BAB 2

LANDASAN TEORI

2. 1 Pengertian Merawat Tanaman

Pemeliharaan tanaman secara umum mencakup segala kegiatan yang berkaitan dengan upaya menjaga kelangsungan hidup tanaman agar tetap hidup sehat dan memiliki produktivitas tinggi. Kegiatan yang dikerjakan pada pemeliharaan tanaman dilakukan dengan menggunakan tenaga manusia. Suatu bentuk tindakan rutin dalam memelihara tanaman melakukan penyiraman, pemupukan secara rutin juga konsisten. Sayuran adalah bagian tanaman yang dikonsumsi beserta makanan utama. Bagian tanaman yang dikonsumsi bisa bagian daun, akar, batang, dan buah muda. Pada daun, komposisi air dan mineral sangat tinggi namun mengandung sedikit energi. Bagian akar dan biji mengandung energi dan pati yang tinggi. Sayur mayur termasuk ke dalam golongan tumbuhan kortikulutura. Tumbuhan tersebut memiliki usia < 1 tahun. Beberapa variasi pada sayuran (warna, aroma, rasa, kekerasan dan sebagainya) membuat peningkatan selera makan. Sayur mayur merupakan sumber serat, vitamin A dan C serta mineral yang dibutuhkan oleh tubuh. Tanaman sayuran yang dapat ditanam dengan mudah di dalam pot tanam.

2.1.1 Seledri

Seledri juga dikenal dengan nama daun sop, adalah tanaman sayuran yang juga banyak dipakai sebagai tanaman obat. Seledri dapat tumbuh pada dataran rendah sampai tinggi. Perbanyakan seledri bisa dengan biji atau anakan. Seledri termasuk tanaman ang mudah perawatannya. Untuk skala rumah tangga seledri juga bisa berfungsi sebagai tanaman hias dengan menanamnya dalam pot, atau pada kotak memanjang seperti talang plastik. Media tanaman yang cocok adalah campuran tanah subur, pupuk kandang dan serbuk gergaji/sekam dengan perbandingan 1:1:1. seperti pada Gambar 2.1 Seledri



Gambar 0.1Seledri

2.1.2 Daun bawang

Merupakan jenis sayuran dari kelpompok bawang. Hanya ada satu batang dari tanah kira – kira sekitar 5cm, biasanya terdapat 2 atau lebih daun yang keluar dari batang dengan bentuk tegak memanjang dan ruang didalam daun. Benih bawang daun dapat berasal dari biji atau dari tunas anakan (stek tunas). Tunas anakan diperoleh dengan cara memisahkan anakan yang sehat dan bagus pertumbuhannya dari induknya.Tanaman daun bawang sangat mudah perawatannya sehingga cocok ditanam di dalam pot. seperti pada Gambar 2.2 Daun Bawang.



Gambar 0.2Daun Bawang

2.1.3 Tomat

Tumbuhan yang berasal dari Amerika Selatan dan Tengah merupakan tumbuhan dengan siklus hidup singkat. Termasuk kedalam kelompok sayuran buah dengan kisaran warna hijau ketika masak, kuning, dan merah. Buah berbentuk bulat pipih bahkan bulat lonjong. seperti pada Gambar 2.3 Tomat



Gambar 0.3Tomat

2.2 Pengertian Game

Dalam bahasa Indonesia "*Game*" berarti "permainan". Permainan yang dimaksud dalam *game* juga merujuk pada pengertia n sebagai "kelincahan intelektual " (*intellectual playability*). Sementara kata "*game*" bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran

sejauh mana *game* itu menarik untuk dimainkan secara maksimal. Dahulu istilah *game* identik dengan anak-anak selaku pemain. Yang ada adalah suatu kegiatan yang dilakukan oleh anak-anak yang menurut mereka itu dapat menyenangkan hati mereka. Segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai *game*.

2.2.1 Jenis Game

Berdasarkan jenis atau klasifikasi*game* dapat diklasifikasikan sebagai berikut :

1. *Shooting* (tembak-tembakan)

   Adalah *game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga *timing*, inti dari *game* jenis ini adalah tembak, tembak dan tembak. Contohnya *GTA, Call Of Duty*.

