# Peningkatan Kinerja Pemodelan Resistivitas DC 3D dengan GPU Berkemampuan CUDA

Hairil Anwar[1,a)], Achmad Imam Kistijantoro[1,b)] dan Wahyu Srigutomo[2,c)]

[1]Laboratorium Sistem Terdistribusi,
Kelompok Keilmuan Informatika,
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung,
Jl. Ganesha no. 10 Bandung, Indonesia, 40132

[2]Laboratorium Fisika Bumi,
Kelompok Keilmuan Fisika Bumi dan Sistem Kompleks,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Bandung,
Jl. Ganesha no. 10 Bandung, Indonesia, 40132

[a)]23514034@std.stei.itb.ac.id (corresponding author)
[b)]imam@stei.itb.ac.id
[c)]wahyu@fi.itb.ac.id

**Abstrak**

*Pemodelan resistivitas DC 3D dilakukan dengan menggunakan metode elemen hingga Galerkin untuk formulasi potensial sekunder. Secara garis besar, komputasi yang terlibat pada metode elemen hingga meliputi: diskritisasi domain pemodelan, penyusunan matriks global, penerapan syarat batas, dan penyelesaian sistem persamaan linier. Domain pemodelan didiskritisasi dengan menggunakan elemen tetrahedron. Matriks global diperoleh menggunakan formulasi metode elemen hingga Galerkin untuk elemen tetrahedron dan disimpan dengan sparse format CSR. Untuk menyelesaikan sistem persamaan linier digunakan metode conjugate gradient (CG). Berdasarkan komputasi yang terlibat, penyelesaian sistem persamaan linier mendominasi sebagian besar waktu komputasi pada pemodelan resistivitas DC 3D. Peningkatan kinerja pemodelan dilakukan dengan memindahkan komputasi penyelesaian sistem persamaan linier pada GPU berkemampuan CUDA dengan memanfaatkan library CUBLAS dan CUSPARSE. Pengukuran kinerja dilakukan pada sistem dengan CPU Intel Core i5-4460, GPU GTX 750Ti dan sistem operasi linux 64-bit. Peningkatan kinerja terhadap serial CPU yang diperoleh dengan menggunakan GPU adalah sebesar 3.86 dan 3.84 untuk single precision dan double precision. Dengan melakukan padding pada format CSR (PCSR) dapat diperoleh peningkatan kinerja 4.81 dan 4.26 untuk single precision dan double precision.*

*Kata-kata kunci: CUDA, GPU, metode elemen hingga, Galerkin, pemodelan resistivitas DC*

## PENDAHULUAN

Pada pemodelan resistivitas DC 3D, potensial akibat adanya sumber arus *direct current* (DC) dimodelkan pada medium dengan resistivitas bervariasi secara tiga dimensi (3D). Pemodelan ini dilakukan dengan menyelesaikan persamaan diferensial orde dua pada ruang 3D beserta syarat batas yang harus dipenuhi pada batas domain. Terdapat dua formulasi persamaan diferensial untuk pemodelan resistivitas DC yaitu formulasi potensial total dan potensial sekunder [1]. Formulasi dengan potensial sekunder dapat menghilangkan efek singular disekitar lokasi sumber arus dan memberikan solusi yang lebih baik dibandingkan formulasi dengan potensial total. Untuk menyelesaikan permasalahan pada pemodelan resistivitas DC 3D secara numerik dapat digunakan metode elemen hingga [2,3]. Secara garis besar, komputasi yang terlibat pada metode elemen

ISBN: 978-602-61045-0-2

hingga meliputi: diskritisasi domain pemodelan, penyusunan matriks global, penerapan syarat batas, dan penyelesaian sistem persamaan linier. berdasarkan waktu komputasinya, penyelesaian sistem persamaan linier merupakan tahap yang mendominasi sebagian besar waktu komputasi pada metode elemen hingga.

