LAMPIRAN

Lampiran 1. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Badung

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	359405	106201	317167	645498	645834
Februari	305047	98016	203581	472495	586598
Maret	158218	139760	322912	449361	498829
April	34169	150415	283151	417855	584635
Mei	32851	147455	480742	560738	644499
Juni	45074	235791	477519	686304	525045
Juli	67450	45197	419536	613022	528516
Agustus	117332	70656	364591	474715	527520
September	112677	153854	358106	514808	541689
Oktober	164086	293123	398351	591582	535982
November	187589	313615	377934	540667	496263
Desember	363438	506465	628402	639762	630092

Lampiran 2. Data Jumlah Biro Perjalanan Wisata di Kabupaten Badung

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	154	50	155	50	62
Februari	154	50	155	50	62
Maret	154	50	155	50	62
April	154	50	155	50	62
Mei	154	50	155	50	62
Juni	154	50	155	50	62
Juli	154	50	155	50	62
Agustus	154	50	155	50	62
September	154	50	155	50	62
Oktober	154	50	155	50	62
November	154	50	155	50	62
Desember	154	50	155	50	62

Lampiran 3. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 4. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 5. *Code* R Hitung Statistik Deskriptif

```
> # Statistik deskriptif
> Mean <- mean(wisatawan$Badung, na.rm = TRUE)
> Median <- median(wisatawan$Badung, na.rm = TRUE)
> Min <- min(wisatawan$Badung, na.rm = TRUE)
> Max <- max(wisatawan$Badung, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Badung, na.rm = TRUE)
> Observations <- length(wisatawan$Badung)
>
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean   Median   Min    Max Std_Dev Observations
1 369869.8 388142.5 32851 686304  199310           60
```

Lampiran 6. *Code* R Plot Data Kabupaten Badung

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,                # Ketebalan garis
+      xaxt = "n")             # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 7. *Code* R Plot Time Series Data Training

```
> #DATA TRAINING
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Tahun",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 8. *Code* R Plot Time Series Data Testing

```
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 9. *Code* R Plot *Seasonal*

```
> # Plot seasonal dan trend untuk data
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020-2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 10. *Code* R Uji *Seasonal*

```
> qs(wisatawan_datatrain.new)
Test used:  QS

Test statistic:  14.05
P-value:  0.0008903258
```

Lampiran 11. *Code* R Uji Stasioner Dalam Ragam

```
> #stasioner dalam ragam
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.77

> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
 [1] 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.60 0.61 0.62 0.63
[20] 0.64 0.65 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 0.80 0.81 0.82
[39] 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.00 1.01
[58] 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.10 1.11
```

Lampiran 12. *Code* R Transformasi Box-Cox

```
> #tranformasi
> wisatawan_new <- BoxCox(wisatawan_datatrain, lambda = 0.77)
> index <- seq(1:48)
> bc = boxcox(wisatawan_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] 0.58 0.59 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74 0.75 0.76
[20] 0.77 0.78 0.79 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 0.94 0.95
[39] 0.96 0.97 0.98 0.99 1.00 1.01 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.10 1.11 1.12 1.13 1.14
[58] 1.15 1.16 1.17 1.18 1.19 1.20 1.21 1.22 1.23 1.24 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32 1.33
[77] 1.34 1.35 1.36 1.37 1.38 1.39 1.40 1.41 1.42 1.43 1.44 1.45
```

Lampiran 13. *Code* R Uji ADF

```
> #stasioneritas Dalam Rataan
> #Note:data dapat dikatakan stasioner apaliba p-value < alpha
> par(mfrow=c(1,1))
> acf(wisatawan_datatrain, main = "ACF Perjalanan Wisatawan")
> tseries::adf.test(wisatawan_datatrain)

        Augmented Dickey-Fuller Test

data:  wisatawan_datatrain
Dickey-Fuller = -4.3218, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Lampiran 14. *Code* R Menampilkan Plot ACF Dan PACF

```
> acf(wisatawan_datatrain, lag.max = 36)
> Pacf(wisatawan_datatrain, lag.max = 36)
```

Lampiran 15. *Code* R Estimasi Model SARIMA

```
> #Estimasi Model
> model1.da <- Arima(wisatawan_datatrain,
+                    order = c(1,0,0),
+                    seasonal = list(order = c(0,0,1), period = 12),
+                    method = "ML")
>
> summary(model1.da)
Series: wisatawan_datatrain
ARIMA(1,0,0)(0,0,1)[12] with non-zero mean

Coefficients:
         ar1    sma1       mean
      0.8211  0.5802  366969.89
s.e.  0.0788  0.1958   95745.43

sigma^2 = 8018720018:  log likelihood = -616.93
AIC=1241.86   AICc=1242.79   BIC=1249.34

z test of coefficients:

               Estimate    Std. Error z value             Pr(>|z|)
ar1            0.821078      0.078766 10.4243 < 0.00000000000000022 ***
sma1           0.580170      0.195785  2.9633             0.0030435 **
intercept 366969.891970  95745.432038  3.8328             0.0001267 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 16. *Code* R Uji *White Noise*

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.75434, df = 1, p-value = 0.3851
```

Lampiran 17. *Code* R Input Data Variabel Eksogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Badung, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro7
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 18. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] -0.3600765
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = -2.9394, df = 58, p-value = 0.004714
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5625646 -0.1168342
sample estimates:
       cor
-0.3600765
```

Lampiran 19. *Code* R Estimasi Model SARIMAX

```
> modelx3 <- Arima(wisatawan_new,
+                  order = c(1,0,0),                  # non-seasonal (p,d,q)
+                  seasonal = list(order = c(0,0,1),  # seasonal (P,D,Q)
+                                  period = 12),      # musim bulanan
+                  xreg = birotrain.ts,              # variabel eksogen
+                  method = "ML")

> summary(modelx3)
Series: wisatawan_new
Regression with ARIMA(1,0,0)(0,0,1)[12] errors

Coefficients:
         ar1    sma1  intercept     xreg
      0.8232  0.4948  24121.784   0.5194
s.e.  0.0798  0.1749   5675.578  20.7844

sigma^2 = 27351577:  log likelihood = -479.3
AIC=968.59   AICc=970.02   BIC=977.95

z test of coefficients:

              Estimate   Std. Error z value             Pr(>|z|)
ar1           0.823193     0.079761 10.3207 < 0.00000000000000022 ***
sma1          0.494802     0.174937  2.8285             0.004677 **
intercept 24121.783720  5675.578341  4.2501          0.00002137 ***
xreg          0.519390    20.784369  0.0250           0.04930032 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lampiran 20. *Code* R Diagnostik Model

```
> #uji white noise
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.0021451, df = 1, p-value = 0.9631
```

Lampiran 21. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> # Forecast menggunakan model ARIMAX dengan data eksternal liburtest.ts
> ramalanx <- forecast::forecast(modelx3, xreg = birotest.ts)
> # Ambil hasil forecast dalam skala transformasi Box-Cox
> hasil.forc <- ramalanx$mean
> # Inverse Box-Cox ke skala asli
> hasil <- InvBoxCox(hasil.forc, lambda = 0.77)
> # Pastikan nilai forecast tidak negatif (jika ada)
> hasil[hasil < 0] <- 0
> # Bulatkan hasil forecast agar mudah dibaca
> hasil_rounded <- round(hasil)
> # Buat time series hasil forecast dengan frekuensi 12, mulai Jan 2024
> hasil_ts <- ts(hasil_rounded, start = c(2024, 1), frequency = 12)
> # Data aktual testing (pastikan ini sudah ts)
> testing.ts <- wisatawan_time_series[49:60]  # Jan 2024 - Des 2024
> # Buat tabel perbandingan aktual vs forecast
> perbandingan <- data.frame(
+   Bulan = time(testing.ts),
+   Aktual = as.numeric(testing.ts),
+   Forecast = as.numeric(hasil_ts)

> print(perbandingan)
   Bulan Aktual Forecast
1      1 645834   630013
2      2 586598   548720
3      3 498829   494437
4      4 584635   480120
5      5 644499   484049
6      6 525045   542796
7      7 528516   480671
8      8 527520   421913
9      9 541689   449264
10    10 535982   485698
11    11 496263   463438
12    12 630092   460236
```

Lampiran 22. *Code* Tingkat Akurasi Ramalan

```
> akurasi <- accuracy(hasil_ts, testing.ts)
> print(akurasi)
                ME     RMSE      MAE      MPE     MAPE
Test set 67012.25 88094.84 69970.75 11.52224 12.08571
```

Lampiran 23. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> #PREDIKSI 2 TAHUN
> h <- 24  # 2 tahun
> rata_biro <- mean(birotrain.ts, na.rm = TRUE)
> biro_future.ts <- ts(
+   rep(rata_biro, h),
+   start = c(2025,1),
+   frequency = 12
+ )
> forecast_2tahun <- forecast::forecast(
+   modelx3,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> forecast_mean <- InvBoxCox(forecast_2tahun$mean, lambda = 0.77)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 0.77)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 0.77)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 0.77)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 0.77)
> bulan_forecast <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )
> hasil_forecast_2tahun <- data.frame(
+   Bulan = format(bulan_forecast, "%b %Y"),
+   Prediksi = round(as.numeric(forecast_mean)),
+   Lower_80 = round(as.numeric(lower80)),
+   Upper_80 = round(as.numeric(upper80)),
+   Lower_95 = round(as.numeric(lower95)),
+   Upper_95 = round(as.numeric(upper95))
```

```
> print(hasil_forecast_2tahun)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1  Jan 2025   630464   489711   778886   418674   860266
2  Feb 2025   549157   374976   737233   289430   841684
3  Mar 2025   494863   304928   703578   213799   820481
4  Apr 2025   480544   279748   703240   184714   828497
5  May 2025   484473   275310   717337   176904   848535
6  Jun 2025   543232   321905   788000   216701   925497
7  Jul 2025   481095   263808   724351   162496   861731
8  Aug 2025   422324   210932   662314   114778   798629
9  Sep 2025   449681   232892   694487   133310   833240
10 Oct 2025   486122   263378   735949   159842   877150
11 Nov 2025   463858   243432   712281   141822   852969
12 Dec 2025   460656   240225   709319   138784   850199
13 Jan 2026   440776   212184   701676   109269   850174
14 Feb 2026   424564   191577   692898    88641   846150
15 Mar 2026   411324   176097   684171    73834   840403
16 Apr 2026   400499   164278   676046    62983   834138
17 May 2026   391637   155136   668753    54894   827990
18 Jun 2026   384377   147991   662355    48776   822282
19 Jul 2026   378423   142358   656828    44091   817154
20 Aug 2026   373539   137886   652106    40463   812647
21 Sep 2026   369529   134314   648103    37629   808746
22 Oct 2026   366236   131446   644731    35395   805407
23 Nov 2026   363529   129135   641902    33623   802572
24 Dec 2026   361305   127264   639539    32209   800180
```

Lampiran 24. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Gianyar

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	101471	93098	94129	164541	194609
Februari	91579	77010	74219	113038	183120
Maret	69875	87255	105190	116872	188957
April	33204	88916	89082	135610	183778
Mei	29374	93062	117852	139479	182955
Juni	46319	96227	115608	170992	172421
Juli	62616	51806	113907	148847	158025
Agustus	81071	61311	105754	148592	181371
September	76800	78547	98146	145830	171715
Oktober	92334	99217	101159	140330	188355
November	97382	96457	101361	134214	179563
Desember	136192	134905	155592	200796	208266

Lampiran 25. Data Jumlah Biro Perjalanan Wisata di Kabupaten Gianyar

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	23	15	8	94	352
Februari	23	15	8	94	352
Maret	23	15	8	94	352
April	23	15	8	94	352
Mei	23	15	8	94	352
Juni	23	15	8	94	352
Juli	23	15	8	94	352
Agustus	23	15	8	94	352
September	23	15	8	94	352
Oktober	23	15	8	94	352
November	23	15	8	94	352
Desember	23	15	8	94	352

Lampiran 26. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 27. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 28. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Gianyar, na.rm = TRUE)
> Median <- median(wisatawan$Gianyar, na.rm = TRUE)
> Min <- min(wisatawan$Gianyar, na.rm = TRUE)
> Max <- max(wisatawan$Gianyar, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Gianyar, na.rm = TRUE)
> Observations <- length(wisatawan$Gianyar)
```

Lampiran 29. *Code* R Plot Data Kabupaten Gianyar

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,              # Ketebalan garis
+      xaxt = "n")           # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 30. *Code* R Plot Time Series Data Training

```
> #DATA TRAINING
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Jumlah Perjalanan 2020-2024",
+      xlab = "Bulan",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 31. *Code* R Plot Time Series Data Testing

```
> #DATA TESTING
> # Data Testing: ambil data dari Jan 2024 sampai Des 2024
> wisatawan_datatest <- wisatawan_time_series[49:60]
> wisatawan_datatest.new <- ts(wisatawan_datatest, start = c(2024, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Plot Data Testing (Tahun 2024)",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 32. *Code* R Plot *Seasonal*

```
> # Plot seasonal dan trend untuk data
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020-2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 33. *Code* R Uji *Seasonal*

```
> qs(wisatawan_time_series)
Test used:  QS

Test statistic:  19.93
P-value:  4.699554e-05
```

Lampiran 34. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.86
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
 [1] 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 0.52 0.53 0.54 0.55 0.56 0.57
[20] 0.58 0.59 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74 0.75 0.76
[39] 0.77 0.78 0.79 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 0.94 0.95
[58] 0.96 0.97 0.98 0.99 1.00 1.01 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.10 1.11 1.12 1.13 1.14
[77] 1.15 1.16 1.17 1.18 1.19 1.20 1.21 1.22 1.23 1.24 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32 1.33
[96] 1.34 1.35 1.36
```

Lampiran 35. *Code* R Transformasi Box-Cox

```
> wisatawan_new <- BoxCox(wisatawan_datatrain, lambda = 0.86)
> index <- seq(1:48)
> bc = boxcox(wisatawan_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] 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.60 0.61 0.62 0.63
 [20] 0.64 0.65 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 0.80 0.81 0.82
 [39] 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.00 1.01
 [58] 1.02 1.03 1.04 1.05 1.06 1.07 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
 [77] 1.21 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 1.36 1.37 1.38 1.39
 [96] 1.40 1.41 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49 1.50 1.51 1.52 1.53 1.54 1.55 1.56 1.57 1.58
[115] 1.59
```

Lampiran 36. *Code* R Uji ADF

```
> tseries::adf.test(wisatawan_new)

        Augmented Dickey-Fuller Test

data:  wisatawan_new
Dickey-Fuller = -3.6392, Lag order = 3, p-value = 0.03961
alternative hypothesis: stationary
```

Lampiran 37. *Code* R Menampilkan Plot ACF dan PACF

```
> acf(wisatawan_new, lag.max = 36)
> Pacf(wisatawan_new, lag.max = 36)
```

Lampiran 38. *Code* R *Differencing* Musiman

```
> train.diff2 <- diff(wisatawan_new, lag = 12, differences = 1)
> acf(train.diff2)
> Pacf(train.diff2)
```

Lampiran 39. *Code* R Estimasi Model SARIMA

```
> model1.da <- Arima(wisatawan_new,
+                    order = c(1,0,0),
+                    seasonal = list(order = c(0,1,0), period = 12),
+                    method = "ML")
> summary(model1.da)
Series: wisatawan_new
ARIMA(1,0,0)(0,1,0)[12]

Coefficients:
         ar1
      0.7445
s.e.  0.1078

sigma^2 = 20744282:  log likelihood = -354.24
AIC=712.48   AICc=712.84   BIC=715.64

z test of coefficients:

    Estimate Std. Error z value  Pr(>|z|)
ar1  0.74446    0.10776  6.9083 4.905e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 40. *Code* R Diagnostik Model

