BAB II

KAJIAN TEORI

A. Lampu Lalu Lintas

Lampu lalu lintas ialah peralatan yang dioperasikan secara mekanis atau elektrik yang berfungsi mengatur kendaraan-kendaraan agar berhenti atau berjalan. Peralatan standar untuk lampu lalu lintas terdiri dari sebuah tiang dan dibagian atas terdapat berwarna merah, kuning, dan hijau yang menyala secara bergantian. Dengan warna merah mengatur supaya kendaraan berhenti, lampu kuning supaya kendaraan pelan-pelan, dan lampu hijau untuk kendaraan jalan.

1. Definsi dalam Operasional Lampu Lalu Lintas

   Menurut Direktorat Jenderal Bina Marga (1997: 2-9) definisi yang digunakan dalam operasional lampu lalu lintas diantaranya sebagai berikut:

   a. Fase

      Satu tahapan sinyal lalu lintas dengan satu atau lebih pergerakan lalu lintas mendapatkan kesempatan bergerak.

   b. Waktu siklus

      Waktu diantara lampu hijau mulai menyala sampai waktu hijau kembali menyala di dalam simpang yang sama.

   c. Waktu antar hijau (fase *clear*)

      Periode waktu lampu menyala merah semua antara dua fase yang berurutan. Hal ini dimaksudkan supaya akhir rombongan kendaraan pada fase sebelumnya tidak berbenturan dengan awal rombongan kendaraan pada fase berikutnya.

d. Konflik lalu lintas

Suatu keadaan dua kendaraan atau lebih yang saling mendekati satu sama lain dan akan terjadi kecelakaan apabila gerakan keduanya tetap atau tidak berubah.

2. Fungsi Lampu Lalu Lintas

Menurut Direktorat Jenderal Bina Marga (1997: 2-2) pada umumnya lampu lalu lintas digunakan untuk satu atau lebih dari alasan berikut:

a. Menghindari kemacetan simpang akibat adanya konflik arus lalu-lintas,
b. Memberi kesempatan kepada kendaraan dan/atau pejalan kaki dari jalan simpang untuk memotong jalan utama,
c. Mengurangi kecelakaan lalu-lintas antara kendaraan-kendaraan dari arah yang bertentangan.

3. Jenis Lampu Lalu Lintas

Berdasarkan cakupanya, jenis kendali dengan lampu lalu lintas pada suatu simpang jalan dibedakan sebagai berikut:

a. Lampu lalu lintas terpisah

Pengoperasian lampu lalu lintas dengan perancanganya hanya berdasar pada pertimbangan satu simpang jalan saja, tidak mempertimbangkan jarak, waktu lampu lalu lintas, dan kepadatan dari simpang jalan yang terdekat.

b. Lampu lalu lintas terkoordinasi

Pengoperasian lampu lalu lintas dengan perancanganya berdasar pada pertimbangan beberapa simpang jalan yang terdapat pada suatu jalur atau arah tertentu.

c. Lampu lalu lintas jaringan

Pengoperasian lampu lalu lintas dengan perancanganya berdasar pada pertimbangan beberapa simpang jalan yang terdekat dalam suatu jaringan jalan dalam suatu kawasan.

Berdasarkan cara pengoperasianya, jenis kendali lampu lalu lintas pada simpang jalan dibedakan sebagai berikut:

a. *Fixed time traffic signals*

Pengoperasian lampu lalu lintas dengan pengaturan waktu (*setting time*) tidak mengalami perubahan atau waktunya tetap sepanjang hari selama 24 jam.

b. *Actuated traffic signals*

Pengoperasian lampu lalu lintas dengan pengaturan waktu (*setting time*) mengalami perubahan dari waktu ke waktu sesuai dengan kepadatan kendaraan suatu simpang.

B. Teori Himpunan *Fuzzy*

Teori himpunan *fuzzy* merupakan perluasan dari teori himpunan tegas (*crisp set*). Teori himpunan *fuzzy* diperkenalkan oleh Prof Lotfi A.Zadeh dari Universitas California di Berkeley pada 1965 dan sekarang telah banyak digunakan di bidang industri, kedokteran, lalu lintas dan lain sebagainya (Wibisono, 2008: 49).

1. Himpunan *Fuzzy*

   Pada teori himpunan *fuzzy*, nilai keanggotaan berada dalam interval 0 sampai 1. Berbeda dengan teori himpunan tegas yang nilai keanggotaannya hanya 0 dan 1.

   Contoh 2.1

   Himpunan A merupakan himpunan merk mobil yang banyak digunakan masyarakat pengguna. Dalam teori himpunan tegas, himpunan A ditulis

   A = {Toyota, Honda, Daihatsu, Nissan, Suzuki}

   Artinya Toyota, Honda, Daihatsu, Nissan, Suzuki merupakan anggota himpunan dengan nilai keanggotaan 1, selain kelima elemen di atas bukan anggota himpunan maka nilai keanggotaanya 0.

   Dari kelima anggota himpunan A tersebut tidak dapat diperoleh informasi mana yang sangat banyak, banyak, cukup, kurang ataupun sedikit diminati oleh masyarakat pengguna, karena derajat keanggotaan kelima anggota himpunan tersebut sama. Dalam teori himpunan *fuzzy*, himpunan A ditulis sebagai berikut:

   A={<Toyora;0,9>,<Honda;0,7>,<Daihatsu;0,5>,<Nissan;0,4>,
   <Suzuki;0,2>}

   Artinya, Toyota paling banyak diminati oleh masyarakat pengguna karena memiliki nilai keanggotaan 0,9 disusul Honda 0,7 dan seterusnya sampai merk mobil yang paling sedikit peminatnya yaitu Suzuki dengan keanggotaan 0,2.

