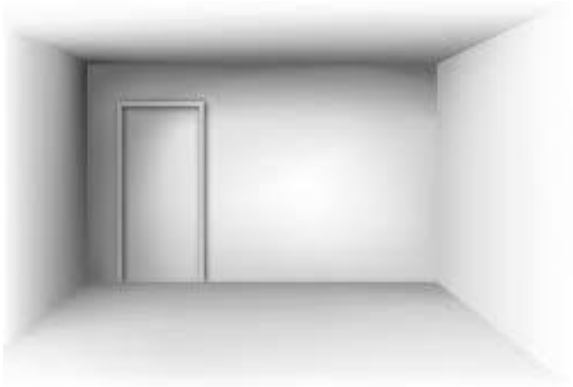
BAB 2
LANDASAN TEORI

2.1 Ruangan

Ruang menurut kamus besar bahasa indonesia (KBBI) adalah sela-sela antara dua (deret) tiang atau sela-sela antar empat riang (di bawah kolong rumah), rongga yang berbatas atau terlingkung oleh bidang, rongga yang tidak terbatas, tempat yang ada. Ruangan menurut kamus besar bahasa indonesia (KBBI) adalah tempat yang legar; kamar (besar); bilik (dalam rumah); kelas (tempat belajar).

Ruang menurut undang undang adalah wadah yang meliputi daratan, ruang lautan dan ruang udara sebagai satu kesatuan wilayah, tempat manusia dan mahluk lainnya hidup dan mahluk lainnya hidup dan melakukan kegiatan serta memelihara kelangsungan hidupnya (Pasal 1 butir 1 UU No.26/2007).

Ruang adalah bahan terpenting di mata seorang perancang dan unsur utama dalam desain interior. Melalui volume ruang, kita tidak hanya bergerak tetapi melihat juga bentuk-bentuk, mendengar berbagai suara, merasakan angin, dan matahari. Ruang mewarisi karateristik estetis dan sensual unru-unsur tersebut untuk bidangnya masing-masing [4]. Ruang adalah substansi materi, seperti batu dan kayu. Walapun demikian, ruang pada umumnya tidak berbentuk dan terdispersi. Ruang iniversil tidak mempunyai definisi. Pada saat suatu unsur diletakan pada suatu bidang, barulah hubungan visualnya terbentuk. Ketika unsur-unsur lain mulai diletakkan pada bidang tersebut, terjadilah hubungan majemuk antara ruang dan unsur-unsur tersebut maupun antara unsur yang satu dengan unsur lainnya. Ruang oleh karenanya terbentuk dari adanya hubungan-hubungan tersebut dan kita yang merasakannya.



Gambar 2.1 Ruang Kosong

Aktifitas manusia dalam kehidupannya dan pembangunan yang memanfaatkan ruang cenderung menimbulkan penurunan kualitas ruang, oleh karena itu mutlak diperlukan langkah-langkah konkrit penataan ruang melalui perencanaan, pemanfaatan dan pengendalian. Penataan ruangan adalah proses perencanaan tata ruang, pemanfaatan ruang dan pengendalian pemanfaatan ruang [3].

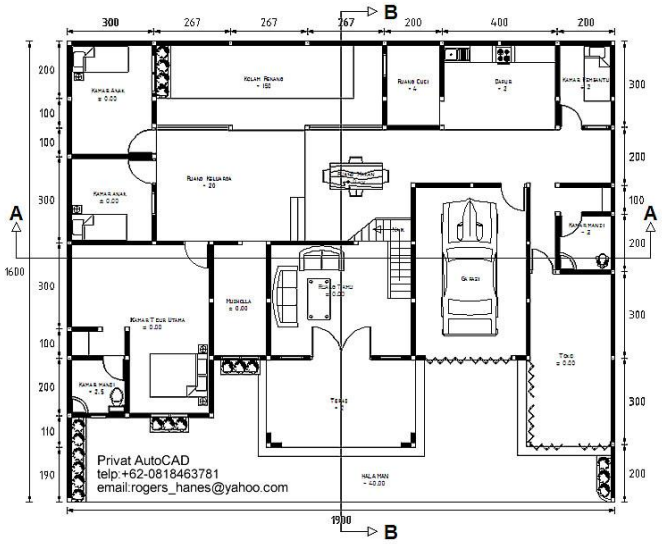
Sistem struktur bangunan mengatur bentuk dan pola dasar ruang-ruang interornya. Meskipun demikian, ruang-ruang tersebut pada akhirnya akan dibentuk strukturnya justru oleh elemen-elemen desain interior tersebut. Terminologi struktur di sini tidak diartikan sebagai penunjang fisik namun berkaitan dengan pemilihan dan penyusunan unsur-unsur interior sedemikian rupa sehingga hubungan-hubungan visual dapat menentukan dan mengatur ruang interior sebuah ruangan. Dalam sebuah ruang yang besar, bentuk dan tata letak perlengkapan ruang juga dapat berfungsi sebagai dinding, menimbulkan kesan terlindung dan membentuk pola-pola spasial. Bahkan sebuah elemen, bentuk skala atau gayanya dapat mendominasi suatu ruang dan mengorganisir munculnya daerah ruang elemen itu sendiri.

