

# Tugas Akhir

Abdan SM

2024-12-04

## Import Library

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(here)
```

```
## here() starts at C:/Users/ABDAN SM/OneDrive/Dokumen/Abdan/Kuliah/UPNVYK/Semester 5/prak data science
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.4.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method           from
##   as.zoo.data.frame zoo
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.4.2
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.4.2
```

```
##  
## Attaching package: 'Metrics'  
##  
## The following object is masked from 'package:forecast':  
##  
## accuracy
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.2
```

```
## Loading required package: lattice  
##  
## Attaching package: 'caret'  
##  
## The following objects are masked from 'package:Metrics':  
##  
## precision, recall  
##  
## The following object is masked from 'package:purrr':  
##  
## lift
```

## Import Data

```
path = here("data_raw", "dota2_chart.csv")  
data_dota2 = read.csv(path, sep=";")  
head(data_dota2)
```

```
##           DateTime Players Average.Players  
## 1 2011-09-22 00:00:00    194             NA  
## 2 2011-09-23 00:00:00    240             NA  
## 3 2011-09-24 00:00:00     NA             NA  
## 4 2011-09-25 00:00:00    233             NA  
## 5 2011-09-26 00:00:00    222             NA  
## 6 2011-09-27 00:00:00    303             NA
```

## Pre-Processing Data

ubah Kolom DateTime menjadi tipe data POSIXct, drop kolom Average.Players (terlalu banyak NA) cari nilai NA lainnya

```
new_dota2 = data_dota2 %>%  
  mutate(DateTime = as.POSIXct(DateTime), Average.Players = NULL)  
which(is.na(new_dota2$Players))
```

```
## [1] 3 9 10 1244 1253 1254 1255 1256 1258 1259 1260 1261 1262 1263 1264
## [16] 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1283 1284 1301 6347
```

```
#head(new_dota2)
```

ganti nilai NA menggunakan forward filling

```
clean_dota2 = na.locf(new_dota2)
which(is.na(clean_dota2$Players))
```

```
## integer(0)
```

Data memiliki record dengan rentang waktu yang berbeda-beda

Kelompokkan data menjadi Peak data harian dan Peak data bulanan

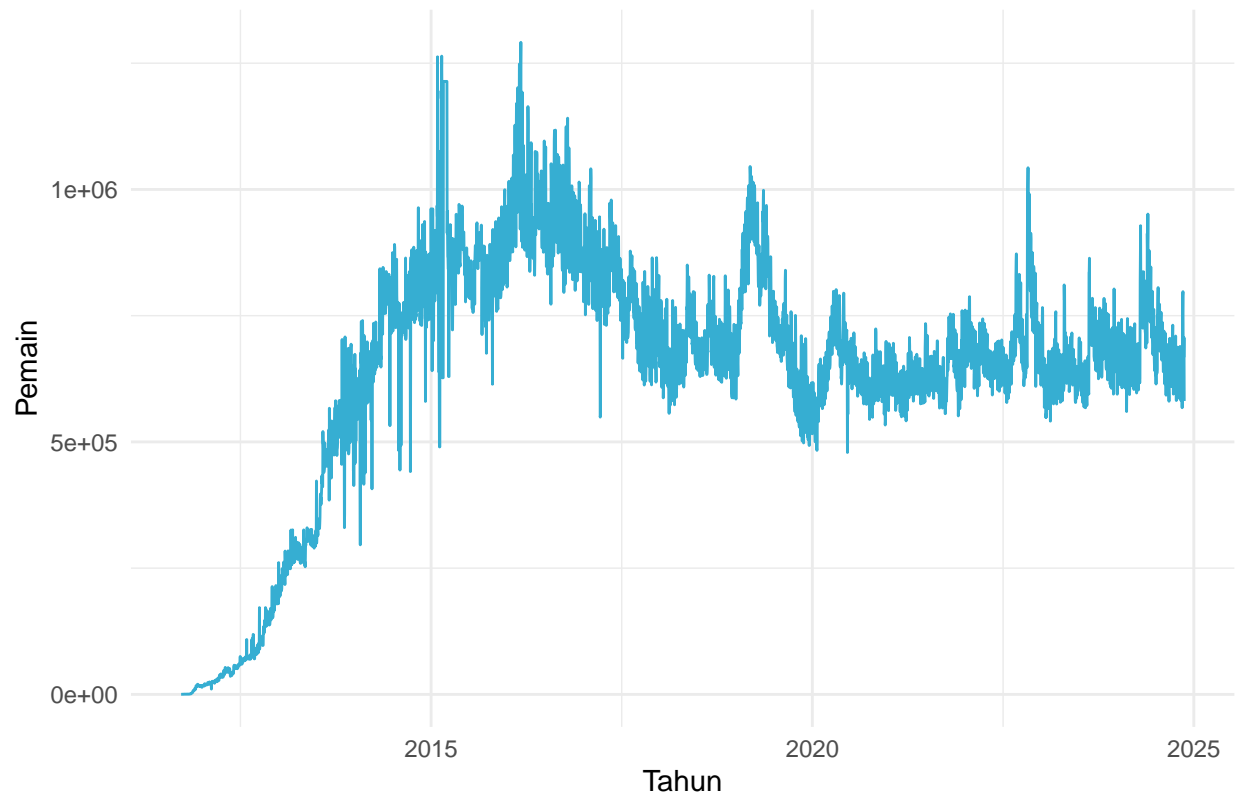
```
daily_dota2 = clean_dota2 %>%
  group_by(Date = lubridate::floor_date(DateTime, 'day')) %>%
  summarize(Peak = max(Players))
head(daily_dota2)
```

