



Lampiran 1. Data NTP Subsektor Tanaman Pangan dari tahun 2009-2025

| Tahun | Bulan   |          |       |       |        |        |       |         |           |         |          |          |
|-------|---------|----------|-------|-------|--------|--------|-------|---------|-----------|---------|----------|----------|
|       | Januari | Februari | Maret | April | Mei    | Juni   | Juli  | Agustus | September | Oktober | November | Desember |
| 2009  | 95,45   | 94,62    | 94,4  | 95,6  | 95,92  | 95,3   | 95,69 | 96,25   | 95,62     | 94,94   | 96,43    | 96,49    |
| 2010  | 95,29   | 95,05    | 94,39 | 95,33 | 95,07  | 93,62  | 93,54 | 93,81   | 94,44     | 95,19   | 95,42    | 94,62    |
| 2011  | 95,05   | 95,25    | 94,7  | 95,47 | 95,56  | 95,71  | 96,34 | 96,34   | 96,89     | 97,18   | 98,05    | 98,18    |
| 2012  | 97,75   | 96,53    | 95,66 | 95,07 | 94,89  | 96,09  | 95,35 | 95      | 95,03     | 95,38   | 94,46    | 94,4     |
| 2013  | 92,35   | 92,78    | 91,86 | 91,03 | 90,55  | 91,15  | 89,14 | 90,08   | 90,62     | 90,51   | 90,98    | 96,32    |
| 2014  | 95,26   | 94,67    | 94,63 | 92,8  | 91,58  | 92,05  | 93,4  | 93,63   | 94,34     | 95,78   | 96,48    | 94,76    |
| 2015  | 96,15   | 96,96    | 97,55 | 94,27 | 93,21  | 95,27  | 95,86 | 97,57   | 98,4      | 100,61  | 101,08   | 100,48   |
| 2016  | 99,81   | 99,62    | 98,04 | 96,34 | 97,09  | 96,71  | 95,9  | 96,53   | 96,73     | 97,36   | 97       | 95,82    |
| 2017  | 95,77   | 95,4     | 94,25 | 93,93 | 93,17  | 93,42  | 94,46 | 94,7    | 96,39     | 97,79   | 97,85    | 98,24    |
| 2018  | 100,45  | 100,94   | 99,24 | 96,93 | 96,24  | 95,72  | 94,85 | 95,43   | 97        | 98,79   | 99,09    | 99,4     |
| 2019  | 99,21   | 100,1    | 99,25 | 97,82 | 97,53  | 98,41  | 98,99 | 99,17   | 100,34    | 101,3   | 101,78   | 101,52   |
| 2020  | 100,5   | 100,96   | 102,1 | 99,87 | 98,86  | 97,95  | 98,77 | 99,73   | 99,41     | 100     | 98,57    | 97,87    |
| 2021  | 96,4    | 94,95    | 92,4  | 89,86 | 89,4   | 90,13  | 90,15 | 91,08   | 91,87     | 92,58   | 94,13    | 93,52    |
| 2022  | 93,1    | 93,11    | 92,41 | 90,02 | 89,72  | 88,38  | 86,96 | 89,21   | 92,17     | 93,47   | 94,53    | 95,15    |
| 2023  | 94,37   | 96,92    | 96,91 | 96,17 | 96,13  | 96,51  | 96,73 | 98,3    | 103,57    | 106,16  | 105,47   | 105,29   |
| 2024  | 105,25  | 109,42   | 105,4 | 95,93 | 94,82  | 98,16  | 98,41 | 99,54   | 99,55     | 100,27  | 96,91    | 97,69    |
| 2025  | 97,85   | 99,41    | 100,4 | 98,63 | 100,35 | 100,25 | 101,6 | 103,22  | 104,05    | 104,02  | 103,02   | 102,69   |

Lampiran 2. Daftar *package* R yang digunakan dalam penelitian

```

> library(ggplot2)
> library(tsibble)
> library(tseries)
> library(MASS)
> library(dplyr)
> library(TTR)
> library(forecast)
> library(lmtest)
> library(HoRM)
> library(dLagM)
> library(dynlm)
> library(MLmetrics)
> library(car)
> library(TSA)
> library(aTSA)
> library(rugarch)
> library(tidyr)
> library(rio)
> library(fGarch)

```

Lampiran 3. *Code* untuk import data

```

> data_NTP <- import("C:\\Users\\R I M A\\OneDrive\\Documents\\DATA NILAI TUKAR PETANI 2009-2024.xlsx")

```

Lampiran 4. *Code* R untuk analisis statistik deskriptif

```

> # Statistik deskriptif
> Mean <- mean(data_NTP$NTP, na.rm = TRUE)
> Median <- median(data_NTP$NTP, na.rm = TRUE)
> Min <- min(data_NTP$NTP, na.rm = TRUE)
> Max <- max(data_NTP$NTP, na.rm = TRUE)
> Std_Dev <- sd(data_NTP$NTP, na.rm = TRUE)
> Observations <- length(data_NTP$NTP)
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)

```

|   | Mean     | Median | Min   | Max    | Std_Dev  | Observations |
|---|----------|--------|-------|--------|----------|--------------|
| 1 | 96.40877 | 96.11  | 86.96 | 109.42 | 3.623907 | 204          |

