

# ANALISIS DAN PERANCANGAN TEKNOLOGI PADA ROBOT SEPAK BOLA

Tony

Fakultas Teknologi Informasi, Universitas Tarumanagara, Jl. Letjen S. Parman No. 1, Jakarta Barat,  
11440, Indonesia

E-mail: tony.b@fti.utara.org

## Abstrak

Pertandingan sepak bola antar robot merupakan salah satu tantangan dalam dunia robotika yang diselenggarakan untuk dapat lebih mengembangkan robotika dan kecerdasan buatan serta sebagai ajang bertukar ilmu bagi para peneliti di seluruh dunia. Hal ini mendorong penulis merancang sebuah strategi untuk pertandingan sepak bola antar robot. Strategi dibuat dengan menggunakan konsep koordinat untuk merepresentasikan posisi robot dalam lapangan. Kemudian strategi diuji dan dianalisis untuk menentukan kinerja strategi dalam berbagai situasi.

**Kata Kunci:** *robotika, sepakbola, strategi*

## Abstract

Inter-robot soccer game is one of the challenges in the world of robotics that is held to develop robotics and artificial intelligence and as well as a forum for researchers to exchange knowledge across the world. This encouraged the authors to design a strategy for inter-robot soccer game. Strategies are made using the concept of coordinates to represent the robot position in the field. Then the strategy is tested and analyzed to determine the performance of strategies in different situations.

**Keywords:** *robotics, soccer, strategy*

## 1. Pendahuluan

Implementasi robot selama ini telah dapat membantu manusia di berbagai bidang terutama bidang Industri. AI (*Artificial Intelligence*) memegang peran penting dalam dunia robotika karena memungkinkan robot dapat bergerak secara otomatis hanya dengan perintah sederhana. Untuk mengembangkan dunia robotika lebih jauh, para peneliti mencoba membuat robot untuk melakukan berbagai macam olahraga dan permainan yang dibuat manusia. Harapannya, agar robot dapat bertindak seperti manusia dan suatu saat dapat digunakan dalam lingkup yang lebih luas.

Salah satu cabang olahraga yang dimasuki dunia robotika adalah sepak bola. Olahraga yang dimainkan di seluruh dunia ini memiliki unsur kerja sama tim yang tinggi, strategi yang kompleks, dan state permainan yang beragam. Konsep permainan sepak bola yang dimainkan robot dimulai dari gagasan Prof. Alan Mackworth

pada 1992, yang menerapkan konsep multi robot yang bekerja sama dalam tim secara kompetitif dan kooperatif.

Pada Juni 1993, sekelompok peneliti di Jepang mengadakan turnamen sepak bola untuk robot. Setelah satu bulan diumumkan, banyak kalangan peneliti di luar Jepang yang tertarik untuk mengikuti turnamen ini. Pada akhirnya, proyek tersebut diberi nama Robot World Cup Initiative atau disingkat RoboCup dan menjadi pemicu bagi pengembangan penelitian sepak bola robot di dunia.

Ide robot yang bermain bola (*robot soccer*) pertama kali dikemukakan pada tahun 1992 oleh Prof. Alan Mackworth dari University of British Columbia, Kanada, dalam tulisannya yang berjudul "On Seeing Robot" [1]. Sementara itu, para peneliti dari Jepang mengadakan *Workshop on Grand Challenges in Artificial Intelligence* pada bulan Oktober 1992 di Tokyo. *Workshop* tersebut membahas mengenai penggunaan permainan sepak bola untuk mempromosikan

sains dan teknologi. Pada bulan Juni 1993 sekelompok peneliti yang terdiri dari Minoru Asada, Yasuo Kuniyoshi, dan Hiroki Kitano, mengadakan kompetisi robot yang disebut *Robot J-League* (*J-League* adalah liga sepak bola profesional Jepang). Dalam sebulan, para peneliti dari luar Jepang meminta kompetisi tersebut diperluas menjadi skala internasional. Nama kompetisi akhirnya diganti menjadi *Robot World Cup* atau disingkat menjadi *RoboCup* [2].

Seiring dengan itu, penelitian mengenai *robot soccer* mulai berkembang. Itsuki Noda dari *Electro Technical Laboratory* (ETL) mulai mengembangkan simulator untuk permainan sepak bola. Simulator ini nantinya akan menjadi *server* resmi dari *RoboCup Simulation*. Sementara itu, Prof. Minoru Asada (Osaka University) dan Prof. Manuela Veloso beserta Peter Stone (Carnegie Mellon University) telah mengembangkan robot sepak bola. Tanpa dukungan dari penelitian di bidang *robot soccer*, kompetisi *RoboCup* mustahil dapat diselenggarakan.