```
## # A tibble: 6 x 2
##   Date                Peak
##   <dtm>              <int>
## 1 2011-09-22 00:00:00    194
## 2 2011-09-23 00:00:00    240
## 3 2011-09-24 00:00:00    240
## 4 2011-09-25 00:00:00    233
## 5 2011-09-26 00:00:00    222
## 6 2011-09-27 00:00:00    303
```

visualisasikan data harian

```
ggplot(
  data = daily_dota2,
  aes(
    x = Date,
    y = Peak
  )
) + geom_line(color = "#36aed2") + theme_minimal() + labs(
  title = "Data Harian Pemain Dota2",
  x = "Tahun",
  y = "Pemain"
)
```

## Data Harian Pemain Dota2



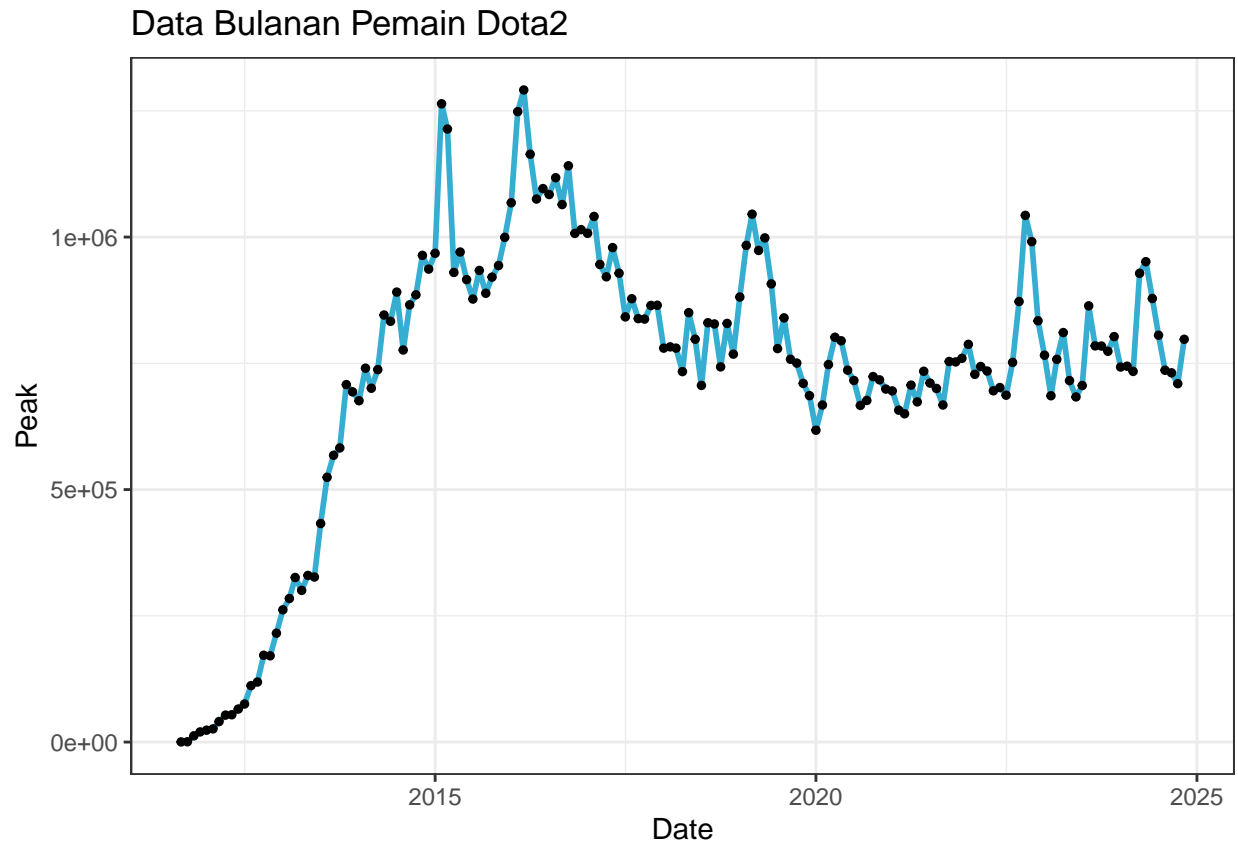
Kelompokkan data menjadi Peak data Bulanan