Permasalahan yang kompleks perlu diperhitungkan dalam upaya mewujudkan sebuah desain yang memberikan penyelesaian masalah secara integral. Dengan menggunakan metodologi desain yang sistematis maka upaya pemecahan masalahnya yaitu dapat dilakukan dengan mendeskripsikan permasalahan tersebut degan cata mendata secara lengkap untuk kemudian diuraikan dalam bentuk analisis masalah. Keberadaan sebuah konsep desain

dalam perencanaan interior sangatlah penting. Dengan adanya konsep maka seluruh permasalahan yang akan dipecahkan dalam perencangan diformulasikan ke dalam satu perumusan.

2.1.1 Desain Interior

Desain interior atau penataan ruang pada prinsipnya merupakan upaya memecahkkan masalah kehidupan yang berkaitan dengan ruang bagian dari sebuah bangunan. Desain interior berkaitan dengan masalah fisik dan non fisik. Masalah fisik berkaitan dengan kondisi ruang yang terdiri atas unsur lantai, dinding, plafon, utilitas seperti jendela untuk memasukan cahaya alam, ventilasi untuk mengalirkan udara alami, pintu untuk mengakses hubungan antar ruang, mekanika dan elektikal seperti saluran pelistrikan dan masalah non fisik berkaitan dengan faktor manusia seperti kondisi psikologis, sosial dan budaya yang membentuk persepsi-persepsi dan perasaan terhadap suasana ruang tertentu [3]. Setiap desain bertujuan untuk menyusun secara teratur bagian demi bagiannya menjadi satu tatanan yang utuh demi maksud-maksud tertentu. Dalam desain interior, elemen-elemen yang dipilih ditata menjadi pola tiga dimensi sesuai dengan garis-garis besar fungsi, estetika dan perilakunya. Hubungan antara elemen-elemen yang terbentuk dari pola-pola ini pada akhirnya menentukan kualitas visual dan kecocokan fungsi ruang interior dan mempengaruhi bagaimana kita memahami dan menggunakannya. Oleh karena itu, maksud dan tujuan desain interior adalah untuk memperbaiki fungsi memperkaya nilai estetika dan meningkatkan aspek psikologis dan ruang interior.



Gambar 2.2 Perancangan Ruangan

Desain interior harus mampu melampui definisi ruang dalam arsitektur. Dalam perancangan tata letak, perlengkapan ruang dan ornamen ruang, perancangan interior harus benar-benar menyadari karakter arsitekturnya termasuk potensi-potensi untuk modifikasi dan peningkatan kualitasnya [4]. Oleh karena itu, desain ruang interior menuntut pemahaman bagaimana ruang tersebut terbentuk oleh sistem bangunannya seperti struktur dan penutup ruangnya. Dengan pemahaman tersebut perancangan interor dapat secara efektif memilih untuk mengerjakan, meneruskan, atau bahkan menawarkan usulan baru kepada konsep utama sebuah ruang arsitektur.

Proses desain menetapkan elemen-elemen yang akan digunakan dan bagaimana menatanya menjadi pola-pola melalui proses desain. Meskipun disajikan sebagai urut-urutan langkah yang linier, proses desain lebih sering berbentuk silus dimana setiap tahapan merupakan serangkaian analisis, sintesis dan evaluasi yang teliti mengenai informasi yang tersedia persepsi serta memungkinan solusi yang diulang terus-menerus sampai diperoleh perpaduan yang sesuai antara apa yang sudah ada dengan yang diinginkan.

Masalah desain harus didefinisikan lebih daulu. Definisi tersebut harus mencakup spesifikasi mengenai bagaimana melaksanakan solusi desain itu. Tujuan akhir dan tujuan-tujuan lain harus ditetapkan terlebih dahulu. Analisis suatu permasalahan adalah memecah permasalahan tersebut menjadi beberapa

bagian, memperjelas permasalahannya dan mencoba menerapkan nilai-nilai dari berbagai aspek pada pemasalahan itu. Analisis juga melibatkan pengumpulan informasi-informasi yang relevan yang dapat membantu kita memahami sidat dari permasalahan tersebut dan membangun tanggapan yang sesuai. Dari titik tolak ini, perlu diketahui juga batasan apa saja yang akan membantu membentuk solusi desain. Faktor-faktor yang diperoleh apa yang dapat dan apa yang tidak dapat diubah harus dipertimbangkan. Keterbatasan-keterbatasan finansial, hukum atau teknis yang dapat mengganggu solusi desain yang harus diperhatikan. Sementara melanjutkan proses desain, semakin tampak pengertian yang lebih jelas tentang permasalahannya. Informasi baru mungkin ditemukan atau dibutuhkan yang mana dapat merubah persepsi kita terhadap masalah maupun solusinya. Oleh karena itu, analisis terus menerus dilakukan dalam proses desain.

