LAPORAN TUGAS 3

TOPIK DALAM ANALISIS DATA DERET WAKTU

GENERALIZED SPACE-TIME AUTOREGRESSIVE (GSTAR) PADA DATA SUHU RATA-RATA DI SURABAYA

Oleh:

Yolla Faradhilla **05111950010029**

Hafara Firdausi **05111950010040**

Dosen Pengampu:

Dr. Ahmad Saikhu, S.Si., M.T.

PROGRAM MAGISTER

JURUSAN INFORMATIKA

FAKULTAS ELEKTRO DAN INFORMATIKA CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2020

1. Deskripsi

Model *Generalized Space Time Autoregressive* (GSTAR) digunakan untuk memodelkan data *timeseries* yang juga mempunyai keterikatan antar lokasi. Salah satu permasalahan utama pada pemodelan GSTAR adalah menentukan bobot lokasi yang dapat membentuk model dengan kesalahan ramalan terkecil. Terdapat beberapa jenis bobot lokasi yang digunakan dalam GSTAR, empat diantaranya yaitu bobot seragam, bobot korelasi, bobot invers jarak, dan bobot biner.

Tujuan dari tugas ini adalah mendapatkan model GSTAR terbaik untuk peramalan berdasarkan tiga jenis bobot lokasi tersebut. Kami mengimplementasikan GSTAR menggunakan bahasa R, kemudian membandingkan hasil evaluasi dari ketiga bobot lokasi tersebut untuk mendapatkan model GSTAR terbaik.

2. Langkah Pengerjaan

2.1. Impor Dataset

Data spasiotemporal yang digunakan pada tugas ini adalah data suhu rata-rata dari Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) dalam kurun waktu 1 Januari 2019 hingga 12 Mei 2020 yang diukur dari tiga stasiun meteorologi yang berbeda di Surabaya, yaitu stasiun meteorologi Perak I, Perak II, dan Juanda. Jumlah data sebanyak 528 baris, diunduh dari http://dataonline.bmkg.go.id/.

	Tanggal	Perak.I	Perak.II	Juanda
	<chr>	<dbl>	<dbl>	<dbl>
1	01-01-2019	28.9	29.5	29.0
2	02-01-2019	29.0	29.2	28.5
3	03-01-2019	27.3	27.2	25.6
4	04-01-2019	27.7	28.0	27.6
5	05-01-2019	29.1	29.7	29.2
6	06-01-2019	30.4	30.4	28.9

	Tanggal	Perak.I	Perak.II	Juanda
	<chr>	<dbl>	<dbl>	<dbl>
523	07-05-2020	29.9	30.0	28.5
524	08-05-2020	30.6	30.3	29.5
525	09-05-2020	29.0	29.1	28.7
526	10-05-2020	29.4	29.3	29.0
527	11-05-2020	29.1	28.9	27.9
528	12-05-2020	29.8	29.5	NA

2.2. Mengecek Korelasi Data

Dataset yang diambil memiliki beberapa null value, sehingga jika dicek korelasinya maka hasilnya NA. Oleh karena itu, kami mengisi *null value* nya terlebih dahulu menggunakan rata-rata. Sehingga di dapatkan korelasi sebagai berikut.

```
> cor(data)
           Perak.I  Perak.II    Juanda
Perak.I  1.0000000 0.9495863 0.8814858
Perak.II 0.9495863 1.0000000 0.8819370
Juanda   0.8814858 0.8819370 1.0000000
```

2.3. Membagi Data

Data dibagi menjadi *train set* dan *test set* dengan perbandingan 80:20

```
Train set: 422
Test set: 106
```

2.4. Mendefinisikan Bobot

2.4.1 Bobot Seragam (*Uniform*)

Bobot seragam didefinisikan sebagai $W_{ij} = {}^{1}/_{n_i}$, n_i adalah banyaknya lokasi yang berdekatan dengan lokasi i. Bobot seragam memberikan nilai bobot yang sama untuk masing-masing lokasi. Oleh karena itu, bobot lokasi ini sering digunakan pada data yang mempunyai jarak antar lokasi yang sama (homogen). Matriks bobot seragam yang digunakan adalah sebagai berikut:

```
> weight_uniform
     [,1] [,2] [,3]
[1,]  0.0  0.5  0.5
[2,]  0.5  0.0  0.5
[3,]  0.5  0.5  0.0
```

2.4.2 Bobot Normalisasi Korelasi Silang

Matriks bobot korelasi menggunakan nilai korelasi antar lokasi, yakni sebagai berikut:

```
> weight_cor
           Perak.I  Perak.II    Juanda
Perak.I  0.5000000 0.4747931 0.4407429
Perak.II 0.4747931 0.5000000 0.4409685
Juanda   0.4407429 0.4409685 0.5000000
```

2.4.3 Bobot Invers Jarak

Pembobotan dengan invers jarak mengacu pada jarak antar lokasi. Jarak antara 3 lokasi stasiun meteorologi didefinisikan sebagai berikut:

- $r_1 =$ Jarak Perak I dan Perak II
- $r_2 =$ Jarak Perak I dan Juanda
- $r_3 =$ Jarak Perak II dan Juanda

$$\begin{matrix} 0 & W_{12} & W_{13} \\ W_{21} & 0 & W_{23} \\ W_{31} & W_{32} & 0 \end{matrix} = \begin{matrix} 0 & \frac{r_2}{r_1+r_2} & \frac{r_1}{r_1+r_2} \\ \frac{r_3}{r_1+r_3} & 0 & \frac{r_1}{r_1+r_3} \\ \frac{r_3}{r_2+r_3} & \frac{r_2}{r_2+r_3} & 0 \end{matrix}$$

Sehingga, matriks yang terbentuk adalah sebagai berikut:

```
          [,1]       [,2]      [,3]
[1,] 0.00000000 0.44796558 0.2598018
[2,] 0.44419283 0.00000000 0.2401982
[3,] 0.05580717 0.05203442 0.0000000
```

2.4.4 Bobot Biner

Bobot biner merupakan pembobotan dengan menggunakan nilai kategorik 0 dan 1. Dengan nilai 0 dan 1 tergantung pada batasan tertentu. Jarak lokasi terdekat bernilai 1, sedangkan jarak lokasi yang lebih jauh bernilai 0. Matriks bobot biner yang digunakan adalah sebagai berikut:

0	1	1
1	0	0
0	0	0

2.5. Evaluasi Model GSTAR

Setelah melakukan *training* dan *testing* model, model GSTAR dievaluasi menggunakan *metrics Mean Squared Error* (MSE) dan *Mean Absolute Percentage Error* (MAPE). Berikut hasil ringkasan dari evaluasi *training:*

Jenis Bobot	**Metrics**	
	MSE	**MAPE**
Seragam	0.5154806	1.900834
Korelasi	0.5154706	1.900645
Invers Jarak	0.5201234	1.904706
Biner	0.5263061	1.920047

Berikut hasil ringkasan dari evaluasi *testing:*

Jenis Bobot	**Metrics**	
	MSE	**MAPE**
Seragam	0.9360444	2.748507
Korelasi	0.9327662	2.743146
Invers Jarak	0.8413433	2.600861
Biner	**0.8267436**	**2.582107**

