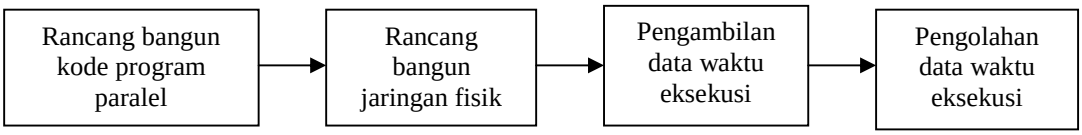
BAB III
METODOLOGI PENELITIAN

Simulasi pemrograman paralel pada medan elektromagnetik berdimensi satu dengan metode *Finite Difference Time Domain* (FDTD) dilakukan untuk menampilkan secara grafis medan listrik dan medan magnet dari suatu medan elektromagnetik serta dianalisis kecepatan waktu eksekusinya. Penelitian dimulai dengan perancangan flowchart simulasi. Selanjutnya, pembuatan kode program dengan menggunakan pustaka *message passing interface* pada rutin komunikasi *point to point* khususnya mode *non-blocking*. Metode penelitian yang digunakan adalah metode eksperimen, dimana kode-kode program yang telah dibuat dalam bahasa pemrograman C dieksekusi pada jaringan komputer fisik yang terdiri dari satu komputer *master* dan satu komputer *slave*. Tahapan-tahapan penelitian yang digunakan dapat digambarkan dalam diagram blok seperti terlihat pada Gambar 3.1.



Gambar 3.1 Diagram blok tahapan penelitian

Diagram blok dibuat untuk menampilkan langkah-langkah yang dilakukan dalam melakukan penelitian ini. Tahapan pertama adalah perancangan *flowchart* dari program simulasi yang dibuat. Kemudian *flowchart* tersebut diterjemahkan ke dalam kode-kode program bahasa C. Tahapan kedua adalah rancang bangun jaringan fisik (*cluster*) yang terdiri dari satu unit komputer *master* dan satu unit komputer *slave* yang dihubungkan secara langsung menggunakan kabel. Tahapan ketiga adalah pengambilan data dengan cara mengeksekusi kode program yang dibuat pada jaringan fisik yang telah dibangun sebelumnya, hal ini dilakukan untuk memperoleh nilai-nilai medan listrik dan medan magnet serta waktu eksekusinya. Tahapan keempat adalah pengolahan data dengan cara menghitung nilai *speedup*, yaitu perbandingan nilai waktu eksekusi program serial dan waktu eksekusi program paralel yang diperoleh dari pengambilan data yang dilakukan pada tahapan sebelumnya.

3.1 Rancang bangun kode program paralel

Rancang bangun kode program paralel dimulai dengan perancangan *flowchart* Simulasi. *Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu, sedangkan antara proses digambarkan dengan garis penghubung. Setelah perancangan *flowchart*, selanjutnya adalah penulisan kode program untuk masing-masing *flowchart*. Pada penelitian ini dibuat lima *flowchart* pemrograman paralel pada rutin komunikasi *point to point* khususnya operasi *non-blocking* yang terdiri dari:

1. Program simulasi medan elektromagnetik berdimensi satu pada ruang terbuka
2. Program simulasi medan elektromagnetik berdimensi satu pada ruang terbuka dengan penambahan kondisi batas serap
3. Program simulasi medan elektromagnetik berdimensi satu saat pulsa menumbuk medium dielektrik
4. Program simulasi medan elektromagnetik berdimensi satu saat gelombang sinusoidal menumbuk medium dielektrik
5. Program simulasi medan elektromagnetik berdimensi satu pada medium dielektrik *lossy*.

3.1.1 Simulasi medan elektromagnetik berdimensi satu pada ruang terbuka

Simulasi medan elektromagnetik berdimensi satu pada ruang terbuka dilakukan untuk mengetahui nilai medan listrik dan medan magnet pada suatu *time step*, serta dihitung waktu eksekusinya.