Pada bulan September 1993 pengumuman resmi dan *draft* peraturan *RoboCup* dibuat. Sementara itu, tim yang dipimpin oleh Itsuki Noda dari ETL mengumumkan *Soccer Server* Versi 0, simulator sistem pertama, kemudian diikuti *Soccer Server* Versi 1.0 yang didistribusikan via *web*. Demonstrasi *server* tersebut pertama kali dilakukan pada IJCAI-95 (*International Joint Conference on Artificial Intelligence*). Konferensi yang diadakan di Montreal, Kanada menghasilkan keputusan bahwa *Robot World Cup Soccer Games and Conferences* akan diadakan bersamaan dengan IJCAI-97 di Nagoya, Jepang. Selain itu, *Pre-RoboCup* akan diadakan pada tahun 1996 sebagai persiapan untuk mengadakan *RoboCup*. Tujuannya untuk memberikan waktu persiapan dan pengembangan robot simulasi yang cukup kepada para peneliti di seluruh dunia. *Pre-RoboCup-96* diselenggarakan bersamaan dengan *International Conference on Intelligence Robotics and Systems* (IROS-96) di Osaka, Jepang pada tanggal 4 – 8 November 1996. Pada ajang tersebut delapan tim bertanding dalam liga simulasi dan diadakan demonstrasi robot untuk *middle size league*. Kompetisi ini adalah kompetisi pertama yang menggunakan sepak bola untuk mempromosikan riset dan pendidikan.

Akhirnya, pertandingan dan konferensi *RoboCup* yang resmi diadakan pada tahun 1997 dan berjalan dengan sukses. Jumlah peserta sebanyak 40 tim dan jumlah penonton lebih dari 5000. Harapannya di masa mendatang lebih dari seratus tim ikut ambil bagian pada kompetisi

tersebut. Adapun tempat penyelenggaraan dan jumlah peserta *RoboCup* dari tahun ke tahun dapat dilihat pada tabel I.

TABEL I  
PENYELENGGARAAN *ROBOCUP* [3]

Tempat	Jumlah tim	Jumlah negara
<i>RoboCup</i> 2010 Singapore	Belum	Belum
<i>RoboCup</i> 2009 Graz - Austria	407	43
<i>RoboCup</i> 2008 Suzhou - China	373	35
<i>RoboCup</i> 2007 Atlanta - USA	321	39
<i>RoboCup</i> 2006 Bremen - Germany	440	35
<i>RoboCup</i> 2005 Osaka - Japan	419	35
<i>RoboCup</i> 2004 Lisbon - Portugal	345	37
<i>RoboCup</i> 2003 Padua - Italy	238	35
<i>RoboCup</i> 2002 Fukuoka/Busan - Japan/Korea	188	29
<i>RoboCup</i> 2001 Seattle - USA	141	22
<i>RoboCup</i> 2000 Melbourne - Australia	110	19
<i>RoboCup</i> 1999 Stockholm - Sweden	85	23
<i>RoboCup</i> 1998 Paris - France	63	19
<i>RoboCup</i> 1997 Nagoya - Japan	38	11
<i>Pre-RoboCup-96</i> - Osaka - Japan	8	-

Tujuan dari *RoboCup* [4] adalah pada tahun 2050 nanti ada sebuah tim yang terdiri dari *autonomous humanoid robot soccer* yang akan memenangkan pertandingan melawan pemenang dari Piala Dunia terakhir dengan mengikuti aturan FIFA (*Federation International Football Association*). Sekarang *RoboCup* terdiri dari tiga domain utama, yakni *RoboCup Soccer*, *RoboCup Rescue*, dan *RoboCup Junior*.

*RoboCup Soccer* dibagi lagi menjadi *Simulation League*, *Small Size Robot League*, *Middle Size Robot*, *Standard Platform League*, dan *Humanoid League*. *Simulation League* dibagi lagi menjadi *2DSimulation League*, *3DSimulation League*, dan *3Ddevelopment*. *Humanoid League* dibagi lagi menjadi *Kid Size* (30-60 cm height) dan *Teen Size* (100-160 cm height).

*RoboCup Rescue* dibagi lagi menjadi *Rescue Simulation League* dan *Rescue Robot League*. Kemudian *RoboCup Junior* dibagi lagi menjadi *Soccer Challenge*, *Dance Challenge*, dan *Rescue Challenge*.

Adapun yang menjadi fokus penelitian dalam tulisan ini adalah *RoboCup 2D Simulation League*. Simulasi robot sepak bola dalam *RoboCup* menggunakan simulator yang disebut *soccer server*. Simulator ini merupakan sebuah sistem simulasi sepak bola yang dibuat oleh Itsuki Noda pada bulan September 1993. Semua pertandingan sepak bola divisualisasikan dengan menunjukkan lapangan simulator pada layar monitor komputer.

*Soccer server* adalah sebuah sistem yang membuat agen atau robot dapat bermain sepak bola melawan robot yang lain [5]. Robot yang digunakan sebenarnya terdiri dari barisan program yang dapat dibuat dengan menggunakan berbagai bahasa pemrograman seperti Java, C/C++, dan sebagainya. Pertandingan sepak bola antar robot dalam *soccer server* dilakukan dengan cara *client/server*. *Server* menyediakan lapangan virtual dan mensimulasikan semua pergerakan bola dan robot. Sementara setiap *client* mengatur gerakan satu robot. Komunikasi antara *server* dan *client* dilakukan via socket UDP/IP (*User Datagram Protocol/Internet Protocol*).

*Soccer server* [6] terdiri dari dua program yaitu *soccer server* dan *soccer monitor*. *Soccer server* adalah program *server* yang mensimulasikan gerakan bola dan robot, komunikasi dengan *client*, dan mengatur pertandingan sesuai dengan aturan yang berlaku. *Soccer monitor* adalah program yang memvisualisasikan lapangan virtual dari *soccer server* pada layar monitor. Beberapa program *soccer monitor* dapat dihubungkan dengan satu *soccer server* sehingga pertandingan sepak bola dapat dilihat pada lebih dari satu layar monitor. Gambar 1 menunjukkan skema *soccer server*.