Menurut Kusumadewi (2003: 158) himpunan *fuzzy* memiliki 2 atribut, yakni sebagai berikut:

a. Linguistik, ialah penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami.

b. Numeris, yakni suatu nilai (angka) yang menunjukan ukuran dari suatu variabel

Contoh 2.2

Misalkan diberikan himpunan *fuzzy* untuk variabel umur dibagi menjadi tiga kategori berikut:

Himpunan *fuzzy* muda = [0,50]

Himpunan *fuzzy* tua = [40, 100]

variabel 0,40,50, dan 100 merupakan atribut numeris

Sedangkan variabel muda, parobaya, dan tua merupakan atribut linguistik

2. Fungsi Keanggotaan (*Membership Function*)

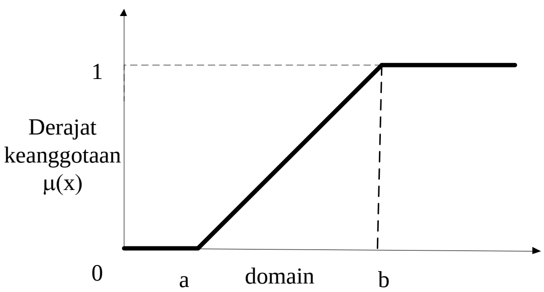
Fungsi keanggotaan adalah suatu kurva yang menunjukan pemetaan titik-titik *input* data (sumbu x) kepada nilai keanggotaan atau derajat keanggotaan yang mempunyai interval dari 0 sampai 1. Terdapat beberapa jenis fungsi yang bisa digunakan untuk mendapatkan nilai keanggotaan dalam fungsi keanggotaan. Menurut Kusumadewi (2003: 160) beberapa jenis fungsi tersebut diantaranya:

a. Representasi Linear

Menurut Kusumadewi (2003: 160) representasi linear adalah pemetaan *input* ke derajat keanggotaan yang digambarkan sebagai suatu garis lurus. Terdapat dua jenis himpunan *fuzzy* yang linear yakni representasi linear naik dan representasi linear turun. Jenis yang pertama yaitu representasi linear naik. Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol bergerak menuju nilai domain yang memiliki derajat keanggotaan yang lebih tinggi. Fungsi keanggotaan untuk kurva representasi linear naik adalah sebagai berikut:

$$\mu(x) = \begin{cases} 0 & , \quad x \leq a \\ \dfrac{x-a}{b-a} & , \quad a \leq x \leq b \\ 1 & , \quad b \leq x \end{cases}$$

Representasi grafik untuk fungsi keanggotaan linear naik ditunjukan pada Gambar 2.1 berikut:



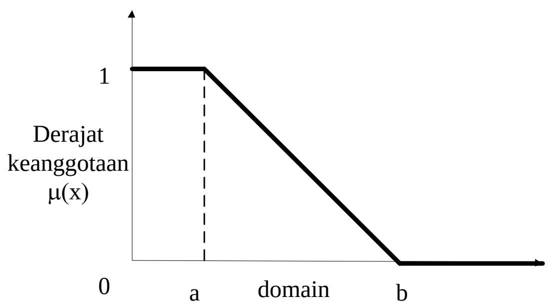
Gambar 2.1 Representasi Linear Naik

Jenis yang kedua ialah representasi linear turun. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi yakni satu pada sisi kiri, kemudian bergerak menuju nilai domain yang memiliki derajat keanggotaan lebih rendah (Kusumadewi, 2003: 161).

Fungsi keanggotaan untuk kurva representasi linear turun ialah sebagai berikut:

$$\mu(x) = \begin{cases} 1 \quad , & x \leq a \\ \dfrac{b-x}{b-a} \quad , & a \leq x \leq b \\ 0 \quad , & b \leq x \end{cases}$$

Representasi grafik untuk fungsi keanggotaan linear turun ditunjukan pada Gambar 2.2 Sebagai berikut:



Gambar 2.2 Representasi Linear Turun

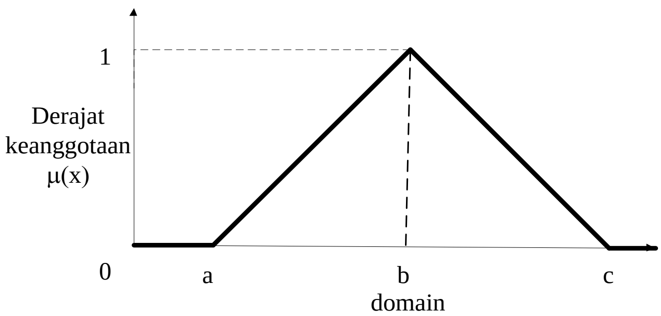
b. Representasi Kurva Segitiga (*triangular*)

Representasi kurva segitiga adalah pemetaan *input* data ke derajat keanggotaan yang digambarkan sebagai suatu kurva berbentuk segitiga. Pada dasarnya kurva segitiga merupakan gabungan antara 2 garis linear (Kusumadewi, 2003: 162).

Fungsi keanggotaan untuk representasi kurva segitiga ialah sebagai berikut:

$$\mu(x) = \begin{cases} 0 & , \quad x \leq a \\ \dfrac{x-a}{b-a} & , \quad a \leq x \leq b \\ \dfrac{c-x}{c-b} & , \quad b \leq x \leq c \\ 0 & , \quad c \leq x \end{cases}$$

Representasi grafik fungsi keanggotaan segitiga di atas ditunjukan pada Gambar 2.3 Berikut:



Gambar 2.3 Representasi Kurva Segitiga

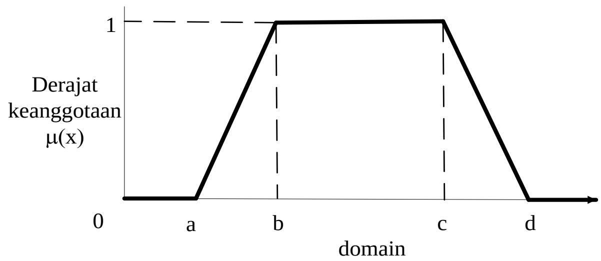
c. Representasi Kurva Trapesium

Representasi kurva trapesium merupakan pemetaan *input* data ke derajat keanggotaan yang digambarkan sebagai suatu kurva dengan bentuk trapesium. Pada dasarnya kurva trapesium ini memiliki bentuk segitiga, tetapi terdapat perbedaan pada beberapa titik yang memiliki keanggotaan 1 (Kusumadewi, 2003: 163).