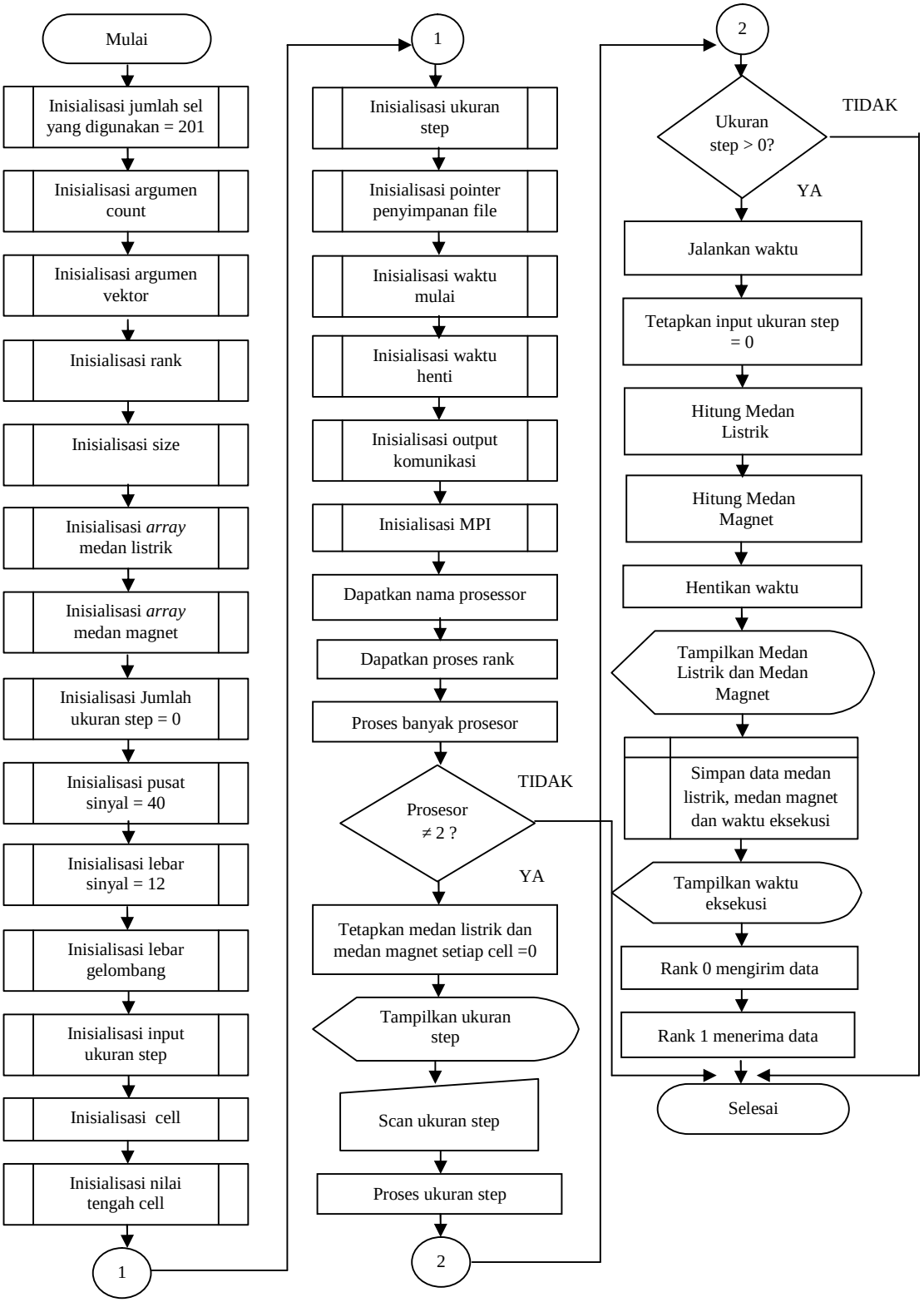
Pada simulasi ini, disisipkan pulsa *gaussian* di antara proses kalkulasi medan listrik dan medan magnet. Pulsa yang disisipkan didefinisikan sebagai:

$$pulse = e^{-5 \times \left(\frac{t0-T}{spread}\right)^2} \tag{3.1}$$

Jika ditulis ke dalam bahasa C, Persamaan (3.1) menjadi:

```
.........
pulse = exp(-.5*pow(t0-T)/spread,2.0) ));
.........
```

Flowchart simulasi medan elektromagnetik berdimensi satu pada ruang terbuka dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Flowchart* simulasi medan elektromagnetik berdimensi satu pada ruang terbuka

Program dimulai dengan inisialisasi data-data yang diperlukan dalam simulasi, mulai dari inisialisasi jumlah sel yang digunakan hingga inisialisasi MPI. Kode program inisialisasi secara berturut-turut yaitu:

```
.....
# define KE 201
int main ()
.....
double ex[KE];
double hy[KE];
.....
FILE *ifp;
char inputFilename[]=”ex.doc”;
FILE *ofp;
char outputFilename[]=”hy.doc”;
clock_t start, end;
MPI_Request request;
MPI_Init (&argc,&argv);
.....
.....
```

Langkah selanjutnya adalah mendapatkan nama prosesor, proses *rank* serta memproses banyak prosesor yang digunakan. Kode program yang digunakan yaitu:

```
.....
MPI_Comm_size(MPI_COMM_WORLD,&
size);
MPI_Comm_rank(MPI_COMM_WORL,&r
ank);
        If (size!=2)
        { printf(“minimal proses/prosesor
2\r\n”):}
……
```

Salah satu proses penting dari program ini adalah menjalankan proses ukuran *step*. kode program yang digunakan untuk proses menampilkan dan men-*scan* ukuran *step* secara berturut-turut yaitu:

```
.....
if (rank ==  0)
{
Printf (“NSTEPS --- > “);
fflush(stdout);
Scanf (“%d”,&NSTEPS);
Printf  (“%d\n”, NSTEPS);
}
.....
```

Setelah proses NSTEPS selesai, maka langkah selanjutnya adalah menjalankan waktu eksekusi, menghitung nilai medan listrik dan nilai medan magnet. Kode program yang digunakan untuk langkah ini yaitu:

```
.....
While (NSTEPS > 0) {
start = MPI_Wtime(); /*mulai waktu eksekusi*/
.....
{ex[k] = ex[k] + .5*(hy[k-1]-hy[k]); /*hitung medan listrik*/
.....
.....
{ hy[k] = hy[k] +.5*(ex[k] – ex[k+1]); } /*hitung medan magnet*/
.....
```

Kode program di atas yang digunakan untuk perhitungan medan listrik dan medan magnet merupakan Persamaan 2.13a dan Persamaan 2.13b yang ditulis ke dalam bahasa pemrograman C.

