Menjalankan Simulasi Jaringan Sensor Nirkabel pada NS-2

(sumber: http://sourceforge.net/projects/wsnlocalisation/files/ oleh Adnan Abu-Mahfouz)

A. Instalasi NS-2 pada Ubuntu 10.04

1. Pada proses instalasi NS2 yang diperlukan operating sistem Ubuntu 10.04

2. Setelah proses instalasi Ubuntu berhasil selanjutnya dapat dilakukan installasi NS2. Sebelumnya download source NS all in one versi 2.34 di http://sourceforge.net/projects/nsnam/file/.
3. Untuk melakukan installasi NS maka dilakukan update dan upgrade terlebih dahulu.
   ```
   #apt-get update
   #apt-get upgrade
   ```
4. Lakukan install compiler pendukung NS2
   ```
   #sudo apt-get install build-essential autoconf automake libxmu-dev gcc-4.3
   ```
5. Buka file ns-allinone-2.34/otcl-1.13/Makefile.in.Cari baris yang menyatakan
   ```
   CC= @CC@ ubah menjadi CC= gcc-4.3
   ```
6. Sebelum melakukan instalasi network simulatornya, ekstrak file NS terlebih dahulu
   ```
   #tar xvfz ./ns-allinone-2.34.tar.gz
   ```

7. Ubah direktori dengan perintah
   ```
   $ cd /home/ninis$ cd /home/ninis/ns-allinone-2.34
   ```
8. Installasi beberapa paket dari repositori jika diperlukan(optional)
   ```
   $ sudo apt-get install build-essential autoconf automake libxmu-dev
   ```
9. Install the ns2
   ```
   $ cd ns-allinone-2.34
   $ ./install
   ```
10. Edit beberapa bagian
    ```
    $ gedit ~/.bashrc
    ```
    Letakkan baris ini pada file di terakhir mengubah /home/ninis/ tergantung di mana file ns-allinone-2.34.tar di ekstrak.

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/ninis/ns-allinone-2.34/otcl-
1.13 NS2_LIB=/home/ninis/ns-allinone-2.34/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS
2_LIB:$X11_LIB:$USR_LOCAL_LIB

# TCL_LIBRARY
TCL_LIB=/home/ninis/ns-allinone-
2.34/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
XGRAPH=/home/ninis/ns-allinone-
2.34/bin:/home/ninis/ns-allinone-
2.34/tcl8.4.18/unix:/home/ninis/ns-allinone-
2.34/tk8.4.18/unix
#the above two lines beginning from xgraph and
ending with unix should come on the same line
NS=/home/micman/ns-allinone-2.34/ns-2.34/
NAM=/home/micman/ns-allinone-2.34/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

11. Selanjutnya lakukan validasi
    ./validate
12. Jika saat diketikkan ns dan meghasilkan % maka proses instalsi NS berhasil

```
root@ninis:~# ns
% ns-version
2.34
```

B. Perancangan Simulasi Jaringan Sensor Nirkabel menggunakan Network Simulator NS 2-34

Untuk melakukan lokalisasi dengan menggunakan network simulator NS-2 ini, beberapa perlengkapan yang perlu dipersiapkan adalah

1. Laptop atau PC dengan OS Ubuntu versi 10.4 (Versi Ubuntu maksimal 10.10, belum support untuk Ubuntu dengan versi diatas versi 10.10)
2. Installer network simulator versi NS2-34 secara lengkap '*allinone'*, installasi NS-2 jangan melalui *package symnaptic*
3. File yang dipersiapkan untuk melakukan modifikasi untuk keperluan lokalisasi pada NS-2, dapat di download http://sourceforge.net/projects/wsnlocalisation/, dimana file dalam folder yang terdownload, meliputi :
   - Folder location
   - Folder untuk melakukan modifikasi pada folder common dan tcl
   - Contoh program tcl
   - Panduan instalasi network simulator
   - Panduan untuk melakukan modifikasi

C. Modifikasi file NS2-34

Selanjutnya untuk proses installasi lokalisasi pada jaringan sensor nirkabel menggunakan simulator jaringan ns2-34 ini, maka akan ditambahkan modul baru dan juga memodifikasi beberapa file yang sudah ada. Untuk keprluan modifikasi ns2-34 ini sebelumnya download file keperluan lokalisasinya pada alamat. Berikut adalah proses modifikasinya.