Fungsi keanggotaan untuk representasi kurva trapesium ialah sebagai berikut:

$$\mu(x) = \begin{cases} 0 & , \quad x \leq a \\ \dfrac{x-a}{b-a} & , \quad a \leq x \leq b \\ 1 & , \quad b \leq x \leq c \\ \dfrac{d-x}{d-c} & , \quad c \leq x \leq d \\ 0 & , \quad d \leq x \end{cases}$$

Representasi grafik untuk fungsi keanggotaan trapesium diatas ditunjukan pada Gambar 2.4 sebagai berikut:

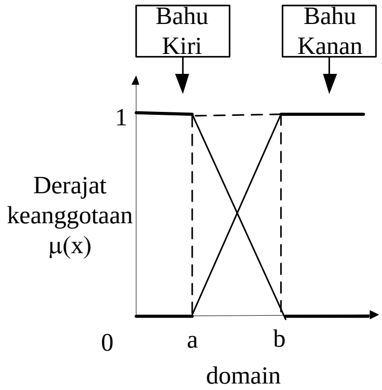


Gambar 2.4 Representasi Kurva Trapesium

d. Representasi Kurva Bentuk Bahu

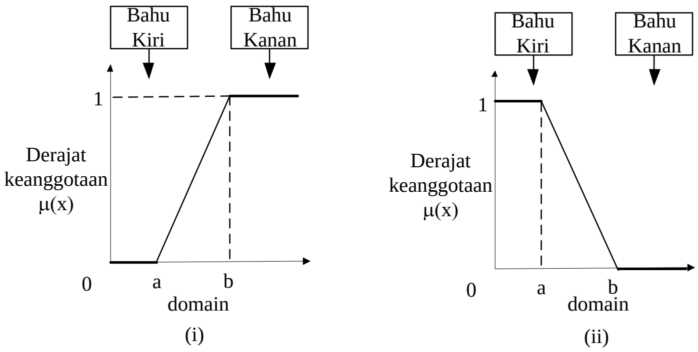
Menurut Kusumadewi (2003: 165) Representasi kurva bentuk bahu ialah pemetaan *input* ke derajat keanggotaan yang digambarkan sebagai suatu garis lurus yang konstan tanpa kenaikan maupun penurunan derajat keanggotaan. Kurva bahu tidak hanya merepresentasikan satu buah himpunan, melainkan terdiri dari beberapa himpunan. Berbeda dengan kurva linear, kurva segitiga, dan kurva trapesium yang merepresentasikan salah satu himpunan saja.

Representasi grafik untuk kurva bentuk bahu ditunjukan pada Gambar 2.5 Sebagai berikut:



Gambar 2.5 Representasi Kurva Bentuk Bahu Satu

Representasi kurva bentuk bahu pada Gambar 2.5 terdiri dari dua himpunan fungsi keanggotaan sebagai berikut:



Gambar 2.6 Representasi Kurva Bentuk Bahu Dua

Fungsi keanggotaan untuk representasi kurva bahu ialah sebagai berikut: $\mu(x) = \begin{cases} 0 & , \quad x \leq a \\ 1 & , \quad b \leq x \end{cases}$ (i) ; $\mu(x) = \begin{cases} 1 & , \quad x \leq a \\ 0 & , \quad b \leq x \end{cases}$ (ii)

3. Operator Dasar untuk Operasi Himpunan *Fuzzy*

Seperti halnya himpunan tegas, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Menurut Kusumadewi (2003: 175) ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu:

a. Operator *AND*

Operator ini berhubungan dengan operasi interseksi pada himpunan, diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangutan.

$$\mu_{A \cap B} = \{\mu_A(x), \mu_B(x)\}$$

b. Operator *OR*

Operator ini berhubungan dengan operasi union pada himpunan. Operator *OR* diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan

$$\mu_{A \cup B} = \max\{\mu_A(x), \mu_B(x)\}$$

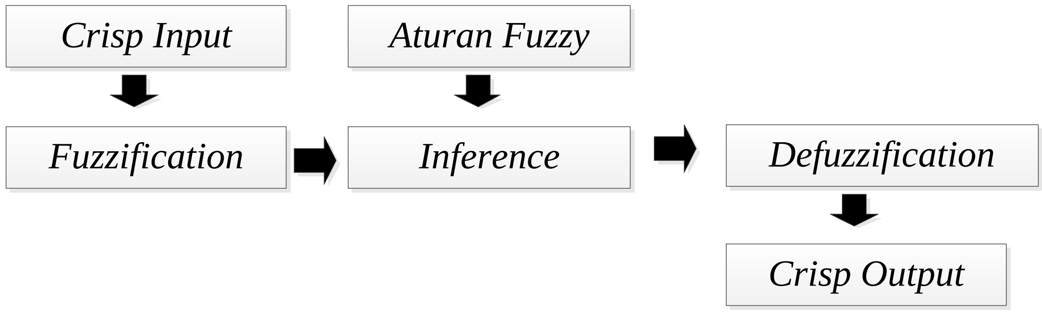
c. Operator *NOT*

Operator ini berhubungan dengan operasi komplemen pada himpunan. Operator *NOT* diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

4. *Fuzzy Inference System* (FIS)

Menurut Suyanto (2014: 105) *Fuzzy Inference System* (FIS) terdiri dari tiga komponen utama: *Fuzzification, Inference,* dan *Defuzzification*.



Gambar 2.7 Alur *Fuzzy Inference System* (Trimartanti & Abadi, 2015: 30)

a. Fuzzification

Proses *fuzzification* merupakan proses mengubah *input* tegas menjadi variabel *fuzzy*.