2.1.2 *Furniture*

Mebel atau *furniture* adalah perlengkapan rumah yang mencakup semua barang seperti kursi, meja, dan lemari. Mebel berasal dari kata *movable,* yang artinya bisa bergerak. Pada zaman dahulu meja kursi dan lemari relatif mudah digerakkan dari batu besar, tembok, dan atap. Sedangkan kata *furniture* berasal dari bahasa Prancis *fourniture* (1520-30 Masehi). *Fourniture* mempunyai asal kata *fournir* yang artinya *furnish* atau perabot rumah atau ruangan. Walaupun mebel dan *furniture* punya arti yang beda, tetapi yang ditunjuk sama yaitu meja, kursi, lemari, dan seterusnya. Dalam kata lain, mebel atau *furniture* adalah semua benda yang ada di rumah dan digunakan oleh penghuninya untuk duduk, berbaring, ataupun menyimpan benda kecil seperti pakaian atau cangkir. Mebel terbuat dari kayu, papan, kulit, sekrup, dll

Dalam buku berjudul desain interior Sufty Nurahmartiyanti (2008:3) Furniture merupakan salah satu elemen untuk menyempurnakan sebuah desain interior. Keberadaan furniture juga berperan penting dalam mewujudkan citra milimalis pada sebuah hunian. *Furniture* minimalis berbentuk sederhana dan

minimalis dalam penggunaan profil atau detail. Selain itu furniture minimalis cenderung tanpa ukiran sehingga tampilannya terlihat lebih bersih dan simple..

2.2 Kecerdasan Buatan (*Artificial Intelligence*)

Bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia, bahkan bisa lebih baik daripada yang dilakukan manusia, untuk mengetahui dan memodelkan proses–proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia.

2.2.1 Beda Kecerdasan Buatan & Kecerdasan Alami

Kelebihan kecerdasan buatan :

1. Lebih bersifat permanen. Kecerdasan alami bisa berubah karena sifat manusia pelupa. Kecerdasan buatan tidak berubah selama sistem komputer & program tidak mengubahnya.
2. Lebih mudah diduplikasi & disebarkan. Mentransfer pengetahuan manusia dari 1 orang ke orang lain membutuhkan proses yang sangat lama & keahlian tidak akan pernah dapat diduplikasi dengan lengkap.Jadi jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut & dapat dipindahkan dengan mudah ke komputer yang lain.
3. Lebih murah. Menyediakan layanan komputer akan lebih mudah & murah dibandingkan mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Bersifat konsisten dan teliti karena kecerdasan buatan adalah bagian dari teknologi komputer sedangkan kecerdasan alami senantiasa berubah-ubah.
5. Keputusan yang dibuat komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
6. Dapat mengerjakan beberapa *task* lebih cepat dan lebih baik dibanding manusia

2.2.2 Kelebihan kecerdasan alami :

1. Kreatif manusia memiliki kemampuan untuk menambah pengetahuan, sedangkan pada kecerdasan buatan untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun
2. Memungkinkan orang untuk menggunakan pengalaman atau pembelajaran secara langsung. Sedangkan pada kecerdasan buatan harus mendapat masukan berupa input-input simbolik.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

2.3 Kecerdasan Buatan (*Artificial Intelligence*)

Algoritma merupakan salah satu cabang ilmu komputer yang membahas prosedur penyelesaian suatu permasalahan. Algoritma adalah urutan langkah-langkah dalam menentukan suatu masalah. Algoritma juga dapat didefinisikan dengan deretan langkah komputasi yang mentransformasikan masukan (*input*) menjadi keluaran (*output*). Jadi algoritma adalah serangkaian urutan langkah-langkah atau prosedur untuk menyelesaikan suatu masalah dengan memproses nilai masukan menjadi nilai keluaran.

Algoritma dapat ditulis dalam bentuk *pseudocode* ataupun *flowchart*. *Pseudocode* berasal dari kata *pseudo* yang artinya mirip atau menyerupai dan *code* berarti kode sehingga *pseudocode* adalah kode yang menyerupai kode program sebenarnya. Sedangkan *flowchart* merupakan bentuk gambar atau diagram yang mempunyai aliran satu atau dua arah secara sekuensial.

Dalam mempelajari algoritma, diperlukan untuk dapat menganalis algoritma dalam menentukan performansinya. Suatu Algoritma tidak saja harus benar, tetapi juga harus efisien dengan menghitung nilai dari performansinya. Setiap algortima memiliki nilai performansi yang berbeda-beda [6].

2.4. Sifat-Sifat Algoritma

Donald E. Knuth menyatakan bahwa ada beberapa ciri-ciri algoritma, yaitu:

a. Algoritma mempunyai awal dan akhir. Suatu algoritma harus berhenti setelah mengerjakan serangkaian tugas atau dengan kata lain suatu algoritma harus memiliki langkah yang terbatas.
b. Setiap langkah harus didefinisikan dengan tepat sehingga tidak memiliki arti ganda.
c. Memiliki masukan atau kondisi awal.
d. Memiliki keluaran atau kondisi akhir.
e. Algoritma harus efektif; Jika diikuti dengan benar akan menyelesaikan permasalahan.

Berdasarkan cici-cirinya, algoritma dapat disimpulkan sifat utamanya yaitu:

a. *Input* : Suatu algoritma memiliki input atau kondisi awal sebelum algoritma dilaksanakan dan bisa berupa nilai-nilai peubah yang diambil dari himpunan khusus.
b. *Output* : Suatu algoritma akan menghasilkan *output* setelah dilaksanakan, atau
   algoritma akan mengubah kondisi awal menjadi kondisi akhir, dimana nilai *output* diperoleh dari nilai *input* yang telah diproses.
c. *Definiteness* : Langkah-langkah yang dituliskan dalam algoritma terdefinisi dengan jelas sehingga mudah dilaksanakan oleh pengguna algoritma.
d. *Finiteness* : Suatu algoritma harus memberi kondisi akhir atau *output* setelah melakukan sejumlah langkah yang terbatas jumlahnya untuk setiap kondisi awal atau *input* yang diberikan.
e. *Effectiveness* : Setiap langkah dalam algoritma dapat dilaksanakan dalam selang waktu tertentu sehingga pada akhirnya member solusi sesuai yang diharapkan.
f. *Generality* : langkah-langkah algoritma berlaku untuk setiap himpunan input yang sesuai dengan persoalan yang akan diberikan, tidak hanya untuk himpunan tertentu.

