

# Penerapan Graf pada Robot *Micromouse*

Nurwanto (13511085)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13511085@std.stei.itb.ac.id

**Abstract**—*Micromouse* merupakan jenis robot *autonomous* atau robot pengendali otomatis yang secara umum di Indonesia dikenal dengan robot cerdas. Visi robot *micromouse* ini mencari jalan ke suatu titik atau posisi tujuan dan kembali lagi ke posisi awal saat start. Dalam penerapannya robot ini menggunakan teori graf untuk pencarian lintasan (*path*). Robot ini sebelumnya tidak mengetahui akan kondisi lintasan, namun disuruh mencari lintasan sendiri dan dari berbagai pencarian tersebut lalu dicari lintasan (*path*) terbaik atau terpendek. Setelah mendapatkan lintasan (*path*) terpendek tersebut maka *micromouse* tersebut akan melakukan tracking sesuai dengan lintasan yang telah ditentukannya. Pada tracking ini robot *micromouse* akan memaksimalkan kecepatannya untuk menempuh lintasan tersebut.

**Index Terms**—*Micromouse*, lintasan (*path*), lintasan atau sirkuit euler, graph, node atau simpul, sisi, dan *microprocessor*.

## I. PENDAHULUAN

*Micromouse* merupakan event yang mana robot kecil mirip tikus yang dapat menyelesaikan misi pada labirin 16 x 16. *Micromouse* ini muncul pada tahun 1970 namun ada yang mengindikasikan kemunculan *micromouse* ini terjadi pada 1950[1]. Tugas utama dari *micromouse* ini yaitu memetakan jalur labirin untuk sampai ke pusat atau pos tujuan dari labirin, setelah sampai pada misi tujuan di labirin maka robot *micromouse* tersebut akan kembali lagi keposisi start.

Dalam perkembangannya *micromouse* dapat bergerak dengan kecepatan 3(tiga) meter per detik, rekor waktu terbaik dalam kompetisi ini yaitu 4 (empat) detik untuk sampai melakukan visi robot tersebut dalam melakukan *tracking* (menuju keposisi tujuan). Kompetisi yang berkembang sekarang ini mempunyai track labirin yang bervariasi, baik dalam ukuran maupun dalam tingkat kesulitan labirin. Kompetisi *micromouse* ini telah berkembang di Inggris, Amerika Serikat, Japan, India, dan Korea Selatan.

Kompetisi ini merupakan pengembangan dari robot line follower yang mengikuti garis dan robot *fire fighting* yang mempunyai visi memadamkan api.

Teori dalam struktur diskrit yang mendukung untuk diaplikasikan dalam *micromouse* ini yaitu teori graf, graf ini digunakan untuk memetakan jalur yang efektif.



Gambar 1. *Micromouse*

<https://www.google.com/search?q=robot+micromouse&hl>



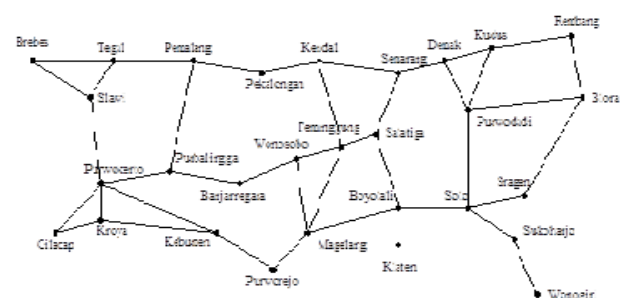
Gambar 2. Lintasan *micromouse*

<https://www.google.com/search?q=lintasan+micromouse&hl7OULGiM4PKrAf01oDoCQ&ved>

## II. DASAR TEORI

Graf merupakan kumpulan simpul (nodes) yang dihubungkan satu sama lain dengan menggunakan sisi atau busur[2].

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut[3].



Gambar 3. Aplikasi graf

Sebuah graf memiliki simpul (nodes) dan sisi/busur yang dapat dinotasikan sebagai berikut.

