

**LAPORAN TUGAS BESAR**  
**METODE SEISMIK TG3105**

---

**Diajukan Sebagai Salah Satu Syarat Kelulusan**  
**Mata Kuliah TG3105 Metode Seismik**

***CROSSHOLE TOMOGRAPHY***

**Disusun Oleh:**

**Kelompok 4**

<b>Muhammad Rendi Jaya</b>	<b>119120050</b>
<b>Risa Solehatin</b>	<b>119120057</b>
<b>Adzan Fareliah Syahputra</b>	<b>119120079</b>
<b>Rifki Nova Suryo</b>	<b>119120082</b>
<b>Annisa Efriyanti</b>	<b>119120127</b>

**Asisten Pembimbing:**

**Muhammad Ichsan, S.T.**

**Dosen Pengampu:**

**Ruhul Firdaus, S.T.,M.T.**



**PROGRAM STUDI TEKNIK GEOFISIKA**  
**JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI**  
**INSTITUT TEKNOLOGI SUMATERA**

**2021**

## DAFTAR ISI

Daftar Isi .....	i
Daftar Gambar .....	ii
Daftar Tabel .....	iii
A. Tujuan Praktikum .....	1
B. Dasar Teori .....	1
2.1 Metode Seismik .....	1
2.2 Tomografi .....	1
2.3 <i>Crosshole Tomography</i> .....	2
C. Langkah Pengerjaan .....	3
3.1 Daftar Langkah .....	3
3.2 Diagram Alir .....	6
D. Hasil dan Pembahasan .....	7
4.1 Hasil .....	7
4.2 Pembahasan .....	9
E. Kesimpulan .....	9
Daftar Pustaka .....	
Lampiran .....	

## DAFTAR GAMBAR

<b>Gambar 1.</b> Diagram alir .....	6
<b>Gambar 2.</b> <i>Final model velocity</i> .....	8
<b>Gambar 3.</b> Grafik RMS <i>error</i> vs iterasi .....	8

## DAFTAR TABEL

<b>Tabel 1.</b> Nilai $m$ pada iterasi awal dan iterasi akhir .....	6
<b>Tabel 2.</b> Iterasi dan <i>error</i> .....	7

## **A. TUJUAN PRAKTIKUM**

Tujuan dari tugas besar Metode Seismik TG3105 kali ini yang membahas tentang *crosshole tomography* adalah sebagai berikut.

1. Mahasiswa mampu menyelesaikan permasalahan dalam menampilkan *crosshole tomography* seismik.
2. Mahasiswa mampu memahami model kecepatan yang didapat dari pembuatan *crosshole tomography* seismik.
3. Mahasiswa mampu dalam memahami algoritma inversi dalam menampilkan *Crosshole Tomography Seismic*.

## **B. DASAR TEORI**

### **2.1 Metode Seismik**

Metode seismik merupakan metode geofisika yang memanfaatkan perambatan gelombang seismik ke dalam bumi (Setiawan, 2008). Metode seismik merupakan salah satu bagian dari seismologi eksplorasi yang dikelompokkan dalam metode geofisika aktif, dimana pengukuran dilakukan dengan menggunakan getaran seismik (palu/ledakan). Setelah usikan diberikan, terjadi gerakan gelombang di dalam medium (tanah/batuan) yang memenuhi hukum-hukum elastisitas ke segala arah dan mengalami pemantulan ataupun pembiasan akibat munculnya perbedaan kecepatan. Gelombang seismik merambat dari sumber ke penerima melalui lapisan bumi dan mentransfer energi sehingga dapat menggerakkan partikel batuan.

### **2.2 Tomografi**

Tomografi merupakan suatu gambaran dari suatu penampang-lintang dari suatu objek. Tomografi dalam geofisika menggabungkan dua buah aspek penting analisis geologi, yakni estimasi sifat-sifat geologi dan pencitraan ke dalam satu konsep.

Tomografi memiliki peran yang sangat penting dalam proses pemantauan *anomaly*, karena tomografi dapat menggambarkan kondisi geologi di bawah permukaan bumi. Tomografi digunakan untuk menganalisis sifat-sifat gelombang dan juga kecepatan gelombang medium yang dilalui. Sehingga,

dapat mencitrakan prediksi dari bawah permukaan dan sifat-sifat geologinya (Prabowo, Riza Eka. dkk, 2006).

