

Variogram

Jeremy Fontaine Chaim

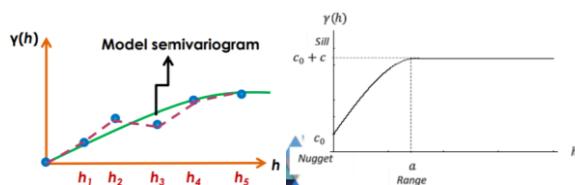
I. Abstrak

Geostatistik adalah cabang ilmu statistika yang menggunakan analisis matematika dan geologi dalam penerapannya. Variogram adalah alat utama dalam ilmu geostatistik. Fungsi dari variogram atau semivariogram adalah menunjukkan korelasi spasial antar data. Data spasial adalah data yang beraserensi pada keadaan geografis bumi. Data tersebut dapat dipandang sebagai koleksi peubah acak dengan sifat $E[Z(s) - Z(s + h)] = 0, \quad \forall s \in D$ (ekspektasi dari 2 data berjarak h bernilai nol) dan $Var[Z(s) - Z(s + h)] = 2\gamma(h)$ (variansi 2 peubah acak hanya bergantung pada jarak antar 2 peubah acak tersebut).

Keterangan:

- s_i : lokasi i , berupa koordinat (x_i, y_i)
- h_k : jarak antar pasangan lokasi dihitung melalui pasangan koordinat lokasi.
- $z(s_i)$: nilai observasi pada lokasi i . Contoh : kandungan minyak bumi

Sebagaimana sudah disebutkan, variabilitas dari peubah acak dapat diukur dengan menghitung variogram (atau semivariogram). Variogram mengukur variabilitas data dengan parameter jarak dengan asumsi hanya jarak antar data yang membuat variabilitas terjadi, sehingga diperoleh: $2\gamma(h) = Var[Z(s) - Z(s + h)]$ dan jika $Z(s_i)$ diketahui maka dapat dihitung variabilitas eksperimental dengan persamaan $\hat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [Z(s_i) - Z(s_i + h)]^2$ dimana h adalah jarak sepasang data dan $N(h)$ adalah banyak pasangan data yang berjarak h . Dari perhitungan tersebut dapat dibandung model variogram dengan sifat-sifat parameter berikut:



1. Variansi selalu monoton naik terhadap satuan jarak antar bagian.
2. Terdapat range dan sill.
3. Terdapat Nugget Variance

Bentuk-bentuk dari model variogram sendiri ada banyak tetapi empat diantaranya yang sering digunakan adalah model linera, spherical, eksponensial, dan gauss.

Kriging adalah metode geostatistik yang digunakan untuk mengestimasi nilai dari sebuah titik (prediksi intrapoliasi). $Var[Z(s_0) - \hat{Z}(s_0)]$ Persamaan kriging biasa adalah $\hat{Z}(s_0) = \sum_{i=1}^n \lambda_i Z(s_i)$ dengan s_0 adalah lokasi tak terobservasi, s_i lokasi terobservasi, $Z(s_i)$ nilai peubah acak pada lokasi s_i , dan λ_i adalah nilai kriging untuk observasi ke-i. Agar prediktor kriging tak bias, penjumlahan seluruh nilai kriging harus bernilai satu dan variansi galat taksiran, bernilai minimum (diselesaikan dengan metode pengali lagrange).

II. Syntax R

Dalam sintaks, akan dibahas data mengenai sumur minyak di Jatibarang. Data diambil dari kolom *k-fracture* yang menunjukkan nilai permeabilitas dalam satuan mili D'arcy (mD).

No	No Sumur Jatibarang	Koordinat lokasi		DZ	Variabel K-Fracture	Pormatrik
		x (km)	y (km)			
8	52	0.6309	-1.3109	213	35.445	12.16
18	62	0.9241	-1.2761	291	26.645	13.53
28	72	0.5677	-1.0127	291	51.72	15.13
38	86	1.0561	-0.8133	388	14.331	10.62
48	95	0.129	-0.5951	138	13.333	8.43
58	107	0.9848	-0.3901	108	4.683	5.57
68	119	1.2489	-0.4034	168	49.754	15.32
78	134	1.22	-0.6375	256	35.832	10.48
88	145	0.351	-0.9461	239	177.021	19.24
98	154	0.5042	-1.5291	238	30.467	12.67
108	164	0.5174	-1.2708	326	26.741	12.33
118	175	0.4646	-0.6614	181	16.776	14.08
128	186	0.4884	-0.9009	359	40.154	14.16