Graf  $G = (V, E)$ , yang mana  $V$  merupakan himpunan tidak kosong dari simpul-simpul (*vertices*) dari  $\{V_1, V_2, \dots, V_n\}$ . Sedangkan notasi  $E$  merupakan himpunan sisi (*edges*) yang menghubungkan sepasang simpul dari  $\{e_1, e_2, \dots, e_n\}$ [3].

Istilah – istilah umum dalam graf yang sering digunakan yaitu node (simpul), sisi, lintasan, panjang lintasan, sirkuit, lintasan dan sirkuit Euler, ketetanggaan, simpul terpercil, graf kosong, derajat, sirkuit, terhubung, upagraf.

Nodes (simpul) merupakan tempat atau posisi pada graf yang biasanya direpresentasikan dalam bentuk titik.

Sisi (*edge*) merupakan garis atau penghubung antara dua node dalam graf.

Lintasan (path) merupakan barisan berselang seling simpul-simpul dan sisi-sisi yang berbentuk  $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n$  sehingga  $e_1 = (v_0, v_1)$ ,  $e_2 = (v_1, v_2)$ ,  $e_n = (v_{n-1}, v_n)$ .

Panjang lintasan adalah jumlah sisi dalam lintasan tersebut[3].

Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung[3].

Untuk sembarang sisi  $e = (v_j, v_k)$  dikatakan  $e$  bersisian dengan simpul  $v_j$ , atau  $e$  bersisian dengan simpul  $v_k$ [3].

Simpul terpercil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya[3].

Graf kosong merupakan graf yang himpunan sisinya merupakan himpunan kosong  $(N_n)$ [3].

Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut[3].

Sirkuit atau siklus merupakan lintasan yang berawal dan berakhir pada simpul yang sama[3].

Panjang sirkuit adalah jumlah sisi dalam sirkuit tersebut[3].

Terhubung jika dua buah simpul  $v_1$  dan simpul  $v_2$  terdapat lintasan dari  $v_1$  ke  $v_2$ [3].

Upagraf, misalkan  $G = (V, E)$  adalah sebuah graf.  $G_1 = (V_1, E_1)$  adalah upagraf (subgraph) dari  $G$  jika  $V_1$  merupakan himpunan bagian dari  $V$  dan  $E_1$  merupakan himpunan bagian dari  $E$ [3].

Lintasan euler merupakan lintasan yang melalui masing-masing sisi di dalam graf tepat satu kali.

Sedangkan sirkuit euler merupakan sirkuit atau siklus yang melewati sisi-sisi pada graf tepat satu kali[3].

Lintasan Hamilton merupakan lintasan yang melalui masing masing tepat satu kali, sedangkan sirkuit Hamilton merupakan sirkuit yang melewati simpul-simpul pada graf tepat satu kali.

### III. SISTEM KERJA MICROMOUSE

Komponen penyusun micromouse terdiri dari :

**Motor**, yang berfungsi sebagai penggerak pada robot.

**Sensor**, untuk mendeteksi adanya tembok labirin.

**Sumber tegangan**, sumber tegangan ini biasanya memakai baterai berasal dari baterai lithium.

**Mikroprocessor**, pusat kendali robot yang dapat merekam jalur yang sudah dilewatinya dan mencari jalur yang terpendek yang harus ditempuhnya.

pada micromouse terdapat dua mode dasar, mode eksplorasi (pemetaan jalur) dan mode memory. Dalam mode eksplorasi, robot akan melalui semua jalan yang ada pada labirin sedangkan mode memory robot akan bergerak sesuai dengan data yang tersimpan pada mikrokontroler EEPROM yang diperoleh dari mode eksplorasi. Robot ini menggunakan algoritma Depth-First Search untuk mencari jalan keluar dalam suatu labirin. Sistem pencarian ini menggunakan data perblok, micromouse ini bergerak bebas di dalam sebuah labirin (maze) tanpa menyentuh obyek sekitarnya, robot mengetahui ke arah mana harus bergerak, berapa dan derajat harus berputar jika menemui jalan buntu pada labirin.

