DAFTAR PUSTAKA

[1] Kumar, Vinay; Fausto, Nelson; Abbas, Abul K.; Cotran, Ramzi S. ; Robbins, Stanley L. (2005). *Robbins and Cotran Pathologic Basis of Disease* (7th ed.). Philadelphia, Pa.: Saunders. hlmn. 1194–1195.

[2] Shoback, edited by David G. Gardner, Dolores (2011).*Greenspan's basic & clinical endocrinology* (9th ed.). New York: McGraw-Hill Medical. hlmn. Chapter 17.

[3] Rusydina, A. W. (2016). Perbandingan Metode Feature Selection Pada High Dimensional Data dan Klasifikasi Menggunakan Support Vector Machine.Surabaya: Institut Teknologi Sepuluh Nopember.

[4] Mukesh Kumar, Sandeep Singh, and Santanu Kumar Rath(2015). Classification of microarray data using functional link neural network. *Procedia Computer Science*, 57:727–737

[5] R M Parry (2010). ***k***-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. *The Pharmacogenomics Journal*

[6] Xindong Zhang, Lin Gao, Zhi-Ping Liu dan Luonan (2015).*Identifying Module Biomarker in Type 2 Diabetes Mellitus By Discriminative Area of Functional Activity*. *BMC bioinformatics, 16(1), 92*.

[7] Stekel, D., 2003, *Microarray Bioinformatics*, Cambridge University Press, New York.

[8] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S. dan Golub, T.R., (2001), Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences 98(26)*, 15149-15154.

[9] Ridwan, R. (2015). *Mengenal alat analisa molekuler DNA sequencing.* Biotrends 3.1, 22-25.

[10] Razavi, A. E. (2012). Dna Microarray, *Isfahan University Of Medical Science*, School Of Pharmacy Department Of Clinical Biochemistry.

[11] Bolstad M, B. I. (2003*). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias*. *Bioinformatics*, 186-187.

[12] Haykin, S. (2009). *Neural Networks and Learning Machine* (3rd ed.). (M. J. Horton, A. Dworkin, D. Mars, W. Opaluch, S. Disanno, & G. Dulles, Eds.) New Jersey: Pearson Education.

itaningrum, D. (2006). *Pengantar Jaringan Syaraf Tiruan.* Yogyakarta:

Penerbit Andi.

[14] Riadi, M. (2016). *Kajian Pustaka*. Retrieved May 2, 2018, from Jaringan Saraf Tiruan (JST):n https://www.kajianpustaka.com/2016/11/ jaringan-saraf-tiruan-jst.html

[15] Siang, J. (2004). *Aplikasi Jaringan Syaraf Tiruan dan Pemrograman Menggunakan MATLAB.* Yogyakarta: Penerbit Andi.

[16] Ramadhani, P. W. (2017). *Deteksi Kanker berdasarkan Klasifikasi Data Microarray menggunakan Functional Link Neural Network dengan Seleksi Fitur Genetic Algorithm. Indonesian Journal on Computing (Indo-JC)*, 11-22.

[17] Ramadhani, P. W. (2017). *Deteksi Kanker berdasarkan Klasifikasi Data Microarray menggunakan Functional Link Neural Network dengan Seleksi Fitur Genetic Algorithm. Indonesian Journal on Computing (Indo-JC)*, 11-22.

[18] Antoni Wibowo, (2017). *10-Fold Cross Validation.* https : //mti.binus.ac.id/ 2017/11/24/ 10-fold-cross-validation/

[19] Florin Gonurescu (2011). *Data Mining concept, models, and technique.* Springer.

[20] Henny Leidiyana (2013). *Penerapan algoritma k-nearest neighbor untuk penentuan resiko kredit kepemilikan kendaraan bemotor* . STIMIK Nusa Mandiri

[21] Obuchowski, N. (2003). Receiver operating characteristic curves and their use radiology. *Radiology*, 229: 3-8.

[22] Park, S. H., Goo, J. M., & Jo, C. H. (2004). Receiver operating characteristic (ROC) curve: practical review for radiologists. *Korean Journal Radiology*, 5(1), 11-18.