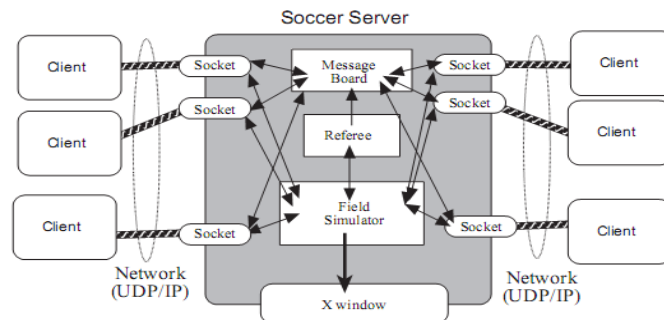
Satu *client* berhubungan dengan *soccer server* dengan menggunakan socket UDP. Dengan socket tersebut, *client* dapat mengirimkan perintah untuk mengatur robot dan menerima informasi dari sensor robot. *Program client* merupakan otak dari robot pemain sepak bola. *Client* akan menerima informasi dari sensor visual dan mengirim perintah kontrol ke *server*. Satu *client*

hanya dapat mengontrol satu robot pemain. Setiap tim terdiri dari sebelas *client* yang bertugas sebagai pemain.

Dalam pembelajaran strategi *robot soccer*, beberapa tim terlihat memiliki sedikit persamaan dari segi algoritma dan strategi yang digunakan. Strategi pertama yang didapat dari sumber-sumber yang ada, yaitu mengenai pembagian lapangan dan pembagian tugas robot sesuai lapangannya. Menurut strategi ini, lapangan dibagi menjadi beberapa area dan beberapa robot diberi satu area untuk mengoper atau menendang bola. Cara ini tidak hanya akan menentukan robot mana yang paling dekat dengan bola untuk mendekati, menembak, atau mengoper bola, namun juga mencegah robot yang berada jauh dari bola untuk membuang-buang waktu mengejar bola.

Strategi kedua adalah strategi pengoperan bola (*passing*) dan evaluasi pengoperan bola. Jika *shooting* bola ke gawang dianggap tidak dapat dilakukan, maka pengoperan akan dilakukan. Untuk melakukan pengoperan, mula-mula dilakukan evaluasi pengoperan untuk menentukan kemungkinan keberhasilan pengoperan yang akan dilakukan. Kedua strategi tersebut akan diimplementasikan secara terintegrasi dalam satu sistem bersama dengan algoritma penentuan lokasi robot dan algoritma untuk menentukan jalannya robot berdasarkan hasil dari strategi yang dibuat.

Dalam tulisan ini, peneliti membahas dua masalah utama terkait dengan penelitian robot sepak bola dan solusinya. Masalah pertama berkenaan dengan perancangan strategi robot soccer yang dapat mencakup semua kemungkinan posisi robot di lapangan. Permasalahan kedua terkait dengan pengaturan robot-robot sehingga robot dapat bekerjasama dengan robot lain dalam satu tim. Untuk kedua permasalahan tersebut, penulis mengusulkan sebuah pendekatan, yakni algoritma pergerakan pemain yang menerapkan kecerdasan buatan dalam pertandingan sepak bola antar robot sehingga robot dapat bertanding secara otomatis tanpa perintah langsung dari manusia. Algoritma tersebut diimplementasikan dalam bentuk simulasi menggunakan *soccer server*.

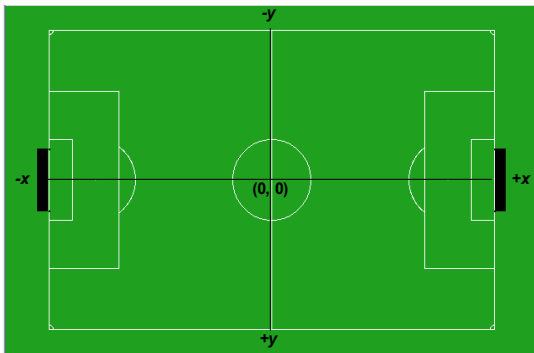


Gambar 1. Soccer server.

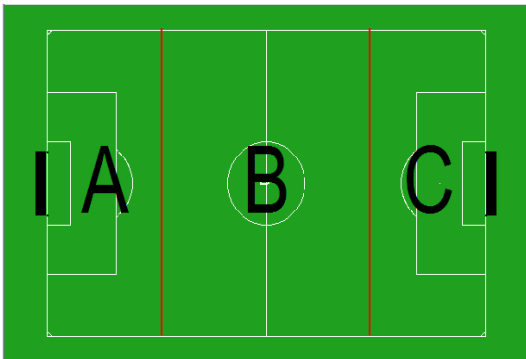
## 2. Metodologi

Prinsip kerja dari algoritma ini adalah membagi lapangan untuk beberapa robot pemain. Lapangan yang ditugaskan kepada robot dinamakan area [7]. Area ini dapat menentukan ruang gerak robot, nomor robot yang berada di lapangan, dan robot mana yang harus mengejar bola.