Robot micromouse merupakan jenis robot Autonomous robot yang berarti dapat dikendalikan secara otomatis. Autonomous Robot merupakan pengendalian gerakan robot yang berdasarkan pada program kemudi yang diberikan sehingga seolah - olah robot tersebut bergerak sendiri.

Pada micromouse terdapat timer yang berfungsi sebagai pengatur delay dan juga berguna untuk menentukan jarak suatu sisi pada aplikasi graf di micromouse. Timer ini akan melakukan perhitungan dengan frekuensi tertentu sesuai dengan kebutuhan.

Agar tidak menabrak labirin maka digunakan sensor jarak pada robot tersebut. Sensor jarak yang paling umum digunakan yaitu sensor inframerah, sistem kerja sensor cahaya inframerah ini yaitu sensor mengirim transmitter berupa cahaya dan jika mengenai tembok labirin maka dipantulkan kembali ke receiver yang terdapat pada robot micromouse. Namun, akhir-akhir ini banyak menggunakan sensor lain seperti sensor suara dengan menerapkan suara ultrasonic. Sensor suara lebih efisien dibandingkan dengan sensor cahaya karena sensor cahaya dapat dipengaruhi oleh kualitas pencahayaan, warna pada tembok labirin dan

Sensor inframerah tersebut akan mempengaruhi kerja dari motor, jika sensor inframerah mengenai labirin maka cahaya tersebut dipantulkan kembali oleh labirin dan receiver mendapat cahaya pantulan maka kerja dari motor

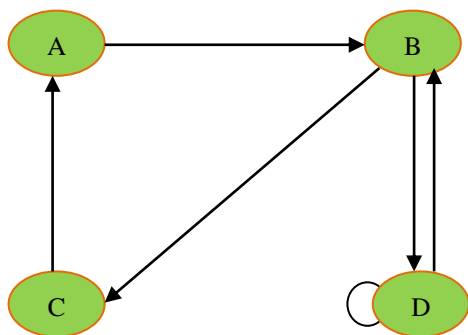
micromouse bagian tersebut tidak aktif. Sehingga micromouse tersebut akan membelok. Jika micromouse menemui jalan buntu maka micromouse diatur supaya dalam kondisi tertentu harus melakukan putaran berpa derajat untuk mencari jalan lain yang bebas, dalam kasus jalan buntu ini maka robot harus berputar 360° untuk melakukan perjalanannya kembali.

#### IV. GRAF PADA MICROMOUSE

##### A. Prinsip pemetaan jalur

Dalam graf kita mengenal panjang lintasan, panjang lintasan merupakan jumlah sisi yang dilaluinya. Dalam graf berbobot sisi dari suatu graf mempunyai nilai yang menyatakan panjang dari sisi tersebut.

Besarnya nilai panjang sisi untuk kasus pencarian jalur oleh micromouse ini diperoleh dari besarnya jarak yang ditempuh micromouse sampai berhenti atau berotasi (berputar). Pengukuran jarak tersebut didapat dari berapa kali roda berputar dengan menghitung keliling lingkaran pada roda tersebut yang sebelumnya nilai jari-jari roda sudah diketahui.



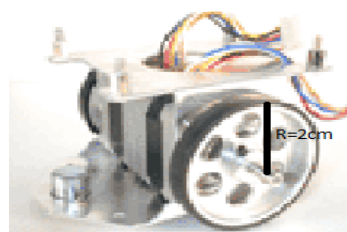
Gambar 4. Contoh graf penentuan jarak / panjang sisi

Keterangan gambar :

- A : Kondisi awal start atau reset
- B : Kondisi dimana robot berhenti
- C : Kondisi robot berputar atau belok
- D : Kondisi robot tidak berputar atau jalan terus

Dari skema tersebut saat robot berada pada kondisi A (kondisi start atau kondisi reset) maka robot akan melakukan alokasi memori baru untuk menyimpan nilai panjang sisi yang baru dibacanya. Robot ini akan bergerak random atau acak dalam menelusuri jalur. Kondisi B merupakan kondisi dimana robot akan berhenti, maksud berhenti disini merupakan kondisi yang mana semua roda berhenti bergerak atau hanya ada satu roda yang berhenti bergerak maka dalam kondisi tersebut dianggap sebagai kondisi berhenti.

