NAFILA HANUM AL HASANIY - 23031554109 DAFANOV DIXIE EINKINDEREN - 23031554201

## 1. Load Data dan Preprocessing

```r
setwd("D:/analisis multivariat_smt4")

dataset <- read.csv("automobile_numeric_data_fix_final.csv")

str(dataset)
```

```
## 'data.frame':    205 obs. of  15 variables:
##  $ price            : num  13495 16500 16500 13950 17450 ...
##  $ highway.mpg      : num  27 27 26 30 22 25 25 25 20 22 ...
##  $ city.mpg         : num  21 21 19 24 18 19 19 19 17 16 ...
##  $ peak.rpm         : num  5000 5000 5000 5500 5500 5500 5500 5500 5500 5500 ...
##  $ horsepower       : num  111 111 154 102 115 110 110 110 140 160 ...
##  $ compression.ratio: num  9 9 9 10 8 8.5 8.5 8.5 8.3 7.4 ...
##  $ stroke           : num  2.68 2.68 3.47 3.4 3.4 3.4 3.4 3.4 3.4 3.4 ...
##  $ bore             : num  3.47 3.47 2.68 3.19 3.19 3.19 3.19 3.19 3.13 3.13 ...
##  $ engine.size      : num  130 130 152 109 136 136 136 136 131 131 ...
##  $ curb.weight      : num  2548 2548 2823 2337 2824 ...
##  $ height           : num  48.8 48.8 52.4 54.3 54.3 53.1 55.7 55.7 55.9 52 ...
##  $ width            : num  64.1 64.1 65.5 66.2 66.4 66.3 71.1 71.1 71.1 67.9 ...
##  $ length           : num  169 169 171 177 177 ...
##  $ wheel.base       : num  88.6 88.6 94.5 99.8 99.4 ...
##  $ normalized.losses: num  115 115 115 164 164 115 158 115 158 115 ...
```

```r
summary(dataset)
```

```
##      price        highway.mpg       city.mpg       peak.rpm      horsepower
##  Min.   : 5118   Min.   :16.00   Min.   :13.0   Min.   :4150   Min.   : 48.0
##  1st Qu.: 7788   1st Qu.:25.00   1st Qu.:19.0   1st Qu.:4800   1st Qu.: 70.0
##  Median :10595   Median :30.00   Median :24.0   Median :5200   Median : 95.0
##  Mean   :12763   Mean   :30.68   Mean   :25.2   Mean   :5126   Mean   :102.9
##  3rd Qu.:16500   3rd Qu.:34.00   3rd Qu.:30.0   3rd Qu.:5500   3rd Qu.:116.0
##  Max.   :29588   Max.   :47.50   Max.   :46.5   Max.   :6550   Max.   :185.0
##  compression.ratio     stroke            bore         engine.size
##  Min.   : 7.400    Min.   :2.660   Min.   :2.540   Min.   : 61.0
##  1st Qu.: 8.600    1st Qu.:3.110   1st Qu.:3.150   1st Qu.: 97.0
##  Median : 9.000    Median :3.290   Median :3.310   Median :120.0
##  Mean   : 9.039    Mean   :3.262   Mean   :3.329   Mean   :124.6
##  3rd Qu.: 9.400    3rd Qu.:3.410   3rd Qu.:3.580   3rd Qu.:141.0
##  Max.   :10.600    Max.   :3.860   Max.   :3.940   Max.   :207.0
##   curb.weight       height          width           length
##  Min.   :1488   Min.   :47.80   Min.   :60.30   Min.   :141.1
##  1st Qu.:2145   1st Qu.:52.00   1st Qu.:64.10   1st Qu.:166.3
##  Median :2414   Median :54.10   Median :65.50   Median :173.2
##  Mean   :2556   Mean   :53.72   Mean   :65.88   Mean   :174.0
##  3rd Qu.:2935   3rd Qu.:55.50   3rd Qu.:66.90   3rd Qu.:183.1
##  Max.   :4066   Max.   :59.80   Max.   :71.10   Max.   :208.1
##    wheel.base     normalized.losses
```

```
##  Min.   : 86.60   Min.   : 65
##  1st Qu.: 94.50   1st Qu.:101
##  Median : 97.00   Median :115
##  Mean   : 98.71   Mean   :120
##  3rd Qu.:102.40   3rd Qu.:137
##  Max.   :114.25   Max.   :191
```

```r
head(dataset)
```