Tomografi biasa juga disebut dengan seismik inversi, yang merupakan salah satu metode geofisika yang telah terbukti dapat menggambarkan keadaan bawah permukaan bumi secara akurat. *seismic tomography* merupakan suatu metoda untuk merekonstruksi model kecepatan bumi dari kecepatan seismik. Prinsip utama dalam *seismic tomography* adalah menyajikan gambaran bawah permukaan dalam Kawasan kecepatan (Zainurandriana, Rizfi. dkk, 2019).

Salah satu keunggulan dari tomografi dibandingkan dengan metoda *seismic* permukaan lainnya adalah jejak sinyal gelombang yang digunakan menjalar pada satu arah dan memiliki waktu tempuh yang lebih pendek, karena sinyal gelombang tersebut hanya menjalar dari satu lubang bor ke lubang bor lainnya., dibandingkan jika sinyal gelombang yang menjalar dari permukaan menuju ke bawah permukaan kemudian kembali lagi ke permukaan. Hal ini dapat memberikan gambaran bahwa permukaan yang beresolusi tinggi dan memperkecil terjadinya sinyal yang hilang dalam proses transmisinya. Keunggulan lainnya daripada metode tomografi adalah tipe transmisi data akuisisi yang bersifat sederhana, tertutup terhadap target bawah permukaan, algoritma inversinya memiliki kesamaan inheren, yang hasilnya memiliki presisi dan akurasi yang tinggi.

### **2.3 *Cross-hole Tomography***

Karakterisasi rinci heterogenitas geologi bawah permukaan yang mengontrol aliran fluida dan transportasi masih merupakan tantangan yang belum terselesaikan dalam komunitas geosains. Sementara metode geofisika permukaan telah membantu dalam memperoleh distribusi parameter dalam skala besar, penurunan resolusi dengan kedalaman membatasi aplikasi mereka untuk menggambarkan heterogenitas pada resolusi tinggi. Keterbatasan ini mendorong penggunaan teknik geofisika melakukan *cross-hole seismic tomography*. Dimana, *cross-hole seismic tomography* yang melibatkan pengukuran waktu tempuh jalur sinar seismik antara dua atau lebih lubang bor untuk mendapatkan gambar kecepatan seismik antara sumur

menawarkan pendekatan untuk mengkarakterisasi heterogenitas geologi pada resolusi tinggi. Aplikasinya berkisar dari studi akuifer dekat permukaan, panas bumi, hingga reservoir minyak dan gas dalam (Sainkov et al 2005; Tselentis et al 2011). Becht et al., (2007), mengevaluasi penggunaan tomografi seismik *cross-hole* untuk karakterisasi akuifer resolusi tinggi (Ehosioka, Solomon, 2014).

## C. LANGKAH Pengerjaan

### 3.1 Daftar Langkah

1. *Input* modul *numpy* as *np*, *matplotlib.pyplot* as *plt*, dan *scipy.sparse*

```
import matplotlib.pyplot as plt
import numpy as np
from numpy.linalg import inv
import scipy.sparse
#from sklearn.metrics import mean_squared_error
```

2. Inputkan *v* berkisar dari 200-3000 m/s

```
v = np.array(([3000,3000,3000,3000],
              [3000,200,200,3000],
              [3000,200,200,3000],
              [3000,3000,3000,3000]))
```

