

BAB II

LANDASAN TEORI

2.1. Persyaratan dan Pengelolaan Habitat Makro Burung Walet

Berdasarkan penelitian yang dilakukan oleh Marzuki, dkk (1999) burung walet dapat dibudidayakan diseluruh Indonesia. Daerah penyebaran burung ini mulai dari Sumatera sampai Irian Jaya. Meskipun daerah sebarannya meliputi hampir seluruh wilayah Indonesia, tetapi Burung Walet lebih tersebar merata di Pulau Sumatera, Jawa, Kalimantan, Bali, dan NTB dibandingkan dengan pulau lainnya. Oleh karena itu, pemilihan dan penentuan lokasi pengembangan yang cocok merupakan suatu bagian penting untuk keberhasilan budidaya burung walet.

Meskipun daerah penyebarannya cukup luas, tetapi ada daerah yang ekstrim sangat optimal dan ada pula daerah yang ekstrim yang sangat tidak cocok. Didaerah yang tidak cocok bukan berarti tidak ada walet, tetapi burung ini tidak bisa berkembangbiak dengan baik. Kecocokan suatu lokasi untuk rumah walet biasanya dikaitkan dengan perilaku dan kebiasaan hidupnya. Berikut ini beberapa daerah yang cocok untuk dikembangkan sebagai lokasi pengembangan budidaya burung walet, yaitu :

1. Daerah basah dengan musim hujan lebih dari 6 bulan
2. Daerah berdekatan dengan lokasi, seperti: pegunungan, pantai, danau, dan sungai.

Dalam hal ini nantinya akan berguna bagi Pemerintah daerah sendiri untuk mengetahui kondisi daerahnya dan bagi kalangan investor berguna untuk

mengetahui kondisi lahan yang mereka butuhkan dalam menginvestasikan modalnya untuk membangun usahanya dalam membudidayakan burung walet.

2.2. Sistem Informasi Geografi

2.2.1. Konsep dasar

Menurut Prahasta (2001) Sejak pertengahan 1970-an telah dikembangkan sistem-sistem yang secara khusus dibuat untuk menangani masalah informasi yang berferensi geografis dalam berbagai cara dan bentuk. Masalah-masalah ini mencakup :

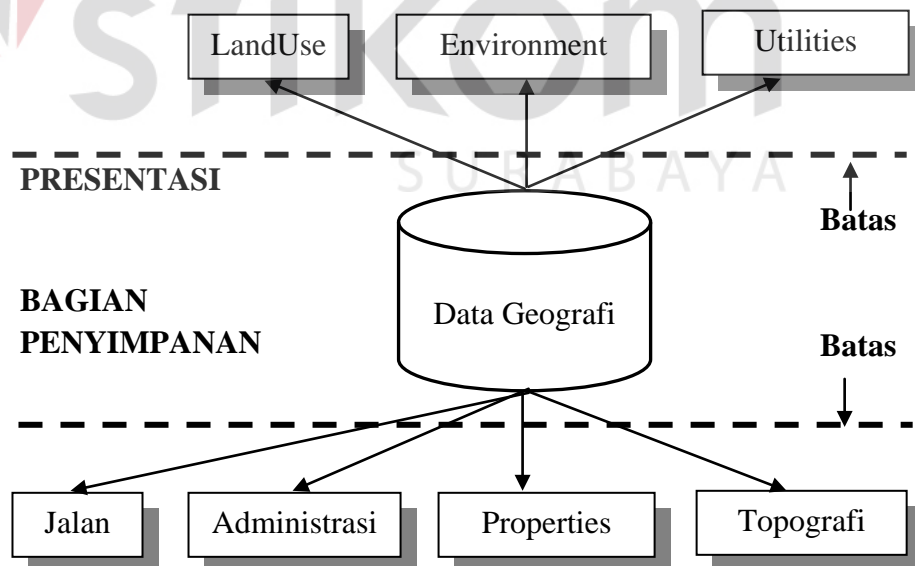
1. Pengorganisasian data dan informasi.
2. Menempatkan informasi pada lokasi tertentu.
3. Melakukan komputasi, memberikan ilustrasi keterhubungan satu sama lainnya (koneksi), beserta analisa-analisa spasial lainnya.

Sebutan umum untuk sistem-sistem yang menangani masalah-masalah atas adalah GIS, sistem informasi geografis. Dalam beberapa literatur, SIG dipandang sebagai hasil dari perkawinan antara sistem komputer untuk bidang kartografi (CAC) atau sistem komputer untuk bidang perancangan (CAD) dengan teknologi basisdata (*database*).

Pada asalnya, data geografi hanya disajikan di atas peta dengan menggunakan simbol, garis, dan warna. Elemen-elemen geometri ini dideskripsikan didalam legendanya misalnya; garis hitam tebal untuk jalan utama, garis hitam tipis untuk jalan sekunder dan jalan-jalan yang berikutnya. Selain itu, berbagai data juga dapat di-*overlay*-kan berdasarkan sistem koordinat yang sama. Akibatnya, sebuah peta menjadi media yang efektif baik sebagai alat presentasi maupun sebagai bank tempat penyimpanan data geografis. Tetapi, media peta

masih mengandung kelemahan atau keterbatasan. Informasi-informasi yang tersimpan, diproses dan dipresentasikan dengan cara tertentu, biasanya untuk tujuan tertentu pula. Tidak mudah untuk merubah bentuk presentasi ini. Sebuah peta selalu menyediakan gambar atau simbol unsur geografi dengan bentuk yang tetap atau statik meskipun diperlukan untuk berbagai kebutuhan yang berbeda.

Peta juga merupakan aset publik yang sangat berharga. Survey-survey pemetaan yang telah dilakukan di berbagai negara telah mengindikasikan bahwa jumlah keuntungan, dari penggunaan peta, akan meningkat hingga beberapa kali lipat biaya produksi peta itu sendiri. Bila dibandingkan dengan peta-peta ini, SIG memiliki keunggulan inheren karena penyimpanan data dan presentasinya dipisahkan. Dengan demikian, data dapat dipresentasikan dalam berbagai cara dan bentuk. Berikut ini merupakan salah satu contoh ilustrasi pemisahan penyimpanan data dan presentasi didalam SIG.



Gambar 2.1 Ilustrasi pemisahan penyimpanan data dan presentasi di dalam SIG

2.2.2. Subsistem SIG

SIG dapat dibagi menjadi beberapa subsistem, sebagai berikut :

1. Data Input

Susbsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini pula yang bertanggungjawab dalam mengkonversi atau mentransformasikan format-format data-data aslinya kedalam format yang dapat digunakan oleh SIG.

2. Data Output

Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basisdata baik dalam bentuk *softcopy* maupun bentuk *hardcopy* seperti: tabel, grafik, peta, dan lain-lain.

3. Data Management

Susbsistem ini mengorganisasikan baik data spasial maupun atribut ke dalam sebuah basisdata sedemikian rupa sehingga mudah dipanggil, di-*update*, dan di-*edit*.

