BAB II
TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan (artificial intelligence) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat berfikir dan melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan lebih baik dari pada yang dilakukan manusia [3].

Beberapa hal yang dapat dikerjakan dalam bagian-bagian AI berikut [5] :

1. Sistem pakar (expert system) : komputer sebagai sarana untuk menyimpan pengetahuan para pakar sehingga komputer memiliki keahlian menyelesaikan permasalahan dengan meniru keahlian yang dimiliki pakar.
2. Pengelolahan bahasa alami (natural language processing) : user dapat berkomunikasi dengan komputer menggunakan bahasa sehari-hari, misalnya bahasa inggris, bahasa Indonesia, dan lain-lain.
3. Pengenalan ucapan (speech recognition) : manusia dapat berkomunikasi dengan komputer menggunakan suara.
4. Robotika dan sistem sensor : difokuskan pada produksi alat-alat mekanik yang dapat mengendalikan gerak.
5. Komputer vision : menginterpretasikan gambar atau objek-objek tampak melalui komputer.
6. Intelligent computer-aided instruction : komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. Game playing : pembuatan program-program bermain permainan.

Area dari Artificial Intelligence adalah gabungan beberapa area study [6], Yaitu :

1. *Logic* : digunakan untuk mempelajari logika dari sebuah argument dengan menerapkan aturan logika standar.

2. *Searching* : diterapkan pada AI mengacu pada metode pencarian untuk penyelesaian suatu masalah.
3. *Vision recognition dan pattern matching* : penting untuk beberapa aplikasi, termasuk robotic dan pengolah citra (image processing).
4. *Natural language processing* : digunakan untuk membangun kemampuan komputer berbahasa manusia secara lansung.
5. *Robotic* : digunakan untuk mengontrol gerakan.
6. *Learning* : program yang dapat belajar dari kesalahan dengan observasi (atau permintaan komputer) serta mempunyai kemampuan untuk mengambil 'benefit' dari pengalaman.
7. *Uncertainty dan fuzzy logic* : komputer dapat berfikir dengan menggunakan pengetahuan yang tidak lengkap dengan menerapkan penggunaan fuzzy logic.

2.2 Searching

Searching atau pencarian adalah mekanisme pemecahan masalah yang paling umum di dalam kecerdasan buatan. Di dalam permasalahan-permasalahan kecerdasan buatan, urutan langkah-langkah yang dibutuhkan untuk memperoleh solusi merupakan suatu isu yang penting untuk diformulasikan. Hal ini harus dilakukan dengan mengidentifikasikan proses *try and error* secara sistematis pada eksplorasi setiap alternative jalur yang ada [7].

Algoritma searching di dalam kecerdasan buatan yang umumnya dikenal adalah :

1. Pencarian Buta (Blind search)

   Algoritma yang tidak memberikan informasi tentang permasalahan yang ada, hanya sebatas definisi dari algoritma tersebut.

2. Pencarian Heuristik (Heuristik search)

   Walaupun dengan menggunakan algoritma pencarian buta, banyak permasalahan dapat dipecahkan, namun tidak semuanya dari algoritma tersebut dapat menyelesaikan masalah dengan efisien.

Selain dari kedua algoritma searching diatas ada juga algoritma yang dikenal dengan adversarial search, dimana didalamnya terdapat beberapa algoritma pencarian seperti algoritma MiniMax, Alpha beta Pruning, NegaMax, dan lain sebagainya.

2.2.1 Alpha Beta Pruning

Alpha beta pruning merupakan sebuah cara untuk mengurangi jumlah simpul yang dieksplorasi dari algoritma MinMax. Dengan adanya alpha beta, waktu yang dibutuhkan dalam pencarian bisa berkurang dengan cara membatasi waktu yang terbuang percuma pada saat mengevaluasi pohon permainan. Implementasi alpha beta akan memberikan jalur terbaik dalam setiap kemungkinan permainan dalam pohon permainan yang terbentuk [3]. Dalam alpha beta pruning, urutan jalannya algoritma akan dimulai sama seperti algoritma minimax. Untuk simpul MIN, nilai yang dihitung dimulai dengan +infinity dan akan menurun seiring jalannya permainan. Untuk simpul MAX, nilai akan dihitung dimulai dengan -infinity dan akan menaik seiring berjalannya waktu permainan. Pseudocode Alpha Beta Pruning dapat dilihat pada gambar 2.1.

```
Alpha-beta (player, board, alpha, beta)
  If (game over in current board position)
   return winner
    children = all legal moves for player from this
board
if(max's turn)
  for each child
    score = alpha-beta(other player, child, alpha, beta)
    if score > alpha then alpha = score
    if alpha >= beta then return alpha
  return alpha
else (min's turn)
  for each child
    score = alpha-beta(other player, child, alpha, beta)
    if score < beta then beta = score
    if alpha >= beta then return beta
  return beta
```

Gambar 2.1 Pseudocode Alpha Beta Pruning

Pada penelitian ini, nilai-nilai yang terdapat pada simpul atau node didapat setelah melakukan perhitungan menggunakan sebuah formula atau rumus.