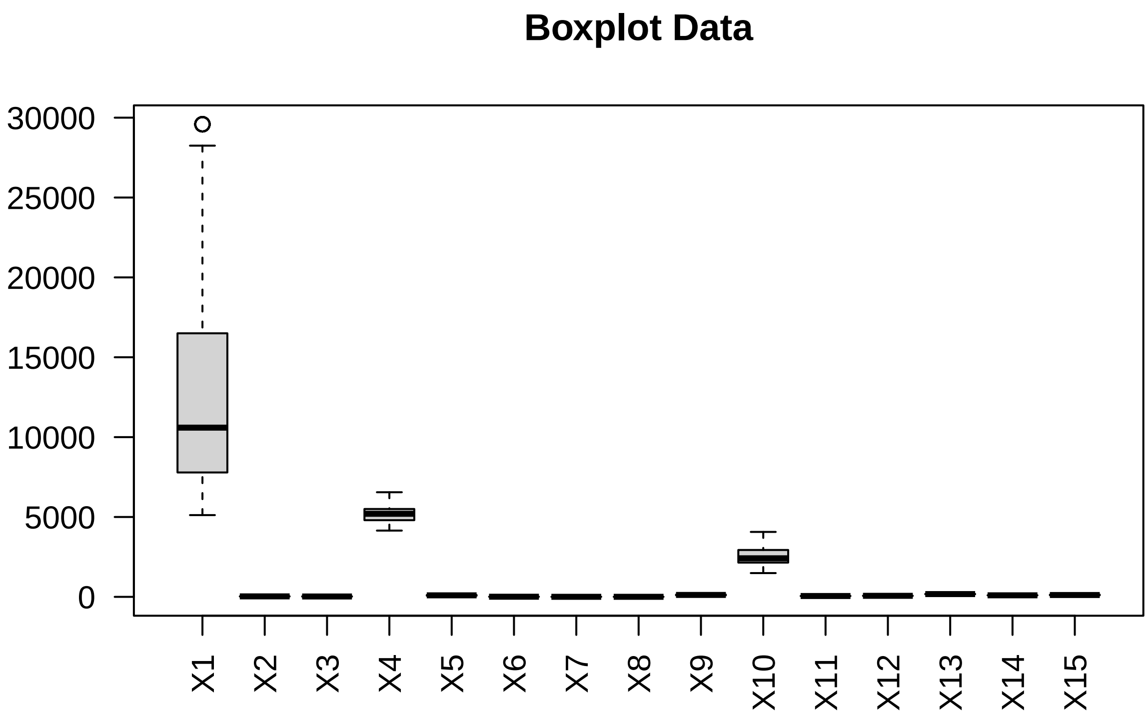
```
##   price highway.mpg city.mpg peak.rpm horsepower compression.ratio stroke bore
## 1 13495          27       21     5000        111              9.0   2.68 3.47
## 2 16500          27       21     5000        111              9.0   2.68 3.47
## 3 16500          26       19     5000        154              9.0   3.47 2.68
## 4 13950          30       24     5500        102             10.0   3.40 3.19
## 5 17450          22       18     5500        115              8.0   3.40 3.19
## 6 15250          25       19     5500        110              8.5   3.40 3.19
##   engine.size curb.weight height width length wheel.base normalized.losses
## 1         130        2548   48.8  64.1  168.8       88.6               115
## 2         130        2548   48.8  64.1  168.8       88.6               115
## 3         152        2823   52.4  65.5  171.2       94.5               115
## 4         109        2337   54.3  66.2  176.6       99.8               164
## 5         136        2824   54.3  66.4  176.6       99.4               164
## 6         136        2507   53.1  66.3  177.3       99.8               115
```

## 2. Penyesuaian Nama Kolom dan Boxplot

```r
original_colnames <- colnames(dataset)

colnames(dataset) <- paste0("X", 1:ncol(dataset))

boxplot(dataset, main = "Boxplot Data", las = 2)
```




### 3. Cek Missing Value

```
sum(is.na(dataset))
```

```
## [1] 0
```

```
p <- ncol(dataset)
```

### 4. Statistika Deskriptif

```
# Mean
means <- apply(dataset, 2, mean, na.rm = TRUE)
means
```

```
##           X1          X2          X3          X4          X5          X6
## 12762.781095   30.680488   25.204878 5125.609756  102.892683    9.039122
##           X7          X8          X9         X10         X11         X12
##     3.262488    3.329366  124.570732 2555.565854   53.724878   65.884390
##          X13         X14         X15
##   174.049268   98.710976  119.990244
```

```r
# Median
medians <- apply(dataset, 2, median, na.rm = TRUE)
medians
```

```
##       X1       X2       X3       X4       X5       X6       X7       X8 
## 10595.00    30.00    24.00  5200.00    95.00     9.00     3.29     3.31 
##       X9      X10      X11      X12      X13      X14      X15 
##   120.00  2414.00    54.10    65.50   173.20    97.00   115.00 
```

```r
# Standard Deviation
sds <- apply(dataset, 2, sd, na.rm = TRUE)
sds
```

```
##           X1           X2           X3           X4           X5           X6 
## 6594.5113426    6.6821470    6.4916595  475.5443155   35.5646084    0.7933561 
##           X7           X8           X9          X10          X11          X12 
##    0.2768807    0.2708575   33.9743433  520.6802035    2.4435220    2.0831138 
##          X13          X14          X15 
##   12.3372885    5.8825860   29.9317836 
```

```r
# Range (Rentang)
mins <- apply(dataset, 2, min, na.rm = TRUE)
maxs <- apply(dataset, 2, max, na.rm = TRUE)
ranges <- maxs - mins
ranges
```

```
##       X1       X2       X3       X4       X5       X6       X7       X8 
## 24469.50    31.50    33.50  2400.00   137.00     3.20     1.20     1.40 
##       X9      X10      X11      X12      X13      X14      X15 
##   146.00  2578.00    12.00    10.80    67.00    27.65   126.00 
```

```r
# Skewness
library(e1071)
skewness_vals <- apply(dataset, 2, skewness, na.rm = TRUE)
skewness_vals
```

```
##          X1          X2          X3          X4          X5          X6 
##  1.22580870  0.34237311  0.59577457  0.04310622  0.80029758  0.03463638 
##          X7          X8          X9         X10         X11         X12 
## -0.38083879  0.02415302  0.89520211  0.67145894  0.06220199  0.76495447 
##         X13         X14         X15 
##  0.15367894  0.91142476  0.58774576 
```