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.0022539, df = 1, p-value = 0.9621
```

Lampiran 41. *Code* R Input Data Variabel Eksogen

```
> #VARIABEL EXOGEN
> data_biro <- wisatawan$Biro8
> str(data_biro)   # pastikan numeric
 num [1:60] 23 23 23 23 23 23 23 23 23 23 ...
> # Time series dependen (full, hanya untuk sinkronisasi)
> wisatawan_time_series <- ts(
+   wisatawan$Gianyar,
+   start = c(2020,1),
+   frequency = 12
+ )
> # Training: 2020-2023
> biro_train <- data_biro[1:48]
> wisata_train <- wisatawan_time_series[1:48]
> # Testing: 2024
> biro_test <- data_biro[49:60]
> birotrain.ts <- ts(
+   biro_train,
+   start = c(2020,1),
+   frequency = 12
+ )
> birotest.ts <- ts(
+   biro_test,
+   start = c(2024,1),
+   frequency = 12
+ )
```

Lampiran 42. *Code* R Uji Korelasi

```
> # Korelasi Pearson (indikasi awal)
> cor.test(X_log, Y_log, method = "pearson")

        Pearson's product-moment correlation

data:  X_log and Y_log
t = 2.93, df = 46, p-value = 0.005261
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1267235 0.6117773
sample estimates:
      cor
0.3965787
```

Lampiran 43. *Code* R Estimasi Model SARIMAX

```
> model_arimax <- Arima(y_train,
+                       order = c(0,0,1),                  # non-seasonal (p,d,q)
+                       seasonal = list(order = c(0,1,0),  # seasonal (P,D,Q)
+                                       period = 12),      # musim bulanan
+                       xreg = birotrain.ts,              # variabel eksogen
+                       method = "ML")
>
> summary(model_arimax)

Series: y_train
Regression with ARIMA(0,0,1)(0,1,0)[12] errors

Coefficients:
         ma1     xreg
      0.5032  85.1439
s.e.  0.1089  20.9594

Sigma^2 = 20260535:    log likelihood = -353.03
AIC=706.94     AICc=707.69   BIC=711.69
z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ma1   0.50319    0.10888  4.6216 3.809e-06 ***
xreg 85.14388   20.95935  4.0623 4.858e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 44. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.68453, df = 1, p-value = 0.408
```

Lampiran 45. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> forecast_arimax <- forecast::forecast(
+   model_arimax,
+   h = h,
+   xreg = ts(birotest.ts[1:h], start = c(2024,1), frequency = 12),  # pastikan sesuai h
+   level = c(80, 95)
+ )
> # Inverse Box-Cox ke skala asli
> forecast_inv <- InvBoxCox(forecast_arimax$mean, lambda = lambda_bc)
> lower80 <- InvBoxCox(forecast_arimax$lower[,1], lambda = lambda_bc)
> upper80 <- InvBoxCox(forecast_arimax$upper[,1], lambda = lambda_bc)
> lower95 <- InvBoxCox(forecast_arimax$lower[,2], lambda = lambda_bc)
> upper95 <- InvBoxCox(forecast_arimax$upper[,2], lambda = lambda_bc)
> # Buat vector bulan/tahun untuk label
> bulan_forecast <- seq(
+   from = as.Date("2024-01-01"),
+   by = "month",
+   length.out = h
+ )
> # Data frame hasil forecast
> hasil_forecast <- data.frame(
+   Bulan = format(bulan_forecast, "%b %Y"),
+   Aktual = if(h == 12) as.numeric(wisatawan_datatest.new) else NA,  # hanya jika ada dat
a aktual
+   Prediksi = round(as.numeric(forecast_inv), 0),
+   Lower_80 = round(as.numeric(lower80), 0),
+   Upper_80 = round(as.numeric(upper80), 0),
+   Lower_95 = round(as.numeric(lower95), 0),
+   Upper_95 = round(as.numeric(upper95), 0)
+ )
```

```
> # Tampilkan tabel forecast
> print(hasil_forecast)
      Bulan Aktual Prediksi
1   Jan 2024 194609   291986
2   Feb 2024 183120   231455
3   Mar 2024 188957   235689
4   Apr 2024 183778   256263
5   May 2024 182955   260489
6   Jun 2024 172421   294680
7   Jul 2024 158025   270694
8   Aug 2024 181371   270417
9   Sep 2024 171715   267412
10 Oct 2024 188355   261418
11 Nov 2024 179563   254737
12 Dec 2024 208266   326709
```

Lampiran 46. *Code* R Tingkat Akurasi

```
> #CEK AKURASI
> aktual <- as.numeric(wisatawan_datatest.new)
> prediksi <- as.numeric(forecast_inv)
>
> RMSE <- sqrt(mean((aktual - prediksi)^2))
> MAE <- mean(abs(aktual - prediksi))
> MAPE <- mean(abs((aktual - prediksi) / pmax(aktual, 1))) * 100
>
> cat("RMSE:", round(RMSE,2), "\n")
RMSE: 88982.52
> cat("MAE:", round(MAE,2), "\n")
MAE: 85734.46
> cat("MAPE:", round(MAPE,2), "%\n")
MAPE: 47.3 %
```

Lampiran 47. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> h <- 24  # 2 tahun
> rata_biro <- mean(birotrain.ts, na.rm = TRUE)
> biro_future.ts <- ts(
+   rep(rata_biro, h),
+   start = c(2025,1),
+   frequency = 12
+ )
> forecast_2tahun <- forecast::forecast(
+   model_arimax,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> forecast_mean <- InvBoxCox(forecast_2tahun$mean, lambda = 0.86)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 0.86)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 0.86)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 0.86)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 0.86)
> bulan_forecast <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )
> hasil_forecast_2tahun <- data.frame(
+   Bulan = format(bulan_forecast, "%b %Y"),
+   Prediksi = round(as.numeric(forecast_mean)),
+   Lower_80 = round(as.numeric(lower80)),
+   Upper_80 = round(as.numeric(upper80)),
+   Lower_95 = round(as.numeric(lower95)),
+   Upper_95 = round(as.numeric(upper95))
```

```
> print(hasil_forecast_2tahun)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1  Jan 2025   141730   111853   172521    96463   189148
2  Feb 2025    87856    56959   120383    41456   138140
3  Mar 2025    91559    60449   124250    44798   142082
4  Apr 2025   109710    77665   143139    61388   161309
5  May 2025   113468    81248   147037    64857   165272
6  Jun 2025   144165   110699   178765    93519   197481
7  Jul 2025   122577    89956   156473    73308   174859
8  Aug 2025   122329    89718   156216    73077   174598
9  Sep 2025   119641    87147   153435    70579   171773
10 Oct 2025   114294    82037   147895    65622   166143
11 Nov 2025   108355    76374   141733    60139   159878
12 Dec 2025   173316   138880   208742   121109   227853
13 Jan 2026   141730    97278   188254    74794   213581
14 Feb 2026    87856    44769   134248    23957   159816
15 Mar 2026    91559    48146   138175    27062   163838
16 Apr 2026   109710    64882   157331    42693   183433
17 May 2026   113468    68378   161280    45995   187467
18 Jun 2026   144165    97221   193387    73534   220208
19 Jul 2026   122577    76888   170835    54072   197221
20 Aug 2026   122329    76656   170575    53851   196956
21 Sep 2026   119641    74141   167759    51459   194082
22 Oct 2026   114294    69149   162148    46724   188354
23 Nov 2026   108355    63624   155905    41507   181976
24 Dec 2026   173316   124945   223676   100317   251020
```

Lampiran 48. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Tabanan

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	113845	78551	77645	116359	164086
Februari	109573	61354	60627	81594	167861
Maret	82378	69427	81241	85970	163656
April	31704	88544	65863	102470	151254
Mei	30305	97878	103725	114885	150937
Juni	46448	83252	93237	139060	149720
Juli	68327	42899	90365	116597	140243
Agustus	95089	54372	74712	120784	141410
September	81184	67228	78261	114896	169645
Oktober	78457	85750	76000	116805	168824
November	84258	79654	76685	113220	169339
Desember	89687	88157	113503	174777	183274

Lampiran 49. Data Jumlah Biro Perjalanan Wisata di Kabupaten Tabanan

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	4	4	0	2	0
Februari	4	4	0	2	0
Maret	4	4	0	2	0
April	4	4	0	2	0
Mei	4	4	0	2	0
Juni	4	4	0	2	0
Juli	4	4	0	2	0
Agustus	4	4	0	2	0
September	4	4	0	2	0
Oktober	4	4	0	2	0
November	4	4	0	2	0
Desember	4	4	0	2	0

Lampiran 50. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 51. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 52. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Tabanan, na.rm = TRUE)
> Median <- median(wisatawan$Tabanan, na.rm = TRUE)
> Min <- min(wisatawan$Tabanan, na.rm = TRUE)
> Max <- max(wisatawan$Tabanan, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Tabanan, na.rm = TRUE)
> Observations <- length(wisatawan$Tabanan)
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean Median   Min    Max  Std_Dev Observations
1 101964.2  90026 30305 183274 38227.66           60
```

Lampiran 53. *Code* R Plot Data Kabupaten Tabanan

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,               # Ketebalan garis
+      xaxt = "n")            # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 54. *Code* R Plot Time Series Data Training

```
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency
= 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Tahun",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 55. *Code* R Plot Time Series Data Testing

```
> # Data Testing: ambil data dari Jan 2024 sampai Des 2024
> wisatawan_datatest <- wisatawan_time_series[49:60]
> wisatawan_datatest.new <- ts(wisatawan_datatest, start = c(2024, 1), frequency = 12)
>
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 56. *Code* R Plot *Seasonal*

```
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020-2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 57. *Code* R Uji *Seasonal*

```
> qs(wisatawan_datatrain.new)
Test used:  QS

Test statistic:  1.39
P-value:  0.4996315
```

Lampiran 58. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.76
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
  [1] 0.19 0.20 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 0.34 0.35
 [18] 0.36 0.37 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 0.52
 [35] 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 0.66 0.67 0.68 0.69
 [52] 0.70 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82 0.83 0.84 0.85 0.86
 [69] 0.87 0.88 0.89 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.00 1.01 1.02 1.03
 [86] 1.04 1.05 1.06 1.07 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
[103] 1.21 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
```

Lampiran 59. *Code* R Transformasi Box-Cox

```
> wisatawan_new <- BoxCox(wisatawan_datatrain, lambda = 0.76)
> index <- seq(1:48)
> bc = boxcox(wisatawan_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] 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 0.38 0.39 0.40 0.41
 [18] 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58
 [35] 0.59 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74 0.75
 [52] 0.76 0.77 0.78 0.79 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
 [69] 0.93 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 1.08 1.09
 [86] 1.10 1.11 1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19 1.20 1.21 1.22 1.23 1.24 1.25 1.26
[103] 1.27 1.28 1.29 1.30 1.31 1.32 1.33 1.34 1.35 1.36 1.37 1.38 1.39 1.40 1.41 1.42 1.43
[120] 1.44 1.45 1.46 1.47 1.48 1.49 1.50 1.51 1.52 1.53 1.54 1.55 1.56 1.57 1.58 1.59 1.60
[137] 1.61 1.62 1.63 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
[154] 1.78
```

Lampiran 60. *Code* R Uji ADF Sebelum *Differencing*

```
> #Note:data dapat dikatakan stasioner apaliba p-value < alpha
> par(mfrow=c(1,1))
> acf(wisatawan_datatrain, main = "ACF Perjalanan Wisatawan")
> tseries::adf.test(wisatawan_datatrain)

        Augmented Dickey-Fuller Test

data:  wisatawan_datatrain
Dickey-Fuller = -2.1978, Lag order = 3, p-value = 0.4951
alternative hypothesis: stationary
```

Lampiran 61. *Code* R Uji ADF Sesudah *Differencing*

```
> train.diff <- diff(wisatawan_datatrain, differences = 1)
> tseries::adf.test(train.diff)

        Augmented Dickey-Fuller Test

data:  train.diff
Dickey-Fuller = -6.0741, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Lampiran 62. *Code* R Menampilkan Plot ACF Sebelum *Differencing*

```
> acf(wisatawan_datatrain, main = "ACF Perjalanan Wisatawan")
```

Lampiran 63. *Code* R Menampilkan Plot ACF dan PACF Sesudah *Differencing*

```
> acf(train.diff, lag.max = 15)
> Pacf(train.diff, lag.max = 15)
```

Lampiran 64. *Code* R Estimasi Model ARIMA

```
> model1.da=Arima(train.diff, order=c(2,1,2),method="ML")
> summary(model1.da)
Series: train.diff
ARIMA(2,1,2)

Coefficients:
         ar1      ar2      ma1     ma2
      0.7437  -0.4495  -1.9536  1.0000
s.e.  0.1483   0.1563   0.1011  0.1021

sigma^2 = 318503163:  log likelihood = -517.82
AIC=1045.65   AICc=1047.15   BIC=1054.79

z test of coefficients:

    Estimate Std. Error  z value             Pr(>|z|)
ar1  0.74371    0.14825   5.0164          0.0000005264 ***
ar2 -0.44952    0.15632  -2.8757              0.004031 **
ma1 -1.95362    0.10110 -19.3246 < 0.00000000000000022 ***
ma2  1.00000    0.10207   9.7967 < 0.00000000000000022 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 65. *Code* R Uji *White Noise*

```
> #uji white noise
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 1.1775, df = 1, p-value = 0.2779
```

Lampiran 66. *Code* R Input Variabel Exogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Tabanan, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro6
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
>
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 67. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] -0.5425789
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = -4.9192, df = 58, p-value = 0.000007522
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.7000542 -0.3347780
sample estimates:
       cor
-0.5425789
```

Lampiran 68. *Code* R Estimasi Model ARIMAX

```
> modelx3 <- Arima(wisatawan_new, order = c(2,1,2),
+                  xreg = birotrain.ts,
+                  method = "ML")
> summary(modelx3)

Series: wisatawan_new
Regression with ARIMA(2,1,2) errors

Coefficients:
         ar1      ar2      ma1     ma2      xreg
      1.1573  -0.8685  -1.3439  0.6655   -2.5441
s.e.  0.1400   0.1364   0.1961  0.2252  206.4221

sigma^2 = 1441094:  log likelihood = -398.36
AIC=808.72   AICc=810.82   BIC=819.82

z test of coefficients:

      Estimate Std. Error z value             Pr(>|z|)
ar1    1.15735    0.13997  8.2687 < 0.00000000000000022 ***
ar2   -0.86845    0.13643 -6.3656      0.000000000194510 ***
ma1   -1.34391    0.19615 -6.8515      0.000000000007308 ***
ma2    0.66547    0.22524  2.9545                0.003132 **
xreg  -2.54405  206.42205 -0.0123                0.009901 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 69. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.041374, df = 1, p-value = 0.8388
```

Lampiran 70. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> # Forecast menggunakan model ARIMAX dengan data eksternal liburtest.ts
> ramalanx <- forecast::forecast(modelx3, xreg = birotest.ts)
> # Ambil hasil forecast dalam skala transformasi Box-Cox
> hasil.forc <- ramalanx$mean
> # Inverse Box-Cox ke skala asli
> hasil <- InvBoxCox(hasil.forc, lambda = 0.76)
> # Pastikan nilai forecast tidak negatif (jika ada)
> hasil[hasil < 0] <- 0
> # Bulatkan hasil forecast agar mudah dibaca
> hasil_rounded <- round(hasil)
> # Buat time series hasil forecast dengan frekuensi 12, mulai Jan 2024
> hasil_ts <- ts(hasil_rounded, start = c(2024, 1), frequency = 12)
> # Data aktual testing (pastikan ini sudah ts)
> testing.ts <- wisatawan_time_series[49:60]  # Jan 2024 - Des 2024
> # Buat tabel perbandingan aktual vs forecast
> perbandingan <- data.frame(
+   Bulan = time(testing.ts),
+   Aktual = as.numeric(testing.ts),
+   Forecast = as.numeric(hasil_ts)
+ )
```

```
> print(perbandingan)
   Bulan Aktual Forecast
1      1 164086   170799
2      2 167861   148312
3      3 163656   126549
4      4 151254   120362
5      5 150937   131680
6      6 149720   150689
7      7 140243   162893
8      8 141410   160067
9      9 169645   146297
10    10 168824   133081
11    11 169339   129533
12    12 183274   136762
```