1. Copy-kan folder location dalam folder ns2-34
2. Untuk keperluan modifikasi network simulator selanjutnya copikan beberapa file yang ada dalam folder modified ke file ns2-34 (lib dan common) yang akan dimodifikasi, untuk file yang lama bisa dilakukan rename file atau di simpan pada folder tersendiri.

Tabel 1. Nama File Modifikasi NS-2

No	Directory	Modified Files
1	ns2-34/common/	packet.h
2	ns2-34/common/	mobilenode.h
3	ns2-34/common/	mobilenode.cc
4	ns2-34/common/	location.h
5	ns2-34/tcl/lib/	ns-packet.tcl
6	ns2-34/tcl/lib/	ns-default.tcl
7	ns2-34/tcl/lib/	ns-lib.tcl
8	ns2-34/tcl/lib/	ns-node.tcl
9	ns2-34/tcl/lib/	ns-namsupp.tcl

3. Selanjutnya buat makefilenya dengan membuka ns2-34/Makefile, lakukan penambahan pada bagian INCLUDES dan OBJ_STL.

```
INCLUDES=
                -I./location
OBJ_STL =
                  location/locationrequest.o \
                  location/locationresponse.o \
                  location/locationdiscovery.o \
                  location/mmse.o \
                  location/position.o \
                  location/nearestposition.o \
                  location/refineposition.o
```

4. Untuk mencoba hasil modiffikasi yang telah dilakukan, run contoh program .tcl yang sudah tersedia dalam folder ns2_localisation dengan mengetikkan perintah

```
# ns nama_file.tcl
```

Terdapat beberapa contoh program tcl yang terdapat dalam folder ns2_location, yaitu loc1, loc2, loc3, loc4, loc5, dan loc6.

5. Untuk mengeksekusi beberapa contoh program tcl ini, harus di ikuti dengan METHOD yang digunakan karena dalam program .tcl merupakan program untuk menggatur jumlah node yang digunakan saja belum termasuk METHOD yang digunakan dalam lokalisasi node sensor. Untuk file loc1.tcl, loc2.tcl, dan loc3.tcl Program dapat dijalankan dengan menambahkan metode pada sintaknya

   ```
   # ns loc1.tcl METHOD
   ```

   ```
   root@ninis:~# ns loc1.tcl 1
   num_nodes is set 4
   INITIALIZE THE LIST xListHead
   Starting Simulation...
   channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
   highestAntennaZ_ = 1.5,  distCST_ = 3511.9
   SORTING LISTS ...DONE!
   1.003183: d: 28.284271, Pr: 2.403478e-07
   1.003183: d: 28.284271, Pr: 2.403478e-07
   ```

   Untuk loc4.tcl, loc5.tcl dan loc6.tcl tidak hanya ditambah dengan metodenya namun juga ditambahkan SEED yang digunakan untuk membandingkan antara dua metode yang berbeda. Maksud dari penambahan SEED ini adalah agar distribusi penyebaran untuk membandingkan antara metode yang berbeda, tetap tidak berubah. Berikut adalah sintaknya

   ```
   #ns nama_file.tcl SEED METHODE
   ```

6. Untuk melihat hasil trace filenya dapat dilakukan perintah #location.tr

   result

   ```
   L 2.334495 15 1.998534 3 0.503827 24.336493 34.978063
   L 2.571295 2 1.998156 3 2.508023 44.517734 28.726999
   L 2.636915 12 1.998156 3 1.982105 0.879476 46.719686
   L 3.283809 6 1.997732 3 2.522068 16.946984 3.136199
   L 3.415675 10 1.997574 3 0.068594 30.673174 34.891277
   ```

D. Implementasi Modul Lokalisasi pada NS 2-34

Untuk melakukan implementasi lokalisasi Jaringan Sensor Nirkabel menggunakan algoritma centroid pada ns2-34 ini, maka perlu ditambahkan modul baru dan juga memodifikasi beberapa file yang sudah ada.