4. Data Manipulation & Analysis

Susbsistem ini menentukan informasi yang dapat dihasilkan oleh SIG. Selain itu , susbsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

2.2.3. Fungsi analisis spasial

Kemampuan SIG dapat juga dikenali dari fungsi-fungsi analisis yang dapat dilakukannya, adapun salah satu fungsi tersebut adalah fungsi analisis spasial yang terdiri atas :

1. Klasifikasi (*reclassify*)

Fungsi ini mengklasifikasikan atau mengklasifikasikan kembali suatu data spasial (atau atribut) menjadi data spasial yang baru dengan menggunakan kriteria tertentu. Misalnya, dengan menggunakan data spasial ketinggian permukaan bumi (topografi), dapat diturunkan data spasial kemiringan atau gradien permukaan bumi yang dinyatakan dalam persentase nilai-nilai kemiringan.

2. *Network* (jaringan)

Fungsi ini merujuk data spasial titik-titik (*point*) atau garis-garis (*lines*) sebagai suatu jaringan yang tidak terpisahkan. Fungsi ini sering digunakan dalam bidang-bidang transportasi dan *utility* (misalnya aplikasi jaringan kabel listrik, telepon, pipa minyak dan gas, air minum, dll).

3. *Overlay*

Fungsi ini menghasilkan data spasial baru dari minimal dua data spasial yang menjadi masukannya. Sebagai contoh, bila untuk menghasilkan wilayah-wilayah yang sesuai untuk budidaya tanaman tertentu (misalnya padi) diperlukan data ketinggian permukaan bumi, kadar air tanah, dan jenis tanah, maka fungsi analisis spasial *overlay* dikenakan terhadap ketiga data spasial tersebut.

4. *Buffering*

Fungsi ini menghasilkan data spasial baru yang berbentuk poligon atau zone dengan jarak tertentu dari data spasial yang menjadi masukannya.

5. 3D analysis

Fungsi ini terdiri dari sub-sub fungsi yang berhubungan dengan presentasi data spasial dalam ruang 3 dimensi. Fungsi analisis ini banyak menggunakan fungsi interpolasi.

6. *Digital image processing* (pengolahan citra digital)

Fungsi ini dimiliki oleh perangkat SIG yang berbasiskan raster. Karena data spasial permukaan bumi (citra digital). Banyak didapat dari perekaman data satelit yang berformat raster, maka banyak SIG raster yang juga dilengkapi dengan fungsi analisis ini.

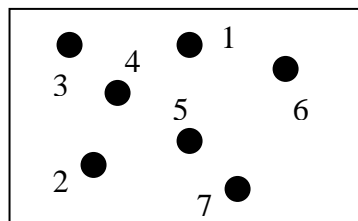
Dan masih banyak fungsi-fungsi analisis spasial lainnya yang umum dan secara rutin digunakan didalam SIG.

2.2.4. Representasi grafis suatu objek

Informasi grafis suatu objek dapat dimasukkan dalam bentuk :

1. Titik (tanpa dimensi)

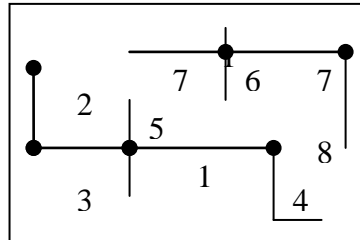
Titik adalah representasi grafis yang paling sederhana untuk suatu objek. Representasi ini tidak memiliki dimensi tetapi dapat diidentifikasi diatas peta dan dapat ditampilkan pada layar monitor dengan menggunakan simbol-simbol.



Gambar 2.2 Contoh representasi objek titik untuk data posisi sumur bor

2. Garis (satu dimensi)

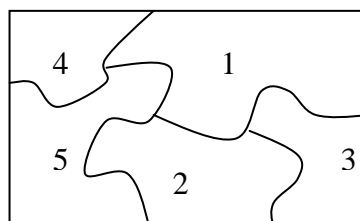
Garis adalah bentuk liner yang akan menghubungkan paling sedikit dua titik dan digunakan untuk merepresentasikan objek-objek satu dimensi.



Gambar 2.3 Contoh representasi objek garis untuk data lokasi jalan-jalan

3. Poligon (dua dimensi)

Poligon digunakan untuk merepresentasikan objek-objek dua dimensi. Suatu danau, batas propinsi, batas kota, batas-batas persil tanah milik adalah tipe-tipe *entity* yang pada umumnya direpresentasikan sebagai poligon. Tetapi representasi ini masih bergantung pada skala tampilan peta (titik atau poligon). Suatu poligon paling sedikit dibatasi oleh tiga garis yang saling terhubung di antara ketiga titik tersebut. Didalam basisdata, semua bentuk area dua dimensi akan direpresentasikan oleh bentuk poligon.



Gambar 2.4 Contoh representasi objek area/poligon untuk data landuse

2.3. Fuzzy Logic / Logika Fuzzy

2.3.1. Sejarah fuzzy logic

Fuzzy logic secara resmi diperkenalkan pada tahun 1965, oleh Lotfi melalui jurnalnya yang berjudul “*Fuzzy Set*” dalam jurnal *Information and Control*. Dan dalam paper penting lain yang ditulisnya yaitu “*Outline of a New Approach to the Analysis of Complex Systems and Decision Process*”, dipublikasikan pada tahun 1973 dalam jurnal *IEEE Transactions on System, Man and Cybernetics*, menjelaskan mengenai pemikiran fuzzy (*fuzzy reasoning*).

Namun Lotfi Zadeh bukanlah orang yang menciptakan *fuzzy logic* / logika fuzzy, karena logika fuzzy telah ada sejak dulu kala. Bisa kita lihat dari teori dari Arisotle dan George Boole, dimana menjelaskan representasi kebenaran dalam dua nilai benar / salah, [0,1]. Dan jauh sebelum itu, ahli filsafat Yunani kuno, yaitu Heraclitus dan Anaximander telah mengenal sistem logika ini. (Earl Cox, *Fuzzy Logic for Business and Industry*, 1995).

Dan sejak permulaan abad ke-20, Max Black telah menulis mengenai *fuzzy set* dalam penelitiannya mengenai ketidakjelasan (*vagueness*). Kemudian pertengahan tahun 1930, Jan Lukasiewicz, dimana dikenal sebagai penemu dari notasi tingkah laku (*polish notation*), mengembangkan sistem logika yang memperluas nilai kebenaran untuk semua bilangan riil dari 0 sampai 1. Ia menggunakan himpunan bilangan tersebut mempresentasikan kemungkinan dari pernyataan yang telah diberikan benar atau salah.

Perkembangan teori logika fuzzy telah menarik pakar sistem kendali untuk memanfaatkannya dalam pengendalian suatu sistem dalam bentuk algoritma

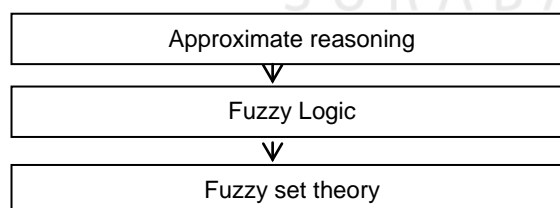
- algoritma otomatis yang dapat dinyatakan, seperti dalam pemakaian pengaturan lalu lintas, sistem transmisi otomatis, alat rumah tangga, industri dan lain-lainnya.