Contoh 2.3

Misalkan himpunan *fuzzy* pada Contoh 2.2 diberikan fungsi keanggotaan sebagai berikut:

$$\mu_{muda}(x;0,25,50) = \begin{cases} \frac{x-0}{25-0} & , \ 0 \leq x \leq 25 \\ \frac{50-x}{50-25} & , \ 25 \leq x \leq 50 \\ 0 & , \ 50 \leq x atau\ x \leq 0 \end{cases}$$

$$\mu_{tua}(x;40,70,100) = \begin{cases} \dfrac{x-40}{70-40} & , \quad 40 \leq x \leq 70 \\ \dfrac{100-x}{100-70} & , \quad 70 \leq x \leq 100 \\ 0 & , \ 100 \leq x\ atau\ x \leq 40 \end{cases}$$

Berdasarkan pada fungsi keanggotaan di atas akan dicari nilai keanggotaan untuk umur 30 tahun

$$\mu_{muda} = \frac{50-30}{50-25} = \frac{20}{25} = 0{,}8$$

Jadi umur dikonversi menjadi muda dengan derajat keanggotaan sama dengan 0,8

b. *Inference*

Inference merupakan tahap evaluasi aturan *fuzzy*. Tahap evaluasi dilakukan berdasarkan penalaran dengan menggunakan *input fuzzy* dan aturan *fuzzy* sehingga diperoleh *output* berupa himpunan *fuzzy*. Dari beberapa macam *inference fuzzy* yang dikenalkan para peneliti, berikut akan dijelaskajn metode Mamdani dan metode Sugeno yang banyak digunakan dalam berbagai penelitian.

1) Metode Mamdani

Metode Mamdani pertama kali diperkenalkan oleh Ibrahim Mamdani pada tahun 1975. Menurut Nuraida et al (2013: 544) Metode Mamdani memiliki kelebihan yakni, lebih intuitif. Metode ini merupakan metode yang paling sederhana dan paling sering digunakan untuk penelitian dibandingkan metode yang lain. Pada metode ini, aturan *fuzzy* didefinisikan sebagai:

IF x_1 *is* A_1 *AND* x_2 *is* A_2*... AND* x_n *is* A_n *THEN z is B* ,

dengan $A_1, \ldots, A_n$, dan *B* ialah nilai-nilai linguistik dan *"*x_1 *is* A_1*"* menyatakan bahwa nilai variabel x_1 ialah anggota *fuzzy set* A_1.

2) Metode Sugeno

Metode ini dikenal juga sebagai **Takagi-Sugeno-Kang** (TSK), yakni suatu varian dari metode mamdani. Metode ini juga menggunakan himpunan *fuzzy* pada *inputnya.* Akan tetapi, *output* yang digunakan ialah konstanta atau persamaan linear. Metode ini menggunakan aturan yang berbentuk:

IF x_1 *is* A_1 *AND* x_2 *is* A_2... *AND* x_n *is* A_n *THEN* $y = f(x_1, x_2, \dots, x_n)$

dengan *f* berupa sembarang fungsi dari variabel-variabel *input* yang nilainya berada dalam interval variabel *output.* Biasanya, fungsi ini dibatasi dengan menyatakan f sebagai kombinasi dari variabel-variabel *input*: $f(x_1, \dots, x_n) = w_0 + w_1 x_1 + \cdots + w_n x_n$,

dengan $w_0, w_1, \dots, w_n$ merupakan konstanta yang berupa bilangan *real* yang termasuk dalam bagian spesifikasi aturan *fuzzy*.

c. *Defuzzification*

Menurut Kusumadewi (2003: 190) terdapat berbagai metode *deffuzification* yang telah berhasil diaplikasikan untuk berbagai macam masalah, antara lain:

1) Metode Centroid

Pada metode ini, solusi *crisp* diperoleh dengan cara menetapkan z* sebagai titik pusat dari daerah fungsi keanggotaan *fuzzy*. Secara umum dirumuskan:

$$z^* = \frac{\int_z z\mu(z)dz}{\int_z \mu(z)dz}$$

untuk domain kontinu, dengan z merupakan nilai domain dan $\mu(z)$ merupakan nilai keanggotaan z.

dan

$$z^* = \frac{\sum_{j=1}^{n} z_j \mu(z_j)}{\sum_{j=1}^{n} \mu(z_j)}$$

untuk domain diskret, dengan z_j merupakan *output* pada aturan ke-j dan $\mu(z_j)$ merupakan nilai keanggotaan pada aturan ke-j sedangkan n merupakan banyaknya aturan yang digunakan.

2) Metode Bisektor

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan setengah dari total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan:

z_p sedemikian sehingga $\int_{R1}^{p} \mu\,(z)dz = \int_{p}^{Rn} \mu\,(z)dz$

3) Metode *Mean of Maximum* (MOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

4) Metode *Largest of Maximum* (LOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

5) Metode *Smallest of Maximum* (SOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

5. *Fuzzy Logic Toolbox* Matlab

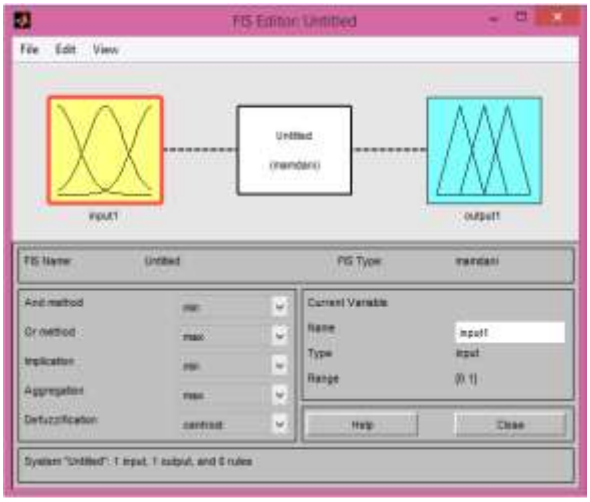
Fuzzy logic toolbox ialah sekumpulan *tool* yang membantu merancang sistem *fuzzy*. *Fuzzy logic toolbox* sangat *user friendly*, memungkinkan untuk berkreasi dengan bebas dalam rancang bangun *Fuzzy*

Inference System (FIS). Semua tool dalam *fuzzy* logic toolbox dikelompokan menjadi tiga kategori, yakni *command lines, Graphical User Interface* (GUI), *Simulink Block*. GUI sangat cocok untuk pemula, sementara *command lines* dan *simulink block* ditujukan untuk pemakai yang sudah berpengalaman. GUI memungkinkan akses fungsi-fungsi yang tersedia dalam *fuzzy logic toolbox*. Dalam kenyataanya *fuzzy logic toolbox* lebih banyak mengandalkan GUI dalam membantu penyelesaian kerja dalam rancang bangun *fuzzy inference system* (Naba, 2009: 79).