Tabel 2. Nama File Modifikasi NS-2

No	Directory	Modified Files
1	ns2-34/common/	packet.h
2	ns2-34/common/	mobilenode.h
3	ns2-34/common/	mobilenode.cc
4	ns2-34/common/	location.h
5	ns2-34/tcl/lib/	ns-packet.tcl
6	ns2-34/tcl/lib/	ns-default.tcl
7	ns2-34/tcl/lib/	ns-lib.tcl
8	ns2-34/tcl/lib/	ns-node.tcl
9	ns2-34/tcl/lib/	ns-namsupp.tcl

Berikut adalah tambahan program yang disisipkan pada beberapa file ns2-34 :

- ns/common/packet.h

```
Add:
#define HDR_LOCREQ(p) (hdr_locreq::access(p))
#define HDR_LOCRES(p) (hdr_locres::access(p))
```

Fungsi dari penambahan pada packet.h adalah untuk mendefinisikan dua tipe packet yang dibuat di locationpaket.h menggunakan 2 struktur (hdr_locreq dn hdr_locres). untuk menggunakan dua jenis paket ini, jenis paket yang sesuai telah didefinisikan dalam packet.h

- ns/tcl/lib/ns-packet.tcl

```
Before the end of foreach prot {} add
   LocReq #Location Request
   LocRes #Location Response
```

File ini digunakan untuk mengaktivasi header class baru

- ns/common/location.h

```
     Add a null constructor
               Location(): X(0), Y(0), Z(0) {}
     dd the getters and setters of individual
parameters
         virtual double getx() {return X;}
         virtual double gety() {return Y;}
         virtual double getz() {return Z;}
         virtual void setx(double x) {X = x;}

         virtual void sety(double y) {Y = y;}
         virtual void setz(double z) {Z = z;}
     Add the following function to check if
two locations are the same
               virtual int is_equal(Location
*loc)
               {
     if((X == loc->X) && (Y == loc->Y) && (Z
== loc->Z))
                           return 1;
                     else
                           return 0;
               }
     Add the following function to estimate
the distance between two locations
               virtual double
distance(Location *loc)
               {
               double dx = X - loc->X;
               double dy = Y - loc->Y;
               double dz = Z - loc->Z;
               return sqrt( dx * dx + dy * dy
+ dz * dz);
               }
```

File ini berisi location class yang digunakan untuk menunjukkan lokasi koordinat node (X,Y, dan Z).

- ns/common/mobilenode.cc..h

```
inline Topography* get_topography() {
return T_;}
  // log the location information
  void log_loc(double, int, double, double);

ns/common/mobilenode.cc
void
MobileNode::log_loc(double error_, int
no_ref_, double x, double y)
{
  if (!log_target_)
        return;
  Scheduler &s = Scheduler::instance();
  sprintf(log_target_->pt_->buffer(),"L %f
%d %f %d %f %f %f",
                      s.clock(),
                      address_,
                      energy_model_-
>energy(),
                      no_ref_,
                      error_,
                      x,
                      y);
  log_target_->pt_->dump();

}
```

Dua metode ditambahkan pada file ini. Pertama untuk mendapatkan sebuah object dari sebuah topologi. Kedua untuk merecord hasil pada trace file. Hasil dapat berupa location error, anchor node yang digunakan sebagai node referensi, akurasi probabilitas dan energy yang tersisa

- Ns/tcl/lib/ns-node.tcl

```
Within Node instproc init args, add the
following instvar
      #Location discovery
      nodeAttribute_

Add the following instproc
      #Location discovery
      Node instproc attribute {} {
      return [$self set nodeAttribute_]
      }
```

Berfungsi untuk menentukan jenis node . Dimana node dapat berupa beacon, unknown atau node referensi

- ns/tcl/lib/ns-namsupp.tcl

```
Modify Node instproc color as following
      #enable changing the node's color
after running the simulator by Ian Downard
      Node instproc color { color } {
            $self instvar attr_ id_
            set ns [Simulator instance]
            set attr_(COLOR) $color
            set attr_(LCOLOR) $color
            if [$ns is-started] {
            # color must be initialized
            $ns puts-nam-config \
            [eval list "n -t [$ns now] -s
$id_ -S COLOR -c $color -o $attr_(COLOR) -
i $color -I $attr_(LCOLOR)"]
            }
      }
```

Selama simulasi ketika unknown node diestimasi posisinya , mereka berubah warna. Agar node berubah warna ketika simulasi berjalan, “Node istproc color” telah dimodifikasi dalam file ini.