## Lampiran 5. Code R plot semua data

```

> plot(NTP_data,
+     main = "Nilai Tukar Petani 2009-2025",
+     xlab = "Tahun",
+     ylab = "Nilai Tukar Petani",
+     col = "black",
+     type = "o",
+     lwd = 2,                # Ketebalan garis
+     xaxt = "n")           # Hilangkan sumbu x default
> # Tambahkan sumbu x manual
> axis(1, at = seq(2009, 2025, by = 1), labels = seq(2009, 2025, by = 1))

```

## Lampiran 5. Code R Plot Data NTP Sub Sektor Tanaman Pangan

```

> # PLOT SEASONAL DAN TREN
> seasonplot(NTP_data,
+     12, # Frekuensi data (12 untuk data bulanan)
+     main = "Seasonal Plot Nilai Tukar Petani (2009-2025)",
+     xlab = "Bulan",
+     ylab = "Nilai Tukar Petani",
+     year.labels = TRUE,
+     col = rainbow(16))

```



## Lampiran 6. Code Data Training

```

> NTP_datatrain <- NTP_data[1:180]
> NTP_datatrain
 [1] 95.45 94.62 94.40 95.60 95.92 95.30 95.69 96.25 95.62 94.94 96.43 96.49 95.29
 [14] 95.05 94.39 95.33 95.07 93.62 93.54 93.81 94.44 95.19 95.42 94.62 95.05 95.25
 [27] 94.70 95.47 95.56 95.71 96.34 96.34 96.89 97.18 98.05 98.18 97.75 96.53 95.66
 [40] 95.07 94.89 96.09 95.35 95.00 95.03 95.38 94.46 94.40 92.35 92.78 91.86 91.03
 [53] 90.55 91.15 89.14 90.08 90.62 90.51 90.98 96.32 95.26 94.67 94.63 92.80 91.58
 [66] 92.05 93.40 93.63 94.34 95.78 96.48 94.76 96.15 96.96 97.55 94.27 93.21 95.27
 [79] 95.86 97.57 98.40 100.61 101.08 100.48 99.81 99.62 98.04 96.34 97.09 96.71 95.90
 [92] 96.53 96.73 97.36 97.00 95.82 95.77 95.40 94.25 93.93 93.17 93.42 94.46 94.70
 [105] 96.39 97.79 97.85 98.24 100.45 100.94 99.24 96.93 96.24 95.72 94.85 95.43 97.00
 [118] 98.79 99.09 99.40 99.21 100.10 99.25 97.82 97.53 98.41 98.99 99.17 100.34 101.30
 [131] 101.78 101.52 100.50 100.96 102.06 99.87 98.86 97.95 98.77 99.73 99.41 100.00 98.57
 [144] 97.87 96.40 94.95 92.40 89.86 89.40 90.13 90.15 91.08 91.87 92.58 94.13 93.52
 [157] 93.10 93.11 92.41 90.02 89.72 88.38 86.96 89.21 92.17 93.47 94.53 95.15 94.37
 [170] 96.92 96.91 96.17 96.13 96.51 96.73 98.30 103.57 106.16 105.47 105.29

```

```
> NTP_train_ts
      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2009 95.45 94.62 94.40 95.60 95.92 95.30 95.69 96.25 95.62 94.94 96.43 96.49
2010 95.29 95.05 94.39 95.33 95.07 93.62 93.54 93.81 94.44 95.19 95.42 94.62
2011 95.05 95.25 94.70 95.47 95.56 95.71 96.34 96.34 96.89 97.18 98.05 98.18
2012 97.75 96.53 95.66 95.07 94.89 96.09 95.35 95.00 95.03 95.38 94.46 94.40
2013 92.35 92.78 91.86 91.03 90.55 91.15 89.14 90.08 90.62 90.51 90.98 96.32
2014 95.26 94.67 94.63 92.80 91.58 92.05 93.40 93.63 94.34 95.78 96.48 94.76
2015 96.15 96.96 97.55 94.27 93.21 95.27 95.86 97.57 98.40 100.61 101.08 100.48
2016 99.81 99.62 98.04 96.34 97.09 96.71 95.90 96.53 96.73 97.36 97.00 95.82
2017 95.77 95.40 94.25 93.93 93.17 93.42 94.46 94.70 96.39 97.79 97.85 98.24
2018 100.45 100.94 99.24 96.93 96.24 95.72 94.85 95.43 97.00 98.79 99.09 99.40
2019 99.21 100.10 99.25 97.82 97.53 98.41 98.99 99.17 100.34 101.30 101.78 101.52
2020 100.50 100.96 102.06 99.87 98.86 97.95 98.77 99.73 99.41 100.00 98.57 97.87
2021 96.40 94.95 92.40 89.86 89.40 90.13 90.15 91.08 91.87 92.58 94.13 93.52
2022 93.10 93.11 92.41 90.02 89.72 88.38 86.96 89.21 92.17 93.47 94.53 95.15
2023 94.37 96.92 96.91 96.17 96.13 96.51 96.73 98.30 103.57 106.16 105.47 105.29
```

### Lampiran 7. Code Plot Data Testing

```
> NTP_datatest <- NTP_data[181:204]
> # Ubah ke time series
> NTP_test_ts <- ts(NTP_datatest, start = c(2024, 1), frequency = 12)
> NTP_test_ts
      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2024 105.25 109.42 105.37 95.93 94.82 98.16 98.41 99.54 99.55 100.27 96.91 97.69
2025 97.85 99.41 100.41 98.63 100.35 100.25 101.60 103.22 104.05 104.02 103.02 102.69
```