Menurut Naba (2009: 82-94) *Fuzzy Logic Toolbox* menyediakan 5 jenis GUI untuk keperluan rancang bangun *Fuzzy Inference System* (FIS), yakni:

a. *Fuzzy Inference System* (FIS) *editor*

Pada halaman perintah MATLAB, ketik: *fuzzy*, muncul FIS *Editor* seperti pada Gambar 2.8 dengan satu variabel masukan dengan label **input1** dan sebuah keluaran dengan label **output1**.

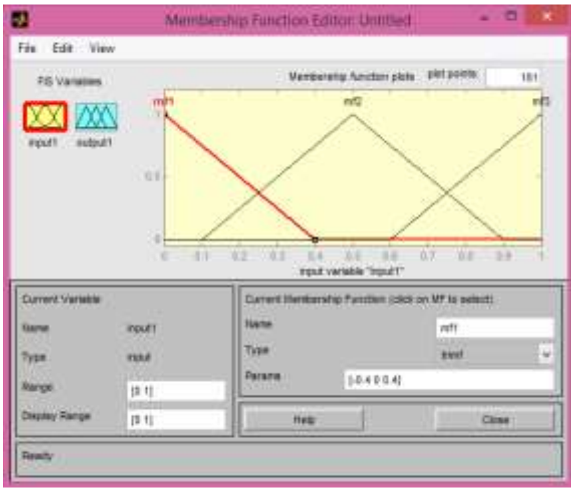


Gambar 2.8 FIS *Editor*

Mengubah label **input1**, klik kotak kuning berlabel **input1** lalu pada *Current Variable* di sebelah kanan bawah, pada kolom **Nama**, ganti label variabel **input1** sesuai yang diinginkan dan tekan **Enter**. Langkah yang sama berlaku untuk mengganti label **output1**. Opsi-opsi pada bagian kiri bawah tetap pada opsi yang sudah diberikan (*default*).

b. *Membership Function Editor*

Fungsi-fungsi keanggotaan variabel masukan dan keluaran didefinisikan melalui *Membership Function Editor* seperti ditunjukan dalam Gambar 2.9 . Membuka *Membership Function Editor* bisa dari FIS *Editor*, **View** → **Edit** atau dengan klik ganda ikon variabel masukan/keluaran.



Gambar 2.9 *Membership Function Editor*

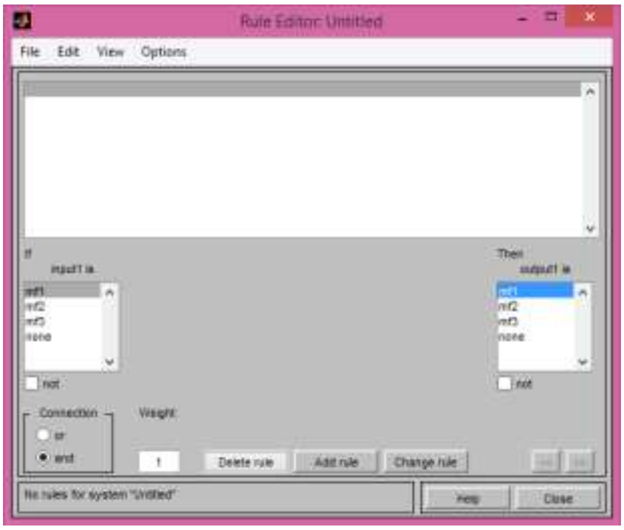
Fungsi keanggotaan untuk setiap variabel FIS bisa dihapus maupun ditambah melalui **Edit-Add Mfs...** . Pada bagian kiri bawah **Field Range** berfungsi untuk merubah semesta fungsi keanggotaan. Sedangkan untuk mendefinisikan fungsi-fungsi keanggotaan variabel

input1, klik ikon variabel FIS **input1** dan klik label **mf1** . Pada bagian kanan bawah, pada **Field Name** berfungsi merubah nama setiap fungsi keanggotaan. **Field Params** untuk fungsi segitiga terdapat 3 kolom, kolom pertama untuk merubah posisi kiri kurva, kolom kedua untuk merubah posisi tengah kurva, kolom ketiga untuk merubah posisi kanan kurva. Langkah yang sama juga berlaku untuk merubah variabel **output1**.

c. *Rule Editor*

Dengan GUI Rule Editor dapat dengan mudah mendefinisikan

if-then rule. Gambar 2.10 merupakan *Rule Editor* Window ketika belum ada masukan dan keluaran yang didefinsikan pada *Membership Function.*



Gambar 2.10 *Rule Editor*

Memasukan *rule* secara otomatis, yakni klik di **input1** untuk fungsi *if* dan **output1** untuk fungsi *then* kemudian klik **Add Rule.**

d. *Rule Viewer*

Rule Viewer menampilkan proses keseluruhan yang terjadi pada FIS. Membuka *Rule Viewer* dengan **View-Rules.** Untuk mengganti masukan klik pada kolom disebelah kiri bawah pada **Field Input.**

e. *Surface Viewer*

Surface viewer digunakan untuk menampilkan keluaran FIS dalam plot 3 dimensi. Membuka desktop Surface Viewer dengan **View-Surface.**

c. Graf

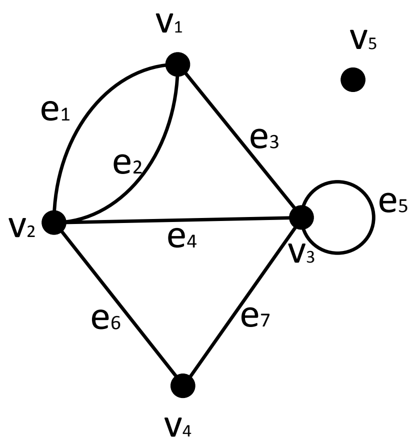
Teori graf merupakan pokok bahasan yang sudah ada sejak lama tapi masih memiliki banyak terapan hingga saat ini. Menurut catatan sejarah, masalah jembatan Könisberg merupakan masalah yang pertama kali menggunakan graf yakni pada tahun 1736 oleh seorang matematikawan Swiss, L.Euler. (Munir, 2010: 354). Secara matematis, graf didefiniskan sebagai berikut