Pada makalah ini, kami melakukan peningkatan kinerja pemodelan resistivitas DC 3D untuk formulasi potensial sekunder dengan memanfaatkan *graphics processing unit* (GPU) berkemampuan CUDA. GPU adalah perangkat keras yang dapat dimanfaatkan untuk melakukan komputasi yang terpisah dari CPU. Kemampuan *floating-point* GPU secara teoritik relatif lebih tinggi dibandingkan dengan CPU [4]. GPU berkemampuan CUDA dapat diprogram dengan bahasa CUDA C/C++. Peningkatan kinerja dilakukan dengan memindahkan komputasi penyelesaian sistem persamaan linier pada GPU. Kami menggunakan metode *conjugate gradient* (CG) untuk penyelesaian sistem persamaan linier dari metode elemen hingga.

## PEMODELAN RESISTIVITAS DC 3D

Formulasi potensial total untuk pemodelan resistivitas DC 3D diberikan oleh persamaan,

$$\vec{\nabla}\cdot\left(\sigma\vec{\nabla}V\right)=-I\delta(x-x_0)\delta(y-y_0)\delta(z-z_0)\,, \tag{1}$$

dengan $V$ adalah potensial total [V], $\sigma$ adalah konduktivitas medium [S/m], $I$ adalah besar arus sumber [A], sedangkan $x_0$, $y_0$, dan $z_0$ adalah koordinat sumber arus [m]. Untuk kasus dengan satu sumber arus, syarat batas yang digunakan adalah,

$$\frac{\partial V}{\partial n}=0\,, \tag{2}$$

dan

$$\frac{\partial V}{\partial n}+\frac{\cos\theta}{r}V=0\,. \tag{3}$$

dengan $r$ adalah jarak sumber arus ke bidang batas domain [m], $\theta$ adalah sudut yang terbentuk antara $r$ dengan normal bidang batas domain [rad]. Persamaan (2) di atas adalah syarat batas untuk permukaan sedangkan persamaan (3) adalah syarat batas untuk bidang-bidang batas lainnya selain permukaan. Hubungan antara resistivitas dan konduktivitas dituliskan dengan,

$$\sigma=\frac{1}{\rho}\,. \tag{4}$$

Pada formulasi potensial sekunder, potensial total ($V$) dibagi menjadi potensial primer ($u_0$) dan potensial sekunder ($u$) dan dihitung dengan,

$$V=u_0+u\,. \tag{5}$$

Potensial primer adalah potensial pada medium dengan konduktivitas homogen $\sigma_0$ yang diakibatkan oleh sumber arus $I$. Pada kasus satu sumber arus, potensial primer dapat dihitung dengan,

$$u_0=\frac{I}{2\pi\sigma_0\sqrt{x^2+y^2+z^2}}\,. \tag{6}$$

Formulasi potensial sekunder untuk pemodelan resistivitas DC 3D diberikan oleh persamaan,

$$\vec{\nabla}\cdot\left(\sigma\vec{\nabla}u\right)=-\vec{\nabla}\cdot\left(\Delta\sigma\vec{\nabla}u_0\right), \tag{7}$$

dengan $u$ adalah potensial sekunder [V], $u_0$ adalah potensial primer [V], $\Delta\sigma$ adalah variasi konduktivitas [S/m] yang dihitung dengan,

$$\Delta\sigma=\sigma-\sigma_0\,. \tag{8}$$

Untuk kasus dengan satu sumber arus, syarat batas yang digunakan untuk formulasi potensial sekunder adalah,

$$\frac{\partial u}{\partial n}=0\,, \tag{9}$$

dan

$$\frac{\partial u}{\partial n}+\frac{\cos\theta}{r}u=0\,. \tag{10}$$

Persamaan (9) di atas adalah syarat batas untuk permukaan sedangkan persamaan (10) adalah syarat batas untuk bidang-bidang batas lainnya selain permukaan. Pada formulasi potensial sekunder, solusi yang dicari adalah solusi potensial sekunder kemudian digunakan persamaan (5) dan (6) untuk menghitung solusi potensial total. Untuk kasus dengan dua sumber arus atau lebih, solusi potensial totalnya dapat diperoleh dengan menghitung solusi masing-masing sumber arus pada medium resistivitas yang sama kemudian menjumlahkan solusi potensial totalnya.

