**PEMBERSIHAN DATA**

**Library(multiUS)**

iris.mis1 = KNNimp(data=iris.mis1, k=10) # *knn imputation*

**Library(mice)**  # *Impute Value*

init = mice(dat, maxit=0)

meth = init$method

predM = init$predictorMatrix

meth[c('Age')] = "pmm"

ImputedData = mice(dat, method=meth, predictorMatrix = predM)

CompletedData = complete(ImputedData)

outlier\_ozone = boxplot.stats(ozone\_reading)$out

out\_ind = which(ozone\_reading%in%outlier\_ozone)

ozone3[c(out\_ind),] # Outlier identification

model.Reg = lm(ozone\_reading~.,data = ozone3)

cooksd = cooks.distance(model.Reg)

plot(cooksd, main="Data Pencil Berdasarkan Jarak Cook")

min\_cook = 4\*mean(cooksd)

abline(h=(min\_cook), col='red', lty=2)

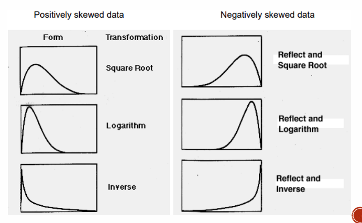
text(x=1:length(cooksd), y=cooksd,

labels = ifelse(cooksd>min\_cook, names(cooksd), ""), col='blue')

outlier\_cook = as.numeric(names(cooksd)[cooksd>min\_cook])

ozone3[outlier\_cook,] # Cooks Outlier

**PENJELMAAN DATA**

****

Sqrt : sqr -> log:exp -> inv:inv

k = max(pressure\_height.hPA) +1 *# Dapatkan pusingan data*

p = k-pressure\_height.hPA # *Pusing balik*

pH2 = sqrt(p) *# jelmakan data*

n = length(pH2) # *Ujian Kolmogorov-Smirnov*

x = rnorm(n)

ks.test(pH2, x) *# bandingkan data dengan data taburan normal*

out = boxcoxnc(data2, method='mle',lambda = seq(-2,2,0.0001))

y = (data2**ˆ**(**-**0.0474)**-**1)**/**(**-**0.0474) # *Boxcox Transformation*

data.Asal = (y**\***(**-**0.0474)**+**1)**ˆ**(1**/**0.0474) # *Revert transformation*

**PCA**

e = eigen(R) # *Nilai Eigen & Vektor*

ev = e$values

evr = e$vectors

Prop.var = ev/length(ev)

cumsum(Prop.var) # *Proportion variance*

y = zdata**%\*%**evr # zdata = scaled data, change to PCA components

**corrplot**(**cor**(z\_skor), order='hclust')

**ANALISIS FAKTOR**

**Library(psych)**

Screeplot()

FA = factanal(z\_skor, factors=2, scores='regression', rotation='varimax')

head(FA.skor$scores, 10) # *To get the FA scores*

**PERLOMBONGAN ATURAN SEKUTUAN**

**Library(arules), library(arulesViz)**

itemFrequencyPlot(tdata, topN=10,main='10 Item paling kerap dibeli')

Aturan.S2 = apriori(data, parameter=list(supp=0.1, conf=0.6), appearance = list(default='lhs',rhs='Survived=Yes')).

Aturan.S2 = sort(Aturan.S1, by='lift', decreasing = T)

plot(Aturan.S1, method='graph')

plot(Aturan.S1, method='paracoord', control=list(reorder=T))

.

**PERLOMBONGAN SIRI MASA**

**Library(forecast)**

**?strptime**

X2 = ts(data1, start=c(2020,3), frequency=4)

daily\_index = seq.Date(from=as.Date('2016-01-01'), to=as.Date('2018-12-31'), by='day')

US\_format\_new = as.Date(dates\_df$US\_format, format = "%m/%d/%Y")

ts.plot(usgas)

acf(x)

pacf(x)

model3 = **ctree**(pattern100**~**.,newdata) # *Klasifikasi Siri Masa*

matriks\_konfusi = table(Predicted = predict(model3, newdata), Actual = newdata$pattern100)

precision\_model = sum(diag(matriks\_konfusi))/sum(matriks\_konfusi)

