

Implementasi Graf Pohon dalam Algoritma Minimax untuk Artificial Intelligence

Praditya Raudi Avinanto 13514087
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514087@std.stei.itb.ac.id

Abstrak—Kemajuan teknologi pada era ini sangatlah pesat, baik dari teknologi komunikasi, informasi, dan mekanik. Semua itu terjadi karena kecerdasan manusia yang dari zaman ke zaman selalu meningkat. Salah satu ciptaan manusia yang sangat berpengaruh pada perubahan zaman ialah *games*. Sekarang manusia dapat membuat pemain virtual untuk bermain dengan pemain tersebut, pemain ini biasa disebut kecerdasan buatan atau *Artificial Intelligence* (AI).

Metode Minimax merupakan salah satu contoh dari metode yang digunakan dalam kecerdasan buatan, minimax adalah suatu algoritma yang menggunakan teknik pencarian Depth-First Search dengan kedalaman terbatas, yang menggunakan dasar peluang dan pohon untuk mendapatkan respon yang diinginkan. Beberapa *games* yang cocok untuk penerapan algoritma minimax adalah catur, tic tac toe, dsb yang dimainkan oleh dua orang atau lebih. Namun jumlah pemain harus dibatasi karena keterbatasan mesin untuk menjalankan algoritma minimax yang kebutuhan memorinya berbanding lurus dengan jumlah player. Permainan diatas dipilih karena sangat mudah menentukan ukuran kesuksesan atau kegagalan, sangat mungkin untuk dibandingkan dengan kemampuan manusia, dan mudah dimainkan.

Kata Kunci—Algoritma Minimax, Artificial Intelligence, Games.

I. PENDAHULUAN

Kemajuan teknologi pada era ini sangatlah pesat, baik dari teknologi komunikasi, informasi, dan mekanik. Semua itu terjadi karena kecerdasan manusia yang dari zaman ke zaman selalu meningkat. Seiring dengan penemuan – penemuan baru manusia belajar untuk mendapatkan hal – hal yang baru dan sifatnya nyaris tidak terbatas. Salah satu ciptaan manusia yang sangat berpengaruh pada perubahan zaman ialah *games*.

Dimulai dari *games* tradisional yang sederhana berkembang menjadi *games* digital, berkembang lagi dengan munculnya *games online* bahkan sekarang yang masih berkembang untuk disempurnakan yaitu *Virtual Reality*. Dikarenakan era yang maju akan teknologi informasi ini membuat *games* tidak harus dimainkan oleh lebih dari satu orang. Komputer yang dulunya hanya sebagai mesin hitung dan pengolah data, seiring perkembangan teknologinya saat ini komputer tidak hanya bertindak sebagai mesin yang disuruh namun dapat berfikir

dan memberi respon bagi pengguna. Sekarang manusia dapat membuat pemain virtual untuk bermain dengan pemain tersebut, pemain ini biasa disebut kecerdasan buatan atau *Artificial Intelligence* (AI).

Metode Minimax merupakan salah satu contoh dari metode yang digunakan dalam kecerdasan buatan, minimax adalah suatu algoritma yang menggunakan teknik pencarian Depth-First Search dengan kedalaman terbatas, yang menggunakan dasar peluang dan pohon untuk mendapatkan respon yang diinginkan dari suatu kondisi. Beberapa *games* yang cocok untuk penerapan algoritma minimax adalah catur, tic tac toe, checkers, dan lain sebagainya, yang dimainkan oleh dua orang atau lebih. Namun jumlah pemain harus dibatasi karena keterbatasan mesin untuk menjalankan algoritma minimax yang kebutuhan memorinya berbanding lurus dengan jumlah player. Alasan saya menyebutkan permainan diatas antara lain karena :

1. Sangat mudah menentukan ukuran kesuksesan atau kegagalan.
2. Sangat mungkin untuk dibandingkan dengan kemampuan manusia.
3. Mudah dimainkan dan asumsi sudah dikenal aturan mainnya oleh pembaca.

Diharapkan dengan terpenuhinya asumsi – asumsi diatas pembaca dapat memahami dasar terbentuknya Artificial Intelligence dan juga pengertian secara umum dari Algoritma Minimax.