2. *Fighting* (Pertarungan)

   Adalah g*ame* yang permainannya memerlukan refleks dan koordinasi mata dan tangan dengan cepat, tetapi inti dari *game* ini adalah penguasaan hafalan jurus. Contohnya Mortal Kombat dan Tekken.

3. Petualangan *(Adventure)*

   Game yang lebih menekankan pada jalan cerita dan kemampuan berfikir pemain dalam menganalisia tempat secara visual, memecahkan teka teki maupun menyimpulkan berbagai peristiwa. Contohnya Kings Quest, dan Space *Quest.*

4. Simulasi, Konstruksi, Manajemen

   Game jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Contohnya The Sims

5. Strategi

   Game jenis ini memerlukan koordinasi dan strategi dalam memainkan permainan ini. Kebanyakan *game* stategi adalah *game* perang. Contohnya Warcraft.

6. Olahraga (*Sport*)

Game ini merupakan adaptasi dari kenyataan, membutuhkan kelincahan dan juga strategi dalam memainkannya. Contohnya Winning Eleven dan NBA.

7. Puzzle

Game teka-teki, pemain diharuskan memecahkan teka-teki dalam *game* tersebut. Contohnya Tetris, Minesweeper dan Bejeweled.

8. Edukasi

Game jenis ini dibuat dengan tujuan spesifik sebagai alat pendidikan, untuk belajar mengenal warna untuk balita, mengenal huruf dan angka, matematika, sampai belajar bahasa asing. *Developer* yang membuatnya, harus memperhitungkan berbagai hal agar *game* ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan design visual ataupun animasinya. Contohnya *The farmer, Dora the explorer*.

2.3 Game Edukasi

Game edukasi adalah *game* digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interatif.

2.3.1 Kriteria *Game* Edukasi

Perancangan *Education game* yang baik haruslah memenuhi kriteria dari *education game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education game,* yaitu

1. Nilai Keseluruhan *(Overall Value)*

Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur *timer.*

2. Dapat Digunakan *(Usability)*

Mudah digunakan dan diakses adalah poin penting bagi pembuat *game.* Apliksi ini merancang sistem dengan *interface* yang *user friendly* sehingga*user* dengan mudah dapat mengakses aplikasi.

3. Keakuratan *(Accuracy)*

Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model game pada tahap perencanaan.

4. Kesesuaian *(Appropriateness)*

Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan *user* dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan *(Relevance)*

Relevan artinya dapat mengaplikasikan isi *game* ke target *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak-anak maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Objektifitas *(Objectives)*

Objektifitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikas ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik *(Feedback)*

Untuk membantu pemahaman *user* bahwa permainan *(performance)* merekasesuai dengan objek *game* atau tidak, *feedback* harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

2.4 *LogikaFuzzy*

Menurut Asus Naba, logika *fuzzy* adalah: “Sebuah metodologi berhitung dengan variabel kata-kata (*linguistic variable*) sebagai pengganti berhitung

dengan bilangan. Kata-kata digunakan dalam *fuzzy logic* memang tidak sepresisi bilangan, namun kata-kata jauh lebih dekat dengan intuisi manusia.

Mengenai *logika fuzzy* pada dasarnya tidak semua keputusan dijelaskan dengan 0 atau 1, namun ada kondisi diantara keduanya, daerah diantara keduanya inilah yang disebut dengan *fuzzy* atau tersamar. Secara umum ada beberapa konsep sistem logika *fuzzy* , sebagai berikut dibawah ini:

a. Himpunan tegas yang merupakan nilai keanggotaan suatu item dalam suatu himpunan tertentu.

b. Himpunan *fuzzy* yang merupakan suatu himpunana yang digunakan untuk mengatasi kekakuan dari himpunan tegas.

c. Fungsi keanggotaan yang memiliki interval 0 sampai 1

d. Variabel *linguistic* yang merupakan suatu variabel yang memiliki nilai berupa kata-kata yang dinyatakan dalam bahasa alamiah dan bukan angka.

e. Operasi dasar himpunan *fuzzy* merupakan operasi untuk menggabungkan dan atau memodifikasi himpunan *fuzzy*.

f. Aturan (*rule*) *if-then fuzzy* merupakan suatu pernyataan *if-then*, dimana beberapa kata-kata dalam pernyataan tersebut ditentukan oleh fungsi keanggotaan.