- ns/tcl/lib/ns-lib.tcl

```
Add to
# New node structure
# added here to specify node attribute
(beacon, reference or unknown)

#-attribute BEACON/REFERECE/UNKNOWN
Add the following instproc

#used for location discovery process
Simulator instproc attribute {val} {$self
set attribute_ $val}
Within Simulator instproc node-config
args, add the following instvar

#Location discovery
attribute_
Within Simulator instproc create-wireless-
node args, add the following instvar

#Location discovery
attribute_
Also add:

#location discovery
if [info exists attribute_]
   {
$node set nodeAttribute_ $attribute_
}
```

File ini berfungsi untuk menagani node attribute

- ns/tcl/lib/ns-default.tcl

```
Add the following to the end of the file:
   #Location discovery
   Agent/LocReq set packetSize_ 100
   Agent/LocRes set packetSize_ 100
   Application/LocDiscovery set
distanceError_ 0
   Application/LocDiscovery set reqFreq_ 5.0
         ;# every 5.0 sec
   Application/LocDiscovery set maxRequests_
268435456      ;# 0x10000000
   Application/LocDiscovery set showColor_ 1
   ;# Node colouring, enable(1) or disable(0)
   ;# Localisation algorithm, general method
(1), nearest3(2), refinement(3)
       Application/LocDiscovery set method_

   Simulator set attribute_ ""
```

Nilai default dari variable terikat diinisialisasi dalam file ns-default.tcl

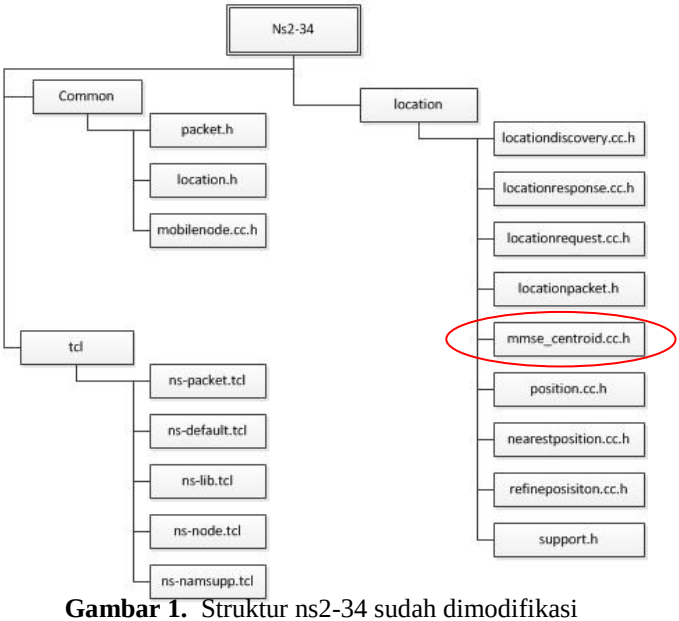
Menambahkan Folder Lokalisasi

Pada simulasi menggunkan NS-2.34 ini mnggunakan modul yang sudah dibuat oleh Adnan Abu Mahfouz. Dimana modul yang telah dibuat menggunakan algoritma trilateral dan multilateral sehingga perlu melakukan penambahan file dalam folder localisation untuk memasukkan algoritma centroid. Ada beberapa file terdapat pada folder localisation adalah sebagai berikut :

Tabel 3. File tambahan ns2-34

No	Nama File	Keterangan
1	Locationdiscovery.h	Header file Locationdiscovery.cc
2	Locationdiscovery.cc	Pilihan untuk metode yang digunakan menggunakan position, refineposition atau nearest posistion
3	Locationpacket.h	Header file locationpacket
4	Locationrequest.h	Header file locationrequest.cc

5	Locationrequest.cc	Mendefinisikan LocreqAgent class
6	Locationresponse.h	Header file locationresponse.cc
7	Locationresponse.cc	Bertanggung jawab untuk menerima paket dari class agent
8	Mmse.h	Header file mmse.cc
9	Mmse.cc	Operasi perhitungan posisi dengan algoritma trilateral dan multilateral
10	Nearestposition.h	Header file nearestposition.cc
11	Nearestposition.cc	File berisi perhitungan untuk estimasi posisi unknown node dengan 3 reference node
12	Position.h	Header file position.cc
13	Position.cc	Perhitungan estimasi posisi dengan memanggil file mmse.cc
14	Refineposition.h	Header dari refineposition.cc
15	Refineposition.cc	Perbaikan estimasi posisi
16	Support.h	Berisi tentang