Gambar 2.1 Data sumur minyak di Jatibarang

Berikut adalah sintaks untuk mengolah data awal (data mentah).

```
#input data
library(readxl)
DATA_VARIOGRAM <- read_excel("D:/praktikum 9 analisis spasial/DATA VARIOGRAM.xlsx")
data <- read_xlsx("D:/praktikum 9 analisis spasial/DATA VARIOGRAM.xlsx", sheet="Sumur Jatibarang")
datasumurjtb <- data.frame(data)

#Mengaktifkan library
library(sp)
library(gstat)
library(dplyr)
library(ggplot2)
library(mapprojects)

#Persiapan data
# 1. mengubah data frame menjadi spatial point data frame
coordinates(datasumurjtb) = ~x+y
glimpse(datasumurjtb)

# 2. Membuat diagram pencar dengan ukuran titik menunjukkan
# besar kecilnya nilai di lokasi tersebut
ggplot(data = as.data.frame(datasumurjtb), aes(x,y)) +
  geom_point(aes(size = k.fracture), col = "blue", alpha = 0.6) +
  ggtitle("Permeabilitas (mD)") + coord_equal() + theme_bw()
```

Gambar 2.2 Sintaks mempersiapkan data

Sintaks ini dilakukan untuk mengambil data dari kolom excell yaitu koordinat dan nilai *k-fracture* kemudian melakukan scatter plot untuk melihat besarnya *k-fracture* pada koordinat tersebut.

Selanjutnya, data diolah untuk dicari semivariogram yang paling baik. Setelah itu, data akan diplot dan dicari model yang paling baik merepresentasikan data.

```

# mencari dan memplot semivariogram
# 1. mencari semivariogram eksperimental
(vgm1 <- variogram(k.fracture ~ 1, datasumurjtb))
plot(vgm1, cex = 1)

# 2. memperlebar interval semivariogram sehingga pasangan data pada suatu kelas h semakin banyak
(vgm2 <- variogram(k.fracture ~ 1, datasumurjtb, width = 0.05))
plot(vgm2, cex = 1)

# 3. mentransformasi semivariogram (misal dgn ln)
(vgm.ln.1 <- variogram(log(k.fracture) ~ 1, datasumurjtb))
plot(vgm.ln.1, main = "var.ln", cex = 1)

(vgm.ln.2 <- variogram(log(k.fracture)~1, datasumurjtb, width = 0.05))
plot(vgm.ln.2, main = "var.ln wdt = 0.05" ,cex=1)

# Melihat contoh-contoh model variogram yang dapat dimodelkan R melalui fungsi vgm
show.vgms()
vgm()

# Menampilkan model variogram eksperimental terbaik diantara model (Exp", "Sph", "Gau")
# Fitting variogram (mencari model yang paling cocok)
(fit <- fit.variogram(vgm.ln.2, model = vgm(model = c("Exp", "Sph", "Gau"))))

#menampilkan model
plot(vgm.ln.2,fit)

#didapat model spherical sbg model terbaik dan didapat sill (co+c) sebesar 0.8 dan range (a) sebesar 0.418
#artinya nilai variogram monoton naik saat h <= 0.8 dan konstan setelah h > 0.418

```

Gambar 2.3 Sintaks R untuk mencari dan memplot model variogram data

Dipilih model dengan transformasi ln pada data serta melebarkan kelas tiap data sebesar 0.05 dari lebar kelas pada awalnya.