Pemodelan resistivitas DC 3D dilakukan untuk formulasi potensial sekunder. Metode elemen hingga digunakan untuk mencari solusi numerik dari persamaan (7) dengan syarat batas pada persamaan (9) dan (10). Domain pemodelan didiskritisasi dengan elemen tetrahedron dan digunakan metode residual terbobot dengan pendekatan Galerkin untuk formulasi matriksnya. Penyelesaian dengan metode elemen hingga akan memberikan sistem persamaan linier,

$$\mathbf{Ax} = \mathbf{b}, \tag{11}$$

dengan **x** berupa vektor solusi yang akan dicari, **A** adalah matriks dan **b** adalah vektor yang diperoleh dengan metode elemen hingga. Matriks disimpan dalam format *compressed sparse row* (CSR) yang efisien dari segi kebutuhan memori dan juga waktu komputasinya. Untuk menyelesaikan sistem persamaan linier yang dihasilkan oleh metode elemen hingga digunakan digunakan metode *conjugate gradient* (CG). Berikut algoritma CG untuk menyelesaikan sistem persamaan linier [5].

$\mathbf{r} := \mathbf{b} - \mathbf{Ax}$
$\mathbf{p} := \mathbf{r}$
$rs := \mathbf{r}^{\mathbf{T}}\mathbf{r}$

**while** (*rs* > *tol*)
$\quad \mathbf{q} := \mathbf{Ap}$
$\quad \alpha := \dfrac{rs}{\mathbf{p}^{\mathbf{T}}\mathbf{q}}$
$\quad \mathbf{x} := \mathbf{x} + \alpha\mathbf{p}$
$\quad \mathbf{r} := \mathbf{r} - \alpha\mathbf{q}$
$\quad rso := rs$
$\quad rs := \mathbf{r}^{\mathbf{T}}\mathbf{r}$
$\quad \beta := \dfrac{rs}{rso}$
$\quad \mathbf{p} := \mathbf{r} + \beta\mathbf{p}$
**end while**

Gambar 1. Algoritma *conjugate gradient* (CG)

Pada algoritma di atas, **x** terlebih dahulu diisi dengan solusi tebakan awal misalkan saja dengan 0. Solusi akhir akan tersimpan pada variable **x** setelah pencarian solusi selesai. Variabel *tol* adalah toleransi yang digunakan untuk menghentikan iterasi pencarian solusi. Pemilihan toleransi mempengaruhi lama komputasi dan akurasi solusi.

## HASIL PENGUKURAN KINERJA

Peningkatan kinerja dilakukan dengan memindahkan komputasi penyelesaian sistem persamaan linier dengan CG pada GPU. Kami menggunakan library CUBLAS dan CUSPARSE untuk melakukan komputasi CG pada GPU. CUBLAS menyediakan fungsi-fungsi untuk melakukan operasi antar vektor sedangkan CUSPARSE menyediakan fungsi-fungsi untuk melakukan operasi-operasi sparse matriks seperti perkalian matriks CSR dengan vektor. Implementasi komputasi pada GPU dilakukan dengan menggunakan pemrograman CUDA C++.