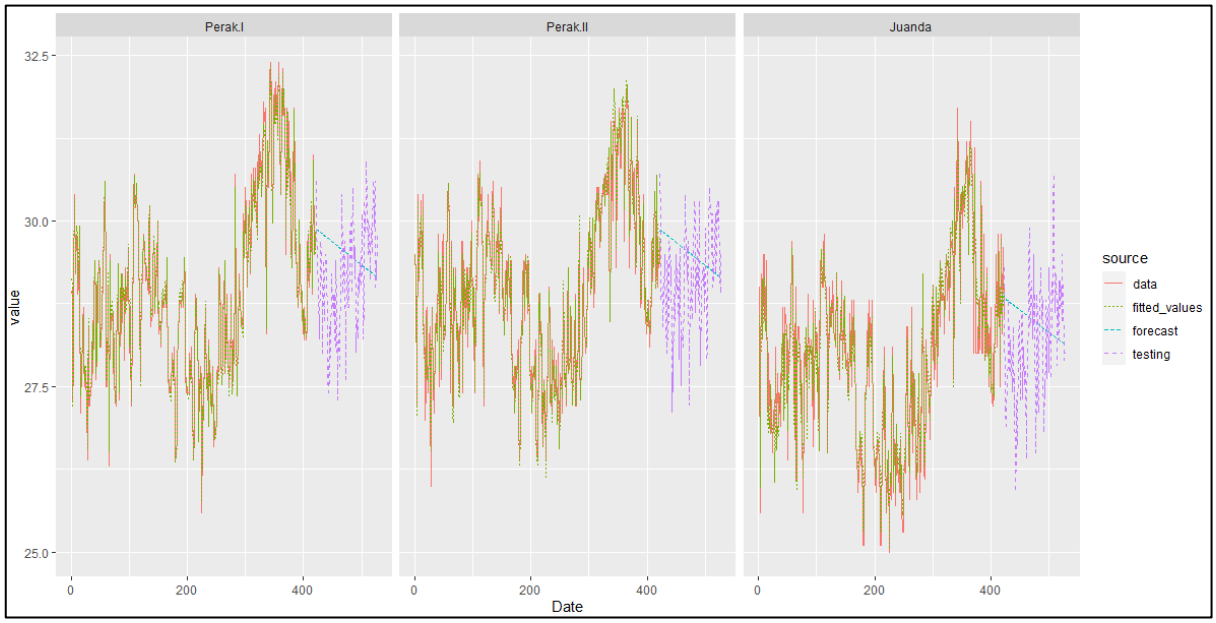
2.5.1. Bobot Seragam (*Uniform*)

2.5.1.1 Hasil *Training*

```
----------Performance training------------

MSE for all data =  0.5154806
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4822518 0.4870102 0.5771798

MAPE for all data =  1.900834
MAPE for each location :
 Perak.I Perak.II   Juanda
1.797284 1.854074 2.051144
```

2.5.1.2 Hasil *Testing*

```
----------Performance testing------------

MSE for all data =  0.9360444
MSE for each location :
  Perak.I  Perak.II    Juanda
0.9341923 0.8702308 1.0037101

MAPE for all data =  2.748507
MAPE for each location :
 Perak.I Perak.II   Juanda
2.762992 2.600131 2.882397
```



2.5.1.3 Prediksi 5 Data

```
> predict(fit, n = 5)
      Perak.I Perak.II   Juanda
[1,] 29.86634 29.84592 28.90100
[2,] 29.86712 29.86043 28.85422
[3,] 29.86332 29.85782 28.83606
[4,] 29.85772 29.85135 28.82586
[5,] 29.85136 29.84417 28.81797
```

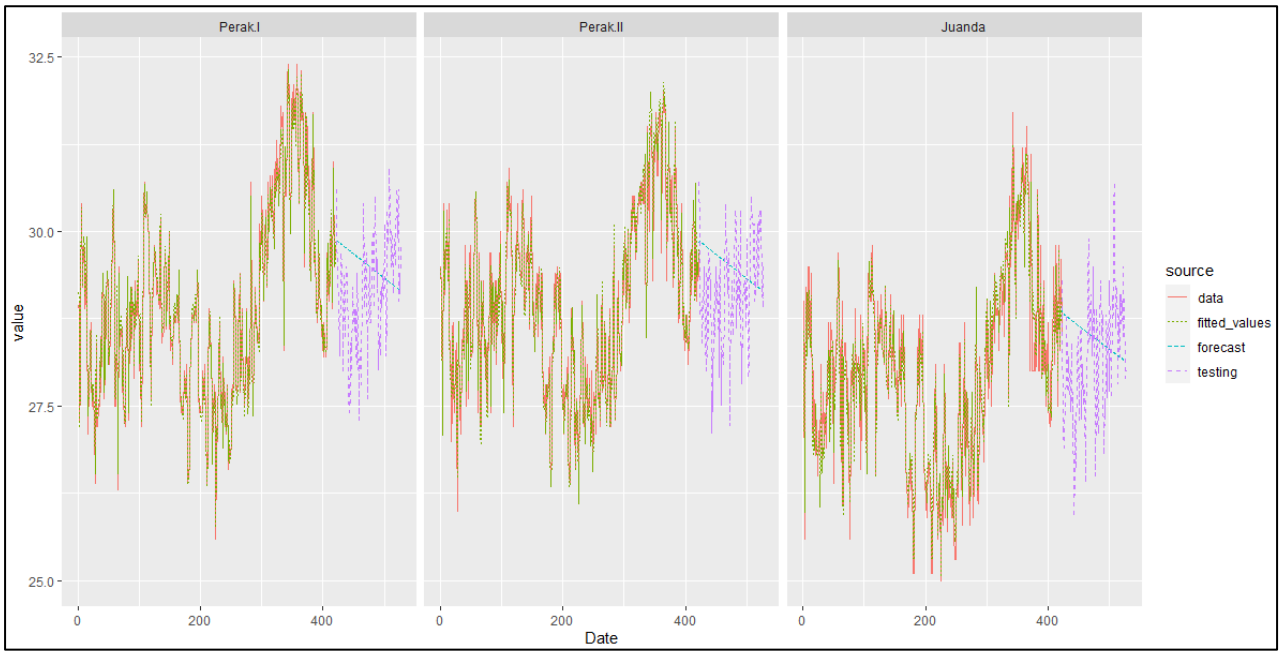
2.5.2. Bobot Korelasi

2.5.2.1 Hasil *Training*

```
----------Performance training------------

MSE for all data =  0.5154706
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4823492 0.4868837 0.5771789

MAPE for all data =  1.900645
MAPE for each location :
 Perak.I Perak.II   Juanda
1.797398 1.853399 2.051139
```

2.5.2.2 Hasil *Testing*

```
----------Performance testing------------

MSE for all data =  0.9327662
MSE for each location :
  Perak.I  Perak.II    Juanda
0.9307215 0.8667233 1.0008539

MAPE for all data =  2.743146
MAPE for each location :
 Perak.I Perak.II   Juanda
2.757630 2.593201 2.878607
```



2.5.2.3 Prediksi 5 Data

```
> predict(fit2, n = 5)
      Perak.I Perak.II   Juanda
[1,] 29.86018 29.84209 28.89877
[2,] 29.86142 29.85608 28.85062
[3,] 29.85798 29.85321 28.83180
[4,] 29.85259 29.84660 28.82131
[5,] 29.84634 29.83938 28.81331
```