3. Untuk mencari *tcal* tambahkan rumus *s*

```
s = 1/v
print('v =',v)
```

4. *Input* fungsi *tcal*

```
#tcal
t11 = 100.7*s[0][0]+100.7*s[0][1]+100.7*s[0][2]+100.7*s[0][3]
t12 = 106.8*s[0][0]+35.6*s[0][1]+71.2*s[1][1]+106.8*s[1][2]+106.8*s[1][3]
t13 = 94.33*s[0][0]+23.5*s[0][1]+117.9*s[1][1]+47.1*s[1][2]+70.7*s[2][2]+117.9*s[2][3]
t21 = 100.7*s[1][0]+100.7*s[1][1]+100.7*s[1][2]+100.7*s[1][3]
t22 = 100.7*s[1][0]+100.7*s[1][1]+100.7*s[1][2]+100.7*s[1][3]
t23 = 106.8*s[1][0]+35.6*s[1][1]+71.2*s[2][1]+106.8*s[2][2]+106.8*s[2][3]
t31 = 106.8*s[2][0]+35.6*s[2][1]+71.2*s[1][1]+106.8*s[1][2]+106.8*s[1][3]
t32 = 100.7*s[2][0]+100.7*s[2][1]+100.7*s[2][2]+100.7*s[2][3]
t33 = 100.7*s[2][0]+100.7*s[2][1]+100.7*s[2][2]+100.7*s[2][3]
t41 = 94.33*s[3][0]+23.5*s[2][0]+117.9*s[2][1]+47.1*s[2][2]+70.7*s[1][2]+117.9*s[1][3]
t42 = 106.8*s[3][0]+35.6*s[3][1]+71.2*s[2][1]+106.8*s[2][2]+106.8*s[2][3]
t43 = 100.7*s[3][0]+100.7*s[3][1]+100.7*s[3][2]+100.7*s[3][3]

tcal = np.array((t11,t12,t13,t21,t22,t23,t31,t32,t33,t41,t42,t43))

print('s =', s)
print('t=',tcal)
```

5. Tentukan nilai matriks G

```
# matriks G
G = np.zeros((len(tcal), 16))
G[0][0] = 100.7
G[0][1] = 100.7
G[0][2] = 100.7
G[0][3] = 100.7
G[1][0] = 106.8
G[1][1] = 35.6
G[1][5] = 71.2
G[1][6] = 106.8
G[1][7] = 106.8
G[2][0] = 94.33
G[2][4] = 23.5
G[2][5] = 117.9
G[2][6] = 47.3
G[2][10] = 70.6
G[2][11] = 117.9
G[3][4] = 100.7
G[3][5] = 100.7
G[3][6] = 100.7
G[3][7] = 100.7
G[4][4] = 100.7
G[4][5] = 100.701 #----
G[4][6] = 100.699 #----
G[4][7] = 100.7
G[5][4] = 106.8
G[5][5] = 35.6
G[5][9] = 71.2
G[5][10] = 106.8
```

6. Lalu lakukan *transpose* pada matriks G, dikalikan dan hasil perkalian tersebut dilakukan *inverse*

```
Gtrans = G.transpose()
perkalian = np.dot(G, Gtrans)
invers = np.linalg.inv(perkalian)
print('Nilai Inversi = ', invers)
print()

gtransdikaliinvers = np.dot(Gtrans, invers)

hasilresize = tcal.reshape(12,1)
print('hasil resize = ', hasilresize.shape)
print()
```

7. *Input* formula untuk mencari nilai m pada inversi pertama

```
m = np.dot(gtransdikaliinvers, hasilresize)
print('m = ',m)
print()
mbenar = 1/m
print('mbenar=',mbenar)
resizem = mbenar.reshape(4,4)
print('resizem = ', resized)
print()

def sparse(i, j, v):
    return scipy.sparse.coo_matrix((v, (i, j)))
```



8. Atur jumlah iterasi, error dan *input* nilai tobs yang telah ditentukan

```
error = 1e9
iterasi = 1
a=1

stop_crit = 0.01
jumlah_iterasi=100

error = 1e9
iterasi = 1
tobs = np.array([[2.014],
                  [0.90186667],
                  [0.82495],
                  [0.2685333],
                  [0.2685333],
                  [0.2848],
                  [0.2848],
                  [0.2685333],
                  [0.2685333],
                  [0.72311667],
                  [0.90186667],
                  [2.014]])

mo=s.reshape(16,1)
mi=mo
lmbd=100000
i=0
errorplot=[]
```

9. *Input* formula untuk mencari nilai tc al

```
tc al = G @ mi #matriks G di inversikan dengan nilai m
dt = tobs - tc al #update waktu
dm = np.linalg.inv(G.T @ G + np.dot(lmbd,np.identity(16))) @ G.T @ dt #update nilai m
mhasil = (1/mi).reshape(4,4)
mi = mi + dm
print('tc al', tc al)
print('mbaru',mi)
print('dm',dm)
print('dt',dt)
```