Lampiran 71. *Code* R Tingkat Akurasi

```
> akurasi <- accuracy(hasil_ts, testing.ts)
> print(akurasi)
                ME     RMSE      MAE      MPE     MAPE
Test set 16935.42 28228.13 25100.25 9.769934 15.45034
```

Lampiran 72. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> forecast_2tahun <- forecast::forecast(
+   modelx3,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> forecast_mean <- InvBoxCox(forecast_2tahun$mean, lambda = 0.76)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 0.76)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 0.76)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 0.76)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 0.76)
> bulan_forecast <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )
> hasil_forecast_2tahun <- data.frame(
+   Bulan = format(bulan_forecast, "%b %Y"),
+   Prediksi = round(as.numeric(forecast_mean)),
+   Lower_80 = round(as.numeric(lower80)),
+   Upper_80 = round(as.numeric(upper80)),
+   Lower_95 = round(as.numeric(lower95)),
+   Upper_95 = round(as.numeric(upper95))
+ )
```

```
> print(hasil_forecast_2tahun)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1  Jan 2025   170684   143520   198931   129612   214293
2  Feb 2025   148201   114667   183674    97790   203165
3  Mar 2025   126442    92876   162314    76172   182136
4  Apr 2025   120256    87095   155790    70644   175455
5  May 2025   131573    97565   167849    80607   187874
6  Jun 2025   150578   114320   189077    96146   210275
7  Jul 2025   162780   122052   206131   101693   230034
8  Aug 2025   159954   114605   208649    92160   235627
9  Sep 2025   146187    98895   197516    75791   226111
10 Oct 2025   132973    85853   184548    63083   213401
11 Nov 2025   129426    82377   181056    59718   209975
12 Dec 2025   136653    88589   189225    65341   218624
13 Jan 2026   148297    98202   202881    73849   233350
14 Feb 2026   155503   102691   213076    77035   245221
15 Mar 2026   153566    98425   214000    71826   247830
16 Apr 2026   145109    89012   207066    62236   241875
17 May 2026   137101    80986   199476    54453   234624
18 Jun 2026   135094    78743   197874    52187   233286
19 Jul 2026   139678    82291   203495    55174   239463
20 Aug 2026   146799    87712   212353    59692   249259
21 Sep 2026   151061    90066   218756    61157   256873
22 Oct 2026   149747    87286   219334    57850   258584
23 Nov 2026   144544    81424   215264    51938   255253
24 Dec 2026   139698    76372   210995    47021   251395
```

Lampiran 73. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Buleleng

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	133942	102220	110011	195260	191321
Februari	123878	89975	92146	123858	242516
Maret	72912	98297	137692	130559	197955
April	34001	130675	113418	148705	196077
Mei	38716	122319	148363	153839	184559
Juni	68611	117441	146358	208691	177433
Juli	108004	66164	138203	195750	183148
Agustus	119463	84063	119263	207345	158756
September	119215	109616	125942	167301	222947
Oktober	107796	125129	129433	176861	161756
November	114208	134561	122357	170719	171298
Desember	126974	128499	162345	185093	182696

Lampiran 74. Data Jumlah Biro Perjalanan Wisata di Kabupaten Buleleng

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	1	4	1	0	0
Februari	1	4	1	0	0
Maret	1	4	1	0	0
April	1	4	1	0	0
Mei	1	4	1	0	0
Juni	1	4	1	0	0
Juli	1	4	1	0	0
Agustus	1	4	1	0	0
September	1	4	1	0	0
Oktober	1	4	1	0	0
November	1	4	1	0	0
Desember	1	4	1	0	0

Lampiran 75. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 76. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 77. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Buleleng, na.rm = TRUE)
> Median <- median(wisatawan$Buleleng, na.rm = TRUE)
> Min <- min(wisatawan$Buleleng, na.rm = TRUE)
> Max <- max(wisatawan$Buleleng, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Buleleng, na.rm = TRUE)
> Observations <- length(wisatawan$Buleleng)
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean Median   Min    Max  Std_Dev Observations
1 139277.5 130617 34001 242516 43889.71           60
```

Lampiran 78. *Code* R Plot Data Kabupaten Buleleng

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,               # Ketebalan garis
+      xaxt = "n")            # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 79. *Code* R Plot Time Series Data Training

```
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Tahun",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 80. *Code* R Plot Time Series Data Testing

```
> # Data Testing: ambil data dari Jan 2024 sampai Des 2024
> wisatawan_datatest <- wisatawan_time_series[49:60]
> wisatawan_datatest.new <- ts(wisatawan_datatest, start = c(2024, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 81. *Code* R Plot *Seasonal*

```
> # Plot seasonal dan trend untuk data
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020-2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 82. *Code* R Uji *Seasonal*

```
> qs(wisatawan_datatrain.new)
Test used:  QS

Test statistic:  0
P-value:  1
```

Lampiran 83. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 1.16
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
  [1] 0.65 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 0.80 0.81
 [18] 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98
 [35] 0.99 1.00 1.01 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.10 1.11 1.12 1.13 1.14 1.15
 [52] 1.16 1.17 1.18 1.19 1.20 1.21 1.22 1.23 1.24 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32
 [69] 1.33 1.34 1.35 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
 [86] 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 1.64 1.65 1.66
[103] 1.67 1.68 1.69 1.70 1.71
```

Lampiran 84. *Code* R Transformasi Box-Cox

```
> wisatawan_new <- BoxCox(wisatawan_datatrain, lambda = 1.16)
> index <- seq(1:48)
> bc = boxcox(wisatawan_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] 0.56 0.57 0.58 0.59 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72
[18] 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89
[35] 0.90 0.91 0.92 0.93 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
[52] 1.07 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 1.22 1.23
[69] 1.24 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32 1.33 1.34 1.35 1.36 1.37 1.38 1.39 1.40
[86] 1.41 1.42 1.43 1.44 1.45 1.46 1.47
```

Lampiran 85. *Code* R Uji ADF

```
> tseries::adf.test(wisatawan_new)

        Augmented Dickey-Fuller Test

data:  wisatawan_new
Dickey-Fuller = -3.8665, Lag order = 3, p-value = 0.02298
alternative hypothesis: stationary
```

Lampiran 86. *Code* R Menampilkan Plot ACF dan PACF

```
> acf(wisatawan_new, lag.max = 15)
> pacf(wisatawan_new, lag.max = 15)
```

Lampiran 87. *Code* R Estimasi Model ARIMA

```
> summary(model1.da)
Series: wisatawan_new
ARIMA(3,0,3) with non-zero mean

Coefficients:
         ar1      ar2     ar3      ma1     ma2      ma3      mean
      1.6315  -1.5301  0.8533  -1.0290  1.2453  -0.6523  752374.8
s.e.  0.1140   0.1321  0.0848   0.1943  0.1563   0.2178  173627.3

sigma^2 = 21459599651:  log likelihood = -638.59
AIC=1293.17   AICc=1296.86   BIC=1308.14

z test of coefficients:

               Estimate    Std. Error  z value             Pr(>|z|)
ar1            1.631538      0.114011  14.3104 < 0.00000000000000022 ***
ar2           -1.530096      0.132094 -11.5834 < 0.00000000000000022 ***
ar3            0.853261      0.084779  10.0645 < 0.00000000000000022 ***
ma1           -1.028972      0.194314  -5.2954  0.000000118742250907 ***
ma2            1.245312      0.156272   7.9689  0.000000000000001601 ***
ma3           -0.652301      0.217772  -2.9953              0.002741 **
intercept 752374.803275 173627.349163   4.3333  0.000014690836853023 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 88. *Code* R Uji *White Noise*

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.45291, df = 1, p-value = 0.501
```

Lampiran 89. *Code* R Input Data Variabel Eksogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Buleleng, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro1
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 90. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] -0.5416786
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = -4.9077, df = 58, p-value = 0.000007841
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6994034 -0.3336453
sample estimates:
       cor
-0.5416786
```

Lampiran 91. *Code* R Estimasi Model ARIMAX

```
> #MODEL FIX
> modelx3 <- Arima(wisatawan_new, order = c(3,0,3),
+                  xreg = birotrain.ts,
+                  method = "ML")

> summary(modelx3)
Series: wisatawan_new
Regression with ARIMA(3,0,3) errors

Coefficients:
         ar1      ar2     ar3      ma1     ma2      ma3  intercept       xreg
      1.6237  -1.4947  0.8158  -1.0285  1.2383  -0.6757   785638.6  -34106.27
s.e.  0.1283   0.1541  0.1061   0.1908  0.1358   0.2045   142812.8   25761.41

sigma^2 = 21319285975:  log likelihood = -637.72
AIC=1293.43   AICc=1298.17   BIC=1310.27

z test of coefficients:

              Estimate   Std. Error z value             Pr(>|z|)
ar1            1.62371      0.12826 12.6598 < 0.00000000000000022 ***
ar2           -1.49465      0.15414 -9.6965 < 0.00000000000000022 ***
ar3            0.81579      0.10606  7.6921   0.00000000000001447 ***
ma1           -1.02855      0.19081 -5.3905   0.00000007024329435 ***
ma2            1.23829      0.13576  9.1213 < 0.00000000000000022 ***
ma3           -0.67575      0.20453 -3.3039              0.0009535 ***
intercept 785638.55449 142812.76836  5.5012   0.00000003772606719 ***
xreg      -34106.26899  25761.40878 -1.3239              0.0185528 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 92. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.42224, df = 1, p-value = 0.5158
```

Lampiran 93. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> # Forecast menggunakan model ARIMAX dengan data eksternal liburtest.ts
> ramalanx <- forecast::forecast(modelx3, xreg = birotest.ts)
> # Ambil hasil forecast dalam skala transformasi Box-Cox
> hasil.forc <- ramalanx$mean
> # Inverse Box-Cox ke skala asli
> hasil <- InvBoxCox(hasil.forc, lambda = 1.16)
> # Pastikan nilai forecast tidak negatif (jika ada)
> hasil[hasil < 0] <- 0
> # Bulatkan hasil forecast agar mudah dibaca
> hasil_rounded <- round(hasil)
> # Buat time series hasil forecast dengan frekuensi 12, mulai Jan 2024
> hasil_ts <- ts(hasil_rounded, start = c(2024, 1), frequency = 12)
> # Data aktual testing (pastikan ini sudah ts)
> testing.ts <- wisatawan_time_series[49:60]  # Jan 2024 - Des 2024
> # Buat tabel perbandingan aktual vs forecast
> perbandingan <- data.frame(
+   Bulan = time(testing.ts),
+   Aktual = as.numeric(testing.ts),
+   Forecast = as.numeric(hasil_ts)
+ )
```

```
> print(perbandingan)
   Bulan Aktual Forecast
1      1 191321   171054
2      2 242516   164474
3      3 197955   165304
4      4 196077   169717
5      5 184559   170273
6      6 177433   165258
7      7 183148   159857
8      8 158756   159072
9      9 222947   161770
10    10 161756   162905
11    11 171298   160081
12    12 182696   155984
```

Lampiran 94. *Code* R Tingkat Akurasi

```
> akurasi <- accuracy(hasil_ts, testing.ts)
> print(akurasi)
                ME     RMSE      MAE      MPE     MAPE
Test set 25392.75 33838.38 25636.92 12.31088 12.46245
```

Lampiran 95. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> # PREDIKSI 2 tahun = 24 bulan
> h <- 24
> # Asumsi variabel libur 2025-2026 = sama dengan 2024
> biro_future <- rep(birotest.ts, length.out = h)
> biro_future.ts <- ts(
+   biro_future,
+   start = c(2025,1),
+   frequency = 12
+ )
> # Forecast ARIMAX
> forecast_2tahun <- forecast::forecast(
+   modelx3,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> # Inverse Box-Cox
> forecast_inv <- InvBoxCox(forecast_2tahun$mean, lambda = 1.16)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 1.16)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 1.16)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 1.16)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 1.16)
> # Buat label bulan
> bulan <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )

> # Tabel hasil
> hasil_2025_2026 <- data.frame(
+   Bulan = format(bulan, "%b %Y"),
+   Prediksi = round(forecast_inv),
+   Lower_80 = round(lower80),
+   Upper_80 = round(upper80),
+   Lower_95 = round(lower95),
+   Upper_95 = round(upper95)
+ )

> print(hasil_2025_2026)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1  Jan 2025   171054   142923   198460   127675   212720
2  Feb 2025   164474   131320   196582   113231   213236
3  Mar 2025   165304   126891   202326   105807   221479
4  Apr 2025   169717   130074   207916   108309   227677
5  May 2025   170273   130598   208506   108818   228286
6  Jun 2025   165258   125206   203798   103177   223721
7  Jul 2025   159857   117985   200021    94860   220753
8  Aug 2025   159072   114615   201597    89966   223518
9  Sep 2025   161770   116344   205213    91149   227606
10 Oct 2025   162905   117414   206421    92192   228853
11 Nov 2025   160081   114362   203770    88978   226281
12 Dec 2025   155984   109438   200371    83516   223221
13 Jan 2026   154491   106596   200081    79846   223530
14 Feb 2026   155892   107286   202143    80128   225929
15 Mar 2026   157042   108344   203394    81146   227235
16 Apr 2026   155604   106765   202062    79462   225951
17 May 2026   152683   103415   199483    75809   223533
18 Jun 2026   151032   101036   198459    72961   222818
19 Jul 2026   151549   101064   199423    72698   224008
20 Aug 2026   152466   101880   200447    73467   225089
21 Sep 2026   151838   101163   199887    72687   224561
22 Oct 2026   149865    98945   198099    70284   222857
23 Nov 2026   148348    97021   196919    68081   221840
24 Dec 2026   148326    96675   197185    67534   222250
```

Lampiran 96. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Karangasem

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	104218	91547	95750	142746	139377
Februari	89108	63970	67433	78976	188408
Maret	57654	78087	179434	90402	184928
April	33550	123795	129477	279421	178377
Mei	29744	93334	99990	121184	154573
Juni	52528	90882	101597	184003	159119
Juli	68630	58102	108801	201417	137286
Agustus	80800	79838	113370	178559	154223
September	84298	89286	103122	166098	189535
Oktober	83574	100746	102779	170961	160179
November	83853	93511	98215	157684	171284
Desember	95079	103741	119371	145281	155654

Lampiran 97. Data Jumlah Biro Perjalanan Wisata di Kabupaten Karangasem

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	2	2	1	0	0
Februari	2	2	1	0	0
Maret	2	2	1	0	0
April	2	2	1	0	0
Mei	2	2	1	0	0
Juni	2	2	1	0	0
Juli	2	2	1	0	0
Agustus	2	2	1	0	0
September	2	2	1	0	0
Oktober	2	2	1	0	0
November	2	2	1	0	0
Desember	2	2	1	0	0

Lampiran 98. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 99. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 100. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Karangasem, na.rm = TRUE)
> Median <- median(wisatawan$Karangasem, na.rm = TRUE)
> Min <- min(wisatawan$Karangasem, na.rm = TRUE)
> Max <- max(wisatawan$Karangasem, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Karangasem, na.rm = TRUE)
> Observations <- length(wisatawan$Karangasem)
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean   Median   Min    Max  Std_Dev Observations
1 118981.5 103431.5 29744 279421 47745.84           60
```

