

Pencarian Rute Terpendek Tempat Penting Melalui *Mobile GMaps* dengan Menggunakan Algoritma Dijkstra

I Nyoman Prama Pradnyana
(13509032)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
prama.pradnyana@itb.ac.id

Abstract—Pencarian rute terpendek merupakan salah satu bentuk pengaplikasian dari teori graf. Dalam teori graf, terdapat pokok bahasan yang mengulas secara detail mengenai pengaplikasian graf yaitu Algoritma Dijkstra. Algoritma Dijkstra ini akan menghasilkan informasi geografis mengenai jarak tempat yang dituju berdasarkan bobot rute kedua tempat. Algoritma ini akan diimplementasikan dengan peralatan navigasi sehingga mempermudah dalam pembacaannya. Alat navigasi yang digunakan berupa software mobile agar mempermudah penggunaan yaitu *Mobile GMaps*. *Mobile GMaps* akan menampilkan map sehingga dapat terlihat rute yang menghubungkan tempat-tempat yang dituju. Sedangkan Algoritma Dijkstra dapat menentukan rute mana yang perlu ditempuh untuk mendapatkan rute terpendek. Melalui kombinasi ini, pengguna alat navigasi dapat mengetahui rute terpendek tempat yang dituju dari media peralatan navigasi.

Index Terms—Algoritma Dijkstra, *Mobile GMaps*

I. PENDAHULUAN

Saat ini sudah banyak kota yang memiliki tingkat pembangunan yang pesat. Sudah banyak kota yang menyediakan berbagai fasilitas masyarakat baik itu rumah sakit, tempat rekreasi, dll. Bagi penduduk tetap, tentu saja telah mengetahui secara detail mengenai tempat-tempat di kotanya, namun untuk para pendatang, tentu saja tidak mengetahui tempat-tempat di kota yang baru mereka kunjungi. Contohnya untuk tempat rekreasi, mereka tentu saja tidak mengetahui secara detail mengenai posisi tempat wisata tersebut.

Pembacaan tempat yang dituju dapat dilakukan melalui media yang disebut dengan peralatan navigasi. Peralatan navigasi ini akan membantu pembaca dalam mengetahui tempat yang ingin diketahui lokasinya. Terdapat banyak jenis alat navigasi, contohnya GPS (*Global Positioning System*), *google earth*, *Mobile GMaps* dll. Seluruh jenis navigasi tersebut memiliki tujuan yang sama yaitu memberikan informasi kedudukan suatu tempat.

Namun yang menjadi masalah saat ini adalah, sangat

jarang terdapat penunjukkan metode rute terpendek pada alat navigasi tersebut. Peralatan navigasi tersebut hanya terbatas menunjukkan rute mana saja yang bisa dilewati. Padahal penunjukkan rute terpendek sangatlah penting mengingat dengan mengetahui rute terpendek yang bisa ditempuh, dapat mengurangi waktu yang dibutuhkan untuk mencapai tempat tersebut. Misalnya saja terdapat pasien yang harus dikirim ke rumah sakit. Tentu saja harus menemukan rute yang dapat ditempuh dalam waktu singkat, dengan kata lain memiliki rute terpendek. Dengan memanfaatkan metode graf, kita dapat mengkombinasikan dengan peralatan navigasi sehingga dengan peralatan navigasi yang ada, dapat diketahui rute mana saja yang menghasilkan jarak terpendek dari tempat asal ke tempat tujuan.

Pada metode graf, terdapat pembahasan khusus mengenai pencarian jarak terpendek yaitu Algoritma Dijkstra. Pada algoritma dijkstra, dilakukan komputasi pencarian rute terpendek antara simpul sumber dan simpul tujuan berdasarkan bobot pada sisi yang menghubungkan simpul-simpul dalam graph.

Metode ini kemudian akan diimplementasikan dalam alat navigasi. Alat Navigasi yang akan diimplementasikan adalah *Mobile GMaps*. *Mobile GMaps* akan menunjukkan map daerah yang akan dicari rutenya, sedangkan dengan penambahan algoritma Dijkstra, *Mobile GMaps* dapat menunjukkan rute terpendek yang menghubungkan tempat asal dan tempat tujuan.