### Lampiran 8. Code uji stasioner dalam ragam

```
> #STASIONER DALAM RAGAM
> index <- seq(1:180)
> bc = boxcox(NTP_train_ts~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.42
```

```

> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
 [1] -2.00 -1.99 -1.98 -1.97 -1.96 -1.95 -1.94 -1.93 -1.92 -1.91 -1.90 -1.89 -1.88 -1.87
 [15] -1.86 -1.85 -1.84 -1.83 -1.82 -1.81 -1.80 -1.79 -1.78 -1.77 -1.76 -1.75 -1.74 -1.73
 [29] -1.72 -1.71 -1.70 -1.69 -1.68 -1.67 -1.66 -1.65 -1.64 -1.63 -1.62 -1.61 -1.60 -1.59
 [43] -1.58 -1.57 -1.56 -1.55 -1.54 -1.53 -1.52 -1.51 -1.50 -1.49 -1.48 -1.47 -1.46 -1.45
 [57] -1.44 -1.43 -1.42 -1.41 -1.40 -1.39 -1.38 -1.37 -1.36 -1.35 -1.34 -1.33 -1.32 -1.31
 [71] -1.30 -1.29 -1.28 -1.27 -1.26 -1.25 -1.24 -1.23 -1.22 -1.21 -1.20 -1.19 -1.18 -1.17
 [85] -1.16 -1.15 -1.14 -1.13 -1.12 -1.11 -1.10 -1.09 -1.08 -1.07 -1.06 -1.05 -1.04 -1.03
 [99] -1.02 -1.01 -1.00 -0.99 -0.98 -0.97 -0.96 -0.95 -0.94 -0.93 -0.92 -0.91 -0.90 -0.89
 [113] -0.88 -0.87 -0.86 -0.85 -0.84 -0.83 -0.82 -0.81 -0.80 -0.79 -0.78 -0.77 -0.76 -0.75
 [127] -0.74 -0.73 -0.72 -0.71 -0.70 -0.69 -0.68 -0.67 -0.66 -0.65 -0.64 -0.63 -0.62 -0.61
 [141] -0.60 -0.59 -0.58 -0.57 -0.56 -0.55 -0.54 -0.53 -0.52 -0.51 -0.50 -0.49 -0.48 -0.47
 [155] -0.46 -0.45 -0.44 -0.43 -0.42 -0.41 -0.40 -0.39 -0.38 -0.37 -0.36 -0.35 -0.34 -0.33
 [169] -0.32 -0.31 -0.30 -0.29 -0.28 -0.27 -0.26 -0.25 -0.24 -0.23 -0.22 -0.21 -0.20 -0.19
 [183] -0.18 -0.17 -0.16 -0.15 -0.14 -0.13 -0.12 -0.11 -0.10 -0.09 -0.08 -0.07 -0.06 -0.05
 [197] -0.04 -0.03 -0.02 -0.01 0.00 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09
 [211] 0.10 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.20 0.21 0.22 0.23
 [225] 0.24 0.25 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37
 [239] 0.38 0.39 0.40 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50 0.51
 [253] 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.60 0.61 0.62 0.63 0.64 0.65
 [267] 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79
 [281] 0.80 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93
 [295] 0.94 0.95 0.96 0.97 0.98 0.99 1.00 1.01 1.02 1.03 1.04 1.05 1.06 1.07
 [309] 1.08 1.09 1.10 1.11 1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19 1.20 1.21
 [323] 1.22 1.23 1.24 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32 1.33 1.34 1.35
 [337] 1.36 1.37 1.38 1.39 1.40 1.41 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49
 [351] 1.50 1.51 1.52 1.53 1.54 1.55 1.56 1.57 1.58 1.59 1.60 1.61 1.62 1.63
 [365] 1.64 1.65 1.66 1.67 1.68 1.69 1.70 1.71 1.72 1.73 1.74 1.75 1.76 1.77
 [379] 1.78 1.79 1.80 1.81 1.82 1.83 1.84 1.85 1.86 1.87 1.88 1.89 1.90 1.91
 [393] 1.92 1.93 1.94 1.95 1.96 1.97 1.98 1.99 2.00 2.01 2.02 2.03 2.04 2.05
 [407] 2.06 2.07 2.08 2.09 2.10 2.11 2.12 2.13 2.14 2.15 2.16 2.17 2.18 2.19

 [421] 2.20 2.21 2.22 2.23 2.24 2.25 2.26 2.27 2.28 2.29 2.30 2.31 2.32 2.33
 [435] 2.34 2.35 2.36 2.37 2.38 2.39 2.40 2.41 2.42 2.43 2.44 2.45 2.46 2.47
 [449] 2.48 2.49 2.50 2.51 2.52 2.53 2.54 2.55 2.56 2.57 2.58 2.59 2.60 2.61
 [463] 2.62 2.63 2.64 2.65 2.66 2.67 2.68 2.69 2.70 2.71 2.72 2.73 2.74 2.75
 [477] 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.89
 [491] 2.90 2.91 2.92 2.93 2.94 2.95 2.96 2.97 2.98 2.99 3.00 3.01 3.02 3.03
 [505] 3.04 3.05 3.06 3.07 3.08 3.09 3.10 3.11 3.12 3.13 3.14 3.15 3.16 3.17
 [519] 3.18 3.19 3.20 3.21 3.22 3.23 3.24 3.25 3.26 3.27 3.28 3.29 3.30 3.31
 [533] 3.32 3.33 3.34 3.35 3.36 3.37 3.38 3.39 3.40 3.41 3.42 3.43 3.44 3.45
 [547] 3.46 3.47 3.48 3.49 3.50 3.51 3.52 3.53 3.54 3.55 3.56 3.57 3.58 3.59
 [561] 3.60 3.61 3.62 3.63 3.64 3.65 3.66