Setelah dipilih model yang baik, akan dilakukan interpolasi kriging. Sintaks selanjutnya akan menaksir titik-titik koordinat yang tidak dilakukan pengekurun sebelumnya. Setelah itu akan dibuat *heat-map* nilai *k-fracture* dari koordinat yang kita dapat melalui data dari nilai yang kita taksir (secara interpolasi).

```

# Membuat grid untuk melihat peta kontur sebaran data untuk lokasi
# tidak terobservasi dengan metode ordinary kriging # n: ukuran grid,
n <- 0.02
kolom <- seq(datasumurjtb@bbox[1,1]-n,datasumurjtb@bbox[1,2]+n,by = n)
baris <- seq(datasumurjtb@bbox[2,1]-n,datasumurjtb@bbox[2,2]+n,by = n)
the.grid <- expand.grid(x = kolom, y = baris)
coordinates(the.grid) <- ~x+y
gridded(the.grid) <- T

# Plot grid dan titik observasi
par(mar=c(1,1,1,1))
plot(the.grid, cex=0.5, col="grey")
points(datasumurjtb, pch=1, col='red', cex=1)

# Melakukan penaksiran untuk membentuk peta kontur dengan ordinary kriging
kriging <- krige(k.fracture~1,datasumurjtb, the.grid, model=fit)

# Plot kontur sebagai hasil taksiran oleh kriging
titik <- SpatialPoints(datasumurjtb@coords)
LayoutPoints <- list('sp.points', titik, pch = 19, cex = 0.8, col = 'red')

# Sesuaikan kolom data yang digunakan pada DataSumurJTB[[n]]
LayoutLabels <- list('sp.pointLabel', titik, label = as.character(datasumurjtb[[4]]), cex = 0.8, col = 'white')
spplot(kriging["var1.pred"], main = "Kontur dan Data", sp.layout=list(LayoutPoints,LayoutLabels))

```

Gambar 2.4 Sintaks R untuk melakukan interpolasi data dan membangun *heat-map*

Sintaks bagian terakhir adalah mencari nilai *k-fracture* pada koordinat yang diinterpolasi. Misalkan diambil titik (0.2,-1),(0.4,-0.8),(0.8,-1.0),(1,-1.4), dan (1.2,-0.5) untuk diinterpolasi.

```

#mencari nilai variogram pada titik tertentu (pada titik yang diinterpolasi)
# Menentukan koordinat;
# contoh 5 lokasi (0.2,-1),(0.4,-0.8),(0.8,-1.0),(1,-1.4), dan (1.2,-0.5)
titik2 <- SpatialPoints(cbind(c(0.2,0.4,0.8,1,1.2),c(-1,-0.8,-1,-1.4,-0.5)))
taksiran <- krige(k.fracture~1, datasumurjtb, titik2, model=fit)[[1]]

LayoutPoints.T <- list('sp.points', titik2, pch=19, cex=0.8, col='green')
LayoutLabels.T <- list('sp.pointLabel', titik2, label = as.character(taksiran), cex = 0.8, col = 'white')
spplot(kriging["vari1.pred"], main = "Set Ordinary Kriging Prediction", sp.layout=list(LayoutPoints.T,LayoutLabels.T))

```

Gambar 2.4 Sintaks R untuk menentukan dan menampilkan hasil interpolasi data

III. Keluaran R

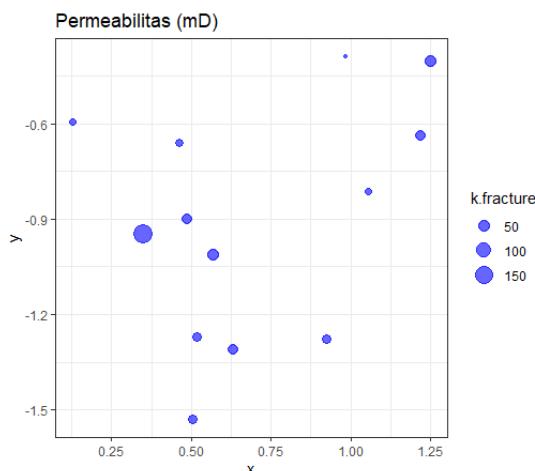
Berikut ini adalah keluaran sintaks R dari bab II.

```

> #input data
> library(readxl)
> DATA_VARIOGRAM <- read_excel("D:/praktikum 9 analisis spasial/DATA VARIOGRAM.xlsx")
> data <- read_xlsx("D:/praktikum 9 analisis spasial/DATA VARIOGRAM.xlsx", sheet="Sumur Jatibarang")
> data$sumurjtb <- data.frame(data)
>
>
> #Mengaktifkan library
> library(sp)
> library(gstat)
> library(dplyr)
> library(ggplot2)
> library(maptools)
>
>
> #Persiapan data
> # 1. mengubah data frame mjd spatial point data frame
> coordinates(data$sumurjtb) = ~x+y
> glimpse(data$sumurjtb)
Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
..@ data     :data.frame': 13 obs. of  5 variables:
... $ no      : num [1:13] 8 18 28 38 48 58 68 78 88 98 ...
... $ nosumur : num [1:13] 52 62 72 86 95 107 119 134 145 154 ...
... $ dz      : num [1:13] 213 291 291 388 138 108 168 256 239 238 ...
... $ k.fracture: num [1:13] 35.4 26.6 51.7 14.3 13.3 ...
... $ pormatrik : num [1:13] 12.16 13.53 15.13 10.62 8.43 ...
..@ coords.nrs : int [1:2] 3 4
..@ coords    : num [1:13, 1:2] 0.631 0.924 0.568 1.056 0.129 ...
.. ..- attr(*, "dimnames")=List of 2
..@ bbox      : num [1:2, 1:2] 0.129 -1.529 1.249 -0.39
.. ..- attr(*, "dimnames")=list of 2
..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
>
> # 2. Membuat diagram pencar dengan ukuran titik menunjukkan
> # besar kecilnya nilai di lokasi tersebut
> ggplot(data = as.data.frame(data$sumurjtb), aes(x,y)) +
+   geom_point(aes(size = k.fracture), col = "blue", alpha = 0.6) +
+   ggtitle("Permeabilitas (mD)") + coord_equal() + theme_bw()
>

