BAB II
LANDASAN TEORI

2.1 Peringatan 17 Agustus

Hari peringatan Kemerdekaan Republik Indonesia, bertepatan tanggal 17 Agustus 1945, Soekarno-Hatta yang membacakan proklamasi tanda kemerdekaan bangsa Indonesia dari kolonialisme Belanda. Sekalipun Belanda belum mengakui kemerdekaan Indonesia pada tanggal 17 Agustus 1945 tapi sebagai bangsa yang sudah merdeka yaitu bangsa Indonesia, hal itu tidaklah menjadi halangan untuk menyatakan kemerdekaan. Tak heran setiap tanggal 17 Agustus selalu diperingatan dengan meriah. Perlombaan yang seringkali diadakan ketika 17 Agustus adalah panjat pinang, balap karung, tarik tambang, balap bakiak, makan kerupuk, perang bantal, dan masih banyak lagi.

Tetapi masyarakat Indonesia sampai saat ini mungkin masih menganggap bahwa perlombaan itu hanya sebuah *game* semata, padahal ada arti dan makna tersendiri dari setiap permainan tersebut. Berikut adalah beberapa ulasan mengenai permainan khas 17 Agustus:

1. Panjat Pinang

   Perlombaan panjat pinang ini seolah-olah menjadi icon perayaang hari kemerdekaan dikarekan selalu diadakan sejak proklamasi kemerdekaan. Perlombaan dengan pohon pinang yang tinggi dan batang yang dilumuri oleh pelumas serta ada rangkaian hadiah menarik di atas menjadi hal yang digemari oleh warga Indonesia. Berdasarkan catatan sejarah, perlombaan panjat pinang sudah ada sejak zaman penjajah. Orang Belanda menggelarnya saat acara besar seperti hajatan atau pernikahan. Saat itu, hadiah yang diperebutkan berupa bahan makanan dan pakaian. Sementara orang pribumi bersusah payah memperebutkan hadiah, para orang Belanda menonton sambil tertawa. Artinya, lomba panjat pinang bukan murni lahir dari budaya

Indonesia, tapi adaptasi dari kegiatan orang Belanda saat itu. Esensi dari lomba ini bagi bangsa Indonesia adalah mengenai semangat kebersamaan dan gotong royong untuk mencapai suatu tujuan. Selain itu, panjat pinang merupakan gambaran bahwa meraih kemerdekaan itu tidak mudah. Ada jalan "licin" yang terkadang membuat bangsa Indonesia jatuh. Dibutuhkan kerja keras, jiwa pantang menyerah, serta gotong royong untuk bisa meneriakkan kata "merdeka".

2. Balap Karung

Perlombaan balap karung ini cukup populer saat memperingati hari kemerdekaan, baik di desa maupun kota. Lomba ini tentu saja mewajibkan para peserta memasukkan bagian bawah badannya ke karung, kemudian berlomba sampai garis finis. Perlombaan balap karung mengingatkan kita pada zaman penjajahan dahulu. Saat itu, sebagian besar rakyat mengalami penderitaan, termasuk kesulitan memperoleh pakaian yang layak. Yang tersisa hanyalah karung goni bekas yang, mau tidak mau, terpaksa mereka kenakan untuk menutupi tubuh. Filosofi dalam lomba ini terletak pada saat peserta menginjak-injak karung, yang berarti sudah saatnya mereka meninggalkan pakaian yang tak layak untuk dikenakan tersebut.