Struktur data yang digunakan pada algoritma ini adalah sebuah *array* yang berisi koordinat robot sendiri, robot musuh, bola, dan arah menghadap robot sendiri. Titik koordinat (0,0) berada di tengah lapangan. Letak koordinat sumbu  $x$  positif dan  $x$  negatif sama dengan koordinat Cartesius, yang berbeda adalah koordinat sumbu  $y$ . Gambar 2 menunjukkan pembagian koordinat pada lapangan sepak bola yang digunakan dalam simulasi. Dengan menggunakan pemodelan lapangan dalam bentuk koordinat, penghitungan jarak antar robot, sudut tembakan, dan analisis posisi musuh dapat dihitung dengan menggunakan persamaan garis dalam bidang Cartesius.



Gambar 2. Pembagian koordinat pada lapangan.



Gambar 3. Pembagian area lapangan.

Daerah area robot adalah daerah di lapangan yang dibatasi oleh beberapa garis lurus. Gambar 3 menunjukkan contoh pembagian area

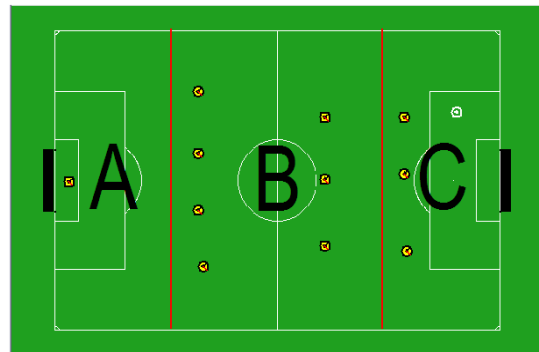
lapangan sepak bola. Lapangan dibagi menjadi tiga area yaitu area A, B, dan C.

Pembagian lapangan ini akan digunakan dalam mengatur pergerakan pemain. Apabila bola berada di area C, letak pemain seperti ditunjukkan pada gambar 4. Apabila bola berada di area B, letak pemain seperti ditunjukkan pada gambar 5. Apabila bola berada di area C, letak pemain seperti ditunjukkan pada gambar 6.

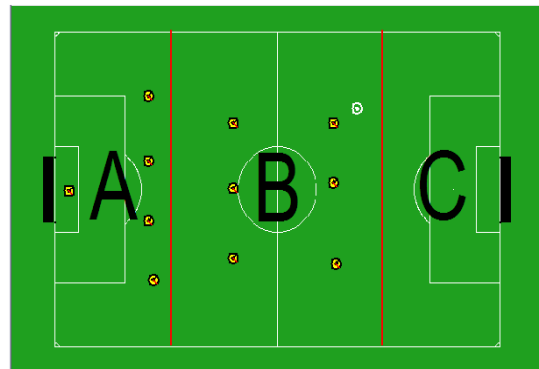
Area tersebut dapat saling lepas atau saling tindih untuk mencegah saling rebut bola antar pemain sendiri dan robot berkumpul di suatu bagian lapangan. Jika hanya satu robot yang mengejar bola, maka robot lain dapat bersiaga untuk menjaga daerah belakang lapangan untuk bersiap menerima operan bola.

Untuk efisiensi, setiap robot memiliki posisi standar. Robot akan menuju ke posisi ini jika tidak mengejar bola (robot aktif), kecuali penjaga gawang yang harus selalu bersiaga terhadap bola yang datang. Posisi standar ini berada di tengah lapangan.

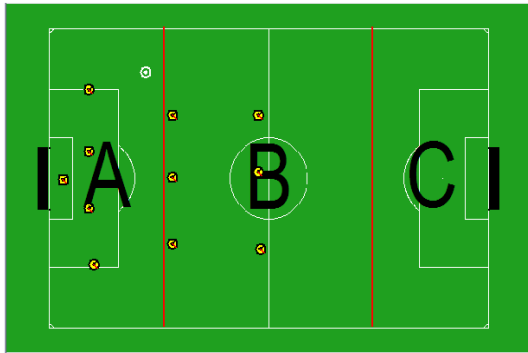
Dalam algoritma ini, penulis menggunakan area yang mengalami saling tindih, namun tidak saling lepas. Hal ini diperlukan untuk mengantisipasi apabila robot secara tidak sengaja keluar dari daerahnya.



Gambar 4. Bola berada pada area C.



Gambar 5. Bola berada pada area B.

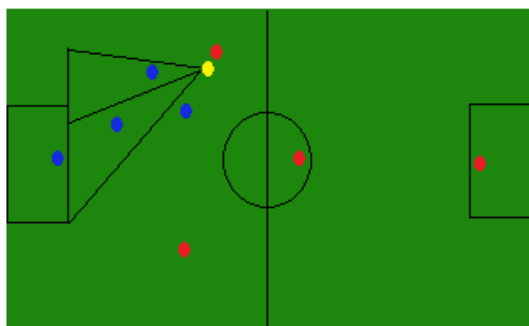


Gambar 6. Bola berada pada area A.

Daerah tembak adalah daerah sekeliling bola yang dianggap cukup dekat dengan bola untuk bersiap melakukan tembakan ke arah gawang. Berikut ini adalah penetapan apakah sebuah robot harus berjalan ke daerah tembak, menembak bola ke gawang, atau mengoper ke robot lain dalam timnya.