Algoritma sebagai langkah-langkah pemecahan masalah dapat dituliskan dengan berbagai cara, seperti:

1) Uraian deskriptif

   Penulisan algoritma dengan uraian deskriptif menggunakan bahasa yang biasa

   Digunakan sehari-hari.

2) *Pseudocode*

   Algoritma yang dituliskan dalam kode-kode yang disepakati dan mempunyai urutan-urutan tertentu. Kode-kode ini dapat dikembangkan sendiri asalkan arti dari setiap kode disepakati bersama.

3) Bagan alir (flowchart)

   Flowchart (bagan alir dokumen) adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program.

2.5. Algoritma *Fuzzy Logic*

Algoritma *fuzzy logic* adalah algoritma yang mengandung unsur ketidakpastian, tidak seperti logika biasa yaitu logika tegas, kita hanya mengenal dua nilai, salah atau benar, atau 0 atau 1. Himpunan *fuzzy* adalah pengelompokan sesuatu bedasarkan parameter dan variable bahasa atau *linguistik variable*, yang dinyatakan dalam fungsi keanggotaan (*Universe of discourse)* U, fungsi keanggotaan dari suatu himpunan *fuzzy* tersebut bernilai antara 0.0 sampai dengan 1.0

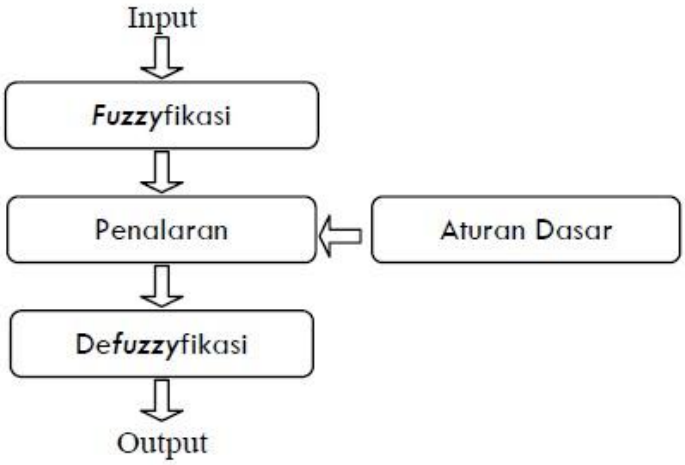
Himpunan *fuzzy* didasarkan pada gagasan untuk memperluas jangkauan fungsi karateristik sedemikian rupa sehingga fungsi tersebut akan mencangkup bilangan real pada interval [0,1]. Nilai keanggotaannya menunjukan bahwa suatu item tidak hanya bernilai benar atau salah. Nilai 0 menujukan salah, nilai 1 menujukan benar dan masih ada nilai nilai yang terletak antara benar dan salah.

2.6 Kendali *Fuzzy Logic*

Sistem kendali logika *fuzzy* disebut juga sistem Inferensi *Fuzzy* (*Fuzzy Inference System*/*FIS*) atau *fuzzy inference engine* adalah sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya. Terdapat beberapa jenis FIS yang dikenal yaitu

Mamdani, Sugeno dan Tsukamoto. FIS yang paling mudah dimengerti, karena paling sesuai dengan naluri manusia adalah FIS Mamdani.

Sistem kendali *fuzzy logic* terdiri dari beberapa tahapan seperti berikut :



Gambar 2.3 Proses Kendali *Fuzzy Logic*

Proses dalam kendali *fuzzy logic* ditunjukan pada gamabr di atas. *Input* yang diberikan adalah berupa bilangan tertentu dan *output* yang dihasilkan juga harus berupa bilangan tertentu.

*Fuzzyf*ikasi adalah pemetaan nilai input yang merupakan nilai tegas ke dalam fungsi keanggotaan himpunan *fuzzy*, untuk kemudian diolah di dalam mesin penalaran

*Fuzzy*fikasi : $x \rightarrow \mu(x)$ (6)

Aturan dasar dalam logika *fuzzy* adalah aturan implikasi dalam bentuk"jika... maka..." . Aturan dasar tersebut ditentukan dengan bantuan seorang pakar yang mengetahui karateristik objek yang akan dikendalikan.

Contoh bentuk implikasi yang digunakan adalah sebagai berikut.

Jika X = A dan Y = B maka Z = C. (7)

Penalaran adalah menalarkan nilai masukan untuk menentukan nilai keluaran sebagai bentuk pengambil keputusan. Sistem terdiri dari beberapa aturan, maka kesimpulan diperoleh dari kumpulan dan korelasi antar aturan.

Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu *max, additive* dan probabilistik *OR*. Secara umum dapat ditulis

$$\mu_{df}(x_i) \leftarrow \max(\mu_{df}(x_i,)\, \mu_{kf}(x_i)) \quad (8)$$

De*fuzzy*fikasi merupakan kebalikan dari *fuzzy*kasi, yaitu pemetaan dari himpunan *fuzzy* ke himpunan tegas. *Input* dari proses de*fuzzy*fikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*. Hasil dari de*fuzzy*fikasi ini merupakan output dari sistem kendali logika *fuzzy*.

De*fuzzy*fikasi di deskripsikan sebagai

$$\boldsymbol{z = \frac{\sum aiZi}{ai}} \tag{9}$$

dengan

αi = Nilai minimal dari derajat keanggotaan

Zi = Nilai domain dari variabel linguistik

z= Nilai defuzzyfikasi

2.7 Fungsi Keanggotaan

Basis data fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukan pemetaan titik-titik input data ke dalam nilai keanggotaannya atau 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 digunakan. Fungsi keanggotaan suatu himpunan *fuzzy* dapat ditentukan dengan fungsi segitiga *(triangle)*, trapesium *(trapezoidal)*, *sigmoid* dan linier .

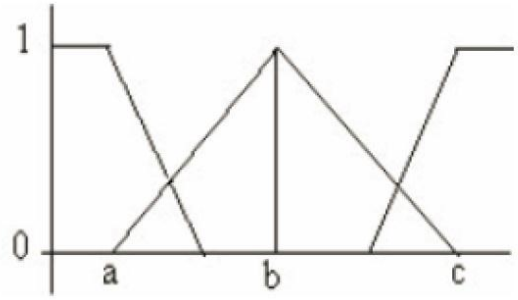
1. Fungsi Segitiga

Fungsi segitiga hanya terdapat satu nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika x = b. Tetapi, nilai-milai disekitar b memiliki derajat keanggotaan yang turun menjauhi nilai 1.

Persamaan fungsi keanggotaan segitiga adalah :

$$\mu(x;a,b,c) = \begin{cases} 0 & x < a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b < x \le c \\ 1 & x > c \end{cases} \tag{1}$$

Persamaan tersebut direpresentasikan dalam bentuk grafik sebagi berikut :



Gambar 2.4 Grafika fungsi keanggotaan segitiga

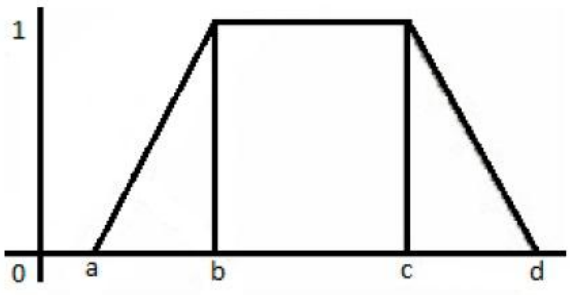
2. Fungsi Trapesium

Fungsi trapesium ini terdapat beberapa nilai x yang memiliki derajat kenaggotaan sama dengan 1, yaitu ketika $b \leq x \leq c$. Tetapi derajat keanggotaan untuk $a < x < b$ dan $c < x < d$ memiliki karateristik yang sama dengan fungsi segitiga.

Persamaan fungsi keanggotaan trapesium adalah :

$$\mu(x;a,b,c,d) = \begin{cases} 0 & x < a \\ \dfrac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x \leq c \\ \dfrac{d-x}{d-c} & c < x \leq d \\ 0 & x > d \end{cases} \qquad (2)$$

Persamaan tersebut direpresentasikan dalam bentuk grafik sebagi berikut :



Gambar 2.5 Grafika fungsi keanggotaan trapesium

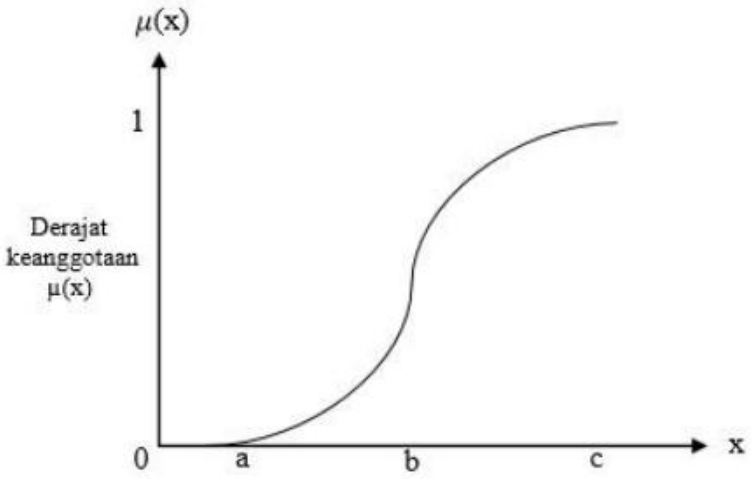
3. Fungsi *Sigmoid*

Fungsi sigmoid ini berbetuk kurva *sigmoidal* seperti huruf S. Setiap nilai x (keanggotaan *crisp set*) dipetakan ke dalam interval [0,1].