[23] Brefeld, U., & Scheffer, T. (2005.. *AUC maximizing support vector learning*. In Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning.

LAMPIRAN

Lampiran 1 ambil package

```
library("GEOquery")
library(preprocessCore)
library(neuralnet)
library(DMwR)
library(dplyr)
library(caret)
library(R.utils)
library(pROC)
library("reshape2")
library("ggplot2")
```

Lampiran 2 Proses ambil data

```
#ambil data
data <- getGEO(file="GSE18732_series_matrix.txt")
load(file="data.Rdata")
```

Lampiran 3 Uji normalitas

```
#uji normalitas
tes_data <- t(exprs(data))


tes <- apply(tes_data,2,shapiro.test)
tes <- data.frame(matrix(unlist(tes),nrow=25770,byrow = T))
tes <- as.numeric(as.character(tes$X2))


norm_data <-t(exprs(data))
featureName <- dimnames(norm_data)
featureName<-featureName[[2]]
```

Lampiran 4 proses standarisasi Min-Max

```
for(i in 1:25770){
                 n(norm_data[,i])
                 x(norm_data[,i])
```

```
  for(j in 1:118){
    norm_data[j,i] = (norm_data[j,i]-min)/(max-min)
  }
}
```

Lampiran 5 Normalisasi quantile

```
norm_data <- normalize.quantiles(norm_data)
colnames(norm_data)<- featureName

tes2 <- apply(norm_data,2,shapiro.test)
tes2 <- data.frame(matrix(unlist(tes2),nrow=25770,byrow = T))
tes2 <- as.numeric(as.character(tes2$X2))
```

Lampiran 6 Seleksi Fitur

```
#seleksi fitur
norm_data <- as.data.frame(norm_data)
label <- data$characteristics_ch1.5

ff <- function(x)(t.test(x~label)$p.value)
p.value <- apply(norm_data,2,ff)
ord <- order(p.value)

signifikansi <- p.value[ord[1:50]]

data10 <- as.data.frame(norm_data[,ord[1:10]])
data20 <- as.data.frame(norm_data[,ord[1:20]])
data30 <- as.data.frame(norm_data[,ord[1:30]])
data40 <- as.data.frame(norm_data[,ord[1:40]])
data50 <- as.data.frame(norm_data[,ord[1:50]])
```


proses ekspansi

7 Ekspansi

```
#10 fitur terbaik dengan orde2
#expansi legendre

expansionL10_ord3 <-matrix(0, ncol = 30,nrow = 118)
expansionL10_ord3 <-as.data.frame(expansionL10_ord3)
expansionL20_ord3 <-matrix(0, ncol = 60,nrow = 118)
expansionL20_ord3 <-as.data.frame(expansionL20_ord3)
expansionL30_ord3 <-matrix(0, ncol = 90,nrow = 118)
expansionL30_ord3 <-as.data.frame(expansionL30_ord3)
expansionL40_ord3 <-matrix(0, ncol = 120,nrow = 118)
expansionL40_ord3 <-as.data.frame(expansionL40_ord3)
expansionL50_ord3 <-matrix(0, ncol = 150,nrow = 118)
expansionL50_ord3 <-as.data.frame(expansionL50_ord3)

for(i in 1:10){
  for(j in 1:118 ){
    expansionL10_ord3[j,i]=data10[j,i]
    expansionL10_ord3[j,i+10]=1/2*((3*data10[j,i]^2)-1)
    expansionL10_ord3[j,i+20]=1/2*((5*data10[j,i]^3)-3*data10[j,i])

  }
}

for(i in 1:20){
  for(j in 1:118 ){
    expansionL20_ord3[j,i]=data20[j,i]
    expansionL20_ord3[j,i+20]=1/2*((3*data20[j,i]^2)-1)
    expansionL20_ord3[j,i+40]=1/2*((5*data20[j,i]^3)-3*data20[j,i])
  }
}

:30){
```

```
  for(j in 1:118 ){
    expansionL30_ord3[j,i]=data30[j,i]
    expansionL30_ord3[j,i+30]=1/2*((3*data30[j,i]^2)-1)
    expansionL30_ord3[j,i+60]=1/2*((5*data30[j,i]^3)-3*data30[j,i])
  }
}

for(i in 1:40){
  for(j in 1:118 ){
    expansionL40_ord3[j,i]=data40[j,i]
    expansionL40_ord3[j,i+40]=1/2*((3*data40[j,i]^2)-1)
    expansionL40_ord3[j,i+80]=1/2*((5*data40[j,i]^3)-3*data40[j,i])
  }
}
for(i in 1:50){
  for(j in 1:118 ){
    expansionL50_ord3[j,i]=data50[j,i]
    expansionL50_ord3[j,i+50]=1/2*((3*data50[j,i]^2)-1)
    expansionL50_ord3[j,i+100]=1/2*((5*data50[j,i]^3)-3*data50[j,i])
  }
}

expansionL10_ord3$y <- as.numeric(label)
expansionL10_ord3$y <- as.numeric(expansionL10_ord3$y==2)
expansionL20_ord3$y <- as.numeric(label)
expansionL20_ord3$y <- as.numeric(expansionL20_ord3$y==2)
expansionL30_ord3$y <- as.numeric(label)
expansionL30_ord3$y <- as.numeric(expansionL30_ord3$y==2)
expansionL40_ord3$y <- as.numeric(label)
expansionL40_ord3$y <- as.numeric(expansionL40_ord3$y==2)
50_ord3$y <- as.numeric(label)
50_ord3$y <- as.numeric(expansionL50_ord3$y==2)
```