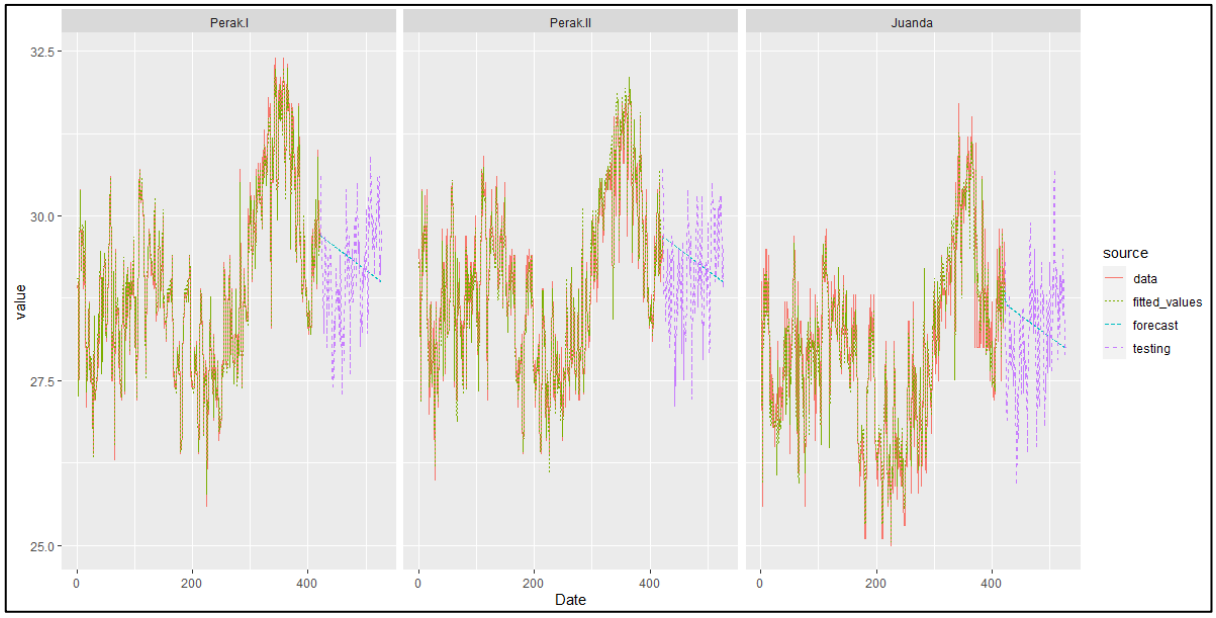
2.5.3. Bobot Invers Jarak

2.5.3.1 Hasil *Training*

```
----------Performance training------------

MSE for all data =  0.5201234
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4874847 0.4955482 0.5773372

MAPE for all data =  1.904706
MAPE for each location :
 Perak.I Perak.II   Juanda
1.797519 1.864729 2.051869
```

2.5.3.2 Hasil *Testing*

```
----------Performance testing------------

MSE for all data =  0.8413433
MSE for each location :
  Perak.I  Perak.II    Juanda
0.8334035 0.7631673 0.9274590

MAPE for all data =  2.600861
MAPE for each location :
 Perak.I Perak.II   Juanda
2.592533 2.411467 2.798583
```



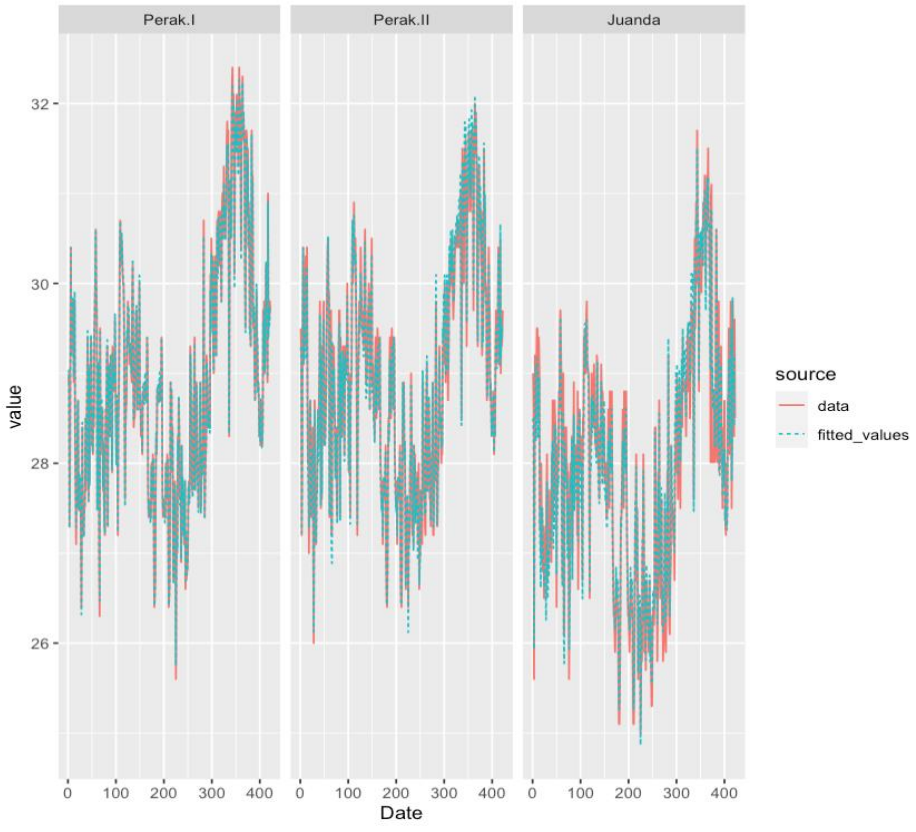
2.5.3.3 Prediksi 5 Data

```
> predict(fit3, n = 5)
      Perak.I Perak.II   Juanda
[1,] 29.70760 29.64885 28.83002
[2,] 29.69430 29.66885 28.73078
[3,] 29.68558 29.67201 28.68522
[4,] 29.67833 29.66891 28.66215
[5,] 29.67152 29.66349 28.64856
```