3. Tarik Tambang

Dua kelompok bertanding dari dua sisi yang berlawanan dan memegang erat satu sisi tali tambang. Masing-masing kelompok mencoba menarik tali tambang sekuat mungkin agar kelompok lawan melewati garis pembatas yang berarti kalah. Pada umumnya, pemain dengan kekuatan paling besar ditempatkan di ujung tali untuk menahan atau menghentak pada saat penarikan. Makna dari lomba tarik tambang sama dengan panjat pinang, yaitu kerja keras dan semangat kebersamaan yang dimiliki setiap anggota kelompok untuk meraih tujuan kemenangan. Hal ini mengambarkan persatuan dan kesatuan bangsa yang saat itu melawan penjajahan untuk meraih kemerdekaan. Tarik tambang bukan lahir dari budaya bangsa Indonesia, namun jauh sebelum kemerdekaan Indonesia diraih. Tercatat dalam perhelatan Olimpiade pada 1900 sampai 1920 silam, tarik tambang

masuk kategori olah raga resmi yang diperlombakan, dengan nama Tug of War.

4. Balap Kelereng

   Biasa dalam perlombaan ini sendok membawa sendiri dari rumah masing-masing peserta. Tantangannya kelereng yang diletakkan di atas sendok tidak boleh terjatuh sambil sendok dibawa memakai mulut dari garis start hingga garis finish. Makna dari permainan ini adalah dalam mencapai tujuan itu harus sabar, tenang dan jangan terburu-buru.

5. Lomba Makan Kerupuk

   Kerupuk terikat pada seutas tali dan digantungkan di atas mulut pererta. Kedua tangan tidak boleh memegang tali ataupun kerupuk, untuk itu kedua tangan disilangkan dibelakang pinggang. Permainan ini mengajarkan kepada kita, di jaman penjajahan dulu rakyat mengalami kesulitan sandang, pangan, dan papan. Untuk makan yang paling sederhana sekali pun mengalami kesulitan, akijbat hasil panen penduduk diambil paksa oleh penguasa. Akibatnya, rakyat banyak yang kekurangan makanan dan menderita gizi buruk bahkan hingga mati kelaparan.

2.2 Game

Game bila diartikan kedalam bahasa Indonesia mempunyai arti permainan. Permainan dalam hal ini merujuk pada pengertian "kelincahan intelektual" (*intellectual playability*). Sebuah game diharuskan mempunyai komponen serta peraturan yang jelas. Karena adanya peraturan tersebut maka ada kondisi dimana pemain akan mengalami menang ataupun kalah dengan tujuan untuk *refreshing.*

2.2.1 Sejarah Game

Sejak pertama kali ditemukan hingga saat ini, *game* telah melewati berbagai fase evolusi penting dimana ini adalah peran besar perkembangan *game. Game* pertama kali dibuat pada tahun 1966 oleh Ralp Baer bersama dengan timnya yang berjumlah 500 orang. Tim ini terdiri dari para insinyur dan teknisi. Setelah sebulan bekerja keras, Baer berhasil menampilkan dua titik putih yang kejar-kejaran dilayar. Pada tahun 1972 muncul nama baru dalam dunia *game*. Nolan Bushel mendirikan perusahaan Atari dan membuat *game arcade* pong.

Pada era baru dalam perkembangan dunia *game* terjadi pada tahun 1988 yang didominasi oleh perusahaan Jepang. Nintendo awalnya hanya memproduksi mesin fotocopy, dengan video game sistemnya telah mencapai omset miliaran dolar. Setelah awal perkembangan game ini, dunia game telah melalui beberapa fase yaitu fase game PC, fase game console, dan fase game online. Sekarang ketiga fase game tersebut sudah semakin banyak variannya.

Divinekids adalah salah satu *game* yang diakui dan mendapat award resmi dari MURI (Museim Rekor Indonesia) sebagai *game* pertama Indonesia pada tahun 2004.[2] Akan tetapi pada kenyataannya sudah banyak jenis-jenis *game* buatan anak negeri (Indonesia) tetapi karena kekurangan syarat-syarat minimal seperti karakter yang menggunakan karakter asing, penyebaran yang kurang luas, bahasa yang digunakan bahasa asing, dan lain sebagainya. Oleh karena itu tidak terpilih sebagai *game* pertama di Indonesia.[2]