Aplikasi logika fuzzy pada industri pertama kali diterapkan setelah tahun 1970 di Eropa. Di Queen Mary College, London, Inggris, Ebrahim Mamdani menggunakan logika fuzzy untuk mengontrol generator uap, dengan menggunakan teknik konvensional.

RWTH University of Aachen, Germany, Hans Zimmermann juga menggunakan logika fuzzy dalam sistem pengambil keputusan (DSS). Dan sampai saat ini logika fuzzy masih tetap masih digunakan.

2.3.2. Definisi fuzzy logic

Suatu logika dimana mencoba menggabungkan suatu pertidaksamaan dengan bahasa dengan kejadian alam dengan perhitungan kekuatan komputer untuk menghasilkan kecerdasan tinggi, kuat dan sistem pemikiran yang fleksibel. Namun logika fuzzy, dalam kehidupan sehari-hari, cakupannya benar-benar luas. Lapisan perbedaannya dapat dilihat pada gambar berikut:



Gambar 2.5 Berbagai aspek dari fuzzy logic

Digambarkan yang paling dibawah adalah *Fuzzy Set Theory*, yang menggambarkan ilmu mekanik dari bagaimana *fuzzy sets* mengatur dan operasi apa yang diijinkan. Logika Fuzzy itu sendiri adalah proses pembuatan kesimpulan

logis dari kumpulan-kumpulan *fuzzy set*. Dalam banyak aplikasi control dan mesin hal ini digambarkan paling atas, karena tak ada teknologi lain yang dibutuhkan.

Dan yang paling atas adalah *approximate reasoning*, suatu kombinasi dari logika matematika dan *heuristic* yang sangat kuat. *Approximate reasoning* merupakan alat yang digunakan oleh *fuzzy expert* dan sistem pendukung keputusan dan termasuk didalamnya penentuan batas (*hedges*) fuzzy set, aturan-aturan (*rules*), dan bentuk operator.

2.3.3. Kemampuan *fuzzy logic*

- Beroperasi tanpa campur tangan manusia secara langsung, tetapi sama efektifitasnya dengan controller manusia
- Mampu menangani sistem-sistem yang kompleks, non linier dan tidak stasioner
- Strukturnya sederhana dan beroperasi secara *real time*
- Dapat melengkapi kekurangan dari model *boolean* dalam mempresentasikan hal-hal pada dunia nyata
- Mampu memenuhi kebutuhan untuk memecahkan masalah yg kompleks dan presisi
- Kecepatan dalam pengembangan dan kemudahan dalam implementasi

2.3.4. Konsep utama fuzzy logic

A. Prinsip ketidakpastian

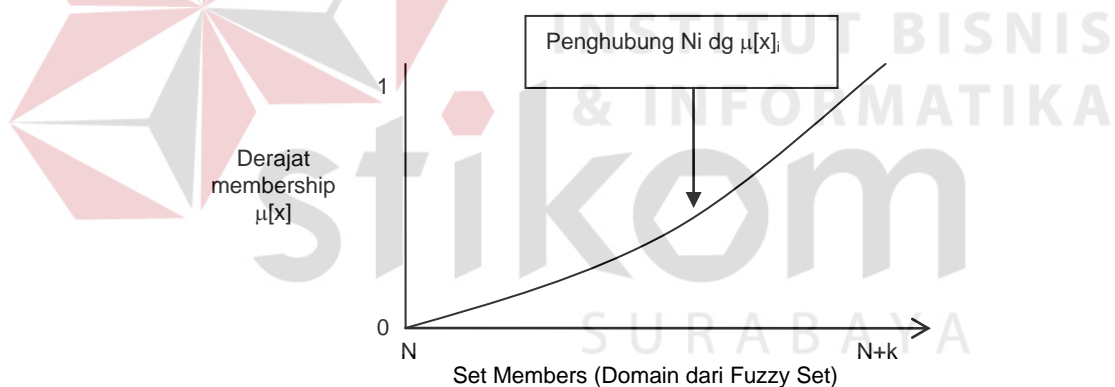
Beberapa ilmu matematika terkadang sulit untuk dipastikan, seperti teori probabilitas, teori informasi, teori *fuzzy set*. Hal ini bisa diklasifikasikan berdasar

tipe ketidakpastian yang dilakukan. Ada beberapa tipe ketidakpastian, dua diantaranya adalah *Stochastic Uncertainty* dan *Lexical Uncertainty*.

Stochastic Uncertainty berhubungan dengan arah ketidakpastian dari kejadian yang pasti. Sedangkan *Lexical Uncertainty* merupakan ketidakpastian yang diungkapkan oleh kata-kata manusia, seperti “orang tinggi”, “hari yang panas”.

B. Fuzzy sets

Fuzzy sets terdiri atas 3 bagian, dimana sumbu horisontal menunjukkan kumpulan member, sumbu vertikal menunjukkan derajat dari membership, dan garis yang menghubungkan masing-masing titik dari member dengan derajat membership yang tepat.



Gambar 2.6 Fuzzy sets

C. Membership Function (MBF)

Derajat dimana angka teknis bernilai sesuai konsep bahasa dari kondisi variabel bahasa (*linguistic*) dinamakan sebagai derajat membership. Untuk variabel berlanjut (*continous variable*) derajat ini disebut *Membership Function* (MBF).

D. Variabel linguistik

Logika fuzzy pada dasarnya menitikberatkan pada pengukuran dan penalaran tentang kekaburan atau bentuk fuzzy yang nampak dalam bahasa alami. Dalam logika fuzzy bentuk fuzzy dinyatakan sebagai variabel linguistik (disebut juga variabel fuzzy).

Variabel linguistik adalah bentuk yang digunakan dalam bahasa alami untuk menggambarkan beberapa konsep yang biasanya mempunyai kekaburan atau nilai fuzzy. Sebagai contoh dalam pernyataan “Jack adalah muda” menyatakan bahwa variabel linguistik umur mempunyai nilai linguistik muda.

Range dari nilai kemungkinan sebuah variabel linguistik disebut semesta pembicaraan dari variabel. Sebagai contoh diberikan range variabel suhu yang digunakan pada rule 1 antara 0-15 derajat. Kata “suhu rendah” anggota dari semesta pembicaraan dari variabel. Ini merupakan himpunan fuzzy.

E. Aturan fuzzy

Aturan dari sistem logika fuzzy (*fuzzy logic system*) menggambarkan pengetahuan dari sistem. Mereka menggunakan variabel linguistik sebagai bahasanya, sebagai contoh untuk mengekspresikan strategi control dari sebuah pengontrol logika fuzzy. Menjelaskan aturan fuzzy berarti menunjukkan, bagaimana menghitung dengan konsep linguistik.

2.3.5. Perhitungan fuzzy

Dalam aplikasi kontrol, komputasi *fuzzy* terdiri atas 3 bagian, yaitu :

A. Fuzzifikasi

Fuzzifikasi berarti menggunakan *membership function* dari variabel Linguistik untuk menghitung masing-masing derajat kondisi validitas dari angka-angka spesifik proses. Fuzzifikasi yang mentransformasi masukan himpunan klasik (*crisp*) ke derajat tertentu yang sesuai dengan aturan besaran fungsi keanggotaan (*membership function*).

