

Penerapan Fuzzy Logic untuk Pembatasan Jumlah Partikel Pada Aplikasi yang Menggunakan Sistem Partikel

Biolardi Yoshogi (13509035)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
vsio@students.itb.ac.id

Abstract—Dalam membangun aplikasi sistem partikel, akan digunakan banyak gambar dan instance. Pembuatan instance yang berlebihan terutama jika akan melebihi batas RAM, dapat memperlambat penggunaan aplikasi lain. Untuk itu digunakan logika fuzzy agar bisa menentukan batas maksimum jumlah partikel yang bisa diemisikan dengan memperhatikan sisa RAM yang dapat digunakan dan besar RAM yang digunakan untuk sebuah partikel.

Kata Kunci— fuzzy logic, sistem partikel, memori, optimasi

I. PENDAHULUAN

Aplikasi sistem partikel banyak digunakan dalam pengembangan software permainan, animasi, dan lain-lain. Dengan menambahkan sistem tersebut, maka software tadi dapat menjadi lebih hidup daripada sebelumnya.

Dalam membangun aplikasi tadi, tentunya harus pandai optimasi resource, algoritma, dan lain-lain. Hal ini penting terutama jumlah partikel yang diproses bisa mencapai ribuan hingga jutaan. Belum lagi perhitungan ramnya. Meskipun relatif kecil, tetapi jika yang digunakan dikalikan ribuan hingga jutaan, maka penggunaannya bisa signifikan.

Masalah ini tentunya juga bergantung pada hardware yang digunakan. Jika yang digunakan adalah komputer dengan ram yang relatif tinggi, tentunya hal ini kurang masalah. Akan tetapi, pada mobile umumnya, RAM yang dimiliki relatif kecil sehingga perlu diperhatikan banyak instance yang dibuat dan digunakan.

Umumnya untuk mengatasi hal tersebut, dilakukan object pooling, yaitu menggunakan ulang suatu instance daripada mengalokasikan atau menghancurkannya menjadi memori yang baru. Dengan menggunakan cara ini, diawal dibuat partikel sebanyak mungkin dan misalnya dimasukkan ke sebuah array. Ketika partikel tersebut digunakan dan mencapai akhir penggunaannya, maka tidak dialokasikan/dihancurkan melainkan

digunakan lagi dengan memindahkan posisinya atau memasukkannya ke array awal jika tidak digunakan.

Untuk menentukan jumlah maksimal partikel yang diawal, maka diperlukan suatu metode untuk menentukannya. Jika menggunakan cara crisp, akan butuh banyak percabangan. Itu pun prediksinya bisa salah dan lebih butuh banyak pemikiran dan percabangan lagi.

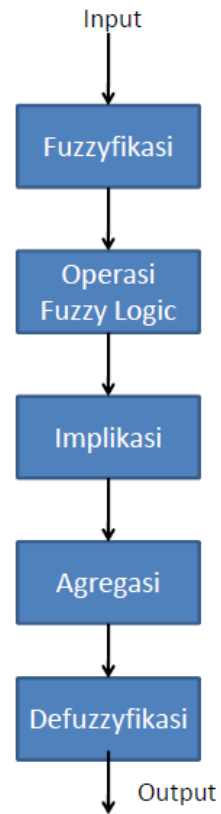
II. LANDASAN TEORI

Fuzzy Logic

- Logika fuzzy adalah suatu bentuk logika yang nilainya lebih ditekankan pada hampiran daripada nilai yang eksak dan pasti.
- Pada himpuna fuzzy, batas-batasnya cenderung kabur sedangkan pada himpunan crips, batas-batasnya dapat terlihat dengan tegas.
- Pada kaidah Fuzzy, terdapat beberapa tahanan dalam penginterpretasiannya, yaitu:
 1. Fuzzifikasi
Menentukan derajat keanggotaannya dari variable masukan
 2. Operasi fuzzy logic
Melakukan operasi-operasi logika fuzzy misalkan AND dan OR. Pada AND, nilai derajat yang dipilih adalah nilai minimum sedangkan pada OR, nilai derajat yang dipilih adalah yang maksimum.
 3. Implikasi
Menerapkan metode implikasi untuk menentukan bentuk akhir keluaran pada himpunan fuzzy. Metode yang saat ini digunakan adalah metode Mamdani dan metode Sugeno