Melalui makalah ini diharapkan dapat memberikan kemudahan bagi masyarakat yang ingin mencari suatu tempat dengan mendapatkan jarak terpendek.

II. DASAR TEORI

Adapun beberapa Dasar Teori yang digunakan sebagai pembandingan dan sumber data yang menyangkut dengan pencarian rute terpendek ini

2.1 *Mobile GMaps*

Aplikasi *Mobile GMaps* merupakan salah satu aplikasi

layanan pada handphone yang berbasis GPS. Aplikasi ini merupakan aplikasi java untuk dapat berkomunikasi dengan berbagai perangkat GPS guna keperluan navigasi [BUY07].

Prinsip kerja aplikasi ini sangat mudah yaitu aplikasi ini menampilkan sebuah map dunia yang dapat diperbesar hingga titik skala 1:300m. Map yang ditampilkan tidaklah realtime sehingga tidak akan terlihat kendaraan yang melintas pada map. Meskipun demikian map yang ditampilkan sangat menggambarkan kondisi sebenarnya dilapangan baik itu kota maupun jalan-jalan yang ada.

Aplikasi ini dipilih karena aplikasi ini merupakan aplikasi mobile yang bersifat freeware sehingga dapat digunakan tanpa keluar biaya untuk mendapatkan software ini. Disamping itu *Mobile GMaps* merupakan aplikasi mobile sehingga penggunaan aplikasi ini menjadi lebih praktis dan lebih mudah dibawa kemana-mana. Cukup menggunakan handphone sendiri bisa menggunakan aplikasi ini. Selain itu aplikasi ini dapat digunakan di berbagai jenis handphone yang tersedia OSnya pada *Mobile GMaps* seperti Java, Blackberry dll.

Sedangkan untuk kelemahan dari aplikasi ini dibutuhkan biaya koneksi GPRS. Biaya yang digunakan bergantung pada tarif yang telah ditetapkan oleh masing-masing provider. Disamping itu, masih ada handphone yang tidak dapat menggunakan aplikasi ini jika tidak memiliki OS yang terdaftar.



Gambar 1 : Interface aplikasi *Mobile GMaps*

2.2 Graf

Secara geometri graf digambarkan sebagai sekumpulan noktah (simpul) di dalam bidang dwimatra yang dihubungkan dengan sekumpulan garis (sisi) [MUN09]. Teori graf sangat sering digunakan pada kehidupan

sehari-hari seperti pada rangkaian listrik dan isomer senyawa karbon. Pada rangkaian listrik memanfaatkan teori graf dalam menyusun rangkaian sehingga rangkaian tersebut dapat dicari turunannya. Pemanfaatan teori ini dalam rangkaian listrik pertama kali digunakan oleh kirchoff untuk penyusunan rangkaian paralel dan seri mengenai persamaan arus.

Disamping diaplikasikan dalam berbagai bidang, teori graf juga dikembangkan dalam berbagai fungsi misalnya untuk menentukan lintasan terpendek, menentukan metode pengiriman barang, metode pewarnaan (dalam pencitraan).

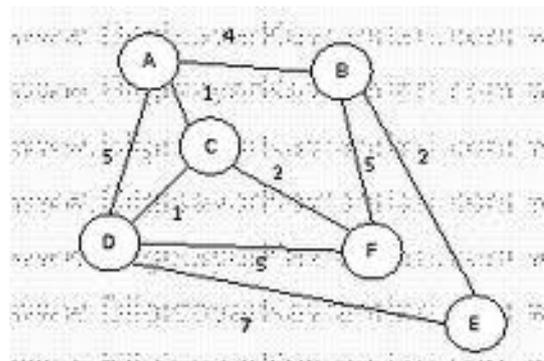
Dalam makalah ini, graf akan langsung dimasukkan ke dalam teori Algoritma Dijkstra

2.3 Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif [WIK10].

Algoritma ini pertama kali dikemukakan oleh Edsger W. Dijkstra (1959) dan telah secara luas digunakan dalam menentukan rute tersingkat atau jalur terpendek berdasarkan kriteria tertentu yang digunakan sebagai batasan. Khususnya dalam bidang transportasi, tentu saja, batasan yang digunakan tersebut adalah dengan menggunakan fungsi biaya arus.