Dalam proses pemanfaatan logika *fuzzy*, ada beberapa hal yang harus diperhatikan salah satunya adalah cara mengolah input menjadi output melalui sistem *inferensi fuzzy*. Metode *inferensi fuzzy* atau cara merumuskan pemetaan dari masukan yang diberikan kepada sebuah keluaran. Proses ini melibatkan fungsi keanggotaan, operasi logika, serta aturan *IF-THEN*. Hasil dari proses ini akan menghasilkan sebuah sistem yang disebut dengan *FIS (Fuzzy InferensiSystem)*. Dalam logika *fuzzy* tersedia beberapa jenis *FIS* diantaranya adalah *Mamdani*, *Sugeno*, dan *Tsukamoto*.

2.4.1 *Logika Fuzzy Sugeno*

Logika *fuzzy* yang pertama kali diperkenalkan oleh Lotfi A. Zadeh, memiliki derajat keanggotaan dalam rentang 0(nol) hingga 1(satu), berbeda

dengan logika digital yang hanya memiliki dua nilai yaitu 1(satu) atau 0(nol). Proses *fuzzy inference* dapat dibagi dalam lima bagian, yaitu:

1. *Fuzzyfikasi Input*

   FIS mengambil masukan-masukan dan menentukan derajat keanggotaannya dalam semua *fuzzy* set.

2. Operasi *logika fuzzy*

   Hasil akhir dari operasi ini adalah derajat kebenaran *antecedent* yang berupa bilangan tunggal.

3. *Implikasi*

   Merupakan proses mendapatkan *consequent* atau keluaran sebuah IF-THEN *rule* berdasarkan derajat kebenaran *antacedent*. Proses ini menggunakan mengambil nilai MIN/terkecil dari dua bilangan : Hasil operasi *fuzzy logic* OR dan *fuzzy set* banyak.

4. *Agregasi*

   Yaitu proses mengkombinasikan keluaran semua IF-THEN *rule* menjadi sebuah *fuzzy set* tunggal. Pada dasarnya agregasi adalah operasi *fuzzy logic* OR dengan masukannya adalah semua *fuzzy set*.

5. *Defuzzyfikas*i

   Keluaran dari *defuzzy*fikasi adalah sebuah bilangan tunggal, cara mendapatkannya ada beberapa versi, yaitu *centroid, bisector, middle of maximum, largest of maximum* dan *smallest of maximum.*

2.5 Himpunan*Fuzzy*

Pada himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A,yang sering ditulis dengan $\mu_A[x]$, memiliki 2 kemungkinan yaitu:

Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan atau Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan

Disini bisa dikatakan bahwa pemakaian himpunan crisp untuk menyatakan umur sangat tidak adil, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yamg sangat significant

Himpunan Fuzy digunakan untuk mengantisipasi hal tersebut. Seseorang dapat masuk dalam 2 himpunan yang berbeda. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya.

Kalau pada himpunan crisp, nilai keanggotaan hanya ada 2 kemungkinan yaitu 0 dan 1, pada himpunan Fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan Fuzzy $\mu_A[x] = 0$ berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai keanggotaan Fuzzy $\mu_A[x] = 1$ berarti x menjadi anggota penuh pada himpunan A.

Terkadang kemiripan antara keanggotaan Fuzzy dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval [0,1], namun interpretasi nilainya sangat berbeda antara kedua kasus terebut. Keanggotaan Fuzzy memberikan suatu ukuran terhadap pendapat dan keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan suatu himpunan Fuzzy Muda adalah 0,9; maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 muda berarti 1 0% dari himpunan tersebut diharapkan tidak muda.