Setelah nilai medan listrik dan medan magnet dihitung, langkah selanjutnya adalah menghentikan waktu eksekusi. Kode program yang digunakan yaitu:

```
.....
end = MPI_Wtime();
.....
```

Setelah waktu eksekusi dihentikan, langkah selanjutnya adalah menampilkan nilai medan listrik dan medan magnet. Kode program yang digunakan untuk langkah ini yaitu:

```
.....
Printf (“3%d %f  %f\n”, k, ex[k],hy[k]);
.....
```

Setelah hasil simulasi ditampilkan, data nilai medan listrik, nilai medan magnet dan waktu eksekusi disimpan dalam bentuk *file* berekstensi doc. Kode program yang digunakan yaitu:

```
.....
/*menyimpan nilai medan listrik*/
ifp=fopen(inputFilename,”w”);
.....
fprintf(ifp,”k\t ex[k]\n”);
.....
.....
fprintf(ifp, “Waktu eksekusi = %f detik.\n”,end-start);
fclose(ifp);
.....
```

```
/*menyimpan nilai medan magnet*/
ofp=fopen(outputFilename,"w");

fprintf(ofp,"k\t hy[k]\n");
.....
.....
fprintf(ofp, "Waktu eksekusi = %f detik.\n",end-start);
fclose(ofp);
/*menampilkan waktu di jendela eksekusi*/
printf ("T = %5.0f\n",T);
fprintf("Waktu eksekusi = %f detik.\n",end-start);
.....
```

Setelah semua proses selesai, maka dimulai pemrosesan paralel dengan cara *rank* 0 mengirim data ke *rank* 1. Sebaliknya *rank* 1 menerima data dari *rank* 0. Kode program yang digunakan untuk proses ini berturut-turut yaitu:

```
.....
/* rank 0 mengirim data*/
if (rank == 0)
{
        MPI_Isend (&NSTEPS, 200, MPI_INT, 0, 1, MPI_COMM_WORLD,
&request);
        printf ("Node %d telah mengirim data ke rank %d\r\n", rank, 1);
}
/* rank 1 menerima data*/
else if (rank == 1)
{
        MPI_Irecv (&NSTEPS, 200, MPI_INT, 1, 1, MPI_COMM_WORLD,
&request);
                printf ("Node %d telah menerima data ke rank %d\r\n", rank, 0);
}
Break;
}
MPI_Finalize ();
return 0;
}
```

Data yang dikirim adalah ukuran *step* sebanyak 200 langkah dengan tipe data *integer* (MPI_INT), *Source* pengiriman data adalah 0 (*rank* 0), *Source* penerimaan data adalah 1 (*rank* 1), nilai *Message tag* adalah 1, dan nama *output* komunikasi adalah *request*.