2.2.2 Klasifikasi *Game*

Game biasanya dikategorikan berdasarkan interaksi *gameplay* daripada perbedaan tampilan atau narasi. Sebuah jenis *game* dapat didefinisikan dengan kumpulan sebuah tipe permainan dari *game* tersebut. *Game* dapat diklasifikasikan menjadi dua, yaitu:

2.2.3 Berdasarkan Jenis *Platform* yang Digunakan

1. *Mobile games*, yaitu permainan yang dapat dimainkan khusus untuk mobile phone atau PDA.
2. PC games, yaitu video game yang dimainkan menggunakan personal computers.
3. Console games, yaitu video games yang dimainkan menggunakan *console* tertentu seperti Playstation 2, Playstation 3, XBOX 360 dan Nintendo wii.
4. Arcade games, yaitu yang sering disebut *ding-dong* dengan design tambahan seperti pistol, kursi khusus, sensor gerak, ataupun stir mobil yang membuat pemain merasa berada dalam permainan langsung dan lebih menikmati.

5. Handheld games, yaitu yang dimainkan di console khusus video game yang dapat dibawa kemana-mana, seperti Sony PSP.

2.2.4 Berdasarkan Genre Game

Menurut Prof. Jason Fritts *genre game* terdiri dari : [2]

1. *Action*

   Kebanyakan dari *arcade games* adalah *action games* juga. Ini dikarenakan *action games* memerlukan reflex yang cepat dan hati-hati dalam mengatur waktu untuk menaklukan rintangan yang ada. Tujuan dari *action game* adalah mengatasi musuh tanpa menghancurkan diri sendiri.

   Adapun *sub-genre* dari *action game*s ini :

   1. *Platformers ,* contoh : Super Mario Bros.
   2. *Shooters (First-person shooters dan Third-person shooters),* contoh : Crysis.
   3. *Racing,* contoh : Burnout Paradise.
   4. *Fighting,* contoh : Tekken 6.

2. *Adventure*

   Sebuah *game* petualangan bukanlah suatu kompetisi atau simulasi. Sebuah *game* petualangan juga tidak menawarkan proses untuk dikelola atau lawan harus dikalahkan melalui strategi dan taktik. Sebaliknya, sebuah game petualangan adalah sebuah cerita interaktif tentang karakter yang dikendalikan oleh pemain.

3. *Action Adventure*

   Adalah paduan antara *action* dan *adventure games.* Yaitu, kecepatan, *real time action* dari sebuah *action games* dan unsur cerita serta teka-teki dari sebuah *adventure games.*Contoh : God of War II , Legend of Zelda : Ocarina of Time.

4. *Role Playing*

   Pemain mengendalikan satu karakter atau sekelompok kecil karakter. Pemain sering membentuk ikatan erat dengan karakter mereka terutama jika itu *single character* (avatar). Karakter mengalami kemajuan

dalam sebuah permainan bila karakter mendapatkan pengalaman dari sebuah pertempuran maupun tantangan. Pengalaman ini membuat sebuah karakter mendapatkan *skills* dan *abilities.*Contoh : Star Wars, Dragon Age, Elder Scrolls IV.

5. *Simulation*

   Simulation games adalah permainan yang mendekati kehidupan nyata. *Video Game* jenis ini seringkali memperhatikan dengan detil berbagai faktor. Dari mencari jodoh dan pekerjaan, membangun rumah, gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. Dunia kehidupan rumah tangga sampai bisnis membangun konglomerasi, dari jualan limun pinggir jalan hingga membangun laboratorium cloning. *Video Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas.

6. *Strategy*

   Game jenis ini menghibur melalui penalaran dan pemecahan masalah. Terkadang genre ini tidak menggunakan banyak *storytelling.*

2.2.5 Game Arcade (Genre Game Terkait)

Game Arcade lebih dikenal dengan sebutan *ding-dong. Game* arcade adalah game yang mengandalkan ketangkasan pemainnya dalam memegang kontrol. Ciri-ciri *game arcade* adalah mempunyai level yang singkat, kontrol yang mudah, serta tingkat kesulitan yang bertambah dengan cepat. Jenis *game* ini dirancang untuk memancing adrenalin pemain serta tidak membutuhkan jalan cerita yang bagus. *Game* seperti ini membutuhkan waktu belajar yang relatif singkat. Sifat dari *game* arcade adalah real-time.