## 5. Uji KMO & Bartlett

```r
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.4.3
```

```
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.4.3

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
r <- cor(dataset)

# KMO Test
KMO(r)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = r)
## Overall MSA =  0.87
## MSA for each item =
##   X1   X2   X3   X4   X5   X6   X7   X8   X9  X10  X11  X12  X13  X14  X15
## 0.94 0.86 0.84 0.56 0.87 0.64 0.43 0.92 0.88 0.93 0.73 0.95 0.92 0.87 0.66
```

```r
dataset <- dataset[, !(colnames(dataset) %in% c("X7"))]

# Cek ulang KMO
KMO(cor(dataset))
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor(dataset))
## Overall MSA =  0.88
## MSA for each item =
##   X1   X2   X3   X4   X5   X6   X8   X9  X10  X11  X12  X13  X14  X15
## 0.96 0.86 0.84 0.56 0.87 0.63 0.95 0.90 0.93 0.74 0.95 0.92 0.87 0.65
```

```r
# Bartlett Test
cor_matrix <- cor(dataset)
cortest.bartlett(cor_matrix, n = nrow(dataset))
```

```
## $chisq
## [1] 3311.784
##
## $p.value
## [1] 0
##
## $df
## [1] 91
```

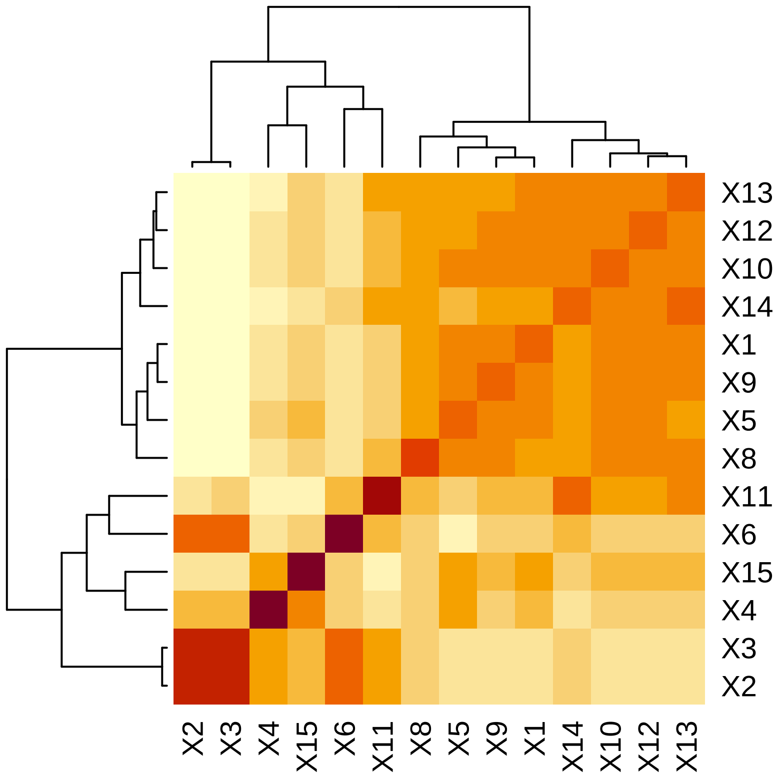
## 6. Standarisasi Data

```
data_scaled <- scale(dataset)
head(data_scaled)
```

```
##             X1         X2         X3         X4          X5          X6
## [1,] 0.1110346 -0.5507942 -0.6477355 -0.2641389  0.22796025 -0.04931197
## [2,] 0.5667166 -0.5507942 -0.6477355 -0.2641389  0.22796025 -0.04931197
## [3,] 0.5667166 -0.7004467 -0.9558231 -0.2641389  1.43702741 -0.04931197
## [4,] 0.1800314 -0.1018367 -0.1856040  0.7872878 -0.02510032  1.21115603
## [5,] 0.7107758 -1.2990567 -1.1098669  0.7872878  0.34043162 -1.30977997
## [6,] 0.3771650 -0.8500992 -0.9558231  0.7872878  0.19984241 -0.67954597
##              X8         X9         X10        X11        X12        X13
## [1,]  0.5192181  0.1598050 -0.01453071 -2.0154834 -0.8565976 -0.4254799
## [2,]  0.5192181  0.1598050 -0.01453071 -2.0154834 -0.8565976 -0.4254799
## [3,] -2.3974442  0.8073524  0.51362457 -0.5422002 -0.1845268 -0.2309477
## [4,] -0.5145356 -0.4583085 -0.41976986  0.2353660  0.1515087  0.2067498
## [5,] -0.5145356  0.3364088  0.51554514  0.2353660  0.2475188  0.2067498
## [6,] -0.5145356  0.3364088 -0.09327386 -0.2557284  0.1995137  0.2634883
##             X14        X15
## [1,] -1.7187978 -0.1667206
## [2,] -1.7187978 -0.1667206
## [3,] -0.7158375 -0.1667206
## [4,]  0.1851268  1.4703352
## [5,]  0.1171295  1.4703352
## [6,]  0.1851268 -0.1667206
```