II. DASAR TEORI

A. Artificial Intelligence

Kecerdasan Buatan (AI) adalah kecerdasan mesin dan cabang ilmu computer yang bertujuan untuk menciptakan komputer cerdas yang dapat berfikir layaknya manusia. Buku AI mendefinisikan bidang ini sebagai “*the study and design of intelligent agent*” dimana agent cerdas adalah system yang merasakan lingkungannya dan mengambil tindakan untuk memaksimalkan peluang yang sukses. John McCarthy menciptakan istilah ini pada tahun 1956 yang di definisikan sebagai “*ilmu dan teknik membuat mesin cerdas*”. AI sangat banyak digunakan pada aplikasi computer terutama pada aplikasi permainan. Pada two

player board games strategy, AI digunakan untuk mengatur strategi dan memutuskan langkah yang dapat mengimbangi permainan player sehingga player yang memainkan aplikasi ini seakan – akan bermain dengan player lain.

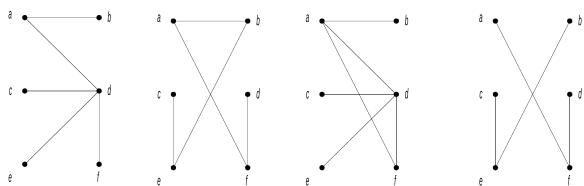
B. Pohon Graf

Defenisi Pohon (Tree) adalah graf terhubung yang tidak mengandung sirkuit. Karena merupakan graf terhubung maka pada pohon selalu terdapat path atau jalur yang menghubungkan kedua simpul di dalam pohon.

Pohon (*tree*) merupakan salah satu bentuk khusus dari struktur suatu graf. Misalkan A merupakan sebuah himpunan berhingga simpul (*vertex*) pada suatu graf G yang terhubung. Untuk setiap pasangan simpul di A dapat ditentukan suatu lintasan yang menghubungkan pasangan simpul tersebut. Untuk itu perlu diingat kembali bahwa :

- Suatu Graf G disebut terhubung apabila untuk setiap dua simpul dari graf G selalu terdapat jalur yang menghubungkan kedua simpul tersebut.
- Sirkuit atau cycle adalah suatu lintasan tertutup dengan derajat setiap simpul dua.

Suatu graf terhubung yang setiap pasangan simpulnya hanya dapat dihubungkan oleh suatu lintasan tertentu, maka graf tersebut dinamakan pohon (*tree*). Dengan kata lain, pohon (*tree*) merupakan graf tak-berarah yang terhubung dan tidak memiliki sirkuit.



Gambar 1. Graf Pohon dan bukan Pohon (Munir, 2006)

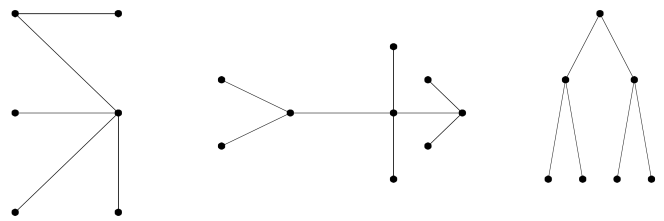
Contoh: Pohon (G_1) pohon (G_2) bukan pohon (G_3) bukan pohon (G_4)

Hutan (forest) merupakan kumpulan pohon yang saling lepas. Dengan kata lain, hutan merupakan graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung tersebut adalah pohon. Dengan kata lain kita dapat katakan (*forest*) adalah

- kumpulan pohon yang saling lepas, atau
- graf tidak terhubung yang tidak mengandung sirkuit.

Setiap komponen di dalam graf terhubung tersebut adalah pohon.