**PERLOMBONGAN TEKS**

***Pembersihan Teks***

Teks <- = readLines("G:/My Drive/…”)

docs = Corpus(VectorSource(text))

toSpace = content\_transformer(function(x, pattern)

gsub(pattern, "",x)) # *(fx) Aksara khas -> whitespace*

docs2 = tm\_map(docs, toSpace, "!") # *Gantikan simbol*

docs5 = tm\_map(docs4, content\_transformer(tolower)) # to\_lower

docs6 = tm\_map(docs5, removeNumbers)

docs7 = tm\_map(docs6, removeWords, stopwords("english"))

docs8 = tm\_map(docs7, removePunctuation) # *rm punctuation*

docs9 = tm\_map(docs8, stripWhitespace)

***Text Stemming (Pembendungan Teks)***

docs10 = tm\_map(docs9, stemDocument) # *Stemming*

***Matriks Sebutan***

dtm = TermDocumentMatrix(docs10)

m = as.matrix(dtm) # *Matriks sebutan*

v = sort(rowSums(m), decreasing=T)

d = data.frame(word=names(v), freq=(v)) # perkataan paling kerap

wordcloud(words=d$word, freq=d$freq, min.freq=2, max.words =

150, random.order=F, colors = brewer.pal(8, "Dark2"))

# *Awan Perkataan*

***Perkaitan Perkataan (Word Association)***

findAssocs(dtm, terms='freedom', corlimit=0.3)$freedom

findAssocs(dtm, terms=findFreqTerms(dtm,lowfreq=10), corlimit=0.5)

***Analisis Sentimen***

**Library(sentimentr)**

Sentiment(x)

sentiment\_text = get\_sentiment(text, method='syuzhet')

hist(sentiment\_text) # *Analisis Sentimen*

***Klasifikasi Emosi***

d2 = get\_nrc\_sentiment(text)

td = data.frame(t(d2)) # *Klasifikasi emosi*

td\_new = data.frame(rowSums(td))

names(td\_new)[1] = 'Count'

td\_new = cbind("sentiment" = rownames(td\_new), td\_new)

rownames(td\_new) = NULL

qplot(sentiment, weight=Count, data=td\_new,

geom='bar', fill=sentiment, ylab='Count') + ggtitle("Sentiment Score")

# *Pengvisualan*

**PERLOMBONGAN DATA JUJUKAN**

mvad.labels = **c**('employment','further education','higher education','joblessness','school','training')

mvad.scode = c('EM','FE','HE','JL','SC','TR')

mvad.seq = seqdef(mvad, 15:86, states=mvad.scode, labels = mvad.labels, xtstep=6)

***Penunjuk Ringkas Statistik***

seqmeant(mvad.seq) # *mean of sequence*

by(mvad.seq,mvad$male,seqmeant) # *mean by group*

seqmtplot(mvad.seq, group = mvad$male, main='Lelaki') # *plot mean by group*

head(seqtransn(mvad.seq),10) # *bilangan transisi*

mvad.trate = seqtrate(mvad.seq) *# kadar peralihan*

***Pengvisualan***

seqiplot(mvad.seq, main='Plot indeks jujukan',idxs=1:20)

seqfplot(mvad.seq, main=’Plot jujukan kekerapan’,

idxs=1:20)

seqdplot(mvad.seq, border=NA,main='plot taburan keadaan')

seqmsplot(mvad.seq) # *Plot keadaan modal*

***Indeks Entropi***

seqHtplot(mvad.seq, main='Entropi Rentas Lintang')

***Data Jujukan Peristiwa***

mvad.seqe = seqecreate(mvad.seq) # *Jujukan Peristiwa*

subseq = seqefsub(mvad.seqe, pmin.support=0.05)

plot(fsubseq[1:15], col='yellow') # *subjujukan paling kerap*

***Mengkategorikan Corak***

library(cluster)

submat = seqsubm(mvad.seq, method = 'TRATE')

dist.om = seqdist(mvad.seq, method='OM',sm=submat)

clusterward = agnes(dist.om, diss=T, method='ward')

plot(clusterward);abline(h=800, lty='dotted', col='red')

cl.4 = cutree(clusterward, 4) # *4 kluster*

cl4fac = factor(cl.4, labels=paste("Kumpulan", 1**:**4))

seqfplot(mvad.seq, group = cl4fac, main='10 jujukan yang paling kerap berlaku', idxs=1**:**10)

seqHtplot(mvad.seq,group=cl4fac, main='Entropi Rentas Lintang')