3.1.2 Simulasi medan elektromagnetik berdimensi satu pada ruang terbuka dengan penambahan kondisi batas serap

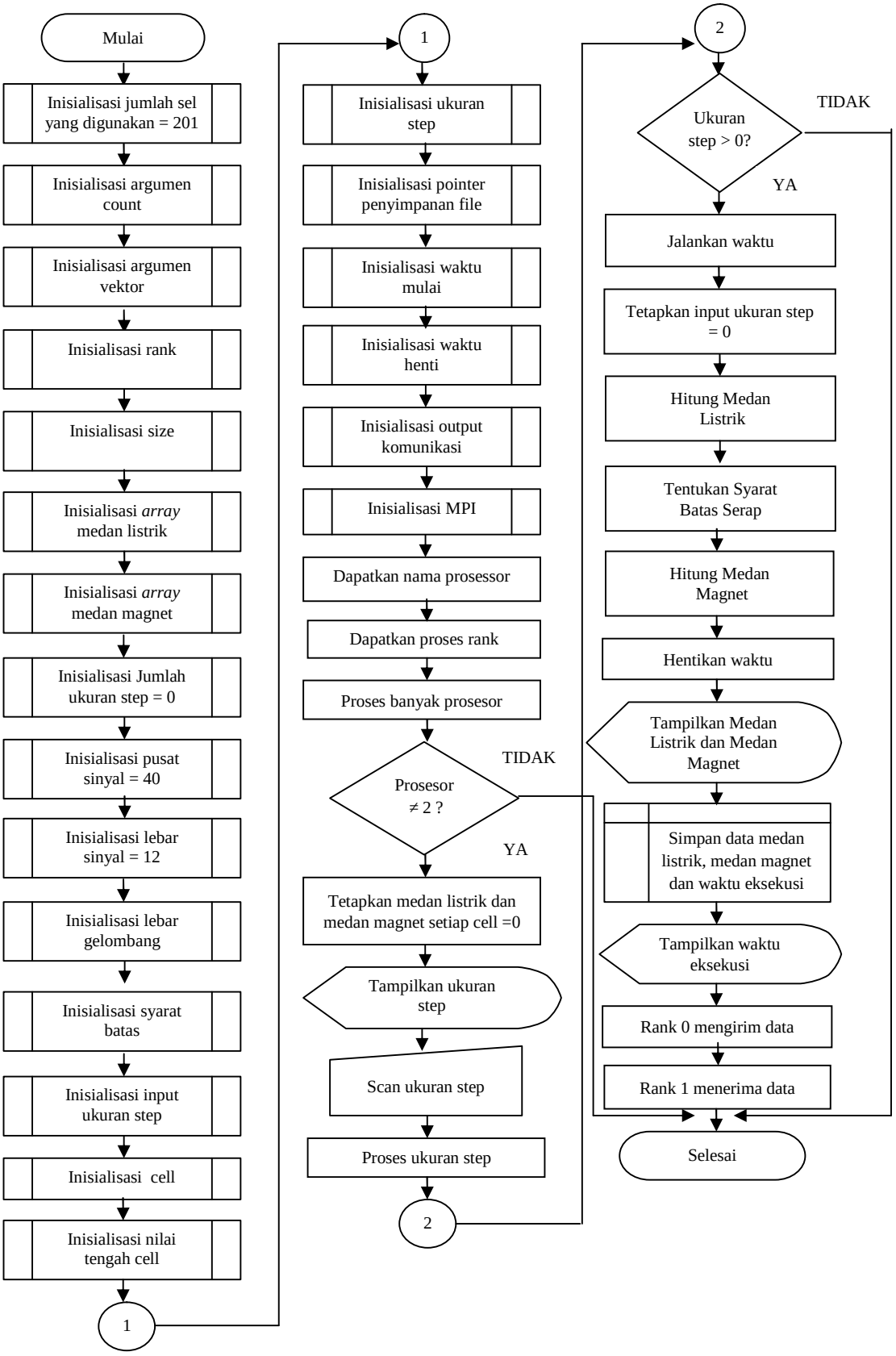
Sama halnya dengan simulasi pertama (Subbab 3.1.1), simulasi ini dilakukan untuk mengetahui nilai medan listrik dan medan magnet pada suatu *time step*, serta dihitung waktu eksekusinya. Akan tetapi, pada simulasi ini ditambahkan kondisi batas serap.

Kode program untuk proses penentuan syarat batas serap yaitu:

```
.................
ex[0]           =       ex_low_m2;
ex_low_m2       =       ex_low_m1;
ex_low_m1       =       ex[1];

ex[KE-1}        =       ex_high_m2;
ex_high_m2      =       ex_high_m1;
ex_high_m1      =       ex[KE-2];
.................
```

Kode program di atas dibuat berdasarkan persamaan syarat batas yaitu Persamaan 2.15. Pada simulasi ini, disisipkan pulsa *gaussian* dengan bentuk sinyal digital diantara proses kalkulasi medan listrik dan medan magnet dengan menggunakan Persamaan 3.1.

Flowchart simulasi medan elektromagnetik berdimensi satu pada ruang terbuka dengan penambahan kondisi batas serap dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Flowchart* simulasi medan elektromagnetik berdimensi satu pada ruang terbuka dengan penambahan kondisi batas serap

3.1.3 Simulasi medan eletromagnetik berdimensi satu saat pulsa menumbuk medium dielektrik

Simulasi ini dilakukan untuk mengetahui nilai medan listrik dan medan magnet pada *time step* tertentu saat pulsa menumbuk medium dielektrik, serta dihitung waktu eksekusinya.

Kode program untuk proses menampilkan konstanta dielektrik relatif, men-*scan* konstanta dielektrik relatif, menampilkan epsilon, dan men-*scan* epsilon secara berturut-turut adalah sebagai berikut :