Himpunan *Fuzzy* memiliki 2 atribut, yaitu:

1. Linguistic,yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi teretentu dengan menggunakan bahasa alami
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu *variabel* seperti: 40, 30, 29.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *Fuzzy*, yaitu:

1. Variabel Fuzzy

Variabel Fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem *Fuzzy*. Contoh: umur, temperature, permintaan, dan sebagainya.

2. Himpunan *Fuzzy*

Himpunan *Fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu *variabel Fuzzy*.

Contoh:

Variabel umur dibagi menjadi 3 himpunan *Fuzzy*, yaitu: muda, parobaya, dan tua.

Variabel temperature, terbagi menjadi 5 himpunan *Fuzzy*, yaitu: dingin, sejuk, normal, hangat, dan panas.

3. Semesta pembicaraan

   Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu *variable Fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh:

Semesta pembicaraan untuk variabel umur: [0+~]

Semesta pembicaraan untuk variabel temterature: [0 40]

4. Domain

   Domain himpunan Fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan Fuzzy. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa nail (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

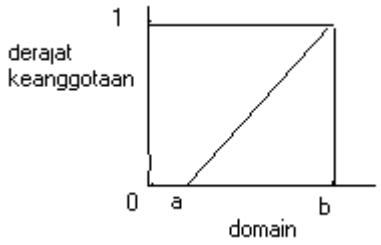
2.6 Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan.

1. Representasi linier

Pada representasi linier, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada dua keaadaan himpunan Fuzzy yang linier. Pertama, kenaikan himpunan dimulai pda nilai domain yang memiliki derajat keanggotaan nol [0] bergerak kekanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi atau sering disebut dengan representasi kurva linear naik. Gambar 2.4 menunjukan Repersentasi kurva naik

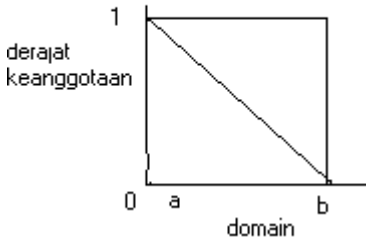


Gambar 0.4Representasi kurva naik (9)

Fungsi keanggotaan

$$\mu(x)=\begin{cases}0; & x \le a \\ (x-a)/(b-a); & a \le x \le b \\ 1; & x \ge b\end{cases} \quad (2.1)$$

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 atau sering disebut dengan kurva linear turun. Gambar 2.5 menunjukan representasi kurva turun



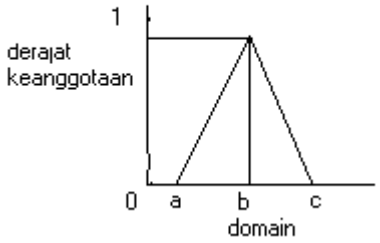
Gambar 0.5Representasi kurva turun (9)

Fungsi keanggotaan:

$$\mu(x)=\begin{cases}(b-x)/(b-a); & a\le x\le b\\ 0; & x\ge b\end{cases} \tag{2.2}$$

2. Representasi kurva segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linier). Gambar 2.6 menunjukan representasi kurva segitiga



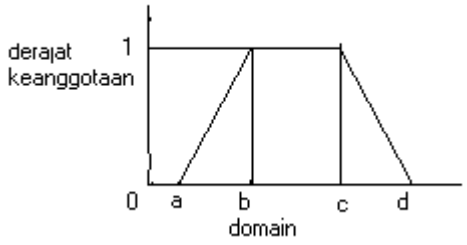
Gambar 0.6Representasi kurva segitiga (9)

Fungsi keanggotaan:

$$\mu[x]=\begin{cases}0; & x\le a\ atau\ x\ge c\\ (x-a)/(b-a); & a\le x\le b\\ (b-x)/(c-b); & b\le x\le c\end{cases} \tag{2.3}$$

