

UAS_SNP_Kelompok K

KELOMPOK K

2023-12-06

Memanggil Library

```
library(readxl)
library(DescTools)
library(nonpar)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
```

Preprocessing Data

```
setwd("C:/Users/AURA/Music/OneDrive/Pictures/Kuliah/Data Sekunder")

data=read_excel("hidup_provinsi.xlsx")
data
```

```
## # A tibble: 34 x 5
##   Provinsi      '2020' '2021' '2022' '2023'
##   <chr>      <dbl> <dbl> <dbl> <dbl>
## 1 ACEH          72.7  72.7  72.9  73.1
## 2 SUMATERA UTARA    73   73.1  73.4  73.7
## 3 SUMATERA BARAT   73.5  73.6  73.9  74.1
## 4 RIAU           73.7  73.7  74.0  74.2
```

```
## 5 JAMBI 73.3 73.4 73.6 73.8
## 6 SUMATERA SELATAN 73.4 73.5 73.8 74.0
## 7 BENGKULU 72.6 72.6 72.9 73.1
## 8 LAMPUNG 73.7 73.7 74.0 74.2
## 9 KEP. BANGKA BELITUNG 73.4 73.5 73.7 73.9
## 10 KEPULAUAN RIAU 74.2 74.4 74.6 74.9
## # i 24 more rows
```

Uji Normalitas

```
uji_normalitas_shapiro <- function(data, alpha = 0.05) {
  # Mengambil hanya variabel numerik dari data
  numeric_vars <- Filter(is.numeric, data)

  # Inisialisasi vektor untuk menyimpan hasil uji
  hasil_uji <- vector("list", length = ncol(numeric_vars))

  # Inisialisasi vektor untuk menyimpan keterangan distribusi
  keterangan_distribusi <- character(length = ncol(numeric_vars))

  # Melakukan uji normalitas Shapiro-Wilk untuk setiap variabel numerik
  for (i in seq_along(numeric_vars)) {
    var <- numeric_vars[[i]]
    shapiro_test <- shapiro.test(var)
    hasil_uji[[i]] <- shapiro_test

    # Menyimpan keterangan distribusi
    if (shapiro_test$p.value < alpha) {
      keterangan_distribusi[i] <- "Distribusi Tidak Normal"
    } else {
      keterangan_distribusi[i] <- "Distribusi Normal"
    }
  }

  # Menyusun hasil uji dan keterangan distribusi menjadi data frame
  hasil_df <- data.frame(
    Variable = names(numeric_vars),
    Shapiro_P_Value = sapply(hasil_uji, function(x) x$p.value),
    Distribution_Status = keterangan_distribusi
  )

  return(hasil_df)
}
```

Uji Shapiro-Wilk

```
uji_normalitas_shapiro(data)
```

```
## Variable Shapiro_P_Value Distribution_Status
```

```
## 1      2020      0.0009418191 Distribusi Tidak Normal
## 2      2021      0.0012019256 Distribusi Tidak Normal
## 3      2022      0.0010560537 Distribusi Tidak Normal
## 4      2023      0.0013456906 Distribusi Tidak Normal
```

Summary Data

```
summary(data)
```

```
##      Provinsi      2020      2021      2022
## Length:34      Min.   :67.59      Min.   :67.72      Min.   :67.98
## Class :character 1st Qu.:71.27      1st Qu.:71.40      1st Qu.:71.67
## Mode  :character Median :73.28      Median :73.36      Median :73.55
##                      Mean  :72.55      Mean  :72.65      Mean  :72.89
##                      3rd Qu.:73.69      3rd Qu.:73.94      3rd Qu.:74.33
##                      Max.   :75.20      Max.   :75.28      Max.   :75.54
##      2023
## Min.   :68.17
## 1st Qu.:71.85
## Median :73.78
## Mean   :73.13
## 3rd Qu.:74.56
## Max.   :75.81
```

Pengambilan Data Numeric

```
datanumeric <- data[, -which(names(data) == "Provinsi")]
datanumeric
```

```
## # A tibble: 34 x 4
##   '2020' '2021' '2022' '2023'
##   <dbl> <dbl> <dbl> <dbl>
## 1  72.7  72.7  72.9  73.1
## 2  73    73.1  73.4  73.7
## 3  73.5  73.6  73.9  74.1
## 4  73.7  73.7  74.0  74.2
## 5  73.3  73.4  73.6  73.8
## 6  73.4  73.5  73.8  74.0
## 7  72.6  72.6  72.9  73.1
## 8  73.7  73.7  74.0  74.2
## 9  73.4  73.5  73.7  73.9
## 10 74.2  74.4  74.6  74.9
## # i 24 more rows
```