2.5.4. Bobot Biner

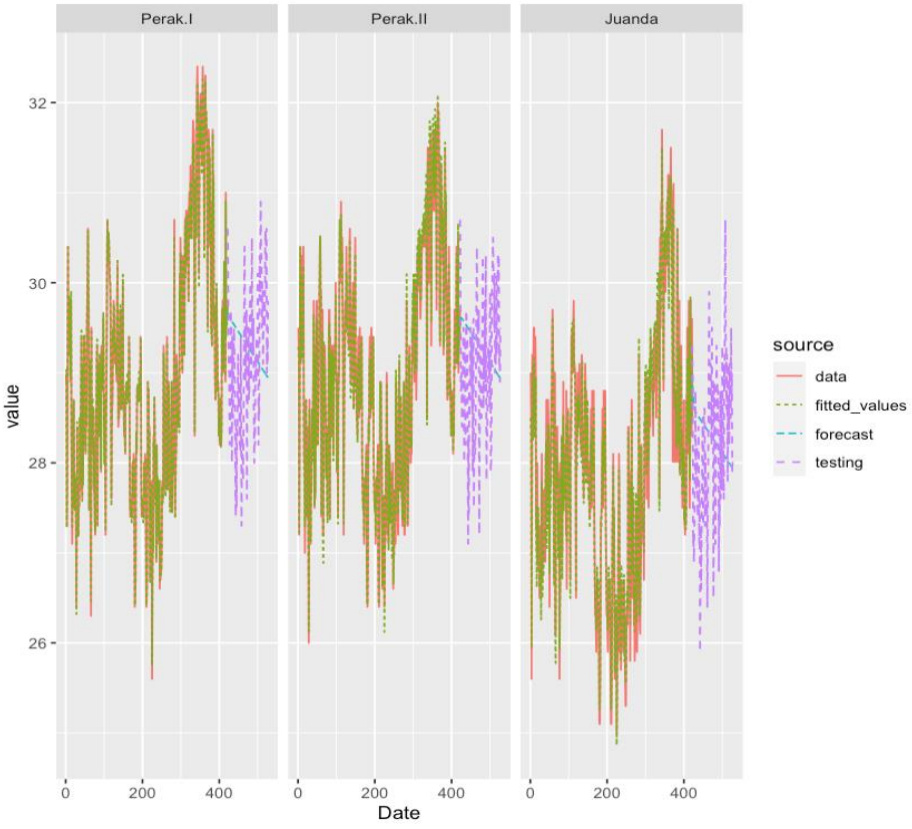
2.5.4.1 Hasil *Training*

```
----------Performance training-----------

MSE for all data =  0.5263061
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4896199 0.5001006 0.5891979

MAPE for all data =  1.920047
MAPE for each location :
 Perak.I Perak.II   Juanda
1.798299 1.873562 2.088281
```



2.5.4.2 Hasil *Testing*

```
----------Performance testing------------

MSE for all data =  0.8267436
MSE for each location :
  Perak.I  Perak.II    Juanda
0.8156797 0.7427546 0.9217963

MAPE for all data =  2.582107
MAPE for each location :
 Perak.I Perak.II   Juanda
2.562141 2.387646 2.796535
```



2.5.4.3 Prediksi 5 Data

Perak.I	Perak.II	Juanda
29.67604	29.58148	28.97457
29.65266	29.60409	28.82107
29.63809	29.61116	28.72627
29.62767	29.61091	28.66766
29.61919	29.60723	28.63081

2.6. Kesimpulan

Berdasarkan hasil evaluasi di atas, **bobot biner mendapatkan hasil MSE dan MAPE yang paling kecil** pada saat *testing* dibandingkan bobot lainnya. Sehingga model GSTAR yang paling baik adalah model yang menggunakan bobot biner.

3. Referensi

https://www.rdocumentation.org/packages/gstar/versions/0.1.0/topics/gstar

1. Import Dataset

In [1]:

```
# import libraries
library(xlsx)
library(Metrics)
library(ggplot2)
library(gstar)
```

In [2]:

```
# import data
data <- read.xlsx(("data-temperature-2019-2020.xlsx"),1, header=TRUE)
head(data)
tail(data)
```

A data.frame: 6 × 4

	Tanggal	Perak.I	Perak.II	Juanda
	<chr>	<dbl>	<dbl>	<dbl>
1	01-01-2019	28.9	29.5	29.0
2	02-01-2019	29.0	29.2	28.5
3	03-01-2019	27.3	27.2	25.6
4	04-01-2019	27.7	28.0	27.6
5	05-01-2019	29.1	29.7	29.2
6	06-01-2019	30.4	30.4	28.9

A data.frame: 6 × 4

	Tanggal	Perak.I	Perak.II	Juanda
	<chr>	<dbl>	<dbl>	<dbl>
523	07-05-2020	29.9	30.0	28.5
524	08-05-2020	30.6	30.3	29.5
525	09-05-2020	29.0	29.1	28.7
526	10-05-2020	29.4	29.3	29.0
527	11-05-2020	29.1	28.9	27.9
528	12-05-2020	29.8	29.5	NA

In [3]:

```
# mendapatkan jumlah data
nrow(data)
```

528

In [4]:

```
# drop kolom tanggal
data$Tanggal <- NULL
head(data)
```

A data.frame: 6 × 3

Perak.I	Perak.II	Juanda
<dbl>	<dbl>	<dbl>

	Perak.I	Perak.II	Juanda
	<dbl>	<dbl>	<dbl>
1	28.9	29.5	29.0
2	29.0	29.2	28.5
3	27.3	27.2	25.6
4	27.7	28.0	27.6
5	29.1	29.7	29.2
6	30.4	30.4	28.9

In [5]:

```
# mendapatkan korelasi
cor(data)
```

A matrix: 3 × 3 of type dbl

	Perak.I	Perak.II	Juanda
Perak.I	1	NA	NA
Perak.II	NA	1	NA
Juanda	NA	NA	1

In [6]:

```
# karena hasilnya NA, maka perlu mengisi null value dengan rata-rata

data$Juanda[is.na(data$Juanda)] <- mean(data$Juanda, na.rm = TRUE)
data$Perak.I[is.na(data$Perak.I)] <- mean(data$Perak.I, na.rm = TRUE)
data$Perak.II[is.na(data$Perak.II)] <- mean(data$Perak.II, na.rm = TRUE)
```

In [7]:

```
# mendapatkan korelasi
cor(data)
```

A matrix: 3 × 3 of type dbl

	Perak.I	Perak.II	Juanda
Perak.I	1.0000000	0.9495863	0.8814858
Perak.II	0.9495863	1.0000000	0.8819370
Juanda	0.8814858	0.8819370	1.0000000

In [8]:

```
## split data menjadi training and testing (80:20)

split_data <- round(nrow(data) * 0.8)
x_train <- data[1:split_data, ]
x_test <- data[-c(1:split_data),]

cat("Train set:",  nrow(x_train), "\n")
cat("Test set:",  nrow(x_test))
```

```
Train set: 422
Test set: 106
```

2. Mendefinisikan Bobot

Kami membandingkan empat jenis bobot:

- Bobot seragam (uniform)
- Bobot korelasi
- Bobot invers jarak
- Bobot biner

2.1 Bobot Seragam (Uniform)

In [9]:

```
################
# bobot seragam
################

weight_uniform = matrix(c(0,1,1,
                          1,0,1,
                          1,1,0), ncol = 3, nrow = 3)

# jumlah bobot adalah 1 untuk setiap baris
weight_uniform = weight_uniform/(ncol(data) - 1)

weight_uniform
```

A matrix: 3 × 3
of type dbl

0.0	0.5	0.5
0.5	0.0	0.5
0.5	0.5	0.0

2.2 Bobot Korelasi

In [10]:

```
################
# bobot korelasi
################

weight_cor = cor(data)

# jumlah bobot adalah 1 untuk setiap baris
weight_cor = weight_cor/(ncol(data) - 1)

weight_cor
```

A matrix: 3 × 3 of type dbl

	Perak.I	Perak.II	Juanda
Perak.I	0.5000000	0.4747931	0.4407429
Perak.II	0.4747931	0.5000000	0.4409685
Juanda	0.4407429	0.4409685	0.5000000

