SIMULASI WAVEGUIDE SEDERHANA MENGGUNAKAN METODE GALERKIN DALAM MATLAB

ALATAS, H.[a,1], A.D. GARNADI[a,2], S. NURDIATI[2], T. PUJANEGARA[a], L. YULIAWATI[a]

Abstrak

Tulisan ini, merupakan sebuah tutorial bagaimana mengimplementasikan metode Galerkin untuk menyelesaikan persamaan Helmholtz. Persamaan ini, misalnya digunakan untuk memodelkan persamaan gelombang skalar. Misalnya untuk memperoleh informasi perilaku waveguide, persamaan Helmholtz perlu diselesaikan secara numerik, mengingat bentuk geometrik bahan penyusun menyebabkan solusi analitik sulit didapat. Persamaan gelombang skalar untuk waveguide isotropik homogen digunakan untuk memperkenalkan FEM. Untuk lebih sederhananya, digunakan elemen segitiga orde pertama. Makalah ini menunjukkan bagaimana pengetahuan tentang metode elemen hingga (FEM) dapat dipelajari dalam waktu singkat dengan menggunakan MATLAB. Hal ini menunjukkan bagaimana pengetahuan yang diperoleh dapat diperluas untuk masalah elektromagnetik lainnya.

PENDAHULUAN

Ketersediaan daya komputasi yang besar dan murah menggunakan komputer desktop atau laptop menjadikan solusi numerik dari permasalahan elektromagnetik menjadi hal yang layak, bahkan bagi mahasiswa sarjana sekalipun. Di kalangan pendidik telah diambil dua pendekatan: menggunakan perangkat lunak yang tersedia secara komersial [1] (yang mungkin menjadi pilihan yang mahal), atau desain antarmuka pengguna dan kode simulasi [2,3] berdasarkan paket matematis terprogram. Kedua pendekatan ini bukan merupakan obyek dari tulisan ini. Tujuan dari tulisan ini adalah memperkenalkan metode momen melalui Matlab dan menyelesaikan permasalahan elektromagnetik. Matlab telah digunakan di seluruh dunia dalam pengajaran banyak mata kuliah rekayasa, misalnya, pemrosesan sinyal dan teknik kontrol. Ini tidak akan mudah bagi para pengajar dalam bidang elektromagnetik untuk mengharapkan siswa untuk memiliki pengetahuan dan akses menggunakan Matlab. Metode Elemen Hingga (FEM) adalah teknik yang relatif mapan dalam elektromagnetik dan masih merupakan area penelitian yang cukup aktif. Hal ini didukung oleh beberapa buku teks dan monograf yang cukup baik tersedia. Tetapi, bahan tersebut hanya cocok bagi para peneliti atau untuk perkuliahan khusus. Selain itu, buku teks tersebut

[a] Research Cluster for Dynamics and Complex System, Fakultas Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor, 16680.
[1] Departemen Fisika, Fakultas Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor, 16680.
[2] Departemen Matematika, Fakultas Ilmu Pengetahuan Alam, Jalan Meranti Kampus IPB Dramaga Bogor, 16680.

sangat menekankan dasar matematika yang ketat dari berbagai formulasi berbagai FEM. Sebuah pendekatan praktis untuk penurunan dari FEM dan pelaksanaannya diberikan pada pustaka [8]. Pada tulisan ini diberikan uraian FEM yang singkat dan mudah dipahami melalui Matlab, mengingat Matlab menyediakan fasilitas untuk operasi matriks dan alokasi memori yang dinamis. Materi dalam tulisan ini dapat dibahas dalam kuliah selama dua jam. Mahasiswa mungkin diberikan kesempatan selama dua minggu untuk menyerap materi dan mengerjakan tugas yang serupa dengan contoh yang diberikan dalam tulisan ini. Perlu dicatat bahwa setidaknya satu paket perangkat lunak FEM komersial, yaitu FEMLAB dari COMSOL Inc, [10] memiliki versi sebagai add-on untuk MATLAB. Paket ini tidak cocok untuk mengajar pemrograman FEM. Namun, seperti yang digambarkan dalam rujukan [11], mahasiswa dapat menggunakan software ini untuk menguji formulasi variational yang diperoleh oleh mereka, mengingat perangkat lunak menerima bentuk formulasi variasional sebagai input. Dengan demikian, [11] dan tulisan ini saling melengkapi jika FEMLAB tersedia.

FORMULASI FEM

Waveguide memainkan peranan penting dalam perambatan gelombang berfrekuensi tinggi. Secara umum, pemodelan waveguide dimulai dari persamaan Maxwell kemudian ke persamaan Helmholtz. Dengan kata lain, dasar analisa perambatan gelombang di waveguide secara umum adalah persamaan Maxwell. Dengan menggunakan hukum Ampere dan hukum Faraday akan dijelaskan penurunan persamaan Maxwell. Pada tahun 1820 Ampere telah menurunkan hubungan arus listrik dan medan magnet ke dalam persamaan matematika. Hukum integral keliling Ampere menyatakan bahwa bila garis tertutup *C* mengelilingi kabel lurus *I* yang dialiri listrik, maka besaran integral tertutup sepanjang garis tertutup *C* untuk medan magnet *H* yang ditimbulkan oleh *I* adalah sama dengan besarnya arus listrik tersebut. Integral tertutup medan listrik dinyatakan sebagai jumlah keseluruhan garis tertutup C yang terbagi pada bagian kecil *dl* dikalikan dengan medan listrik sejajar *dl*.

$$\oint_C H \bullet dl = I. \tag{1}$$

Misalkan *J* menyatakan kepadatan arus listrik pada tiap satuan unit arus listrik dan arus listrik tersebut menembus penampang *S*. Asumsikan bahwa penampang tersebut bukan dalam batang tetapi penampang yang terhingga. Penampang tersebut memiliki vektor normal *n* dan memiliki keliling *C*. Komponen arus listrik yang melalui penampang S secara tegak lurus dinyatakan oleh $J \bullet n$. Dengan mengintegralkan permukaan penampang S diperoleh keseluruhan arus listrik pada penampang *S*. Oleh karena itu bagian kanan pada persamaan (1) dapat ditulis

$$\oint_C H \bullet dl = \int_S J \bullet n \, dS. \tag{2}$$

Pada tahun 1831 Faraday membuktikan fenomena perubahan medan magnet yang menimbulkan arus listrik. Hukum Faraday menyatakan bahwa perubahan medan listrik mengakibatkan tegangan di sepanjang loop tertutup *C*. Pada saat induksi magnet *B* yang menembus loop tertutup *C* berubah maka besaran tersebut sama dengan berkurangnya tegangan sesuai dengan perubahan waktu di loop tertutup *C* tersebut. Fenomena ini dapat dinyatakan oleh

$$\oint_C E \bullet dl = -\frac{\partial}{\partial t} \int_S B \bullet n \, dS. \tag{3}$$