```

Gambar 3.1 Keluaran sintaks untuk menyiapkan data dan membuat diagram pencar

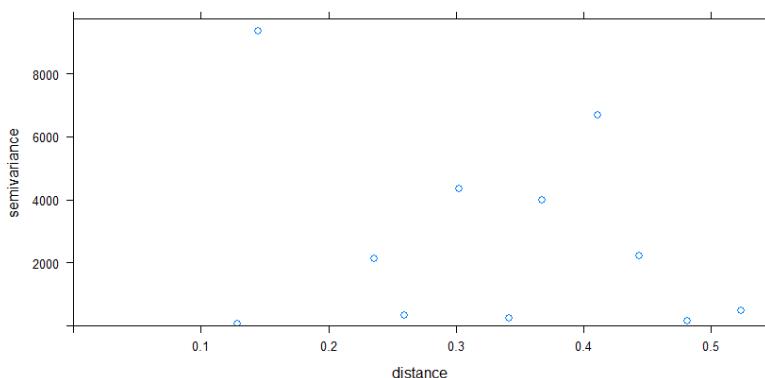


Gambar 3.2 Diagram pencar nilai *k-fracture* pada koordinat data

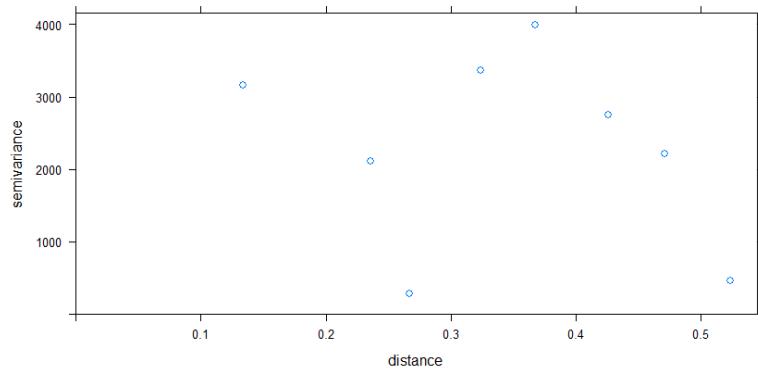
```
> # 2. memperlebar interval semivariogram sehingga pasangan data pada suatu kelas h semakin banyak
> (vgm2 <- variogram(k.fracture ~ 1, datasmurjtb, width = 0.05))
  np      dist    gamma dir.hor dir.ver id
1 3 0.1340292 3157.0179     0     0 var1
2 4 0.2359029 2112.8733     0     0 var1
3 5 0.2667206 277.1449     0     0 var1
4 4 0.3236990 3365.6813     0     0 var1
5 3 0.3673355 3997.5117     0     0 var1
6 5 0.4256882 2753.7785     0     0 var1
7 5 0.4712329 2218.4161     0     0 var1
8 2 0.5239131 462.4068     0     0 var1
> plot(vgm2, cex = 1)
>
> # 3. mentransformasi semivariogram (misal dgn ln)
> (vgm.ln.1 <- variogram(log(k.fracture) ~ 1, datasmurjtb))
  np      dist    gamma dir.hor dir.ver id
1 2 0.1287219 0.03586834     0     0 var1
2 1 0.1446437 1.10045482     0     0 var1
3 4 0.2359029 0.40290521     0     0 var1
4 4 0.2595862 0.75744251     0     0 var1
5 3 0.3022030 0.96274361     0     0 var1
6 2 0.3417225 1.04839830     0     0 var1
7 3 0.3673355 0.83421135     0     0 var1
8 2 0.4110238 1.67188612     0     0 var1
9 5 0.4438359 0.58422400     0     0 var1
10 3 0.4811260 0.26967470    0     0 var1
11 2 0.5239131 0.48180470    0     0 var1
> plot(vgm.ln.1, main = "var.ln", cex = 1)
>
> (vgm.ln.2 <- variogram(log(k.fracture)~1, datasmurjtb, width = 0.05))
  np      dist    gamma dir.hor dir.ver id
1 3 0.1340292 0.3907305     0     0 var1