2.3 Bobot Invers Jarak

In [11]:

```
################
# bobot invers jarak
################

# fungsi hitung euclidean
euclidean_dist <- function(locA, locB){
    dist <- sqrt((locB[1]-locA[1])^2 + (locB[2]-locA[2])^2)
    return(dist)
}
```

In [12]:

```
# define jarak
```

```
loc_perak_1 = c(-7.22360, 112.72390)
loc_perak_2 = c(-7.20530, 112.73530)
loc_juanda = c(-7.38460, 112.78330)

# r1 jarak Perak I - Perak II
r1 <- euclidean_dist(loc_perak_1, loc_perak_2)
# r2 jarak Perak I - Juanda
r2 <- euclidean_dist(loc_perak_1, loc_juanda)
# r2 jarak Perak II - Juanda
r3 <- euclidean_dist(loc_perak_2, loc_juanda)

cat("Jarak Perak I - Perak II:", r1, "\n")
cat("Jarak Perak I - Juanda:", r2, "\n")
cat("Jarak Perak II - Juanda:", r3, "\n")
```

```
Jarak Perak I - Perak II: 0.02156038
Jarak Perak I - Juanda: 0.1716082
Jarak Perak II - Juanda: 0.1856138
```

In [13]:

```
# hitung bobot berdasar invers jarak
w12 <- r2/(r1+r2)
w13 <- r1/(r1+r2)
w21 <- r3/(r1+r3)
w23 <- r1/(r1+r3)
w31 <- r3/(r2+r3)
w32 <- r2/(r2+r3)

weight_dist = matrix(c(0,w12,w13,
                       w21,0,w23,
                       w31,w32,0), ncol = 3, nrow = 3)

# jumlah bobot adalah 1 untuk setiap baris
weight_dist = weight_dist/(ncol(data) - 1)

weight_dist
```

A matrix: 3 × 3 of type dbl

0.00000000	0.44796558	0.2598018
0.44419283	0.00000000	0.2401982
0.05580717	0.05203442	0.0000000

2.4 Bobot Biner

In [14]:

```
#################
# bobot biner
#################

# define fungsi untuk mendapatkan min value
is_less_than <- function(distA, distB){
    if(distA < distB) {
        return (1)
    } else {
        return (0)
    }
}
```

In [15]:

```
# menggunakan jarak yang telah dihitung sebelumnya pada poin 2.3
wb12 <- is_less_than(r1, r2)
wb13 <- is_less_than(r2, r1)
wb21 <- is_less_than(r1, r3)
wb23 <- is_less_than(r3, r1)
wb31 <- is_less_than(r2, r3)
wb32 <- is_less_than(r3, r2)
```

```
weight_biner = matrix(c(0,wb12,wb13,
                        wb21,0,wb23,
                        wb31,wb32,0), ncol = 3, nrow = 3)

weight_biner
```

A matrix:
3 × 3 of
type dbl

0	1	1
1	0	0
0	0	0

3. Train Model

3.1 Bobot Seragam (Uniform)

In [16]:

```
# train model dengan bobot seragam
fit <-  gstar(x_train, weight = weight_uniform, p = 1, d = 0, est = "OLS")
summary(fit)
```

```
Coefficients:
                Estimate Std.Err t value Pr(>|t|)
psi10(Perak.I)    0.7335  8.1693   0.090    0.928
psi10(Perak.II)   0.5482  9.8647   0.056    0.956
psi10(Juanda)     0.4716  5.0730   0.093    0.926
psi11(Perak.I)    0.2710  8.4600   0.032    0.974
psi11(Perak.II)   0.4594 10.2108   0.045    0.964
psi11(Juanda)     0.5099  4.7310   0.108    0.914


AIC: 2761
```

3.2 Bobot Korelasi

In [17]:

```
# train model dengan bobot korelasi
fit2 <-  gstar(x_train, weight = weight_cor, p = 1, d = 0, est = "OLS")
summary(fit2)
```

```
Warning message in gstar(x_train, weight = weight_cor, p = 1, d = 0, est = "OLS"):
“the sum of weight is equal to 1 every row”
```

```
Coefficients:
                Estimate Std.Err t value Pr(>|t|)
psi10(Perak.I)    0.5827 20.1828   0.029    0.977
psi10(Perak.II)   0.2891 24.3336   0.012    0.991
psi10(Juanda)     0.1824 12.1496   0.015    0.988
psi11(Perak.I)    0.2979 10.2942   0.029    0.977
psi11(Perak.II)   0.5074 12.4034   0.041    0.967
psi11(Juanda)     0.5784  6.0856   0.095    0.924


AIC: 2761
```

3.3 Bobot Invers Jarak

In [18]:

```
# train model dengan bobot invers jarak
```

```
# train model dengan bobot invers jarak
fit3 <-  gstar(x_train, weight = weight_dist, p = 1, d = 0, est = "OLS")
summary(fit3)
```

```
Coefficients:
               Estimate Std.Err t value Pr(>|t|)
psi10(Perak.I)    0.7708  9.9810   0.077    0.938
psi10(Perak.II)   0.6022 10.5249   0.057    0.954
psi10(Juanda)     0.4728  5.1043   0.093    0.926
psi11(Perak.I)    0.4598 40.2459   0.011    0.991
psi11(Perak.II)   0.7978 42.3902   0.019    0.985
psi11(Juanda)     1.0175 19.0408   0.053    0.957


AIC: 2773
```

3.4 Bobot Biner

In [19]:

```
# train model dengan bobot biner
fit4 <-  gstar(x_train, weight = weight_biner, p = 1, d = 0, est = "OLS")
summary(fit4)
```

```
Coefficients:
               Estimate Std.Err t value Pr(>|t|)
psi10(Perak.I)    0.8084  9.5440   0.085    0.933
psi10(Perak.II)   0.6619  9.5472   0.069    0.945
psi10(Juanda)     0.5518  4.3400   0.127    0.899
psi11(Perak.I)    0.1914  9.5472   0.020    0.984
psi11(Perak.II)   0.3378  9.5440   0.035    0.972
psi11(Juanda)     0.4325  4.0466   0.107    0.915


AIC: 2788
```

4. Evaluate Model

4.1 Bobot Seragam (Uniform)

In [20]:

```
# cek performa model dengan bobot seragam
performance(fit, x_test)
```

```
----------Performance training------------

MSE for all data =  0.5154806
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4822518 0.4870102 0.5771798

MAPE for all data =  1.900834
MAPE for each location :
 Perak.I Perak.II   Juanda
1.797284 1.854074 2.051144

----------Performance testing------------

MSE for all data =  0.9360444
MSE for each location :
  Perak.I  Perak.II    Juanda
0.9341923 0.8702308 1.0037101

MAPE for all data =  2.748507
MAPE for each location :
 Perak.I Perak.II   Juanda
2.762992 2.600131 2.882397
```

4.2 Bobot Korelasi

In [21]:

```
# cek performa model dengan bobot korelasi
performance(fit2, x_test)
```

```
----------Performance training------------

MSE for all data =  0.5154706
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4823492 0.4868837 0.5771789

MAPE for all data =  1.900645
MAPE for each location :
 Perak.I Perak.II   Juanda
1.797398 1.853399 2.051139

----------Performance testing------------

MSE for all data =  0.9327662
MSE for each location :
  Perak.I  Perak.II    Juanda
0.9307215 0.8667233 1.0008539

MAPE for all data =  2.743146
MAPE for each location :
 Perak.I Perak.II   Juanda
2.757630 2.593201 2.878607
```