2.3 Formula

Formula merupakan rumus yang digunakan untuk mencari nilai-nilai yang nantinya nilai tersebut dimasukkan kedalam simpul atau node untuk setiap posisi pemain. Berikut ini merupakan rumus yang digunakan pada penelitian ini.

$$H = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad \text{.....(2.1)}$$

Dimana :

H = total nilai

X1 = koordinat x pemain

X2 = koordinat x bola

Y1 = koordinat y pemain

Y2 = koordinat y bola

2.4 RoboCup

Robot bermain sepakbola pertama kali disebutkan oleh professor Alan Mackworth (University of British Columbia, Kanada) dalam sebuah makalah yang berjudul "*On seeing Robots*" pada tahun 1992 dan kemudian diterbitkan dalam sebuah buku *Computer Vision : System, Theory, and Applications*. Pada bulan juni 1993, beberapa peneliti seperti Minoru Asada, Yasuo Kuniyoshi, dan Hiroaki Kitono memutuskan untuk meluncurkan sebuah kompetisi robot yang diberi nama J-League. Dalam sebulan J-League mendapat reaksi yang luar biasa dari para peneliti diluar Jepang, dan meminta inisiatif untuk diperpanjang menjadi proyek internasional. Oleh karena itu J-League berganti nama menjadi Robot World Cup disingkat RoboCup [8].

RoboCup adalah sebuah federasi untuk mengakomodasi para peneliti utamanya pada robot sepak bola. Salah satu kegiatannya adalah dengan cara

menyelenggarakan turnamen piala dunia setiap tahun. Anggota-anggota Federasi RoboCup adalah para peneliti yang aktif melakukan penelitian dalam bidang ini dan mereka adalah wakil-wakil dari universitas-universitas atau perusahaan-perusahaan yang berkepentingan dan berkontribusi di bidang ini.

Federasi RoboCup telah menetapkan tujuan-tujuan dan suatu jadwal penelitian. Penetapan tujuan ini dimaksudkan untuk mendesak perkembangan terkini (*the state-of-the-art)* selanjutnya, sehubungan dengan *test-beds* yang telah diformalkan. Artinya bahwa sebenarnya tujuan utama yang paling penting dari RoboCup adalah memajukan tingkat teknologi masyarakat di segala bidang.

Hingga saat ini, federasi RoboCup internasional telah mempromosikan kompetisi atau liga robot sepak bola sebagai berikut.

a. *RoboCup Soccer,* terdiri atas :
    1. *Simulation League,* terdiri dari :
        a. *2D Simulation League,*
        b. *3D Simulation League,*
        c. *3D Development.*
    2. *Small Size Robot League*
    3. *Middle Size Robot League*
    4. *Standard Platform League*
    5. *Humanoid League,* terdiri dari :
        a. *Kid Size* (30-60 cm *height),*
        b. *Teen Size* (100-160 cm *height).*
b. *RoboCup Rescue,* terdiri atas :
    1. *Rescue Simulation League,*
    2. *Rescue Robot League.*
c. *RoboCup Junior,* terdiri atas :
    1. *Soccer Challenge,*
    2. *Dance Challenge,*
    3. *Rescue A Challenge,*

4. *Rescue B Challenge.*

d. *RoboCup @Home*

Pada penelitian ini lebih difokuskan pada *RoboCup 2D Simulation League.* Simulasi robot sepak bola dalam RoboCup yang menggunakan simulator yang disebut dengan *Soccer Server.* Simulator ini merupakan sebuah simulasi pertandingan sepak bola yang divisualisasikan dengan menunjukan lapangan simulator pada layar monitor komputer.

2.4.1 Simulator RoboCup Soccer

Simulator yang telah disediakan untuk liga RoboCup Soccer terdiri dari 3 komponen utama, yaitu: Soccer Server, Soccer Monitor dan Log Player. Sebuah simulasi pertandingan sepak bola dilakukan dalam bentuk *client-server.* Soccer server menyediakan sebuah domain (sebuah lapangan sepak bola virtual) yang mensimulasikan semua gerakan-gerakan obyek dalam domain ini, dan mengontrol sebuah game sepak bola sesuai aturan-aturan tertentu.