Jika robot berada jauh dari bola (selisih koordinat  $x$  dan  $y$  dari robot dan bola tidak masuk dalam *range* yang ditentukan), maka robot akan berjalan ke arah bola. Jika robot berada dalam daerah tembak (selisih koordinat  $x$  dan  $y$  dari robot dan bola masuk dalam *range* yang ditentukan). Dalam hal ini jika jumlah musuh lebih sedikit, maka robot diberikan perintah untuk menembak ke gawang. Namun jika jumlah musuh lebih banyak, maka robot mengoper bola ke robot lain.

Untuk menghitung jumlah musuh yang berada di antara robot dengan gawang, dibutuhkan *input* berupa koordinat robot dan koordinat sasaran, serta koordinat setiap robot musuh. Tidak semua musuh yang dihitung, hanya musuh yang mungkin dapat menahan laju bola yang harus dihitung. Gambar 7 menunjukkan pengecekan jumlah robot musuh.



Gambar 7. Pengecekan jumlah musuh.

Untuk perhitungan jumlah robot musuh tersebut, mula-mula titik sasaran diputar sejauh  $\alpha$  dan  $-\alpha$  dengan titik pusat posisi bola.

Implementasinya dengan menggunakan persamaan 1 dan persamaan 2.

$$(x' - x_1) = (x - x_1) \cos \theta - (y - y_1) \sin \theta \quad (1)$$

$$(y' - y_1) = -(x - x_1) \sin \theta + (y - y_1) \cos \theta \quad (2)$$

Selanjutnya dihitung gradien garis antara masing-masing titik dengan koordinat bola dengan menggunakan persamaan 3. Setelah gradien garis diketahui, maka persamaan garis diperoleh dengan pola yang ditunjukkan oleh persamaan 4. Variabel  $c$  untuk kedua garis dapat diketahui dengan memasukkan kedua titik ke persamaan masing-masing.

$$m = \frac{y-y_1}{x-x_1} \quad (3)$$

$$y = mx + c \quad (4)$$

Menggunakan keempat persamaan di atas, selanjutnya setiap koordinat robot musuh dicek satu persatu. Jika koordinat robot tersebut diantara kedua garis, maka robot tersebut berpotensi untuk menghalangi bola.

### 3. Hasil dan Pembahasan

Eksperimen yang dilakukan dalam penelitian ini bertujuan untuk mendapatkan konfigurasi strategi yang optimal dari algoritma yang digunakan. Selain itu, eksperimen dilakukan untuk mendapatkan data-data statistik yang diperlukan untuk analisis dan evaluasi algoritma. Adapun parameter-parameter yang akan dievaluasi adalah jumlah menang, seri, dan kalah. Selain itu, ditambah dengan parameter statistik tambahan berupa jumlah gol yang dihasilkan, tingkat penguasaan bola (*ball possession*), jumlah tendangan sudut (*corne*), dan jumlah *offside*.

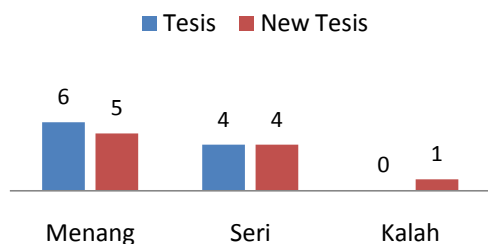
Adapun skenario eksperimen yang dilakukan adalah melakukan simulasi pertandingan sepak bola menggunakan *soccer server*. Ada dua tim yang dikembangkan penulis yaitu Tesis dan NewTesis yang keduanya menggunakan formasi 4-3-3. Tim Tesis hanya menggunakan algoritma pergerakan pemain, sedangkan NewTesis menggunakan algoritma pergerakan pemain ditambah dengan algoritma pengecekan jumlah musuh. Dua tim tersebut akan dipertandingkan dengan lima tim lain yang merupakan peserta dari RoboCup. Secara keseluruhan tim yang bertanding dalam simulasi ada tujuh tim, yaitu Tesis, NewTesis, Brainstormers (University of Osnabrueck, Germany) [8], Helios (National Institute of

Advanced Industrial Science and Technology, Japan), OxBlue (University of Oxford, UK) [9], OPU\_hana (Osaka Prefecture, Japan), dan UvA\_Trilearn (Universiteit Van Amsterdam, Holland) [10].

Simulasi pertandingan akan dilakukan selama 10 kali untuk setiap pertandingan. Satu kali simulasi pertandingan berjalan selama lima menit atau 3000 *time cycle* (waktu dalam *soccer server*). Sebagai contoh tim Tesis vs Brainstormers akan dilakukan sebanyak lima kali simulasi dan kemudian dibalik Brainstormers vs Tesis yang juga akan dilakukan sebanyak lima kali. Dikarenakan total ada tujuh tim dalam simulasi pertandingan, setiap tim berarti akan melakukan sebanyak 60 pertandingan.

Secara default simulasi pertandingan yang dilakukan dalam *soccer server* akan menghasilkan dua file yaitu file \*.rcg dan \*.rcl. File \*.rcg adalah file simulasi pertandingan yang dapat diputar kembali dengan menggunakan logplayer. Sementara file \*.rcl merupakan file yang berisi log aktivitas setiap tim selama simulasi pertandingan. Dari file \*.rcl akan diperoleh data statistik mengenai penguasaan bola, jumlah *corner*, dan jumlah *offside*.