4.3 Bobot Invers Jarak

In [22]:

```
# cek performa model dengan bobot invers jarak
performance(fit3, x_test)
```

```
----------Performance training------------

MSE for all data =  0.5201234
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4874847 0.4955482 0.5773372

MAPE for all data =  1.904706
MAPE for each location :
 Perak.I Perak.II   Juanda
1.797519 1.864729 2.051869

----------Performance testing------------

MSE for all data =  0.8413433
MSE for each location :
  Perak.I  Perak.II    Juanda
0.8334035 0.7631673 0.9274590

MAPE for all data =  2.600861
MAPE for each location :
 Perak.I Perak.II   Juanda
2.592533 2.411467 2.798583
```

4.4 Bobot Biner

In [23]:

```
# cek performa model dengan bobot biner
performance(fit4, x_test)
```

```
----------Performance training------------
```

```
----------Performance training------------

MSE for all data =  0.5263061
MSE for each location :
  Perak.I  Perak.II    Juanda
0.4896199 0.5001006 0.5891979

MAPE for all data =  1.920047
MAPE for each location :
 Perak.I Perak.II   Juanda
1.798299 1.873562 2.088281

----------Performance testing------------

MSE for all data =  0.8267436
MSE for each location :
  Perak.I  Perak.II    Juanda
0.8156797 0.7427546 0.9217963

MAPE for all data =  2.582107
MAPE for each location :
 Perak.I Perak.II   Juanda
2.562141 2.387646 2.796535
```

5. Predict

5.1 Bobot Seragam (Uniform)

In [24]:

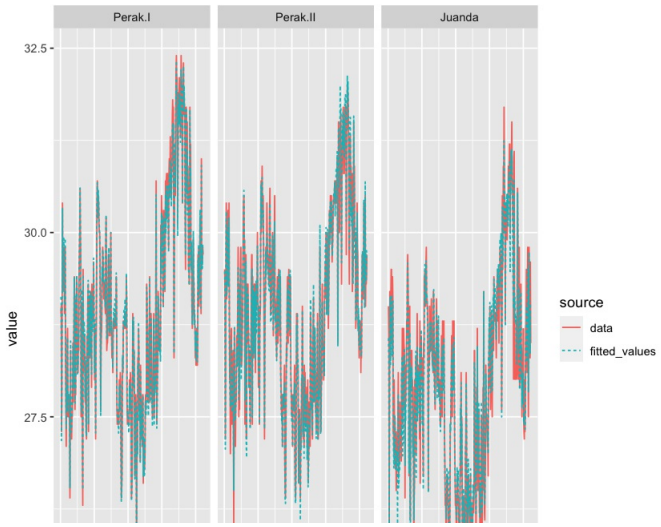
```
# forecast 5 data ahead
predict(fit, n = 5)
```

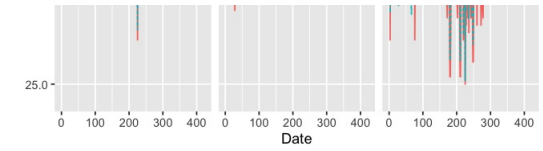
A matrix: 5 × 3 of type dbl

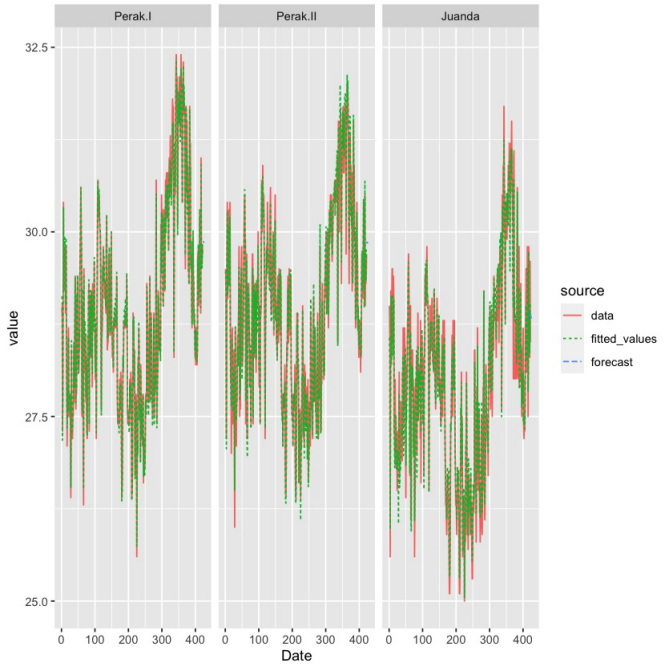
Perak.I	Perak.II	Juanda
29.86634	29.84592	28.90100
29.86712	29.86043	28.85422
29.86332	29.85782	28.83606
29.85772	29.85135	28.82586
29.85136	29.84417	28.81797

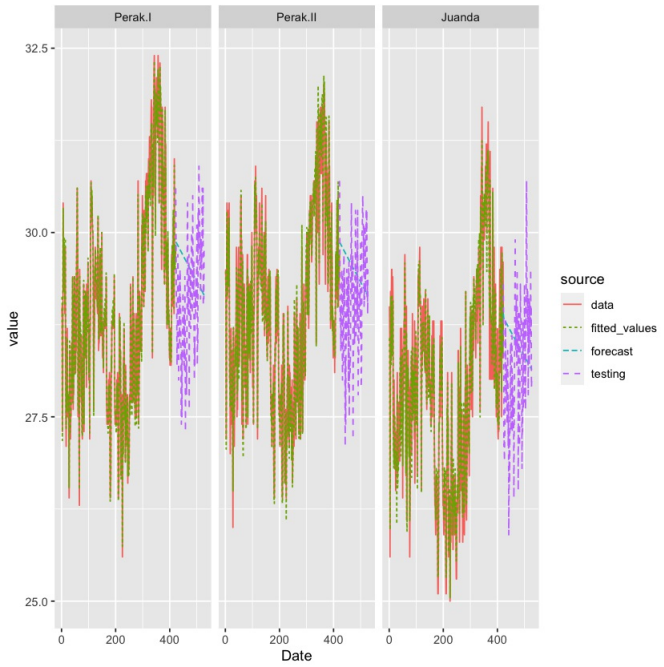
In [25]:

```
# plot with 5 forecasting data
plot(fit)
plot(fit, n_predict = 5)
plot(fit, testing = x_test)
```









