_{x2} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 17{,}06465921$$

$$\sigma_{x3} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 91{,}21584788$$

$$\sigma_{x4} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 86897{,}47045$$

5,1576839

$\left(\boldsymbol{\mu} = \frac{\sum_{i=1}^{n} X_i}{n}\right)$

3520579

$$\mu_{x2} = \frac{727351}{51679} = 14{,}07440159$$

$$\mu_{x3} = \frac{9498904}{51679} = 183{,}8058786$$

$$\mu_{x4} = \frac{7922965262}{51679} = 153311{,}1179$$

$$\mu_{x5} = \frac{43000278}{51679} = 832{,}0648232$$

Standar Deviasi kelas Sensing $\boldsymbol{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}}$

$$\sigma_{x1} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 9{,}350851869$$

$$\sigma_{x2} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 30{,}66120137$$

$$\sigma_{x3} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 83{,}09733696$$

$$\sigma_{x4} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 105446{,}0731$$

$$\sigma_{x5} = \sqrt{\frac{\sum_{i=1}^{n}(X_i-\mu)^2}{n}} = 1139{,}101693$$

Lampiran 6. Perhitungan probabilitas kelas intuitive dan sensing

Forum Post	Jumlah Kejadian		Probabilitas	
	Intuitive	*Sensing*	*Intuitive*	*Sensing*
0	23124	20088	0,639	0,388
1	8186	17943	0,226	0,347
2	4852	13629	0,134	0,263
3	4	19	0,000	0,000
	36166	51679	1	1

ngan Densitas Gausian Kelas Intuitive dan Sensing

$$g(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Densitas Gausian pada kelas Intuitive

Atribut	Uji	*Intuitive*	*Sensing*
Jumlah revisi X_1	0	0.017	0,000
Jumlah kunjungan tugas X_2	20	0,022	0,012
Jumlah konten yang dikunjungi X_3	24	0,003	0,000
Lama waktu pada konten X_4	46045	4,485	0,000
Forum diskusi yang dikunjungi X_5	1	0,000	0,000