## 7. Korelasi dan Heatmap

```
correlation_matrix <- cor(data_scaled)
heatmap(correlation_matrix)
```




### 8. Principal Component Analysis (PCA)

```r
pca_result <- prcomp(data_scaled, center = TRUE, scale. = TRUE)

summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.7465 1.5173 1.03919 0.92089 0.82968 0.68307 0.57162
## Proportion of Variance 0.5388 0.1644 0.07714 0.06057 0.04917 0.03333 0.02334
## Cumulative Proportion  0.5388 0.7032 0.78037 0.84095 0.89011 0.92344 0.94678
##                            PC8     PC9    PC10    PC11    PC12   PC13    PC14
## Standard deviation     0.49254 0.38873 0.34548 0.29443 0.25636 0.2337 0.15809
## Proportion of Variance 0.01733 0.01079 0.00853 0.00619 0.00469 0.0039 0.00179
## Cumulative Proportion  0.96411 0.97490 0.98343 0.98962 0.99431 0.9982 1.00000
```

```r
pca_result$rotation
```

```
##             PC1         PC2         PC3         PC4          PC5          PC6
## X1  -0.31858161 -0.03843826 -0.20341903  0.02934944  0.208162585 -0.256993676
## X2   0.32017641  0.20727947 -0.18466137  0.05925938  0.008156224 -0.048129259
## X3   0.31018299  0.25865221 -0.18690861  0.04905976 -0.002701116 -0.097750084
## X4   0.07075063 -0.44192561  0.09497768 -0.48821352  0.610966576  0.204097153
```