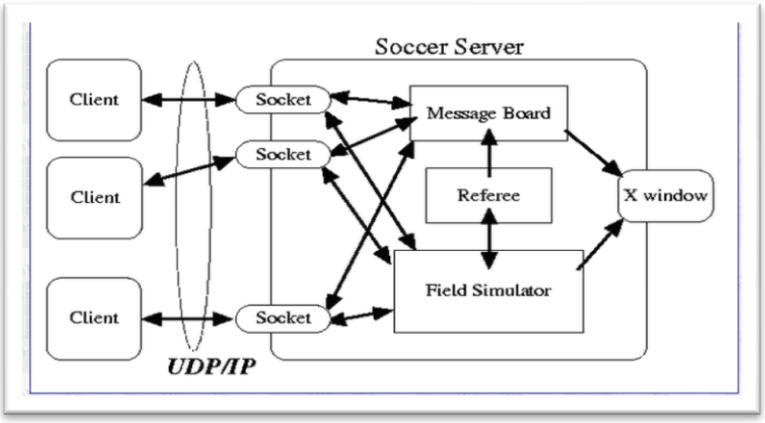
Sebuah pertandingan sepak bola virtual melibatkan 3 komponen yaitu Sistem Soccer Server, Sistem Soccer Monitor, dan Sistem Soccer Client.

Selanjutnya tiga kompenen utama tersebut, dijabarkan sebagai berikut:

a. Soccer Server.

   Soccer Server adalah sebuah file *executable* yang mengatur hubungan antara klien-klien (monitor, klien kesebelasan), dan menghasilkan lingkungan virtual bagi klien-klien. Soccer Server juga mengatur semua peraturan permainan dan protocol-protokol komunikasi antar klien. Soccer Server ini adalah properti RoboCup, dan konfigurasinya tidak dapat diubah lagi. Pada penelitian ini Soccer Server yang digunakan adalah versi 15.0.1 dikarenakan pada saat menginstallkan Soccer Server

versi yang terbaru terdapat error, dikarenakan ada beberapa library yang tidak mendukung. Connection antara client dengan server dapat dilihat pada gambar 2.2.



Gambar 2.2 Connection antara Client dengan Server

b. Soccer monitor.

Soccer Monitor adalah sebuah file executable yang menyediakan tampilan visual yang berupa grafis lapangan sepak bola dan tampilan visual para *agent* / pemain di atas lapangan. Informasi-informasi mengenai posisi dan aksi yang dilakukan oleh pemain berasal dari Soccer Server. Seperti halnya Soccer Server, Soccer Monitor ini adalah properti RoboCup, dimana konfigurasinya sudah permanen dan tidak dapat diubah-ubah lagi. Tampilan soccer monitor dapat dilihat pada gambar 2.3.



Gambar 1.3 Tampilan Soccer Monitor

Soccer monitor yang digunakan adalah versi 15.0.0 . soccer monitor juga tidak menggunakan versi yang terbaru karena ada library yang tidak mendukung.

c. Soccer Client.

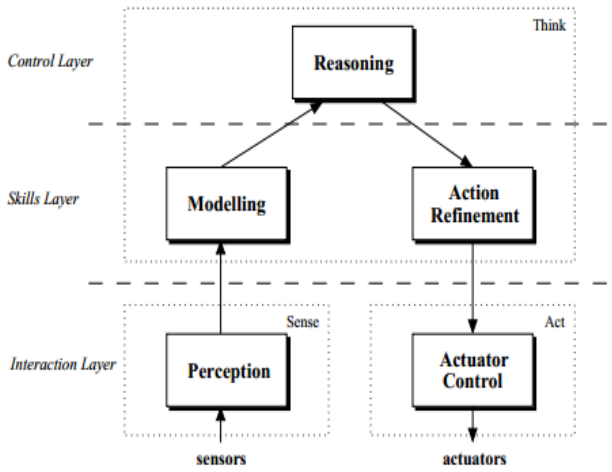
Soccer Client adalah sebuah file *executable* yang menghadirkan satu set pemain sepakbola (*soccer players)* untuk bermain sepakbola dalam lingkungan Soccer Server. Soccer Client telah banyak dikembangkan oleh banyak universitas, misalnya: Carnegie Melon University, Tsing Hua University, Universiteit van Amsterdam, dan lain-lain. Masing-masing universitas tersebut telah mengimplementasikan berbagai kecakapan pada agent-agentnya, sedemikian supa sehingga para agent dapat memainkan pertandingan sepakbola melawan agent dari Soccer Client lain secara *autonomous.* Soccer Client adalah properti dari masing-masing pengembang, dan bukan properti RoboCup. Pengembang dapat dengan bebas mengembangkan sistem kliennya masing-masing selama sistem tersebut dapat terhubung dan berkomunikasi dengan Soccer Server. Soccer Client yang dikembangkan adalah Uva Trilearn pertama kali

dikembangkan pada tahun 2001 oleh Jelle R. Kok. Uva Trilearn bersifat open source.