```
monthly_dota2 = clean_dota2 %>%
  group_by(Date = lubridate::floor_date(DateTime, 'month')) %>%
  summarize(Peak = max(Players))
monthly_dota2
```

```
## # A tibble: 159 x 2
##   Date                Peak
##   <dtm>              <int>
## 1 2011-09-01 00:00:00    312
## 2 2011-10-01 00:00:00    587
## 3 2011-11-01 00:00:00  12385
## 4 2011-12-01 00:00:00  20156
## 5 2012-01-01 00:00:00  23539
## 6 2012-02-01 00:00:00  26129
## 7 2012-03-01 00:00:00  40661
## 8 2012-04-01 00:00:00  53426
## 9 2012-05-01 00:00:00  54098
## 10 2012-06-01 00:00:00  65425
## # i 149 more rows
```

Visualisasikan data bulanan

```
ggplot(
  data = monthly_dota2,
  aes(
    x = Date,
    y = Peak
  )
) + geom_line(color = "#36aed2", linewidth = 1) + geom_point(size = 1) + theme_bw() + labs(
  title = "Data Bulanan Pemain Dota2"
)
```



## filter dan Scalling

```
filtered_daily_dota2 = monthly_dota2 %>% filter(Date >= as.POSIXct("2011-01-01"))

predict1 = preProcess(filtered_daily_dota2, method=c("range"))
scaled_daily_dota2 = predict(predict1, filtered_daily_dota2)
scaled_daily_dota2
```

```
## # A tibble: 159 x 2
##   Date                Peak
##   <dtm>              <dbl>
## 1 2011-09-01 00:00:00 0
```

```
## 2 2011-10-01 00:00:00 0.000213
## 3 2011-11-01 00:00:00 0.00935
## 4 2011-12-01 00:00:00 0.0154
## 5 2012-01-01 00:00:00 0.0180
## 6 2012-02-01 00:00:00 0.0200
## 7 2012-03-01 00:00:00 0.0313
## 8 2012-04-01 00:00:00 0.0411
## 9 2012-05-01 00:00:00 0.0417
## 10 2012-06-01 00:00:00 0.0504
## # i 149 more rows
```

## Membagi Data menjadi Training dan Testing