B. Inference

Sistem Inferensi Fuzzy adalah sistem kerja komputer yang didasarkan pada konsep teori *fuzzy*, aturan *if-then*, dan logika fuzzy. Struktur dasar dari sistem Inferensi Fuzzy terdiri dari basis aturan yang berisi aturan *if-then*, basis data yang mendefinisikan fungsi keanggotaan dari himpunan fuzzy. Dua teknik inferensi fuzzy yang terkenal digunakan dalam penerapan adalah *max-min inference* dan *max-product inference*.

❖ Max-Min Inference

Dalam *max-min inference* implikasi operasi yang digunakan adalah min, yaitu : $m_{ij} = \text{truth}(a_i \rightarrow b_j) = \min(a_i, b_j) \dots \dots \dots \text{persamaan 1}$

Apabila diberikan 2 fuzzy set A dan B kita bisa menggunakan persamaan tadi untuk membentuk matrix M. Kemudian kita bisa menggunakan persamaan : $b_j = \max_{1 \leq i \leq n} \{ \min(a'_i, m_{ij}) \} \dots \dots \dots \text{persamaan 2}$

Untuk menentukan penyebab vektor B' dari suatu subset A yang dilambangkan A'.

Sebagai gambaran, diasumsi bahwa kita memiliki suatu parameter x yang mewakili temperatur dan suatu *fuzzy set* A dalam x yang menggambarkan normal temperatur, kemudia kita memiliki suatu parameter y yang mewakili kecepatan

dan suatu *fuzzy set* B didalam y yang menggambarkan kecepatan medium.

Akhirnya kita mempunyai rule fuzzy set sebagai berikut :

IF temperatur adalah normal THEN velocity adalah medium

atau

IF A THEN B

Kemudian asumsikan bahwa fuzzy set diwakili ole vektor berikut, dimana untuk keterangan elemen vektor ditunjukkan dengan hubungan utama sebagai berikut :

Temperatur Normal = (0/100, .5/125 , 1/150, .5/175, 0/200)

Kecepatan Medium = (0/10, .6/20 , 1/30, .6/50, 0/50)

Kita mulai dengan membentuk matrik M menggunakan persamaan 1. :

$$\begin{aligned}
 M &= m_{ij} = \min(a_i, b_j) \\
 &= \begin{vmatrix} \min(0., 0.) & \min(0., .6) & \min(0., 0.) & \min(0., .6) & \min(0., 0.) \\ \min(.5, 0.) & \min(.5, .6) & \min(.5, 0.) & \min(.5, .6) & \min(.5, 0.) \\ \min(1., 0.) & \min(1., .6) & \min(1., 0.) & \min(1., .6) & \min(1., 0.) \\ \min(.5, 0.) & \min(.5, .6) & \min(.5, 0.) & \min(.5, .6) & \min(.5, 0.) \\ \min(0., 0.) & \min(0., .6) & \min(0., 0.) & \min(0., .6) & \min(0., 0.) \end{vmatrix} \\
 &= \begin{vmatrix} 0. & 0. & 0 & 0. & 0. \\ 0. & 0.5 & 0.5 & 0.5 & 0. \\ 0. & 0.6 & 1 & 0.6 & 0. \\ 0. & 0.5 & 0.5 & 0.5 & 0. \\ 0. & 0. & 0. & 0. & 0. \end{vmatrix}
 \end{aligned}$$

kemudian asumsikan subset A' sebagai berikut :

A' = (0/100, .5/125 , 1/150, .5/175, 0/200)

Subset ini mewakili suatu pembacaan *crisp* dalam temeratur 125 °, angka ini termasuk nilai *membership function* dari 0,5 untuk *fuzzy set* temperatur normal. hal ini menyebabkan *fuzzy set* B' (yaitu suatu anggota dalam B) yang ingin kita tentukan kemudian.

$$b_j = \max \{ \min(a'_i, m_{ij}) \}$$

$$b_1 = \max[\min(0., 0.), \min(.5, 0.), \min(0., 0.), \min(0., 0.), \min(0., 0.)]$$

$$b_2 = \max[\min(0., 0.), \min(.5, .5), \min(0., .6), \min(0., .5), \min(0., 0.)]$$

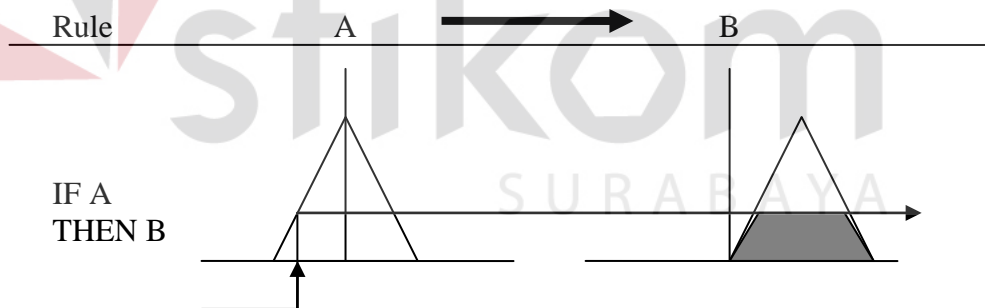
$$b_3 = \max[\min(0., 0.), \min(.5, .5), \min(0., 1.), \min(0., .5), \min(0., 0.)]$$

$$b_4 = \max[\min(0., 0.), \min(.5, .5), \min(0., .6), \min(0., .5), \min(0., 0.)]$$

$$b_5 = \max[\min(0., 0.), \min(.5, 0.), \min(0., 0.), \min(0., 0.), \min(0., 0.)]$$

$$B' = (0/10, .5/20, .5/30, .5/40, 0/50)$$

Akibatnya, bentuk *fuzzy set* adalah clipped versi dari B, dimana ketinggiannya ditentukan oleh A'. Hal ini merupakan akibat umum *max-min inference* seperti pada gambar 2.7. Max-Min Inference. Inti dari contoh ini adalah hasil yang kita tentukan dengan membatasi A' ke dalam suatu nilai tunggal. Kita menyatakan temperatur yang terbaca adalah 125 ° yang memberikan kita vektor A' dari (0 .5 0 0 0) yang dihasilkan sebuah B' dari (0 .5 .5 .5 0).