Selain tiga komponen utama diatas, ada sebuah komponen yang berfungsi sebagai pemutar video yang disebut dengan log player. Log player yang digunakan adalah versi 15.0.0, log player juga tidak menggunakan versi terbaru karena ada library yang tida mendukung. Fungsi dari log player adalah untuk memutar kembali sebuah pertandingan sepak bola.

2.5 UvA Trilearn

Uva trilean pertama kali dikembangkan pada tahun 2001 oleh Jelle R. Kok [9]. Dan pertamakalinya ikut kompetisi pada jerman terbuka 2001 di Paderborn dan di kejuaraan dunia 2001 di seattle (USA). Pada tahun 2003 Jelle R. Kok mengembangkan lebih lanjut UvA Trlearn sebelumnya dengan memperbaiki keahlian merebut bola dengan mempertimbangkan kemampuan tim lawan untuk aksi yang sama, memperbaiki teknik *passing* dengan mempertimbangkan berbagai piilihan aksi *passing* yang dapat dilakukan, dan perbaikan terakhir adalah pengunaan c*oordination graph* untuk untuk menentukan koordinasi antar agen. Dan hasilnya yaitu UvA Trilearn menjadi juara pada Robocup-2003. Berikut ini merupakan struktur dari *soccer client* UvA Trilearn dapat dilihat pada gambar 2.4.



Gambar 2.4 Arsitektur Agent Uva Trilearn

2.6 Soccer

Soccer atau sepakbola merupakan salah satu cabang olahraga yang sangat populer di seluruh dunia. Hal tersebut dapat dilihat dari perkembangan sepakbola yang semakin maju dan memiliki jumlah penggemar paling banyak di dunia dibandingkan dengan cabang olahraga lainnya. Permainan sepakbola sendiri dimainkan oleh dua tim dan satu tim terdiri dari sebelas pemain, untuk dapat memainkan permainan ini dengan baik dibutuhkan kerjasama tim dan teknik dalam bermain sepakbola.

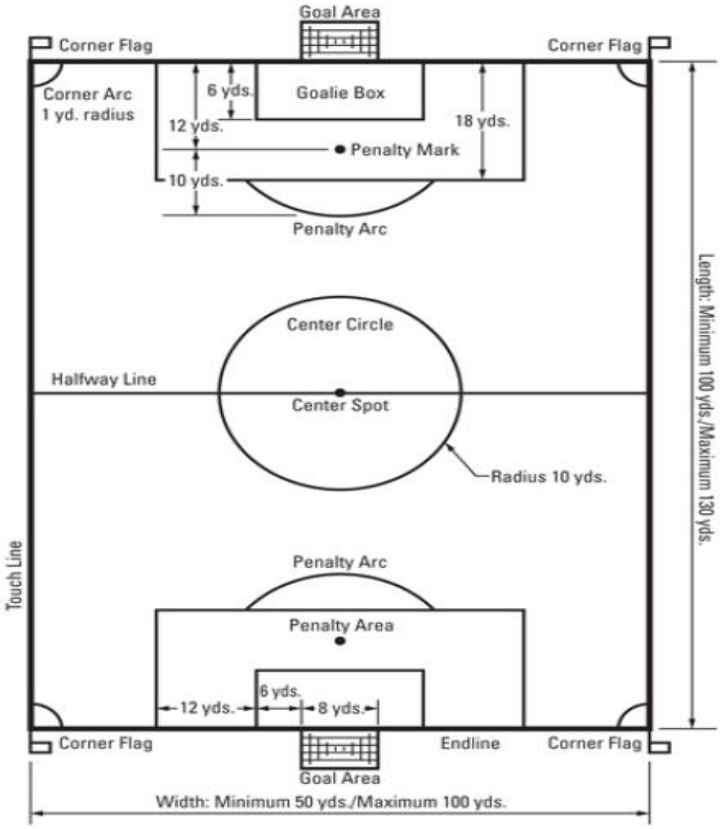
Menurut sejarah kontemporer permainan favorit dunia berkembang lebih dari 100 didunia. Semuanya dimulai pada tahun 1863 di inggris. Ketika rugby dan sepakbola berada pada cabang olahraga yang berbeda. Asosiasi sepakbola di inggris dibentuk menjadi badan pertama yang mengatur olahraga [10].