Hukum integral keliling Ampere dan hukum Faraday dapat digabungkan dengan menggunakan perubahan listrik. Hal ini dapat diilustrasikan oleh arus listrik AC yang dialirkan ke kondensator. Di dalam kondensator tersimpan muatan listrik yang mengalami perubahan menurut waktu sehingga kondensator mengalirkan arus listrik. Rasio perubahan waktu terhadap perubahan listrik *D* disebut sebagai perubahan arus listrik yang sama dengan arus listrik dalam hukum Ampere. Akibatnya persamaan (2) menjadi

$$\oint_C H \bullet dl = \int_S J \bullet n \, dS + \frac{\partial}{\partial t} \int_S D \bullet n \, dS. \tag{4}$$

Hukum integral keliling Ampere ini dapat digabungkan dengan hukum Faraday dan persamaan tersebut merupakan persamaan dasar Maxwell. Dengan menggunakan teori Stokes, ruas kiri pada persamaan (3) dan (4) dapat diubah menjadi integral permukaan. Permukaan yang diintegralkan harus diambil sekecil mungkin supaya berlaku di seluruh ruang. Akibatnya berlaku penurunan persamaan Maxwell berikut

$$\nabla \times H = J + \frac{\partial D}{\partial t} \tag{5}$$

$$\nabla \times E = -\frac{\partial B}{\partial t}. \tag{6}$$

Dalam hukum dasar elektromagnet nilai muatan listrik sama dengan jumlah perubahan listrik yang ditimbulkannya dan dikenal sebagai hukum Gauss untuk perubahan listrik. Hukum Gauss yang lain yang berhubungan dengan perubahan listrik yaitu tidak ada fenomena muatan listrik yang hanya mempunyai satu kutub saja. Kedua fenomena di atas dapat digambarkan oleh persamaan di bawah ini

$$\nabla \bullet D = \rho \tag{7}$$

$$\nabla \bullet B = 0 \tag{8}$$

Persamaan (5) hingga 8) memuat banyak variabel diantaranya variabel medan listrik *E*, perubahan listrik *D*, medan listrik *H* dan induksi magnet *B*. Untuk mendapatkan medan listrik dan medan magnet akan dilakukan penurunan pada persamaan-persamaan tersebut.

Suatu medium berdielektrik yang homogen digambarkan di seluruh ruang analisa dengan permitivitas ε dan permeabilitas μ. Pada medium homogen,

perubahan listrik dan induksi magnet dapat dinyatakan sebagai $D = \varepsilon E$ dan $B = \mu H$. Dengan mensubstitusikan kedua persamaan ini pada persamaan dasar Maxwell maka diperoleh persamaan medan listrik dan medan magnet. Untuk mendapatkan persamaan medan listrik dilakukan penghapusan medan magnet pada persamaan Maxwell dengan menggunakan operasi putaran pada persamaan (6) sehingga diperoleh

$$\nabla \times \nabla \times E + \mu \frac{\partial}{\partial t}\left(\nabla \times H\right) = 0. \tag{9}$$

Dengan menggunakan hubungan $\nabla \bullet E = \rho / \varepsilon$ dan $\nabla \times \nabla \times A = \nabla\left(\nabla \bullet A\right) - \nabla^2 A$ maka persamaan (9) menjadi

$$\nabla \frac{\rho}{\varepsilon} - \nabla^2 E + \mu \frac{\partial J}{\partial t} + \varepsilon\mu \frac{\partial^2 J}{\partial t^2} = 0. \tag{10}$$

Dengan menggunakan operasi putaran pada persamaan (5) kemudian disubstitusikan pada persamaan (6) untuk menghilangkan medan listrik dari persamaan Maxwell maka diperoleh persamaan di bawah ini.

$$-\nabla^2 H - \nabla \times J + \varepsilon\mu \frac{\partial^2 H}{\partial t^2} = 0. \tag{11}$$

Medan listrik dan medan magnet dalam permasalahan elektromagnet sebagian besar berubah sebagai persamaan sinus dengan frekuensi angular ω pada setiap perubahan waktu, misalnya $e^{j\omega t}$. Akibatnya persamaan medan listrik dan medan magnet dapat ditulis sebagai berikut.

$$\nabla^2 E + l^2 E = \nabla \frac{\rho}{\varepsilon} + j\omega\mu J \tag{12}$$

$$\nabla^2 H + l^2 H = -\nabla \times J \tag{13}$$

$$l = \omega\sqrt{\varepsilon\mu} = \frac{2\pi f}{v} = \frac{2\pi}{\lambda} \tag{14}$$

Persamaan (12) dan (13) disebut sebagai persamaan Helmholtz dengan l adalah tetapan hantar. Karena $1/\sqrt{\varepsilon\mu}$ memiliki satuan kecepatan yang ditunjukkan oleh $v(m = s)$ sedangkan $f(Hz)$ frekuensi dan kecepatan angular $\omega = 2\pi f \cdot f / v$ mempunyai satuan panjang maka λ disebut sebagai panjang gelombang. Jika suatu medan listrik dan medang maget di titik yang sangat jauh dari sumber gelombang maka keberadaan muatan listrik ρ dan arus listrik J sebagai sumber gelombang dapat tidak dianggap. Oleh karena itu diperoleh persamaan Helmholtz berikut.

$$\nabla^2 E + l^2 E = 0 \tag{15}$$

$$\nabla^2 H + l^2 H = 0. \tag{16}$$

Untuk mempelajari penggunaan FEM melalui Matlab, pada pembahasan lebih lanjut digunakan persamaan Helmholtz berikut

$$-\nabla^2 u + ku = 0, \tag{17}$$

dengan $k < 0, k = -l^2$ dan u sebagai variabel medan listrik atau medan magnet.

FEM memecahkan persamaan (17) dengan meminimumkan fungsional bentuk lemah yang diberikan oleh:

$$F(u,u) = \int_{\Omega} (\nabla u \cdot \nabla u + kuu)\, dx. \tag{18}$$

Gelombang datar adalah gelombang ideal dimana gelombang elektromagnet ini terhantar di ruang bebas dengan kecepatan cahaya. Oleh karena itu, gelombang datar merupakan gelombang di lokasi tak terhingga dari sumber gelombang yang menimbulkannya. Pada saat gelombang datar terhantar di ruang yang homogen, amplitudo gelombang tidak akan berubah dan fasenya tidak mengalami ketidakkontinyuan. Tetapi pada saat gelombang datar melewati medium yang mempunyai tetapan medium yang berlainan maka komponen medan listrik dan medan magnet di perbatasan antar medium tersebut harus memenuhi syarat batas. Untuk kemudahan, kita perhatikan beberapa kasus berikut.