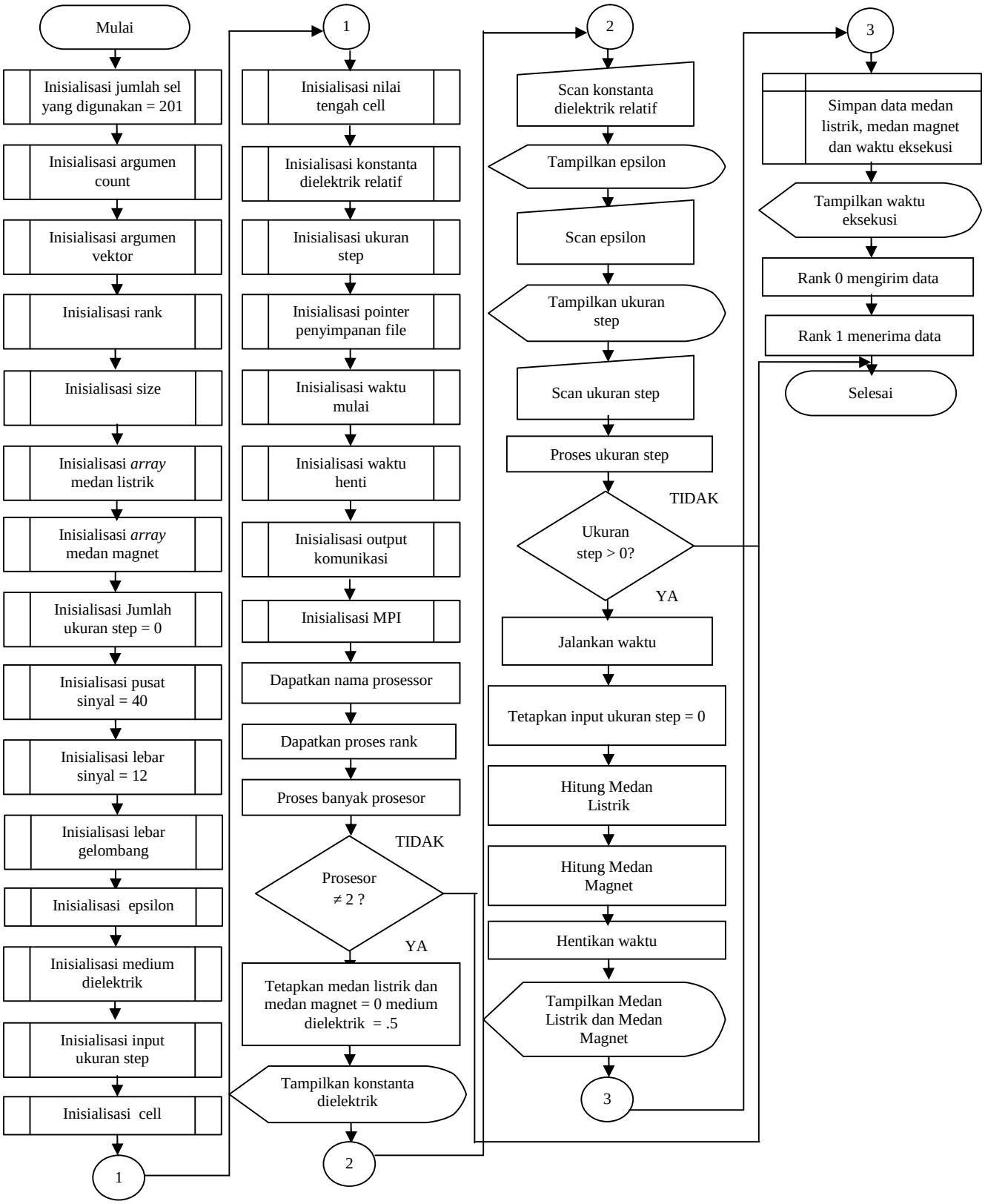
```
.....
{
printf (“dielectric start at --> “);
fflush(stdout);
scanf (“%d”,&kstart);
printf (“Epsilon --> “);
fflush(stdout);
scanf (“%f”,&epsilon);
for (k=kstart; k <= KE; k++) {
cb[k]= .5/epsilon;
}
.....
```

Kode program untuk perhitungan medan listrik adalah sebagai berikut:

```
.....
/* calculate the Ex field*/
.....
{ ex[k] = ex[k] + cb[k] * (hy[k-1] – hy[k]); }
.....
```

Kode program untuk perhitungan medan listrik di atas berdasarkan Persamaan 2.20a yang ditulis ke dalam bahasa pemrograman C. Pada simulasi ini, disisipkan pulsa *gaussian* dengan bentuk sinyal digital diantara proses kalkulasi medan listrik dan medan magnet dengan menggunakan Persamaan 3.1.

Flowchart simulasi medan elektromagnetik berdimensi satu pada saat pulsa menumbuk medium dielektrik dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Flowchart* simulasi medan eletromagnetik berdimensi satu saat pulsa menumbuk medium dielektrik

3.1.4 Simulasi medan eletromagnetik berdimensi satu saat gelombang sinusoidal menumbuk medium dielektrik

Simulasi ini dilakukan untuk menghitung nilai medan listrik dan medan magnet pada *time step* tertentu saat gelombang sinusoidal menumbuk medium dielektrik, serta dihitung waktu eksekusinya.

Pada program simulasi sebelumnya (Subbab 3.1.3), ditambahkan kode program proses menampilkan konstanta dielektrik relatif, men-*scan* konstanta dielektrik relatif, menampilkan epsilon, men-*scan* epsilon secara berturut-turut. Pada program simulasi kali ini, terdapat penambahan satu inisialisasi yaitu frekuensi gelombang (*freq_in*). Kode program untuk proses menampilkan dan men-*scan* frekuensi yaitu:

```
.............
/* these parameters specify the input pulse*/
printf ("Input freq (MHz) --> ");
fflush(stdout);
scanf  ("%f",&freq_in);
printf  ("%f \n", freq_in);
.............
```

Kode program untuk perhitungan medan listrik yaitu:

```
.............
{ ex[k] = hy[k] + cb[k] * ( hy[k-1] – hy[k] );
.............
```