Pada saat peralihan dari kondisi A ke kondisi B maka memori akan mulai melakukan perhitungan, perhitungan tersebut seperti yang dikatakan sebelumnya dengan menggunakan data keliling lingkaran dari roda dan mengecek berapa kali roda berputar (*looping*), maka dari data tersebut dapat diketahui jarak atau panjang lintasan lurus yang dalam hal ini dianggap sebagai sisi sebuah graf.



Gambar 5. Ukuran roda micromouse

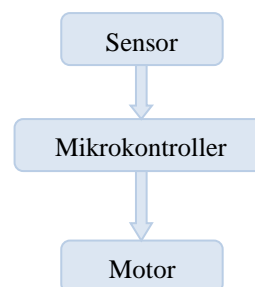
Sebagai contoh pada gambar 5 tersebut kita ketahui bahwa data jari-jari dari roda micromouse tersebut ialah 2 (dua) cm misalkan kondisi sampai berhenti dan pindah node (simpul) tersebut memerlukan 5 (lima) putaran maka dapat dilakukan perhitungan secara matematis sebagai berikut :

$$\begin{aligned} \text{Keliling roda} &= 2 \times \pi \times R \\ \text{Dengan } \pi &: \text{konstanta } 3.14 \\ R &: \text{jari-jari roda} \end{aligned}$$

$$\begin{aligned} \text{Maka jarak sisi tersebut,} \\ \text{Jarak} &= \text{jumlah putaran} \times \text{keliling} \\ &= 5 \text{ putaran} \times 2 \times \pi \times 2 \text{ cm} \\ &= 62.8 \text{ cm} \end{aligned}$$

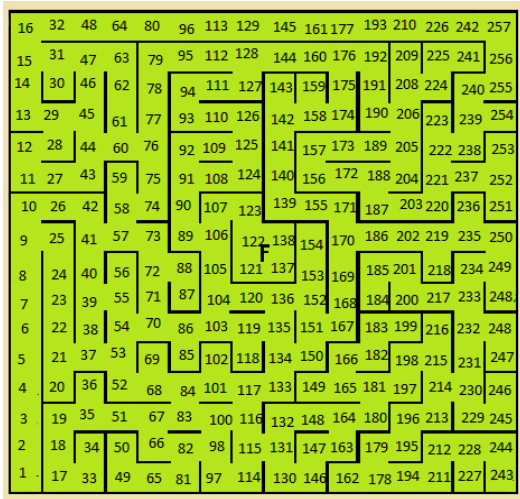
Saat berada pada kondisi B maka robot akan mengecek kondisi selanjutnya apakah kondisi C yang mana micromouse akan berputar atau belok atau ke kondisi D yang mana robot akan berhenti lalu melanjutkan jalan lagi. Pada saat mencapai pada kondisi C maka perhitungan diakhiri dengan mereset dan bikin alokasi memori baru dengan pindah ke kondisi A. Jika kondisi berhenti (kondisi C) tersebut ke kondisi D yang mana micromouse akan melanjutkan perjalanan maka perhitungan dilanjutkan sampai pada kondisi berhenti (kondisi B) dan seterusnya dilakukan pengulangan.

Namun sebelum kita membahas lebih lanjut tentang aplikasi graf pada micromouse ini, alangkah baiknya kita mengetahui skema kerja dari micromouse.



Gambar 6. Skema kerja micromouse

Berikut contoh pemetaan labirin(maze) atau lintasan micromouse.



Gambar 6. Sirkuit micromouse

Pada lintasan tersebut robot *micromouse* melakukan start pada posisi 1. Pada kesempatan pertama robot tersebut melakukan pencarian jalur tersendiri dengan mengeksplorasi atau menelusuri semua jalur yang ada. Hasil penelusuran tersebut disimpan memori mikrokontroler. Setelah selesai melakukan *searching* atau pencarian jalan untuk sampai kefinish. Diposisi tujuan atau finish tersebut robot akan melakukan pemetaan dengan menggunakan prinsip graf hamilton yang memilih lintasan terpendek dengan syarat melewati node atau simpul tepat satu kali.