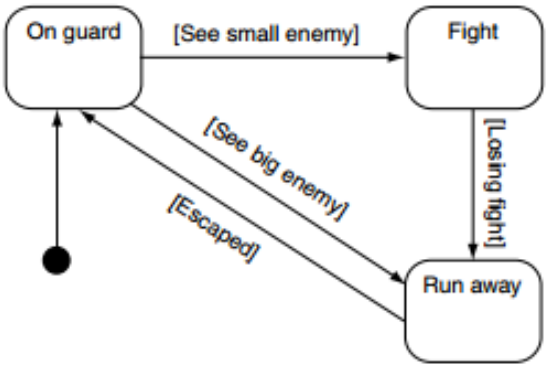
2.3 Algoritma

Algoritma adalah urutan langkah-langkah untuk memecahkan suatu masalah. Berikut ini adalah beberapa contoh dan penjelasan dari algoritma.[12]

2.3.1 Finite State Machine [4]

Ada beberapa definisi mengenai Finite State Machine (FSM) atau sering juga disebut dengan Finite State Automata (FSA).

1. FSM dapat diartikan sebagai perangkat komputasi yang memiliki input berupa string dan output yang merupakan satu dari dua nilai yang dapat diaccept dan reject.
2. Finite Automata adalah model matematika sistem dengan masukan dan keluaran diskrit. Sistem dapat berada di salah satu dari sejumlah berhingga konfigurasi internal disebut state.
3. FSM adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: State (Keadaan), Event (kejadian) dan action (aksi). Pada beberapa waktu, system akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks.



Gambar II-1 Contoh Diagram State [3]

Pada diagram state II.1 tersebut memperlihatkan FSM dengan dua buah state dan dua buah input serta empat buah aksi output yang berbeda : seperti terlihat pada gambar, ketika ada event yang terjadi karakter hanya akan melakukan satu aksi setiap statenya, karakter akan melakukan perlawan jika

melihat musuh yang kecil, sedangkan jika karakter kalah dalam pertarungan dia akan melarikan diri.

Secara formal FSM dinyatakan oleh 5 tupel atau M=(Q, Σ, δ, S, F), dimana:

Q = himpunan state/kedudukan

Σ = himpunan symbol input/masukan/abjad

δ = fungsi transisi

S = state awal/ kedudukan awal (initial state), S Q

F = himpunan state akhir, F Q

FSM terdiri dari dua jenis, yaitu FSM ber-output dan FSM tidak beroutput. FSM tidak ber-output digunakan untuk pengenalan bahasa dalam komputer, dengan input yang dimasukkan akan diperoleh apakah input tersebut dikenal oleh bahasa komputer atau tidak. Salah satu penggunaan FSM tidak beroutput adalah program compiler, yaitu program untuk memeriksa apakah perintah yang digunakan pengguna benar atau salah. Sementara untuk FSM ber-output digunakan untuk merancang mesin atau sistem.