Persamaan fungsi keanggotaan *sigmoid* adalah :

$$sigmoid(x,a,b,c) = \begin{cases} 0; x \leq a \\ \frac{2(x-a)}{(c-a)^2}; a \leq x \leq b \\ 1 - 2\left(\frac{c-x}{c-a}\right)^2; b \leq x \leq c \\ 1; x \geq c \end{cases} \quad (3)$$

Persamaan tersebut direpresentasikan dalam bentuk grafik sebagi berikut :



Gambar 2.6 Grafika fungsi keanggotaan sigmoid

4. Fungsi Linier

Pada fungsi linier, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati konsep yang kurang jelas. Ada 2 bentuk untuk himpunan fuzzy linear, yaitu:

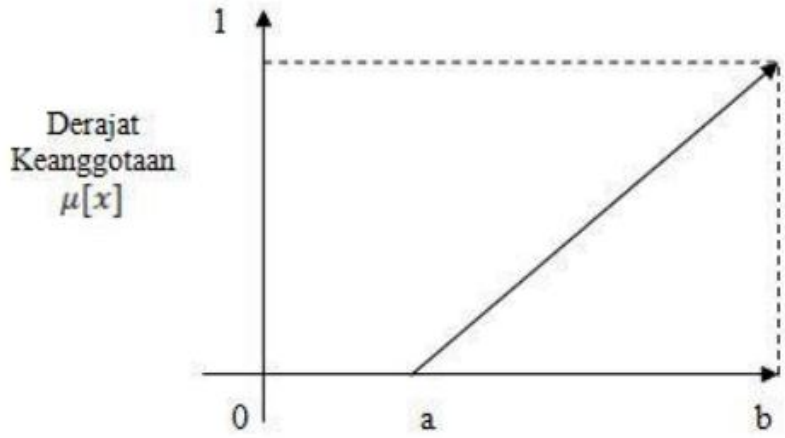
a. Linier Naik

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.

Persamaan fungsi keanggotaan linear naik adalah :

$$\mu[x] = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x \geq b \end{cases} \quad (4)$$

Persamaan tersebut direpresentasikan dalam bentuk grafik sebagi berikut :



Gambar 2.7 Grafika fungsi keanggotaan Linier Naik

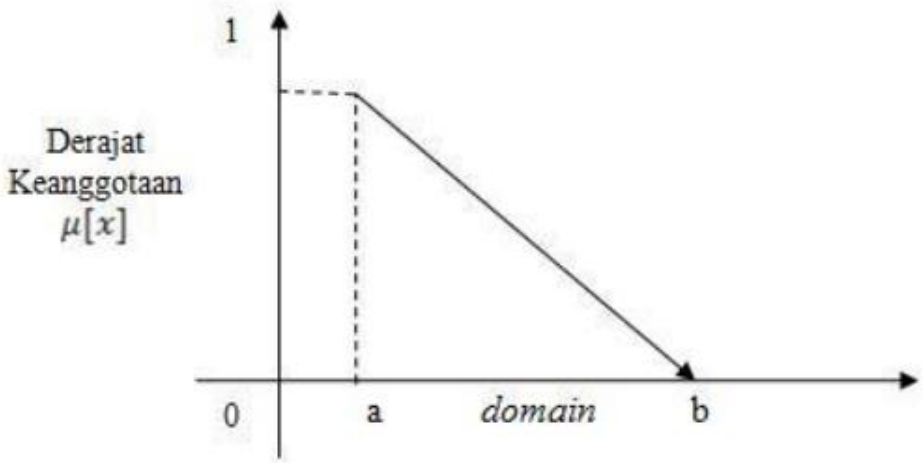
a. Linier Turun

Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat kenaggotaan lebih rendah.

Persamaan fungsi keanggotaan linear naik adalah :

$$\mu[x] = \begin{cases} \frac{b-x}{b-a} if a \leq x \leq b \\ 0 \quad if\ x \geq b \end{cases} . \quad (5)$$

Persamaan tersebut direpresentasikan dalam bentuk grafik sebagi berikut :



Gambar 2.8 Grafika fungsi keanggotaan Linier Turun

2.8 UML

UML adalah kependekan dari *Unified Modeling Language* yang merupakan suatu cara untuk menyelesaikan suatu masalah dengan mendeskripsikannya yang telah menjadi standar dalam dunia industri untuk memvisualisasikan, merancang dan mendokumentasikan sistem perangkat lunak. Dengan menggunakan UML kita dapat membangun model untuk segala bentuk dan jenis aplikasi perangkat lunak, yang mana aplikasi yang dibangun dapat berjalan pada perangkat lunak dengan sistem operasi dan jaringan apapun. Selain itu, dengan menggunakan UML, programmer dapat mengerti, memahami dengan jelas maksud, tujuan dan arah rancangan sistem, sehingga sistem perngakat lunak yang dirancang dapat dibangun dengan bahasa program apapun. Namun karena UML merupakan suatu model pengembangan sistem perangkat lunak yang berbasis *object oriented* sehingga menggunakan bentuk *class* dan *operation* dalam konsep dasarnya, maka lebih cocok dalam pemrogramannya dengan menggunakan bahasa-bahasa pemrograman yang berbasis OOP (*Object Oriented Programming*) seperti C, C++, Java, VB dan lain sebagainya.