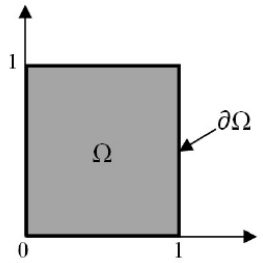
Kasus 1

Perhatikan persamaan Helmholtz dengan syarat batas Dirichlet berikut

$$\begin{aligned} -\nabla^2 u + ku &= 0 \text{ di } \Omega \\ u &= g \text{ di } \partial\Omega \end{aligned} \tag{19}$$

di mana k dan g adalah sembarang fungsi yang diketahui. Hal ini dapat diilustrasikan untuk gelombang datar yang melewati medium kedua yang merupakan penghantar sempurna.

Misalkan $\Omega = [0,1] \times [0,1]$ dan syarat batas $\partial\Omega$ yang diilustrasikan pada Gambar 1.



Gambar 1 Daerah Ω

Kasus 2

Hal lain yang mungkin jika gelombang datar tersebut mengalami perubahan setelah melewati medium kedua. Perhatikan persamaan Helmholtz dengan syarat batas Dirichlet dan Neumann berikut

$$\begin{aligned} -\nabla\cdot\nabla u + ku &= 0 \text{ di } \Omega \\ u &= p \text{ di } L_1 \\ n\cdot\nabla u &= q \text{ di } L_2 \end{aligned} \tag{20}$$

dimana *k*, *p*, dan *q* adalah fungsi-fungsi yang diketahui. Misalkan $\Omega = [0,1]\times[0,1]$ dengan syarat batas Dirichlet di L_1 dan syarat batas Neumann di L_2.

HAMPIRAN GALERKIN

Untuk menyelesaikan kasus-kasus yang sudah diuraikan di atas secara numerik digunakan FEM dengan hampiran Galerkin. Simulasi untuk kedua kasus ini akan diuraikan dengan menggunakan Matlab.

Kasus 1

Misal $v \in \mathcal{H}^1$ di mana $\mathcal{H}^1$ adalah ruang Hilbert $\mathcal{H}^1 \left\{v : \|v\| + \|\nabla v\| < \infty\right\}$. Sehingga bentuk lemah kasus 1 adalah

$$\int_\Omega \left(\nabla u\cdot\nabla v + kuv\right) dx = 0 \tag{21}$$

Misal $V_h \in \mathcal{H}^1$ adalah ruang dari semua fungsi *piecewise linear* yang kontinu dalam sebuah partisi K = {T} dari S menjadi segitiga berukuran |T|. Himpunan hat function didefinisikan $\varphi_i\ _{i=1}^{N}$ di mana *N* menyatakan banyaknya node dalam triangulasi T yang merupakan basis untuk V_h. Pendekatan elemen hingga (finite element approximation) dari bentuk lemah (weak form) persamaan (19) adalah $u_h \in V_h$ sedemikian sehingga

$$\int_\Omega \left(\nabla u_h\cdot\nabla v + ku_h v\right) dx = 0 \tag{22}$$

Karena $u_h \in V_h$ maka u_h dapat ditulis

$$u_h = \sum_{j=1}^{N} z_j \varphi_j .$$

Oleh karena itu persamaan (22) dapat ditulis

$$\sum_{j=1}^{N} z_j \left(\int_\Omega \nabla\varphi_j \nabla\varphi_i \ dx + \int_\Omega k\varphi_j\varphi_i \ dx \right) = 0. \tag{23}$$

Dari persamaan (23) diperoleh sistem persamaan linear *Az* = *b*, di mana elemen-elemen sistem persamaan linear tersebut sebagai berikut

$$A_{i,j} = \int_{\Omega} \nabla\varphi_j \nabla\varphi_i \, dx + \int_{\Omega} k\varphi_j \varphi_i \, dx \tag{24}$$

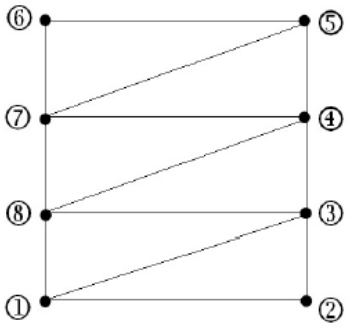
$$b_i = 0 \tag{25}$$

$$z = \left(z_1, z_2, ..., z_N\right)^T . \tag{26}$$

Pada persamaan di atas *j* bergerak sepanjang node dan *i* bergerak sepanjang node dengan nilai *u* diketahui. Nilai *u* pada $\partial\Omega$ yang diketahui dinotasikan dalam vektor z_e dan nilai *u* selainnya dinotasikan dalam z_n,

$$\left(A_e \mid A_n\right)\begin{pmatrix} z_e \\ z_n \end{pmatrix} = A_e z_e + A_n z_n = b$$

matriks A_e dihitung pada syarat batas. Sehingga solusi akhir z_n dapat dihitung dengan sistem persamaan $A_n z_n = b - A_e z_e$ dimana A_n sudah diketahui. Misalkan daerah dibagi ke dalam enam ($N = 6$) segitiga yang diperlihatkan pada Gambar 2.



Gambar 2 Ilustrasi mesh (enam elemen segitiga) dalam Ω

Misalkan koordinat dari setiap node diberikan pada tabel 1.

TABEL 1
Nomor-nomor koordinat sebagai node

Node	1	2	3	4	5	6	7	8
x	0	1	1	1	1	0	0	0
y	0	0	1/3	2/3	1	1	2/3	2/3

Matlab mampu membaca data dari file yang diberikan dalam format ascii dengan file .dat, sehingga dalam Matlab kita dapat menyimpan data koordinat untuk setiap node dengan pemrograman sebagai berikut.

```
node1 = [1 0 0;
     2 1 0;
     3 1 1/3;
     4 1 2/3;
     5 1 1;
     6 0 1;
     7 0 2/3;
     8 0 1/3];
```

```
dlmwrite('koordinat1.dat', node1,' ')
type koordinat1.dat
```

Pada Tabel 2 diberikan elemen-elemen yang berpadanan dengan node-node pada Tabel 1.

TABEL 2
Nomor-nomor elemen segitiga

Elemen	1	2	3	4	5	6
Node 1	7	7	8	8	1	1
Node 2	5	4	4	3	3	2
Node 3	6	5	7	4	8	3

Dengan menggunakan Matlab kita dapat menyimpan data elemen-elemen di atas dengan pemrograman berikut.

```
segitiga1=[1 7 5 6;
    2 7 4 5;
    3 8 4 7;
    4 8 3 4;
    5 1 3 8;
    6 1 2 3];
dlmwrite('elemen1.dat', segitiga1,' ')
type elemen1.dat
```

Syarat batas untuk Gambar 1 diperlihatkan pada tabel berikut.