```

Lampiran 9. Code R untuk transformasi data

```

> NTP_datatrain.new <- BoxCox(NTP_train_ts, lambda = 0.42)
> index <- seq(1:180)
> bc = boxcox(NTP_datatrain.new~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 1

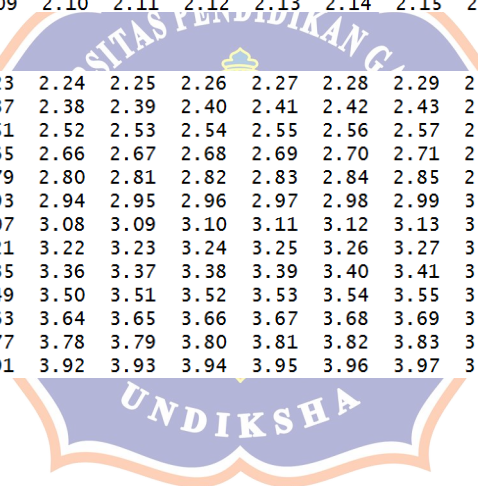
```

```

> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
[1] -2.00 -1.99 -1.98 -1.97 -1.96 -1.95 -1.94 -1.93 -1.92 -1.91 -1.90 -1.89 -1.88 -1.87
[15] -1.86 -1.85 -1.84 -1.83 -1.82 -1.81 -1.80 -1.79 -1.78 -1.77 -1.76 -1.75 -1.74 -1.73
[29] -1.72 -1.71 -1.70 -1.69 -1.68 -1.67 -1.66 -1.65 -1.64 -1.63 -1.62 -1.61 -1.60 -1.59
[43] -1.58 -1.57 -1.56 -1.55 -1.54 -1.53 -1.52 -1.51 -1.50 -1.49 -1.48 -1.47 -1.46 -1.45
[57] -1.44 -1.43 -1.42 -1.41 -1.40 -1.39 -1.38 -1.37 -1.36 -1.35 -1.34 -1.33 -1.32 -1.31
[71] -1.30 -1.29 -1.28 -1.27 -1.26 -1.25 -1.24 -1.23 -1.22 -1.21 -1.20 -1.19 -1.18 -1.17
[85] -1.16 -1.15 -1.14 -1.13 -1.12 -1.11 -1.10 -1.09 -1.08 -1.07 -1.06 -1.05 -1.04 -1.03
[99] -1.02 -1.01 -1.00 -0.99 -0.98 -0.97 -0.96 -0.95 -0.94 -0.93 -0.92 -0.91 -0.90 -0.89
[113] -0.88 -0.87 -0.86 -0.85 -0.84 -0.83 -0.82 -0.81 -0.80 -0.79 -0.78 -0.77 -0.76 -0.75
[127] -0.74 -0.73 -0.72 -0.71 -0.70 -0.69 -0.68 -0.67 -0.66 -0.65 -0.64 -0.63 -0.62 -0.61
[141] -0.60 -0.59 -0.58 -0.57 -0.56 -0.55 -0.54 -0.53 -0.52 -0.51 -0.50 -0.49 -0.48 -0.47
[155] -0.46 -0.45 -0.44 -0.43 -0.42 -0.41 -0.40 -0.39 -0.38 -0.37 -0.36 -0.35 -0.34 -0.33
[169] -0.32 -0.31 -0.30 -0.29 -0.28 -0.27 -0.26 -0.25 -0.24 -0.23 -0.22 -0.21 -0.20 -0.19
[183] -0.18 -0.17 -0.16 -0.15 -0.14 -0.13 -0.12 -0.11 -0.10 -0.09 -0.08 -0.07 -0.06 -0.05
[197] -0.04 -0.03 -0.02 -0.01 0.00 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09
[211] 0.10 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.20 0.21 0.22 0.23
[225] 0.24 0.25 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37
[239] 0.38 0.39 0.40 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50 0.51
[253] 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.60 0.61 0.62 0.63 0.64 0.65
[267] 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79
[281] 0.80 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93
[295] 0.94 0.95 0.96 0.97 0.98 0.99 1.00 1.01 1.02 1.03 1.04 1.05 1.06 1.07
[309] 1.08 1.09 1.10 1.11 1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19 1.20 1.21
[323] 1.22 1.23 1.24 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32 1.33 1.34 1.35
[337] 1.36 1.37 1.38 1.39 1.40 1.41 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49
[351] 1.50 1.51 1.52 1.53 1.54 1.55 1.56 1.57 1.58 1.59 1.60 1.61 1.62 1.63
[365] 1.64 1.65 1.66 1.67 1.68 1.69 1.70 1.71 1.72 1.73 1.74 1.75 1.76 1.77
[379] 1.78 1.79 1.80 1.81 1.82 1.83 1.84 1.85 1.86 1.87 1.88 1.89 1.90 1.91
[393] 1.92 1.93 1.94 1.95 1.96 1.97 1.98 1.99 2.00 2.01 2.02 2.03 2.04 2.05
[407] 2.06 2.07 2.08 2.09 2.10 2.11 2.12 2.13 2.14 2.15 2.16 2.17 2.18 2.19

[421] 2.20 2.21 2.22 2.23 2.24 2.25 2.26 2.27 2.28 2.29 2.30 2.31 2.32 2.33
[435] 2.34 2.35 2.36 2.37 2.38 2.39 2.40 2.41 2.42 2.43 2.44 2.45 2.46 2.47
[449] 2.48 2.49 2.50 2.51 2.52 2.53 2.54 2.55 2.56 2.57 2.58 2.59 2.60 2.61
[463] 2.62 2.63 2.64 2.65 2.66 2.67 2.68 2.69 2.70 2.71 2.72 2.73 2.74 2.75
[477] 2.76 2.77 2.78 2.79 2.80 2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.89
[491] 2.90 2.91 2.92 2.93 2.94 2.95 2.96 2.97 2.98 2.99 3.00 3.01 3.02 3.03
[505] 3.04 3.05 3.06 3.07 3.08 3.09 3.10 3.11 3.12 3.13 3.14 3.15 3.16 3.17
[519] 3.18 3.19 3.20 3.21 3.22 3.23 3.24 3.25 3.26 3.27 3.28 3.29 3.30 3.31
[533] 3.32 3.33 3.34 3.35 3.36 3.37 3.38 3.39 3.40 3.41 3.42 3.43 3.44 3.45
[547] 3.46 3.47 3.48 3.49 3.50 3.51 3.52 3.53 3.54 3.55 3.56 3.57 3.58 3.59
[561] 3.60 3.61 3.62 3.63 3.64 3.65 3.66 3.67 3.68 3.69 3.70 3.71 3.72 3.73
[575] 3.74 3.75 3.76 3.77 3.78 3.79 3.80 3.81 3.82 3.83 3.84 3.85 3.86 3.87
[589] 3.88 3.89 3.90 3.91 3.92 3.93 3.94 3.95 3.96 3.97 3.98 3.99 4.00