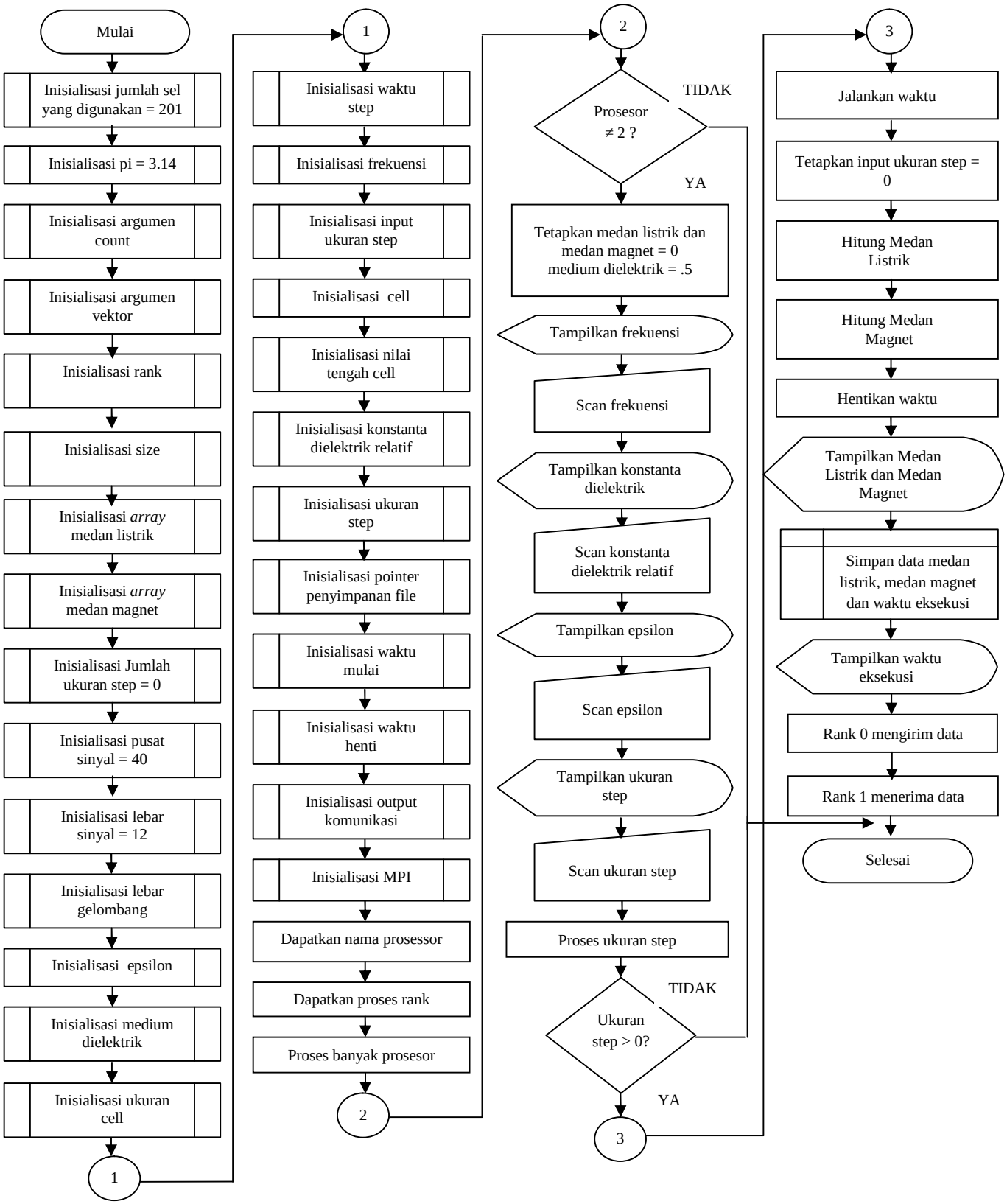
Seperti halnya kode program untuk perhitungan medan listrik yang digunakan pada Subbab 3.1.3, perhitungan medan listrik pada subbab ini juga berdasarkan Persamaan 2.20a yang telah ditulis ke dalam bahasa pemrograman C. *Flowchart* simulasi medan elektromagnetik berdimensi satu saat gelombang sinusoidal menumbuk medium dielektrik dapat dilihat pada Gambar 3.5.

Pada simulasi ini, disisipkan pulsa sinusoidal di antara proses kalkulasi medan listrik dan medan magnet. Pulsa yang disisipkan didefinisikan sebagai:

$$pulse = \sin(2\pi f T\, dt) \tag{3.2}$$

Jika ditulis ke dalam bahasa C, Persamaan 3.2 menjadi:

```
.........
pulse = exp(2*pi*freq_in*dt*T);
.........
```



Gambar 3.5 *Flowchart* simulasi medan eletromagnetik berdimensi satu saat gelombang sinusoidal menumbuk medium dielektrik

3.1.5 Simulasi medan elektromagnetik berdimensi satu pada medium dielektrik *lossy*

Simulasi ini dilakukan untuk menghitung nilai medan listrik dan medan magnet pada medium yang memiliki nilai konduktivitas rendah, serta dihitung waktu eksekusinya.