3. Representasi kurva trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada titik yang memiliki nilai keanggotaan 1. Gambar 2.7 menunjukan repersentasi kurva trapezium



Gambar 0.7Representasi kurva trapezium (9)

Fungsi keanggotaan

$$\mu[x] = \begin{cases} 0; & x \le a \; atau \; x \ge d \\ (x-a)/(b-a); & a \le x \le b \\ 1; & b \le x \le c \\ (d-x)/(d-c); & x \ge d \end{cases} \qquad (2.4)$$

2.7 Operasi Himpunan Crisp

Pada logika tradisional, fungsi keanggotaan suatu himpunan terbagi atas dua daerah, yaitu:

$$\mu_A[x] = 0, \text{ jika } x \notin A \text{ atau} \qquad (2.5)$$

$$\mu_A[x] = 1, \text{ jika } x \in A$$

dengan kata lain, fungsi keanggotaan himpunan A bernilai nol (0), jika x bukan merupakan elemen dari himpunan A. Sebaliknya fungsi keanggotaan himpunan A akan bernilai satu (1) jika x merupakan anggota A. Keanggotaan himpunan crisp selalu dapat dikategorikan secara penuh tanpa adanya dikotomi atau ambiguitas. Pada himpunan crisp, ada empat operasi dasar.

Union dari himpunan A dan B ($A \cup B$) berisi semua elemen yang berada pada himpunan A atau himpunan B. interseksi dari himpunan A dan B ($A \cup B$) berisi semua element yang berada pada himpunan A atau pada himpunan B.

2.7.1 Interaksi Himpunan *Fuzzy*

Pada sisten crisp, interseksi antara dua himpunan yang berisi elemen-elemen yang berada pada kedua himpunan. Hal ini ekuivalen dengan operasi aritmatik atau logika AND. Pada logika fuzzy konvensional, operator AND diperlihatkan dengan derajat keanggotaan minimum antar kedua himpunan. Berikut adalah aturan Zadeh dasar untuk interseksi fuzzy, daerah antara dua himpunan ditentukan oleh aplikasi operasi tersebut:

$$\mu_{A\cap B} = \min(\mu_A[x], \mu_B[y]). \quad (2.9)$$

2.8 Pemodelan Berorientasi Objek

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunkan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam suatu entitas. Model berorientasi objek bermanfaat untuk memahami masalah, komunikasi dengan ahli aplikasi, pemodelan suatu organisasi, meyiapkan dokumentasi serta perancangan program dan basis data. Pertama-tama suatu model analisis dibuat untuk menggambarkan aspek dasar dari domain aplikasi, dimana model tersebut berisi objek yang terdapat dalam domain aplikasi termasuk deskripsi dari keterangan objek dan perilakunya.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP (*Object Oriented Programing*) mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat. Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut.

2.9 Karateristik Berorientasi Objek

Pengembangan sistem berorientasi objek mempunyai tiga karateristik utama, yaitu :

1. *Encapsulation*

Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas dalam 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 atribut dan metoda dari induknya langsung. Atribut dan metoda dari objek induk diturunkan kepada anak objek demikian seterusnya. Pendefinisian objek dipergunakan untuk membangun suatu hirarki dari objek turunannya, sehingga tidak perlu membuat atribut dan metode lagi pada anaknya, karena telah mewarisi sifat induknya.

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.

2.10 UML (Unified Modeling Language)

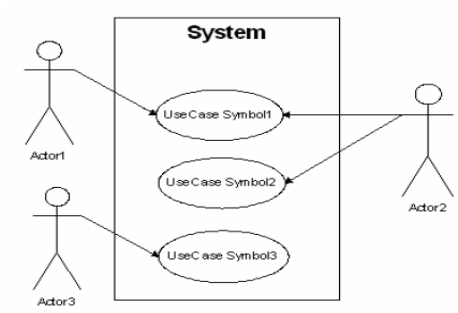
UML merupakan bahasa pemodelan dimana unsur – unsur dan aturan – aturan yang dimilikinya berfokus pada persentasi konseptual dan fisikal sistem. Unsur-unsur dan aturan-aturan tersebut dapat digunakan untuk merancang dan membaca model objek.