Prinsip kerja algoritma ini adalah dengan mengecek simpul-simpul pada graph dimulai dengan simpul sumber. Algoritma Dijkstra akan memilih jarak terpendek dari simpul-simpul terdekat dan menghitung total bobot semua sisi yang dilewati untuk mencapai simpul tujuan.



Gambar 2 : Graf

Dalam penentuan jalur terpendek pada graf di atas di butuhkan titik awal dan titik tujuan. Misalkan titik awal A dengan titik tujuan E dengan graf kita dapat mengetahui jalur mana yang dapat dilalui, dengan ketentuan jalur jalan yang telah dilalui dengan rute yang sama tidak boleh

dilalui lagi dan jika telah sampai pada titik tujuan maka pencarian pada rute tersebut berhenti dan melanjutkan dengan rute lain, jika semua rute telah dilalui kita dapat mengambil kesimpulan rute mana yang akan dilalui dengan jarak yang optimum atau terpendek.

Dari graf tersebut dapat ditentukan rute yang berhasil mencapai tujuan dari vertices A ke vertices G yaitu :

- A ⇒ B ⇒ E dengan total jarak = 6
- A ⇒ B ⇒ F ⇒ C ⇒ D ⇒ E dengan total jarak = 19
- A ⇒ B ⇒ F ⇒ D ⇒ E dengan total jarak = 21
- A ⇒ C ⇒ F ⇒ B ⇒ E dengan total jarak = 10
- A ⇒ C ⇒ F ⇒ D ⇒ E dengan total jarak = 15
- A ⇒ C ⇒ D ⇒ E dengan total jarak = 9
- A ⇒ D ⇒ C ⇒ F ⇒ B ⇒ E dengan total jarak = 15
- A ⇒ D ⇒ E dengan total jarak = 12

Dari hasil tersebut jarak terpendek adalah A ⇒ B ⇒ E dengan total jarak 6. Dalam penerapan algoritma dijkstra pada graf di atas merupakan graf tidak berarah, tetapi algoritma Dijkstra dapat juga diterapkan pada graf berarah.

III. METODE

Adapun beberapa Metode yang akan digunakan dalam implementasi program ini.

3.1 Instalasi *Mobile GMaps*

Instalasi aplikasi *Mobile GMaps* ini sangat mudah, yaitu dengan mengunjungi WAP <http://wap.mgmaps.com> dengan menggunakan handphone langsung. Setelah masuk kedalam situs, carilah link yang sesuai dengan OS handphone yang digunakan misalnya Nokia dengan OS JAVA bisa menggunakan Link Generic (full) release. Setelah mengklik link tersebut, aplikasi *Mobile GMaps* akan langsung terinstall pada handphone. Setelah itu buka icon mobile Maps, dan lakukan setting GPRS sehingga handphone dapat terkoneksi dengan internet, dan lakukan setting GPS misalnya dengan menggunakan GPS Bluetooth. *Mobile GMaps* siap digunakan.

3.2 Penggunaan *Mobile GMaps*

Setelah berhasil dengan setting GPRS dan GPS pada aplikasi *Mobile GMaps*, carilah daerah lokasi yang akan dicari rutanya. Pencarian dapat dilakukan dengan menggunakan joystick pada handphone. Mengenai pembesaran dapat dilakukan dengan tombol * untuk memperbesar dan tombol # untuk memperkecil skala map.

IV. PEMBAHASAN

A. Pseudocode Algoritma Dijkstra

Algoritma Dijkstra dapat dituliskan dalam berbagai bentuk bahasa baik itu bahasa pemrograman maupun HTML. Pada kasus ini digunakan penulisan Algoritma