```
test_set_start_date <- as.POSIXct("2023-01-01")
train_dota2 <- subset(scaled_daily_dota2, Date <= test_set_start_date)
test_dota2 <- subset(scaled_daily_dota2, Date > test_set_start_date)

dim(train_dota2)
```

```
## [1] 137 2
```

```
dim(test_dota2)
```

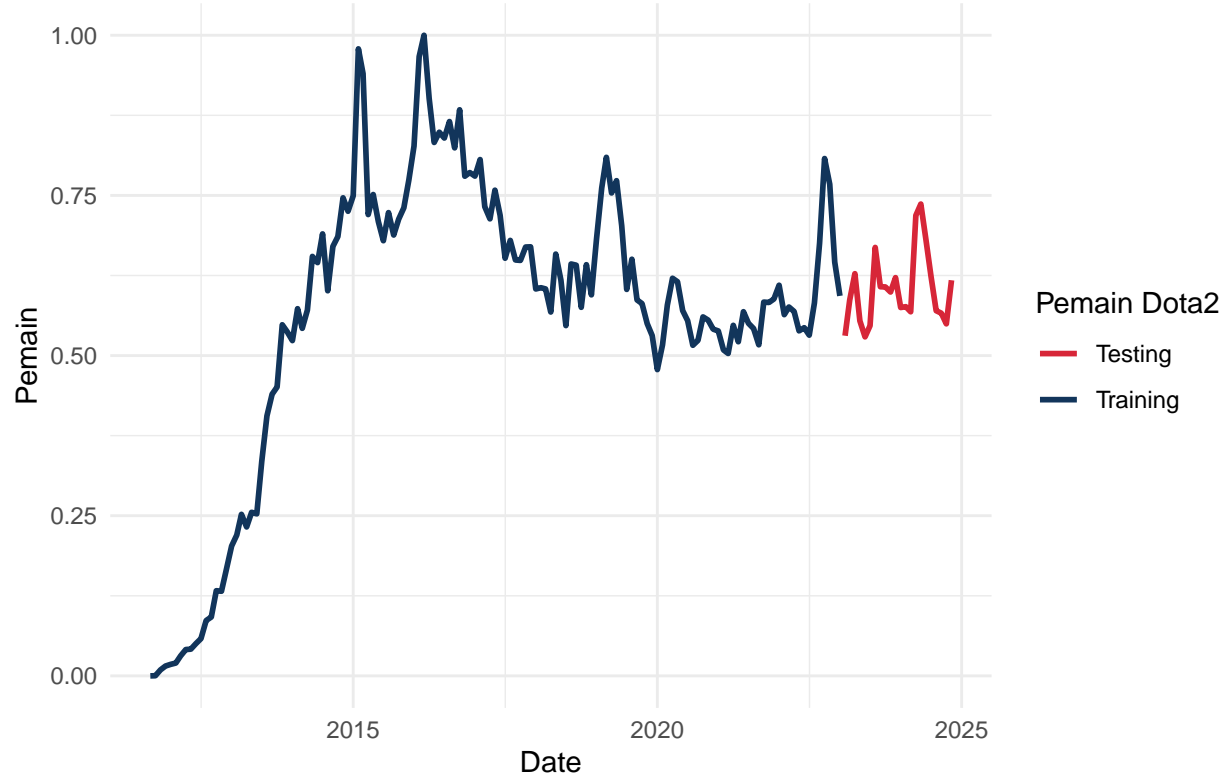
```
## [1] 22 2
```

visualisasikan data yang sudah dibagi

```
ggplot() +
  geom_line(data = train_dota2, aes(x = Date, y = Peak, color = "Training"), size = 1) +
  geom_line(data = test_dota2, aes(x = Date, y = Peak, color = "Testing"), size = 1) +
  labs(
    title = "Data Bulanan Pemain Dota2 - Training and Testing Sets",
    x = "Date",
    y = "Pemain"
  ) +
  scale_color_manual(values = c("Training" = "#12355B", "Testing" = "#D72638"), name = "Pemain Dota2") +
  theme_minimal() +
  theme(plot.title = element_text(size = 18))
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

## Data Bulanan Pemain Dota2 – Training and Testing Set



konversi data training menjadi object Time Series

```
ts_monthly_dota2 <- ts(train_dota2$Peak, start = c(2011,9), frequency = 12)
ts_monthly_dota2
```