Keungulan UML sebagai berikut :

A. UML sebagai bahasa visualisasi digunakan untuk merancang suatu model yang dapat dibaca oleh banyak orang dengan pegertian yang sama.
B. UML bahasa pendefinisian digunakan untuk mendefinisikan dengan rinci seluruh hasil analisis, desain, dan implementasi yang harus dilakukan dalam pengembangan sistem.
C. UML sebagai bahasa dokumentasi digunakan untuk mendokumentasikan arsitektur beserta perinciannya, unsur-unsur yang dibutuhkan dalam pengembangannya. Serta perencanaan dan implementasi proyek secara keseluruhan dengan simbol-simbol yang mudah dimengerti.

2.10.1 Use Case Diagram

Secara grafik menggambarkan interaksi antara sistem, hal-hal diluar sistem,dan *user*. Dengan kata lain, mereka mengambarkan siapa yang akan memakai sistem dan dengan cara itu seorang *user* akan berinteraksi dengan sistem. Use-case merupakan urutan langkah yang secara tindakan saling terkait (Skenario), baik termotivasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal.



Gambar 0.8Contoh Diagram Use-case

Notasi yang digunakan dalam *use-case* diagram antara lain :

1. *Actor*

   Adalah posisi yang dimiliki user terhadap sistem. Yang menjadi *actor* dapat berupa manusia, hardware atau sistem lain yang berhubungan dengan sistem yng berjalan. Atau *actor* adalah sesuatu yang memerlukan interaksi dengan sistem untuk bertukar informasi.

2. *Use – case* simbol

   Menggambarkan sekumpulan urutan, dimana setiap urutan mewakili intervensi antara *actor* dengan sistem. *Use-case* symbol mempresentasikan fungsionalitas sistem secara keseluruhan.

2.10.2 Deskripsi *Use-case*

Bagian umum kejadian *use-case* adalah deskripsi *use-case,* yaitu langkah demi langkah mulai dengan pelaku menginisialisasi *use-case* dan melanjutkan hingga akhir kejadian bisnis. Dimana dalam deskripsi *use-case* terdapat :

a. Nama *use-case (use-case),* merupakan nama dari suatu *use-case.*
b. Aktor (*actor),* pelaku yang berpartisipasi didalam suatu *use-case.*
c. Tujuan (*goal),* tujuan dari *use-case.*
d. Prakondisi (*precondition),* batasan pada keadaan sistem sebelum *use-case* dapat dieksekusi.
e. Ringkasan (*summary),* deskripsi ringkasan pendek yang berisi jumlah kalimat yang menunjukan secara garis besar tujuan *use-case* dan berbagai kegiatanya.
f. *Related use-case, use-case* yang saling berhubungan.
g. Langkah – langkah (*steps),* menjelaskan setiap langkah dari *use-case* dengan mengunakan dua kolom, dimana kolom sebelah kiri menunjukan aksi dari pelaku dan kolom sebelah kanan merupakan respon yang diberikan oleh sistem.

2.10.3 Activity Diagram

*Activity diagram*atau *diagram* aktivitas adalah sebuah *diagram* yang dapat digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah-langkah sebuah *use-case* atau logika *behavior object.* Diagram ini serupa dengan *flowchart,* dimana secara grafis diagram ini menggambarkan aliran sekuential dengan *flowchart* dimana diagram ini menggambarkan mekanisme kegiatan paralel

Oleh karena itu, diagram ini sangat berguna untuk memodelkan kegiatan yang akan dilakukan saat sebuah operasi dieksekusi dan untuk memodelkan hasil-hasil dari kegiatan ini, seperti memodelkan *event* yang menyebabkan *window* akan ditampilkan atau ditutup.