10. Tampilkan hasil model inisial dan iterasi yang telah dibuat sebelumnya

```
x,y = np.arange(0, 5, 1),np.arange(0, 5, 1)
plt.imshow(mhasil, extent=[0, 4, 0, 4])
if i==0:
    plt.title("Model Inisial")
else:
    plt.title("iterasi ke-"+str(i))
plt.pcolormesh(x, y, mhasil)
plt.gca().invert_yaxis()
plt.clim(200,3000)
plt.colorbar()

plt.show()
```

11. *Input* formula untuk mencari error pada pemodelan yang dibuat

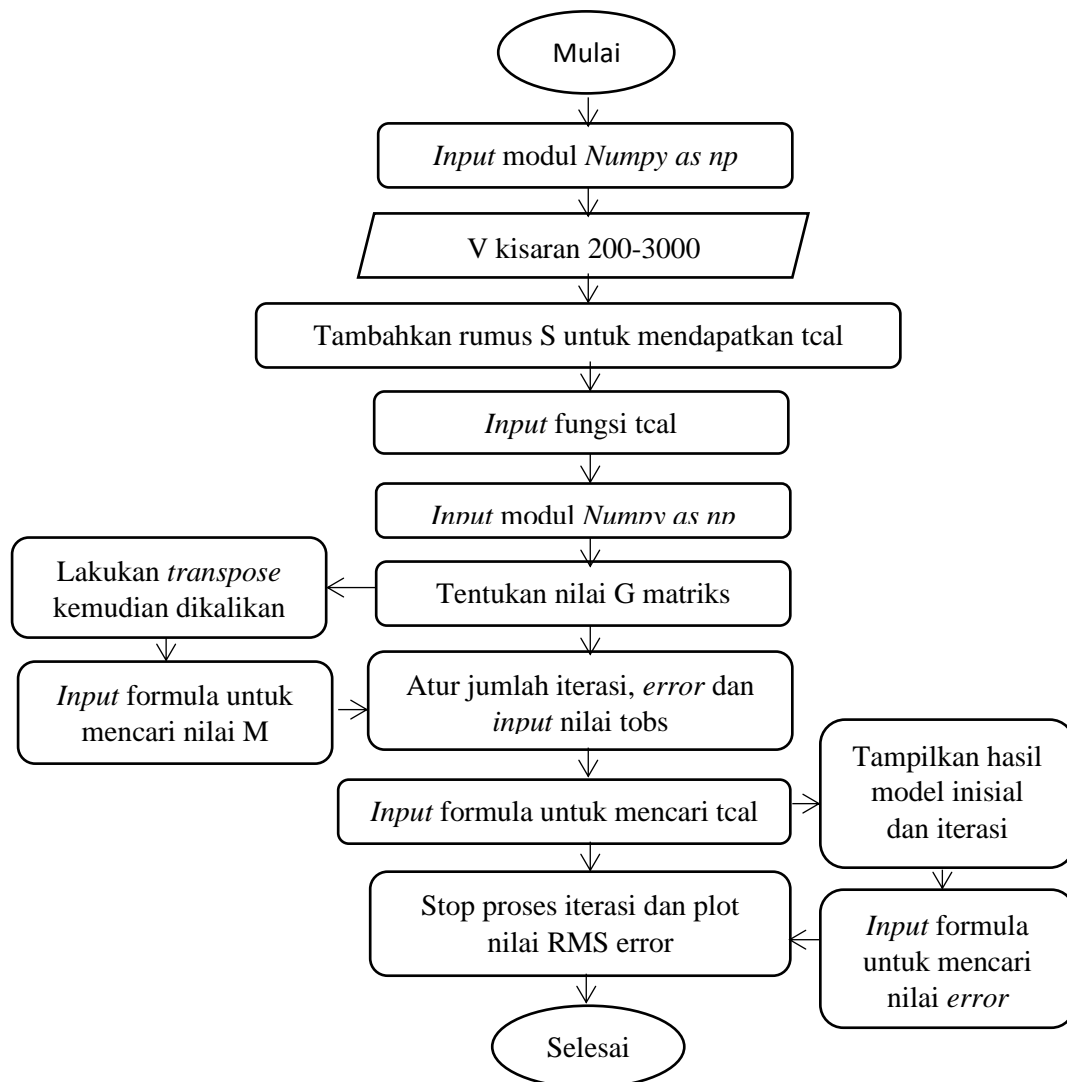
```
error = np.sqrt(np.mean(np.power(dt,2)))
errorplot.append(error)
print('error',error*100,"%")
```