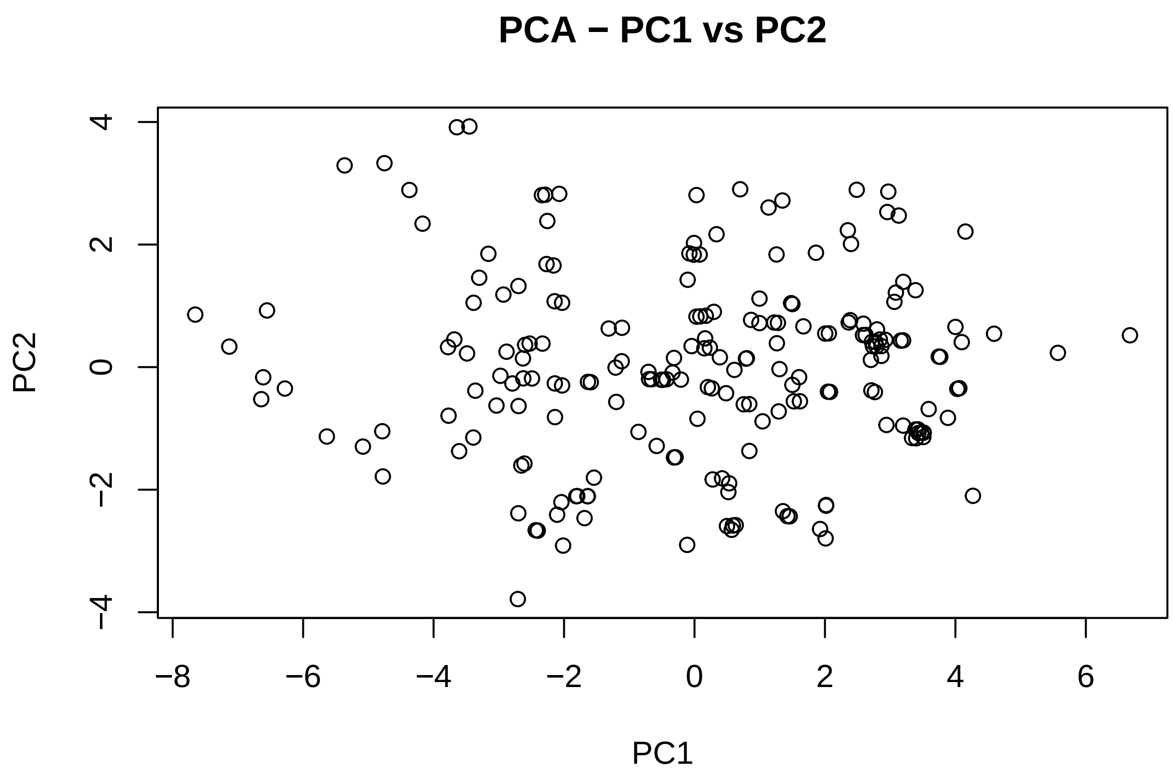
```
## X5  -0.30750867 -0.27077397 -0.01266669  0.14199910  0.182638679 -0.119755678
## X6   0.07950117  0.35313163 -0.65297811 -0.04541031  0.459183508  0.187030359
## X8  -0.26196644  0.04556564  0.02955828  0.40707227  0.002316343  0.827780608
## X9  -0.32018978 -0.03429636 -0.18636955  0.27379632  0.075924617 -0.230531001
## X10 -0.35053204  0.06117560 -0.09402687  0.00313735  0.005772934 -0.111776875
## X11 -0.10394183  0.47346522  0.33273837 -0.40881222  0.038671511  0.123288232
## X12 -0.32138476  0.10139008 -0.13313233 -0.13333961  0.002849588 -0.121024797
## X13 -0.32575359  0.18325847  0.01940883 -0.20630999 -0.059210655  0.060697375
## X14 -0.28066704  0.30518779  0.03189367 -0.31226516 -0.131578003 -0.006103356
## X15 -0.02982592 -0.35660985 -0.52890748 -0.40863121 -0.557658015  0.216049908
##              PC7         PC8         PC9        PC10        PC11         PC12
## X1   0.286631538  0.10054488  0.77906966  0.03780049  0.20113001 -0.058270075
## X2   0.044136450  0.52895930 -0.08058391 -0.04023390  0.25459495  0.046404600
## X3   0.026158829  0.49313895  0.02390819 -0.06862720 -0.03463193  0.122724780
## X4  -0.074108008  0.27641643 -0.04917903 -0.07184518 -0.04295721 -0.185431124
## X5   0.197575162  0.15176543 -0.27704810 -0.07191334  0.01543491  0.769407907
## X6  -0.011756591 -0.39680876 -0.10208747 -0.02362453 -0.07050356  0.093907084
## X8   0.005010222  0.23449514  0.13031323  0.04976821 -0.01808137 -0.021589922
## X9   0.290322733  0.22315482 -0.41704509 -0.09120040 -0.19962496 -0.568186109
## X10 -0.020019982 -0.08096472 -0.11267830 -0.08563404 -0.14373124  0.029768499
## X11  0.607614940  0.01382808 -0.11327623  0.29144803 -0.03680316  0.044291766
## X12 -0.494556134  0.22011566 -0.08916506  0.72521844  0.03382202  0.009653786
## X13 -0.162910283 -0.02242373 -0.19582373 -0.36773926  0.75624158 -0.089939456
## X14 -0.291267159  0.21622282  0.18147152 -0.46311864 -0.50103366  0.097876231
## X15  0.246318283  0.05265168 -0.02690195  0.05127316 -0.01429217  0.016926566
##             PC13         PC14
## X1   0.028458016 -0.010309274
## X2  -0.014379427 -0.676380510
## X3  -0.255374826  0.677755737
## X4  -0.106715308  0.017045967
## X5   0.165498569  0.022021964
## X6   0.110828041 -0.007598504
## X8  -0.044171118  0.016556076
## X9   0.212306871  0.060048966
## X10 -0.875394868 -0.192768936
## X11  0.001185020 -0.003290060
## X12  0.084461267  0.031199429
## X13  0.034967659  0.168331512
## X14  0.246990549 -0.107662412
## X15 -0.001623393  0.016317203
```

```r
head(pca_result$x)
```