- Terdapat Sistem Inferensi Fuzzy untuk menentukan output keluaran sebagai penarikan kesimpulan kaidah fuzzy. Proses-prosesnya adalah:
 1. Fuzzyfikasi
Memetakan nilai crisp ke dalam himpunan fuzzy dan menentukan derajat keanggotaannya di dalam himpunan fuzzy
 2. Operasi fuzzy logic
Melakukan operasi and, or, dan not yang nilai dihitung berdasarkan kaidah fuzzy
 3. Implikasi
Proses mendapatkan keluaran dari IF-THEN rule
 4. Agregasi
Jika terdapat lebih dari satu kaidah fuzzy yang dievaluasi, keluaran semua IF-THEN rule dikombinasikan menjadi sebuah fuzzy set tunggal.
 5. Defuzzyfikasi
Proses memetakan besaran dari himpunan fuzzy ke dalam bentuk nilai crisp.

Berikut diagramnya:



Sistem Partikel

Teknik dalam grafis komputer yang menggunakan banyak gambar berukuran sangat kecil sehingga dapat mensimulasikan suatu fenomena.

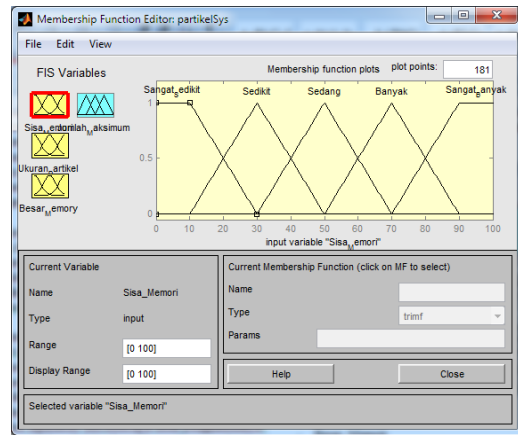
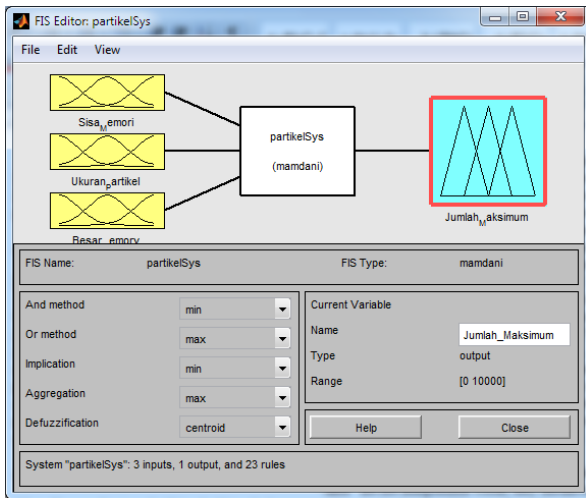
Contoh: Api, Air, Kembang api

RAM (Random Access Memory)

RAM adalah suatu bentuk penyimpanan data. Dengan hardware ini, data dapat diakses langsung secara acak.

III. METODE PEMBAHASAN

Metode yang digunakan untuk menentukan maksimum partikel adalah logika fuzzy. Berikut ini beberapa hal yang akan digunakan untuk estimasi nilai maksimum.



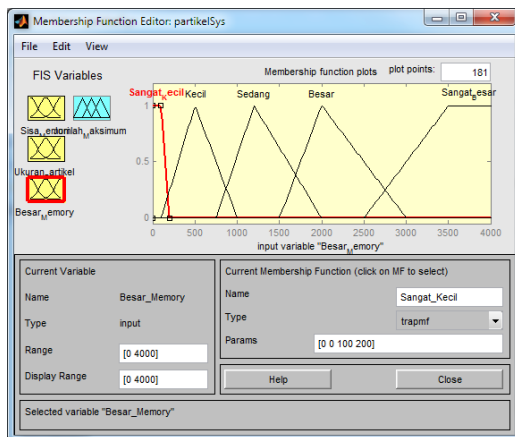
Asumsi range: 0 - 100 persen.