2.3.2 Fuzzy Logic

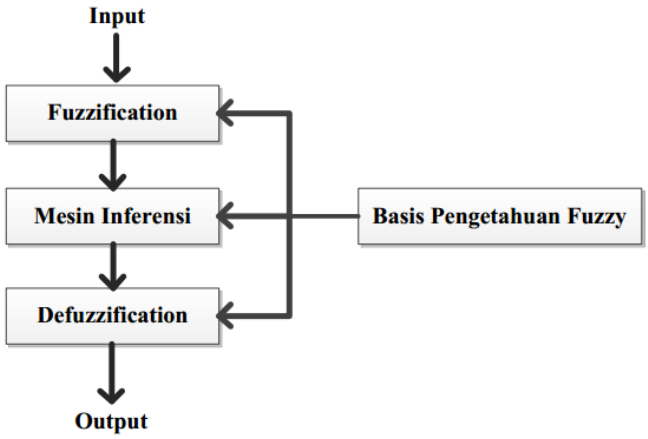
Konsep tentang *fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika fuzzy adalah metodologi sistem control pemecahan masalah, yang cocok diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akurasi data dan sistem control. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, dan lain-lain. Oleh karena itu semua ini mempunyai nilai keanggotaan 0 dan 1. Akan tetapi, dalam logika fuzzy memungkinkan nilai keanggotaan berada diantara 0 dan 1.

Konsep himpunan fuzzy memiliki 2 atribut, yaitu [9] :

1. Linguistik, yaitu nama suatu kelompok yang me wakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya dingin, sejuk, panas mewakili variable temperatur.
2. Numeris, yaitu suatu nilai yang menunjukan ukuran dari suatu variabel,

misalnya 10, 35, 40 dan sebagainya.

Cara kerja logika fuzzy dapat digambarkan sebagai struktur elemen dasar sistem inferensi fuzzy seperti pada gambar II.2:



Gambar II-2 Struktur Sistem Inferensi Fuzzy

Keterangan :

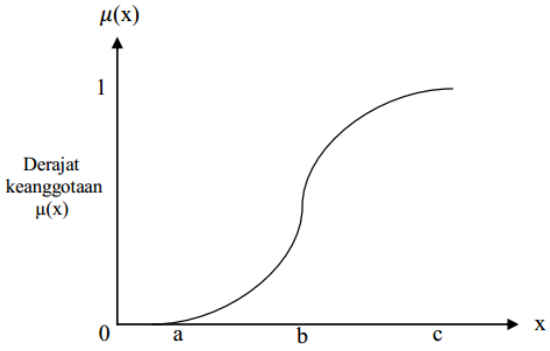
1. Basis Pengetahuan Fuzzy merupakan kumpulan rule-rule fuzzy dalam bentuk pernyataan IF…THEN.
2. Fuzzyfikasi adalah proses untuk mengubah input sistem yang mempunyai nilai tegas menjadi variabel linguistic menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan fuzzy.
3. Mesin Inferensi merupakan proses untuk mengubah input fuzzy dengan cara mengikuti aturan-aturan (IF -THEN Rules) yang telah ditetapkan pada basis pengetahuan fuzzy.
4. DeFuzzyfikasi merupakan proses mengubah output fuzzy yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzyfikasi.

2.3.2.1 Fungsi Keanggotaan (Member function)

Pada fuzzy system, fungsi keanggotaan memainkan peran yang sangat penting untuk merepresentasikan masalah dan menghasilkan keputusan yang akurat. Terdapat banyak sekali fungsi keanggotaan yang dapat digunakan, diantaranya
adalah [10] :

1. Fungsi sigmoid

Sesuai dengan namanya, fungsi ini berbentuk kurva sigmoidala seperti huruf S. Setiap nilai x (anggota crips set) dipetakan ke dalam interval [0,1]. Grafik dan notasi matematika untuk fungsi ini adalah sebagai berikut :



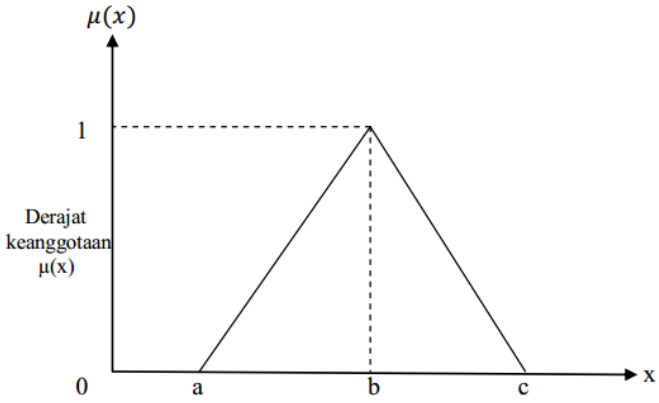
Gambar II-3 ***Grafik Fungsi Sigmoid***

Fungsi keanggotaan :