12. Hentikan proses iterasi dan plot nilai RMS error vs Iterasi dalam bentuk grafik

```
iterasi += 1
i+=1
if iterasi==jumlah_iterasi:
    break

plt.plot(errorplot)
plt.title("RMS error vs Iterasi")
plt.show()
```

### 3.2 Diagram Alir



**Gambar 1.** Diagram alir

## D. HASIL DAN PEMBAHASAN

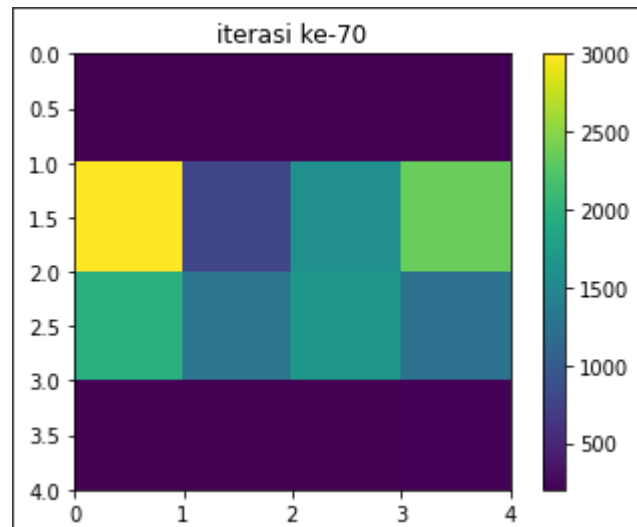
### 4.1 Hasil

**Tabel 1.** Nilai  $m$  pada iterasi awal dan iterasi akhir

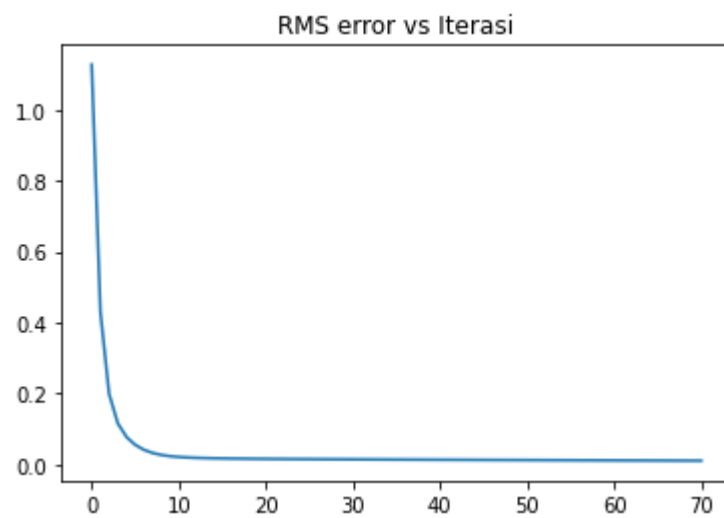
Nilai $m$ pada iterasi pertama	Nilai $m$ pada iterasi akhir
0.00334382	0.00509169
0.00431619	0.00473964
0.00436589	0.00507678
0.00436589	0.00507678
0.00228736	0.00032925
0.0019843	0.00130157
0.00185147	0.00063035
0.00179482	0.00042246
0.00228062	0.00050443
0.00182197	0.00078296
0.00183694	0.00059283
0.00189358	0.00080071
0.00385536	0.00507307
0.00431121	0.00484218
0.00437939	0.00536641
0.00371272	0.00469974

**Tabel 2.** Iterasi dan Error

Nilai <i>error</i> pada iterasi pertama (%)	Nilai <i>error</i> pada iterasi terakhir (%)
112.73303566433657	0.9959285900831996