Definisi 2.1. (Munir, 2010: 356) *Suatu graf G terdiri atas pasangan himpunan (V,E), ditulis dengan notasi G=(V,E), dua himpunan yaitu himpunan tak kosong V yang unsur-unsurnya disebut simpul (vertices atau node) dan himpunan E yang unsur-unsurnya disebut sisi (edges atau arcs) yang menghubungkan sepasang simpul.*

Simpul pada graf dapat dilabeli dengan huruf, seperti a,b,c,... dengan bilangan asli 1,2,3,... atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul dinyatakan dengan pasangan simpul tersebut atau biasa juga dinyatakan dengan $e_1, e_2, \dots, e_n$.

1. Terminologi Dasar

Berikut ini didefinisikan beberapa terminologi (istilah) yang sering digunakan dalam graf. Graf pada Gambar 2.11 akan digunakan untuk memperjelas terminologi yang didefinisikan.



Gambar 2.11 Graf

a. Bertetangga (*Adjacent*)

Menurut Munir (2010: 365) Dua buah simpul pada graf G dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain, u bertetangga dengan v jika (u, v) adalah sebuah sisi pada graf G. Pada Gambar 2.11 , simpul v_1 bertetangga dengan simpul v_2 dan v_3, tetapi simpul v_1 tidak bertetangga dengan simpul v_4 dan v_5.

b. Beririsan (*Incident*)

Untuk sembarang sisi $e = (u, v)$, sisi e dikatakan bersisian dengan simpul u dan v. Pada Gambar 2.11 , sisi (v_1, v_2) bersisian dengan simpul v_1 dan simpul v_2, untuk contoh sisi yang tidak bersisian pada sisi (v_1, v_2) tidak bersisian dengan simpul v_3.

c. Sisi Ganda (*Multi Edge*)

Dua buah rusuk atau lebih yang menghubungkan simpul-simpul yang sama.

Pada Gambar 2.11 sisi ganda ialah sisi $e_1 = (v_1, v_2)$ dan $e_2 = (v_1, v_2)$ yang menghubungkan dua simpul yang sama, yaitu v_1 dan v_2.

d. Sisi Gelang (*loop*)

Sebuah rusuk yang menghubungkan sebuah simpul dengan dirinya sendiri.

Pada Gambar 2.11 sisi gelang (*loop*) ialah sisi e_5 = (v_3, v_3).

e. Simpul Terpencil (*Isolated Vertex*)

Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian denganya. Atau, dapat juga dinyatakan simpul terpencil adalah simpul yang tidak satupun bertetangga dengan simpul-simpul yang lain. Pada Gambar 2.11 , simpul terpencil adalah simpul v_5 (Munir, 2010: 365).

f. Derajat (*Degree*)

Menurut Munir (2010: 365) derajat suatu simpul pada graf ialah banyaknya ujung sisi rusuk yang hadir pada suatu simpul. Dinotasikan dengan $d(v)$. Pada Gambar 2.1 $d(v_1) = d(v_2) = 3$, $d(v_3)$ = 5, $d(v_4)$ = 2, $d(v_5)$ = 0

g. Graf Bagian (*Subgraph*)

Sebuah graf H disebut graf bagian dari graf G, ditulis $H \subseteq G$, jika $V(H) \subseteq V(G)$ dan $E(H) \subseteq E(G)$. artinya semua simpul H termuat di G dan semua sisi di H termuat di G. (Munir, 2010: 372)

h. Graf Berbobot (*Weighted Graph*)

Menurut Munir (2010: 376) graf berbobot adalah graf yang setiap sisi, simpul, atau sisi dan simpul diberi sebuah harga (bobot)

2. Keterhubungan

Beberapa istilah yang berhubungan dengan keterhubungan sebuah graf, yaitu:

a. Jalan (*Walk*)

Sebuah jalan di graf G adalah sebuah barisan tak kosong dan berhingga $W = v_1, e_1, v_2, e_2, \dots, e_n, v_n$ yang suku-sukunya bergantian simpul dan rusuk.

b. Jejak (*Trail*)

Jalan W disebut jejak jika semua rusuk $e_1, e_2, \dots, e_k$ berbeda.

c. Lintasan (*Path*)

Jalan W disebut lintasan jika semua simpul dan rusuk dalam jalan W berbeda (Munir, 2010: 369).

d. Sirkuit (*Circuit*)

Lintasan yang berawal dan berakhir pada simpul yang sama (Munir, 2010: 370)

e. Sikel *(Cycle*)

Sebuah Jejak tertutup yang simpul awal dan simpul yang termuat didalamnya berbeda disebut sikel.

f. Terhubung (*Connected*)

Sebuah graf G disebut terhubung jika untuk setiap dua simpul u dan v di G terdapat lintasan di G yang menghubungkan kedua simpul tersebut. Jika tidak, maka G disebut graf tak-terhubung. Pada Gambar 2.11 merupakan graf tak-terhubung karena pada simpul v_5 tidak terdapat lintasan yang menghubungkan. (Munir, 2010: 371)

3. Jenis-Jenis Graf

Graf dapat dikelompokan menjadi beberapa kategori bergantung pada sudut pandang pengelompokanya. Menurut Munir (2010:357-358) berdasarkan ada tidaknya sisi ganda atau gelang pada suatu graf, secara umum graf dapat dikelompokan menjadi dua jenis

a. Graf sederhana (*simple graph*).