Kedua olahraga tersebut berasal dari cabang yang sama. Sebuah pencarian selama berabat-abat mengungkapkan bahwa setidaknya ada enam permainna yang berbeda, bervariasi untuk derajat yang berbeda, dan sejarah perkembangan sepakbola telah diteliti kembali. Apakah ini dapat dibenarkan dalam beberapa kasus yang diperdebatkan. Namun demikian, faktanya tetap bahwa orang telah menikmati sepakbola selama ribuan tahun dan sama sekali tidak ada alasan untuk menganggapnya sebagai penyimpangan terhadap permainan yang menggunakan tangan.

Pada dinasti han sepakbola disebut Tsu’Chu dengan cara menendang bola kulit yang diisi bulu atau rambut dengan ukuran bola yang hanya 30-40cm, dan terdapat sebuah jarring kecil dengan tongkat bambu panjang. Bentuk lain dari permainan sepakbola berasal dari jepang yang disebut kemari. Di yunani kuno juga ada permainan yang disebut *Episkyros,* di yunani permainan sepakbola dimainkan oleh dua tim dilapangan persegi panjang dengan ditandai oleh garis batas dan garis tengah. Tujuan dari permainan adalah untuk memenangkan pertandingan dengan berbagai strategi atau taktik.

2.6.1 Aturan Sepakbola

Sepakbola dimainkan dalam sebuah lapangan oleh sebelas pemain. Permainan biasanya dilakukan dalam waktu 2 X 45 menit dengan jeda istirahat lima belas menit yang dilakukan oleh para pemainnya. Kalau dalam waktu 2 X 45 menit masih dalam keadaan seri maka akan di adakan babak perpanjangan waktu selama 2 X 15 menit. Setelah perpanjangan waktu masih seri maka akan dilakukan adu pinalti. Pemimpin atau pengadil pertandingan dalam sepakbola adalah wasit. Biasanya dia dibantu oleh dua orang hakim garis dan dua orang wasit cadangan yang bertugas ketika terjadi pergantian pemain. Ukuran lapangan sepakbola adalah 91,4 m dan 54,8 m dengan ukuran gawang 7,32 x 2,44 meter [11]. Lapangan sepakbola dapat dilihat pada gambar 2.5.



Gambar 2.5 Lapangan Sepakbola

Ada beberapa kondisi khusus dalam sepakbola yang umumnya terjadi, yaitu [11]:

1. Kickoff : pada permulaan permainan dan setelah gol, tim melakukan kockoff dari tengah untuk memulai permainan.
2. Penalty kick : ketika pemain melakukan pelanggaran di area penaltinya sendiri, tim yang mendapatkan tendangan penalty ini adalah tendangan berhadapan lansung dengan kipper yang berjarak 12 yard.
3. Throw-ins : jika bola keluar dari batas garis samping kiri atau kanan maka lemparan bebas ke dalam lapangan diberikan kepada tim yang terakhir tidak menyentuh bola. Jika bola keluar dari batas garis belakang, tendangan goal kick diberikan jika tim penyerang yang menyentuh bola bola terakhir, dan tendangan sudut diberikan jika tim bertahan yang menyentuh bola terakhir.
4. Goal kick : ketika penyerang menendang bola keluar dari garis gawang, tim yang bertahan diberikan goal kick, bola dapat ditempatkan di manapun di dalam area gawang dan setiap pemain (termasuk kipper) dapat melakkan tendangan.
5. Corner kick : jika bola keluar melewati garis gawang dan terakhir tersentuh oleh pemain tim bertahan, tendangan sudut diberikan kepada penyerang. Bola ditempatkan disudut lingkaran samping area gawang dimana bola itu keluar dan seperti halnya tendangan bebas, pemain bertahan harus berdiri setidaknya 10 meter dari bola.
6. Offside : pemain berada pada posisi offside jika, pada saat rekan satu tim memainkan bola, dia berada dekat dengan gawang dari pada bola dan dua pemain lawan. Jika pemain melakukan offside, pemain lawan mendapatkan tendangan bebas dari tempat dimana offside itu terjadi.