Kode program untuk proses menampilkan dan men-*scan* konduktivitas sebagai berikut:

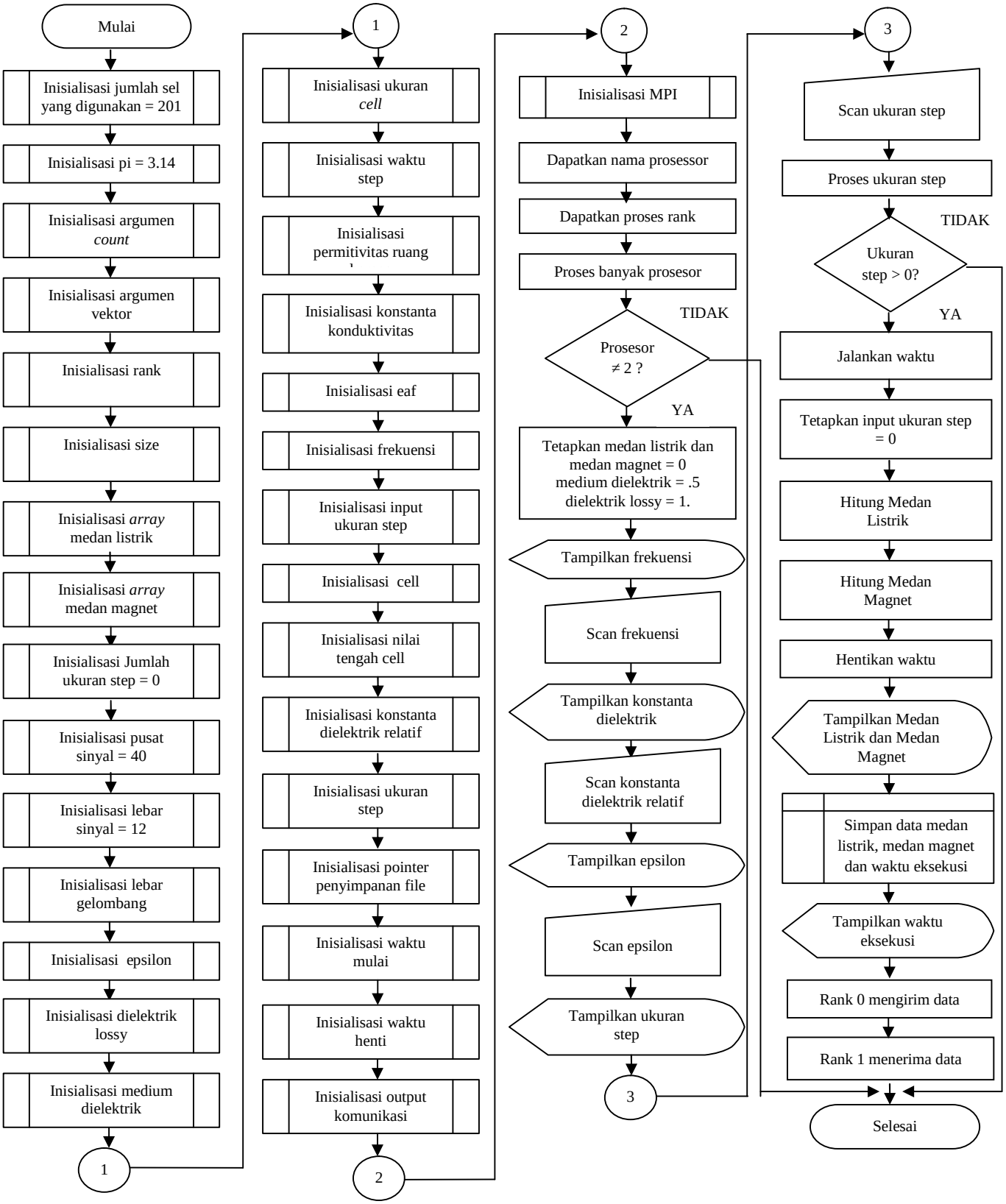
```
.....................
.....................
printf (“Conductivity -->  “);
fflush(stdout);
scanf (“%f”, &sigma);
printf (“%d %f %f\n”, kstart,epsilon, sigma);

eaf = dt*sigma/(2*epsz*epsilon);
printf (“%f\n”,eaf);
for (k=kstart; k<=KE; k++)
{
ca[k] = (1. –eaf) / (1 + eaf);
cb[k] = .5 / (epsilon*(1 + eaf);
}
.....................
```

Kode program untuk perhitungan nilai medan listrik yaitu:

```
.....................
{ ex[k] = ca[k]*ex[k] + cb[k] * (hy[k-1] – hy[k]) ;}
.....................
```

Kode program yang digunakan untuk perhitungan medan listrik menggunakan Persamaan 2.24 yang ditulis ke dalam bahasa pemrograman C. Pada simulasi ini, disisipkan pulsa *gaussian* dengan bentuk sinyal analog diantara proses menghitung medan listrik dan medan magnet dengan menggunakan Persamaan 3.2.