Graf yang tidak mempunyai sisi ganda ataupun gelang (*loop*).

b. Graf tak-sederhana

Graf yang mengandung sisi ganda ataupun gelang. Terdapat dua macam graf tak-sederhana, yakni graf semu (*pseudograph*) dan graf ganda (*multigraph*). Graf semu ialah graf yang mengandung gelang (*loop*) sedangkan graf ganda ialah graf yang didalamnya terdapat sisi ganda.

Ada beberapa graf sederhana khusus yang dijumpai pada banyak aplikasi. Beberapa di antaranya ialah:

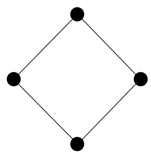
a. Graf Nol (*Null Graph*)

Graf yang himpunan sisinya merupakan himpunan kosong atau tidak mempunyai sisi hanya mempunyai simpul disebut sebagai graf kosong dan ditulis sebagai N_n, dengan n adalah banyak simpul (Munir, 2010: 366).

b. Graf Teratur (*Regular Graphs*)

Graf teratur ialah graf yang setiap simpulnya mempunyai derajat yang sama, apabila setiap simpul ialah *r*, maka graf tersebut disebut sebagai graf teratur berderajat *r*.

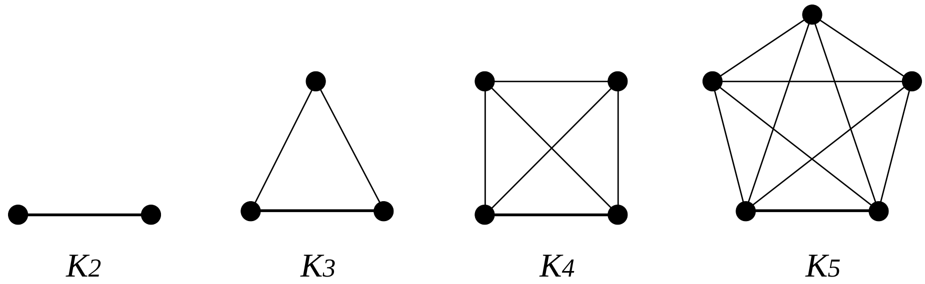
Banyak sisi pada graf teratur *r* dengan *n* buah simpul ialah $\frac{nr}{2}$ (Wibisono, 2008, 129).



Gambar 2.12 Graf Teratur Derajat 2

c. Graf Lengkap (*Complete Graph*)

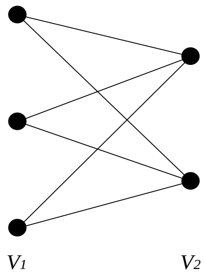
Graf lengkap ialah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainya. Graf lengkap dengan *n* buah simpul dilambangkan dengan K_n. Graf lengkap merupakan graf teratur dengan setiap simpul berderajat *n-1*. Banyak sisi pada graf lengkap yang terdiri dari *n* buah simpul ialah $\frac{n(n-1)}{2}$ (Munir, 2010: 377):



Gambar 2.13 Graf lengkap K_n, $2 \leq n \leq 5$

d. Graf Bipartet (*Bipartite Graph*)

Sebuah graf G disebut bipartet jika V(G) dapat dipartisi menjadi dua himpunan V_1 dan V_2 sedemikian sehingga setiap rusuk dari G menghubungkan sebuah simpul di V_1 dan sebuah simpul di V_2. (Wibisono, 2008, 129).



Gambar 2. 14 Graf Teratur Bipartit G(V_1,V_2)

4. Graf Kompatibel (Graf Terapan)

Graf kompatibel ialah suatu graf dengan himpunan simpul menyatakan objek yang diatur dan himpunan sisinya menunjukan pasangan objek yang mampu bekerja atau bergerak secara serasi. Menurut Hardianti et al (2013: 2) penerapan graf kompatibel salah satunya digunakan untuk mengatur arus lalu lintas pada suatu simpang jalan dengan simpul menyatakan arus lalu lintas dan sisi menyatakan pasangan arus yang dapat berjalan secara serasi tanpa menyebabkan konflik.

Menurut Yolanda et al (2014: 2) pemodelan graf kompatibel dalam sistem arus lalu lintas pada sebuah simpang jalan dengan langkah-langkah:

a. Menggambarkan sistem arus lalu lintas dari sebuah simpang jalan.
b. Menggambarkan graf kompatibel, dengan simpul melambangkan arus lalu lintas yang akan diatur, dan sisi menunjukan pasangan objek yang kompatibel. Dua buah simpul dihubungkan dengan sisi jika kedua arus saling kompatibel.
c. Menentukan *subgraph* lengkap terbesar.
d. Menentukan fase arus lalu lintas berdasarkan banyak *subgraph* lengkap.

d. Pelabelan *Fuzzy* pada Graf

Pelabelan *fuzzy* pada graf adalah suatu pemetaan yang memetakan unsur-unsur graf (simpul atau sisi) dengan himpunan bilangan *fuzzy*. Dalam hal ini pelabelan yang dilakukan ialah pelabelan *fuzzy*, maka himpunan bilangan yang digunakan ialah himpunan bilangan yang berada pada interval [0,1]. Ada banyak jenis pelabelan graf yang telah dikembangkan. Penelitian mengenai pelabelan graf terus berkembang baik dari bentuk pelabelanya atau dari graf yang dilabeli. Jika yang dilabeli hanya simpul maka disebut pelabelan simpul (*vertex labelling*). Begitu juga bila hanya sisi yang dilabeli, maka disebut pelabelan sisi (*edge labelling)*. Sedangkan jika suatu pelabelan melabeli simpul maupun sisinya, maka disebut pelabelan total (*total labelling)*

Berikut ini akan diberikan definisi untuk pelabelan total graf dengan label himpunan bilangan *fuzzy* :

Definisi 2.2. (Mordeson et al, 2001: 21) *Misal V adalah himpunan simpul tak kosong dan E adalah himpunan sisi pada graf G = (V, E). Pelabelan fuzzy adalah sepasang fungsi μ dan σ , dengan μ memetakan simpul di G ke* A⊆ [0,1] *dan σ memetakan sisi di G ke himpunan* B ⊆ [0,1], *serta memenuhi* $\sigma(u, v) \leq \min\{\mu(u),\ \mu(v)\}$, $\forall\, u, v \in V$.