Perhitungan Statistika Deskriptif

```

statistika_deskriptif <- function(data) {
  # Mendapatkan hanya kolom-kolom numerik
  kolom_numerik <- data[, sapply(data, is.numeric)]
  # Inisialisasi vektor kosong untuk menyimpan hasil perhitungan
  variables <- names(kolom_numerik)
  means <- medians <- q1s <- q3s <- variances <- sds <- ranges <- sums <- mins <- maxs <- numeric(length(variables))
  # Loop untuk setiap variabel numerik
  for (i in seq_along(variables)) {
    variable <- variables[i]
    values <- kolom_numerik[[variable]]
    # Hitung statistika deskriptif
    means[i] <- mean(values)
    medians[i] <- median(values)
    q1s[i] <- quantile(values, 0.25)
    q3s[i] <- quantile(values, 0.75)
    variances[i] <- var(values)
    sds[i] <- sd(values)
    ranges[i] <- max(values) - min(values)
    sums[i] <- sum(values)
    mins[i] <- min(values)
    maxs[i] <- max(values)
  }

  hasil <- data.frame(Variabel = variables, Mean = means, Median = medians, Q1 = q1s, Q3 = q3s, Variance = variances, SD = sds, Range = ranges, Sum = sums, Min = mins, Max = maxs)

  return(hasil)
}

```

Nilai Statistika Deskriptif

```
statistika_deskriptif(data)
```

```

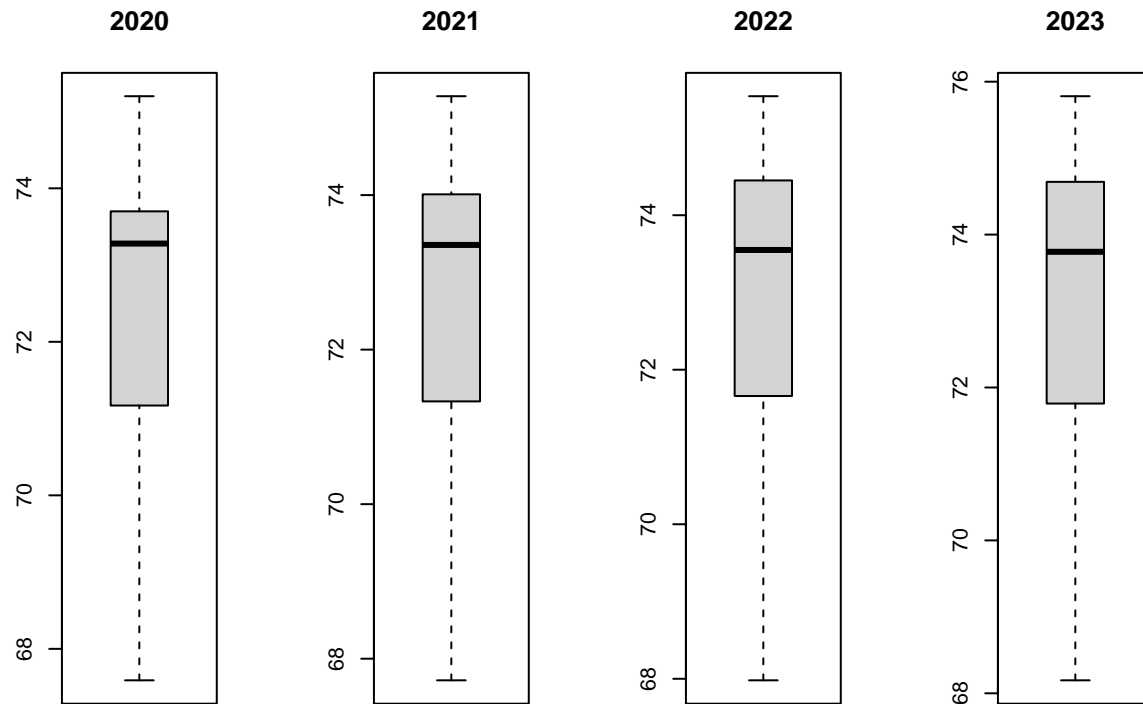
##   Variabel      Mean Median      Q1      Q3 Variance      SD Range      Sum
## 1    2020 72.55059 73.280 71.2675 73.6900 3.798109 1.948874 7.61 2466.72
## 2    2021 72.64706 73.355 71.4000 73.9400 3.766470 1.940740 7.56 2470.00
## 3    2022 72.89000 73.550 71.6700 74.3250 3.722109 1.929277 7.56 2478.26
## 4    2023 73.12559 73.775 71.8475 74.5625 3.692153 1.921498 7.64 2486.27
##      Min      Max
## 1 67.59 75.20
## 2 67.72 75.28
## 3 67.98 75.54
## 4 68.17 75.81