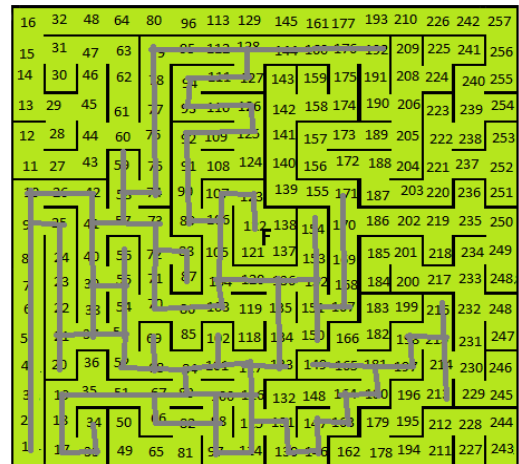
Pada kebanyakan kasus, sering kali kita melakukan prioritas pemilihan jalur saat pencarian jalur ke finish tersebut. Bentuk prioritas yang sering dipakai yaitu prioritas dalam memilih jalur yang paling kanan. Pemilihan tersebut karena dalam suatu lomba atau permainan biasanya awal mulai start berada pada posisi disebelah kiri sedangkan untuk posisi tujuan atau posisi akhirnya berada pada sebelah kanan sehingga kondisi ini dimanfaatkan untuk mempercepat dalam pencarian jalur.

Pada gambar 6, jalur tercepat dari posisi start sampai ke finish (digambar ditunjukkan dengan huruf F) dapat direpresentasikan dalam bentuk matriks 1/0.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0
1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	0
1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0
1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Salah satu contoh dalam suatu permainan robot *micromouse* tersebut mendapatkan jalur yang

diimplementasikan dalam graf seperti pada gambar berikut.



Gambar 7. Contoh pencarian jalur sampai garis finish

Lintasan atau jalur pencarian pada gambar 7 yang ditunjukkan dengan garis warna abu – abu, menunjukkan banyaknya kemungkinan untuk mencapai tempat tujuan (finish). Pada gambar tersebut angka-angka yang terdapat dalam kotak merupakan suatu node atau simpul. Sedangkan kotak tersebut merupakan grafnya. Pada saat pencarian jalur maka node yang tidak dilalui oleh *micromouse* maka akan diabaikan oleh system pada mikrokontroler. Sisi-sisi yang terbentuk yaitu penghubung antar node (simpul) yang dilewatinya sesuai dengan jalur yang dicari. Untuk jalur yang tidak diperlukan atau menjauh serta buntu maka jalur tersebut akan dilakukan *cut-set* atau dilakukan pemotongan.

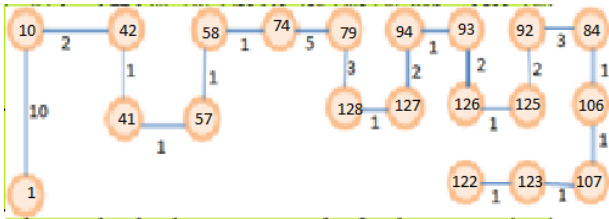
Sisi yang dilakukan *cut-set* pada gambar 7 dapat dilihat pada table dibawah ini.