Setelah melakukan simulasi pertandingan dari tujuh tim, akan dihitung jumlah menang, seri, dan kalah yang diperoleh dari setiap tim. Menang akan mendapat nilai 3, seri mendapat nilai 1, dan kalah tidak mendapat nilai. Adapun klasemen akhir simulasi pertandingan dari tujuh tim tersebut dapat dilihat pada tabel II.

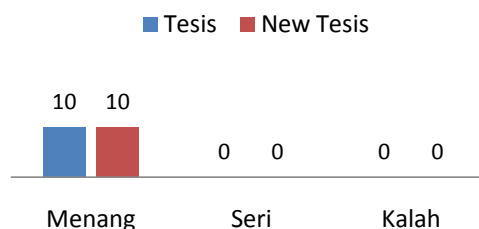


Gambar 8. Melawan Brainstormers.

Selanjutnya akan dilakukan analisis dari

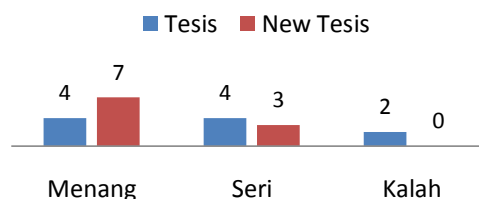
setiap pertandingan yang dilakukan tim Tesis dan NewTesis. Adapun parameter yang menjadi bahan perbandingan antara dua tim tersebut adalah jumlah menang, seri, dan kalah yang diperoleh ketika melawan 5 tim lawan yang lain, yaitu Brainstormers, OPU\_hana, OxBlue, Helios, dan UvA\_Trilearn. Gambar 8 menunjukkan jumlah menang, seri, dan kalah yang diperoleh tim Tesis dan NewTesis pada saat melawan Brainstormers. Tim NewTesis memiliki jumlah menang yang lebih sedikit dan jumlah kalah yang lebih banyak dari tim Tesis.

Gambar 9 menunjukkan jumlah menang, seri, dan kalah yang diperoleh tim Tesis dan NewTesis pada saat melawan OPU\_hana. Tim NewTesis memiliki jumlah menang, seri, dan kalah yang sama dengan tim Tesis.



Gambar 9. Melawan OPU\_hana.

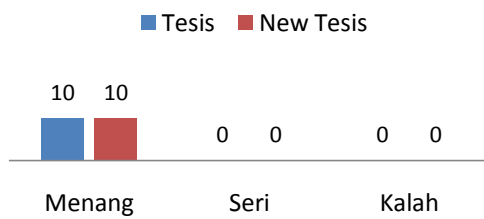
Gambar 10 menunjukkan jumlah menang, seri, dan kalah yang diperoleh tim Tesis dan NewTesis pada saat melawan OxBlue. Tim NewTesis memiliki jumlah menang lebih banyak, jumlah seri lebih sedikit, dan jumlah kalah lebih sedikit dari tim Tesis.



Gambar 10. Melawan OxBlue.

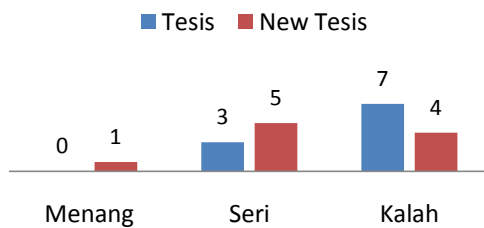
TABEL II  
KLASEMEN AKHIR SIMULASI

Peringkat	Tim	Main	Menang	Seri	Kalah	Selisih gol	Poin
1	Helios	60	46	12	2	233 - 15	150
2	New Tesis	60	36	16	8	164 - 23	124
3	Tesis	60	33	15	12	155 - 28	114
4	Brainstormers	60	21	21	18	55 - 29	84
5	Oxblue	60	19	16	25	132 - 74	73
6	OPU_hana	60	13	4	43	80 - 131	43
7	UvA Trilearn	60	0	0	60	1 - 520	0



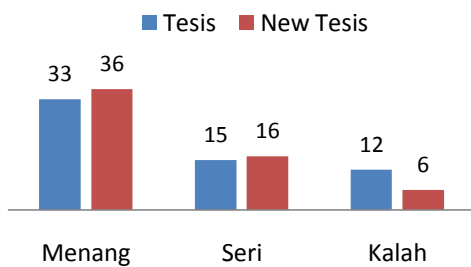
Gambar 11. Melawan UvA\_Trileam.

Gambar 11 menunjukkan jumlah menang, seri, dan kalah yang diperoleh tim Tesis dan NewTesis pada saat melawan Uva\_Trilearn. Tim NewTesis memiliki jumlah menang, seri, dan kalah yang sama dengan tim Tesis.



Gambar 12. Melawan Helios.

Gambar 12 menunjukkan jumlah menang, seri, dan kalah yang diperoleh tim Tesis dan NewTesis pada saat melawan Helios. Tim NewTesis memiliki jumlah menang lebih banyak, jumlah seri lebih banyak, dan jumlah kalah lebih sedikit dari tim Tesis. Secara keseluruhan dari 60 pertandingan yang dilakukan kedua tim menunjukkan hasil seperti terlihat pada gambar 13.