Secara lebih mendalam, UML lebih dari sekedar sebuah standard dan penemuan dari suatu notasi-notasi yang disatukan, tetapi juga berisi konsep-

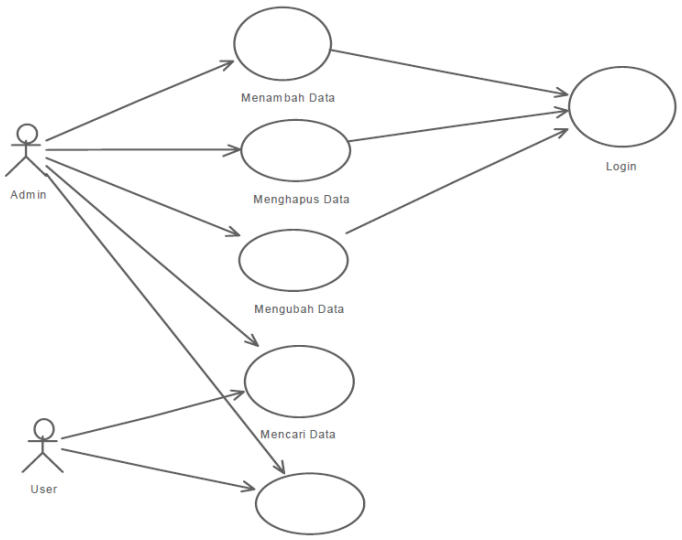
konsep baru yang menarik yang tidak ditemukan secara umum dalam komunitas object oriented. Dalam membangun suatu model perangkat lunak dengan UML, digunakan bentuk-bentuk diagram atau symbol untuk merepresentasikan elemen-elemen dalam sistem. Bentuk diagram yang digunakan untuk merepresentasikannya adalah sebagai berikut:

1. *Use-case Diagram*
2. *Class Diagram*
3. *Sequence diagram*
4. *Collaboration Diagram*
5. *Activity Diagram*
6. *Component Diagram*
7. *Deployment Diagram*

2.8.1 Use Case Diagram

Use-case diagram merupakan suatu bentuk diagram yang menggambarkan fungsi-fungsi yang diharapkan dari sebuah sistem yang dirancang. Dalam *Use-case diagram* penekanannya adalah "apa" yang diperbuat oleh sistem, dan bukan "bagaimana". Sebuah *use-case* akan merepresentasikan sebuah interaksi antara pelaku atau *actor* dengan sistem.

Use-case diagram yang digunakan dalam mercancang suatu sistem dapat sangat membantu pada saat menyusun *requirement* sebuah sistem, mengomunikasikannya dengan klien, dan merancang pengujian untuk semua fitur yang terdapat dalam sistem. Dalam suatu sistem aplikasi database, *use-case diagram* sangat membantu *requierement* apa saja yang diperlukan.



Gambar 2.9 Contoh *Use-case Diagram* [5]

2.8.2 Class Diagram

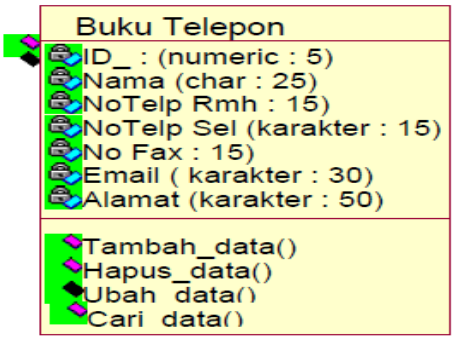
Sebuah *Class Diagram* menunjukkan struktur yang statis dari beberapa *class* dalam suatu sistem. *Class-class* merepresentasikan suatu keadaan (atribut/properti) dan yang akan dikerjakan oleh sistem (metoda/fungsi). *Class* memiliki tiga area pokok:

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dalam *class* diagram dapat memiliki salah satu sifat seperti berikut di bawah ini:

1. *Private*, hanya dapat diakses oleh class itu sendiri.
2. *Protected*, hanya dapat diakses oleh class itu sendiri dan turunan dari class tersebut.
3. *Public*, dapat diakses oleh class selain dari class yang bersangkutan.

Class dapat direpresentasikan dalam sebuah *interface* atau sebaliknya merupakan implementasi dari sebuah *interface* yang berupa *class* abstrak yang hanya tidak memiliki *attribute* dan hanya memiliki metoda. Berikut merupakan bentuk *class* diagram secara umum:

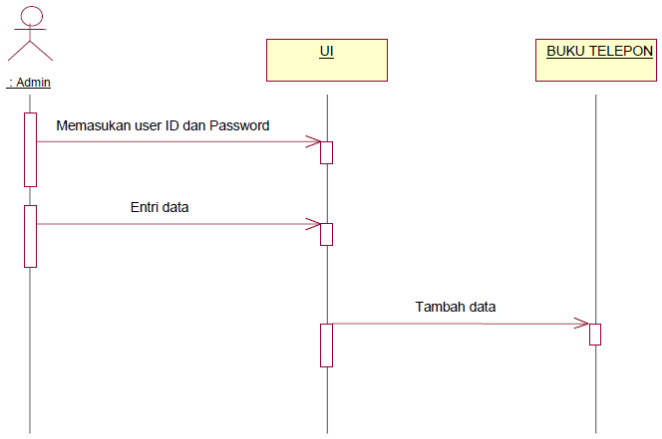


Gambar 2.10 Contoh *Class* Diagram [5]

2.8.3 Sequance Diagram

Sequence Diagram merupakan diagram yang mengambarkan kolaborasi yang dinamis antara obyek satu dengan yang lain. Kolaborasi ini ditunjukkan dengan adanya interaksi antar obyek di dalam dan di sekitar sistem yang berupa pesan atau instruksi yang berurutan.