$$Sigmoid(x,a,b,c) = \begin{cases} 0;\ x \le a \\ 2(x-a)/(c-a)^2; a \le x \le b \\ 1-2((c-x)/(c-a))^2; b \le x \le c \\ 1; x \ge c \end{cases} \quad \ldots\ldots\ldots\ldots (2.1)$$

1. Fungsi Segitiga

Fungsi segitiga hanya terdapat satu nilai x yang memiliki derajat kenanggotaan sama dengan 1, yaitu ketika x = b. Tetapi, nilai-nilai disekitar b memiliki derajat keanggotaan yang turun cukup tajam (menjauhi 1) . Grafik dan notasi matematika dari fungsi segitiga adalah seperti pada gambar II.4 :



Gambar II-4 ***Grafik Fungsi Segitiga***

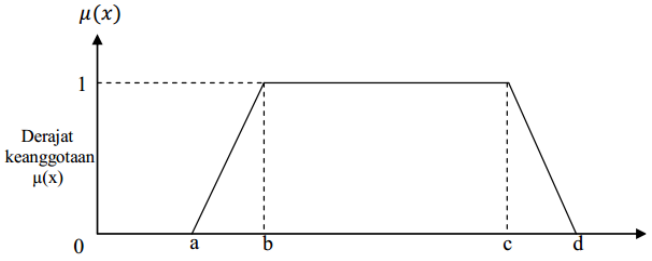
Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0;\ x \leq a\ atau\ x\ \geq c \\ (x-a)/(b-a)\,;\, a \leq x \leq b \\ (b-x)/(c-b)\,;\, b \leq x \leq c \end{cases}$$

…… (2.2)

2. Fungsi Trapesium

Berbeda dengan fungsi segitiga, pada fungsi ini terdapat beberapa nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika $b \leq x \leq c$. Tetapi, derajat keanggotaan untuk $c \leq x \leq d$ memiliki karakteristik yang sama dengan fungsi segitiga. Grafik dan notasi matematika dari fungsi Trapesium adalah seperti pada gambar II.5 :



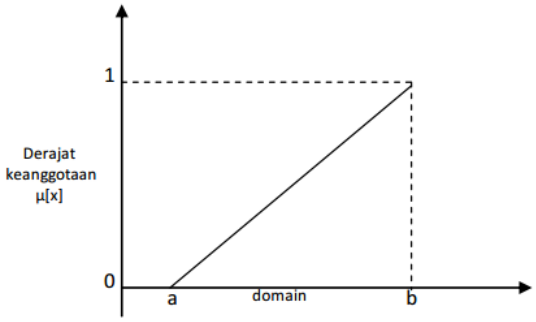
Gambar II-5 ***Grafik Fungsi Trapesium***

Fungsi Keanggotaan :

$$\mu(x) = \begin{cases} 0; & x \le a \text{ atau } x \ge d \\ (x-a)/(b-a); & a \le x \le b \\ 1; & \\ (d-x)/(d-c); & b \le x \le c \end{cases} \quad \ldots\ldots\ldots (2.3)$$

3. Fungsi Linear

Pada fungsi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk kurva ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Keadaan tersebut digambarkan seperti pada gambar II.6.