5.2 Bobot Korelasi

In [26]:

```
# forecast 5 data ahead
predict(fit2, n = 5)
```

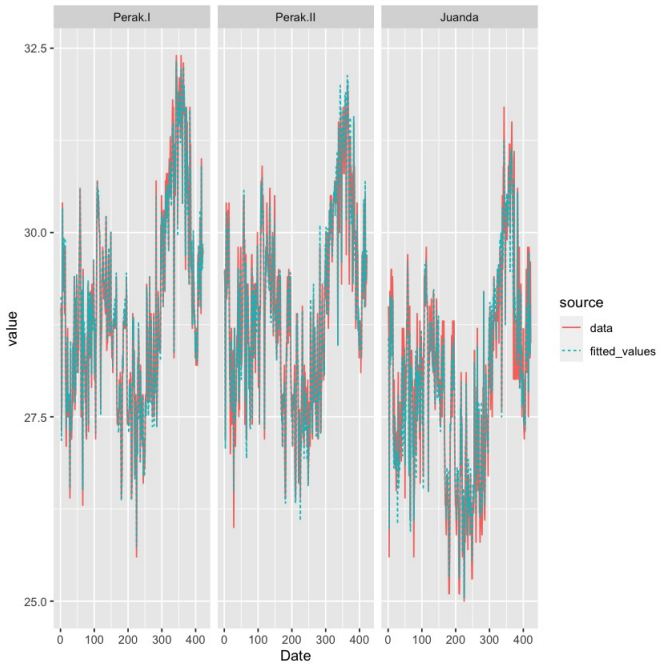
A matrix: 5 × 3 of type dbl

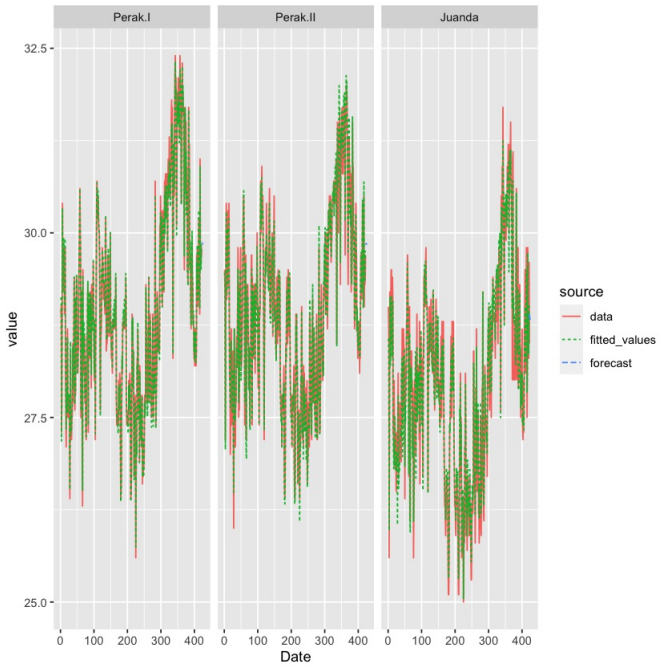
Perak.I	Perak.II	Juanda
29.86018	29.84209	28.89877
29.86142	29.85608	28.85062
29.85798	29.85321	28.83180

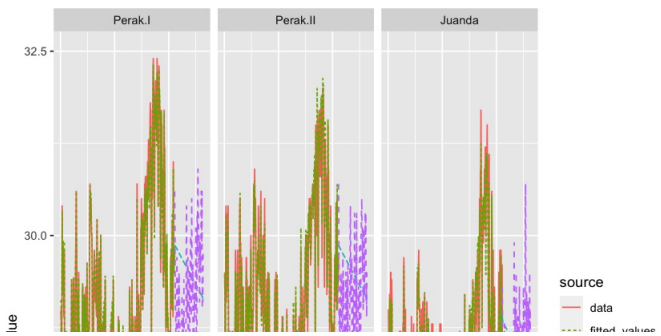
Perak.I	Perak.II	Juanda
29.84634	29.83938	28.81331

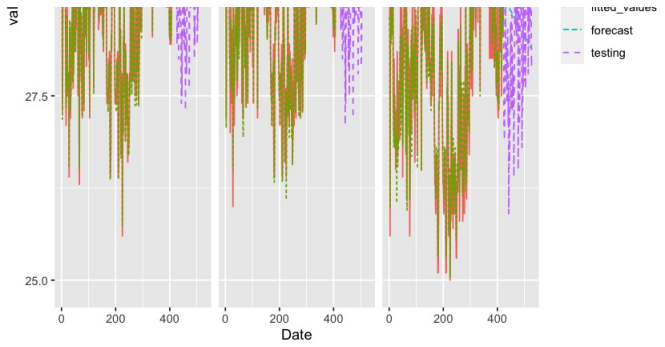
In [27]:

```
# plot with 5 forecasting data
plot(fit2)
plot(fit2, n_predict = 5)
plot(fit2, testing = x_test)
```









5.3 Bobot Invers Jarak

In [28]:

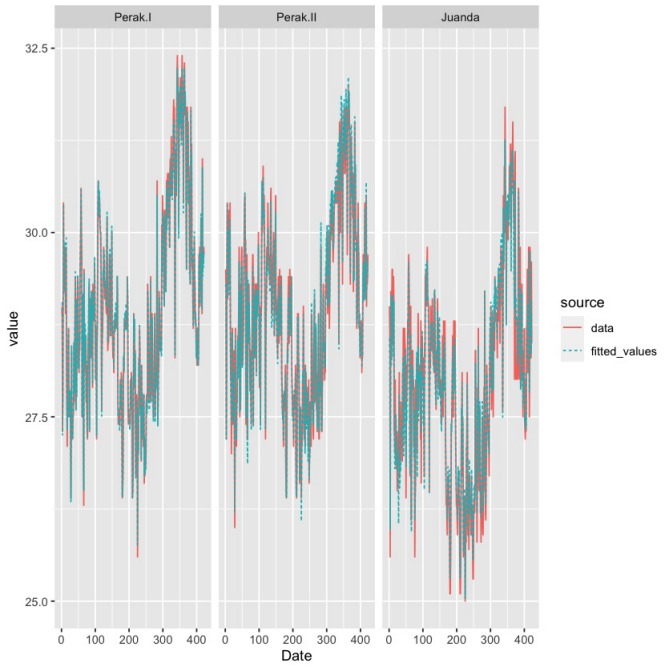
```
# forecast 5 data ahead
predict(fit3, n = 5)
```

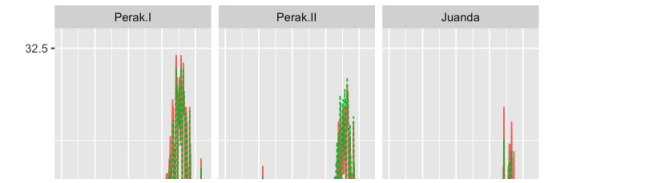
A matrix: 5 × 3 of type dbl

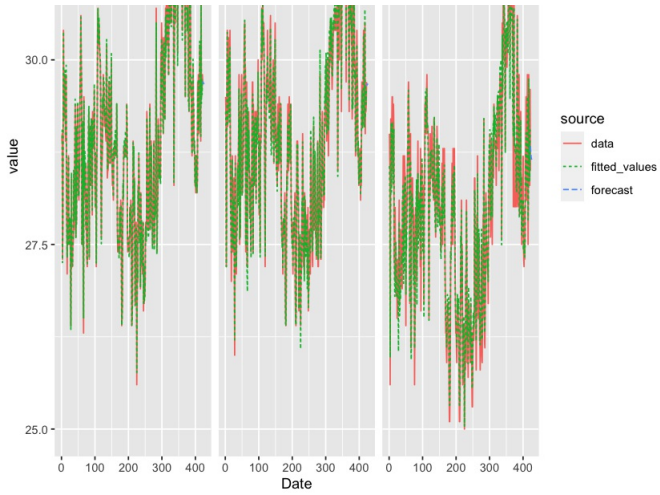
Perak.I	Perak.II	Juanda
29.70760	29.64885	28.83002
29.69430	29.66885	28.73078
29.68558	29.67201	28.68522
29.67833	29.66891	28.66215
29.67152	29.66349	28.64856