TABEL 3
Nomor-nomor edge dengan syarat batas

Edge	1	2	3	4	5	6	7	8
Node 1	1	2	3	4	5	6	7	8
Node 2	2	3	4	5	6	7	8	1

```
deltaomega=[1 1 2;
     2 2 3;
     3 3 4;
     4 4 5;
     5 5 6;
     6 6 7;
     7 7 8;
     8 8 1];
dlmwrite('deltaomega.dat', deltaomega,' ')
type deltaomega.dat
```

Menyusun matriks *stiffness*

Perhatikan bahwa untuk sebuah elemen segitiga T misalkan (x_1, y_1), (x_2, y_2), dan (x_3, y_3) adalah titik-titik verteks dan $\varphi_1, \varphi_2,$ dan φ_3 adalah fungsi basis yang berkorespondensi di K, yaitu

$$\varphi_i\left(x_j, y_j\right) = \delta_{ij}, \qquad i, j = 1,2,3$$

Oleh karena itu, diperoleh

$$\varphi_i\left(x, y\right) = \frac{\det\begin{pmatrix} 1 & x & y \\ 1 & x_{i+1} & y_{i+1} \\ 1 & x_{i+2} & y_{i+2} \end{pmatrix}}{\det\begin{pmatrix} 1 & x_i & y_i \\ 1 & x_{i+1} & y_{i+1} \\ 1 & x_{i+2} & y_{i+2} \end{pmatrix}}$$

dengan

$$\nabla\varphi_i\left(x, y\right) = \frac{1}{2|T|}\begin{pmatrix} y_{i+1} - y_{i+2} \\ x_{i+2} - x_{i+1} \end{pmatrix}. \tag{28}$$

Indeks yang digunakan pada persamaan di atas di dalam modulo 3, sehingga diperoleh

$$2|T| = \det\begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix}.$$

Matriks *stiffness* yang diperoleh adalah

$$M_{ij} = \int_T \nabla\varphi_i \left(\nabla\varphi_j\right)^T dx$$

$$= \frac{|T|}{\left(2|T|\right)^2}\begin{pmatrix} y_{i+1} - y_{1+2} & x_{i+2} - x_{i+1} \end{pmatrix}\begin{pmatrix} y_{j+1} - y_{j+2} \\ x_{j+2} - x_{j+1} \end{pmatrix}.$$

dengan indeks di dalam modulo 3. Bentuk sederhana persamaan di atas adalah

$$M = \frac{|T|}{2} GG^T,$$

di mana

$$G = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix}^{-1}\begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Pemrograman dalam Matlab untuk menyusun matriks *stiffness* dapat kita tuliskan sebagai berikut.

```
function M = stima(vertices)
d = size(vertices,2);
G = [ones(1,d+1);vertices'] \ [zeros(1,d);eye(d)];
```

```
M = det([ones(1,d+1);vertices']) * G * G' / prod(1:d);
end
```

Menyusun Matriks Massa

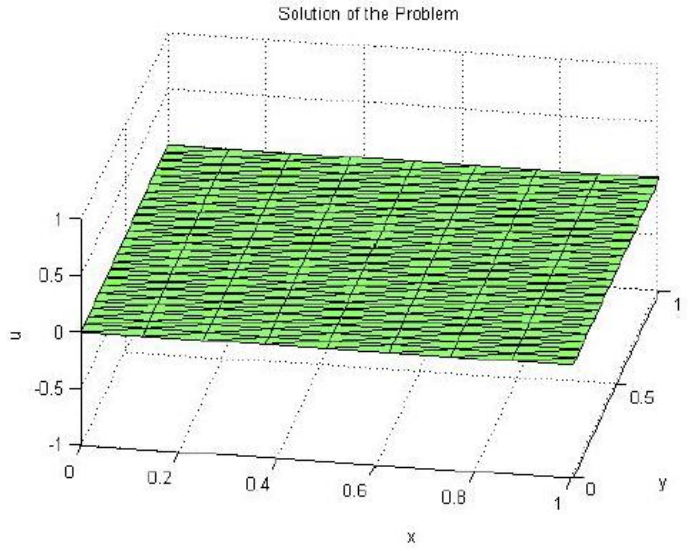
Perhatikan bahwa

$$B_{ij} = \sum_{T\in\Omega}\int_T \varphi_i\varphi_j k\ dx.$$

Dengan cara yang hampir sama dalam menyusun matriks *stiffness*, diperoleh matriks massa

$$\int_T \varphi_i\varphi_j k dx = \frac{1}{24}\det\begin{pmatrix} x_2-x_1 & x_3-x_1 \\ y_2-y_1 & y_3-y_1 \end{pmatrix}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} k\left(x_s, y_s\right)$$

di mana (x_s, y_s) adalah titik tengah pada segitiga T.
Untuk $k = 0$, dengan menggunakan Matlab diperoleh solusi dalam bentuk grafik berikut ini.



Gambar 3 Solusi kasus 1 untuk $k = 0$

Solusi analitik pada kasus 1 untuk $k = 0$ adalah $u = 0$ mempunyai kesimpulan yang sama dengan solusi numerik yang ditampilkan pada Gambar 3.

```
function k = k0(x,y);
k = 0;
end
function SyaratBatas = deltaO(x)
SyaratBatas = zeros(size(x,1),1);
end
```

Pemrograman dengan Matlab untuk kasus 1 sebagai berikut:

```
load koordinat1.dat; koordinat1(:,1)=[]; % koordinat-
koordinat
load elemen1.dat; elemen1(:,1)=[]; % elemen hingga berbentuk
segitiga
load deltaomega.dat; deltaomega(:,1)=[];
koordinat = koordinat1;
elemen = elemen1;
for k = 1:3
[Kantennr,elemen] = GeneriereKantennr(elemen,koordinat);
VK = (1:full(max(max(Kantennr))))' + size(koordinat,1);
[koordinat,elemen] = ...
Verfeinerung(koordinat,elemen,[],[],Kantennr,VK);
end
FreeNodes=setdiff(1:size(koordinat,1),unique(deltaomega));
A = sparse(size(koordinat,1),size(koordinat,1));
b = sparse(size(koordinat,1),1);
% 1. menyusun matriks A1
% dengan memanggil fungsi stima
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) = A(elemen(i,:),elemen(i,:)) ...
+ stima(koordinat(elemen(i,:),:));
end
% 2. menyusun matriks A2
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) = A(elemen(i,:),elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
(diag(ones(3,1))+
ones(3))*k0(sum(koordinat(elemen(i,:),:))/3)/24;
end
u = sparse(size(koordinat,1),1);
u(unique(deltaomega)) =
delta0(koordinat(unique(deltaomega),:));
b = b - A*u;
% 3. Computation of the solution
u(FreeNodes) = A(FreeNodes,FreeNodes) \ b(FreeNodes);
% 4. Graphic representation
show(elemen,[],koordinat,full(u));
xlabel('x');ylabel('y');zlabel('u');
```

Kasus 2

Bentuk lemah untuk kasus 2 adalah

$$\int_{\Omega} \left(\nabla u \cdot \nabla v + kuv\right) dx = \int_{L_2} qv \, ds. \tag{29}$$