Lampiran 8 Klasifikasi data 70% data latih dengan 30% data uji

```
best.train10_ord3 <- NULL

best.acc10_ord3 <- 0

acc <- NULL

pb <- txtProgressBar(min = 0, max = 1000, style = 3)


for(i in 1:1000){

  Sys.sleep(0.1)

  train.proportion <- 0.70

  train.index <- sample(x = 1:nrow(expansionL10_ord3),

                        size = floor(train.proportion *
nrow(expansionL10_ord3)),

                        replace = F)

  expansion.train.data <- expansionL10_ord3[train.index, ]

  expansion.test.data <- expansionL10_ord3[-train.index, ]

  expansion.test.labels <- expansion.test.data$y

  expansion.test.data <- subset(expansion.test.data, select = -c(y))


  formula <- sprintf("%s%s", "y ~ ", paste("V", 1:30, collapse = " + ",
sep = ""))


  expansion.net <- neuralnet(formula, data = expansion.train.data,

                             linear.output = F, rep = 1,

                             err.fct = "sse", act.fct = 'logistic',
threshold = 0.1,algorithm = 'backprop',

                             learningrate = 0.1, hidden=(0))


  net.predictions <- neuralnet::compute(expansion.net,
expansion.test.data)$net.result

  thresholded.net.predictions <- ifelse(net.predictions > 0.5, 1, 0)

  ect <- sum(as.numeric(thresholded.net.predictions ==
test.labels))
```

```
  acc[i] <- num.correct / length(expansion.test.labels)

  if(acc[i]>best.acc10_ord3){

    best.acc10_ord3<- acc[i]

    best.train10_ord3<- NULL

    best.train10_ord3 <- expansion.train.data

    best.test10_ord3 <- expansion.test.labels

    matrixConfussion10_ord3 <-
confusionMatrix(as.factor(expansion.test.labels),as.factor(thresholded.ne
t.predictions))

    neuraModel10_ord3 <- expansion.net

    auc10_ord3 <-
multiclass.roc(expansion.test.labels,thresholded.net.predictions)

  }

  setTxtProgressBar(pb,i)

}

close(pb)
```