Dibawah ini merupakan simbol-simbol yang ada pada diagram aktivitas:

a. Initial node

Lingkaran yang diisi penuh merupakan awal mulainya dari diagram ini. Initial ode tidak harus ada, tetapi dengan mengunakanya membuat diagramnya lebih mudah untuk dibaca.

b. *Activity*

Segi empat bersudut tumpul menggambarkan kegiatan yang perlu dilakukan. Secara *activity* bisa secara fisik, seperti *inspect Forms*, atau secara elektronik, seperti *Display Create Student Screen.*

c. *Control Flow*

Panah yang ada di diagram, menggambarkan sasaran yang mengawa kegiatan.

d. *Fork*

Bar hitam dengan satu *flow* yang ke dalamnya dan beberapa flow lainnya meninggalkannya, merupakan bar sinkronissi dimana kegiatan dapat dilakukan secara paralel.

e. *Join*

Bar hitam dengan satu *flow* mengarah ke dalamnya dan ada satu flow yang meninggalkannya, merupakan bar sinkronisasi dimana beberapa aktivitas yang mempunyai tujuan yang sama dari keduannya digabung menjadi satu.

f. *Decicion*

Gambarsebuah wajik, menggambarkan sebuah kegiatan keputusan.

g. *Final Node*

Gambar lingkaran yang diisi penuh dan ada batas lagi di luarnya, menggambarkan akhir dari sebuah proses.

2.10.4 Class Diagram

Class diagram merupakan salah satu diagram struktur statis yang menggambarkan struktur hubungan antar kelas. *Class disgram* digunakan untuk mensimulasikan objek-objek dalam dunia nyata ke dalam sistem yang akan dibangun. Notasi UML pada *class diagram* adalah sebuah persegi yang dibagi menjadi 3 area, yaitu nama kelas, atribut, dan operasi (*method).*

Dalam sistem yang akan dibangun, kelas-kelas akan dihubungkan secara konseptual. Hubungan ini disebut asosiasi.

Class diagram dapat juga menggambarkan keanekaragaman (*multiplicity),* yaitu jumlah objek dari suatu kelas yang berhubungan dengan sebuah objek dari kelas yang berasosiasi.

2.10.5 Sequence Diagram

Sequance diagram digunakan terutama untuk menunjukan interaksi antar objek dalam urutan sekuensial. *Sequance diagram* sangat berguna untuk mengkomunikasikan bagaimana objek-objek berinteraksi dalam suatu proses bisnis. Analisis sistem umumnya mengunakan *sequance diagram* untuk memperjelas *use case.*

Sequance diagram terdiri dari objek-objek yang digambarkan dengan sebuah persegi yang memiliki nama. Objek-objek tersebut diletakan diatas garis hidup tersebut, ada kotak kecil memanjang yang dinamakan aktivasi. Aktivasi mempresentasikan eksekusi dari operasi yang objek lakukan.

Suatu objek dapat bertukar pesan dengan objek lain. Pesan tersebut digambarkan sebagai sebuah panah dari garis hidup suatu objek ke objek lain. Ada empat jenis pesan yang bisa digunakan, yakni pesan sederhana (*simple message),* pesan sinkron (*synchronous message),* pesan asinkron (*asynchronous message),* dan pesan nilai pengembalian (*retrun value).* Pesan sederhana digunakan untuk memindahkan kontrol dari satu objek ke objek lain. Pesan sinkron digunakan apabila suatu objek harus menunggu jawaban dari objek lain (yang dipanggil) terhadap pesan tersebut sebelum melanjutkan proses lainnya.

Pesan asinkron digunakan apabila suatu objek tidak perlu menunggu jawaban dari objek lain sebelum melanjutkan proses lainnya.

Sequance diagram merepresentasikan waktu dan arah *vertikal*, dari atas ke bawah. Pesan yang lebih atas menandakan bahwa pesan tersebut terjadi terlebih dahulu.

2.11 Adobe Photoshop CS4

Adobe Photoshop adalah *software* pengolah gambar yang sangat *powerfull* dengan segala fasilitasnya. Hasil gambar olah dengan Adobe Photoshop ini banyak dilihat di berbagai website, brosur, koran, majalah, dan media lainnya.