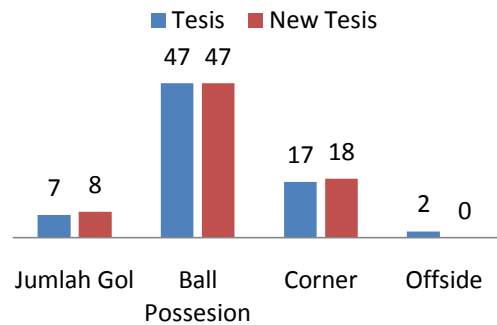


Gambar 13. Statistik total.

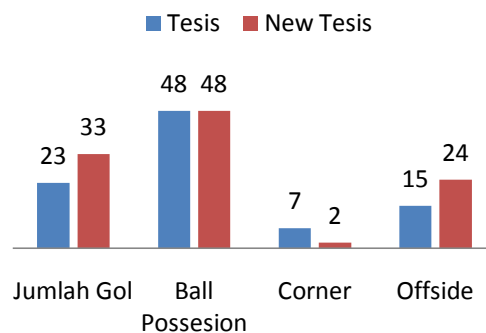
Tim Tesis memperoleh jumlah menang sebanyak 33, seri sebanyak 15, dan kalah sebanyak 12. Sementara tim NewTesis memperoleh jumlah menang sebanyak 36, jumlah seri sebanyak 16, dan jumlah kalah sebanyak 8. Secara keseluruhan tim NewTesis memiliki statistik jumlah menang, seri, dan kalah yang lebih baik dari tim Tesis.

Analisis yang akan dilakukan selanjutnya adalah penambahan parameter seperti jumlah gol,

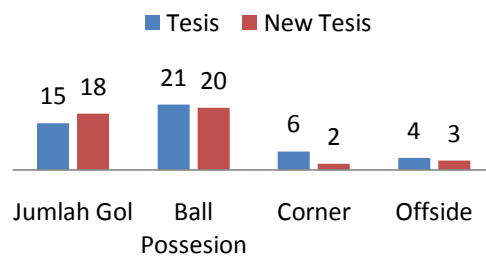
*ball possession*, jumlah *corner*, dan jumlah *offside*. Parameter tersebut diperoleh dari *file \*.rcl*. Dari *file* yang berisi log pertandingan akan dihitung parameter-parameter tersebut.



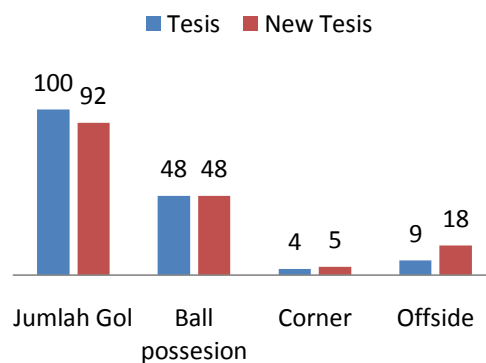
Gambar 14. Statistik ketika melawan Brainstormers.



Gambar 15. Statistik ketika melawan OPU\_hana.

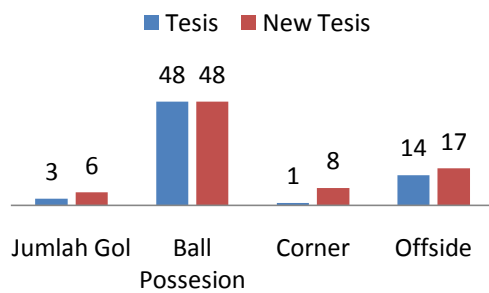


Gambar 16. Statistik ketika melawan OxBlue.

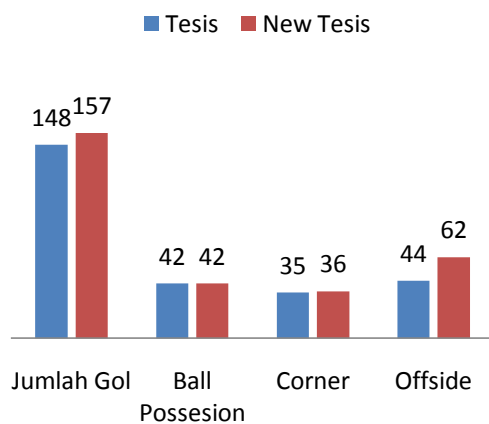


Gambar 17. Statistik ketika melawan UvA\_Trileam.





Gambar 18. Statistik ketika melawan Helios.



Gambar 19. Statistik keseluruhan.

Gambar 14, 15, 16, 17, dan 18 menunjukkan perbandingan jumlah gol, *ball possession*, jumlah *corner*, dan jumlah *offside* antara tim Tesis dan NewTesis ketika melawan tim Brainstormers, OPU\_hana, OxBlue, UvA\_Trilearn, dan Helios. Secara keseluruhan statistik yang diperoleh tim Tesis dan NewTesis dapat dilihat pada gambar 19. Tim NewTesis memiliki statistik jumlah gol dan jumlah *offside* yang lebih baik dari pada tim Tesis. *Ball possession* dan jumlah *corner* kedua tim kurang lebih sama.