Nilai Fuzzy: Sangat Sedikit, Sedikit, Sedang, Banyak, Sangat Banyak

Variabel yang digunakan:

- Besar_Memori
Menunjukkan besar memori yang dimiliki oleh hardware. Untuk nilai 0, diasumsikan sebagai nilai memori yang sangat kecil.

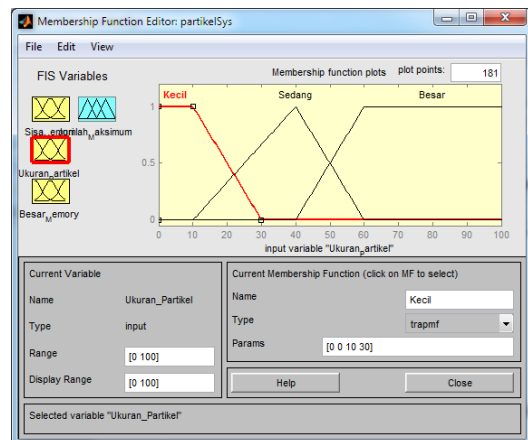
- Ukuran_Partikel
Besar memori yang digunakan untuk menjalankan partikel ketika aktif.



Asumsi range: 0 - 4000 KB.

Nilai Fuzzy: Sangat Kecil, Kecil, Sedang, Besar, Sangat Besar

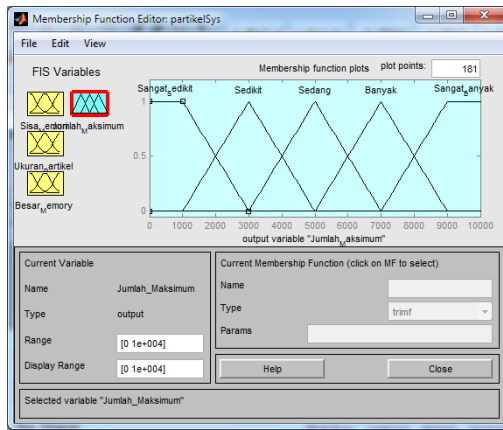
- Sisa_Memori
Besar_Memori dikurangi pemakaian seperti membuka aplikasi, sistem, dan lain-lain.



Asumsi range: 0 - 100 KB.

Nilai Fuzzy: Kecil, Sedang, Besar

- Output Jumlah_Maksimum:
Jumlah maksimum partikel yang boleh dibuat. Jika sistem yang berjalan menggunakan sama dengan atau melebihi batas jumlah maksimum, partikel tidak akan dibuat dan sebaiknya dilakukan optimasi dengan memanfaatkan partikel yang tidak akan digunakan atau kurang diperlukan.



Asumsi range: 0 – 10000 partikel

Nilai Fuzzy: Sangat Sedikit, Sedikit, Sedang, Banyak, Sangat Banyak

Rules:

1. If (Besar_Memori is Sangat_Kecil) then (Jumlah_Maksimum is Sangat_Sedikit) (1)
2. If (Sisa_Memori is Sangat_Sedikit) and (Besar_Memori is Kecil) then (Jumlah_Maksimum is Sangat_Sedikit) (1)
3. If (Sisa_Memori is Sedikit) and (Besar_Memori is Kecil) then (Jumlah_Maksimum is Sangat_Sedikit) (1)
4. If (Sisa_Memori is Sedang) and (Besar_Memori is Kecil) then (Jumlah_Maksimum is Sangat_Sedikit) (1)
5. If (Sisa_Memori is Banyak) and (Besar_Memori is Kecil) then (Jumlah_Maksimum is Sedikit) (1)
6. If (Sisa_Memori is Sangat_Banyak) and (Besar_Memori is Kecil) then (Jumlah_Maksimum is Sedikit) (1)
7. If (Sisa_Memori is Sangat_Sedikit) and (Besar_Memori is Sedang) then (Jumlah_Maksimum is Sangat_Sedikit) (1)
8. If (Sisa_Memori is Sedikit) and (Besar_Memori is Sedang) then (Jumlah_Maksimum is Sedikit) (1)
9. If (Sisa_Memori is Sedang) and (Besar_Memori is Sedang) then (Jumlah_Maksimum is Sedikit) (1)
10. If (Sisa_Memori is Banyak) and (Besar_Memori is Sedang) then (Jumlah_Maksimum is Sedang) (1)
11. If (Sisa_Memori is Sangat_Banyak) and (Besar_Memori is Sedang) then (Jumlah_Maksimum is Sedang) (1)
12. If (Sisa_Memori is Sangat_Sedikit) and (Besar_Memori is Besar) then (Jumlah_Maksimum is Sangat_Sedikit) (1)
13. If (Sisa_Memori is Sedikit) and (Besar_Memori is Besar) then (Jumlah_Maksimum is Sedikit) (1)
14. If (Sisa_Memori is Sedang) and (Besar_Memori is Besar) then (Jumlah_Maksimum is Sedang) (1)
15. If (Sisa_Memori is Banyak) and (Besar_Memori is Besar) then (Jumlah_Maksimum is Banyak) (1)
16. If (Sisa_Memori is Sangat_Banyak) and (Besar_Memori is Besar) then (Jumlah_Maksimum is Banyak) (1)
17. If (Sisa_Memori is Sangat_Sedikit) and (Besar_Memori is Sangat_Besar) then (Jumlah_Maksimum is Sedang) (1)
18. If (Sisa_Memori is Sedikit) and (Besar_Memori is Sangat_Besar) then (Jumlah_Maksimum is Sedang) (1)
19. If (Sisa_Memori is Sedang) and (Besar_Memori is Sangat_Besar) then (Jumlah_Maksimum is Banyak) (1)
20. If (Sisa_Memori is Banyak) and (Besar_Memori is Sangat_Besar) then (Jumlah_Maksimum is Sangat_Banyak) (1)
21. If (Ukuran_Partikel is Kecil) then (Jumlah_Maksimum is Sangat_Banyak) (1)
22. If (Ukuran_Partikel is Sedang) then (Jumlah_Maksimum is Sedang) (1)
23. If (Ukuran_Partikel is Besar) then (Jumlah_Maksimum is Sedikit) (1)

IV. HASIL EKSPERIMEN

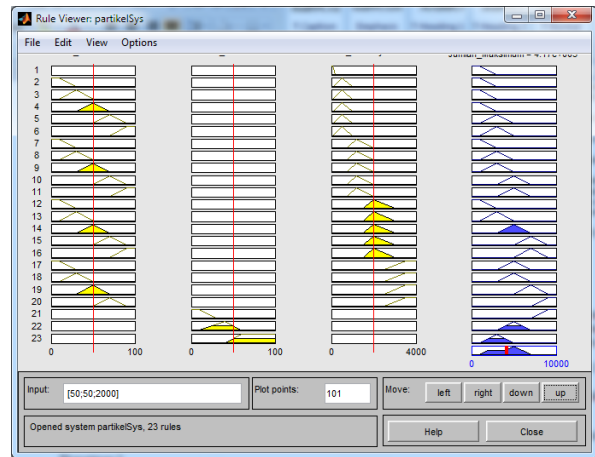
Metode yang digunakan untuk adalah metode Mamdani.
Urutan variabel sebagai berikut:

[Sisa_Memori; Ukuran_Partikel; Besar_Memori]

Eksperimen 1

Input: [50;50;2000]