Lampiran 9 proses klasifikasi 80% data latih dengan 20% data uji

```
best.train10_ord3_80 <- NULL

best.acc10_ord3_80 <- 0

acc <- NULL

pb <- txtProgressBar(min = 0, max = 1000, style = 3)


for(i in 1:1000){

  Sys.sleep(0.1)

  train.proportion <- 0.80

  train.index <- sample(x = 1:nrow(expansionL10_ord3),

                        size = floor(train.proportion *
nrow(expansionL10_ord3)),

                        replace = F)

  expansion.train.data <- expansionL10_ord3[train.index, ]

  expansion.test.data <- expansionL10_ord3[-train.index, ]

  n.test.labels <- expansion.test.data$y

  n.test.data <- subset(expansion.test.data, select = -c(y))
```

```
  formula <- sprintf("%s%s", "y ~ ", paste("V", 1:30, collapse = " + ",
sep = ""))


  expansion.net <- neuralnet(formula, data = expansion.train.data,

                             linear.output = F, rep = 1,

                             err.fct = "sse", act.fct = 'logistic',
threshold = 0.1,algorithm = 'backprop',

                             learningrate = 0.1, hidden=(0))


  net.predictions <- neuralnet::compute(expansion.net,
expansion.test.data)$net.result

  thresholded.net.predictions <- ifelse(net.predictions > 0.5, 1, 0)

  num.correct <- sum(as.numeric(thresholded.net.predictions ==
expansion.test.labels))


  acc[i] <- num.correct / length(expansion.test.labels)

  if(acc[i]>best.acc10_ord3_80){

    best.acc10_ord3_80<- acc[i]

    best.train10_ord3_80<- NULL

    best.train10_ord3_80 <- expansion.train.data

    best.test10_ord3_80 <- expansion.test.labels

    matrixConfussion10_ord3_80 <-
confusionMatrix(as.factor(expansion.test.labels),as.factor(thresholded.ne
t.predictions))

    neuraModel10_ord3_80 <- expansion.net

    auc10_ord3_80 <-
multiclass.roc(expansion.test.labels,thresholded.net.predictions)

  }

  setTxtProgressBar(pb,i)

}

close(pb)
```

10 Plot grafik hasil klasifikasi

```
result_tabel <-
rbind(best.acc10_ord3,best.acc20_ord3,best.acc30_ord3,best.acc40_ord3,bes
t.acc50_ord3)

result_tabel2 <-
rbind(best.acc10_ord3_80,best.acc20_ord3_80,best.acc30_ord3_80,best.acc40
_ord3_80,best.acc50_ord3_80)

result_tabel3 <- rbind(result_tabel,result_tabel2)

result_index <- c("10","20","30","40","50","10","20","30","40","50")

group_plot <- rep(c("70% training, 30% testing","80% training, 20%
testing"),each=5)

df <- data.frame(Banyak_fitur = result_index, accuracy = result_tabel3,
Jenis = group_plot)

p<-ggplot(data=df, aes(x=Banyak_fitur, y=accuracy, group=Jenis))
+geom_line(aes(color=Jenis))+ geom_point(aes(color=Jenis))

  +ylim(min,max)

plot(p)
```

Lampiran 11 Cross Validasi

```
  acc.cv_10 <- NULL

  foldL10 <- createFolds(expansionL10_ord3$y,10, list = FALSE)

  formula10 <- sprintf("%s%s", "y ~ ", paste("V", 1:30, collapse = " + ",
sep = ""))

  pb <- txtProgressBar(min = 0, max = 10, style = 3)


  for(i in 1:10){

    index <- which(foldL10==i)

    train.cv <- expansionL10_ord3[-index,]

    tes.cv <- expansionL10_ord3[index,]

    tes.label.cv <- tes.cv$y

    tes.cv  <- subset(tes.cv, select=-c(y))

    expansion.net <- neuralnet(formula10, data = train.cv,

                               linear.output = F, rep = 1,

                               err.fct = "sse", act.fct = 'logistic',
threshold = 0.1,algorithm = 'backprop',

                               learningrate = 0.1,hidden = 0)

          edictions <- neuralnet::compute(expansion.net,
          t.result
```

```
    thresholded.net.predictions <- ifelse(net.predictions > 0.5, 1, 0)

    num.correct <- sum(as.numeric(thresholded.net.predictions ==
tes.label.cv))

    acc.cv_10[i] <- num.correct / length(tes.label.cv)

    setTxtProgressBar(pb,i)

  }


  mean.acc.cv_10 <- mean(acc.cv_10)
```