Pendekatan elemen hingga (*finite element approximation*) dari bentuk lemah persamaan (29) adalah $u_h \in V_h$ sedemikian sehingga

$$\int_{\Omega} \left(\nabla u_h \cdot \nabla v + k u_h v \right) dx = \int_{L_2} qv \, ds. \tag{30}$$

Dengan cara yang sama seperti yang sudah diuraikan pada kasus 1, terdapat sistem persamaan linear dengan elemen-elemen berikut.

$$A_{ij} = \left(\int_{\Omega} \nabla \varphi_i \nabla \varphi_j \, dx + \int_{\Omega} \varphi_i \varphi_j \, dx \right)$$

$$b_i = \int_{L_2} \varphi_i q \, ds$$

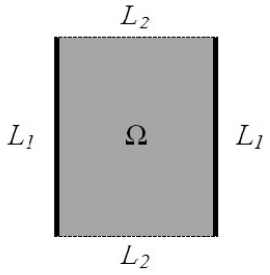
$$z = \left(z_1, z_2, ..., z_N \right)^T .$$

Dengan menggunakan cara yang serupa pada kasus 1, matriks A_e dihitung pada syarat batas Dirichlet. Sehingga solusi akhir z_n dapat dihitung dengan sistem persamaan

$$A_n z_n = b - A_e z_e .$$

Beberapa simulasi untuk kasus 2 akan dibahas di bawah ini.

a. Misal Ω dan syarat batas diberikan pada gambar berikut. Jika Ω dibagi



Gambar 4 Syarat batas untuk kasus 2a

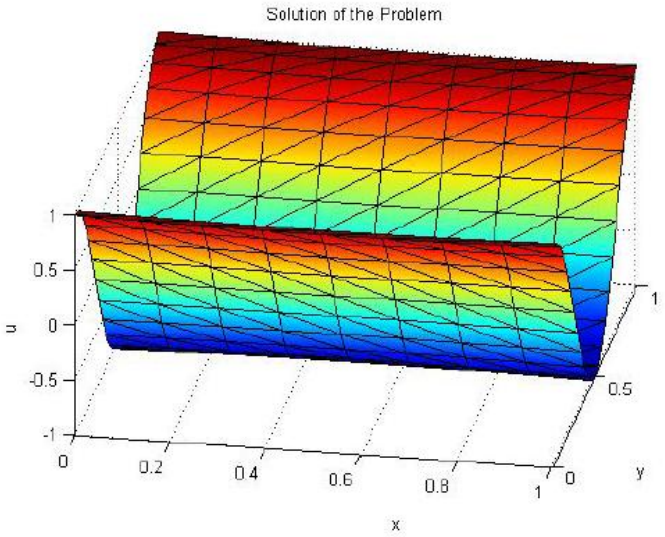
ke dalam enam segitiga seperti pada gambar 1 dengan syarat batas seperti pada Gambar 4, maka dalam Matlab kita dapat menyimpan data syarat batas dengan pemrograman berikut.

```
neumann1=[1 1 2;
2 5 6];
dlmwrite('neumann1.dat',neumann1,' ')
type neumann1.dat
dirichlet1=[1 2 3;
2 3 4;
3 4 5;
4 6 7;
5 7 8;
6 8 1];
dlmwrite('dirichlet1.dat', dirichlet1,' ')
type dirichlet1.dat
```

Misalkan dipilih $k = -(2\pi)^2$, $p = \cos(2\pi y)$ dan $q = 0$.

```
function k = k1(x,y);
k = -((2*pi))^2;
end
function S = g(x)
S = zeros(size(x,1),1);
end
function a=q1(x);
a=0;
end
function BatasDirichletKasus2a = ud1(x)
BatasDirichletKasus2a = zeros(size(x,1),1);
I=find(x(:,1)==0|x(:,1)==1);
BatasDirichletKasus2a (I) = cos(2*pi*x(:,2));
end
```

Dengan menggunakan Matlab diperoleh solusi pada Gambar 5.



Gambar 5 Solusi kasus 2a

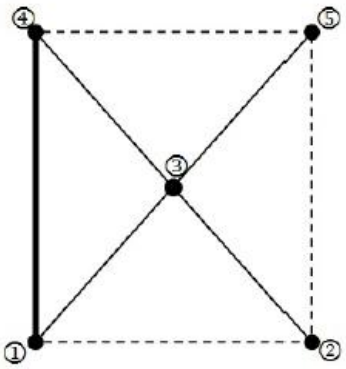
Pemrograman dengan Matlab sebagai berikut:

```
% inisialisasi
load koordinat1.dat; koordinat1(:,1)=[]; % koordinat-
koordinat
load elemen1.dat; elemen1(:,1)=[]; % elemen hingga
berbentuk segitiga
load neumann1.dat; neumann1 (:,1)=[];
load dirichlet1.dat; dirichlet1 (:,1)=[];
koordinat = koordinat1;
elemen = elemen1;
```

```
neumann = neumann1;
dirichlet = dirichlet1;
for k = 1:3
[Kantennr,elemen] = GeneriereKantennr(elemen,koordinat);
VK = (1:full(max(max(Kantennr)))' + size(koordinat,1);
[koordinat,elemen,dirichlet,neumann] = ...
Verfeinerung(koordinat,elemen,dirichlet,neumann,Kantennr
,VK);
end
FreeNodes=setdiff(1:size(koordinat,1),unique(dirichlet))
;
A = sparse(size(koordinat,1),size(koordinat,1));
b = sparse(size(koordinat,1),1);
% 1. menyusun matriks A1
% dengan memanggil fungsi stima_1
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) =
A(elemen(i,:),elemen(i,:)) ...
+ stima(koordinat(elemen(i,:),:));
end
% 2. menyusun matriks A2
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) =
A(elemen(i,:),elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
(diag(ones(3,1))+
ones(3))*k1(sum(koordinat(elemen(i,:),:))/3)/24;
end
% 3. volume forces
for i = 1:size(elemen,1)
b(elemen(i,:)) = b(elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
g(sum(koordinat(elemen(i,:))/3))/6;
end
% 4. Neumann condition
for i = 1:size(neumann,1)
b(neumann(i,:))=b(neumann(i,:)) +...
norm(koordinat(neumann(i,1),:)- ...
koordinat(neumann(i,2),:)) *...
q1(sum(koordinat(neumann(i,:),:))/2)/2;
end
% 5. Dirichlet conditions
u = sparse(size(koordinat,1),1);
u(unique(dirichlet)) =
ud1(koordinat(unique(dirichlet),:));
b = b - A*u;
% 6. Computation of the solution
u(FreeNodes) = A(FreeNodes,FreeNodes) \ b(FreeNodes);
```

```
% 7. Graphic representation
show(elemen,[],koordinat,full(u));
xlabel('x');ylabel('y');zlabel('u');
```