	Jan	Feb	Mar	Apr	May
## 2011					
## 2012	0.0179912565	0.0199974284	0.0312536793	0.0411412407	0.0416617610
## 2013	0.2026380773	0.2199213643	0.2521928466	0.2324541292	0.2553531482
## 2014	0.5233444047	0.5732175279	0.5424820451	0.5710339763	0.6547959127
## 2015	0.7494717339	0.9788259789	0.9400565136	0.7200940964	0.7514167137
## 2016	0.8269742590	0.9667440218	1.0000000000	0.9014055597	0.8326736462
## 2017	0.7801134920	0.8060047281	0.7323511095	0.7133962708	0.7581803789
## 2018	0.6039793465	0.6059312975	0.6040211740	0.5679774689	0.6584418783
## 2019	0.6823648971	0.7615110889	0.8094717649	0.7538496812	0.7730601325
## 2020	0.4781001940	0.5167387546	0.5787991783	0.6206995111	0.6152720028
## 2021	0.5384240009	0.5088209596	0.5032757146	0.5472434114	0.5215644113
## 2022	0.6098274537	0.5638202780	0.5758363955	0.5688721131	0.5385231477
## 2023	0.5931638338				
	Jun	Jul	Aug	Sep	Oct
## 2011				0.0000000000	0.0002130105
## 2012	0.0504354710	0.0581270875	0.0863598902	0.0918571110	0.1329332867
## 2013	0.2529194061	0.3350268316	0.4058586416	0.4396452097	0.4509734968
## 2014	0.6450988989	0.6899573669	0.6011451446	0.6704037750	0.6857769385
## 2015	0.7090647986	0.6792727588	0.7231746160	0.6881525868	0.7129377173
## 2016	0.8486974600	0.8395604702	0.8653703750	0.8242074459	0.8837063212

```
## 2017 0.7189051104 0.6519376987 0.6798056724 0.6492382744 0.6485488948
## 2018 0.6176956753 0.5467910545 0.6429633715 0.6410973993 0.5752678511
## 2019 0.7025699139 0.6032829957 0.6504179654 0.5868935784 0.5810958191
## 2020 0.5701935530 0.5544021143 0.5160671905 0.5237518358 0.5603772533
## 2021 0.5685173538 0.5503835739 0.5421567200 0.5167836805 0.5835055491
## 2022 0.5435184382 0.5318191254 0.5820818642 0.6754834952 0.8076019197
## 2023
##           Nov           Dec
## 2011 0.0093515495 0.0153708397
## 2012 0.1320936379 0.1667252768
## 2013 0.5481155927 0.5369615868
## 2014 0.7463098831 0.7252202916
## 2015 0.7306826561 0.7739175967
## 2016 0.7799732924 0.7857059866
## 2017 0.6694510370 0.6697260142
## 2018 0.6418967697 0.5947618000
## 2019 0.5499513561 0.5311305205
## 2020 0.5553385860 0.5412450349
## 2021 0.5828649684 0.5883474721
## 2022 0.7673638437 0.6458912980
## 2023
```