**PERLOMBONGAN DATA GRAF**

***Jenis Data Graf***

**Library(igraph)**

g <- graph\_from\_literal(1-2,1-3,1-7,3-4,2-3,2-4,3-5,4-5,4-6,4-7,5-6,5-8,6-7,7-8) # *Graf tidak berarah*

V(g)$name <- c("Adam", "Judy", "Bobby", "Sam", "Frank","Tom", "Jerry", "Jay")

dg<- graph\_from\_literal(KL-+CHINA,KL-+London,CHINA++London)

t # *Graf berarah*

ig <- graph\_from\_adjacency\_matrix(m, weighted=T)

plot(ig, edge.label=E(ig)$weight) # *graf berwajaran*

tr <- make\_tree(40,

children=3, mode='undirected') # *Graf pokok*

gb <- sample\_bipartite(10,5,p=0.4)

colo<- c('blue','red')

shape <- c('circle','square')

plot(gb, vertex.color=colo[as.numeric(V(gb)$type)+1], vertex.shape=shape[as.numeric(V(gb)$type)+1]) # *Graf bipartit*

***Hypergraph***

**Library(HyperG)**

h <- hypergraph\_from\_edgelist(list(1:2, 2:5,3:7, c(1,3,5,7,9)))

***Perwakilan Data Graf***

Adj.list1 <- as\_adj\_list(g) # *Senarai Bersebelahan*

Ed.list1<- as.data.frame(as\_edgelist(g)) # *Senarai Sisi*

Adj.M1<- as\_adjacency\_matrix(g) #*Matriks Bersebelahan*

***Teknik Manipulasi Graf***

h <- g-vertices(c("Jerry","Bobby")) # *Keluarkan verteks*

h2 <- graph\_from\_literal("Adam"-"Judy","Adam"-"Tom",

"Judy"-"Aminah", "Aminah"-"Frank") # Jana *Subgraf*

h3 <- union(h2,g) #  *Gabungkan Graf*

***Mengubah Suai Sisi Data***

E(h3)$type<- c("email", "phone", "FB", "email", "class",

"Twitter", "neighbour", "phone", "FB", "email",

"class", "neighbour", "phone", "email", "email",

"FB", "neighbour")

E(h3)$weight<- c(10, 1, 3, 2, 2, 2, 1, 5, 9, 8,1, 6, 2, 9,

3, 10, 7)

edge\_attr(h3) # *Ubah suai sisi data*

plot(h3, vertex.label=V(h3)$name, edge.label=E(h3)$weight)

plot(h3, vertex.label=V(h3)$name, edge.label=E(h3)$type)

***Analisis Prominen Nod***

**Library(statnet), library(devtools), library(UserNetR)**

Name<- Bali%v%"vertex.names" # *Nama nod*

Role<- Bali%v%"role" # *Role nod*

Attr<- Bali%e%"IC" # *Atribut Nod*

deg<- degree(Bali) *# Kepusatan Darjah*

cls<- closeness(Bali) # *Kepusatan Kedekatan*

btw<- betweenness(Bali) # *Kepusatan Antara*

rolecat <- Bali**%v%**"role"

name <- Bali**%v%**"vertex.names"

**library**(RColorBrewer)

**library**(ggplot2)

my\_pal <- brewer.pal(5, "Set2")

gplot(Bali, displaylabels=T,

vertex.col=my\_pal[as.factor(rolecat)])

legend("topright", legend=c("BM","CT","OA","SB","TL"),

col=my\_pal, pch=19, cex=1) # *Plot graf*

centralization(Bali, degree) # *Ukuran pemusatan*

centralization(Bali, closeness)

cpnet <- cutpoints(net, return.indicator = T) # *Titik Potong*

***Analisis Sub Kumpulan***

clique <- cliques(Facebook) # *Clique*

max\_clique <- max\_cliques(Facebook) # *Clique terbesar*

k\_core <- coreness(Facebook) # *K-teras*

plot(Facebook, main='sub=kumpulan mengikut k-teras',

vertex.size=7, vertex.label.cex=0.7) # *Plot teras sama*

Komuniti\_D <- cluster\_louvain(Facebook)