Pada gambar berikut adalah hutan yang terdiri dari 3 buah pohon



Gambar 2. Hutan yang terdiri dari tiga pohon (Munir, 2006)

Sifat-sifat Pohon

Misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n . Maka, semua pernyataan di bawah ini adalah ekuivalen:

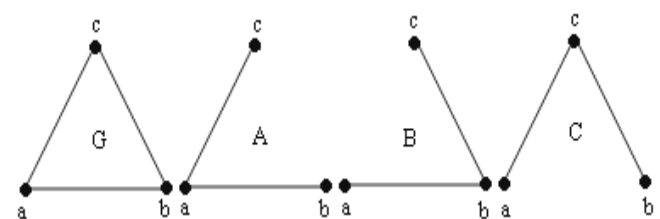
1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki $m = n - 1$ buah sisi.
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan. (jembatan adalah sisi yang bila dihapus menyebabkan graf terpecah menjadi dua komponen)

Pohon Merentang (spanning tree)

Definisi Misalkan G adalah sebuah graph. Sebuah pohon di G yang memuat semua titik G disebut *pohon rentang (spanning tree)* dari G.

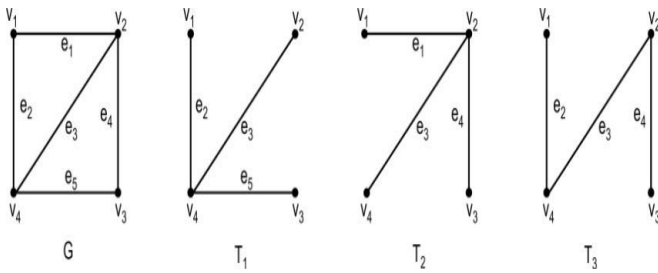
Contoh

Misalkan kita mempunyai graph G seperti pada gambar di bawah ini. Terdapat 3 pohon rentang dari graph G, yaitu graph A, B, dan C. Tampak jelas bahwa graph A, B, dan C masing-masing memuat semua simpul dari graph G serta mengandung sisi-sisi dari G demikian sehingga tidak terbentuk sikel.



Gambar 3. Graf dan Pohon Merentang (Sumber : www.meyouusblogshare.files.wordpress.com 6 Desember 2015)

Sebuah graph terhubung mungkin memuat lebih dari satu pohon rentang, seperti terlihat pada Gambar. Graph G memuat pohon rentang T1, T2, dan T3.



Gambar 4. Graf dan Pohon Merentang (Sumber : www.meyouusblogshare.files.wordpress.com 6 Desember 2015)

Jadi, pohon merentang:

- Pohon merentang dari graf terhubung adalah subgraf merentang yang berupa pohon.
- Pohon merentang diperoleh dengan memotong sirkuit di dalam graf.
- Setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang.
- Graf tak-terhubung dengan k komponen mempunyai k buah hutan merentang yang disebut hutan merentang (*spanning forest*).

Pohon Rentang Minimum

Graf terhubung-berbobot mungkin mempunyai lebih dari 1 pohon merentang. Pohon rentang yang berbobot minimum – dinamakan **pohon merentang minimum** (*minimum spanning tree*)

Dalam kehidupan nyata, salah satu contoh aplikasi *spanning tree* adalah menentukan rangkaian jalan dengan jarak total seminimum mungkin yang menghubungkan semua kota sehingga setiap kota tetap terhubung satu sama lain.

Dalam menentukan suatu *minimum spanning tree* dari suatu graf terhubung, kita dapat menentukannya dengan menggunakan dua cara yaitu algoritma Prim dan algoritma Kruskal, yang tidak akan dibahas pada paper ini.

Terminologi pada pohon berakar.

1. Anak dan Orangtua

Akar pada pohon berakar dikatakan sebagai orangtua dan simpul dari akar tersebut adalah anaknya

2. Lintasan (path)

Jalur pohon dari suatu simpul ke simpul lain

3. Saudara Kandung (sibling)

Dua buah simpul dikatakan sebagai saudara kandung apabila akar dari kedua simpul tersebut sama.

4. Upapohon (subtree)

Upapohon adalah bagian dari sebuah pohon utuh

5. Derajat

Derajat sebuah simpul adalah jumlah upapohon pada simpul tersebut.