```



Lampiran 10. *Code* Uji stasioner dalam rataaan

```

> #STASIONER DALAM RATAAN
> #sebelum differencing
> par(mfrow=c(1,1))
> plot(NTP_datatrain.new, col = "black",
+      main = "Data Sebelum Difference",
+      xlab = "bulan ke-", type="o",
+      ylab="Nilai Tukar Petani",lty=1, lwd=1.5)
> plot.ts(NTP_datatrain.new, lty=1, xlab="tahun",
+        ylab="Data Sebelum Difference",
+        main="Plot First Difference ")
> acf(NTP_datatrain.new, lag.max = 36)
> tseries::adf.test(NTP_datatrain.new)


```

## Augmented Dickey-Fuller Test

```

data: NTP_datatrain.new
Dickey-Fuller = -2.3745, Lag order = 5, p-value = 0.4197
alternative hypothesis: stationary

```

Lampiran 11. *Code* untuk *differencing* data


```

> #Uji ADF setelah difference
> train.diff <- diff(NTP_datatrain.new, differences = 1)
> plot.ts(train.diff, lty=1, xlab="Tahun",
+        ylab="Data Difference",
+        main="Plot First Difference ")
> acf(train.diff, main = "ACF Setelah Difference")
> tseries::adf.test(train.diff)

```

## Augmented Dickey-Fuller Test

```

data: train.diff
Dickey-Fuller = -6.5964, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary

```

Lampiran 12. *Code* untuk menampilkan plot ACF dan PACF

```

> acf(train.diff, lag.max = 36)
> pacf(train.diff, lag.max = 36)

```

Lampiran 12. *Code differencing musiman*

```
> train.diff2 <- diff(train.diff, lag = 12, differences = 1)
> acf(train.diff2, lag.max = 36)
> pacf(train.diff2, lag.max = 36)
```

Lampiran 13. *Code estimasi model SARIMA (0,1,0)(1,1,0)<sup>12</sup>*

```
> #PENENTUAN MODEL SARIMA
> #MODEL 1
> Arima.1=Arima(NTP_datatrain.new, order=c(0,1,0),seasonal=list(order=
c(1,1,0), period=12),method="ML")
> summary(Arima.1)
Series: NTP_datatrain.new
ARIMA(0,1,0)(1,1,0)[12]

Coefficients:
      sar1
      -0.4291
s.e.    0.0701

sigma^2 = 0.007917:  log likelihood = 166.51
AIC=-329.03  AICC=-328.96  BIC=-322.79

Training set error measures:
              ME      RMSE      MAE      MPE
Training set 0.004077607 0.08544506 0.06202294 0.02863719
              MAPE      MASE      ACF1
Training set 0.4492921 0.2903861 0.1060168
> lmtest::coeftest(Arima.1)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
sar1 -0.429107  0.070138 -6.1181 9.471e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> lmtest::coeftest(Arima.1)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
sar1 -0.429107  0.070138 -6.1181 9.471e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lampiran 14. *Code* estimasi model SARIMA (0,1,0)(2,1,0)<sup>12</sup>

```

> #MODEL 2
> Arima.1=Arima(NTP_datatrain.new, order=c(0,1,0),seasonal=list(order=
c(2,1,0), period=12),method="ML")
> summary(Arima.1)
Series: NTP_datatrain.new
ARIMA(0,1,0)(2,1,0)[12]

Coefficients:
      sar1      sar2
-0.5753  -0.3011
s.e.    0.0773   0.0766

sigma^2 = 0.007205:  log likelihood = 173.67
AIC=-341.34  AICc=-341.19  BIC=-331.99