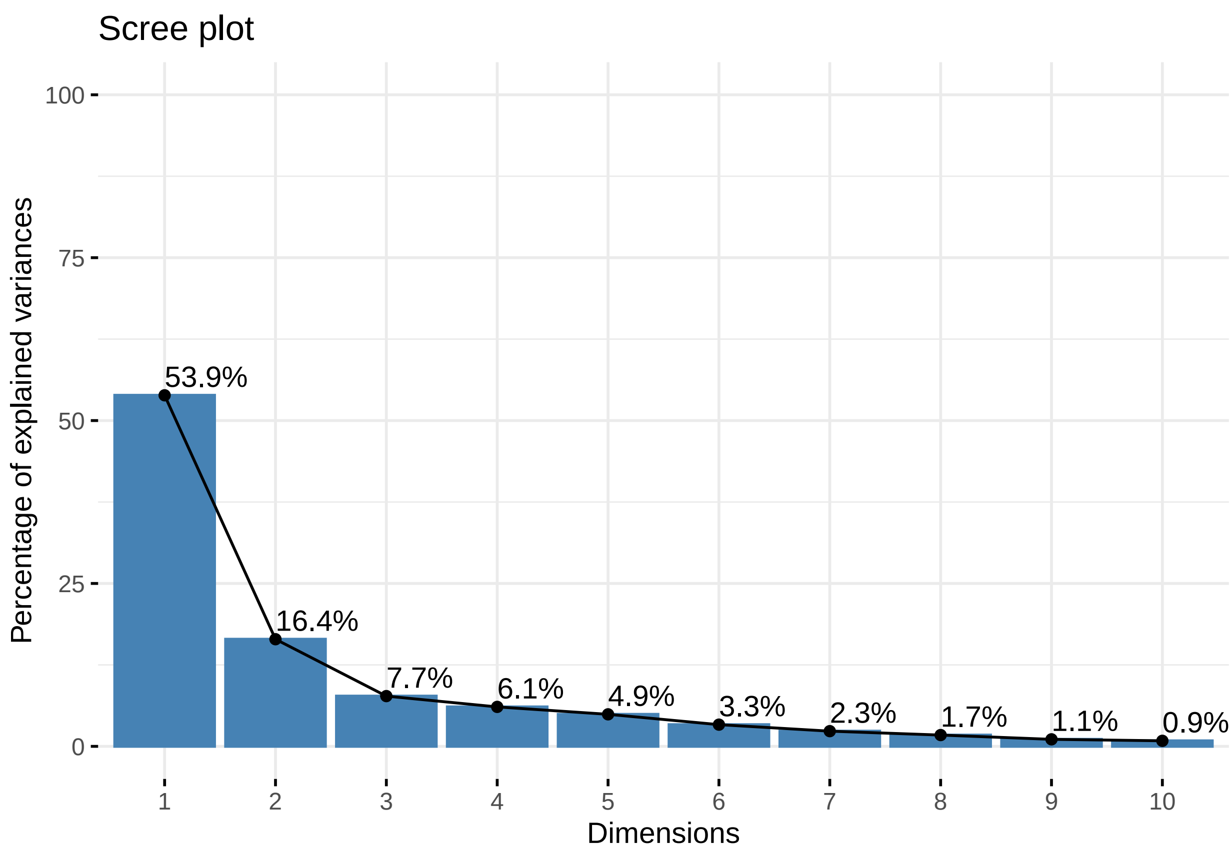
```
##             PC1        PC2        PC3         PC4        PC5         PC6
## [1,]  0.4233329 -1.8152817 -0.3401382  1.98827594  0.1551727  0.16926996
## [2,]  0.2781610 -1.8327973 -0.4328326  2.00164995  0.2500286  0.05216258
## [3,] -0.5795188 -1.2862560 -0.1829764  0.09578083  0.4313238 -2.57199620
## [4,]  0.2045364 -0.3260206 -1.2988251 -1.60674839  0.1842887  0.43340817
## [5,] -1.5414726 -1.8038970  0.3775237 -1.31212396 -0.7283518 -0.36924595
## [6,] -0.8591831 -1.0572349  0.6932924 -0.49506614  0.3471327 -0.52280799
##              PC7        PC8        PC9        PC10        PC11       PC12
## [1,] -0.16707112 -1.0471653  0.1534996 -0.17763505  0.470673516 -0.2245435
## [2,] -0.03645829 -1.0013488  0.5085076 -0.16041004  0.562324838 -0.2510961
## [3,]  0.58956145 -1.2503393 -0.6143147 -0.08887860  0.014219594  0.4975928
```

```
## [4,]  0.18012842 -0.4336133  0.0733847  0.08450312  0.097285725  0.2057588
## [5,]  0.54141798 -0.3054675  0.2596138  0.19797902 -0.141149100 -0.3790776
## [6,] -0.26029874 -0.3378122  0.1156021 -0.11014514 -0.009691767 -0.4353824
##            PC13         PC14
## [1,] -0.2532137  0.044894615
## [2,] -0.2402459  0.040196864
## [3,]  0.1576832 -0.211093270
## [4,]  0.4565793  0.032589155
## [5,] -0.1526949  0.115412927
## [6,]  0.3883536  0.004702654
```

```
plot(pca_result$x[,1], pca_result$x[,2],
     xlab = "PC1", ylab = "PC2",
     main = "PCA - PC1 vs PC2")
```




```
# Scree Plot
library(factoextra)
fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 100))
```