6. Daun

Daun adalah simpul yang memiliki derajat nol

atau tidak mempunyai anak

7. Simpul dalam

Simpul dalam adalah simpul yang memiliki anak

8. Tinggi dan kedalaman

Tinggi maksimum sebuah pohon disebut juga tinggi atau kedalaman dari pohon tersebut

Pohon Keputusan

Pohon keputusan adalah pohon berakar yang simpul –

simpul dalamnya menuju suatu hail atau keputusan – keputusan yang dilalui.

Pohon keputusan pada gambar 5 halaman 4 adalah contoh pohon

yang membandingkan jalur pilihan langkah terbaik dari suatu player. Angka tersebut ialah nilai dari perhitungan minimax dan maximin. Player lingkaran akan lebih besar kemungkinan menang apa bila mengikuti panah biru. Panah merah ialah membandingkan pilihan terbaik dari seluruh peluang yang ada.

Dengan pohon keputusan, kita dapat melihat pertahapnya pemilihan keputusan yang dilakukan oleh AI namun pohon ini hanya untuk ilustrasi saja karena pada AI yang sebenarnya pilihan kemungkinannya sangatlah banyak dan akan sulit untuk melacaknya.

C. Algoritma Minimax

Algoritma minimax merupakan basis dari semua permainan berbasis AI seperti permainan catur misalnya. AI permainan catur tentunya sudah sangat terkenal dimana AI tersebut bahkan dapat mengalahkan juara dunia sekalipun. Pada algoritma minimax, pengecekan akan seluruh kemungkinan yang ada sampai akhir permainan dilakukan. Pengecekan tersebut akan menghasilkan pohon permainan yang berisi semua kemungkinan tersebut. Tentunya dibutuhkan resource yang berskala besar untuk menangani komputasi pencarian pohon solusi tersebut terhubung kombinasi kemungkinan untuk sebuah permainan catur pada setiap gerakannya sangat banyak sekali.

Keuntungan yang didapat dengan menggunakan algoritma minimax yaitu algoritma minimax mampu menganalisis segala kemungkinan posisi permainan untuk menghasilkan keputusan yang terbaik karena algoritma minimax ini bekerja secara rekursif dengan mencari langkah yang akan membuat lawan mengalami kerugian minimum. Semua strategi lawan akan dihitung dengan algoritma yang sama dan seterusnya. Ini berarti, pada langkah pertama komputer akan menganalisis seluruh pohon permainan. Dan untuk setiap langkahnya, komputer akan memilih langkah yang paling membuat lawan mendapatkan keuntungan minimum, dan yang paling membuat komputer itu sendiri mendapatkan keuntungan maksimum.

Dalam penentuan keputusan tersebut dibutuhkan suatu nilai yang merepresentasikan kerugian atau keuntungan yang akan diperoleh jika langkah tersebut dipilih. Untuk itulah disini digunakan sebuah fungsi heuristik untuk mengevaluasi nilai sebagai nilai yang

merepresentasikan hasil permainan yang akan terjadi jika langkah tersebut dipilih. Biasanya pada permainan tic tac toe ini digunakan nilai 1,0,-1 untuk mewakili hasil akhir permainan berupa menang, seri, dan kalah. Dari nilai-nilai heuristic inilah komputer akan menentukan simpul mana dari pohon permainan yang akan dipilih, tentunya simpul yang akan dipilih tersebut adalah simpul dengan nilai heuristic yang akan menuntun permainan ke hasil akhir yang menguntungkan bagi komputer.

III. PENGGUNAAN MINIMAX DALAM PEMBUATAN ARTIFICIAL INTELLIGENCE

Dalam Algoritma Minimax terdapat dua nilai :

- Maximin Value

Nilai terbesar yang dapat di dapat seorang pemain tanpa harus mengetahui langkah lawan.

Nilai tersebut dapat dicari menggunakan rumus :

$$v_i = \max_{a_i} \min_{a_{-i}} v_i(a_i, a_{-i})$$

i = pemain yang ingin diuntungkan

-i = pemain selain i

a_i = aksi yang dilakukan i

a_{-i} = aksi yang dilakukan selain player i

V_i = nilai untuk player i

Cara kerjanya ialah kita cek seluruh kemungkinan nilai dari maximin, kita pilih nilai aksi yang akan memberikan nilai minimum terbesar untuk segala kombinasi dengan langkah pemain lain.