2 4 0.2359029 0.4029052     0     0 var1
3 5 0.2667206 0.6140982     0     0 var1
4 4 0.3236990 1.2360766     0     0 var1
5 3 0.3673355 0.8342113     0     0 var1
6 5 0.4256882 0.8394007     0     0 var1
7 5 0.4712329 0.5753826     0     0 var1
8 2 0.5239131 0.4818047     0     0 var1
> plot(vgm.ln.2, main = "var.ln wth = 0.05" ,cex=1)
>

> # Melihat contoh-contoh model variogram yang dapat dimodelkan R melalui fungsi vgm
> show.vgms()
> vgm()
  short                                long
1 Nug                               Nug (nugget)
2 Exp                               Exp (exponential)
3 Sph                               sph (spherical)
4 Gau                               Gau (gaussian)
5 Exc      Exclass (Exponential class/stable)
6 Mat      Mat (Matern)
7 Ste Mat (Matern, M. Stein's parameterization)
8 Cir      cir (circular)
9 Lin      Lin (linear)
10 Bes     Bes (bessel)
11 Pen     Pen (pentaspherical)
12 Per     Per (periodic)
13 wav    wav (wave)
14 Hol    Hol (hole)
15 Log     Log (logarithmic)
16 Pow    Pow (power)
17 Spl     spl (sppline)
18 Leg     Leg (Legendre)
19 Err     Err (Measurement error)
20 Int     Int (Intercept)
>
> # Menampilkan model variogram eksperimental terbaik diantara model (Exp, "Sph", "Gau")
> # Fitting variogram (mencari model yang paling cocok)
> (fit <- fit.variogram(vgm.ln.2, model = vgm(model = c("Exp", "sph", "Gau"))))
model   psill   range
1  Sph 0.7998145 0.4178029
> #menampilkan model
> plot(vgm.ln.2,fit)
```

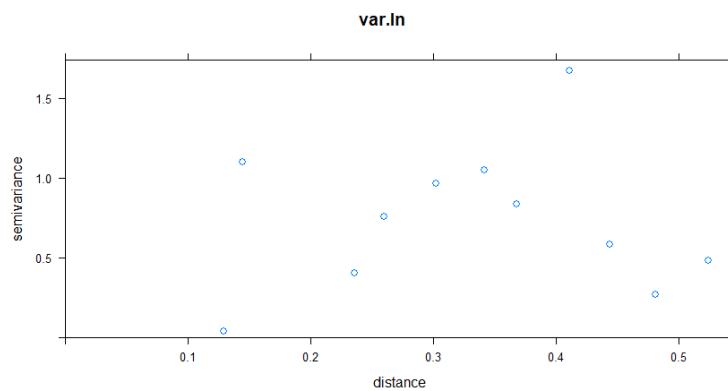
Gambar 3.3 Keluaran sintaks R untuk mencari dan memplot semivariogram