plot(Komuniti\_D, Facebook) # *Pengesanan komuniti*

skor\_modulariti <- modularity(Komuniti\_D)

skor\_modulariti # *Ukuran modulariti*

**PERLOMBONGAN DATA RERUANG**

**library(sp)**

**library(raster)**

***Spatial Points***

lonlat = cbind(longitude, latitude)

crdref = CRS('+proj=longlat +datum=WGS84') # *Takrifkan CRS*

pts =SpatialPoints(lonlat, proj4string=crdref)

df = data.frame(ID = name, precip)

ptsdf = SpatialPointsDataFrame(pts, data=df) # *Masukkan maklumat*

plot(pts, axes=T)

showDefault(ptsdf) # *Lihat detail data*

***Spatial Lines***

lns = spLines(lonlat, crs=crdref)

***Spatial Polygon***

pols = spPolygons(lonlat, crs=crdref)

***Data Raster***

r=raster(ncol=20,nrow=20, xmx=-80,xmn=-150, ymn=20, ymx=60)

x = rexp(ncell(r), rate=1)

values(r) = x

s = stack(r,r2,r3,r4) # *Raster Stack*

**library**(terra)

f = system.file('external/lux.shp', package='raster')

p = shapefile(f)

***Manipulasi Peta***

zon = as(zon, 'SpatialPolygonsDataFrame')

plot(p, axes=T)

plot(zon, add=T, border='blue', lwd=2, lty=2)# *Bahagi peta kepada 4 zon*

zon2 = zon[2,] # *Pilih Zon tertentu*

e = intersect(p, zon2)

plot(e,axes=T) *# Zoom Zon*

e2 = erase(p, zon2) # *Buang Zon*

e3 = extent(5.8,6.3,49.6,50)

pe = crop(p, e3)

plot(p, axes=T)

plot(pe, axes=T, add=T, col='red') # *Takrif lokasi nak pilih*

***Manipulasi Data Raster***

l2 = crop(s,extent(-130,**-**100,30,40)) # *Pangkas data*

m = merge(l2,l3) *# Gabung*

cellStats(s, mean) # *Fungsi Deskriptif*

***Autokorelasi Reruang (Moran-ii)***

**Library(spdep)**

w <- poly2nb(p)

ww <- nb2listw(w)

moran.test(p$AREA, ww) # Nak p >0.05

***Interpolasi Ruang***

**Library(devtools)**

**Library(rspat) install\_github(‘rspatial/rspat’)**

d <- spat\_data('precipitation')

d$prec <- rowSums(d[,6:17])

dsp<- vect(d, c("LONG", "LAT"),

crs="+proj=longlat +datum=WSG84")

cuts <- c(0,200,300,500,1000,3000)

library(ggplot2)

blues <- colorRampPalette(c('yellow','orange','purple','blue','darkblue'))

plot(CA)

plot(dsp, "prec", type="interval", col=blues(10),

breaks=cuts, add=T) # *cipta graf*

v <- **voronoi**(dsp) # *Voronoi*

vca <- crop(v, CA)

plot(vca, 'prec')

***Regressi Setempat***

countries<- spat\_data('counties')

crs(hvect) <- crs(countries)

***Geographically Weighted Regression (GWR)***

countrynames <- unique(hd$NAME)

regfun <- function(x) {

dat <- hd[hd$NAME == x,]

m <- glm(houseValue~income+houseAge+roomhead+bedroomhead+population, data=dat)

coefficients(m)

}

hd2 <- hd[!is.na(hd$NAME),]

countrynames <- unique(hd2$NAME)

res <- sapply(countrynames, regfun)

dotchart(sort(res['income',]), cex=0.65)

resdf<- data.frame(NAME=colnames(res), t(res))

dcounties <- aggregate(countries[,'NAME'], 'NAME')

cnres <- merge(dcounties, resdf, by='NAME')

plot(cnres, 'income')

values(cnres2) <- as.data.frame(scale(as.data.frame(cnres)[-1]))

plot(cnres2, 2:7, plg=list(x='topright'), mar=c(1,1,1,1))