Gambar II-6 ***Representasi Linear Naik[10]***

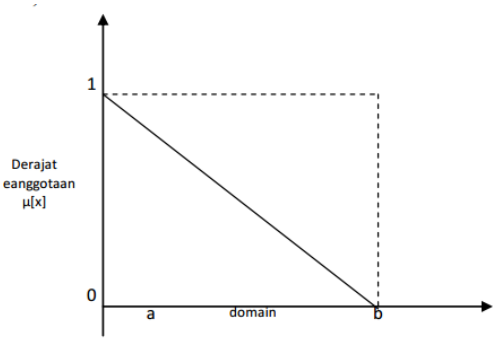
Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x < a \\ (x-a)/(b-a); & a < x \le b \\ 1; & x \ge b \end{cases}$$

……..(2.4)

Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian

bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Keadaan tersebut digambarkan seperti pada gambar II.7.



Gambar II-7 ***Representasi Linear Turun[10]***

Fungsi keanggotaan :

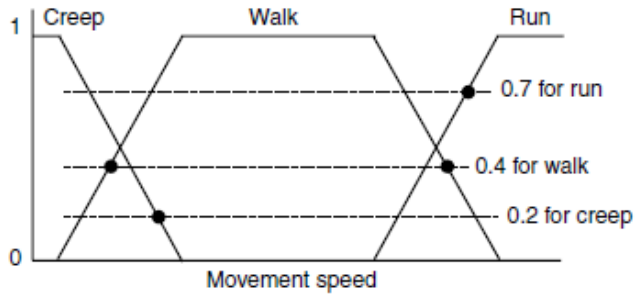
$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \le x < b \\ 0; & x \ge b \end{cases}$$

……..(2.5)

2.3.2.2 Defuzzifikasi

Setelah menerapkan apa logika fuzzy yang dibutuhkan, biarkan dengan seperangkat keanggotaan nilai untuk fuzzy set. Untuk mengubahnya kembali menjadi data yang berguna, perlu menggunakan sebuah teknik defuzzifikasi. Teknik fuzzifikasi melihat di bagian terakhir adalah cukup jelas dan hampir di mana-mana.. Ada beberapa teknik defuzzifikasi mungkin, dan tidak ada konsensus yang jelas mengenai mana yang terbaik untuk digunakan. Semua memiliki struktur dasar yang sama, tetapi berbeda dalam efisiensi dan stabilitas hasil. Defuzzifikasi melibatkan mengubah seperangkat nilai keanggotaan menjadi output tunggal nilai. Nilai output hampir selalu nomor. Hal ini bergantung pada memiliki satu set fungsi keanggotaan untuk nilai.Kami output mencoba untuk membalikkan fuzzifikasi yang Metode: untuk menemukan nilai output yang akan mengakibatkan nilai-nilai keanggotaan yang dimiliki. Pada Gambar selanjutnya, kita memiliki nilai keanggotaan 0,2, 0,4, dan 0,7 untuk fuzzy set "merayap",

“berjalan," dan "lari." Fungsi keanggotaan menunjukkan bahwa tidak ada nilai yang mungkin untuk gerakan kecepatan yang akan memberikan nilai-nilai keanggotaan, jika memberi ke dalam fuzzifikasi.



Gambar II-8 ***Defuzzyfikasi [3]***

2.4 Object Oriented Programming (OOP)

OOP adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metode OOP didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode ini meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemograman berorientasi objek, dan pengujian berorientasi objek. [5]Ketika sebuah program berjalan, objek individu biasanya tidak dapat berdiri sendiri.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut.
2. Kecepatan pengembangan

   Karen sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi

Karena sifat pewarisan dan penggunaan notasi yang sama pasa saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistenso pada saat pengembangannya, perangkat lunak dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.[5]

Konsep dasar dari sebuah OOP adalah sebagai berikut :