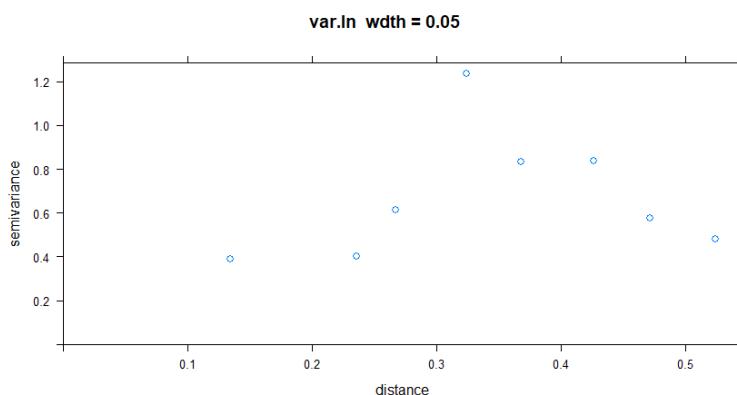
Gambar 3.4 Plot variogram eksperimental



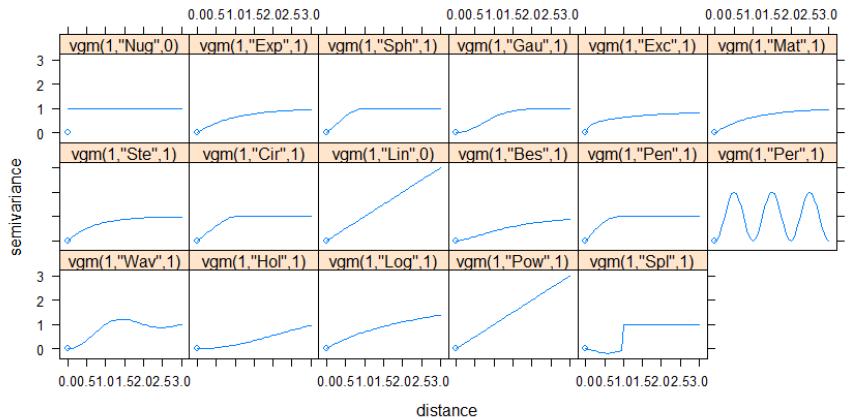
Gambar 3.5 Plot variogram dengan interval kelas diperlebar sebesar 0.05



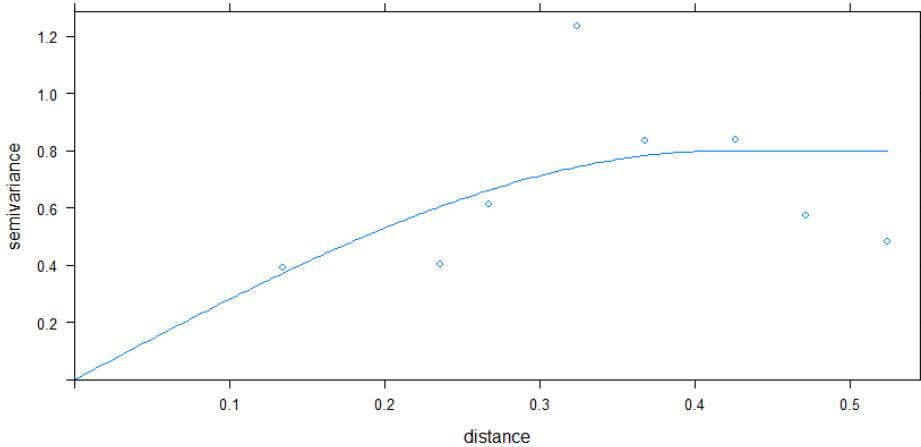
Gambar 3.6 Plot variogram data yang sudah ditransformasi dengan ln



Gambar 3.7 Plot variogram data yang sudah ditransformasi dengan ln serta interval kelas diperlebar sebesar 0.05