Lampiran 101. *Code* R Plot Data Kabupaten Karangasem

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,               # Ketebalan garis
+      xaxt = "n")            # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 102. *Code* R Plot Time Series Data Training

```
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Tahun",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 103. *Code* R Plot Time Series Data Testing

```
> wisatawan_datatest <- wisatawan_time_series[49:60]
> wisatawan_datatest.new <- ts(wisatawan_datatest, start = c(2024, 1), frequency = 12)
>
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 104. *Code* R Plot *Seasonal*

```
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020-2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 105. *Code* R Uji *Seasonal*

```
> qs(wisatawan_time_series)
Test used:  QS

Test statistic:  0
P-value:  1
```

Lampiran 106. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.21
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
 [1] -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 -0.04
[15] -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  0.10
[29]  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  0.24
[43]  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  0.38
[57]  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  0.52
[71]  0.53  0.54  0.55  0.56  0.57  0.58  0.59  0.60
```

Lampiran 107. *Code* R Transformasi Box-Cox

```
> wisatawan_new <- BoxCox(wisatawan_datatrain, lambda = 0.21)
> index <- seq(1:48)
> bc = boxcox(wisatawan_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] -0.66 -0.65 -0.64 -0.63 -0.62 -0.61 -0.60 -0.59 -0.58 -0.57 -0.56 -0.55 -0.54 -0.53
 [15] -0.52 -0.51 -0.50 -0.49 -0.48 -0.47 -0.46 -0.45 -0.44 -0.43 -0.42 -0.41 -0.40 -0.39
 [29] -0.38 -0.37 -0.36 -0.35 -0.34 -0.33 -0.32 -0.31 -0.30 -0.29 -0.28 -0.27 -0.26 -0.25
 [43] -0.24 -0.23 -0.22 -0.21 -0.20 -0.19 -0.18 -0.17 -0.16 -0.15 -0.14 -0.13 -0.12 -0.11
 [57] -0.10 -0.09 -0.08 -0.07 -0.06 -0.05 -0.04 -0.03 -0.02 -0.01  0.00  0.01  0.02  0.03
 [71]  0.04  0.05  0.06  0.07  0.08  0.09  0.10  0.11  0.12  0.13  0.14  0.15  0.16  0.17
 [85]  0.18  0.19  0.20  0.21  0.22  0.23  0.24  0.25  0.26  0.27  0.28  0.29  0.30  0.31
 [99]  0.32  0.33  0.34  0.35  0.36  0.37  0.38  0.39  0.40  0.41  0.42  0.43  0.44  0.45
[113]  0.46  0.47  0.48  0.49  0.50  0.51  0.52  0.53  0.54  0.55  0.56  0.57  0.58  0.59
[127]  0.60  0.61  0.62  0.63  0.64  0.65  0.66  0.67  0.68  0.69  0.70  0.71  0.72  0.73
[141]  0.74  0.75  0.76  0.77  0.78  0.79  0.80  0.81  0.82  0.83  0.84  0.85  0.86  0.87
[155]  0.88  0.89  0.90  0.91  0.92  0.93  0.94  0.95  0.96  0.97  0.98  0.99  1.00  1.01
[169]  1.02  1.03  1.04  1.05  1.06  1.07  1.08  1.09  1.10  1.11  1.12  1.13  1.14  1.15
[183]  1.16  1.17  1.18  1.19  1.20  1.21  1.22  1.23  1.24  1.25  1.26  1.27  1.28  1.29
[197]  1.30  1.31  1.32  1.33  1.34  1.35  1.36  1.37  1.38  1.39  1.40  1.41  1.42  1.43
[211]  1.44  1.45  1.46  1.47  1.48  1.49  1.50  1.51  1.52  1.53  1.54  1.55  1.56  1.57
[225]  1.58  1.59  1.60  1.61  1.62  1.63  1.64  1.65  1.66  1.67  1.68  1.69  1.70  1.71
[239]  1.72  1.73  1.74  1.75  1.76  1.77  1.78  1.79  1.80  1.81  1.82  1.83  1.84  1.85
[253]  1.86  1.87  1.88  1.89  1.90  1.91  1.92  1.93  1.94  1.95  1.96  1.97  1.98  1.99
[267]  2.00  2.01  2.02  2.03  2.04  2.05  2.06  2.07  2.08  2.09  2.10  2.11  2.12  2.13
[281]  2.14  2.15  2.16  2.17  2.18  2.19  2.20  2.21  2.22  2.23  2.24  2.25  2.26  2.27
[295]  2.28  2.29  2.30  2.31  2.32  2.33  2.34  2.35  2.36  2.37  2.38  2.39  2.40  2.41
[309]  2.42  2.43  2.44  2.45  2.46  2.47  2.48  2.49  2.50  2.51  2.52  2.53  2.54  2.55
[323]  2.56  2.57  2.58  2.59  2.60  2.61  2.62  2.63  2.64  2.65  2.66  2.67  2.68  2.69
[337]  2.70
```

Lampiran 108. *Code* R Uji ADF

```
> tseries::adf.test(wisatawan_new)

        Augmented Dickey-Fuller Test

data:  wisatawan_new
Dickey-Fuller = -3.6876, Lag order = 3, p-value = 0.03566
alternative hypothesis: stationary
```

Lampiran 109. *Code* R Menampilkan Plot ACF dan PACF

```
> acf(wisatawan_new, lag.max = 15)
> pacf(wisatawan_new, lag.max = 15)
```

Lampiran 110. *Code* R Estimasi Model ARIMA

```
> summary(model1.da)
Series: wisatawan_new
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1     mean
      0.6263  48.8998
s.e.  0.1101   1.3435

sigma^2 = 13.48:  log likelihood = -129.76
AIC=265.52   AICc=266.06   BIC=271.13

z test of coefficients:

           Estimate Std. Error z value  Pr(>|z|)
ar1         0.62626    0.11007  5.6899 1.271e-08 ***
intercept 48.89975    1.34351 36.3971 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 111. *Code* R Uji *White Noise*

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.42036, df = 1, p-value = 0.5168
```

Lampiran 112. *Code* R Input Data Variabel Exogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Karangasem, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro3
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
>
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 113. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] -0.7713112
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = -9.2294, df = 58, p-value = 0.0000000000005559
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8573246 -0.6433994
sample estimates:
       cor
-0.7713112
```

Lampiran 114. *Code* R Estimasi Model ARIMAX

```
> #MODEL FIX
> modelx3 <- Arima(wisatawan_new, order = c(0,0,1),
+                  xreg = birotrain.ts,
+                  method = "ML")

> summary(modelx3)
Series: wisatawan_new
Regression with ARIMA(0,0,1) errors

Coefficients:
         ma1  intercept     xreg
      0.3316    53.5279  -3.8128
s.e.  0.1427     1.0922   0.7262

sigma^2 = 10.87:  log likelihood = -123.88
AIC=255.75   AICc=256.68   BIC=263.24

z test of coefficients:

           Estimate Std. Error z value  Pr(>|z|)
ma1         0.33160    0.14270  2.3238   0.02014 *
intercept 53.52793    1.09220 49.0094 < 2.2e-16 ***
xreg      -3.81283    0.72617 -5.2506 1.516e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 115. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.0098785, df = 1, p-value = 0.9208
```

Lampiran 116. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> # Forecast menggunakan model ARIMAX dengan data eksternal liburtest.ts
> ramalanx <- forecast::forecast(modelx3, xreg = birotest.ts)
> # Ambil hasil forecast dalam skala transformasi Box-Cox
> hasil.forc <- ramalanx$mean
> # Inverse Box-Cox ke skala asli
> hasil <- InvBoxCox(hasil.forc, lambda = 0.21)
> # Pastikan nilai forecast tidak negatif (jika ada)
> hasil[hasil < 0] <- 0
> # Bulatkan hasil forecast agar mudah dibaca
> hasil_rounded <- round(hasil)
> # Buat time series hasil forecast dengan frekuensi 12, mulai Jan 2024
> hasil_ts <- ts(hasil_rounded, start = c(2024, 1), frequency = 12)
> # Data aktual testing (pastikan ini sudah ts)
> testing.ts <- wisatawan_time_series[49:60]  # Jan 2024 - Des 2024
> # Buat tabel perbandingan aktual vs forecast
> perbandingan <- data.frame(
+   Bulan = time(testing.ts),
+   Aktual = as.numeric(testing.ts),
+   Forecast = as.numeric(hasil_ts)
+ )
```

```
> print(perbandingan)
   Bulan Aktual Forecast
1      1 139377   149227
2      2 188408   151376
3      3 184928   151376
4      4 178377   151376
5      5 154573   151376
6      6 159119   151376
7      7 137286   151376
8      8 154223   151376
9      9 189535   151376
10    10 160179   151376
11    11 171284   151376
12    12 155654   151376
```

Lampiran 117. *Code* R Tingkat Akurasi

```
> akurasi <- accuracy(hasil_ts, testing.ts)
> print(akurasi)
                ME     RMSE   MAE      MPE     MAPE
Test set 13215 21502.37 17205 7.032124 9.920526
```

Lampiran 118. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> # PREDIKSI 2 tahun = 24 bulan
> h <- 24
> # Asumsi variabel libur 2025-2026 = sama dengan 2024
> biro_future <- rep(birotest.ts, length.out = h)
> biro_future.ts <- ts(
+   biro_future,
+   start = c(2025,1),
+   frequency = 12
+ )
> # Forecast ARIMAX
> forecast_2tahun <- forecast::forecast(
+   modelx3,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> # Inverse Box-Cox
> forecast_inv <- InvBoxCox(forecast_2tahun$mean, lambda = 0.21)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 0.21)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 0.21)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 0.21)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 0.21)
> # Buat label bulan
> bulan <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )

> # Tabel hasil
> hasil_2025_2026 <- data.frame(
+   Bulan = format(bulan, "%b %Y"),
+   Prediksi = round(forecast_inv),
+   Lower_80 = round(lower80),
+   Upper_80 = round(upper80),
+   Lower_95 = round(lower95),
+   Upper_95 = round(upper95)
+ )

> print(hasil_2025_2026)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1  Jan 2025   149227   104175   208436    85135   246527
2  Feb 2025   151376   103702   214899    83800   256139
3  Mar 2025   151376   103702   214899    83800   256139
4  Apr 2025   151376   103702   214899    83800   256139
5  May 2025   151376   103702   214899    83800   256139
6  Jun 2025   151376   103702   214899    83800   256139
7  Jul 2025   151376   103702   214899    83800   256139
8  Aug 2025   151376   103702   214899    83800   256139
9  Sep 2025   151376   103702   214899    83800   256139
10 Oct 2025   151376   103702   214899    83800   256139
11 Nov 2025   151376   103702   214899    83800   256139
12 Dec 2025   151376   103702   214899    83800   256139
13 Jan 2026   151376   103702   214899    83800   256139
14 Feb 2026   151376   103702   214899    83800   256139
15 Mar 2026   151376   103702   214899    83800   256139
16 Apr 2026   151376   103702   214899    83800   256139
17 May 2026   151376   103702   214899    83800   256139
18 Jun 2026   151376   103702   214899    83800   256139
19 Jul 2026   151376   103702   214899    83800   256139
20 Aug 2026   151376   103702   214899    83800   256139
21 Sep 2026   151376   103702   214899    83800   256139
22 Oct 2026   151376   103702   214899    83800   256139
23 Nov 2026   151376   103702   214899    83800   256139
24 Dec 2026   151376   103702   214899    83800   256139
```

Lampiran 119. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Jembrana

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	64488	50540	58706	106249	82950
Februari	65352	45069	48733	71574	101889
Maret	44490	55506	68703	71884	90059
April	23690	66506	60133	82318	88915
Mei	25509	68679	73367	88195	75606
Juni	38492	61906	78724	105507	78410
Juli	48422	36483	73816	104866	72273
Agustus	53100	43858	64130	92477	63120
September	55455	56530	67888	78054	80591
Oktober	51856	63079	68863	84577	66130
November	55510	63211	61106	79194	69280
Desember	64982	64344	103877	73476	77548

Lampiran 120. Data Jumlah Biro Perjalanan Wisata di Kabupaten Jembrana

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	0	0	0	5	0
Februari	0	0	0	5	0
Maret	0	0	0	5	0
April	0	0	0	5	0
Mei	0	0	0	5	0
Juni	0	0	0	5	0
Juli	0	0	0	5	0
Agustus	0	0	0	5	0
September	0	0	0	5	0
Oktober	0	0	0	5	0
November	0	0	0	5	0
Desember	0	0	0	5	0

Lampiran 121. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 122. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 123. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Jembrana, na.rm = TRUE)
> Median <- median(wisatawan$Jembrana, na.rm = TRUE)
> Min <- min(wisatawan$Jembrana, na.rm = TRUE)
> Max <- max(wisatawan$Jembrana, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Jembrana, na.rm = TRUE)
> Observations <- length(wisatawan$Jembrana)
>
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean Median   Min    Max  Std_Dev Observations
1 68004.08  67197 23690 106249 18412.39           60
```

Lampiran 124. *Code* R Plot Data Kabupaten Jembrana

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,               # Ketebalan garis
+      xaxt = "n")            # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 125. *Code* R Plot Time Series Data Training

```
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Tahun",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 126. *Code* R Plot Time Series Data Testing

```
> # Data Testing: ambil data dari Jan 2024 sampai Des 2024
> wisatawan_datatest <- wisatawan_time_series[49:60]
> wisatawan_datatest.new <- ts(wisatawan_datatest, start = c(2024, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 127. *Code* R Plot *Seasonal*

```
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020-2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 128. *Code* R Uji *Seasonal*

```
> qs(wisatawan_time_series)
Test used:  QS

Test statistic:  0
P-value:  1
```

Lampiran 129. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.81
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
  [1] 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.40 0.41 0.42
 [18] 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59
 [35] 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74 0.75 0.76
 [52] 0.77 0.78 0.79 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
 [69] 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 1.08 1.09 1.10
 [86] 1.11 1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19 1.20 1.21 1.22 1.23 1.24 1.25 1.26 1.27
[103] 1.28 1.29 1.30 1.31 1.32 1.33 1.34 1.35 1.36 1.37 1.38 1.39 1.40
> # Tambahkan garis vertikal di lambda terbaik
> abline(v = lambda, col = "red", lty = 2)
```

Lampiran 130. *Code* R Transformasi Box-Cox

```
> wisatawan_new <- BoxCox(wisatawan_datatrain, lambda = 0.81)
> index <- seq(1:48)
> bc = boxcox(wisatawan_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] 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.40 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48
 [18] 0.49 0.50 0.51 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
 [35] 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 0.80 0.81 0.82
 [52] 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99
 [69] 1.00 1.01 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.10 1.11 1.12 1.13 1.14 1.15 1.16
 [86] 1.17 1.18 1.19 1.20 1.21 1.22 1.23 1.24 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32 1.33
[103] 1.34 1.35 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 1.50
[120] 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 1.64 1.65 1.66 1.67
[137] 1.68 1.69 1.70 1.71 1.72
> # Tambahkan garis vertikal di lambda terbaik
> abline(v = lambda, col = "red", lty = 2)
```

Lampiran 131. *Code* R Uji ADF

```
> par(mfrow=c(1,1))
> acf(wisatawan_new, main = "ACF Perjalanan Wisatawan")
> tseries::adf.test(wisatawan_new)

        Augmented Dickey-Fuller Test

data:  wisatawan_new
Dickey-Fuller = -4.3628, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Lampiran 132. *Code* R Menampilkan Plot ACF dan PACF

```
> acf(wisatawan_new, lag.max = 15)
> pacf(wisatawan_new, lag.max = 15)
```

Lampiran 133. *Code* R Estimasi Model ARIMA

```
> summary(model1.da)
Series: wisatawan_new
ARIMA(3,0,1) with non-zero mean

Coefficients:
         ar1      ar2     ar3      ma1      mean
      1.5388  -0.9828  0.4103  -0.6644  9736.247
s.e.  0.1946   0.2420  0.1413   0.1785  1380.492

sigma^2 = 1973646:    log likelihood = -414.16
AIC=840.32     AICc=842.37   BIC=847.52

z test of coefficients:

            Estimate Std. Error z value            Pr(>|z|)
ar1          1.53876    0.19459  7.9078 0.00000000000000262 ***
ar2         -0.98285    0.24202 -4.0610 0.00004886789835132 ***
ar3          0.41030    0.14133  2.9031           0.0036950 **
ma1         -0.66436    0.17848 -3.7224           0.0001974 ***
intercept 9736.24719 1380.49238  7.0527 0.00000000000175435 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 134. *Code* R Uji *White Noise*

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.00083136, df = 1, p-value = 0.977
```

Lampiran 135. *Code* R Input Variabel Exogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Jembrana, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro2
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 136. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] 0.5073536
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = 4.4838, df = 58, p-value = 0.00003515
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2909062 0.6743961
sample estimates:
      cor
0.5073536
```

Lampiran 137. *Code* R Estimasi Model ARIMAX

```
> #MODEL FIX
> #Estimasi Model ARIMAX
> modelx1 <- Arima(wisatawan_new, order = c(1,0,0),
+                  xreg = birotrain.ts,
+                  method = "ML")

> summary(modelx1)
Series: wisatawan_new
Regression with ARIMA(1,0,0) errors

Coefficients:
         ar1  intercept      xreg
      0.6151  9151.3676  467.5204
s.e.  0.1278   594.4507  226.4330

sigma^2 = 2132546:  log likelihood = -416.55
AIC=841.09   AICc=842.02   BIC=848.58

z test of coefficients:

            Estimate Std. Error z value             Pr(>|z|)
ar1          0.61514    0.12784  4.8117          0.000001497 ***
intercept 9151.36760  594.45072 15.3947 < 0.00000000000000022 ***
xreg       467.52044  226.43302  2.0647               0.03895 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 138. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 1.5466, df = 1, p-value = 0.2136
```

Lampiran 139. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> # Forecast menggunakan model ARIMAX dengan data eksternal liburtest.ts
> ramalanx <- forecast::forecast(modelx1, xreg = birotest.ts)
> # Ambil hasil forecast dalam skala transformasi Box-Cox
> hasil.forc <- ramalanx$mean
> # Inverse Box-Cox ke skala asli
> hasil <- InvBoxCox(hasil.forc, lambda = 0.81)
> # Pastikan nilai forecast tidak negatif (jika ada)
> hasil[hasil < 0] <- 0
> # Bulatkan hasil forecast agar mudah dibaca
> hasil_rounded <- round(hasil)
> # Buat time series hasil forecast dengan frekuensi 12, mulai Jan 2024
> hasil_ts <- ts(hasil_rounded, start = c(2024, 1), frequency = 12)
> # Data aktual testing (pastikan ini sudah ts)
> testing.ts <- wisatawan_time_series [49:60]  # Jan 2024 - Des 2024
> # Buat tabel perbandingan aktual vs forecast
> perbandingan <- data.frame(
+   Bulan = time(testing.ts),
+   Aktual = as.numeric(testing.ts),
+   Forecast = as.numeric(hasil_ts)
+ )
```

```
> print(perbandingan)
   Bulan Aktual Forecast
1      1  82950    56498
2      2 101889    57824
3      3  90059    58642
4      4  88915    59146
5      5  75606    59456
6      6  78410    59648
7      7  72273    59765
8      8  63120    59838
9      9  80591    59882
10    10  66130    59910
11    11  69280    59927
12    12  77548    59937
```

Lampiran 140. *Code* R Tingkat Akurasi

```
> akurasi <- accuracy(hasil_ts, testing.ts)
> print(akurasi)
                ME     RMSE      MAE      MPE     MAPE
Test set 19691.5 22656.65 19691.5 23.5508 23.5508
```

Lampiran 141. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> # PREDIKSI 2 tahun = 24 bulan
> h <- 24
> # Asumsi variabel libur 2025-2026 = sama dengan 2024
> biro_future <- rep(birotest.ts, length.out = h)
> biro_future.ts <- ts(
+   biro_future,
+   start = c(2025,1),
+   frequency = 12
+ )
> # Forecast ARIMAX
> forecast_2tahun <- forecast::forecast(
+   modelx1,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> # Inverse Box-Cox
> forecast_inv <- InvBoxCox(forecast_2tahun$mean, lambda = 0.81)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 0.81)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 0.81)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 0.81)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 0.81)
> # Buat label bulan
> bulan <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )

> # Tabel hasil
> hasil_2025_2026 <- data.frame(
+   Bulan = format(bulan, "%b %Y"),
+   Prediksi = round(forecast_inv),
+   Lower_80 = round(lower80),
+   Upper_80 = round(upper80),
+   Lower_95 = round(lower95),
+   Upper_95 = round(upper95)
+ )
```

```
> print(hasil_2025_2026)
       Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1   Jan 2025    56498    41933    71821    34579    80201
2   Feb 2025    57824    40724    75953    32171    85910
3   Mar 2025    58642    40652    77759    31680    88270
4   Apr 2025    59146    40816    78638    31682    89359
5   May 2025    59456    40991    79095    31791    89898
6   Jun 2025    59648    41126    79346    31897    90181
7   Jul 2025    59765    41220    79489    31978    90338
8   Aug 2025    59838    41281    79572    32033    90427
9   Sep 2025    59882    41320    79622    32070    90479
10  Oct 2025    59910    41345    79652    32093    90511
11  Nov 2025    59927    41360    79670    32107    90529
12  Dec 2025    59937    41370    79681    32116    90541
13  Jan 2026    59943    41376    79688    32122    90548
14  Feb 2026    59947    41379    79692    32125    90552
15  Mar 2026    59950    41381    79695    32127    90555
16  Apr 2026    59951    41383    79696    32129    90556
17  May 2026    59952    41384    79697    32129    90557
18  Jun 2026    59953    41384    79698    32130    90558
19  Jul 2026    59953    41385    79698    32130    90558
20  Aug 2026    59953    41385    79698    32130    90559
21  Sep 2026    59953    41385    79699    32131    90559
22  Oct 2026    59953    41385    79699    32131    90559
23  Nov 2026    59953    41385    79699    32131    90559
24  Dec 2026    59953    41385    79699    32131    90559
```

Lampiran 142. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Bangli

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	83592	66501	64735	109985	89855
Februari	71607	47574	53131	76537	89197
Maret	47151	54028	109015	77243	86389
April	26287	83213	76189	169873	84794
Mei	27346	76087	85009	101167	85218
Juni	56138	71111	76207	152070	84753
Juli	70351	46733	76553	149765	83432
Agustus	95009	60597	72820	147192	88800
September	72635	62992	75645	136002	99748
Oktober	68302	70501	69787	138640	93604
November	64601	58480	70672	133990	97402
Desember	70678	66465	92501	94423	96466

Lampiran 143. Data Jumlah Biro Perjalanan Wisata di Kabupaten Bangli

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	2	2	0	135	0
Februari	2	2	0	135	0
Maret	2	2	0	135	0
April	2	2	0	135	0
Mei	2	2	0	135	0
Juni	2	2	0	135	0
Juli	2	2	0	135	0
Agustus	2	2	0	135	0
September	2	2	0	135	0
Oktober	2	2	0	135	0
November	2	2	0	135	0
Desember	2	2	0	135	0

Lampiran 144. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 145. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 146. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Bangli, na.rm = TRUE)
> Median <- median(wisatawan$Bangli, na.rm = TRUE)
> Min <- min(wisatawan$Bangli, na.rm = TRUE)
> Max <- max(wisatawan$Bangli, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Bangli, na.rm = TRUE)
> Observations <- length(wisatawan$Bangli)
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean Median   Min    Max  Std_Dev Observations
1 83446.47  76545 26287 169873 28983.94           60
```

Lampiran 147. *Code* R Plot Data Kabupaten Bangli

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,               # Ketebalan garis
+      xaxt = "n")            # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 148. *Code* R Plot Time Series Data Training

```
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Tahun",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 149. *Code* R Plot Time Series Data Testing

```
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 150. *Code* R Plot *Seasonal*

```
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020–2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 151. *Code* R Uji *Seasonal*

```
> qs(wisatawan_time_series)
Test used:  QS

Test statistic:  0
P-value:  1
```

Lampiran 152. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.24
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
 [1] -0.21 -0.20 -0.19 -0.18 -0.17 -0.16 -0.15 -0.14 -0.13 -0.12 -0.11 -0.10 -0.09 -0.08
[15] -0.07 -0.06 -0.05 -0.04 -0.03 -0.02 -0.01  0.00  0.01  0.02  0.03  0.04  0.05  0.06
[29]  0.07  0.08  0.09  0.10  0.11  0.12  0.13  0.14  0.15  0.16  0.17  0.18  0.19  0.20
[43]  0.21  0.22  0.23  0.24  0.25  0.26  0.27  0.28  0.29  0.30  0.31  0.32  0.33  0.34
[57]  0.35  0.36  0.37  0.38  0.39  0.40  0.41  0.42  0.43  0.44  0.45  0.46  0.47  0.48
[71]  0.49  0.50  0.51  0.52  0.53  0.54  0.55  0.56  0.57  0.58  0.59  0.60  0.61  0.62
[85]  0.63  0.64  0.65  0.66  0.67  0.68  0.69  0.70  0.71  0.72
```

Lampiran 153. *Code* R Transformasi Box-Cox

```
> wisatawan_new <- BoxCox(wisatawan_datatrain, lambda = 0.24)
> index <- seq(1:48)
> bc = boxcox(wisatawan_new~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 1.01
```

```
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
  [1] -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
 [15] -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
 [29] -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
 [43] -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
 [57] -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
 [71] -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
 [85]  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
 [99]  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
[113]  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
[127]  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
[141]  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
[155]  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
[169]  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
[183]  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
[197]  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
[211]  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
[225]  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
[239]  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
[253]  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
[267]  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
[281]  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
[295]  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
[309]  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
[323]  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
[337]  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
[351]  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
```

Lampiran 154. *Code* R Uji ADF Sebelum *Differencing*

```
> tseries::adf.test(wisatawan_new)

        Augmented Dickey-Fuller Test

data:  wisatawan_new
Dickey-Fuller = -3.0095, Lag order = 3, p-value = 0.171
alternative hypothesis: stationary
```

Lampiran 155. *Code* R Uji ADF Sesudah *Differencing*

```
> tseries::adf.test(train.diff)

        Augmented Dickey-Fuller Test

data:  train.diff
Dickey-Fuller = -5.6106, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Lampiran 156. *Code* R Menampilkan Plot ACF Sebelum *Differencing*

```
> acf(wisatawan_new, main = "ACF Perjalanan Wisatawan")
```

Lampiran 157. *Code* R Menampilkan Plot ACF dan PACF Setelah *Differencing*

```
> acf(train.diff)
> Pacf(train.diff)
```

Lampiran 158. *Code* R Estimasi Model ARIMA

```
> summary(model1.da)
Series: wisatawan_new
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.4420  -0.8338
s.e.  0.2002   0.1098

sigma^2 = 16.72:  log likelihood = -132.1
AIC=270.21   AICc=270.76   BIC=275.76

z test of coefficients:

    Estimate Std. Error z value            Pr(>|z|)
ar1  0.44201    0.20022  2.2077             0.02727 *
ma1 -0.83384    0.10980 -7.5944 0.00000000000003092 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 159. *Code* R Uji *White Noise*

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.003077, df = 1, p-value = 0.9558
```

Lampiran 160. *Code* R Input Data Variabel Eksogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Bangli, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro5
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 161. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] 0.6985436
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = 7.4346, df = 58, p-value = 0.0000000005453
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5404896 0.8089744
sample estimates:
      cor
0.6985436
```

Lampiran 162. *Code* R Estimasi Model ARIMAX

```
> #MODEL FIX
> modelx3 <- Arima(wisatawan_new, order = c(1,1,1),
+                  xreg = birotrain.ts,
+                  method = "ML")

> summary(modelx3)
Series: wisatawan_new
Regression with ARIMA(1,1,1) errors

Coefficients:
         ar1      ma1    xreg
      0.4167  -0.9599  0.0578
s.e.  0.1633   0.1217  0.0251

sigma^2 = 15.21:  log likelihood = -129.96
AIC=267.91   AICc=268.87   BIC=275.32
```

```
z test of coefficients:

      Estimate Std. Error z value             Pr(>|z|)
ar1   0.416715   0.163322  2.5515              0.01073 *
ma1  -0.959850   0.121693 -7.8875 0.000000000000003084 ***
xreg  0.057842   0.025074  2.3068              0.02106 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 163. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.025662, df = 1, p-value = 0.8727
```

Lampiran 164. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> # Forecast menggunakan model ARIMAX dengan data eksternal birotest.ts
> ramalanx <- forecast::forecast(modelx3, xreg = birotest.ts)
> # Ambil hasil forecast dalam skala transformasi Box-Cox
> hasil.forc <- ramalanx$mean
> # Inverse Box-Cox ke skala asli
> hasil <- InvBoxCox(hasil.forc, lambda = 0.24)
> # Pastikan nilai forecast tidak negatif (jika ada)
> hasil[hasil < 0] <- 0
> # Bulatkan hasil forecast agar mudah dibaca
> hasil_rounded <- round(hasil)
> # Buat time series hasil forecast dengan frekuensi 12, mulai Jan 2024
> hasil_ts <- ts(hasil_rounded, start = c(2024, 1), frequency = 12)
> # Data aktual testing (pastikan ini sudah ts)
> testing.ts <- wisatawan_time_series[49:60]  # Jan 2024 - Des 2024
> # Buat tabel perbandingan aktual vs forecast
> perbandingan <- data.frame(
+   Bulan = time(testing.ts),
+   Aktual = as.numeric(testing.ts),
+   Forecast = as.numeric(hasil_ts)
+ )

> print(perbandingan)
   Bulan Aktual Forecast
1      1  89855    63588
2      2  89197    67229
3      3  86389    68792
4      4  84794    69451
5      5  85218    69728
6      6  84753    69843
7      7  83432    69891
8      8  88800    69911
9      9  99748    69920
10    10  93604    69923
11    11  97402    69924
12    12  96466    69925
```

Lampiran 165. *Code* R Tingkat Akurasi

```
> akurasi <- accuracy(hasil_ts, testing.ts)
> print(akurasi)
                ME     RMSE      MAE      MPE     MAPE
Test set 20961.08 21660.21 20961.08 23.04354 23.04354
```

Lampiran 166. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> #PREDIKSI 2 TAHUN
> h <- 24  # 2 tahun
> rata_biro <- mean(birotrain.ts, na.rm = TRUE)
> biro_future.ts <- ts(
+   rep(rata_biro, h),
+   start = c(2025,1),
+   frequency = 12
+ )
> forecast_2tahun <- forecast::forecast(
+   modelx3,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> forecast_mean <- InvBoxCox(forecast_2tahun$mean, lambda = 0.24)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 0.24)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 0.24)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 0.24)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 0.24)
> bulan_forecast <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )
```

```
> hasil_forecast_2tahun <- data.frame(
+   Bulan = format(bulan_forecast, "%b %Y"),
+   Prediksi = round(as.numeric(forecast_mean)),
+   Lower_80 = round(as.numeric(lower80)),
+   Upper_80 = round(as.numeric(upper80)),
+   Lower_95 = round(as.numeric(lower95)),
+   Upper_95 = round(as.numeric(upper95))
+ )
```