- Minimax Value

Nilai minimum yang pemain bisa berikan pada pemain lain tanpa harus mengetahui aksinya.

$$\bar{v}_i = \min_{a_{-i}} \max_{a_i} v_i(a_i, a_{-i})$$

	L	R
T	3,1	2,-20
M	5,0	-10,1
B	-100,2	4,4

Misalkan pada kasus tabel diatas, angka pertama pada tiap sel adalah nilai untuk pemain baris, dan angka kedua ialah nilai untuk pemain kolom. Untuk nilai Maximin nya :

Pemain baris akan memilih langkah T karena angka terburuk yang dia dapat ialah dua jadi maximin value pemain baris adalah 2.

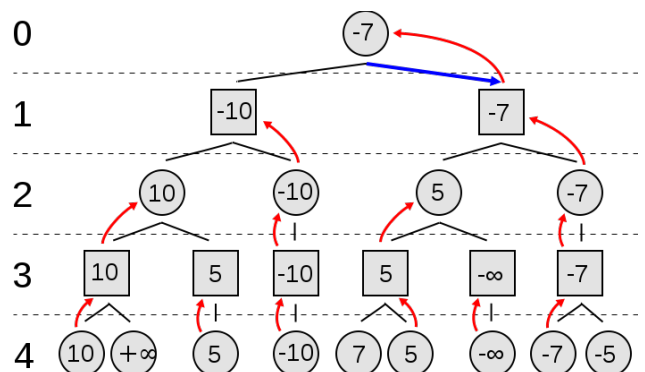
Sedangkan pemain kolom langkah L akan memberikan maximin value 0.

Untuk Minimax Value pada pemain baris ialah 4 karena kemungkinan terbesar yang dia dapat bila pemain kolom memilih R ialah 5 sedangkan L ialah 4. kita pilih yang paling kecil. Untuk pemain kolom Minimax Valuenya ialah 1.

Contoh Penerapan Pohon pada Algoritma Minimax

Misalkan permainan yang dimainkan hanya memiliki maksimal dua kemungkinan langkah per pemain setiap giliran. Algoritma menghasilkan pohon di sebelah kanan, di mana lingkaran mewakili pergerakan dari pemain yang menjalankan algoritma (maximizing player), dan kotak mewakili pergerakan dari lawan (minimizing player). Karena keterbatasan sumber daya komputasi, seperti dijelaskan di atas, pohon terbatas pada 4 tahap pergerakan saja.

Algoritma mengevaluasi setiap node daun menggunakan fungsi evaluasi heuristic, memperoleh nilai yang ditampilkan. Pergerakan yang menyebabkan maximize player menang ialah ditandai dengan infinity positif, sementara pergerakan yang menyebabkan kemenangan dari minimize player ditandai dengan infinity negatif. Pada tingkat 3, algoritma akan memilih, untuk setiap node, yang terkecil dari nilai node anak, dan menetapkan ke node yang sama (misalnya node di sebelah kiri akan memilih minimum antara "10" dan "+ ∞", oleh karena itu ia akan memilih "10" untuk dirinya sendiri). Langkah berikutnya, di tingkat 2, terdiri dari memilih untuk setiap node yang terbesar dari nilai-nilai node anak. Sekali lagi, nilai-nilai dipilih untuk setiap node induk. Algoritma terus mengevaluasi nilai-nilai maksimum dan minimum dari node anak bergantian sampai mencapai simpul akar, di mana ia memilih pergerakan dengan nilai terbesar (diwakili pada gambar dengan panah biru). Ini adalah langkah yang pemain harus buat untuk meminimalkan kerugian maksimum yang mungkin.