Training set error measures:
              ME      RMSE      MAE      MPE      MAPE
Training set 0.003634369 0.08127053 0.05831632 0.02493228 0.422641
              MASE      ACF1
Training set 0.273032 0.106084
> lmtest::coeftest(Arima.1)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
sar1 -0.575312   0.077320 -7.4407 1.002e-13 ***
sar2 -0.301146   0.076561 -3.9334 8.375e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Lampiran 15. *Code* estimasi model SARIMA (0,1,0)(1,1,1)<sup>12</sup>

```

> #MODEL 3
> Arima.1=Arima(NTP_datatrain.new, order=c(0,1,0),seasonal=list(order=
c(1,1,1), period=12),method="ML")
> summary(Arima.1)
Series: NTP_datatrain.new
ARIMA(0,1,0)(1,1,1)[12]

Coefficients:
      sar1      sma1
  0.0270  -0.7823
s.e.    0.1035   0.0830

sigma^2 = 0.006371:  log likelihood = 181
AIC=-356  AICc=-355.85  BIC=-346.64

Training set error measures:
              ME      RMSE      MAE      MPE
Training set 0.003131861 0.07641803 0.05444424 0.02107694
              MAPE      MASE      ACF1
Training set 0.3946713 0.2549033 0.1116345
> lmtest::coeftest(Arima.1)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
sar1  0.027034   0.103459  0.2613  0.7939
sma1 -0.782265   0.082976 -9.4276 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Lampiran 16. *Code* estimasi model SARIMA (0,1,0)(0,1,1)<sup>12</sup>

```

> #MODEL 4
> Arima.1=Arima(NTP_datatrain.new, order=c(0,1,0),seasonal=list(order=
c(0,1,1), period=12),method="ML")
> summary(Arima.1)
Series: NTP_datatrain.new
ARIMA(0,1,0)(0,1,1)[12]

Coefficients:
      sma1
      -0.7698
s.e.    0.0693

sigma^2 = 0.006338:  log likelihood = 180.96
AIC=-357.93  AICc=-357.85  BIC=-351.69

Training set error measures:
              ME      RMSE      MAE      MPE
Training set 0.003127961 0.07645082 0.05439541 0.02102354
              MAPE      MASE      ACF1
Training set 0.3943068 0.2546747 0.1111674
> lmtest::coeftest(Arima.1)

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
sma1 -0.769840   0.069281 -11.112 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Lampiran 17. *Code* untuk Diagnostik Model

```

> library(forecast)
> #Diagnosis Model
> residual <- Arima.1$residuals
> tsdiag(Arima.1)

```

Lampiran 18. *Code* Uji *White noise*

```

> #uji whitenoise
> Box.test(residual, type = "Ljung") #tak tolak H0 > sisaan saling bebas

Box-Ljung test

data: residual
X-squared = 2.2618, df = 1, p-value = 0.1326

```

Lampiran 19. *Code R* perbandingan data testing dan peramalan menggunakan

Model SARIMA (0,1,0)(1,1,0)<sup>12</sup>

```
> #Peralaman Sarima
> ramalan.da <- forecast::forecast(Arima.4, h = 12)
> hasil1 <- InvBoxCox(ramalan.da$mean, lambda = 0.42)
> # Bandingkan
> perbandingan <- cbind(
+   Aktual = as.numeric(NTP_test_ts),
+   Hasil_Forecast = as.numeric(hasil1)
+ )
> perbandingan
```

|       | Aktual | Hasil_Forecast |
|-------|--------|----------------|
| [1,]  | 105.25 | 104.7542       |
| [2,]  | 109.42 | 105.2870       |
| [3,]  | 105.37 | 104.5632       |
| [4,]  | 95.93  | 102.8316       |
| [5,]  | 94.82  | 102.4360       |
| [6,]  | 98.16  | 102.3999       |
| [7,]  | 98.41  | 102.2919       |
| [8,]  | 99.54  | 103.5120       |
| [9,]  | 99.55  | 105.7616       |
| [10,] | 100.27 | 107.1899       |
| [11,] | 96.91  | 107.4108       |
| [12,] | 97.69  | 107.3282       |
| [13,] | 97.85  | 104.7542       |
| [14,] | 99.41  | 105.2870       |
| [15,] | 100.41 | 104.5632       |
| [16,] | 98.63  | 102.8316       |
| [17,] | 100.35 | 102.4360       |
| [18,] | 100.25 | 102.3999       |
| [19,] | 101.60 | 102.2919       |
| [20,] | 103.22 | 103.5120       |
| [21,] | 104.05 | 105.7616       |
| [22,] | 104.02 | 107.1899       |
| [23,] | 103.02 | 107.4108       |
| [24,] | 102.69 | 107.3282       |





```

+ # Sumbu Y mulai dari 0
+ scale_y_continuous(limits = c(0, NA)) +
+ theme_minimal(base_size = 14) +
+ theme(
+   plot.title = element_text(face = "bold", size = 16),
+   legend.position = "top",
+   legend.title = element_blank(),
+   axis.text.x = element_text(angle = 45, hjust = 1)
+ )

```

### Lampiran 21. Code Pengukuran Akurasi Model

```

> accuracy(
+   ts(hasil1, start = start(NTP_test_ts), frequency = frequency(NTP_test_ts)),
+   NTP_test_ts
+ )