Untuk mengukur kinerja pemodelan resistivitas DC 3D pada GPU, digunakan model resistivitas berlapis yang telah didiskritisasi dengan elemen balok berukuran 150x150 x90. Setiap elemen balok kemudian didiskritisasi lagi dengan 5 elemen tetrahedron tanpa adanya penambahan simpul. Model berlapis digunakan karena solusi analitiknya tersedia dan dapat digunakan untuk memeriksa kebenaran solusi numerik yang diperoleh. Pengukuran waktu komputasi CG dilakukan untuk komputasi serial pada CPU dan GPU untuk *single precision* dan *double precision*. Pengukuran waktu komputasi juga dilakukan untuk struktur matriks CSR yang diberikan *padding* (PCSR). *Padding* pada matriks CSR dilakukan dengan menggenapkan panjang tiap baris matriks menjadi kelipatan 4 dan kelebihannya diisi dengan nilai nol. Sistem yang digunakan untuk pegukuran kinerja dilengkapi dengan sistem operasi Linux 64-bit (Ubuntu 15.04), CPU Intel Core i5-4460, GPU Nvidia GeForce GTX 750Ti, dan CUDA Toolkit 7.5. Berikut hasil pengukuran kinerja pada CPU dan GPU untuk *single precision* dan *double precision*.

Tabel 1. Hasil pengukuran kinerja pada CPU dan GPU untuk *single precision*

| **Implementasi (*single precision*)** | **Waktu komputasi (detik)** | **Peningkatan kinerja relatif** |
|---|---|---|
| Serial CPU | 43.524 | 1.00 |
| GPU CSR | 11.285 | 3.86 |
| GPU PCSR | 9.042 | 4.81 |

Tabel 2. Hasil pengukuran kinerja pada CPU dan GPU untuk *double precision*

| **Implementasi (*double precision*)** | **Waktu komputasi (detik)** | **Peningkatan kinerja relatif** |
|---|---|---|
| Serial CPU | 63.687 | 1.00 |
| GPU CSR | 16.565 | 3.84 |
| GPU PCSR | 14.944 | 4.26 |

Hasil pengukuran di atas adalah pengukuran waktu yang dibutuhkan untuk menyelesaikan sistem persamaan linier dengan model seperti yang telah dijelaskan sebelumnya. Peningkatan kinerja relatif dihitung dengan membandingkan waktu komputasi Serial CPU dengan waktu implementasi lainnya. Hasil Berdasarkan hasil di atas, kinerja yang lebih baik diperoleh pada implementasi pada GPU baik untuk *single precision* maupun *double precision*. Peningkatan kinerja dengan GPU diperoleh sebesar 3.86 dan 3.84 untuk *single precision* dan *double precision*. Pemberian *padding* pada format CSR juga memberikan peningkatan kinerja pada implementasi dengan GPU dibandingkan dengan tanpa adanya *padding*. Dengan *padding* diperoleh peningkatan kinerja 4.81 dan 4.26 untuk *single precision* dan *double precision*.

## KESIMPULAN

Implementasi pada GPU memberikan kinerja yang lebih baik dibandingkan implementasi serial pada CPU untuk keperluan pemodelan resistivitas DC 3D pada sistem yang digunakan. Peningkatan kinerja terhadap serial CPU yang diperoleh dengan menggunakan GPU adalah sebesar 3.86 dan 3.84 untuk *single precision* dan *double precision*. Dengan melakukan *padding* pada format CSR (PCSR) dapat diperoleh peningkatan kinerja 4.81 dan 4.26 untuk *single precision* dan *double precision*.

## UCAPAN TERIMA KASIH

Penulis pertama mengucapkan terima kasih kepada Lembaga Pengelola Dana Pendidikan Indonesia (LPDP) yang telah membantu dalam pembiayaan pendidikan pada program magister.

## REFERENSI

1. X. Wu, *A 3-D finite-element algorithm for DC resistivity modelling using the shifted incomplete Cholesky conjugate gradient method*. Geophysical Journal International, 154: 947–956. (2003)
2. A. C. Polycarpu, *An Introduction to the Finite Element Method in Electromagnetism*, Morgan & Claypool Publisher, USA, (2006)
3. D. L. Logan, *A First Course in the Finite Element Method*, Thomsons, Canada, (2007)
4. NVIDIA, *CUDA C Programming Guide*, (2015)

5. Y. Saad, *Iterative Methods for Sparse Linear Systems*, The Society for Industrial and Applied Mathematics (2003)