Gambar 5. Pohon Keputusan Maximin Value Artificial Intelligence

(Sumber : <https://en.wikipedia.org/wiki/Minimax> 6 Desember 2015)

Berikut merupakan contoh kode sederhana atau dasar dari Algoritma Minimax pada tingkat tertentu

```
function minimax (node, depth, maximizingPlayer)
  if depth = 0 or node is a terminal node
    return the heuristic value of node
  if maximizingPlayer
    bestValue := -∞
    for each child of node
      val := minimax(child, depth - 1, FALSE)
      bestValue := max(bestValue, val)
    return bestValue
  else
    bestValue := +∞
```

```

for each child of node
    val := minimax(child, depth - 1, TRUE)
    bestValue := min(bestValue, val)
return bestValue

```

```

(* Initial call for maximizing player *)
minimax(origin, depth, TRUE)

```

Namun jangan menganggap algoritma ini membuat AI tidak bisa dikalahkan, karena dalam sebuah permainan masih ada faktor – faktor lain yang tidak masuk ke dalam perhitungan AI, seperti keberuntungan dalam bentuk giliran atau yang lainnya. Maka tentu ada kalanya AI menemui ketidakpastian dalam memberikan respon. Di saat itu AI akan mencari yang resikonya paling kecil dengan rumus dibawah ini.

Minimax Dalam Teori Keputusan Statistik

Dalam teori keputusan statistik klasik, kita memiliki estimator δ yang digunakan untuk memperkirakan parameter $\theta \in \Theta$. Kami juga menganggap fungsi risiko $R(\theta, \delta)$, biasanya ditetapkan sebagai integral dari fungsi kerugian. Dalam kerangka ini, Δ disebut minimax jika memenuhi

$$\sup R(\theta, \Delta) = \inf \sup R(\theta, \delta)$$

Kriteria alternatif dalam rangka keputusan teoritis adalah estimator Bayes dengan adanya sebelum distribusi Π . Estimator adalah Bayes jika meminimalkan risiko rata-rata.

$$\int_{\Theta} R(\theta, \delta) d\Pi(\theta).$$

IV. KESIMPULAN

Kesimpulan dari makalah ini adalah :

1. Teori graf dapat digunakan sebagai salah satu metode penentuan respon Artificial Intelligence.
2. Graf yang digunakan ialah graf Pohon, karena dapat menelusuri tiap tahap dengan semua kemungkinan yang ada.
3. Masih diperlukan algoritma yang lebih efisien sehingga tidak menguras terlalu banyak memori seperti metode diatas untuk kemajuan pembuatan Artificial Intelligence.

V. ACKNOWLEDGMENT

Pertama, penulis mengucapkan puji syukur kepada Tuhan

Yang Maha Esa atas berkat dan rahmat-Nya sehingga penulis dapat

menyelesaikan tugas makalah IF2120 Matematika Diskrit ini.

Penulis juga ingin berterima kasih kepada Dosen IF 2120 Tn. Rinaldi Munir dan Ny. Harlili yang sudah menjadi mengajarkan setiap materi dengan jelas sehingga

penulis dapat menyelesaikan tugas tanpa kesulitan berarti. Terima kasi juga pada teman-teman yang telah memberikan masukan pada makalah ini.

Semoga ke depannya teori yang saya bahas di tugas ini dapat diimplementasikan pada praktik pembuatan Artificial Intelligence di masa mendatang.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi, "Diktat Kuliah IF2120 Matematika Diskrit", Program Studi Teknik Informatika ITB, hal. IX-1 – IX-33
- [2] Michael Maschler, Eilon Solan & Shmuel Zamir (2013). *Game Theory*. Cambridge University Press. pp. 176–180.
- [3] Russel, Stuart J; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach (2nd ed.)*, Upper Saddle River, New Jersey: Prentice Hall, pp. 163–171
- [4] Hazewinkel, Michiel, ed. (2001), "Minimax principle", *Encyclopedia of Mathematics*, Springer.,
- [5] Osborne, Martin J., and Ariel Rubinstein. *A Course in Game Theory*. Cambridge, MA: MIT, 1994. Print.
- [6] Von Neumann, J: *Zur Theorie der Gesellschaftsspiele Math. Annalen*. **100** (1928) 295-320.
- [7] John L Casti (1996). *Five golden rulse: great theories of 20th-century mathematics – and why they matter*. New York: Wiley-Interscience. p. 19.

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, 8 Desember 2015



Praditya Raudi Avinanto 13514087