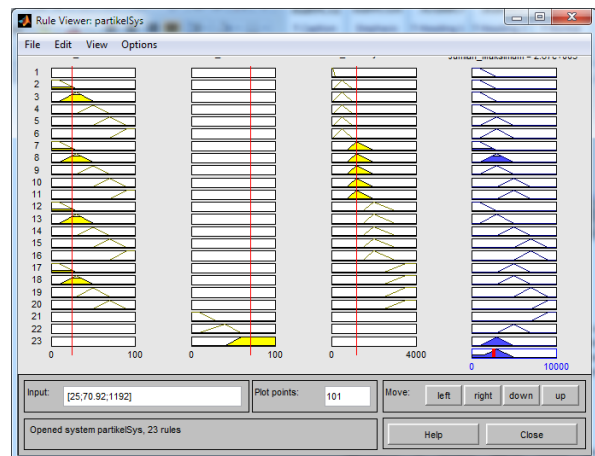
Makalah IF4058 Topik Khusus Informatika I – Sem. II Tahun 2011/2012



Output: 4170 partikel

Eksperimen 2

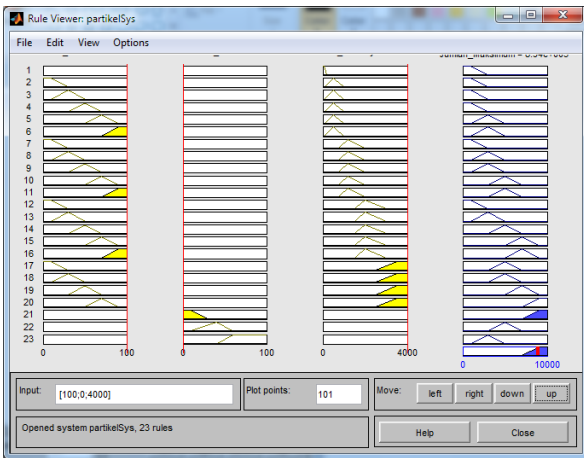
Input: [25;70.92;1192]



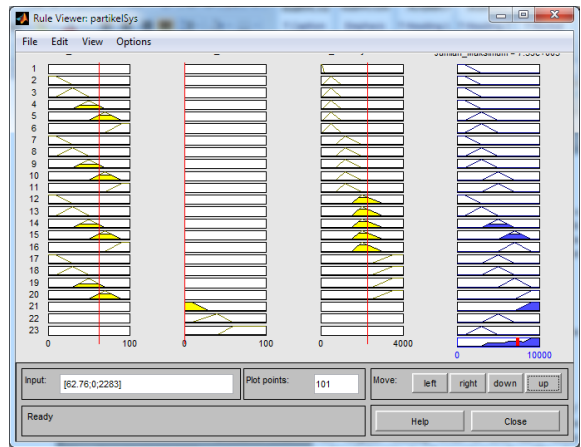
Output: 2670 partikel

Eksperimen 3

Input: [100;0;4000]