**Gambar 2.** *Final model velocity*



**Gambar 3.** Grafik RMS error Vs iterasi

## 4.2 Pembahasan

Berdasarkan percobaan *crosshole* yang telah dilakukan didapatkan nilai *slowness* berkisar 0.000333-0.004 nilai jni dipengaruhi oleh nilai model kecepatan yang dimasukkan. Model kecepatan yang dipakai pada percobaan ini adalah sebagai berikut:

```
[300 250 250 250
 250 250 250 250
 250 250 250 250
 250 250 250 300]
```

Pada nilai model kecepatan diatas dimasukkan nilai kecepatan secara acak sehingga didapatkan nilai model kecepatan awal seperti data diatas, pada model kecepatan ini didapatkan nilai *error* yang cukup kecil sehingga model kecepatan ini yang digunakan pada percobaan ini. Iterasi pada model ini sebanyak 71 iterasi atau iterasi ke-70 dikarenakan pada hasil python nilai 1 dibaca 0, pada iterasi ke-71 nilai dari tobs yang sudah diinput sama dengan nilai tcsl sehingga iterasi berhenti pada iterasi ke-71.

*Crosshole* dilakukan dengan cara *invers modelling* dikarenakan dari data tobs yang sudah diketahui diubah menjadi *final model velocity* jika tidak menggunakan *invers modelling* tidak bisa didapatkan hasil *final model velocity* yang dicari. Nilai *final model velocity* yang didapat setelah melakukan *invers modelling* adalah:

```
[ 196.44049344 210.80470864 197.03456846 197.03456846
 3030.65430301 770.61985551 1581.99660572 2358.11677018
 1978.77098681 1284.27091383 1682.1420379 1246.08712725
 197.14071364 206.34269587 186.40979603 212.8629562 ]
```

## E. KESIMPULAN

Adapun kesimpulan dari tugas besar Metode Seismik TG3105 kali ini yang membahas tentang *crosshole tomography* adalah sebagai berikut :

1. Untuk menampilkan hasil *Crosshole Tomography Seismic* perlu dilakukannya *invers modelling* dari data yang sudah ada sehingga didapatkan hasil *final model tomography*.
2. Model kecepatan yang didapat adalah:

[ 196.44049344 210.80470864 197.03456846 197.03456846  
3030.65430301 770.61985551 1581.99660572 2358.11677018  
1978.77098681 1284.27091383 1682.1420379 1246.08712725  
197.14071364 206.34269587 186.40979603 212.8629562 ]

3. Algoritma inversi dalam menampilkan *crosshole Tomography seismic* mencari nilai  $t_{cal}$  terlebih dahulu lalu melakukan inversi terhadap matriks  $G$  dan mencari nilai  $m$  atau inisial model yang baru kemudian nilai  $G$  dan  $m$  di inversikan untuk memperoleh data  $t_{obs}$  sama dengan  $t_{cal}$ .

## DAFTAR PUSTAKA

- Ehosioko, Solomon. (2014). Researchgate. *Application of Cross-hole Seismic Tomography in Characterization of Heterogeneous Aquifers*, 1.
- Fansury, Dhesy. (n.d.). *Metode Seismik*. Retrieved from Scribd: <https://id.scribd.com/doc/304359588/Metode-Seismik>
- Prabowo, Riza Eka. dkk. (2006). Pemodelan Tomografi Cross-hole Metode Geolistrik Resistivitas (Bentuk Anomali Silindris). *Berkala Fisika Vol. 9 No. 1*, 23.
- Setiawan, B. (2008). Pemetaan Tingkat Kekerasan Batuan Menggunakan Metode Seismik Refraksi. *Skripsi.*, Depok : Universitas Indonesia.
- Zainurandiana, Rizfi. dkk. (2019). Penentuan Struktur Kecepatan Dangkal dengan Metode Seismik Tomografi Refraksi Pada Data 2D. *Repo ITERA*, 2.

## **LAMPIRAN**

### **Script:**

[https://colab.research.google.com/drive/1\\_XVTNeF\\_fTwtRQvtz2wdYwvEzNVI\\_Nqag?usp=sharing](https://colab.research.google.com/drive/1_XVTNeF_fTwtRQvtz2wdYwvEzNVI_Nqag?usp=sharing)