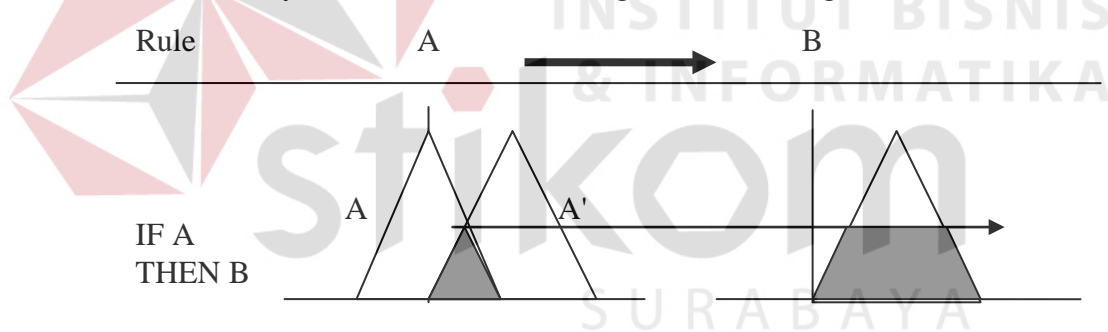
Gambar 2.7 Max-Min Inference

Dalam kebanyakan aplikasi yang nyata, suatu sistem *fuzzy logic* kita memiliki suatu nilai *crisp* dalam suatu perhitungan (misal $x_k = 125$ derajat). dengan perhitungan tunggal x_k kita dapat menggunakan $\mu_A(x_k)$ secara langsung dengan *fuzzy set* yang mewakili B, dengan nama $\mu_B(y)$, untuk menyebabkan terbentuknya suatu fuzzy set B' : $B' = \mu_A(x_k) \wedge \mu_B(y)$.

Untuk cepatnya dalam contoh diatas kita mengasumsikan temperatur 125° yang diikuti dengan $\mu_A = 0,5$ dan

$$\begin{aligned} B' &= [\min(.5, 0.), \min(.5, .6), \min(.5, .1), \min(.5, .6), \min(.5, 0.)] \\ &= (0, .5, .5, .5, 0). \end{aligned}$$

Ini adalah hasil yang sama seperti pada matrix fuzzy. Oleh karena itu, ketika memasukkan informasi dalam form crisp, kita tidak perlu menghitung *matrix fuzzy*, tetapi dapat dikerjakan dengan mudah dan lebih sederhana dalam informasi fuzzy set. Walaupun dimasukkan dalam rule yang mewakili pembacaan *fuzzy*, kita masih menggunakan pendekatan sederhana. Dengan melihat rule IF A THEN B, dan pembacaan fuzzy A yang digambarkan dalam A'. Kita dapat mengambil secara mudah interseksi keduanya sebagai input $\min(a'_i, a'_j)$ untuk membentuk fuzzy set B'. Pendekatan ini digambarkan sebagai berikut :



Gambar 2.8 Max-Min Inference untuk fuzzy input

C. Defuzzifikasi

Defuzzifikasi yang mentransformasi hasil *fuzzy* ke bentuk keluaran yang crisp. *Membership function* digunakan dalam menterjemahkan keluaran *fuzzy* ke bentuk keluaran *crisp*. Terjemahan kembali disini dapat menggunakan beberapa metode, dimana diantaranya adalah CoM (*Center of Maximum*) / CoG (*Center of*

Gravity), CoA (*Center of Area*), MoM (*Mean of Maximum*), MoM BSUM (*Mean of Maximum Bounded Sum*).

2.3.6. Himpunan fuzzy

Teori himpunan tradisional menggambarkan dunia sebagai hitam dan putih. Ini berarti sebuah obyek berada didalam atau diluar himpunan yang diberikan. Dalam teori himpunan tradisional untuk anggota diberi nilai 1 dan untuk bukan anggota diberi nilai 0; ini disebut himpunan *crisp*. Sebagai contoh anggota himpunan orang muda dapat berisi hanya orang yang berumur kurang dari 10. Penggunaan interpretasi ini pada seseorang yang berulang tahun ke-11, maka orang tersebut bukan anggota himpunan orang muda.

Himpunan fuzzy memberikan nilai keanggotaan antara 0 dan 1 yang menggambarkan secara lebih alami sebuah kumpulan anggota dengan himpunan. Sebagai contoh, jika seorang berumur 5 tahun dapat diberikan nilai keanggotaan 0.9 atau jika umurnya 13 tahun nilai keanggotaannya 0.1. Dalam contoh ini “umur” adalah variabel linguistik dan “muda” adalah salah satu himpunan fuzzy.

Definisi himpunan Fuzzy :

Misalkan X semesta pembicaraan, dengan elemen dari X dinotasikan x. Sebuah himpunan fuzzy A dari X dikarakteristikkan dengan fungsi keanggotaan

$$\mu_A(x) : X \rightarrow [0,1]$$

Pada logika fuzzy, kejadian atau elemen x diberikan nilai keanggotaan dengan fungsi keanggotaan μ . Nilai ini mempresentasikan derajat keanggotaan elemen x pada himpunan fuzzy A.

$$\mu_A(x) = \text{Degree}(x \in A)$$

Nilai keanggotaan dari x berada pada interval :

$$0 \leq \mu_A(x) \leq 1$$

Himpunan fuzzy adalah perluasan dari teori himpunan tradisional. Himpunan fuzzy menyamakan konsep keanggotaan dengan menggunakan fungsi keanggotaan μ yang menghasilkan nilai antara 0 dan 1 yang mempresentasikan derajat keanggotaan obyek x pada himpunan A .

2.3.7. Membentuk himpunan fuzzy

Untuk mempresentasikan himpunan fuzzy dalam komputer perlu didefinisikan fungsi keanggotaannya. Sebagai contoh : orang tinggi. Dapat dinyatakan pada setiap individu, pada tingkatan mana bahwa mereka yakin seseorang itu dikatakan tinggi. Setelah mengumpulkan jawaban untuk interval ukuran tinggi, dapat disajikan tingkat rata-rata untuk menghasilkan suatu himpunan fuzzy dari orang-orang yang tinggi. Fungsi ini dapat digunakan sebagai suatu keyakinan (nilai keanggotaan). Bagi individu yang menjadi anggota himpunan fuzzy dari orang tinggi.

Dengan membentuk fuzzy subset untuk berbagai bentuk fuzzy, dapat dianggap nilai keanggotaan dari obyek yang diberikan pada setiap himpunan. Pendekatan lain yang sering ditemukan pada praktek untuk membentuk himpunan fuzzy sangat berhubungan dengan interpretasi dari seorang ahli. Seperti teknik pengumpulan data, dapat ditanyakan pada pakar untuk kepercayaannya bahwa berbagai obyek merupakan bagian himpunan yang diberikan.

2.3.8. Batasan (*Hedges*)

Dalam pembicaraan normal, manusia mungkin menambahkan kekaburan untuk memberikan pernyataan dengan menggunakan kata keterangan seperti

sangat, agak. Kata keterangan adalah sebuah kata yang memodifikasi kata benda, kata sifat, kata keterangan lain, atau keseluruhan kalimat. Sebagai contoh, kata keterangan memodifikasi kata sifat, “orang itu sangat tinggi”.

Sebuah hedges memodifikasi himpunan fuzzy yang sudah ada secara matematis untuk menghitung beberapa kata keterangan yang ditambahkan.

2.3.9. Operasi himpunan fuzzy

Terdapat 3 operasi dalam himpunan fuzzy, yaitu :

1. Irisan (*Intersection*)

Dalam teori himpunan klasik, irisan dari dua himpunan berisi elemen-elemen yang sama dari keduanya. Dalam himpunan fuzzy, sebuah elemen mungkin sebagian dalam kedua himpunan. Oleh karena itu ketika mengingat irisan dari kedua himpunan, tidak dapat dikatakan bahwa sebuah elemen adalah lebih mungkin menjadi dalam irisan daripada dalam suatu himpunan asli.