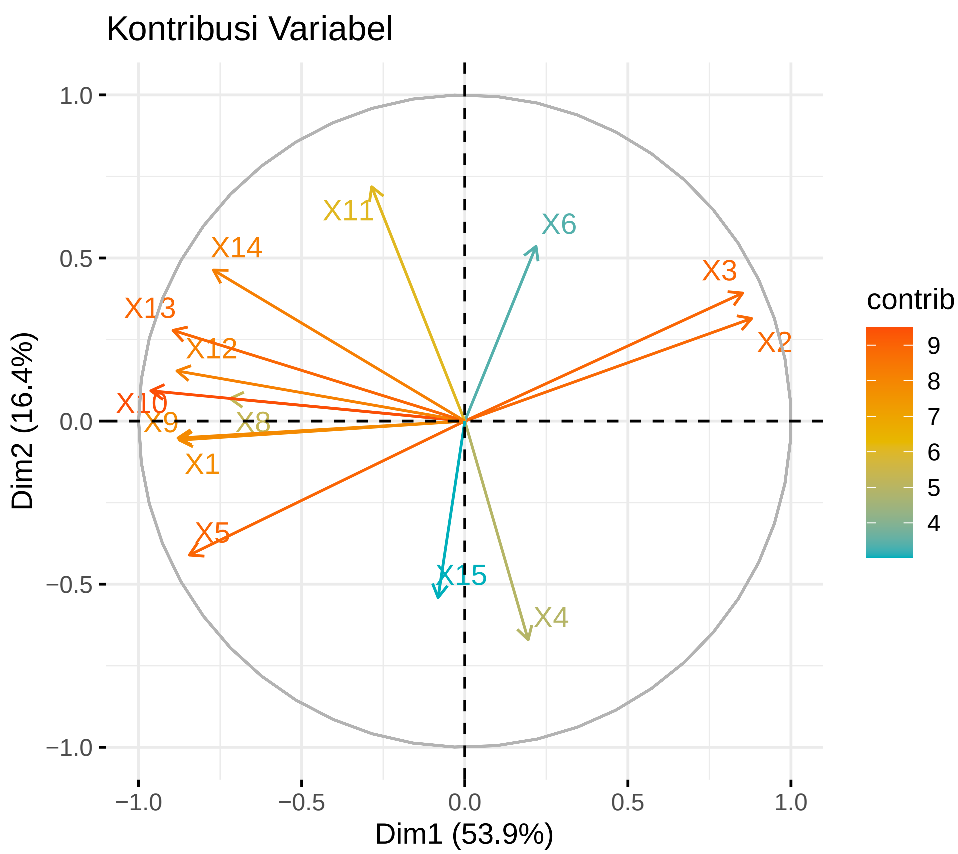



### 9. Korelasi Variabel PCA (Correlation Circle)

```
pca_df <- as.data.frame(pca_result$x)

contrib_circle <- fviz_pca_var(pca_result, col.var = "contrib",
                               gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                               repel = TRUE) +
  ggtitle("Kontribusi Variabel")

contrib_circle
```