1. Objek, yaitu benda di dunia nyata yang bisa dibedakan antara satu dengan yang lainnya. Objek mempunyai identitas, property, dan tingkah laku.
2. Kelas,yaitu sekumpulan objek yang memiliki kemiripan dalam hal properti, atribut, *behaviour*, dan semantik.
3. Abstraksi, yaitu mekanisme yang memungkinkan situasi dunia nyata dapat diwakili menggunakan model yang disederhanakan. Orientasi objek abstrak dunia nyata didasarkan pada objek dan interaksi antar objek lainnya.
4. Encapsulasi dikenal sebagai informasi yang tersembunyi juga berperan pada pembungkusan properti dan *behaviour*. Enkapsulasi ini memastikan pengguna sebuah objek tidak dapat mengganti keadaan dari sebuah objek dengan cara yang tidak layak.
5. Polimorfisme, yaitu karakteristik sistem orientasi objek. Ketika warisan digunakan untuk memperpanjang *class* umum ke *class* yang lebih khusus, biasanya akan mencakup beberapa perilaku dari *class* umum. Perilaku dari *class* khusus ini akan berbeda dengan *class* umum, tetapi nama yang digunakan untuk mendefinikan perilaku tersebut akan menjadi sama.
6. Inheritas, yaitu mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dariobjek yang sudah ada objek-objek ini dapat membagi dan

memperluasperilaku mereka tanpa haru mengimplementasi ulang perilaku tersebut(bahasa berbasis-objek tidak selalu memiliki inheritas).

2.5 Unified Modeling Language (UML)

UML adalah salah satu bentuk *language* atau bahasa untuk menggambarkan *object oriented analysis and design.* Sebagian besar metode terdiri bahasa dan proses. UML disebut bahasa pemodelan bukan metode. Bahasa pemodelan adalah notasi (terutama grafis) bahwa metode digunakan untuk mengekspresikan desain. Logo dari UML tersebut seperti pada gambar II.9.



Gambar II-9 ***Logo UML***

2.5.1 Sejarah UML

Bahasa pemograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemograman ini kurang berkembang dan dikembangkan lebih lanjut, namun kemunculannya menjadi titik besar bagi developer pengembang bahasa pemograman berorientasi objek selanjutnya.

Karena banyaknya metodologi yang berkembang pesat pada saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuatlah bahasa gabungan dari beberapa konsep seperti *Object Modelling Technique (OMT)* dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivan Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML).* [6]

2.5.2 Diagram UML

Menggunakan berbagai macam diagram dengan fungsi masing-masing untuk menggambarkan setiap proses dari sistem berorientasi objek. Berikut merupakan beberapa diagram UML diantaranya:

1. *Use Case Diagram*

   Use Case atau diagram *use case* merupakan pemodelan yang digunakan untuk menggambarkan kelakuan (*behavior)* dari sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

   1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.
   2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling vertukar pesan antarunit atau actor.

2. *Activity Diagram*

   Diagram aktivitas atau *activity* diagram adalah sebuah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Dalam diagram aktivitas yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

   Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

   1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis system yang didefinisikan
   2. Urutan atau pengelompokan tampilan dari sistem/*user interface* di mana setiap aktivitas dianggap memiliki sebuah rancangan antaruka tampilan.

3. Rancangan pengunjian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

3. *Class Diagram*

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

4. *Sequence Diagram*

Diagram sekuen adalah diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki prose situ sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

2.6 Adobe Illustrator (AI)

Adobe Illustrator adalah salah satu *software* editor yang berbasis vektor. Dalam pembuatan desain ada dua macam jenis yaitu desain berbasis vector dan desain berbasis bitmap. Desain berbasis vektor ini memiliki kelebihan yaitu bila gambar ditarik sebesar apapun tidak akan mengalami penurunan kualitas (pecah).

2.7 Construct 2

Construct 2 adalah salah satu editor *game* dengan bahasa pemograman HTML5. *Game* yang dibuat dengan construct 2 dapat dijalankan di browser yang mendukung bahasa HTML 5, seperti android, windows 8, facebook, windows phone, dll. Software ini dirancang untuk pembuatan dan pengembangan video game. Dalam pembuatan sebuah *game* memakai construct 2, pengguna tidak harus

memiliki skill coding atau programming. Karena construct 2 telah menyediakan berbagai macam *behavior*, termasuk salah satunya adalah *physics*(fisika).