2. Gabungan (*Union*)

Cara kedua dari penggabungan himpunan fuzzy adalah gabungannya. Penggabungan dari dua himpunan adalah terdiri dari dua himpunan adalah terdiri dari elemen-elemen yang menjadi satu atau dua himpunan. Dalam situasi ini anggota dari gabungan tidak dapat mempunyai nilai keanggotaan yang kurang dari nilai keanggotaan yang lain dari himpunan aslinya.

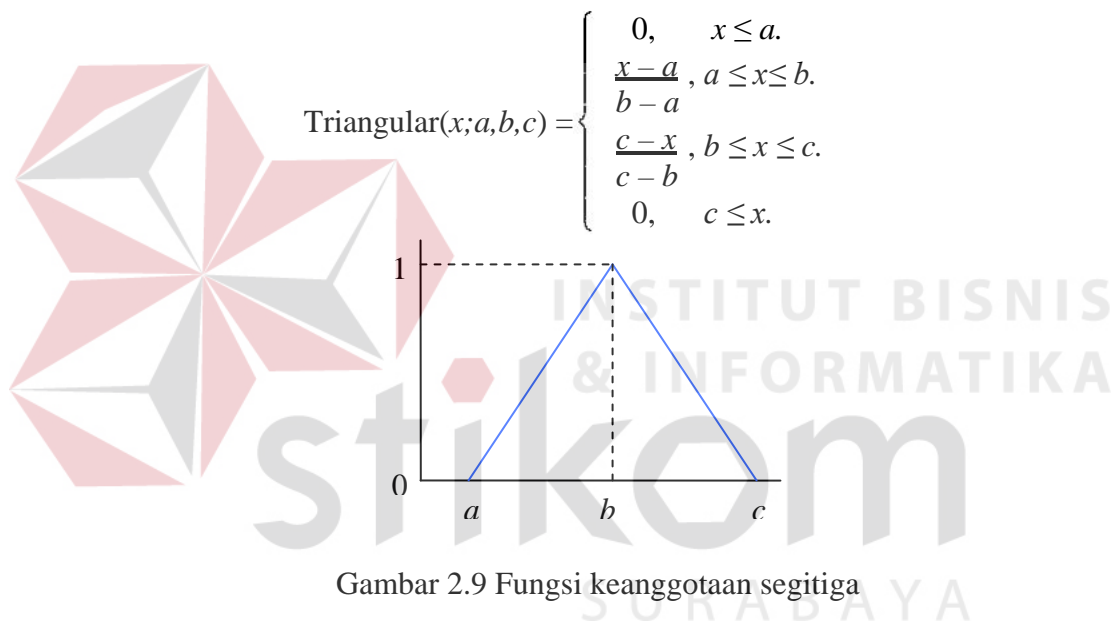
3. Komplemen (*Complement*)

Komplemen dari himpunan fuzzy A dinotasikan dengan $(\sim A)$ dinyatakan dengan persamaan sebagai berikut : $\mu_{\sim A}(x) = 1 - \mu_A(x)$

2.3.10. Fungsi keanggotaan

Fungsi keanggotaan digunakan dalam mempresentasikan himpunan fuzzy. Dalam *fuzzy* fungsi keanggotaan yang biasa dipakai adalah fungsi keanggotaan segitiga, trapesium, Gaussian, fungsi keanggotaan S, fungsi keanggotaan lonceng dan sebagainya. Dalam sistem ini fungsi keanggotaan digunakan adalah fungsi keanggotaan segitiga :

- Fungsi keanggotaan segitiga



Gambar 2.9 Fungsi keanggotaan segitiga

2.4. Analisa dan Perancangan Sistem

2.4.1. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah gambaran pada sistem dimana didalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdidefinisikan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai atribut yang merupakan ciri *entity* tersebut. Sedangkan relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity*.

Adapun elemen-elemen dari ERD ini adalah :

- Entitas
- Atribut
- Pengidentifikasi
- Hubungan atau relasi

2.4.2. DFD (*Data Flow Diagram*)

Meskipun suatu analisa yang disebut dengan DFD mempunyai struktur tersendiri, namun sistem analisa dapat meletakkan secara bersamaan sebuah gambar yang merepresentasikan seluruh proses-proses data dalam sebuah organisasi. Pendekatan *data flow* menitik beratkan pada logika yang tersirat dari suatu sistem.

Dengan menggunakan kombinasi simbol, sistem analisa dapat membuat sebuah gambaran dari suatu proses yang sebenarnya dengan menggunakan dokumen sistem.

A. Keuntungan pembuatan data flow

Data flow mempunyai lima keuntungan utama dari penjelasan-penjelasan jalannya data dalam sistem, yaitu :

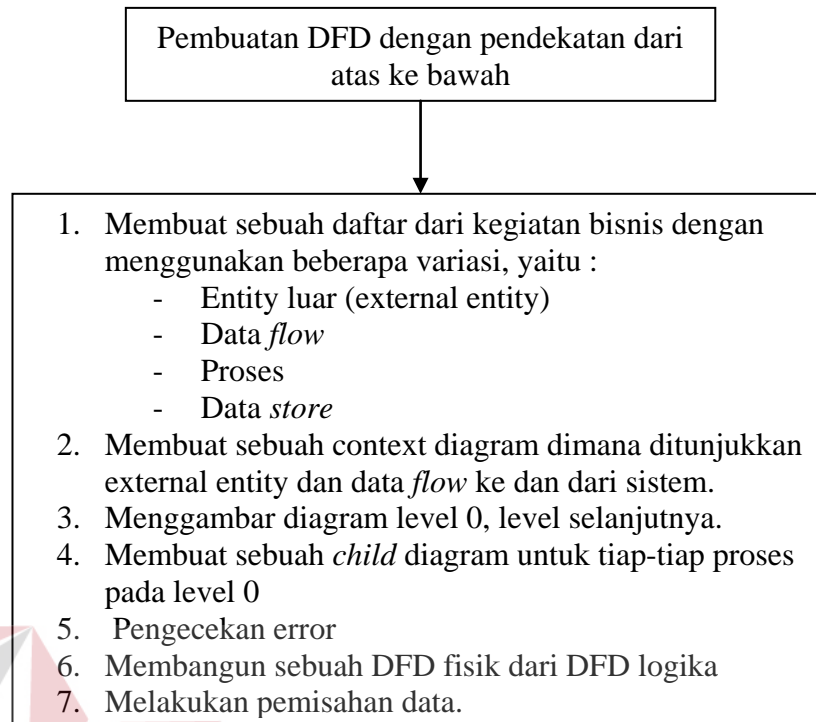
1. Kebebasan yang berasal dari kepercayaan untuk mengimplementasikan secara benar teknik sistem dari suatu sistem yang baru.
2. Memberikan pengertian dari hubungan sistem-sistem dan subsistem yang ada.
3. Komunikasi mengenai pengetahuan sistem bagi *user* melalui DFD
4. Analisa dari sebuah usulan sistem untuk menentukan jika data dan proses-proses yang ada dapat didefinisikan secara mudah.