### 10. Factor Analysis (FA)

```
varcov <- cov(data_scaled)
pc <- eigen(varcov)
eigenvalues <- pc$values

cat("Eigenvalues:\n")
```

```
## Eigenvalues:
```

```
print(eigenvalues)
```

```
##  [1] 7.54306907 2.30222599 1.07991428 0.84803302 0.68836440 0.46658177
##  [7] 0.32674539 0.24259098 0.15110919 0.11935564 0.08668754 0.06572282
## [13] 0.05460597 0.02499395
```

```
cat("\nEigenvectors (Principal Components):\n")
```

```
##
## Eigenvectors (Principal Components):
```

```r
print(pc$vectors)
```

```
##             [,1]        [,2]        [,3]        [,4]         [,5]         [,6]
##  [1,]  0.31858161 -0.03843826 -0.20341903  0.02934944 -0.208162585  0.256993676
##  [2,] -0.32017641  0.20727947 -0.18466137  0.05925938 -0.008156224  0.048129259
##  [3,] -0.31018299  0.25865221 -0.18690861  0.04905976  0.002701116  0.097750084
##  [4,] -0.07075063 -0.44192561  0.09497768 -0.48821352 -0.610966576 -0.204097153
##  [5,]  0.30750867 -0.27077397 -0.01266669  0.14199910 -0.182638679  0.119755678
##  [6,] -0.07950117  0.35313163 -0.65297811 -0.04541031 -0.459183508 -0.187030359
##  [7,]  0.26196644  0.04556564  0.02955828  0.40707227 -0.002316343 -0.827780608
##  [8,]  0.32018978 -0.03429636 -0.18636955  0.27379632 -0.075924617  0.230531001
##  [9,]  0.35053204  0.06117560 -0.09402687  0.00313735 -0.005772934  0.111776875
## [10,]  0.10394183  0.47346522  0.33273837 -0.40881222 -0.038671511 -0.123288232
## [11,]  0.32138476  0.10139008 -0.13313233 -0.13333961 -0.002849588  0.121024797
## [12,]  0.32575359  0.18325847  0.01940883 -0.20630999  0.059210655 -0.060697375
## [13,]  0.28066704  0.30518779  0.03189367 -0.31226516  0.131578003  0.006103356
## [14,]  0.02982592 -0.35660985 -0.52890748 -0.40863121  0.557658015 -0.216049908
##               [,7]        [,8]        [,9]       [,10]       [,11]        [,12]
##  [1,] -0.286631538 -0.10054488  0.77906966  0.03780049  0.20113001 -0.058270075
##  [2,] -0.044136450 -0.52895930 -0.08058391 -0.04023390  0.25459495  0.046404600
##  [3,] -0.026158829 -0.49313895  0.02390819 -0.06862720 -0.03463193  0.122724780
##  [4,]  0.074108008 -0.27641643 -0.04917903 -0.07184518 -0.04295721 -0.185431124
##  [5,] -0.197575162 -0.15176543 -0.27704810 -0.07191334  0.01543491  0.769407907
##  [6,]  0.011756591  0.39680876 -0.10208747 -0.02362453 -0.07050356  0.093907084
##  [7,] -0.005010222 -0.23449514  0.13031323  0.04976821 -0.01808137 -0.021589922
##  [8,] -0.290322733 -0.22315482 -0.41704509 -0.09120040 -0.19962496 -0.568186109
##  [9,]  0.020019982  0.08096472 -0.11267830 -0.08563404 -0.14373124  0.029768499
## [10,] -0.607614940 -0.01382808 -0.11327623  0.29144803 -0.03680316  0.044291766
## [11,]  0.494556134 -0.22011566 -0.08916506  0.72521844  0.03382202  0.009653786
## [12,]  0.162910283  0.02242373 -0.19582373 -0.36773926  0.75624158 -0.089939456
## [13,]  0.291267159 -0.21622282  0.18147152 -0.46311864 -0.50103366  0.097876231
## [14,] -0.246318283 -0.05265168 -0.02690195  0.05127316 -0.01429217  0.016926566
##              [,13]        [,14]
##  [1,]  0.028458016 -0.010309274
##  [2,] -0.014379427 -0.676380510
##  [3,] -0.255374826  0.677755737
##  [4,] -0.106715308  0.017045967
##  [5,]  0.165498569  0.022021964
##  [6,]  0.110828041 -0.007598504
##  [7,] -0.044171118  0.016556076
##  [8,]  0.212306871  0.060048966
##  [9,] -0.875394868 -0.192768936
## [10,]  0.001185020 -0.003290060
## [11,]  0.084461267  0.031199429
## [12,]  0.034967659  0.168331512
## [13,]  0.246990549 -0.107662412
## [14,] -0.001623393  0.016317203
```

```r
num_factors <- sum(eigenvalues > 1)
cat("Jumlah faktor berdasarkan Kaiser's Criterion:", num_factors, "\n")
```

```
## Jumlah faktor berdasarkan Kaiser’s Criterion: 3
```

```r
L <- matrix(nrow = ncol(data_scaled), ncol = num_factors)
for (i in 1:num_factors) {
  L[, i] <- sqrt(eigenvalues[i]) * pc$vectors[, i]
}
L
```

```
##              [,1]        [,2]        [,3]
##  [1,]  0.87497319 -0.05832271 -0.21139087
##  [2,] -0.87935326  0.31450697 -0.19189811
##  [3,] -0.85190668  0.39245527 -0.19423341
##  [4,] -0.19431412 -0.67053761  0.09869979
##  [5,]  0.84456175 -0.41084773 -0.01316309
##  [6,] -0.21834716  0.53580973 -0.67856783
##  [7,]  0.71948161  0.06913714  0.03071665
##  [8,]  0.87938997 -0.05203817 -0.19367323
##  [9,]  0.96272392  0.09282228 -0.09771171
## [10,]  0.28547258  0.71839294  0.34577814
## [11,]  0.88267193  0.15384006 -0.13834969
## [12,]  0.89467079  0.27805968  0.02016945
## [13,]  0.77084216  0.46306411  0.03314356
## [14,]  0.08191584 -0.54108727 -0.54963497
```

```r
cor_matrix <- cor(data_scaled)
eigen(cor_matrix)$values
```

```
##  [1] 7.54306907 2.30222599 1.07991428 0.84803302 0.68836440 0.46658177
##  [7] 0.32674539 0.24259098 0.15110919 0.11935564 0.08668754 0.06572282
## [13] 0.05460597 0.02499395
```

```r
fa_result <- fa(r = data_scaled,
                nfactors = num_factors,
                rotate = "varimax",
                scores = "tenBerge")
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

```r
fa_result$loadings
```

```
##
## Loadings:
##     MR1    MR2    MR3
## X1   0.891
## X2  -0.756         0.622
## X3  -0.729         0.679
## X4  -0.208 -0.397 -0.341
## X5   0.833 -0.267 -0.374
## X6          0.133  0.586
## X8   0.654  0.135 -0.116
## X9   0.930
```

```
## X10  0.955  0.197 -0.101
## X11  0.153  0.789
## X12  0.851  0.245
## X13  0.814  0.467 -0.143
## X14  0.681  0.616
## X15  0.114 -0.371 -0.116
##
##                  MR1   MR2   MR3
## SS loadings    6.731 1.730 1.515
## Proportion Var 0.481 0.124 0.108
## Cumulative Var 0.481 0.604 0.713
```

```
fa.diagram(fa_result$loadings)
```