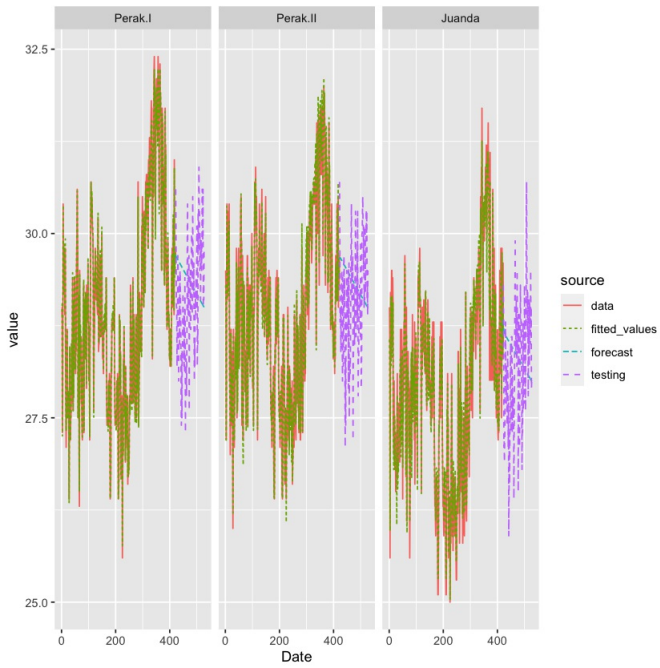
In [29]:

```
# plot with 10 forecasting data
plot(fit3)
plot(fit3, n_predict = 5)
plot(fit3, testing = x_test)
```









5.4 Bobot Biner

In [30]:

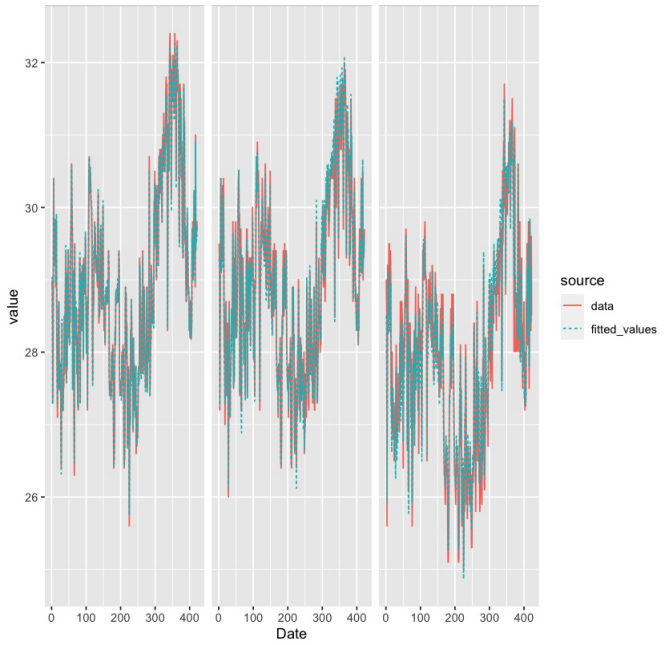
```
# forecast 5 data ahead
predict(fit4, n = 5)
```

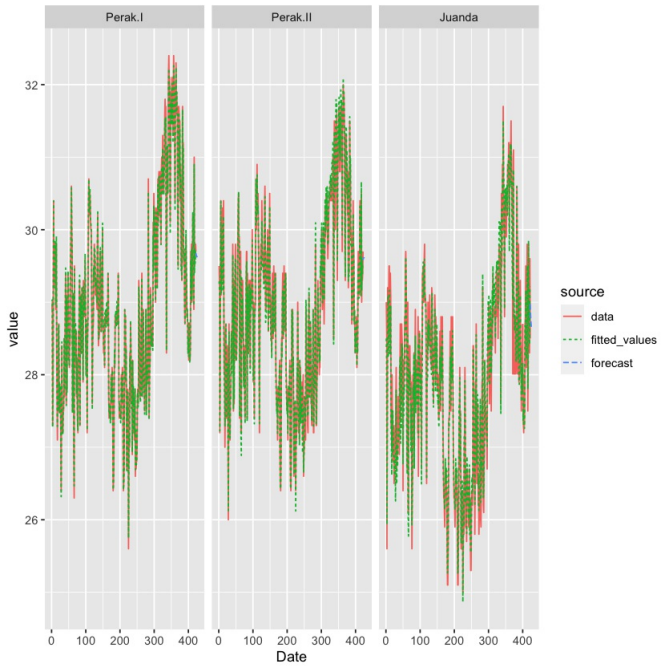
A matrix: 5 × 3 of type dbl

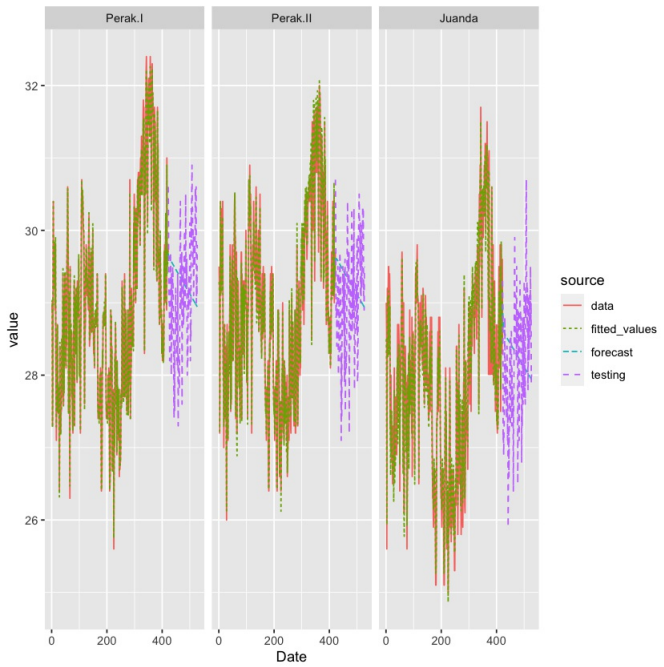
Perak.I	Perak.II	Juanda
29.67604	29.58148	28.97457
29.65266	29.60409	28.82107
29.63809	29.61116	28.72627
29.62767	29.61091	28.66766
29.61919	29.60723	28.63081

In [31]:

```
# plot with 5 forecasting data
plot(fit4)
plot(fit4, n_predict = 5)
plot(fit4, testing = x_test)
```


value
32
30
28
26
0
100
200
300
400
Date
source
data
fitted_values


Perak.I
Perak.II
Juanda
value
32
30
28
26
0
100
200
300
400
Date
source
data
fitted_values
forecast


Perak.I
Perak.II
Juanda
value
32
30
28
26
0
200
400
Date
source
data
fitted_values
forecast
testing

References

- https://www.rdocumentation.org/packages/gstar/versions/0.1.0/topics/gstar