5. Penggunaan data *flow* merupakan keuntungan tambahan yang dapat digunakan sebagai latihan bagi sistem analis, kesempatan sistem analis menjadi lebih baik untuk mengerti tentang hubungan sistem dan subsistem yang ada didalamnya.

Keuntungan dari kelima penggunaan data *flow* tersebut dapat digunakan sebagai *tools* yang interaktif dengan *user*. Hal yang menarik dalam penggunaan DFD adalah ditunjukkannya kepada *user* gambaran-gambaran secara lengkap dari sistem. *User* dapat menanyakan guna memberikan komentar pada konsep, sistem analis dapat merubah sistem berdasarkan keinginan *user*. Keuntungan terakhir dari penggunaan DFD adalah dapat mengikuti sistem analis untuk mendeskripsikan komponen-komponen yang digunakan dalam suatu diagram. Analisa dapat ditampilkan untuk menjamin bahwa semua *output* mempunyai isi atau memperoleh data inputan dari prosesnya.

B. Pembuatan DFD (Data Flow Diagram)

DFD dapat dan harus digambarkan secara sistematis. Pertama, dibutuhkan sistem analis untuk mengkonsep data *flow*, dari atas ke bawah seperti ditunjukkan pada gambar berikut :



Gambar 2.10 Pembuatan data flow diagram

Untuk memulai sebuah DFD dari suatu sistem biasanya dituangkan dalam sebuah daftar dengan empat kategori yaitu entity luar, arus data, proses, dan penyimpanan data. Daftar ini akan membantu menentukan batasan-batasan dari suatu sistem yang akan digambarkan. Pada dasarnya daftar itu berisi elemen-elemen data yang dikarang. Elemen-elemen tersebut terdiri dari :

a. Pembuatan konteks diagram

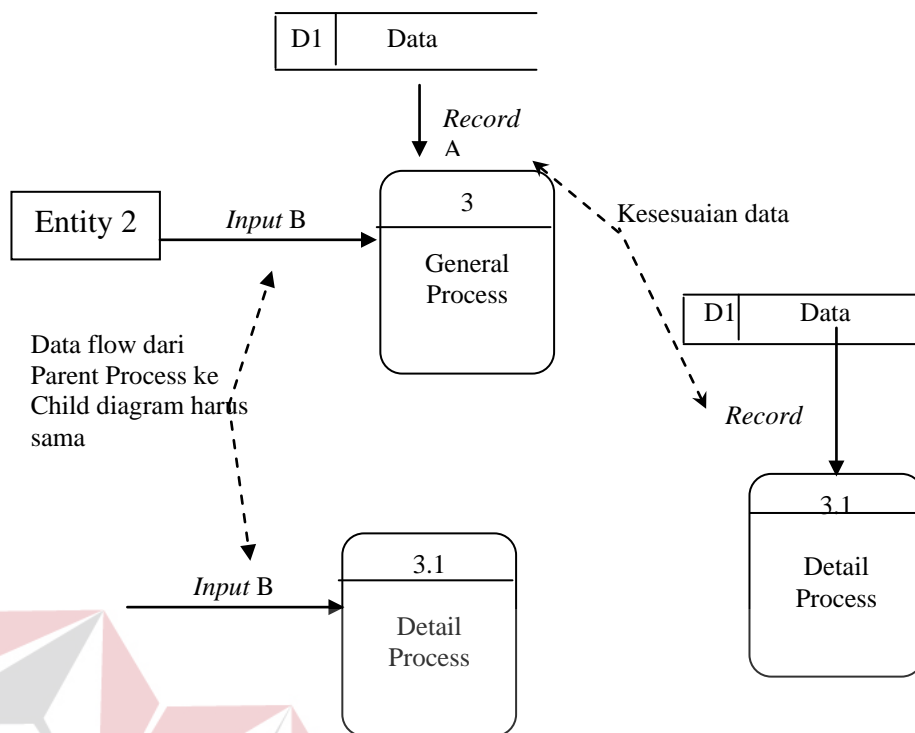
Konteks diagram adalah level yang tertinggi dalam sebuah DFD dan hanya berisi satu proses serta merupakan representasi dari sebuah sistem. Proses dimulai dengan penomoran ke-0 dan untuk seluruh entity luar akan ditunjukkan dalam konteks diagram yang sama seperti data awal yang dikirim dari entity luar. Konteks diagram tidak berisi penyimpanan data.

b. Pembuatan diagram level 0 serta level berikutnya

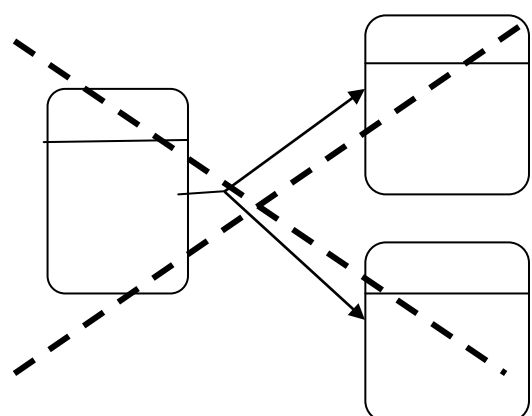
Diagram level 0 dihasilkan oleh konteks diagram dan berisi proses-proses. Pengisian proses-proses yang berlebihan pada level ini akan menghasilkan sebuah diagram yang salah, sehingga sulit untuk dimengerti. Masing-masing proses diberikan penomoran dengan sebuah bentuk integer. Umumnya dimulai dari kiri atas dan penyelesaiannya di kanan bawah dalam sebuah bentuk diagram.

c. Pembuatan *child* diagram

Child diagram diberikan nomor yang sama seperti proses di atasnya (parent proses) dalam diagram level 0. Contohnya, proses 3 harus diturunkan ke diagram 3, proses pada *child* diagram menggunakan penomoran unik untuk masing-masing proses dengan mengikuti penomoran proses di atasnya. Contohnya, dalam diagram 3 proses-proses diberikan nomor 3.1, 3.2, 3.3 dan seterusnya. Konversi ini diikuti oleh analisis sistem untuk menelusuri seri-seri dari proses-proses yang dikeluarkan oleh beberapa level, jika pada proses diagram level 0 digambarkan sebagai 1, 2, , dan 3 maka *child* diagram-diagramnya adalah 1, 2, dan 3 pada level yang sama. Ilustri level detail dengan sebuah *child* DFD dapat ditunjukkan pada gambar :