```

Visualisasi Data

Pengecekan Outlier

```
par(mfrow=c(1,4))
for(i in 1:4) {boxplot(datanumeric[,i], main=names(datanumeric)[i])}
```



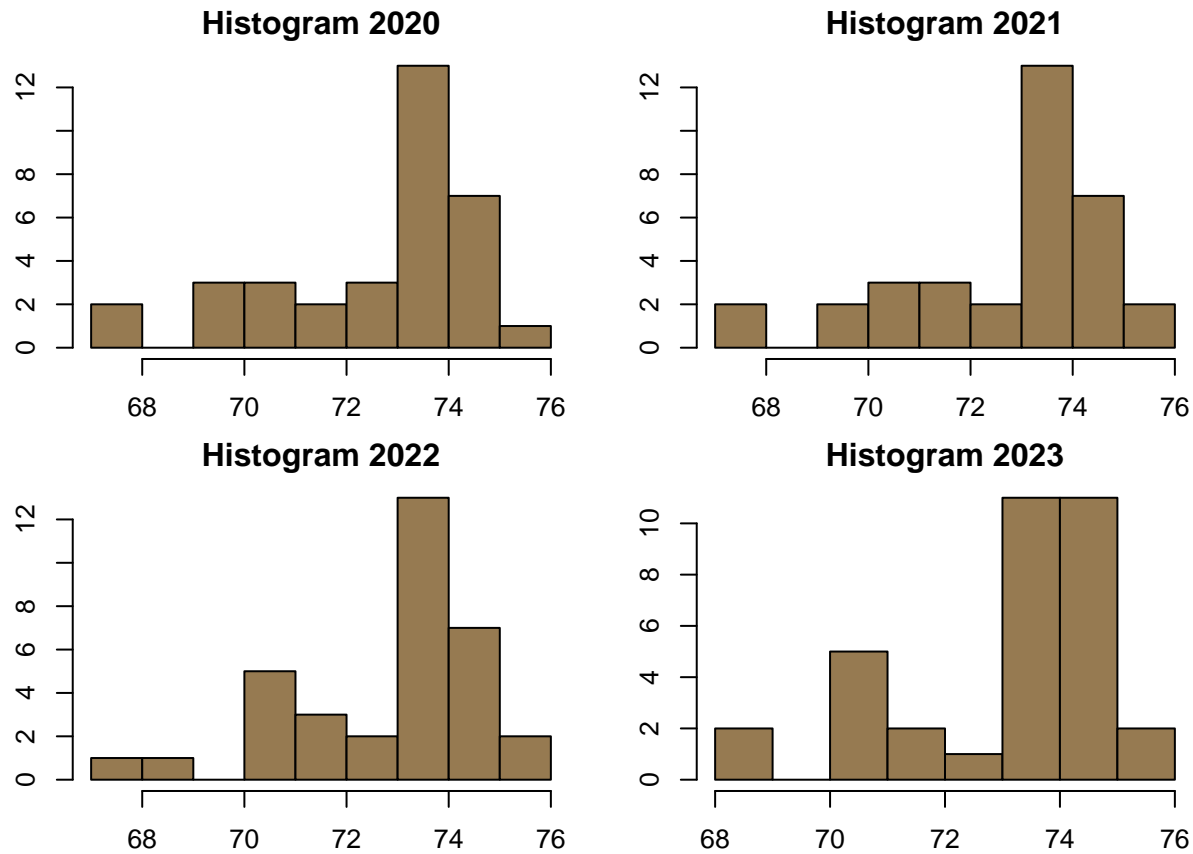
Pengecekan Distribusi Data

```
variabel_numerik <- c()
# Loop for untuk menelusuri kolom-kolom di dbdjabar
for (col_name in names(datanumeric)) {
  if (is.numeric(datanumeric[[col_name]])) {
    variabel_numerik <- c(variabel_numerik, col_name)
  }
}
# Menampilkan Hasil
print(variabel_numerik)
```

```
## [1] "2020" "2021" "2022" "2023"
```

```
# Inisialisasi plot
par(mfrow = c(2, 2), mar = c(2, 2, 2, 2))
# Loop for untuk membuat histogram untuk setiap variabel numerik
for (i in 1:length(variabel_numerik)) {
  var <- variabel_numerik[i]
```

```
hist_data <- datanumeric[[var]]
# Membuat Histogram
hist(hist_data, main = paste("Histogram", var), col = "#967A51", xlab = var, ylab = "Frequency")
}
```