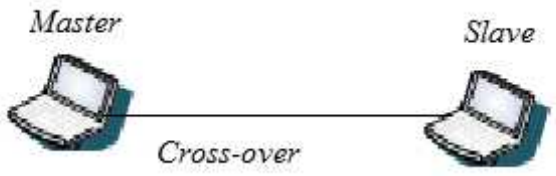
Flowchart simulasi medan elektromagnetik berdimensi satu pada medium dielektrik *lossy* dapat dilihat pada Gambar 3.6.



Gambar 3.6 *Flowchart* simulasi medan elektromagnetik berdimensi satu pada medium dielektrik *lossy*

3.2 Rancang bangun jaringan fisik

Dalam penelitian ini dihubungkan satu unit komputer *master* dan satu unit komputer *slave* dalam sebuah *cluster* seperti terlihat pada Gambar 3.7.



Gambar 3.7 Jaringan fisik komputer

Master adalah komputer yang digunakan sebagai tempat mengakses dan memberikan tugas ke *slave. Master* digunakan sebagai tempat bagi pengguna *cluster* untuk mengakses *cluster* sehingga pengguna dapat memberikan *tasks* komputasi ke dalam *cluster*.

Slave adalah komputer pekerja yang menerima tugas dari *master* dan memproses tugas tersebut. Komputer yang difungsikan sebagai *slave* hanya dapat diakses oleh *master* untuk melakukan proses komputasi.

Master dan *Slave* dihubungkan secara langsung menggunakan kabel *Local Area Network* (LAN) tipe *Cross-over* dengan topologi *point to point*. Perangkat lunak yang digunakan pada masing-masing komputer adalah *Virtualbox* versi 4.1.4. Adapun spesifikasi komputer *master* dan *slave* seperti terlihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi komputer *master* dan *slave*

Perangkat	***Master***	***Slave***
RAM (*Random Access Memory*)	512 *megabyte*	384 *megabyte*
Prosessor	CPU Intel *Core 2 Duo*. @2.20 GHz	CPU Intel *Core 2 Duo*. @2.00 GHz
Sistem Operasi	*Pelican*HPC versi 2.2 (32 bit)	*Pelican*HPC versi 2.2 (32 bit)

Kecepatan masing-masing komputer dalam *cluster* dapat dinyatakan dalam *teraflops* seperti terlihat pada Gambar 3.8.

Gambar 3.8 kecepatan cluster

Gambar 3.8 menunjukkan jumlah prosesor yang digunakan adalah 2 prosesor,waktu kalkulasi *cluster* adalah 1,96 detik, kecepatan *cluster* adalah 918 MFLOPS, kecepatan *master* adalah 459 MFLOPS, dan kecepatan *slave* adalah 764 MFLOPS.

3.3 Pengambilan data waktu eksekusi

Dalam penelitian ini kompilasi dan eksekusi kode program serial dan paralel dilakukan pada terminal sistem operasi *Linux PelicanHPC.* Perintah kompilasi dan eksekusi pada kode serial dan kode paralel seperti terlihat pada Tabel 3.2.

Tabel 3.2 Perintah kompilasi dan eksekusi kode serial dan kode paralel

	Kode serial	**Kode paralel**
Kompilasi	*gcc test.c -o test –lm*	*mpicc test.c -o test*
Eksekusi	*./test*	*mpiexec -n 2 test*

Tahapan kompilasi kode serial menggunakan tambahan *–lm* (*library of math*) pada akhir baris kompilasi. *–lm* berfungsi sebagai penghubung ke *header* matematika (*math.h*) saat kode program dikompilasi pada terminal *Linux* dengan *compiler* GCC (*GNU C Compiler*) versi 4.3.2. Tahapan selanjutnya adalah kompilasi kode paralel dengan menggunakan *compiler* MPI untuk bahasa C (*mpicc*) versi 1.4.2.

Dalam penelitian ini, ada lima kode serial (Lampiran B) dan lima kode paralel (Lampiran C) dalam bahasa C yang dikompilasi dan dieksekusi. Setelah program berhasil dieksekusi, maka didapatkan nilai medan listrik dan medan magnet, serta waktu eksekusinya (Lampiran D). Langkah selanjutnya adalah menggambarkan secara grafis nilai medan listrik, medan magnet serta waktu eksekusi masing-masing program. Waktu eksekusi yang digunakan merupakan waktu eksekusi dari 10 kali percobaan.

3.4 Pengolahan data waktu eksekusi

Dalam penelitian ini, pengolahan data dilakukan dengan cara menghitung rata-rata waktu eksekusi yang diperoleh dari data 10 kali percobaan terhadap setiap kode program yang dilakukan pada tahapan sebelumnya. Langkah selanjutnya adalah melakukan perbandingan rata-rata waktu eksekusi program serial dan rata-rata waktu eksekusi program paralel untuk memperoleh berapa banyak peningkatan kecepatan (*speedup*) yang diperoleh. Perhitungan nilai *speedup* didasarkan pada Persamaan 2.25