2.6.2 Posisi Pemain Sepakbola

Setiap tim sepakbola memiliki 11 pemain yang terdiri dari 1 kiper dan 10 pemain. Tujuan dari permainan sepakbola adalah untuk memasukkan bola ke gawang lawan. Tim yang menang adalah tim yang paling banyak mencetak gol. Berikut ini peran setiap posisi pemain [11]:

1. Kiper : hanya kiper yang diperbolehkan menggunakan tangannya, selama masih dalam area 18 x 44 yard yang disebut area penalty. Jika seorang penjaga gawang memegang bola diluar kotak dia akan mendapatkan kartu dan tim penyerang diberikan tendangan bebas dari tempat itu.
2. Pemain belakang : mereka bermain didepan kipper dan tugas utama mereka adalah untuk menghentikan serangan lawan. Tugas mereka dan tanggung jawab mereka dapat bervariasi dari jangkauan zona pertahanan mereka dimana mereka membela daerah tertentu. Pemain sayap belakang yang ada dikiri dan kanan, umumnya jarang bergerak dari sisi mereka di lapangan, dan pemain belakang tengah bermain ditengan lapangan dan biasanya menjaga terhadap pencetak gol atau penyerang tengah.
3. Pemian tengah : para pemain ini adalah penghubung antara pertahanan dan penyerangan. Gelandang harus memiliki fisik yang baik karena diharapkan untuk menjalankan sebagian besar permainan. Mereka harus mampu menembus jauh di dalam wilayah musuh untuk menyerang dan membuat transisi ke pertahanan ketika lawan berhasil mengambil bola. Gelandang bisa mengkhususkan diri sebagai pemain gelandang menyerang atau bertahan.
4. Pemain depan : tugas utama mereka adalah untuk mencetak gol sebanyak mungkin. Ada beberapa jenis penyerang, penyerang sayap bermain di kedua sisi kiri atau kanan dan biasanya berlari

dari atas ke bawah pada sisi lapangan. Mereka dapat membawa bola ke area penalty untuk menembak atau tetap di sisi dan mencoba untuk mengoper kepada temannya di daerah tersebut. Penyerang tengah bermain di tengah lapangan, tapi mereka dibolehkan berkeliling jika ada ruang terbuka disana, penyerang tengah juga dikenal sebagai strike, harus menjadi pemimpin tim dalam mencetak gold an menjadi pemain paling berbahaya yang berada didepan.

2.7 Object Oriented

Pendekatan berorientasi objek adalah cara memandang persoalan menggunakan model – model yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas [12]. Pada pendekatan ini, organisasi perangkat lunak adalah sebagai kumpulan objek diskrit yang saling bekerja sama, berkomunikasi dan berinteraksi menuju sasaran tertentu. Pendekatan berorientasi objek mempunyai tiga karakteristik utama, yaitu :

1. *Encaptulation*

   Encaptulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama – sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau objek lain kecuali prosedur yang berada dalam objek itu sendiri.

2. *Inheritance*

   Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data / atribut dan metoda dari induknya langsung. Atribut dan metoda dari objek induk diturunkan kepada anak objek, demikian seterusnya. Pendefinisian objek dipergunaan untuk membangun suatu hirarki dari objek turunannya, sehingga tidak perlu membuat atribut dan metoda lagi pada anaknya, karena telah mewarisi sifat indunya. *Inheritance* mempunyai arti bahwa atribut dan operasi yang

dimiliki bersama diantara kelas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan secara spesifik mejadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.

3. *Polymorphism*

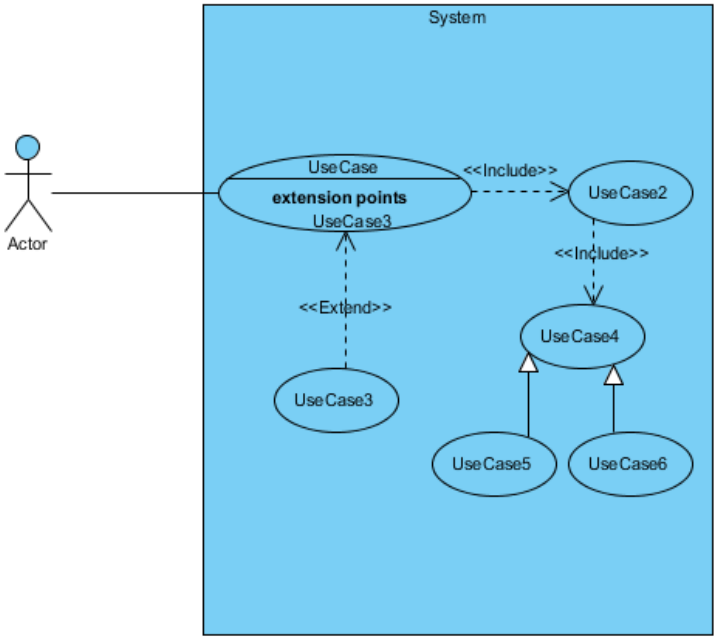
Polymorphism (polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Suatu implementasi yang spesifik dari suatu operasi dari kelas tertentu disebut metoda, karena operator berorientasi objek adalah bersifat polimorfisme, mungkin dapat mempunyai lebih dari satu metoda.