Sequence diagram umumnya digunakan untuk menggambarkan suatu skenario atau urutan langkah-langkah yang dilakukan baik oleh *actor* maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau *output*.

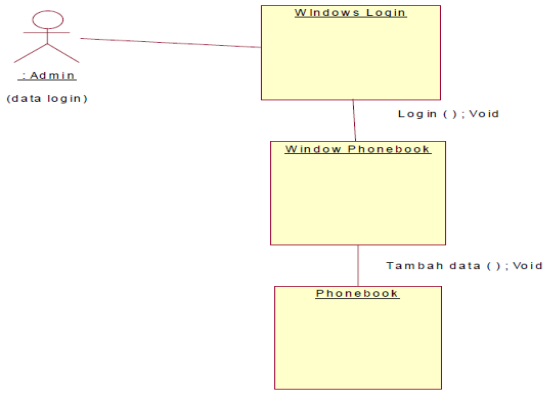


Gambar 2.11 Contoh Sequance Diagram [5]

2.8.4 Collaboration Diagram

Sebuah *collaboration diagram* menunjukkan kolaborasi yang dinamis yang mirip dengan *sequence diagram*. *Collaboration diagram* digambarkan

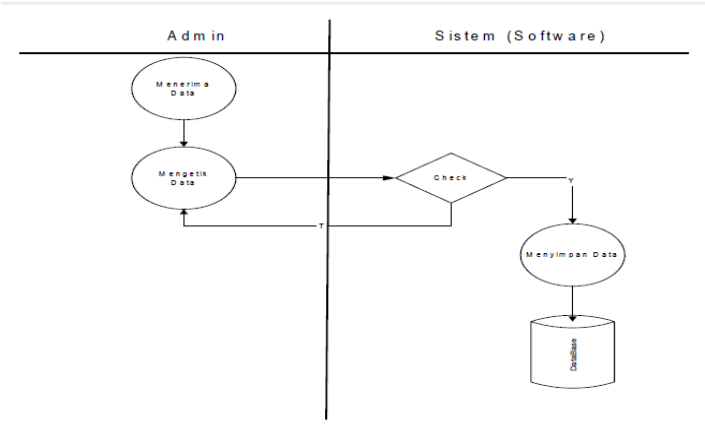
sebagai sebuah *object* diagram dimana sejumlah obyek ditunjukkan disekitarnya dengan hubungan-hubungannya.



Gambar 2.12 Contoh Collaboration Diagram [5]

2.8.5 Activity Diagram

Sebuah *Activity Diagram* menunjukkan suatu alur kegiatan secara berurutan. *Activity Diagram* digunakan untuk mendiskripsikan kegiatan-kegiatan dalam sebuah operasi meskipun juga dapat digunakan untuk mendeskripsikan alur kegiatan yang lainnya seperti *use case* atau suatu interaksi.

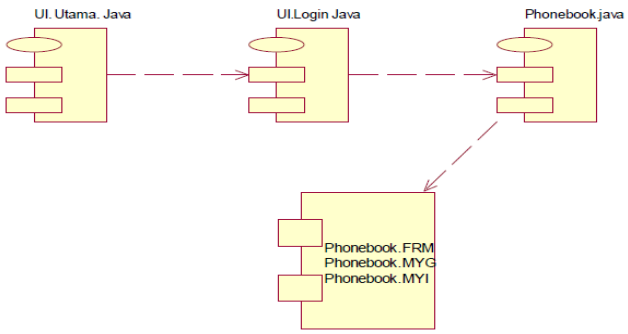


Gambar 2.13 Contoh Activity Diagram [5]

2.8.6 Component Diagram

Component Diagram menunjukkan struktur dan hubungan antar komponen *software* termasuk ketergantungan (*dependency*) diantara komponen-komponen tersebut. Komponen pada piranti lunak adalah berupa modul-modul

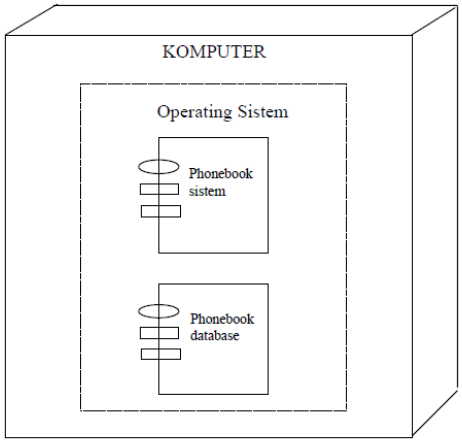
yang berisikan *code*, baik *library* maupun *executable*. Umumnya komponen yang terbentuk dari beberapa *class* dan/atau *package*, atau juga dapat dari komponen-komponen yang lebih kecil.



Gambar 2.14 Component Diagram [5]

2.8.7 Deployment Diagram

Deployment Diagram menunjukkan arsitektur fisik pada *hardware* dan *software* pada suatu sistem yang dirancang. *Deployment diagram* juga dapat menunjukkan perngkat-perangkat dan *nodes* diantara hubungan yang dimilikinya antar komponen.



Gambar 2.15 Contoh Deployment Diagram [5]