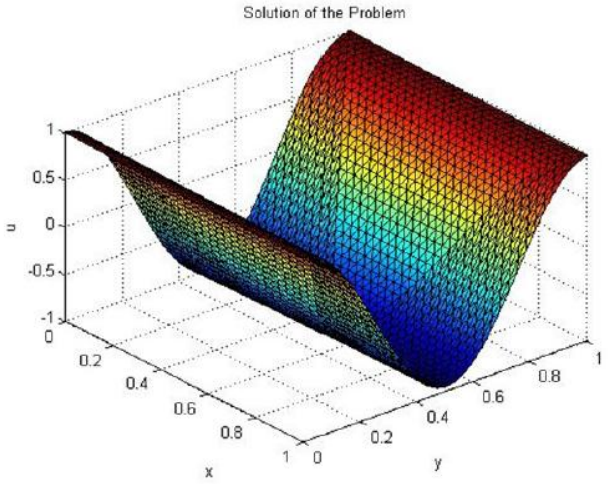
b. Misalkan daerah dibagi ke dalam empat (N = 4) segitiga yang diperlihatkan pada gambar berikut ini Dengan nomor-nomor node, elemen segitiga dan edge syarat batas pada tabel berikut: Misalkan $k=-(2\pi)^2$, $p=\cos(2\pi y)$ dan q = 0, dengan menggunakan Matlab, diperoleh solusi dalam bentuk grafik berikut ini:



Gambar 6 Daerah dibagi ke dalam empat segitiga

TABEL 4
Nomor-nomor node, elemen, dan edge syarat batas

Nomor	Node (koordinat)	Elemen Segitiga	Syarat Batas Neumann	Syarat Batas Dirichlet
1	(0,0)	1-2-3	1-2	4-1
2	(1,0)	1-3-4	2-5	
3	(1/2,1/2)	3-2-5	5-4	
4	(0,1)	3-5-4		
5	(1,1)			



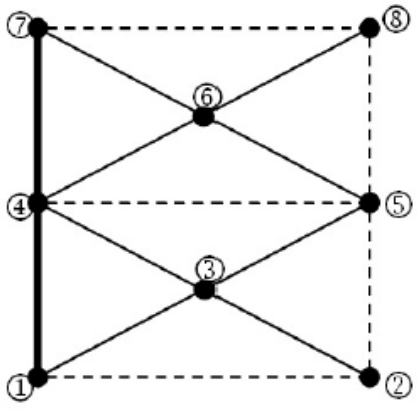
Gambar 7 Solusi Kasus 2b

Pemrograman dengan Matlab sebagai berikut:

```
load koordinat2.dat; koordinat2(:,1)=[]; % koordinat-
koordinat
load elemen2.dat; elemen2(:,1)=[]; % elemen hingga
berbentuk segitiga
load neumann3.dat; neumann3(:,1)=[];
load dirichlet3.dat; dirichlet3(:,1)=[];
koordinat=koordinat2;
elemen=elemen2;
neumann = neumann3;
dirichlet = dirichlet3;
for k = 1:5
[Kantennr,elemen] = GeneriereKantennr(elemen,koordinat);
VK = (1:full(max(max(Kantennr))))' + size(koordinat,1);
[koordinat,elemen,dirichlet,neumann] = ...
Verfeinerung(koordinat,elemen,dirichlet,neumann,Kantennr
,VK);
end
FreeNodes=setdiff(1:size(koordinat,1),unique(dirichlet))
;
A = sparse(size(koordinat,1),size(koordinat,1));
b = sparse(size(koordinat,1),1);
% 1. menyusun matriks A1
% dengan memanggil fungsi stima_1
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) =
A(elemen(i,:),elemen(i,:)) ...
+ stima(koordinat(elemen(i,:),:));
end
% 2. menyusun matriks A2
for i = 1:size(elemen,1)
```

```
A(elemen(i,:),elemen(i,:)) =
A(elemen(i,:),elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
(diag(ones(3,1))+
ones(3))*k1(sum(koordinat(elemen(i,:),:))/3)/24;
end
% 3. volume forces
for i = 1:size(elemen,1)
b(elemen(i,:)) = b(elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
g(sum(koordinat(elemen(i,:))/3))/6;
end
% 4. Neumann condition
for i = 1:size(neumann,1)
b(neumann(i,:))=b(neumann(i,:)) +...
norm(koordinat(neumann(i,1),:)- ...
koordinat(neumann(i,2),:)) *...
q1(sum(koordinat(neumann(i,:),:))/2)/2;
end
% 5. Dirichlet conditions
u = sparse(size(koordinat,1),1);
u(unique(dirichlet)) =
ud1(koordinat(unique(dirichlet),:));
b = b - A*u;
% 6. Computation of the solution
u(FreeNodes) = A(FreeNodes,FreeNodes) \ b(FreeNodes);
% 7. Graphic representation
show(elemen,[],koordinat,full(u));
xlabel('x');ylabel('y');zlabel('u');
```

c. Misalkan daerah dibagi ke dalam delapan ($N = 8$) segitiga yang diperlihatkan pada gambar berikut ini. Dengan nomor-nomor node, elemen



Gambar 8 Daerah dibagi ke dalam delapan segitiga

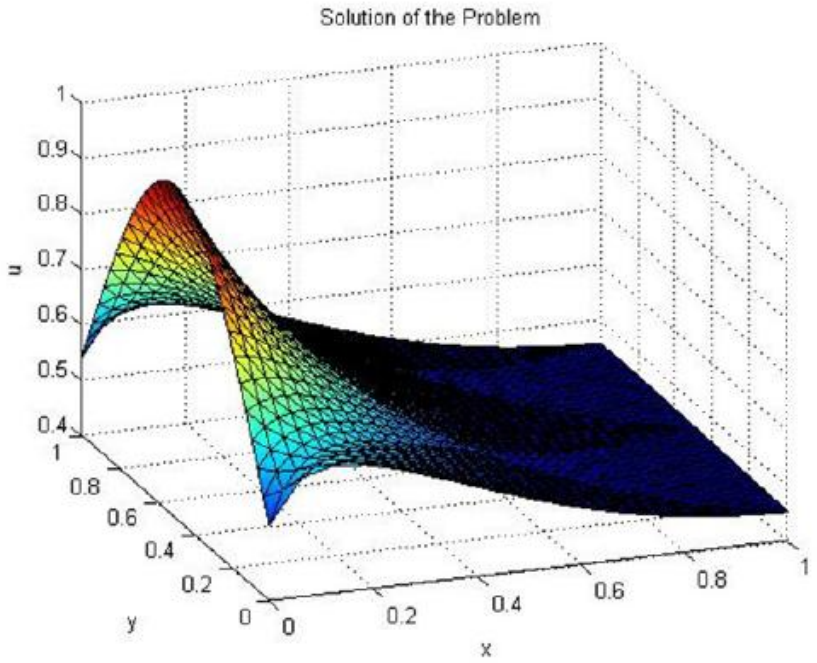
segitiga dan edge syarat batas pada Tabel 5.