```
> print(hasil_forecast_2tahun)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1   Jan 2025    73073    51237   101348    41911   119325
2   Feb 2025    77117    52374   109873    42017   131003
3   Mar 2025    78851    53194   112977    42499   135056
4   Apr 2025    79583    53572   114230    42745   136668
5   May 2025    79889    53716   114780    42829   137387
6   Jun 2025    80017    53755   115049    42836   137757
7   Jul 2025    80070    53747   115203    42809   137985
8   Aug 2025    80093    53720   115310    42767   138154
9   Sep 2025    80102    53685   115397    42719   138299
10 Oct 2025    80106    53647   115475    42669   138434
11 Nov 2025    80107    53608   115550    42618   138564
12 Dec 2025    80108    53568   115623    42566   138692
13 Jan 2026    80108    53528   115696    42515   138820
14 Feb 2026    80108    53487   115768    42463   138947
15 Mar 2026    80108    53447   115841    42412   139074
16 Apr 2026    80108    53407   115913    42360   139200
17 May 2026    80108    53368   115985    42309   139327
18 Jun 2026    80108    53328   116056    42258   139453
19 Jul 2026    80108    53288   116128    42207   139579
20 Aug 2026    80108    53248   116200    42157   139705
21 Sep 2026    80108    53209   116271    42106   139831
22 Oct 2026    80108    53170   116343    42056   139957
23 Nov 2026    80108    53130   116414    42005   140082
24 Dec 2026    80108    53091   116486    41955   140208
```

Lampiran 167. Data Jumlah Perjalanan Wisatawan Nusantara ke Kabupaten Klungkung

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	68936	60375	50966	83472	80985
Februari	77372	44321	44703	54365	91078
Maret	50894	50095	75745	55931	93750
April	20700	71670	63481	78240	84503
Mei	17533	69344	71253	84001	81263
Juni	28236	62039	87117	101165	96639
Juli	39072	36636	64230	102685	83543
Agustus	49161	52152	70547	105667	83900
September	52632	53708	61606	92792	92604
Oktober	53002	63041	61203	93698	84829
November	56734	70352	49947	92015	80994
Desember	61885	61155	63456	93402	83482

Lampiran 168. Data Jumlah Biro Perjalanan Wisata di Kabupaten Klungkung

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	1	4	0	79	96
Februari	1	4	0	79	96
Maret	1	4	0	79	96
April	1	4	0	79	96
Mei	1	4	0	79	96
Juni	1	4	0	79	96
Juli	1	4	0	79	96
Agustus	1	4	0	79	96
September	1	4	0	79	96
Oktober	1	4	0	79	96
November	1	4	0	79	96
Desember	1	4	0	79	96

Lampiran 169. Package R yang Digunakan

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,                # Ketebalan garis
+      xaxt = "n")             # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 170. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 171. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 172. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Klungkung, na.rm = TRUE)
> Median <- median(wisatawan$Klungkung, na.rm = TRUE)
> Min <- min(wisatawan$Klungkung, na.rm = TRUE)
> Max <- max(wisatawan$Klungkung, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Klungkung, na.rm = TRUE)
> Observations <- length(wisatawan$Klungkung)
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean Median   Min    Max  Std_Dev Observations
1 68505.03  69140 17533 105667 20325.09           60
```

Lampiran 173. *Code* R Plot Data Kabupaten Klungkung

```
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,                # Ketebalan garis
+      xaxt = "n")             # Hilangkan sumbu x default
```

Lampiran 174. *Code* R Plot Time Series Data Training

```
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Tahun",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 175. *Code* R Plot Time Series Data Testing

```
> # Data Testing: ambil data dari Jan 2024 sampai Des 2024
> wisatawan_datatest <- wisatawan_time_series[49:60]
> wisatawan_datatest.new <- ts(wisatawan_datatest, start = c(2024, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 176. *Code* R Plot *Seasonal*

```
> seasonplot(wisatawan_time_series,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Seasonal Plot Perjalanan Wisatawan (2020–2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 177. *Code* R Uji *Seasonal*

```
> qs(wisatawan_datatrain.new)
Test used:  QS

Test statistic:  0
P-value:  1
```

Lampiran 178. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 1.07
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
  [1] 0.51 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 0.66 0.67
 [18] 0.68 0.69 0.70 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82 0.83 0.84
 [35] 0.85 0.86 0.87 0.88 0.89 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.00 1.01
 [52] 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.10 1.11 1.12 1.13 1.14 1.15 1.16 1.17 1.18
 [69] 1.19 1.20 1.21 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
 [86] 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 1.50 1.51 1.52
[103] 1.53 1.54 1.55 1.56 1.57 1.58 1.59 1.60 1.61 1.62 1.63 1.64 1.65 1.66 1.67 1.68 1.69
```

Lampiran 179. *Code* R Uji ADF Sebelum *Differencing*

```
> tseries::adf.test(wisatawan_datatrain)

        Augmented Dickey-Fuller Test

data:  wisatawan_datatrain
Dickey-Fuller = -3.3568, Lag order = 3, p-value = 0.0739
alternative hypothesis: stationary
```

Lampiran 180. *Code* R Uji ADF Sesudah *Differencing*

```
> tseries::adf.test(train.diff)

        Augmented Dickey-Fuller Test

data:  train.diff
Dickey-Fuller = -5.5736, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Lampiran 181. *Code* R Menampilkan Plot ACF Sebelum *Differencing*

```
> acf(wisatawan_datatrain, main = "ACF Perjalanan Wisatawan")
```

Lampiran 182. *Code* R Menampilkan Plot ACF dan PACF Sesudah *Differencing*

```
> acf(train.diff)
> Pacf(train.diff)
```

Lampiran 183. *Code* R Estimasi Model ARIMA

```
> summary(model1.da)
Series: wisatawan_datatrain
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.6341  -0.8802
s.e.  0.1769   0.1027

sigma^2 = 177289259:  log likelihood = -512.2
AIC=1030.39   AICc=1030.95   BIC=1035.94

z test of coefficients:

    Estimate Std. Error z value             Pr(>|z|)
ar1  0.63408    0.17687  3.5850             0.000337 ***
ma1 -0.88018    0.10272 -8.5686 < 0.00000000000000022 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 184. *Code* R Uji *White Noise*

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.32473, df = 1, p-value = 0.5688
```

Lampiran 185. *Code* R Input Variabel Exgogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Klungkung, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro9
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
>
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 186. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] 0.7204011
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = 7.9105, df = 58, p-value = 0.00000000008652
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5709119 0.8236563
sample estimates:
      cor
0.7204011
```

Lampiran 187. *Code* R Estimasi Model ARIMAX

```
> #MODEL FIX
> modelx3 <- Arima(wisatawan_datatrain, order = c(1,1,1),
+                  xreg = birotrain.ts,
+                  method = "ML")

> summary(modelx3)
Series: wisatawan_datatrain
Regression with ARIMA(1,1,1) errors

Coefficients:
         ar1      ma1      xreg
      0.6511  -1.0000  342.7640
s.e.  0.1202   0.1018  115.5186

sigma^2 = 156921984:  log likelihood = -509.81
AIC=1027.62   AICc=1028.57   BIC=1035.02

z test of coefficients:

      Estimate Std. Error z value             Pr(>|z|)
ar1    0.65115    0.12022  5.4164         0.00000006079 ***
ma1   -0.99998    0.10185 -9.8184 < 0.00000000000000022 ***
xreg 342.76398  115.51864  2.9672             0.003006 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 188. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.84257, df = 1, p-value = 0.3587
```

Lampiran 189. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> # Forecast menggunakan model ARIMAX dengan data eksternal birotest.ts
> ramalanx <- forecast::forecast(modelx3, xreg = birotest.ts)
> # Ambil hasil forecast dalam skala transformasi Box-Cox
> data.ramalan.da <- ramalanx$mean
> plot(ramalanx)
> hasil.forc.Diff <- data.ramalan.da
> hasil1_rounded <- round(hasil.forc.Diff)
> perbandingan <- data.frame(
+   Aktual = as.numeric(head(wisatawan_datatest, 12)),
+   Forecast = as.numeric(hasil1_rounded)
+ )
```

```
> head(perbandingan,12)
      Aktual Hasil Forecast
 [1,]  80985           96181
 [2,]  91078           94195
 [3,]  93750           92903
 [4,]  84503           92061
 [5,]  81263           91513
 [6,]  96639           91156
 [7,]  83543           90924
 [8,]  83900           90773
 [9,]  92604           90674
[10,]  84829           90610
[11,]  80994           90568
[12,]  83482           90541
```

Lampiran 190. *Code* R Tingkat Akurasi

```
> accuracy(ts(hasil1_rounded), head(wisatawan_datatest, n = 12))
                 ME     RMSE      MAE       MPE     MAPE
Test set -5377.417 7713.375 6754.083 -6.600044 8.043595
```

Lampiran 191. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> # PREDIKSI 2 tahun = 24 bulan
> h <- 24
> # Asumsi variabel libur 2025-2026 = sama dengan 2024
> biro_future <- rep(birotest.ts, length.out = h)
> biro_future.ts <- ts(
+   biro_future,
+   start = c(2025,1),
+   frequency = 12
+ )
> # Forecast ARIMAX
> forecast_2tahun <- forecast::forecast(
+   modelx3,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> # Inverse Box-Cox
> forecast_inv <- InvBoxCox(forecast_2tahun$mean, lambda = 1.07)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 1.07)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 1.07)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 1.07)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 1.07)
> # Buat label bulan
> bulan <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )

> # Tabel hasil
> hasil_2025_2026 <- data.frame(
+   Bulan = format(bulan, "%b %Y"),
+   Prediksi = round(forecast_inv),
+   Lower_80 = round(lower80),
+   Upper_80 = round(upper80),
+   Lower_95 = round(lower95),
+   Upper_95 = round(upper95)
```

```
> print(hasil_2025_2026)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1   Jan 2025    48368    40706    55946    36609    59927
2   Feb 2025    47434    38186    56557    33228    61344
3   Mar 2025    46826    36925    56581    31610    61696
4   Apr 2025    46429    36230    56472    30752    61737
5   May 2025    46171    35826    56354    30268    61691
6   Jun 2025    46002    35583    56258    29983    61632
7   Jul 2025    45893    35433    56187    29810    61582
8   Aug 2025    45821    35338    56138    29703    61544
9   Sep 2025    45775    35278    56104    29635    61517
10  Oct 2025    45745    35240    56082    29592    61499
11  Nov 2025    45725    35215    56067    29564    61487
12  Dec 2025    45712    35199    56057    29546    61478
13  Jan 2026    45704    35188    56051    29535    61473
14  Feb 2026    45698    35182    56047    29527    61469
15  Mar 2026    45695    35177    56044    29522    61467
16  Apr 2026    45693    35174    56042    29519    61466
17  May 2026    45691    35173    56041    29517    61465
18  Jun 2026    45690    35171    56040    29516    61464
19  Jul 2026    45689    35171    56040    29515    61464
20  Aug 2026    45689    35170    56040    29514    61463
21  Sep 2026    45689    35170    56039    29514    61463
22  Oct 2026    45689    35169    56039    29514    61463
23  Nov 2026    45688    35169    56039    29514    61463
24  Dec 2026    45688    35169    56039    29514    61463
```

Lampiran 192. Data Jumlah Perjalanan Wisatawan Nusantara ke Kota Denpasar

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	168149	86201	122534	266772	395424
Februari	142624	100102	101453	211632	367266
Maret	100568	107089	136648	200244	374960
April	36145	109299	135110	201995	383188
Mei	30028	105705	188465	255795	405483
Juni	45454	123816	197999	342535	382791
Juli	63864	57776	203033	322564	351187
Agustus	82861	68745	166903	275708	338360
September	82577	96995	157011	331770	317816
Oktober	99591	122600	164866	361006	364181
November	103321	115907	160453	355902	355267
Desember	135584	164771	249950	400811	442726

Lampiran 193. Data Jumlah Biro Perjalanan Wisata di Kota Denpasar

Bulan	Tahun				
	2020	2021	2022	2023	2024
Januari	270	275	251	275	857
Februari	270	275	251	275	857
Maret	270	275	251	275	857
April	270	275	251	275	857
Mei	270	275	251	275	857
Juni	270	275	251	275	857
Juli	270	275	251	275	857
Agustus	270	275	251	275	857
September	270	275	251	275	857
Oktober	270	275	251	275	857
November	270	275	251	275	857
Desember	270	275	251	275	857

Lampiran 194. Package R yang Digunakan

```
> 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)
> library(seastests)
```

Lampiran 195. *Code* R Import Data

```
> wisatawan <- import("D:\\SKRIPSI\\data\\wisatawan.xlsx")
```

Lampiran 196. *Code* R Hitung Statistik Deskriptif

```
> Mean <- mean(wisatawan$Denpasar, na.rm = TRUE)
> Median <- median(wisatawan$Denpasar, na.rm = TRUE)
> Min <- min(wisatawan$Denpasar, na.rm = TRUE)
> Max <- max(wisatawan$Denpasar, na.rm = TRUE)
> Std_Dev <- sd(wisatawan$Denpasar, na.rm = TRUE)
> Observations <- length(wisatawan$Denpasar)
>
> deskriptif <- data.frame(Mean, Median, Min, Max, Std_Dev, Observations)
> print(deskriptif)
      Mean   Median   Min    Max  Std_Dev Observations
1 205659.7 165884.5 30028 442726 119264.7           60
```

Lampiran 197. *Code* R Plot Data Kota Denpasar

```
> options(scipen = 999)
> # Plot semua data
> plot(wisatawan_time_series,
+      main = "Jumlah Perjalanan Wisatawan 2020-2024",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "blue",
+      type = "o",
+      lwd = 2,               # Ketebalan garis
+      xaxt = "n")            # Hilangkan sumbu x default
> axis(1, at = seq(2020, 2024, by = 1), labels = seq(2020, 2024, by = 1))
```

Lampiran 198. *Code* R Plot Time Series Data Training

```
> # Ambil data training dari Jan 2020 sampai Des 2023 (48 bulan)
> wisatawan_datatrain <- wisatawan_time_series[1:48]
> wisatawan_datatrain.new <- ts(wisatawan_datatrain, start = c(2020, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatrain.new,
+      main = "Data Training",
+      xlab = "Bulan",
+      ylab = "Jumlah",
+      col = "black",
+      lty = 1)
```

Lampiran 199. *Code* R Plot Time Series Data Testing

```
> # Data Testing: ambil data dari Jan 2024 sampai Des 2024
> wisatawan_datatest <- wisatawan_time_series[49:60]
> wisatawan_datatest.new <- ts(wisatawan_datatest, start = c(2024, 1), frequency = 12)
> # Plot
> plot(wisatawan_datatest.new,
+      main = "Data Testing",
+      xlab = "Tahun",
+      ylab = "Jumlah Perjalanan",
+      col = "black",
+      lty = 1,
+      xaxt = "n")  # nonaktifkan sumbu X default
> axis(1, at = seq(2024, 2024 + 11/12, by = 1/12),
+      labels = month.abb, las = 2, cex.axis = 0.8)
```

Lampiran 200. *Code* R Plot *Seasonal*

```
> # Plot seasonal dan trend untuk data
> seasonplot(wisatawan_datatrain.new,
+            12,  # Frekuensi data (12 untuk data bulanan)
+            main = "Plot Musiman Perjalanan Wisatawan (2020-2024)",
+            xlab = "Bulan",
+            ylab = "Jumlah Perjalanan",
+            year.labels = TRUE,
+            col = rainbow(16))
```

Lampiran 201. *Code* R Uji *Seasonal*

```
> qs(wisatawan_datatrain.new)
Test used:  QS

Test statistic:  4.89
P-value:  0.08691536
```

Lampiran 202. *Code* R Uji Stasioner Dalam Ragam

```
> index <- seq(1:48)
> bc = boxcox(wisatawan_datatrain.new~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 0.39
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
 [1] 0.09 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 0.24 0.25
[18] 0.26 0.27 0.28 0.29 0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.40 0.41 0.42
[35] 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59
[52] 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71
```