2.8 UML

UML (Unified Modeling Language) adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi. Ada beberapa diagram yang disediakan dalam UML antara lain [12]:

1. Use case diagram

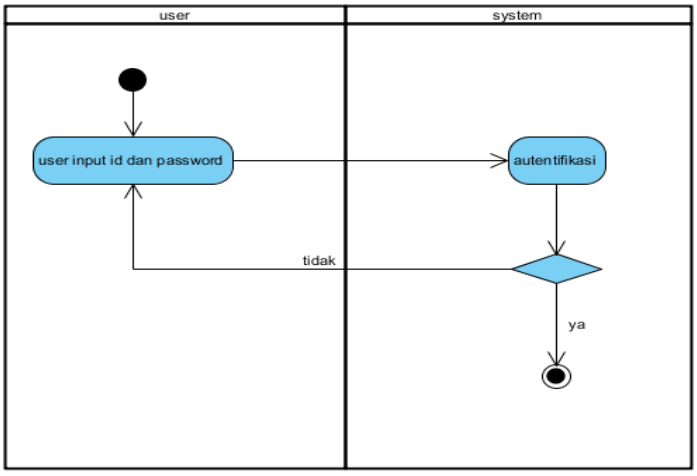
Diagram ini menunjukkan sekumpulan kasus fungsional dan aktor (jenis kelas khusus) dan keterhubungannya. Berikut ini adalah contoh use case diagram.



Gambar 2.6 Contoh Use Case Diagram

2. Activity diagram

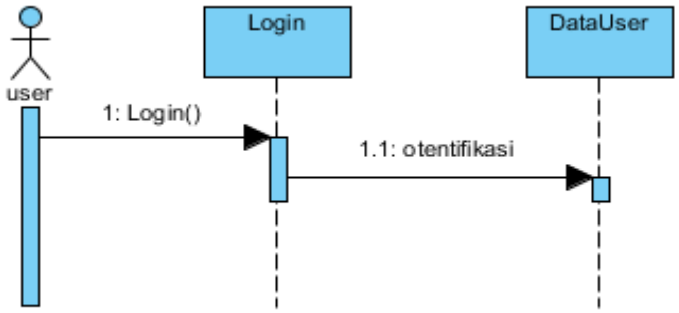
Diagram ini untuk menunjukkan aliran aktifitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem. Diagram ini penting untuk memodelkan fungsi sistem dan menekankan pada aliran kendali diantara objek – objek. Berikut ini adalah contoh activity diagram.



Gambar 2.7 Contoh Activity Diagram

3. Sequence diagram

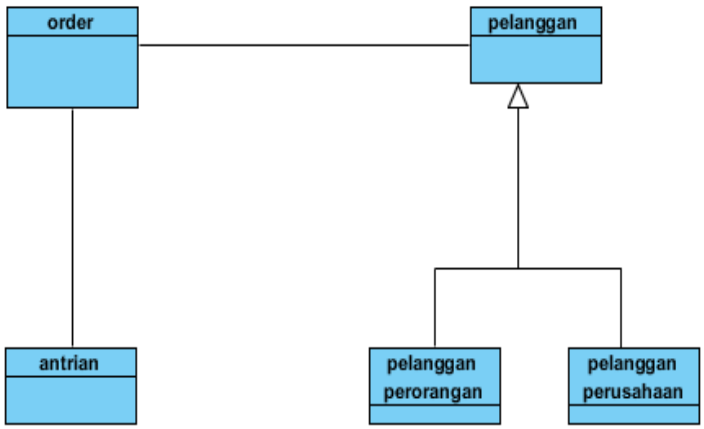
Diagram ini menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada basis keberurutan waktu dari pesan – pesan yang terjadi. Berikut ini adalah contoh dari sequence diagram.