Gambar 3.8 Bentuk-bentuk plot variogram yang dapat dikeluarkan oleh program R



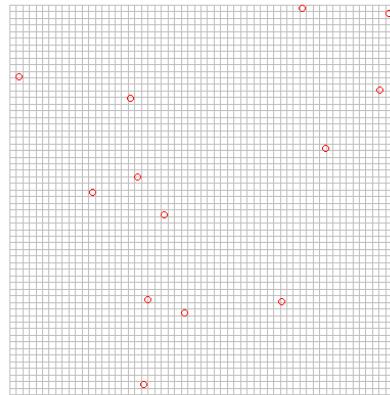
Gambar 3.9 Plotting dan fitting model variogram terbaik (model yang ditransformasi dengan \ln serta interval yang diperlebar sebesar 0.05)

```

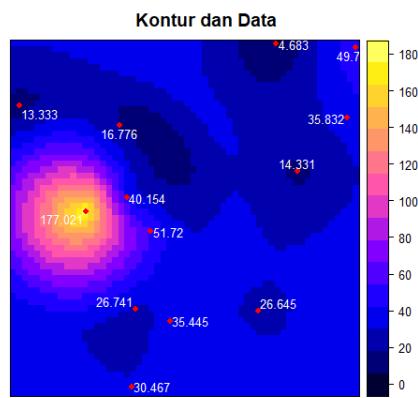
> # Membuat grid untuk melihat peta kontur sebaran data untuk lokasi
> # tidak terobservasi dengan metode ordinary kriging # n: ukuran grid,
> n <- 0.02
> kolin <- seq(datasumurjtb@bbox[1,1]-n,datasumurjtb@bbox[1,2]+n,by = n)
> baris <- seq(datasumurjtb@bbox[2,1]-n,datasumurjtb@bbox[2,2]+n,by = n)
> the.grid <- expand.grid(x = kolin, y = baris)
> coordinates(the.grid) <- ~x+y
> gridded(the.grid) <- T
>
> # Plot grid dan titik observasi
> par(mar=c(1,1,1,1))
> plot(the.grid, cex=0.5, col="grey")
> points(datasumurjtb, pch=1, col="red", cex=1)
>
> # Melakukan penaksiran untuk membentuk peta kontur dengan ordinary kriging
> kriging <- krig(k.fracture~1,datasumurjtb, the.grid, model=fit)
> [using ordinary kriging]
>
> # Plot kontur sebagai hasil taksiran oleh kriging
> titik <- spatialPoints(datasumurjtb@coords)
> LayoutPoints <- list('sp.points', titik, pch = 19, cex = 0.8, col = 'red')
>
> # Sesuaikan kolom data yang digunakan pada DataSumurJTB[[n]]
> LayoutLabels <- list('sp.pointLabel', titik, label = as.character(datasumurjtb[[4]]), cex = 0.8, col = 'white')
> spplot(kriging["var1.pred"], main = "Kontur dan Data", sp.layout=list(LayoutPoints,LayoutLabels))
Warning messages:
1: Function moved to https://github.com/oscarperpinan/lable
2: Function moved to https://github.com/oscarperpinan/lable
3: Function moved to https://github.com/oscarperpinan/lable

```

Gambar 3.10 Keluaran sintaks untuk melakukan interpolasi kriging serta membangun *heat-map*



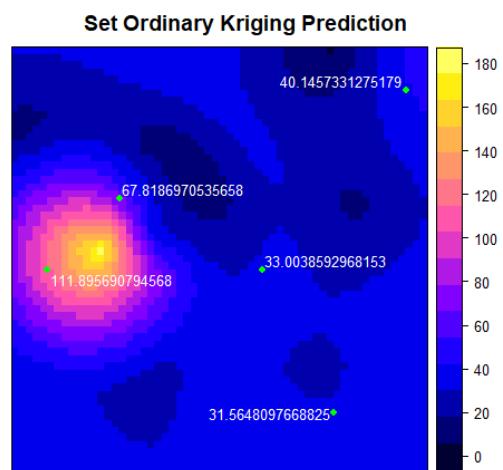
Gambar 3.11 Hasil plot koordinat data yang diketahui serta menampilkan grid



Gambar 3.12 Hasil *heat-map* data

```
> #mencari nilai variogram pada titik tertentu (pada titik yang diinterpolasi)
Warning messages:
1: Function moved to https://github.com/oscarperpinan/label
2: Function moved to https://github.com/oscarperpinan/label
> # Menentukan koordinat;
> # contoh 5 lokasi (0.2,-1),(0.4,-0.8),(0.8,-1.0),(1,-1.4),dan (1.2,-0.5)
> titik2 <- SpatialPoints(cbind(c(0.2,0.4,0.8,1,1.2),c(-1,-0.8,-1,-1.4,-0.5)))
> taksiran <- krig(k.fracture-1, datasumurjtb, titik2, model=fit)[[1]]
[using ordinary kriging]
>
> LayoutPoints.T <- list('sp.points', titik2, pch=19, cex=0.8, col='green')
> LayoutLabels.T <- list('sp.pointLabel', titik2, label = as.character(taksiran), cex = 0.8, col = 'white')
> spplot(kriging["vari1.pred"], main = "set ordinary Kriging Prediction", sp.layout=list(LayoutPoints.T,LayoutLabels.T))
Warning messages:
1: Function moved to https://github.com/oscarperpinan/label
2: Function moved to https://github.com/oscarperpinan/label
3: Function moved to https://github.com/oscarperpinan/label
```

Gambar 3.13 Mencari nilai *k-fracture* pada beberapa titik yang diinterpolasi serta menampilkannya pada *heat-map*



Gambar 3.14 Plot nilai dan titik yang diinterpolasi