TABEL 5
Nomor-nomor node, elemen, dan edge syarat batas

Nomor	Node (koordinat)	Elemen Segitiga	Syarat Batas Neumann	Syarat Batas Dirichlet
1.	(0,0)	1-2-3	1-2	7-4
2.	(1,0)	1-3-4	2-5	4-1
3.	(1/2,1/4)	3-2-5	5-8	
4.	(0,1/2)	4-3-5	8-7	
5.	(1,1/2)	4-5-6		
6.	(1/2,3/4)	4-6-7		
7.	(0,1)	6-5-8		
8.	(1,1)	7-6-8		

```
function k = k3(x)
if x(:,2)<=0.5
k = 1.5;
else if x(:,2)>0.5
k = 1.4;
19
end
end
end
function DirichletBoundaryValueCase2d = ud3(x)
DirichletBoundaryValueCase2d = zeros(size(x,1),1);
I=find(x(:,1)==0);
DirichletBoundaryValueCase2d(I) = exp((-(x(I,2)-
0.5).^2)./0.4);
end
```

Dengan menggunakan Matlab, diperoleh solusi dalam bentuk grafik berikut ini:



Gambar 9 Solusi Kasus 2c

Pemrograman dengan Matlab sebagai berikut:

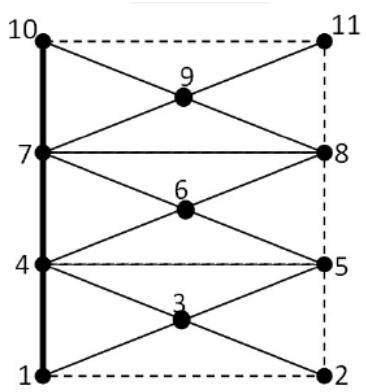
```
load koordinat3.dat; koordinat3(:,1)=[]; % koordinat-
koordinat
load elemen3.dat; elemen3(:,1)=[]; % elemen hingga
berbentuk segitiga
load neumann4.dat; neumann4(:,1)=[];
load dirichlet4.dat; dirichlet4(:,1)=[];
koordinat=koordinat3;
elemen=elemen3;
neumann = neumann4;
dirichlet = dirichlet4;
for k = 1:4
[Kantennr,elemen] = GeneriereKantennr(elemen,koordinat);
VK = (1:full(max(max(Kantennr))))' + size(koordinat,1);
[koordinat,elemen,dirichlet,neumann] = ...
Verfeinerung(koordinat,elemen,dirichlet,neumann,Kantennr
,VK);
end
20
FreeNodes=setdiff(1:size(koordinat,1),unique(dirichlet))
;
A = sparse(size(koordinat,1),size(koordinat,1));
b = sparse(size(koordinat,1),1);
% 1. menyusun matriks A1
% dengan memanggil fungsi stima_1
```

```
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) =
A(elemen(i,:),elemen(i,:)) ...
+ stima(koordinat(elemen(i,:),:));
end
% 2. menyusun matriks A2
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) =
A(elemen(i,:),elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
(diag(ones(3,1))+
ones(3))*k2(sum(koordinat(elemen(i,:),:))/3)/24;
end
% 3. volume forces
for i = 1:size(elemen,1)
b(elemen(i,:)) = b(elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
g(sum(koordinat(elemen(i,:))/3))/6;
end
% 4. Neumann condition
for i = 1:size(neumann,1)
b(neumann(i,:))=b(neumann(i,:)) +...
norm(koordinat(neumann(i,1),:)- ...
koordinat(neumann(i,2),:)) *...
q1(sum(koordinat(neumann(i,:),:))/2)/2;
end
% 5. Dirichlet conditions
u = sparse(size(koordinat,1),1);
u(unique(dirichlet)) =
ud3(koordinat(unique(dirichlet),:));
b = b - A*u;
% 6. Computation of the solution
u(FreeNodes) = A(FreeNodes,FreeNodes) \ b(FreeNodes);
% 7. Graphic representation
show(elemen,[],koordinat,full(u));
xlabel('x');ylabel('y');zlabel('u');
```

d. Misalkan daerah Ω dibagi ke dalam duabelas (N = 12) segitiga yang diperlihatkan pada Gambar 10. Sedangkan nomor-nomor node, elemen segitiga dan node dengan syarat batas pada Tabel 6. Misalkan

$$k = \begin{cases} 1.4; & y > 2/3 \text{ atau } y < 1/3 \\ 1.5; & 1/3 \le y \le 2/3 \end{cases}$$

sedangkan p dan q sama seperti pada kasus 2c.



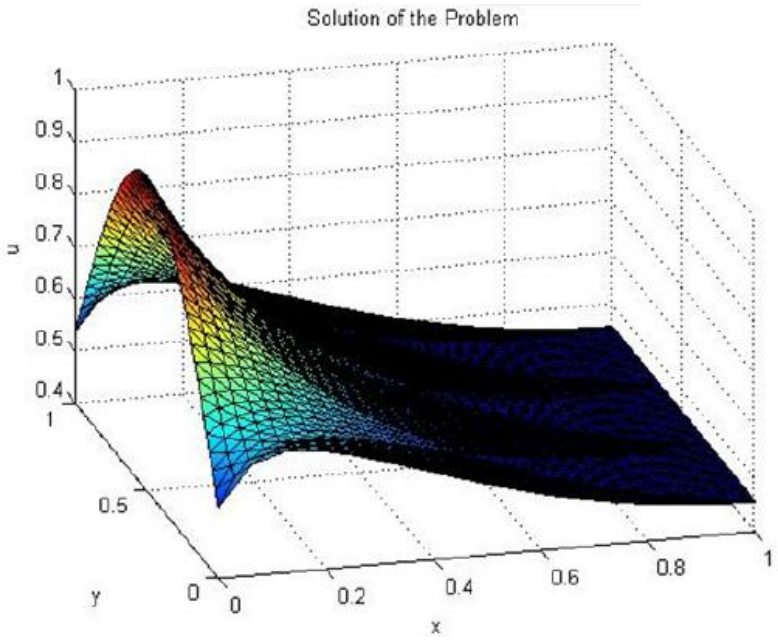
Gambar 10 Daerah Ω dibagi ke dalam duabelas segitiga