Gambar 1. Struktur ns2-34 sudah dimodifikasi

Membuat file baru untuk simulasi lokalisasi JSN pada network simulator ns2-34 membuat file dengan nama mmse_centroid . dan mmse_centroid.cc

- mmse_centroid.h

Membuat header dari file mmse_centroid.cc

```
#ifndef ns_mmse_centroid_h
#define ns_mmse_centroid_h
#include "locationdiscovery.h"
#include "location.h"

struct ReferenceNode;

class MMSE_CENTROID
{

public:
       MMSE_CENTROID() : num_ref_(0)
       ~MMSE_CENTROID();

       virtual bool estimate(ReferenceNode
       *, int, Location*);

protected:
       virtual void init(ReferenceNode *);
       double *x;
       double *y;

       // Number of references that will be
       used in the estimation
       int num_ref_;
};

#endif
```

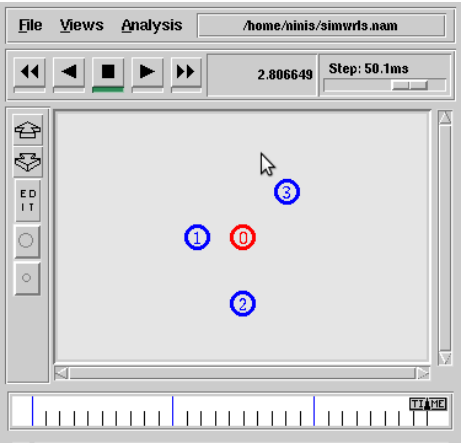
- mmse_centroid.cc

Program untuk perhitungan estimasi posisi berada di file mmse_centroid.cc

```
#include "mmse_centroid.h"
// Distructor
MMSE_CENTROID::~MMSE_CENTROID()
{
	delete [] x;
	delete [] y;
}
bool MMSE_CENTROID::estimate(ReferenceNode
*ref_nodes_, int n, Location* loc_)
{
	double sumx, sumy;
	num_ref_ = n;
	init(ref_nodes_);
	double x, y;
	unsigned int i
	x =sumx/num_ref_;
	y =sumy/num_ref_;
	loc_->setx(x);
	loc_->sety(y);
	return SUCCESS;
}
void MMSE_CENTROID::init(ReferenceNode
*ref_nodes_)
{
	double x0, y0, d0;
	double sumx=0, sumy=0;
	unsigned int i;
	double max_range=MAX_RANGE;
	ReferenceNode Ref;
if ( ref_nodes_[i].distance_ < max_range)
{
	for (i = 0; i < num_ref_ - 1; i++)
	{
	x0 = ref_nodes_[i + 1].loc_.getx();
	y0 = ref_nodes_[i + 1].loc_.gety();
	d0 = ref_nodes_[i + 1].distance_;
	}
		sumx = sumx + x0;
		sumy = sumy + y0;
	}
}
```

E. Hasil Simulasi

Hasil running program loc1.tcl ini menggunakan satu unknown node dengan tiga anchor node. Dimana posisi unknown node (50,50) dengan anchor node berada dikoordinat A1(30,50); A2(50,20) dan A3 (70,70) dengan communication range yang digunakan yaitu 50 meter. Hasil dari simulasi ini berupa tampilan pada nam console yang menyatakan unknown node akan berwarna hijau sebelum diestimasi namun akan berubah warna menjadi merah saat proses estimasi selesai.