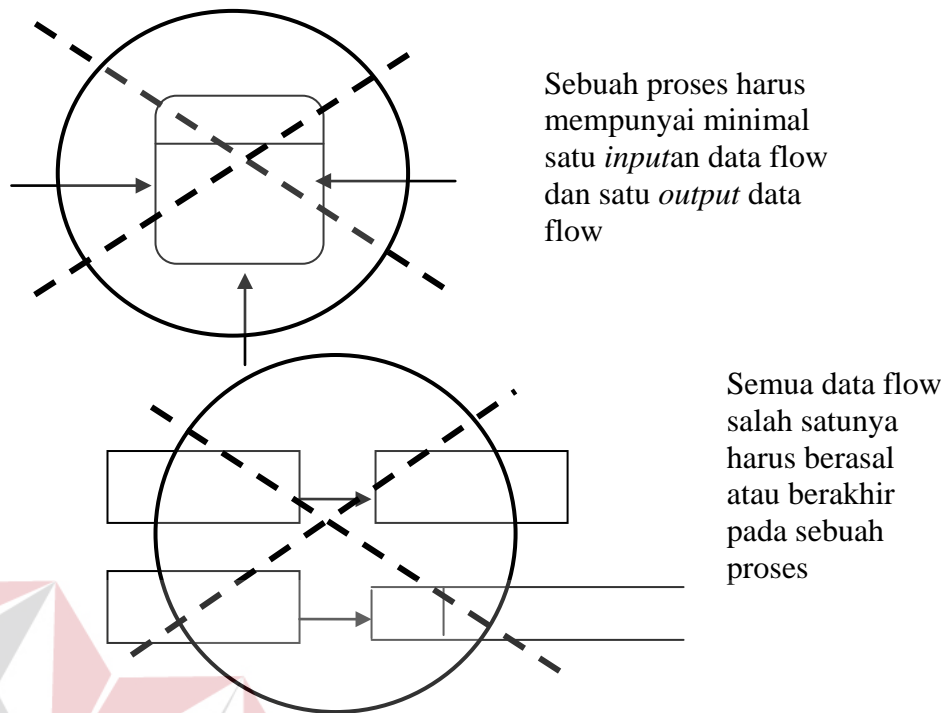
Gambar 2.11 Pembuatan *Child Diagram*

- d. Pengecekan kesalahan-kesalahan pada diagram digunakan untuk melihat kesalahan-kesalahan yang terdapat pada sebuah DFD. Beberapa kesalahan-kesalahan yang umum terjadi ketika penggambaran/ pembuatan DFD, ditunjukkan pada gambar berikut, adalah :



Sebuah data flow tidak diperbolehkan mempunyai percabangan / memisahkan diri (flow) ke dalam dua atau lebih data flow yang berbeda

Gambar 2.12 Contoh 1 kesalahan penggambaran DFD

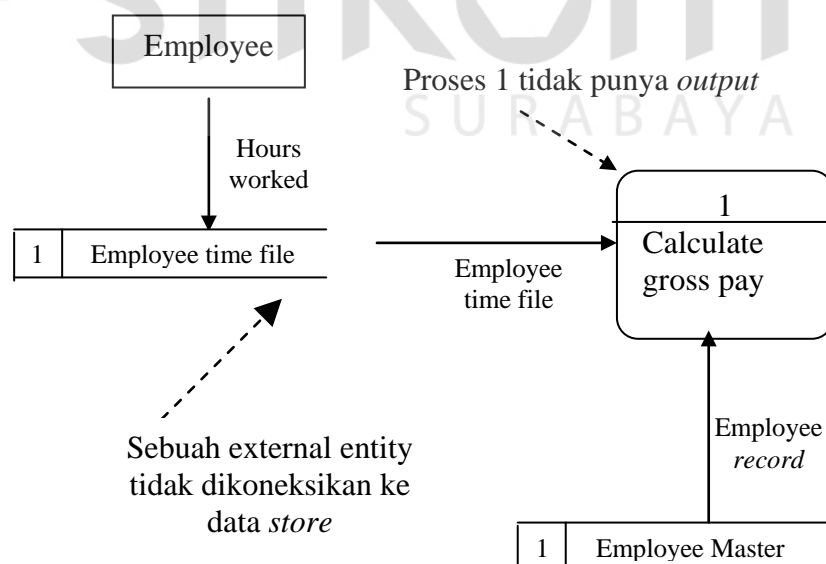


Gambar 2.13 Contoh 2 kesalahan penggambaran DFD

1. Lupa untuk menginputkan sebuah arus data atau arah panah langsung. Sebagai contoh adalah penggambaran proses yang menunjukkan sebuah data *flow* seperti *input* atau seperti *output*. Tiap-tiap proses pengubahan data harus menerima *input* dan *output*. Tipe kesalahan ini terjadi ketika sistem analis lupa memasukkan sebuah data *flow* atau meletakkan sebuah arah panah ditempat yang salah.
2. Hubungan penyimpanan data dan entity luar secara langsung satu sama lain. Data *store* dan entity tidak mungkin dikoneksikan satu sama lain ; data *store* dan entity luar harus dikoneksikan melalui sebuah proses.
3. Kesalahan penamaan (label) pada proses-proses atau data *flow*. Pengecekan DFD untuk memastikan bahwa tiap-tiap objek atau data *flow* telah diberikan label. Sebuah proses haruslah di indikasikan seperti nama dari sistem atau

menggunakan format kata kerja-kata benda. Tiap data *flow* haruslah dideskripsikan dengan sebuah kata benda.

4. Memasukkan lebih dari sembilan proses dalam sebuah DFD. Memiliki banyak proses akan mengakibatkan kekacauan pada diagram sehingga dapat menyebabkan kebingungan dalam pembacaan sebuah proses dan akan menghalangi tingkat komunikasi. Jika lebih dari sembilan proses dalam sebuah sistem, maka beberapa grup dalam proses dilakukan bersama-sama ke dalam sebuah sub sistem dan meletakkannya dalam sebuah *child* diagram.
5. Menghilangkan suatu arus data. Pengujian dari suatu diagram yang menunjukkan garis / arah (*flow*), dimana untuk setiap proses data *flow* hanya mempunyai *input* data, *output* kecuali dalam kasus dari detil (*child*). Setiap *child* data dari DFD, arah arus data seringkali digambarkan untuk mengidentifikasi bahwa diagram tersebut kehilangan data *flow*. Seperti di tunjukkan pada gambar :



Gambar 2.14 Contoh mengidentifikasi diagram kehilangan data *flow*

6. Buat ketidaksesuaian komposisi dalam *child* diagram , dimana tiap *child* diagram harus mempunyai *input* dan *output* arus data yang sama seperti proses dilevel atasnya (parent proses). Pengecualian untuk rule ini adalah kurangnya *output*, seperti kesalahan garis yang ada didalam *child* diagram.

C. Perbedaan DFD (Logika dan Fisik)

Disain	Logika	Fisik
Gambaran model	Operasi-operasi bisnis	Bagaimana sistem akan diimplementasikan (atau bagaimana sistem dijalankan)
Apa yang ditampilkan oleh proses	Aktivitas bisnis	Program-program, modul program, dan prosedur-prosedur manual
Apa yang ditampilkan oleh data store	Koleksi-koleksi dari data yang dikesampingkan dari bagaimana data tersebut di simpan	File-file fisik dan database-database dari file-file manual
Kontrol sistem	Menunjukkan kontrol-kontrol bisnis	Menunjukkan kontrol-kontrol untuk validasi <i>input</i> data, untuk memperoleh sebuah <i>record</i> , untuk memastikan kesuksesan proses dan untuk keamanan sistem