Dijkstra dalam bahas pemrograman. Algoritma inilah yang diimplementasikan ke dalam *Mobile GMaps*.

```

{ Algoritma Dijkstra }

procedure Dijkstra
(
    input m: matriks,
    a: integer {simpul awal}
)

{mencari lintasan terpendek dari
simpul awal a ke semua simpul
lainnya
Masukan : matriks ketetangga (m)
dari graf berbobot G dan simpul
awal a
Keluaran: lintasan terpendek dari
a ke semua simpul lainnya}

KAMUS
s1, s2, ..., sn : integer {larik
interger}

d1, d2, ..., dn : integer {larik
interger}
i : integer

ALGORITMA
{Langkah 0 (inisialisasi) : }
for i ← 1 to n do
    si ← 0
    di ← mai
endfor

{Langkah 1: }
sa ← 1
{karena simpul a adalah simpul asal
lintasan terpendek, jadi terpilih
dalam lintasan terpendek}
da ← infinity
{tidak ada lintasan terpendek dari
simpul a ke a}

{Langkah 2,3,...,n1 :}
for i ← 2 to n-1 do
    {Cari j sedemikian sehingga sj = 0
    dan dj = min (d1,d2,...,dn)}
    sj ← 1
    {simpul j sudah terpilih ke dalam
    lintasan terpendek}
    {perbarui di, untuk i = 1,2,3,...,n
    dengan : di (baru) = min{di (lama),
    dj + mji}
endfor

```

Algoritma di atas menjelaskan tentang bagaimana kinerja dari “Algoritma Dijkstra” menganalisis masalah untuk dapat menentukan rute terpendek. Pada algoritma ini pertama menganalisis kondisi masukan simpul awal pada graf tertentu kemudian dicari simpul mana yang memiliki jarak terdekat dengan simpul a. Kemudian algoritma ini diulang terus dan hasil akhirnya dibandingkan sehingga didapat rute terpendek dari graph yang diuji

Pada algoritma di atas terdapat dua variable yang menjadi input yaitu m dideklarasikan sebagai matriks, dan a dideklarasikan dengan integer.

```

procedure Dijkstra
(
  input m: matriks,
  a: integer {simpul awal}
)

```

M yang dideklarasikan sebagai matriks merupakan variable untuk graph yang akan dicari rute terpendeknya. Implementasi pada *Mobile GMaps* adalah map yang ditampilkan. *Mobile GMaps* akan menampilkan map yang terdapat lokasi acuan, lokasi tujuan dan beberapa lokasi yang menjadi penghubung diantara kedua lokasi tersebut. Seluruh lokasi tersebut merupakan simpul yang nantinya akan digunakan untuk mengakses lokasi tujuan dengan membandingkan jarak lokasi-lokasi tersebut dengan titik awal. Sedangkan a yang juga merupakan input merupakan variable yang menunjukkan simpul awal. Simpul ini yang akan digunakan sebagai titik acuan awal. Implementasi pada *Mobile GMaps* adalah berupa lokasi awal user. *GMaps* akan mendeteksi lokasi awal user dengan bantuan GPS pada *Mobile GMaps*. Setelah itu lokasi ini akan dihubungkan dengan lokasi lain yang berhubungan langsung dengan lokasi awal ini. Setiap lokasi yang berhubungan akan dicatat dan dibandingkan jaraknya.

Pada algoritma di atas juga terdapat pendeklarasian beberapa variable yang dipakai nantinya saat membandingkan lokasi lainnya.

```

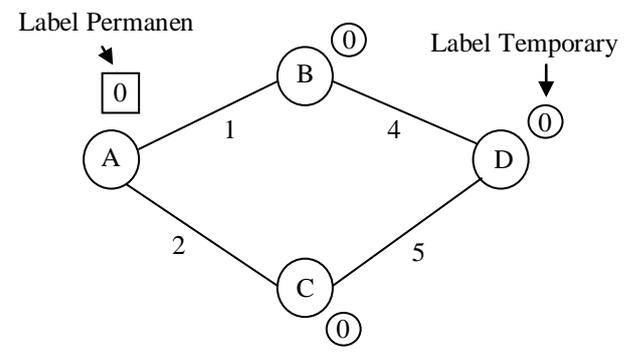
KAMUS
s1, s2, ..., sn : integer {larik
integer}
d1, d2, ..., dn : integer {larik
integer}
i : integer

```

Pada pendeklarasian Kamus digunakan s_n sebagai penunjuk simpul. Jarak dari titik awal akan dibandingkan dan dicari jarak terpendek. Implementasi dengan *Mobile GMaps* adalah tempat yang berhubungan langsung dengan posisi awal. Untuk d_n digunakan untuk melakukan proses pengulangan sebagai media penyimpanan matriks