Gambar 2. Estimasi Posisi dengan 1 Unknown Node dan 3 Anchor Node

```
root@ninis:~# grep "L" location.tr
L 2.761160 0 0.499788 3 137.701771 0.000000 2.666667
```

Gambar 3. Estimasi Menggunakan Metode 1

Gambar diatas adalah hasil simulasi menggunakan metode estimasi posisi dengan menggunakan algoritma centroid dengan topologi sebaran node ada digambar 4.98.

```
root@ninis:~# grep "L" location.tr
L 2.761160 0 0.499788 3 128.062488 0.000000 9.999997
```

Gambar 4. Estimasi menggunakan Metode 2

Gambar diatas adalah hasil estimasi menggunakan metode perbaikan dari estimasi posisi menggunakan centroid dengan adanya

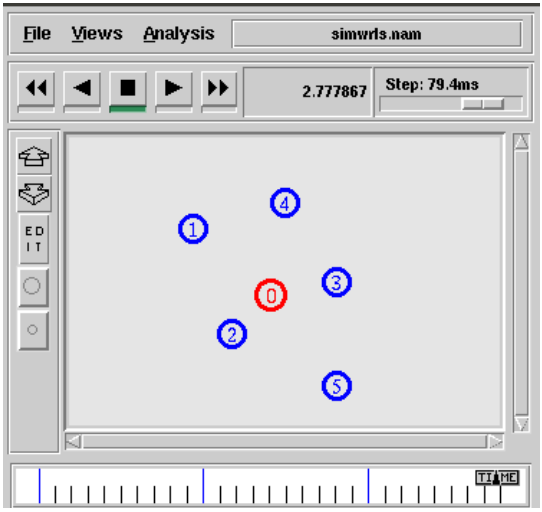
perbaikan dengan menghilangkan error posisi node paling besar, topologi sebaran node ada digambar 4.98.

```
root@ninis:~# grep "L" location.tr
L 2.761160 0 0.499788 3 137.701771 0.000000 2.666667
L 7.508378 0 0.499766 3 137.701771 0.000000 2.666667
L 12.083878 0 0.499743 3 137.701771 0.000000 2.666667
L 19.927362 0 0.499720 3 137.701771 0.000000 2.666667
L 31.295594 0 0.499698 3 137.701771 0.000000 2.666667
L 41.697267 0 0.499675 3 137.701771 0.000000 2.666667
L 49.245477 0 0.499653 3 137.701771 0.000000 2.666667
```

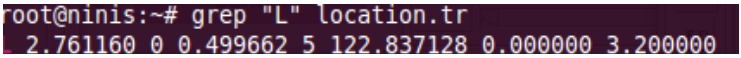
Gambar 5. Estimasi Menggunakan Metode 3

Gambar 5 hasil estimasi menggunakan metode centroid dengan menggunakan 3 node referensi. Berdasarkan hasil pengamatan, estimasi terbaik menggunakan metode ke dua yaitu algoritma centroid dengan perbaikan. Namun hasil estimasi masih jauh dari posisi sebenarnya.

Hasil running program loc2.tcl ini menggunakan satu unknown node dengan 5 anchor node dan 1 unknown node

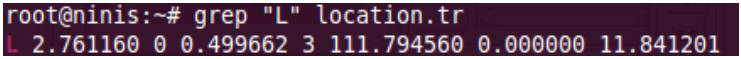


Gambar 6. Estimasi Posisi dengan 3 Unknown Node dan 5 Anchor Node



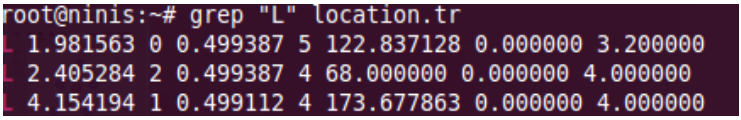
Gambar 7. Estimasi Menggunakan Metode 1

Menggunakan metode estimasi posisi tanpa perbaikan, perhitungan dengan menggunakan algoritma centroid dengan topologi sebaran node ada pada gambar diatas



Gambar 8. Estimasi menggunakan Metode 2

Gambar 8 adalah hasil estimasi menggunakan metode perbaikan dari estimasi posisi menggunakan centroid dengan topologi sebaran node ada pada gambar diatas



Gambar 9. Estimasi Menggunakan Metode 3

Gambar 9 merupakan hasil estimasi menggunakan metode centroid dengan menggunakan 3 node referensi. Berdasarkan hasil pengamatan, estimasi terbaik menggunakan metode ke dua yaitu algoritma centroid dengan perbaikan. Namun hasil estimasi masih jauh dari posisi sebenarnya.

Disusun oleh: Ninis Ari Fianti (Juni 2013)