TABEL 6
Nomor-nomor node, elemen, dan edge syarat batas

Nomor	Node (koordinat)	Elemen Segitiga	Syarat Batas Neumann	Syarat Batas Dirichlet
1	(0,0)	1-2-3	1-2	10-7
2	(1,0)	1-3-4	2-5	7-4
3	(1/2,1/6)	3-2-5	5-8	4-1
4	(0,1/3)	3-5-4	8-11	
5	(1,1/3)	4-5-6	11-10	
6	(1/2,1/2)	4-6-7		
7	(0,2/3)	6-5-8		
8	(1,2/3)	6-8-7		
9	(1/2,5/6)	7-8-9		
10	(0,1)	7-9-10		
11	(1,1)	9-8-11		
12		9-10-11		

```
function k = k4(x)
if (x(:,2)<1/3)||(x(:,2)>2/3)
k = 1.4;
else
k = 1.5;
end
end
```

Dengan menggunakan Matlab, diperoleh solusi dalam bentuk grafik berikut ini:



Gambar 11 Solusi Grafik Kasus 2d

Pemrograman dengan Matlab sebagai berikut:

```
load koordinat4.dat; koordinat4(:,1)=[]; % koordinat-
koordinat
load elemen4.dat; elemen4(:,1)=[]; % elemen hingga berbentuk
segitiga
load neumann5.dat; neumann5(:,1)=[];
load dirichlet5.dat; dirichlet5(:,1)=[];
koordinat=koordinat4;
elemen=elemenB4;
neumann = neumann5;
dirichlet = dirichlet5;
for k = 1:4
[Kantennr,elemen] = GeneriereKantennr(elemen,koordinat);
VK = (1:full(max(max(Kantennr)))' + size(koordinat,1);
[koordinat,elemen,dirichlet,neumann] = ...
Verfeinerung(koordinat,elemen,dirichlet,neumann,Kantennr,VK)
;
end
FreeNodes=setdiff(1:size(koordinat,1),unique(dirichlet));
A = sparse(size(koordinat,1),size(koordinat,1));
b = sparse(size(koordinat,1),1);
% 1. menyusun matriks A1
% dengan memanggil fungsi stima_1
for i = 1:size(elemen,1)
```

```
A(elemen(i,:),elemen(i,:)) = A(elemen(i,:),elemen(i,:)) ...
+ stima(koordinat(elemen(i,:),:));
end
% 2. menyusun matriks A2
for i = 1:size(elemen,1)
A(elemen(i,:),elemen(i,:)) = A(elemen(i,:),elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
(diag(ones(3,1))+
ones(3))*k3(sum(koordinat(elemen(i,:),:))/3)/24;
end
% 3. volume forces
for i = 1:size(elemen,1)
b(elemen(i,:)) = b(elemen(i,:))+...
det([ones(1,3);koordinat(elemen(i,:),:)'])*...
g(sum(koordinat(elemen(i,:))/3))/6;
end
% 4. Neumann condition
for i = 1:size(neumann,1)
b(neumann(i,:))=b(neumann(i,:)) +...
norm(koordinat(neumann(i,1),:)- ...
koordinat(neumann(i,2),:)) *...
q1(sum(koordinat(neumann(i,:),:))/2)/2;
end
% 5. Dirichlet conditions
u = sparse(size(koordinat,1),1);
u(unique(dirichlet)) = ud3(koordinat(unique(dirichlet),:));
b = b - A*u;
% 6. Computation of the solution
u(FreeNodes) = A(FreeNodes,FreeNodes) \ b(FreeNodes);
% 7. Graphic representation
show(elemen,[],koordinat,full(u));
xlabel('x');ylabel('y');zlabel('u');
```

SIMPULAN

Kode Matlab berhubungan langsung dengan operasi FEM. Ditunjukkan bagaimana FEM diperkenalkan kepada mahasiswa hanya menggunakan beberapa baris kode Matlab. Kedua contoh yang diberikan dapat dijalankan pada PC standar. Hal ini juga menunjukkan bagaimana konsep-konsep yang diperoleh dapat dengan mudah diperluas ke masalah lain yang diberikan bersifat 2D. Namun, teknik pemrograman yang sama dapat digunakan untuk masalah yang lebih kompleks, seperti masalah 3D yang misalnya disajikan oleh Silvester dan Ferrari [6]. Lembaga pendidikan dengan sumber daya keuangan yang terbatas dapat mengambil manfaat dari metodologi yang diberikan, karena hanya memerlukan Matlab, tidak ada add-ons (bahkan pembangkit mesh sekalipun). Materi yang

disajikan dalam makalah ini mungkin juga berguna untuk para pemula penelitian yang belum mengenal FEM.

Ucapan Terima Kasih. Pekerjaan in didanai oleh Kementrian Pendidikan dan Kebudayaan, Republik Indonesia, melalui Penelitian Strategis Unggulan, hibah DIPA-IPB, 0558/023-04.2.01/12/2012, dengan kontrak no: **44/I3.24.4/SPK-PUS/IPB/2012**.

DAFTAR PUSTAKA

[1] J. Lu and D. V. Thiel, Computational and visual electromagnetics using an integrated programming language for undergraduate engineering students, IEEE Trans. Magnetics, 36 (2000), 10001003.
[2] S. Selleri, A Matlab experimental framework for electromagnetic education, IEEE Antennas and Propagat. Mag., 45 (2003), 8590.
[3] S. Selleri, A Matlab application programmer interface for educational electromagnetics in Antennas and Propagat. Soc. Symp, 2227 June, 2003.
[4] S. Makarov, MoM antenna simulations with Matlab RWG basis functions, IEEE Antennas and Propagat. Mag., 43 (2001), 100107.
[5] MATLAB, The Math Works Inc., http://www.mathworks.com
[6] P. P. Silvester and R. L. Ferrari, Finite Elements for Electrical Engineers (Cambridge University Press, Cambridge, 1990).
[7] J. Jin, The Finite Element Method in Electromagnetics (John Wiley and Sons, New York, 1993).
[8] S. R. H. Hoole, Computer-Aided Design of Electromagnetic Devices (Elsevier, New York, 1989).
[9] T. Itoh, G. Pelosi and P. P. Silvester, Finite Element Software for Microwave Engineering (John Wiley and Sons, New York, 1996).
[10] A. Bondeson, Computational Electromagnetics. (Springer. New York, 2005.)
[11] K. Foster, Retaking the _eld an old computational favourite is overhauled, IEEE Spectrum, July (2004), 5455.
[12] L. Y. Tio, A. A. P. Gibson, B. M. Dillon and L. E. Davis, Weak form finite element formulation for Helmholtz equation, Int. J. Elect. Enging. Educ., 41 (2004), 19.
[13] S. Ramo, J. R. Whinnery and T. Van Duzer, Fields and Waves in Communication Electronics (John Wiley, New York, 1994).
[14] A. K. Ghatak and K. Thyagarajan, Optical Electronics (Cambridge University Press, Cambridge, 1989).
[15] S. Hopfer, Design of ridged waveguides, IRE Trans. Microw. Theory. Tech., 3 (1955), 2029.
[16] P. Silvester, A general high-order _nite element waveguide analysis program, IEEE Trans. Microw. Theory Tech., 17 (1969), 204210.