Uji K-Sampel Independen

Uji Kruskal-Wallis

```
#Uji Kruskal-Wallis
kruskal.test(datanumeric)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  datanumeric
## Kruskal-Wallis chi-squared = 4.7256, df = 3, p-value = 0.193
```

Uji Jonckheere-Terpstra

```
#Uji Jonckheere-Terpstra
JonckheereTerpstraTest(datanumeric)
```

```
##
## Jonckheere-Terpstra test
##
## data: datanumeric
## JT = 4054.5, p-value = 0.02252
## alternative hypothesis: two.sided
```

Uji K-Sampel Dependen

Uji Friedman

```
#Uji Friedman
friedman.test(data.matrix(datanumeric))
```

```
##
## Friedman rank sum test
##
## data: data.matrix(datanumeric)
## Friedman chi-squared = 102, df = 3, p-value < 2.2e-16
```

Uji Perbandingan Ganda

```
#Uji Perbandingan Ganda
tabel_z = 0.05/(4*(3))

z = 2.64
print(paste("Nilai Z adalah", z))
```

```
## [1] "Nilai Z adalah 2.64"
```

```
rumus = z*(sqrt((34*4*5)/6))
print(paste("Hasil Perhitungan Uji Perbandingan Ganda", rumus))
```

```
## [1] "Hasil Perhitungan Uji Perbandingan Ganda 28.1049461839015"
```

```
sum_r_2020 <- 34
sum_r_2021 <- 68
sum_r_2022 <- 102
sum_r_2023 <- 136
sprintf("Nilai sum ranking dari tahun 2020 adalah %d, tahun 2021 adalah %d, tahun 2022 adalah %d, dan t
```

```
## [1] "Nilai sum ranking dari tahun 2020 adalah 34, tahun 2021 adalah 68, tahun 2022 adalah 102, dan t
```

```

if (abs(sum_r_2020-sum_r_2021)>rumus) {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2021 memberikan perbedaan")
} else {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2021 tidak memberikan perbedaan")
}

```

```
## [1] "Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2021 memberikan perbedaan"
```

```

if (abs(sum_r_2020-sum_r_2022)>rumus) {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2022 memberikan perbedaan")
} else {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2022 tidak memberikan perbedaan")
}

```

```
## [1] "Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2022 memberikan perbedaan"
```

```

if (abs(sum_r_2020-sum_r_2023)>rumus) {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2023 memberikan perbedaan")
} else {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2023 tidak memberikan perbedaan")
}

```

```
## [1] "Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2020 dan Tahun 2023 memberikan perbedaan"
```

```

if (abs(sum_r_2021-sum_r_2022)>rumus) {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2021 dan Tahun 2022 memberikan perbedaan")
} else {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2021 dan Tahun 2022 tidak memberikan perbedaan")
}

```

```
## [1] "Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2021 dan Tahun 2022 memberikan perbedaan"
```

```

if (abs(sum_r_2021-sum_r_2023)>rumus) {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2021 dan Tahun 2023 memberikan perbedaan")
} else {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2021 dan Tahun 2023 tidak memberikan perbedaan")
}

```

```
## [1] "Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2021 dan Tahun 2023 memberikan perbedaan"
```

```

if (abs(sum_r_2022-sum_r_2023)>rumus) {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2022 dan Tahun 2023 memberikan perbedaan")
} else {
  print("Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2022 dan Tahun 2023 tidak memberikan perbedaan")
}