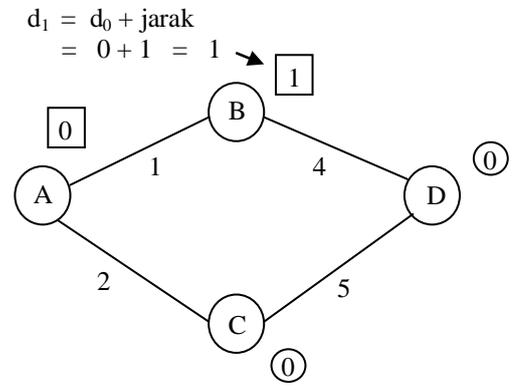
berikutnya yang akan dicek setelah matriks dengan titik awal selesai dibandingkan. Sedangkan I digunakan untuk menunjukkan kota ke berapa yang digunakan sebagai perbandingan.

Untuk bagian algoritmanya merupakan proses pencarian jarak terpendek dari lokasi awal dengan simpul-simpul di sekitarnya. Pada Langkah 0, merupakan langkah awal yang digunakan untuk menjadikan seluruh titik bernilai 0 [NUR06].



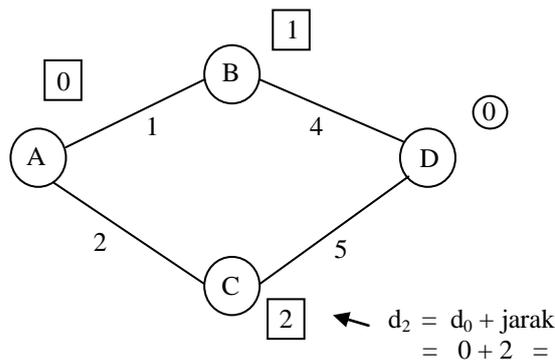
Gambar 3 : Inisialisasi Awal

Setelah melakukan implementasi, dilakukan pencarian rute terpendek dari titik asal dengan simpul yang berhubungan secara langsung dengan titik awal tersebut. Pada gambar terlihat bahwa simpul B lah yang memiliki jarak yang paling dekat dengan simpul A. Maka Label temporary pada simpul B diubah menjadi nilai jarak yang telah ditempuh yaitu sebesar 2 dan label temporary diubah menjadi label Permanen.



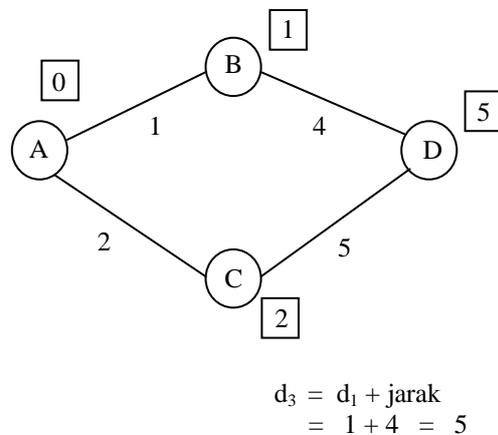
Gambar 4 : Perubahan Simpul Pertama

Selanjutnya ialah menemukan nilai sisi terendah dari simpul dengan label sementara dari simpul yang telah memiliki label permanen. Pada gambar simpul yang menjadi acuan adalah simpul A ataupun simpul B yang mencari jarak terpendek dengan simpul sekitarnya.



Gambar 5 : Perubahan Simpul Pertama

Proses tersebut terus diulang hingga menemukan titik akhir dari graf tersebut yaitu titik tujuan. Jika pada gambar berikut yang merupakan titik tujuan adalah titik D. Sebelumnya telah dilakukan proses perubahan label temporary menjadi label permanen pada seluruh simpul kecuali pada titik akhir. Proses terakhir kali ini sama dengan sebelumnya yaitu dengan membandingkan jarak terpendek dari setiap titik yang sudah memiliki label permanen pada gambar merupakan titik B dan C. namun terdapat sedikit perbedaan pada akhir ini yaitu setelah mendapatkan jarak terpendek akan dilakukan perbandingan ulang sebagai cek ulang sehingga didapatkan jawaban yang tepat.