Lampiran 203. *Code* R Transformasi Box-Cox

```
> wisatawan_new <- BoxCox(wisatawan_datatrain.new, lambda = 0.39)
> index <- seq(1:48)
> bc = boxcox(wisatawan_new~index, lambda = seq(-2, 4, by=.01))
> lambda <- bc$x[which.max(bc$y)]
> lambda
[1] 1.01
> bc$x[bc$y > max(bc$y) - 1/2 * qchisq(.95,1)]
  [1] 0.23 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 0.38 0.39
 [18] 0.40 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56
 [35] 0.57 0.58 0.59 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73
 [52] 0.74 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.90
 [69] 0.91 0.92 0.93 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
 [86] 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 1.22 1.23 1.24
[103] 1.25 1.26 1.27 1.28 1.29 1.30 1.31 1.32 1.33 1.34 1.35 1.36 1.37 1.38 1.39 1.40 1.41
[120] 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49 1.50 1.51 1.52 1.53 1.54 1.55 1.56 1.57 1.58
[137] 1.59 1.60 1.61 1.62 1.63 1.64 1.65 1.66 1.67 1.68 1.69 1.70 1.71 1.72 1.73 1.74 1.75
[154] 1.76 1.77 1.78 1.79 1.80 1.81 1.82
```

Lampiran 204. *Code* R Uji ADF Sebelum *Differencing*

```
> acf(wisatawan_new, main = "ACF Perjalanan Wisatawan")
> tseries::adf.test(wisatawan_new)

        Augmented Dickey-Fuller Test

data:  wisatawan_new
Dickey-Fuller = -3.0159, Lag order = 3, p-value = 0.1684
alternative hypothesis: stationary
```

Lampiran 205. *Code* R Uji ADF Sesudah *Differencing*

```
> tseries::adf.test(train.diff)

        Augmented Dickey-Fuller Test

data:  train.diff
Dickey-Fuller = -6.1855, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Lampiran 206. *Code* R Menampilkan Plot ACF Sebelum *Differencing*

```
> acf(wisatawan_new, main = "ACF Perjalanan Wisatawan")
```

Lampiran 207. *Code* R Menampilkan Plot ACF dan PACF Sesudah *Differencing*

```
> acf(train.diff,lag.max = 15)
> Pacf(train.diff, lag.max = 15)
```

Lampiran 208. *Code* R Estimasi Model ARIMA

```
> summary(model1.da)
Series: train.diff
ARIMA(2,1,2)

Coefficients:
         ar1      ar2      ma1     ma2
      0.8438  -0.3410  -1.9738  1.0000
s.e.  0.1447   0.1541   0.0912  0.0916

sigma^2 = 553.7:  log likelihood = -212.43
AIC=434.85   AICc=436.35   BIC=443.99

z test of coefficients:

     Estimate Std. Error  z value             Pr(>|z|)
ar1  0.843816   0.144681   5.8322        0.000000005469 ***
ar2 -0.341001   0.154100  -2.2129                0.02691 *
ma1 -1.973828   0.091192 -21.6447 < 0.00000000000000022 ***
ma2  0.999971   0.091645  10.9113 < 0.00000000000000022 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 209. *Code* R Uji *White Noise*

```
> Box.test(sisaan.da, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.da
X-squared = 0.75434, df = 1, p-value = 0.3851
```

Lampiran 210. *Code* R Input Data Variabel Eksogen

```
> #Variabel Exogen
> wisatawan_time_series<- ts(wisatawan$Denpasar, start = c(2020, 1), frequency = 12)
> data_biro <- wisatawan$Biro4
> databiro.train <- ts(data_biro[1:48])
> databiro.test <- ts(data_biro[49:60])
> #time series data
> birotrain.ts <- ts(databiro.train, frequency = 12, start = c(2020,1), end = c(2023,12))
> birotest.ts <- ts(databiro.test, frequency = 12, start = c(2024,1), end = c(2024,12))
```

Lampiran 211. *Code* R Uji Korelasi

```
> #Uji Korelasi
> correlation <- cor(data_biro, wisatawan_time_series)
> correlation
[1] 0.7102431
> cor.test(data_biro, wisatawan_time_series)

        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = 7.6838, df = 58, p-value = 0.0000000002079
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5567222 0.8168502
sample estimates:
      cor
0.7102431
```

Lampiran 212. *Code* R Estimasi Model ARIMAX

```
> # Dependen training (sesuai preprocessing)
> y_train <- wisatawan_new   # hasil Box-Cox / differencing
>
> model_arimax <- Arima(
+   y_train,
+   order = c(2, 1, 2),        # SESUAI HASIL IDENTIFIKASI
+   xreg  = birotrain.ts,
+   method = "ML"
+ )
```

```
> summary(model_arimax)
Series: y_train
Regression with ARIMA(2,1,2) errors

Coefficients:
         ar1      ar2      ma1     ma2    xreg
      1.2432  -0.9526  -1.4007  0.9999  0.0093
s.e.  0.0601   0.0483   0.2517  0.3498  0.6048

sigma^2 = 584.3:  log likelihood = -215.91
AIC=443.82   AICc=445.92   BIC=454.92

z test of coefficients:

       Estimate Std. Error  z value             Pr(>|z|)
ar1   1.2431641  0.0601184  20.6786 < 0.00000000000000022 ***
ar2  -0.9526466  0.0482839 -19.7301 < 0.00000000000000022 ***
ma1  -1.4007041  0.2516887  -5.5652          0.00000002618 ***
ma2   0.9998984  0.3497913   2.8586                0.004256 **
xreg  0.0092733  0.6047938   0.0153                0.034353 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Lampiran 213. *Code* R Diagnostik Model

```
> Box.test(sisaan.arimax, type = "Ljung")  #tak tolak H0 > sisaan saling bebas

        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.32496, df = 1, p-value = 0.5686
```

Lampiran 214. *Code* R Hasil Perbandingan Data Aktual dan Hasil Ramalan

```
> hasil_forecast <- data.frame(
+   Bulan = format(bulan_forecast, "%b %Y"),
+   Aktual = if(h == 12) as.numeric(wisatawan_datatest.new) else NA,

> # Tampilkan tabel forecast
> print(hasil_forecast)
      Bulan Aktual Prediksi
1  Jan 2024 395424   397550
2  Feb 2024 367266   396463
3  Mar 2024 374960   399984
4  Apr 2024 383188   405435
5  May 2024 405483   408858
6  Jun 2024 382791   407884
7  Jul 2024 351187   403423
8  Aug 2024 338360   398835
9  Sep 2024 317816   397357
10 Oct 2024 364181   399873
11 Nov 2024 355267   404435
12 Dec 2024 442726   407715
```

Lampiran 215. *Code* R Tingkat Akurasi

```
> if(h == 12){
+   MAE  <- mean(abs(as.numeric(wisatawan_datatest.new) -
+                      as.numeric(forecast_inv)))
+
+   MAPE <- mean(abs((wisatawan_datatest.new - forecast_inv) /
wisatawan_datatest.new)) * 100
+   cat("RMSE:", round(RMSE,2), " | MAPE:", round(MAPE,2), "%
\n", "MAE:", round(MAE,2))
+ }
RMSE: 88982.52  | MAPE: 9.81 %
 MAE: 34932.13
```

Lampiran 216. *Code* R Ramalan Perjalanan Wisatawan Nusantara Tahun 2025-2026

```
> # PREDIKSI 2 tahun = 24 bulan
> h <- 24
> # Asumsi variabel libur 2025-2026 = sama dengan 2024
> biro_future <- rep(birotest.ts, length.out = h)
> biro_future.ts <- ts(
+   biro_future,
+   start = c(2025,1),
+   frequency = 12
+ )
> # Forecast ARIMAX
> forecast_2tahun <- forecast::forecast(
+   model_arimax,
+   h = h,
+   xreg = biro_future.ts,
+   level = c(80, 95)
+ )
> # Inverse Box-Cox
> forecast_inv <- InvBoxCox(forecast_2tahun$mean, lambda = 0.39)
> lower80 <- InvBoxCox(forecast_2tahun$lower[,1], lambda = 0.39)
> upper80 <- InvBoxCox(forecast_2tahun$upper[,1], lambda = 0.39)
> lower95 <- InvBoxCox(forecast_2tahun$lower[,2], lambda = 0.39)
> upper95 <- InvBoxCox(forecast_2tahun$upper[,2], lambda = 0.39)
> # Buat label bulan
> bulan <- seq(
+   from = as.Date("2025-01-01"),
+   by = "month",
+   length.out = h
+ )

> # Tabel hasil
> hasil_2025_2026 <- data.frame(
+   Bulan = format(bulan, "%b %Y"),
+   Prediksi = round(forecast_inv),
+   Lower_80 = round(lower80),
+   Upper_80 = round(upper80),
+   Lower_95 = round(lower95),
+   Upper_95 = round(upper95)
```

```
> print(hasil_2025_2026)
      Bulan Prediksi Lower_80 Upper_80 Lower_95 Upper_95
1  Jan 2025   409859   331291   498853   293758   550342
2  Feb 2025   408751   307069   528695   260256   599947
3  Mar 2025   412338   297493   550809   245706   634209
4  Apr 2025   417891   292799   571099   237242   664270
5  May 2025   421378   285509   590758   226218   694877
6  Jun 2025   420385   271849   610112   208616   728431
7  Jul 2025   415842   253987   628695   187175   763689
8  Aug 2025   411168   237823   645419   168376   796272
9  Sep 2025   409662   227557   660690   156227   824114
10 Oct 2025   412226   223144   676152   150152   849138
11 Nov 2025   416872   221434   692162   146792   873473
12 Dec 2025   420213   218416   707420   142294   897617
13 Jan 2026   419913   211457   721068   134244   922065
14 Feb 2026   416361   201138   733366   123324   947054
15 Mar 2026   412252   190795   744917   112727   971394
16 Apr 2026   410518   183579   756748   105203   994278
17 May 2026   412258   180177   769986   101125  1016639
18 Jun 2026   416091   178885   784395    98890  1039242
19 Jul 2026   419205   177018   798233    96232  1061502
20 Aug 2026   419408   172569   810125    91532  1082987
21 Sep 2026   416692   165583   820199    84905  1104004
22 Oct 2026   413137   158121   829540    78094  1124603
23 Nov 2026   411302   152505   839561    72908  1144889
24 Dec 2026   412393   149561   851363    69860  1165643
```

Lampiran 217. Hasil Perhitungan Manual Uji ADF

a. Perhitungan dengan bantuan Excel

t	Y_t	*diff*	lag 1	lag 2	lag 3	*timed trend*	*lagged*
1	359,405						
2	305,047	-54,358				1	359,405
3	158,218	-146,829	-54,358			2	305,047
4	34,169	-124,049	-146,829	-54,358		3	158,218
5	32,851	-1,318	-124,049	-146,829	-54,358	4	34,169
6	45,074	12,223	-1,318	-124,049	-146,829	5	32,851
7	67,450	22,376	12,223	-1,318	-124,049	6	45,074
8	117,332	49,882	22,376	12,223	-1,318	7	67,450
9	112,677	-4,655	49,882	22,376	12,223	8	117,332
10	164,086	51,409	-4,655	49,882	22,376	9	112,677
11	187,589	23,503	51,409	-4,655	49,882	10	164,086
12	363,438	175,849	23,503	51,409	-4,655	11	187,589
13	106,201	-257,237	175,849	23,503	51,409	12	363,438
14	98,016	-8,185	-257,237	175,849	23,503	13	106,201
15	139,760	41,744	-8,185	-257,237	175,849	14	98,016
16	150,415	10,655	41,744	-8,185	-257,237	15	139,760
17	147,455	-2,960	10,655	41,744	-8,185	16	150,415
18	235,791	88,336	-2,960	10,655	41,744	17	147,455
19	45,197	-190,594	88,336	-2,960	10,655	18	235,791
20	70,656	25,459	-190,594	88,336	-2,960	19	45,197
21	153,854	83,198	25,459	-190,594	88,336	20	70,656
22	293,123	139,269	83,198	25,459	-190,594	21	153,854
23	313,615	20,492	139,269	83,198	25,459	22	293,123
24	506,465	192,850	20,492	139,269	83,198	23	313,615
25	317,167	-189,298	192,850	20,492	139,269	24	506,465

Parameter	Estimate	Std. Error	t-stat
intercept	32165.7284	27617.6701	1.1647
trend	13623.6285	3209.5273	4.2447
lagged	-1.0373	0.2400	-4.3218
lag 1	0.3793	0.1885	2.0126
lag 2	0.1908	0.1568	1.2169
lag 3	0.0602	0.1410	0.4266

coef	-1.0373
standart error	0.2400
t-stat	-4.3218
***p-value* (R)**	0.01

b. Perhitungan dengan bantuan RStudio

```
        Augmented Dickey-Fuller Test

data:  wisatawan_datatrain
Dickey-Fuller = -4.3218, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

Berdasarkan hasil Uji *Augmented Dickey-Fuller* pada Kabupaten Badung yang dilakukan secara manual dengan bantuan Excel, diperoleh nilai koefisien sebesar -1,0373 dengan standar error 0,2400. Nilai *t*-statistik yang dihasilkan adalah -3,6454 dengan nilai *p-value* sebesar 0,01, sehingga data dapat dikatakan stasioner. Langkah perhitungan yang sama kemudian diterapkan pada kabupaten lainnya.

Lampiran 218. Perhitungan Manual Nilai ACF dan PACF

a. Perhitungan dengan bantuan Excel

t	Y_t	$Y_t - \overline{Y}$	$(Y_t - \overline{Y})^2$
1	359,405	37599	1413692634
2	305,047	-16759	280860590
3	158,218	-163588	26760999663
4	34,169	-287637	82734983845
5	32,851	-288955	83494931826
6	45,074	-276732	76580542172
7	67,450	-254356	64696921745
8	117,332	-204474	41809574077
9	112,677	-209129	43734895072
10	164,086	-157720	24875565542
11	187,589	-134217	18014175127
12	363,438	41632	1733232097
13	106,201	-215605	46485471107
14	98,016	-223790	50081917477
15	139,760	-182046	33140708190
16	150,415	-171391	29374839175
17	147,455	-174351	30398234878
18	235,791	-86015	7398562305
19	45,197	-276609	76512481254
20	70,656	-251150	63076270177
21	153,854	-167952	28207839314
22	293,123	-28683	822708513
23	313,615	-8191	67090775
24	506,465	184659	34098984752
25	317,167	-4639	21519355

Lag	Numerator	Denominator	ACF	PACF
1	1434822877947	1756647109128	0.6795	0.8168
2	1193702246545	1756647109128	0.5850	0.0372
3	1027717278770	1756647109128	0.5538	0.0610
4	972876107486	1756647109128	0.5903	0.1587
5	1036870245926	1756647109128	0.5931	0.2482
6	1041936750633	1756647109128	0.4767	0.0376
7	837310721928	1756647109128	0.3573	-0.2867
8	627692484182	1756647109128	0.3338	-0.1101
9	586427081525	1756647109128	0.3070	0.2061
10	539253617248	1756647109128	0.2983	-0.0861
11	523988254204	1756647109128	0.3223	-0.1072

Lag	Numerator	Denominator	ACF	PACF
12	566183934581	1756647109128	0.2232	0.2203
13	392051534182	1756647109128	0.1323	-0.1603
14	232361885419	1756647109128	0.0578	-0.1628
15	101597948366	1756647109128	-0.0186	-0.1395
16	-32759968425	1756647109128	-0.0473	-0.0826
17	-83058673455	1756647109128	-0.0688	-0.0068
18	-120918980614	1756647109128	-0.1166	-0.1367
19	-204890260586	1756647109128	-0.2113	0.0297
20	-371158923813	1756647109128	-0.2260	-0.0735
21	-396970409225	1756647109128	-0.2600	0.0942
22	-456667311022	1756647109128	-0.2747	-0.0500
23	-482554220092	1756647109128	-0.2520	-0.0682
24	-442705167563	1756647109128	-0.3073	0.0571
25	-539765254752	1756647109128	-3.5190	-0.1032

b. Perhitungan dengan bantuan RStudio

```
> hasil_acf_pacf
       acf_values  pacf_values
 [1,]  0.67953446  0.816796311
 [2,]  0.58504481  0.037189361
 [3,]  0.55382558  0.060980979
 [4,]  0.59025529  0.158748170
 [5,]  0.59313948  0.248198784
 [6,]  0.47665278  0.037645142
 [7,]  0.35732418 -0.286724085
 [8,]  0.33383317 -0.110101662
 [9,]  0.30697891  0.206131134
[10,]  0.29828885 -0.086120479
[11,]  0.32230943 -0.107160794
[12,]  0.22318173  0.220337193
[13,]  0.13227579 -0.160342926
[14,]  0.05783629 -0.162823167
[15,] -0.01864915 -0.139465787
[16,] -0.04728250 -0.082567807
[17,] -0.06883510 -0.006772661
[18,] -0.11663712 -0.136735079
[19,] -0.21128827  0.029719528
[20,] -0.22598188 -0.073486420
[21,] -0.25996531  0.094201537
[22,] -0.27470186 -0.050027813
[23,] -0.25201713 -0.068169466
[24,] -0.30727017  0.057179083
[25,] -0.35194243 -0.103257340
```

Berdasarkan hasil perhitungan manual ACF dan PACF pada Kabupaten Badung dengan bantuan Excel dan RStudio diperoleh hasil yang sama. Langkah perhitungan yang sama juga diterapkan pada kabupaten lainnya untuk memperoleh model yang sesuai pada masing-masing wilayah.