No	Sisi(node-node)	Penghubung ke upagraf dengan nodes yang dihapus/diabaikan
1	84 - 83	17, 18, 19, 33, 34, 35, 51, 56, 57, 82, 97, 98, 100, 114, 115, 131, 130, 146,147, 163, 164, 180, 149, 166, 181, 197, 198, 213, 214, 215, 216
2	116 - 117	17, 18, 19, 33, 34, 35, 51, 56, 57, 82, 97, 98, 100, 114, 115, 131, 130, 146,147, 163, 164, 180, 149, 166, 181, 197, 198, 213, 214, 215, 216
3	20 – 21	
4	68 - 69	
5	101 - 102	
6	136 - 152	150, 151, 152, 153, 154, 167, 168,169, 170, 171
7	128 - 144	160, 176, 192
8	58 -59	
9	72 - 88	87
10	55 - 56	

Tabel 1. Tapel upagraf yang dicut-set

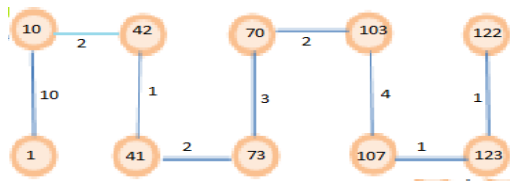
Dari tabel tersebut maka terdapat 4 (empat) jalur yang dapat menghubungkan ke tempat tujuan atau finish.

Berikut uraian empat jalur utama yang direpresentasikan dalam graf.



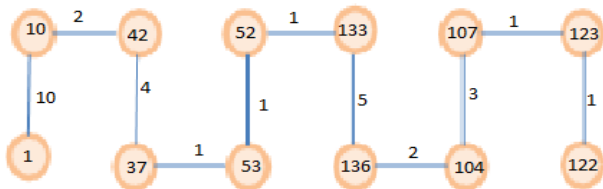
Gambar 8. Jalur pertama

Panjang lintasan pada jalur pertama yaitu  $10+2+1+1+1+1+5+3+1+2+1+2+1+2+3+1+1+1+1=40$



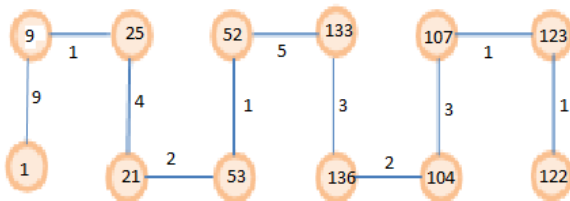
Gambar 9. Jalur kedua

Panjang lintasan untuk jalur kedua ini yaitu  $10+2+1+2+3+2+4+1+1=26$



Gambar 10. Jalur ketiga

Panjang lintasan untuk jalur ketiga ini yaitu  $10+2+4+1+1+1+5+2+3+1+1=31$



Gambar 11. Jalur keempat

Panjang lintasan pada jalur yang keempat yaitu  $9+1+4+2+1+5+3+2+3+1+1=32$

Dari kondisi ini maka *micromouse* akan memilih dari 4 kemungkinan jalan tersebut, sehingga *micromouse* akan melakukan keputusan untuk memilih jalur kedua dengan jarak terpendek dari 4 (empat) jalur tersebut.

## V. KESIMPULAN

*Micromouse* merupakan robot yang mempunyai misi untuk melakukan pencarian jalur tercepat untuk mencapai posisi tujuan atau posisi finish. Untuk mencapai tujuan tersebut, robot ini menggunakan sistem pencarian jalur dengan mencoba semua jalur yang ada (merandom jalur).

Hasil *randomize* tersebut dipetakan atau mapping ke dalam mikrokontroler. Dalam mikrokontroler tersebut akan dicari jalur yang terhubung ke finish dengan menggunakan prinsip graf dengan membuang simpul-simpul yang tidak dibutuhkan. Setelah itu dicari lintasan yang terpendek dari data jarak tiap sisi pada graf tersebut. Setelah jarak ditentukan maka lintasan tersebut direkam dan saat robot dilakukan start maka robot akan jala sesuai dengan lintasan yang dibuatnya. Jika robot telah mencapai *finish* maka robot akan menggunakan jalur yang sama untuk kembali keposisi awal saat *start*.

## REFERENCES

- [1] <http://www.ntf.or.jp/mouse/>
- [2] <http://ienx.files.wordpress.com/2007/12/bab-ii-landasan-teori.pdf>
- [3] Munir Rinaldi, 2009, *Matematika Diskrit*, Bandung : Penbit Informatika. Bab 8.

## 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, 18 Desember 2012

Nurwanto (13511085)