```

|          | ME        | RMSE     | MAE      | MPE       | MAPE     | ACF1      | Theil's U |
|----------|-----------|----------|----------|-----------|----------|-----------|-----------|
| Test set | -4.537164 | 6.205265 | 5.443103 | -4.692436 | 5.528097 | 0.6135656 | 1.850261  |

```

> dugaan <- as.numeric(hasil1)
> aktual <- as.numeric(NTP_test_ts)
> MAPE <- mean(abs((aktual - dugaan) / aktual)) * 100
> MAE <- mean(abs(aktual - dugaan))
> hasil_evaluasi <- data.frame(
+   MAPE = MAPE,
+   MAE = MAE
+ )
> print(hasil_evaluasi)

```

|   | MAPE     | MAE    |
|---|----------|--------|
| 1 | 4.431631 | 4.3993 |

### Lampiran 22. Code model SARIMA terbaik untuk tahun 2026 dan 2027

```

> NTP_full_bc <- BoxCox(NTP_data, lambda = 0.42)
> Arima.final <- Arima(
+   NTP_full_bc,
+   order = c(0,1,0),
+   seasonal = list(order = c(0,1,1), period = 12),
+   method = "ML"
+ )
> summary(Arima.final)
Series: NTP_full_bc
ARIMA(0,1,0)(0,1,1)[12]

Coefficients:
      sma1
      -0.7917
s.e.      0.0528

sigma^2 = 0.008839: log likelihood = 175.3
AIC=-346.59 AICC=-346.53 BIC=-340.09

Training set error measures:

```

|              | ME          | RMSE       | MAE        | MPE         | MAPE      |
|--------------|-------------|------------|------------|-------------|-----------|
| Training set | 0.001319806 | 0.09073174 | 0.06108914 | 0.007725402 | 0.4407507 |

```


```

|              | MASE      | ACF1      |
|--------------|-----------|-----------|
| Training set | 0.2645639 | 0.1493734 |

Lampiran 23. *Code R* untuk forecast dengan interval prediksi

```

> library(forecast)
> # Forecast dengan interval 95%
> forecast_2627 <- forecast::forecast(
+   Arima.final,
+   h = 24,
+   level = 95
+ )
> # Inverse Box-Cox
> forecast_mean <- InvBoxCox(forecast_2627$mean, lambda = 0.42)
> lower95 <- InvBoxCox(forecast_2627$lower[,1], lambda = 0.42)
> upper95 <- InvBoxCox(forecast_2627$upper[,1], lambda = 0.42)
> # Time series hasil prediksi
> forecast_2627_ts <- ts(
+   forecast_mean,
+   start = c(2026,1),
+   frequency = 12
+ )
> # Time series interval
> lower95_ts <- ts(lower95, start = c(2026,1), frequency = 12)
> upper95_ts <- ts(upper95, start = c(2026,1), frequency = 12)

> print(hasil_forecast_2627)

```

|    | Bulan    | Prediksi | Lower_80 | Upper_80 | Lower_95 | Upper_95 |
|----|----------|----------|----------|----------|----------|----------|
| 1  | Jan 2026 | 102.40   | 100.64   | 104.18   | 99.72    | 105.12   |
| 2  | Feb 2026 | 103.70   | 101.20   | 106.24   | 99.90    | 107.59   |
| 3  | Mar 2026 | 102.81   | 99.77    | 105.90   | 98.18    | 107.56   |
| 4  | Apr 2026 | 99.82    | 96.37    | 103.33   | 94.58    | 105.22   |
| 5  | May 2026 | 99.74    | 95.90    | 103.67   | 93.90    | 105.79   |
| 6  | Jun 2026 | 100.27   | 96.05    | 104.60   | 93.86    | 106.93   |
| 7  | Ju1 2026 | 100.53   | 95.97    | 105.22   | 93.61    | 107.75   |
| 8  | Aug 2026 | 101.78   | 96.88    | 106.83   | 94.34    | 109.56   |
| 9  | Sep 2026 | 103.27   | 98.02    | 108.67   | 95.31    | 111.60   |
| 10 | Oct 2026 | 104.24   | 98.68    | 109.97   | 95.81    | 113.07   |
| 11 | Nov 2026 | 103.60   | 97.80    | 109.59   | 94.81    | 112.84   |
| 12 | Dec 2026 | 103.61   | 97.56    | 109.88   | 94.44    | 113.28   |
| 13 | Jan 2027 | 103.32   | 96.93    | 109.96   | 93.64    | 113.56   |
| 14 | Feb 2027 | 104.63   | 97.86    | 111.67   | 94.38    | 115.51   |
| 15 | Mar 2027 | 103.73   | 96.68    | 111.08   | 93.06    | 115.09   |
| 16 | Apr 2027 | 100.73   | 93.50    | 108.27   | 89.79    | 112.39   |
| 17 | May 2027 | 100.65   | 93.14    | 108.50   | 89.30    | 112.80   |
| 18 | Jun 2027 | 101.18   | 93.37    | 109.36   | 89.39    | 113.83   |
| 19 | Ju1 2027 | 101.45   | 93.36    | 109.92   | 89.24    | 114.57   |
| 20 | Aug 2027 | 102.70   | 94.30    | 111.52   | 90.02    | 116.36   |
| 21 | Sep 2027 | 104.19   | 95.47    | 113.36   | 91.03    | 118.40   |
| 22 | Oct 2027 | 105.17   | 96.15    | 114.66   | 91.57    | 119.87   |
| 23 | Nov 2027 | 104.53   | 95.31    | 114.24   | 90.63    | 119.59   |
| 24 | Dec 2027 | 104.54   | 95.09    | 114.52   | 90.30    | 120.01   |

```

> forecast_2526_asli <- InvBoxCox(forecast_2526$mean, lambda = 0.42)
> forecast_2526_ts <- ts(
+   forecast_2526_asli,
+   start = c(2025, 1),
+   frequency = 12
+ )
> forecast_2526_ts
      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
2025 97.27955 98.70904 97.22106 93.64847 93.09649 93.89233 93.86842 95.05105
2026 97.00022 98.42735 96.94183 93.37525 92.82420 93.61869 93.59483 94.77546
      Sep      Oct      Nov      Dec
2025 96.74924 97.97294 97.27424 97.40999
2026 96.47080 97.69246 96.99493 97.13045