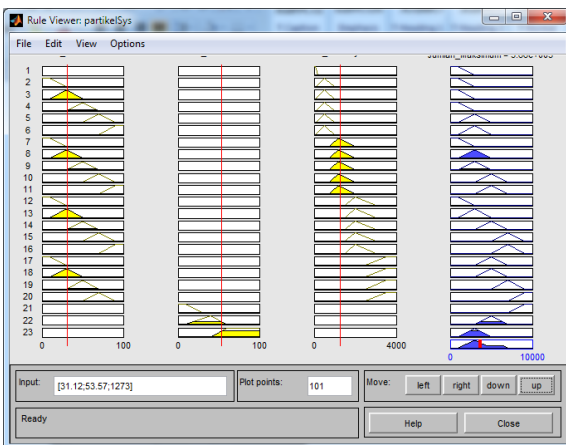
Output: 8940 partikel



Output: 7390 partikel

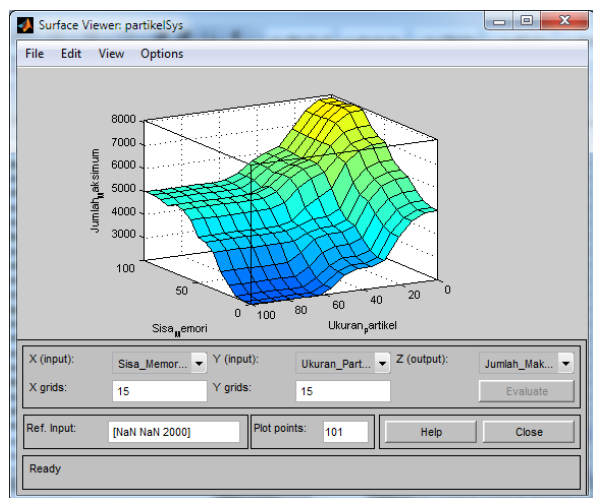
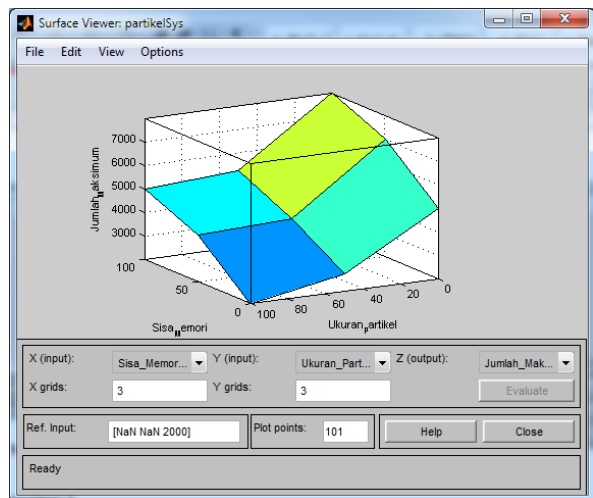
Eksperimen 4

Input: [31.12;53.57;1273]



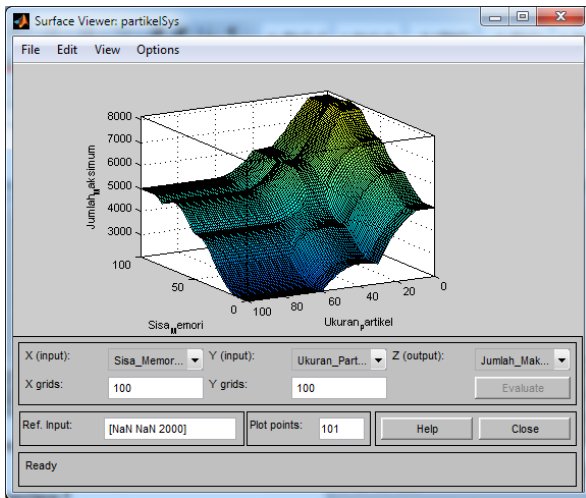
Output: 3660 partikel

Hasil Tampilan Surface



Eksperimen 5

Input: [62.76;0;2283]



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, 6 Mei 2013

V. ANALISIS

Dari hasil yang didapat, terlihat bahwa output keluaran untuk menentukan jumlah maksimum partikel berbeda berdasarkan input dan rulesnya. Dalam tampilan surface juga terlihat distribusi hasil untuk menentukan nilai keluaran.

Pada variable Ukuran_Partikel tadi, perubahan nilai output cukup drastis. Dari keluaran sekitar 8000 partikel ke kiri dan sekitar 3000 ke kanan.

Biolardi Yoshogi (13509035)

VI. KESIMPULAN

Berdasarkan hasil analisis, dapat ditarik kesimpulan bahwa untuk menentukan suatu nilai yang masih kabur batasan-batasannya, logika fuzzy dapat digunakan untuk menentukan nilainya karena nilai yang didapat tadi sangat bervariasi berdasarkan inputnya.

Ditambah lagi, bisa ditentukan nilai output berdasarkan nilai variable yang paling berpengaruh sehingga tiap variabel, bisa dilihat mana yang lebih penting dan paling berpengaruh.

Dengan implementasi fuzzy logic pada optimasi maksimum sistem partikel, diharapkan dapat lebih memudahkan mencari nilai maksimumnya sehingga aplikasi tetap berjalan lancar meskipun banyak instance partikel dalam suatu aplikasi.

REFERENSI

- [1] <http://informatika.stei.itb.ac.id/~rinaldi.munir/MetNum/2012-2013/metnum12-13.htm>
Waktu akses: 6 Mei 2013 10.00 AM
- [2] <http://dl.acm.org/citation.cfm?id=357320>
Waktu akses: 6 Mei 2013 11.00 AM