Adobe Photoshop adalah salah satu software pengolah gambar. *Software* ini digunakan untuk membuat (*create),* mengubah *(edit*) dan manipulasi gambar. *Software* ini merupakan *software* paling *powerfull* di dunia dalam pengelolaan gambar. Adobe Photoshop *compatible* dengan banyak jenis file gambar. Dan file outputnya pun support di berbagai *software.* Untuk *bundle Adobe* seri CS tentu saja sudah terasosiasi dengan baik antar software, misalnya ke Adobe After Effect, Adobe Premiere, Adobe Indesign, Adobe Illustrator, Adobe Flash dan Adobe Dreamweaver. Sedangkan software ex-CS yang sudah terasosiasi antara lain Adobe Director, CorelDraw, 3D Max dan lain-lain.

2.12 Unity 3D

Unity merupakan salah satu game engine yang banyak digunakan. Unity menyediakan fitur pengembangan game dalam berbagai platform, yaitu Unity Web, Windows, Mac, Android, iOS, XBox, Playstation 3 dan Wii.

Dalam unity disediakan berbagai pilihan bahasa pemrograman untuk mengembangkan game, antara lain JavaScript, C#, dan BooScript. Namun meskipun disediakan tiga bahasa pemrograman, kebanyakan developer menggunakan JavaScript dan C# sebagai bahasa yang digunakan untuk mengembangkan game mereka.

Unity Mensuport pembuatan game 2D dan 3D, namun lebih ditekankan pada 3D. Pengembangan game lebih ditekankan pada desain dan tampilan visual daripada pemrograman.



Gambar 0.9Gambar Unity 3D

2.13 Teknik Pengujian Perangkat Lunak

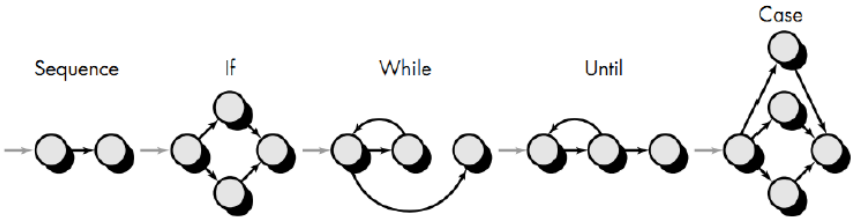
Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean[11].

2.13.1 Pengujian *White Box*

Pengujian *white box*, yang kadang-kadang disebut pengujian *glass box*, adalah metode desain *test case* yang menggunakan struktur kontrol desain procedural untuk memperoleh *test case* [7]. Dengan menggunakan metode pengujian *white box*, perekayasa sistem dapat melakukan *test case* sebagai berikut:

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*.
3. Mengeksekusi semua *loop* pada batasan dan pada batas operasional.
4. Menggunakan struktur data internal untuk menjamin validitasnya.

Pengujian *white box* yang berupa notasi diagram alir dapat dilihat pada gambar 2.10.



Gambar 0.10Notasi Diagram Alir [7]

2.13.2 Pengujian *Black Box*

Black-box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian *white-box*). pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional.

Perancang uji memilih input yang valid dan tidak *valid* dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan.

Metode ujicoba blackbox memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox*. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang

2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. kesalahan inisialisasi dan terminasi

2.13.3 Kuesioner

Kuesioner adalah daftar pertanyaan operasional yang ditanyakan pada responden terpilih untuk menjawab hipotesis-hipotesis yang dikembangkan sesuai tujuan penelitian. Pertanyaan-pertanyaan dalam kuesioner harus dapat mengumpulkan keterangan-keterangan responden yang diperlukan untuk menghasilkan indikator-indikator atau memenuhi rancangan tabulasi yang ingin dikaji.

Tahapan kegiatan Penelitian Statistik menentukan latar belakang dan permasalahan, menentukan tujuan dan menentukan kerangka konseptual atau kerangka pikir.

Teori memberikan penjelasan atas suatu gejala.Teori memberikan jawaban atas pertanyaan mengapa dan bagaimana.