Gambar 2.8 Contoh Sequence Diagram

4. Class diagram

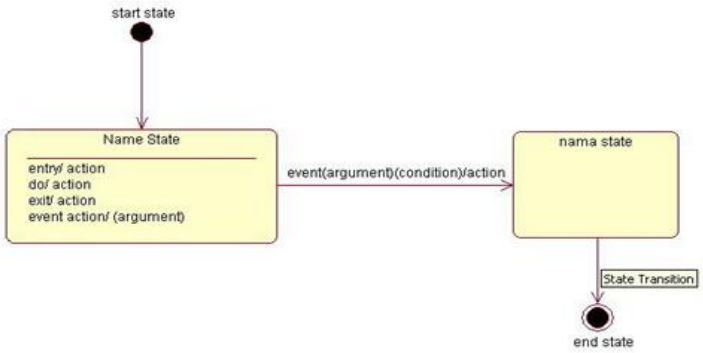
Diagram ini menunjukkan sekumpulan kelas, *interface* dan kolaborasi keterhubugannnya. Diagram kelas ditunjukkan untuk pandangan statik terhadap sistem. Berikut ini adalah contoh dari class diagram.



Gambar 2.9 Contoh Class Diagram

5. *Statechart diagram*

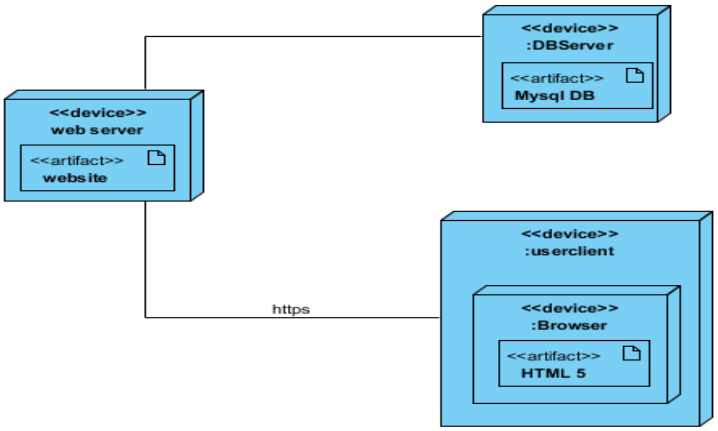
Diagram ini adalah *state machine diagram*, berisi *state*, transisi, kejadian dan aktifitas. Statechart merupakan pandangan dinamis dari sistem. Diagram ini penting dalam memodelkan perilaku antarmuka, kelas, kolaborasi dan menekankan pada urutan kejadian. Penting untuk sistem reaktif yang dipicu di dunia nyata. Berikut ini adalah contoh dari statechart diagram.



Gambar 2.10 Contoh Statechart Diagram

6. *Deployment diagram*

Diagram ini menunjukkan konfigurasi pemrosesan saat jalan dan komponen – komponen yang terdapat didalamnya. Diagram ini merupakan pandangan statik dari arsitektur. Berikut ini adalah contoh dari deployment diagram.



Gambar 2.11 Contoh Deployment Diagram

2.9 C++

Bahasa C++ diciptakan oleh Bjarne Stroudtrup di AT & T Bell laboratories awal tahun 1980-an berdasarkan C ANSI (*American National Standard Institute).* Pertama kali prototype C++ muncul sebagai C yang dipercanggih dengan fasilitas kelas. Bahasa tersebut disebut dengan nama *C with Class.* Selama tahun 1983-1984, bahasa C dengan kelas disempurnakan dengan menambahkan fasilitas operator dan fungsi yang kemudian melahirkan apa yang disebut C++. Simbol ++ merupakan operator C untuk operasi penaikan, simbol itu muncul untuk menunjukkan bahwa bahasa baru ini merupakan versi yang lebih canggih dari C.

2.10 Linux Ubuntu

Ubuntu adalah salah satu distribusi linux yang berbasiskan pada Debian dan memiliki interface desktop. Nama ubuntu diambil dari nama sebuah konsep ideology di Afrika Selatan. Ubuntu berasal dari bahasa kuno afrika yang berarti "rasa perikemanusiaan terhadap sesama manusia" [13]. Ubuntu sendiri bersifat open source, dan siap untuk dipergunakan dalam kondisi yang stabil. Ubuntu didukung oleh perusahaan bernama Canonical, Ltd yang memiliki tujuan untuk membantu perkembangan, distribusi, dan promosi dari produk-produk yang bersifat open source.

Penulis disini menggunakan system operasi linux ubuntu untuk menjalankan simulator robocup soccer dengan alasan lebih mudah untuk menginstall library yang dibutuhkan untuk menjalankan robocup soccer simulator.