```

#### Lampiran 24. Code R Plot hasil peramalan dengan interval prediksi

```

> # Plot data aktual
> plot(NTP_data,
+   xlim = c(2009, 2028),
+   ylim = range(c(NTP_data, lower95_ts, upper95_ts)),
+   main = "Peramalan Nilai Tukar Petani (NTP) Tahun 2026–2027",
+   xlab = "Tahun",
+   ylab = "Nilai Tukar Petani",
+   col = "black",
+   lwd = 2)
> # Area interval prediksi 95%
> polygon(c(time(lower95_ts), rev(time(upper95_ts))),
+   c(lower95_ts, rev(upper95_ts)),
+   col = rgb(0,0,1,0.25),
+   border = NA)
> # Garis forecast
> lines(forecast_2627_ts, col = "red", lwd = 2)
> # Garis batas mulai forecast
> abline(v = 2026, lty = 2, col = "gray40")
> # Legenda
> legend("topleft",
+   legend = c("Data Aktual", "Forecast", "Interval Prediksi 95%"),
+   col = c("black", "red", rgb(0,0,1,0.25)),
+   lwd = c(2,2,10),
+   bty = "n")

```



#### Lampiran 25. Code R untuk peramalan NTP Provinsi Bali tahun 2026-2027

```

> # PERAMALAN 2026-2027
> tahun <- floor(time(forecast_2627_ts))
> bulan <- cycle(forecast_2627_ts)
> tabel_forecast_2627 <- data.frame(
+   Tahun_Bulan = paste0(tahun, "-", sprintf("%02d", bulan)),
+   Forecast_NTP = round(as.numeric(forecast_2627_ts), 2)
+ )

```

```
> head(tabel_forecast_2627, 24)
```

|    | Tahun_Bulan | Forecast_NTP |
|----|-------------|--------------|
| 1  | 2026-01     | 102.40       |
| 2  | 2026-02     | 103.70       |
| 3  | 2026-03     | 102.81       |
| 4  | 2026-04     | 99.82        |
| 5  | 2026-05     | 99.74        |
| 6  | 2026-06     | 100.27       |
| 7  | 2026-07     | 100.53       |
| 8  | 2026-08     | 101.78       |
| 9  | 2026-09     | 103.27       |
| 10 | 2026-10     | 104.24       |
| 11 | 2026-11     | 103.60       |
| 12 | 2026-12     | 103.61       |
| 13 | 2027-01     | 103.32       |
| 14 | 2027-02     | 104.63       |
| 15 | 2027-03     | 103.73       |
| 16 | 2027-04     | 100.73       |
| 17 | 2027-05     | 100.65       |
| 18 | 2027-06     | 101.18       |
| 19 | 2027-07     | 101.45       |
| 20 | 2027-08     | 102.70       |
| 21 | 2027-09     | 104.19       |
| 22 | 2027-10     | 105.17       |
| 23 | 2027-11     | 104.53       |
| 24 | 2027-12     | 104.54       |



## RIWAYAT HIDUP



Kadek Rima Suryani lahir di Desa Sulanyah pada tanggal 8 Maret 2004, dari pasangan Bapak Made Darmada dan Ibu Ni Luh Serigati. Penulis berkewarganegaraan Indonesia dan beragama hindu. Saat ini penulis berdomisili di Banjar Dinas Tegal Sari, Desa Sulanyah, Kecamatan Seririt, Kabupaten Buleleng, Provinsi Bali.

Penulis memulai pendidikan formal pada tahun 2010 di Taman Kanak-kanak (TK) Widya Suta Kerti. Selanjutnya, penulis melanjutkan pendidikan di Sekolah Dasar (SD) Negeri 2 Sulanyah dan berhasil menyelesaikan pada tahun 2016. Pada tahun yang sama, penulis melanjutkan pendidikan di Sekolah Menengah Pertama (SMP) Negeri 1 Seririt dan lulus pada tahun 2019. Kemudian penulis melanjutkan pendidikan di Sekolah Menengah Atas (SMA) Negeri 1 Seririt pada tahun 2019 dan menyelesaikan pendidikan pada tahun 2022. Setelah lulus dari SMA, penulis melanjutkan pendidikan ke perguruan tinggi di Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Ganesha pada tahun 2022 dan masih berstatus sebagai mahasiswa hingga penulisan skripsi ini. Dalam masa studi di perguruan tinggi, penulis aktif dalam kegiatan organisasi dan pernah menjadi pengurus HMJ Matematika masa bakti 2022/2023 sebagai anggota Sie Kesenian dan anggota Sie Kesekretariatan masa bakti 2023/2024 hingga 2024/2025.