```

```
## [1] "Nilai UHH di Provinsi-Provinsi Indonesia Tahun 2022 dan Tahun 2023 memberikan perbedaan"
```

Uji Page


```
#Uji Page
uji_page <- as.matrix(data.frame(datanumeric))
PageTest(uji_page)
```

```
##
## Page test for ordered alternatives
##
## data: uji_page
## L = 1020, p-value < 2.2e-16
```

Uji Asosiatif

Uji Korelasi Spearman

Fungsi Uji Spearman

```
uji_spearman <- function(data, variabel1, variabel2, alpha = 0.05) {
  # Mengambil kolom-kolom yang sesuai dengan variabel yang diberikan
  data_subset <- data[, c(variabel1, variabel2)]
  # Melakukan uji Spearman
  hasil_uji <- cor.test(data_subset[[variabel1]], data_subset[[variabel2]], method = "spearman")
  # Menyusun hasil ke dalam data frame
  hasil_df <- data.frame(
    Variabel1 = variabel1,
    Variabel2 = variabel2,
    rho = hasil_uji$estimate,
    P_value = hasil_uji$p.value,
    alpha = alpha,
    Kesimpulan = ifelse(hasil_uji$p.value < alpha, "Tolak H0", "Gagal Tolak H0")
  )
  return(hasil_df)
}
```

Nilai Uji Spearman

```
hasil_uji_2020_2021 <- uji_spearman(data, "2020", "2021")
hasil_uji_2020_2021
```

```
##      Variabel1 Variabel2      rho      P_value alpha Kesimpulan
## rho      2020      2021 0.997249 9.680857e-38 0.05      Tolak H0
```

```
hasil_uji_2020_2022 <- uji_spearman(data, "2020", "2022")
hasil_uji_2020_2022
```

```
##      Variabel1 Variabel2      rho      P_value alpha Kesimpulan
## rho      2020      2022 0.9886902 6.066845e-28 0.05      Tolak H0
```

```
hasil_uji_2020_2023 <- uji_spearman(data, "2020", "2023")
hasil_uji_2020_2023
```

```
##      Variabel1 Variabel2      rho      P_value alpha Kesimpulan
## rho      2020      2023 0.9739396 3.449706e-22 0.05      Tolak H0
```

```
hasil_uji_2021_2022 <- uji_spearman(data, "2021", "2022")
hasil_uji_2021_2022
```

```
##      Variabel1 Variabel2      rho      P_value alpha Kesimpulan
## rho      2021      2022 0.9920538 2.190837e-30 0.05      Tolak H0
```

```
hasil_uji_2021_2023 <- uji_spearman(data, "2021", "2023")
hasil_uji_2021_2023
```

```
##      Variabel1 Variabel2      rho      P_value alpha Kesimpulan
## rho      2021      2023 0.9770001 4.777492e-23 0.05      Tolak H0
```

```
hasil_uji_2022_2023 <- uji_spearman(data, "2022", "2023")
hasil_uji_2022_2023
```

```
##      Variabel1 Variabel2      rho      P_value alpha Kesimpulan
## rho      2022      2023 0.993505 8.785583e-32 0.05      Tolak H0
```

Nilai Seluruh Uji Spearman beserta Kesimpulannya

```
hasil_akhir_spearman <- rbind(hasil_uji_2020_2021, hasil_uji_2020_2022, hasil_uji_2020_2023,
hasil_uji_2021_2022, hasil_uji_2021_2023, hasil_uji_2022_2023)
hasil_akhir_spearman
```

```
##      Variabel1 Variabel2      rho      P_value alpha Kesimpulan
## rho      2020      2021 0.9972490 9.680857e-38 0.05      Tolak H0
## rho1      2020      2022 0.9886902 6.066845e-28 0.05      Tolak H0
## rho2      2020      2023 0.9739396 3.449706e-22 0.05      Tolak H0
## rho3      2021      2022 0.9920538 2.190837e-30 0.05      Tolak H0
## rho4      2021      2023 0.9770001 4.777492e-23 0.05      Tolak H0
## rho5      2022      2023 0.9935050 8.785583e-32 0.05      Tolak H0
```