Gambar 5 : Perubahan Simpul Pertama

Pada gambar di atas diperlihatkan bahwa titik akhir telah memiliki label permanen. Label permanen inilah yang menunjukkan jarak rute terpendek yang bisa digunakan. Label permanen tersebut merupakan hasil dari perbandingan dengan rute-rute lain yang memungkinkan.

Demikianlah proses kerja algoritma Dijkstra secara umum. Algoritma ini pada intinya menggunakan prinsip dari graph yang diterjemahkan ke dalam notasi algoritmik sehingga dapat digunakan pada berbagai bahasa pemrograman sehingga dapat memanfaatkan teori graph dalam berbagai program. Secara umum algoritma ini membandingkan rute-rute yang memiliki jarak terpendek

dibandingkan dengan rute alternative lain. Jika pada simpul terdapat lebih dari satu rute (kondisi dimana label permanen memiliki lebih dari satu nilai) maka yang diambil adalah nilai terkecil yang akan dituliskan sebagai label permanen selanjutnya. Begitu seterusnya hingga berakhir di titik akhir

Sedangkan terdapat kondisi lain dimana terdapat rute yang memiliki jarak yang sama. Untuk kondisi seperti ini maka akan diberi kebebasan memilih rute yang mana tergantung pada penulisan algoritmiknya apakah memilih lintasan yang dicek pertama kali atau sesudahnya.

B. Kelemahan Program

Meskipun dalam aplikasi *Mobile GMaps* telah diimplementasikan dengan menggunakan Algoritma Dijkstra, masih terdapat beberapa yang menjadi kelemahan dari program ini.

- Masih belum menjamin sampai di tempat lokasi lebih cepat meskipun telah melalui rute terpendek. Hal ini disebabkan beberapa factor misalnya keadaan jalan yaitu macet. Jika melalui rute terpendek namun kondisi jalannya sedang macet, maka waktu untuk sampai ditujuan belum tentu lebih cepat dibandingkan jika menggunakan rute lainnya meskipun lebih panjang.
- Pencarian lokasi tidak dapat digunakan untuk lokasi yang tidak terdeteksi oleh *Mobile GMaps* atau yang tidak memiliki akses langsung dengan lokasi penting.

IV. LINK

Adapun beberapa Link yang dapat membantu dalam proses penggunaan aplikasi ini.

Untuk mendownload *Mobile GMaps* : <http://wap.mgmaps.com>

V. KESIMPULAN

Implementasi Algoritma Dijkstra ke dalam alat navigasi *Mobile GMaps* dapat membantu dalam menentukan rute terpendek yang bisa ditempuh untuk mencapai tujuan.

REFERENCES

[MUN09] Munir, Rinaldi.2009.*Diktat Kuliah IF 2091 Struktur Diskrit*.Bandung : INFORMATIKA

[MAY08] Mayhoney, "Algoritma Dijkstra", http://www.itelkom.ac.id/library/index.php?view=article&catid=20%3Ainformatika&id=161%3Aalgoritma-dijkstra&option=com_content&Itemid=15
Waktu : Sabtu, 11 Desember 2010 [23:00]

[GER08] Gerwandi, "Aplikasi GPS Navigasi di ponsel Java (J2ME), *Mobile GMaps*",

<http://aplikasihp.blogspot.com/2008/03/aplikasi-gps-navigasi-di-ponsel-java.html>

Waktu : Sabtu, 11 Desember 2010 [23:30]

[BUY07] Buyung Akram, "Aplikasi GPS Navigasi di ponsel Java (J2ME), *Mobile GMaps*",

<http://navigasi.net/goart.php?a=gpmgmaps>

Waktu : Sabtu, 11 Desember 2010 [23:30]

[WIK10] Wikipedia, "Algoritma Dijkstra",

http://id.wikipedia.org/wiki/Algoritma_Dijkstra

Waktu : Sabtu, 11 Desember 2010 [23:30]

[NUR06] Nur Fajriah, "Aplikasi Algoritma Dijkstra dalam pencarian lintasan terpendek",

<http://www.informatika.org/~rinaldi/Matdis/2007-2008/Makalah/MakalahIF2153-0708-113.pdf>

Waktu : Sabtu, 11 Desember 2010 [23:30]

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Desember 2010

ttd

I Nyoman Prama Pradnyana
13509032