## Modelling

forecast dengan Auto SARima lalu visualisasikan hasil

```
arima_monthly_dota2 = auto.arima(ts_monthly_dota2, seasonal = TRUE)
predict_arima_monthly_dota2 = forecast(arima_monthly_dota2, h = nrow(test_dota2))

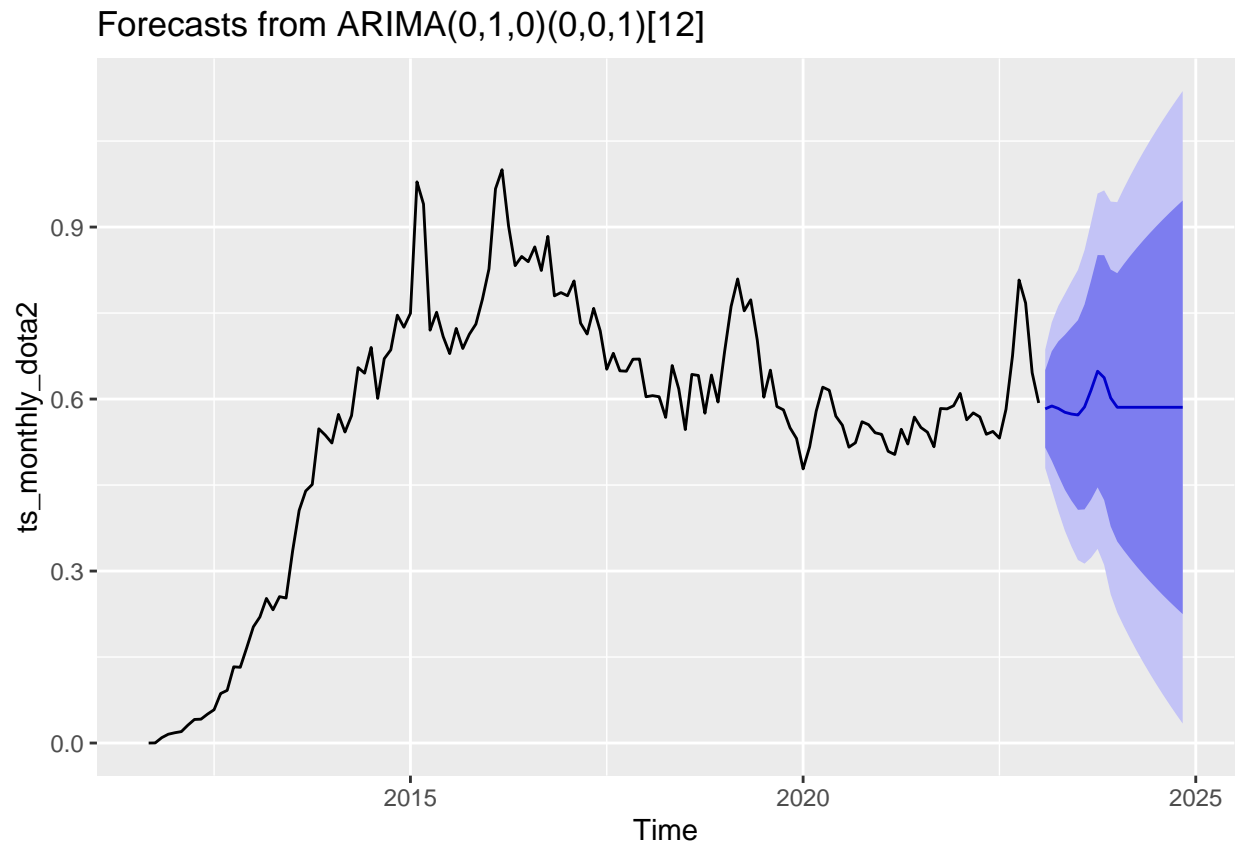
arima_monthly_dota2_df = data.frame(
  Date = test_dota2$Date,
  peak = predict_arima_monthly_dota2$mean
)
arima_monthly_dota2_df
```

```
##           Date      peak
## 1 2023-02-01 0.5828216
## 2 2023-03-01 0.5878676
## 3 2023-04-01 0.5835801
## 4 2023-05-01 0.5768939
## 5 2023-06-01 0.5738653
## 6 2023-07-01 0.5721969
## 7 2023-08-01 0.5861724
## 8 2023-09-01 0.6154958
## 9 2023-10-01 0.6485335
## 10 2023-11-01 0.6372927
## 11 2023-12-01 0.6018946
## 12 2024-01-01 0.5855859
## 13 2024-02-01 0.5855859
## 14 2024-03-01 0.5855859
## 15 2024-04-01 0.5855859
## 16 2024-05-01 0.5855859
## 17 2024-06-01 0.5855859
```