Lampiran 219. Perhitungan Manual Estimasi Parameter Model SARIMA

a. Perhitungan dengan bantuan Excel

Parameter	*Estimate*	*Std. Error*	*z value*	*p-value*
ar1	0.821078	0.078766	10.4242694	0
sma1	0.58017	0.195785	2.96330158	0.003043582
intercept	366969.892	95745.43204	3.83276658	0.00012671

b. Perhitungan dengan bantuan RStudio

```
z test of coefficients:

               Estimate     Std. Error z value             Pr(>|z|)
ar1            0.821078       0.078766 10.4243 < 0.00000000000000022 ***
sma1           0.580170       0.195785  2.9633             0.0030435 **
intercept 366969.891970  95745.432038  3.8328             0.0001267 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Berdasarkan hasil perhitungan manual estimasi parameter model SARIMA pada Kabupaten Badung diperoleh nilai *p-value* pada setiap parameter lebih kecil dari 0,05, yang menunjukkan bahwa parameter tersebut signifikan secara statistik sehingga layak digunakan dalam model. Langkah perhitungan yang sama juga diterapkan pada kabupaten lainnya.

Lampiran 220. Perhitungan Manual Uji Korelasi

a. Perhitungan dengan bantuan Excel

t	Perjalanan Wisatawan (Y)	Biro Perjalanan (X)	$Y_t - \bar{Y}$	$X_t - \bar{X}$	$(Y_t - \bar{Y})(X_t - \bar{X})$	$(Y_t - \bar{Y})^2$	$(X_t - \bar{X})^2$
1	359405	154	-10464.8	59.8	-625792.05	109510992.6	3576.04
2	305047	154	-64822.8	59.8	-3876400.45	4201988918	3576.04
3	158218	154	-211652	59.8	-12656774.7	44796463278	3576.04
4	34169	154	-335701	59.8	-20074904.9	1.12695E+11	3576.04
5	32851	154	-337019	59.8	-20153721.3	1.13582E+11	3576.04
6	45074	154	-324796	59.8	-19422785.9	1.05492E+11	3576.04
7	67450	154	-302420	59.8	-18084701.1	91457705190	3576.04
8	117332	154	-252538	59.8	-15101757.5	63775315175	3576.04
9	112677	154	-257193	59.8	-15380126.5	66148110653	3576.04
10	164086	154	-205784	59.8	-12305868.3	42346951764	3576.04
11	187589	154	-182281	59.8	-10900388.9	33226271821	3576.04
12	363438	154	-6431.75	59.8	-384618.65	41367408.06	3576.04
13	106201	50	-263669	-44.2	11654158.8	69521209727	1953.64
14	98016	50	-271854	-44.2	12015935.8	73904461389	1953.64
15	139760	50	-230110	-44.2	10170851	52950497045	1953.64
16	150415	50	-219455	-44.2	9699899.95	48160387298	1953.64
17	147455	50	-222415	-44.2	9830731.95	49468321018	1953.64
18	235791	50	-134079	-44.2	5926280.75	17977111202	1953.64
19	45197	50	-324673	-44.2	14350535.6	1.05412E+11	1953.64
20	70656	50	-299214	-44.2	13225247.8	89528868189	1953.64

t	Perjalanan Wisatawan (Y)	Biro Perjalanan (X)	$Y_t - \bar{Y}$	$X_t - \bar{X}$	$(Y_t - \bar{Y})(X_t - \bar{X})$	$(Y_t - \bar{Y})^2$	$(X_t - \bar{X})^2$
21	153854	50	-216016	-44.2	9547896.15	46662804248	1953.64
22	293123	50	-76746.8	-44.2	3392206.35	5890063636	1953.64
23	313615	50	-56254.8	-44.2	2486459.95	3164596898	1953.64
24	506465	50	136595.3	-44.2	-6037510.05	18658262323	1953.64
25	317167	155	-52702.8	60.8	-3204327.2	2777579858	3696.64
26	203581	155	-166289	60.8	-10110356	27651948377	3696.64
27	322912	155	-46957.8	60.8	-2855031.2	2205030285	3696.64
28	283151	155	-86718.8	60.8	-5272500	7520141602	3696.64
29	480742	155	110872.3	60.8	6741032.8	12292655820	3696.64
30	477519	155	107649.3	60.8	6545074.4	11588361026	3696.64
31	419536	155	49666.25	60.8	3019708	2466736389	3696.64
32	364591	155	-5278.75	60.8	-320948	27865201.56	3696.64
33	358106	155	-11763.8	60.8	-715236	138385814.1	3696.64
34	398351	155	28481.25	60.8	1731660	811181601.6	3696.64
35	377934	155	8064.25	60.8	490306.4	65032128.06	3696.64
36	628402	155	258532.3	60.8	15718760.8	66838924290	3696.64
37	645498	50	275628.3	-44.2	-12182768.7	75970932198	1953.64
38	472495	50	102625.3	-44.2	-4536036.05	10531941938	1953.64
39	449361	50	79491.25	-44.2	-3513513.25	6318858827	1953.64
40	417855	50	47985.25	-44.2	-2120948.05	2302584218	1953.64
41	560738	50	190868.3	-44.2	-8436376.65	36430688858	1953.64
42	686304	50	316434.3	-44.2	-13986393.9	1.00131E+11	1953.64
43	613022	50	243152.3	-44.2	-10747329.5	59123016680	1953.64

t	Perjalanan Wisatawan (Y)	Biro Perjalanan (X)	$Y_t - \bar{Y}$	$X_t - \bar{X}$	$(Y_t - \bar{Y})(X_t - \bar{X})$	$(Y_t - \bar{Y})^2$	$(X_t - \bar{X})^2$
44	474715	50	104845.3	-44.2	-4634160.05	10992526448	1953.64
45	514808	50	144938.3	-44.2	-6406270.65	21007096313	1953.64
46	591582	50	221712.3	-44.2	-9799681.45	49156321800	1953.64
47	540667	50	170797.3	-44.2	-7549238.45	29171700608	1953.64
48	639762	50	269892.3	-44.2	-11929237.5	72841826610	1953.64
49	645834	62	275964.3	-32.2	-8886048.85	76156267278	1036.84
50	586598	62	216728.3	-32.2	-6978649.65	46971134348	1036.84
51	498829	62	128959.3	-32.2	-4152487.85	16630488161	1036.84
52	584635	62	214765.3	-32.2	-6915441.05	46124112608	1036.84
53	644499	62	274629.3	-32.2	-8843061.85	75421224956	1036.84
54	525045	62	155175.3	-32.2	-4996643.05	24079358213	1036.84
55	528516	62	158646.3	-32.2	-5108409.25	25168632639	1036.84
56	527520	62	157650.3	-32.2	-5076338.05	24853601325	1036.84
57	541689	62	171819.3	-32.2	-5532579.85	29521854671	1036.84
58	535982	62	166112.3	-32.2	-5348814.45	27593279600	1036.84
59	496263	62	126393.3	-32.2	-4069862.65	15975253646	1036.84
60	630092	62	260222.3	-32.2	-8379156.45	67715619395	1036.84
				SUM	**-211066449**	**2.34374E+12**	**146601.6**

$\bar{Y}$	$\bar{X}$	Pearson r
369869.8	94.2	-0.36008

b. Perhitungan dengan bantuan RStudio

```
        Pearson's product-moment correlation

data:  data_biro and wisatawan_time_series
t = -2.9394, df = 58, p-value = 0.004714
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5625646 -0.1168342
sample estimates:
       cor
-0.3600765
```

Berdasarkan hasil perhitungan manual uji korelasi antara jumlah perjalanan wisatawan nusantara ke Kabupaten Badung dan jumlah biro perjalanan wisata di Kabupaten Badung diperoleh nilai koefisien korelasi sebesar −0,36008 yang menunjukkan hubungan negatif dengan tingkat korelasi yang rendah. Hasil perhitungan manual tersebut juga konsisten dengan hasil yang diperoleh menggunakan bantuan program R. Dengan langkah perhitungan yang sama diterapkan pada kabupaten lainnya.

Lampiran 221. Perhitungan Manual Estimasi Parameter Model SARIMAX

a. Perhitungan dengan bantuan Excel

Parameter	*Estimate*	*Std. Error*	*z value*	*p-value*
ar1	0.823193	0.079761	10.32074573	0
sma1	0.494802	0.174937	2.828458245	0.00467728
intercept	24121.78372	5675.578341	4.250101447	2.13674E-05
xreg	0.51939	20.784369	0.024989452	0.049003169

b. Perhitungan dengan bantuan RStudio

```
z test of coefficients:

             Estimate   Std. Error z value             Pr(>|z|)
ar1          0.823193     0.079761 10.3207 < 0.00000000000000022 ***
sma1         0.494802     0.174937  2.8285              0.004677 **
intercept 24121.783720  5675.578341  4.2501            0.00002137 ***
xreg          0.519390    20.784369  0.0250              0.04930032 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Berdasarkan hasil perhitungan manual estimasi parameter model SARIMAX pada Kabupaten Badung diperoleh nilai *p-value* pada setiap parameter lebih kecil dari 0,05, yang menunjukkan bahwa parameter tersebut signifikan secara statistik sehingga layak digunakan dalam model. Langkah perhitungan yang sama juga diterapkan pada kabupaten lainnya.

Lampiran 217. Perhitungan Manual Uji *White Noise*

a. Perhitungan dengan bantuan Excel

t	Residual e_t	$e_t - \bar{e}$	$(e_t - \bar{e})^2$
1	205	183	33597
2	-2558	-2579	6652439
3	-8127	-8148	66393975
4	-9916	-9937	98751142
5	-3350	-3372	11371327
6	-2315	-2337	5460058
7	-1506	-1528	2335349
8	371	349	121761
9	-2612	-2634	6937974
10	614	592	350757
11	-612	-634	401557
12	7262	7240	52414501
13	-13874	-13896	193089603
14	-1926	-1948	3794442
15	3734	3712	13780082
16	2903	2882	8303841
17	-679	-700	490640
18	4309	4287	18379472
19	-12888	-12910	166671661
20	-1397	-1419	2013549
21	3902	3880	15053530
22	5931	5909	34921989
23	978	956	913966
24	6438	6416	41164000
25	-1912	-1934	3738725
26	-5833	-5855	34282477
27	3481	3459	11963490
28	-3839	-3861	14908727
29	9921	9899	97999643
30	-1075	-1097	1202702
31	4426	4405	19400595
32	-1548	-1570	2464508
33	-2111	-2133	4549073
34	-718	-740	546890
35	-1105	-1126	1268527
36	9351	9329	87027838
37	4188	4167	17360161
38	-2818	-2839	8062261
39	-1761	-1782	3176512
40	1194	1172	1373317

t	Residual e_t	$e_t - \bar{e}$	$(e_t - \bar{e})^2$
41	2758	2737	7489764
42	8214	8193	67117708
43	-2658	-2680	7180618
44	-3578	-3600	12957844
45	4116	4095	16767061
46	5479	5457	29779064
47	236	214	45889
48	1749	1727	2982127

Mean residual	21.73210625
Denominator	1203446736

Lag	ρ_k	ρ_k^2	$\frac{\rho_k}{n-1}$	Q Ljung-Box	df	*p-value*
1	0.00645	4.20073E-05	8.93773E-07	0.002145	1	0.963059

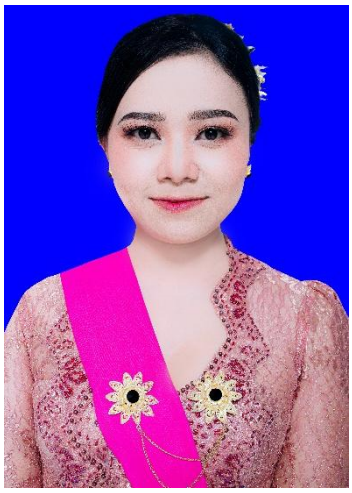
b. Perhitungan dengan bantuan RStudio

```
        Box-Ljung test

data:  sisaan.arimax
X-squared = 0.0021451, df = 1, p-value = 0.9631
```

Berdasarkan hasil perhitungan manual uji Ljung-Box pada residual model SARIMAX Kabupaten Badung diperoleh nilai *p-value* sebesar 0,963. Nilai tersebut menunjukkan bahwa residual bersifat *white noise*, yang berarti residual tidak memiliki autokorelasi. Hasil perhitungan manual ini juga konsisten dengan hasil yang diperoleh menggunakan perangkat lunak RStudio. Langkah perhitungan yang sama juga diterapkan pada kabupaten lainnya

RIWAYAT HIDUP



Ni Komang Imelda Fitriyani lahir di Desa Batumadeg pada tanggal 11 April 2004 dari pasangan suami istri Bapak I Gede Suamba dan Ibu Ni Komang Suryani. Penulis berkebangsaan Indonesia dan beragama Hindu serta beralamat di Banjar Dinas Batumadeg, Desa Tista, Kecamatan Abang, Kabupaten Karangasem, Provinsi Bali. Penulis memulai pendidikan di TK Pembina Abang dan lulus pada tahun 2010, kemudian menyelesaikan pendidikan dasar di SD Negeri 1 Tista pada tahun 2016, melanjutkan pendidikan di SMP Negeri 1 Abang dan lulus pada tahun 2019, serta menamatkan pendidikan di SMA Negeri 1 Bebandem jurusan MIPA pada tahun 2022. Selanjutnya penulis melanjutkan studi pada Program Studi S1 Matematika di Universitas Pendidikan Ganesha. Selama menempuh pendidikan di perguruan tinggi, penulis aktif dalam organisasi Himpunan Mahasiswa Jurusan Matematika sebagai anggota Sie Kesenian masa bakti 2022/2023 serta anggota Sie Kesekretariatan pada masa bakti 2023/2024 dan 2024/2025. Penulis juga mengikuti program MBKM sebagai Bendahara pada Program Penguatan Kapasitas Organisasi Kemahasiswaan (PPK Ormawa) BEM FMIPA Undiksha tahun 2024 yang berhasil memperoleh pendanaan, serta sebagai anggota Program Pengembangan Usaha Mahasiswa (P2MW) Jurusan Matematika tahun 2025 dengan peran koordinator logistik yang lolos KMI Expo.