#### 4. Kesimpulan

Strategi yang dirancang untuk pertandingan robot sepak bola menggunakan pemetaan yang berbasis sistem koordinat Cartesius. Strategi tersebut terdiri dari strategi pembagian partisi dan strategi *passing*. Pembagian partisi berguna untuk mencegah robot berkumpul pada daerah tertentu dan menentukan robot yang harus mengejar bola. Strategi *passing* berguna untuk menjauhkan bola dari robot musuh dan mengatur posisi bola untuk ditembakkan ke gawang lawan.

Hasil eksperimen menunjukkan nilai persentase yang rendah tidak selalu berarti bahwa kinerja robot buruk. Pada persentase penguasaan bola, nilai yang rendah justru menunjukkan bahwa

strategi *passing* ternyata efektif karena hanya memerlukan sedikit *passing* untuk menghasilkan gol. Hal ini terlihat dari jumlah gol yang dihasilkan lebih banyak dari robot musuh. Pada strategi partisi, kecepatan bola yang rendah akan mengurangi kinerja robot yang menggunakan partisi. Sebaliknya kecepatan bola yang tinggi akan meningkatkan kinerja robot. Untuk meningkatkan kinerja robot, strategi partisi dan strategi *passing* harus diimplementasikan secara bersamaan.

Saran untuk penelitian selanjutnya, perlu dikembangkan strategi yang lain selain strategi pergerakan pemain dan *passing*. Adapun strategi yang dapat dikembangkan adalah pemilihan pemain berdasarkan posisi yang paling dekat dengan gawang untuk menembak bola. Hal ini dapat meningkatkan peluang terjadinya gol. Selain itu, juga perlu melihat pengaruh ruang tembak dalam melakukan *shooting*.

Penelitian selanjutnya dapat dikembangkan dengan mengimplementasikan algoritma optimasi dalam simulasi robot sepak bola. Salah satunya adalah PSO (*particle swarm optimization*). Peranan dan pengaruh PSO dalam simulasi robot sepak bola masih merupakan bidang yang sangat terbuka untuk diteliti.

Selain algoritma PSO, masih banyak lagi algoritma yang dapat diimplementasikan dalam simulasi robot sepak bola, seperti algoritma genetika, ACO (*ant colony optimization*), *neural network*, *machine learning*, dan sebagainya. Akan lebih baik lagi jika simulasi robot sepak bola menggunakan gabungan dari algoritma-algoritma tersebut. Dengan menggabungkan algoritma tersebut, diharapkan akan mendapatkan hasil yang lebih baik bila dibandingkan hanya menggunakan satu jenis algoritma kecerdasan.

Terakhir perlu dipikirkan bagaimana cara untuk menterjemahkan simulasi robot sepak bola ke dalam robot sebenarnya. Salah satu robot yang dapat digunakan adalah robot *Lego Mindstorms NXT*. Robot tersebut merupakan robot edukasi yang dapat digunakan dalam penelitian robot sepak bola. Selain robot *Lego*, tidak tertutup kemungkinan menggunakan robot-robot yang lain atau merakit sendiri robot dari awal.

#### Referensi

- [1] A.K. Mackworth, In: A. Basu and X. Li (Eds.), *On Seeing Robots*, Computer Vision: Systems, Theory, and Applications, World Scientific Press, Singapore, pp. 1-13, 1993.
- [2] H. Kitano, M. Asada, Y. Kuniyoshi, I. Nota, & E. Osawa, "RoboCup: The World Cup Initiative" In *Proceeding of IJCAI-95*



- Workshop on Entertainment and AI/Alife*, pp. 19-24, 1995.
- [3] Aprilif, Past Symposiums, RoboCup, <http://www.robocup.org/2010/10/past-symposiums/>, 2010, retrieved February 9, 2010.
- [4] RoboCup Federation, About RoboCup, <http://www.RoboCup.org/about-RoboCup>, 2012, retrieved February 9, 2010.
- [5] I. Noda, H. Matsubara, K. Hiraki, & I. Frank, "Soccer Server: A Tool For Research On Multi-Agent Systems," *Applied Artificial Intelligence* vol. 12 , pp. 233-250, 1998.
- [6] M. Chen, E. Foroughi, et.al, Users Manual RoboCup Soccer Server for Soccer Server Version 7.07 and later, Sourceforge, <http://sourceforge.net/projects/sserver/>, 2003, retrieved February 9, 2010.
- [7] S.L. Wu, Y.R. Liou, W.H. Lin, & M.H. Wu, "A Multi-agent Algorithm for Robot Soccer Games in Fira Simulation League" (to be published).
- [8] M. Riedmiller, T. Gabel, F. Trost, & T. Schwegmann, In: L. Iocchi, H. Matsubara, A. Weitzenfeld, C. Zhou, (Ed.), *Brainstormers 2D – Team Description 2008*, RoboCup 2008: Robot Soccer World Cup XII, LNCS, Springer, 2008.
- [9] J. Ma & S. Cameron, "Combining Policy Search with Planning in Multi-Agent Cooperation", *RoboCup 2008: Robot Soccer World Cup XII*, Springer, pp. 532-543, 2009.
- [10] J.R. Kok, "Coordination and Learning in Cooperative Multiagent Systems," Ph.D thesis, Faculty of Science, University of Amsterdam, 2006.