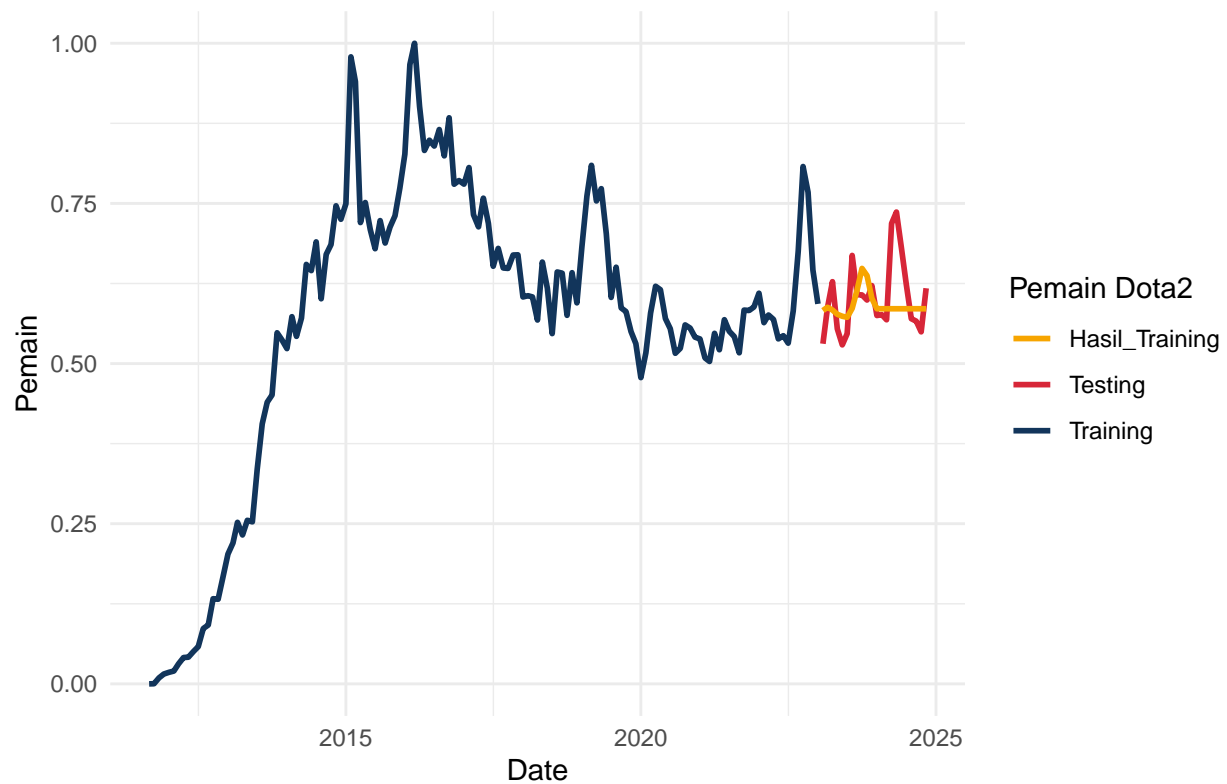
```
## 18 2024-07-01 0.5855859
## 19 2024-08-01 0.5855859
## 20 2024-09-01 0.5855859
## 21 2024-10-01 0.5855859
## 22 2024-11-01 0.5855859
```

```
autoplot(predict_arima_monthly_dota2)
```



```
ggplot() +
  geom_line(data = train_dota2, aes(x = Date, y = Peak, color = "Training"), size = 1) +
  geom_line(data = test_dota2, aes(x = Date, y = Peak, color = "Testing"), size = 1) +
  geom_line(data = arima_monthly_dota2_df, aes(x = Date, y = peak, color = "Hasil_Training"), size = 1)
labs(
  title = "Data Bulanan Pemain Dota2",
  x = "Date",
  y = "Pemain"
) +
scale_color_manual(values = c("Training" = "#12355B", "Testing" = "#D72638", "Hasil_Training" = "#f7a1d9"))
theme_minimal() +
theme(plot.title = element_text(size = 18))
```

## Data Bulanan Pemain Dota2



## Evaluasi

```
evaluasi_dota2 = data.frame(
  Date = test_dota2$Date,
  ARIMA = predict_arima_monthly_dota2$mean,
  Testing = test_dota2$Peak
)
nilai_mae = mae(evaluasi_dota2$Testing,evaluasi_dota2$ARIMA)
nilai_rmse = rmse(evaluasi_dota2$Testing,evaluasi_dota2$ARIMA)
nilai_mape = mape(evaluasi_dota2$Testing,evaluasi_dota2$ARIMA)

metrics_arima_monthly_dota2 = data.frame(
  MAE = nilai_mae,
  MAPE = nilai_mape,
  RMSE = nilai_rmse
)
metrics_arima_monthly_dota2
```

```
##           MAE           MAPE           RMSE
## 1 0.04259988 0.06713771 0.05745897
```