Contoh 2.5 (Ratnasari, 2009: 24)

Misalkan diberikan graf *fuzzy G* dengan himpunan simpul $V = \{V_1, V_2, V_3, V_4\}$ dan himpunan sisi $E = \{(V_1, V_2), (V_1, V_3), (V_2, V_3), (V_3, V_4), (V_1, V_4)\}$. Derajat keanggotaan dari himpunan simpulnya ialah

$\mu(V_1) = 0{,}5$, $\mu(V_2) = 0{,}5$

$\mu(V_3) = 0{,}8$, $\mu(V_4) = 0{,}4$

Derajat keanggotaan himpunan sisinya adalah

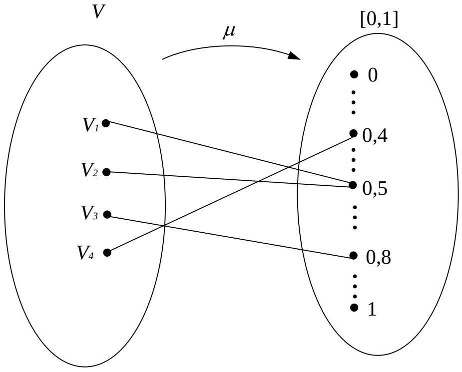
$\sigma\ (V_1, V_2) = 0{,}3$, $\sigma\ (V_2, V_3) = 0{,}5$, $\sigma\ (V_3, V_4) = 0{,}1$

$\sigma\ (V_1, V_3) = 0{,}2$, $\sigma\ (V_1, V_4) = 0{,}2$

1. Relasi μ dan σ memenuhi syarat fungsi sebagai berikut:

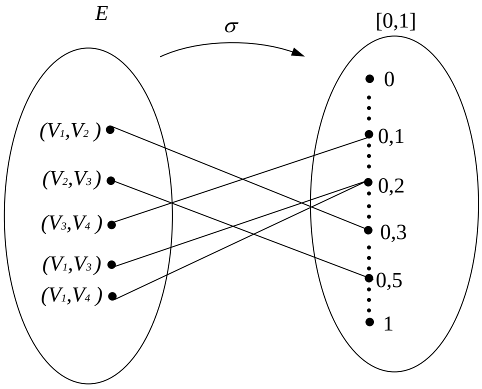
   Relasi dari A ke B dikatakan fungsi jika $v = w \rightarrow f(v) = f(w)$, $\forall v, w \in A$

   Relasi $\mu : V \rightarrow [0{,}1]$



Relasi μ mengawankan setiap himpunan simpul secara tunggal dengan himpunan derajat keanggotaan simpul maka relasi μ merupakan fungsi.

Relasi $\sigma : E \rightarrow [0,1]$



Relasi σ mengawankan setiap himpunan sisi secara tunggal dengan himpunan derajat keanggotaan sisi maka relasi σ merupakan fungsi.

2. Memenuhi $\sigma(u,v) \leq \min\{\mu(u),\ \mu(v)\}$, $\forall\ u,v \in V$

a. $\sigma\ (V_{1,}V_2)\ ) \leq \min\{\mu(V_1),\ \mu(V_2)\}$

$0{,}3\ \leq \min\{0{,}5\ , 0{,}5\}$

$0{,}3 \leq 0{,}5$

b. $\sigma\ (V_{2,}V_3)\ ) \leq \min\{\mu(V_2),\ \mu(V_3)\}$

$0{,}5\ \leq 0{,}5$

c. $\sigma\ (V_{3,}V_4)\ ) \leq \min\{\mu(V_3),\ \mu(V_4)\}$

$0{,}1\ \leq 0{,}4$

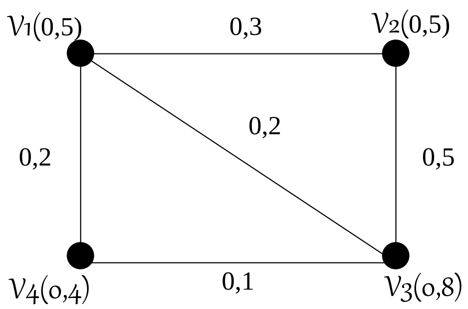
d. $\sigma\ (V_{1,}V_4)\ ) \leq \min\{\mu(V_1),\ \mu(V_4)\}$

$0{,}2\ \leq 0{,}4$s

e. $\sigma\ (V_{1,}V_3)\ ) \leq \min\{\mu(V_1),\ \mu(V_3)\}$

$0{,}2\ \leq 0{,}5$

Graf *fuzzy* G tersebut seperti Gambar 2.15 berikut:



Gambar 2.15 Graf Dengan Pelabelan *Fuzzy*

Sedangkan untuk *Subgraph fuzzy* didefinisikan sebagai berikut:

Definisi 2.3. (Nurshiami et al, 2014: 4) *Misal graf G = (μ, σ) adalah graf dengan pelabelan fuzzy. Graf H =* (τ, ρ) *disebut subgraph dengan pelabelan fuzzy dari G jika memenuhi* $\tau(u) \leq \mu(u)$*untuk setiap* $u \in V$ *dan* $\rho(u, v) \leq \sigma(u, v)$ *untuk setiap u,v* ∊ *V*

Contoh 2.6

Berdasar graf *fuzzy G* pada Contoh 2.5 dibuat *subgraph fuzzy* dengan himpunan simpul V = {V_1, V_2, V_3} dan himpunan sisi E = {(V_1, V_2), (V_2, V_3)}. Derajat keanggotaan dari himpunan simpulnya ialah

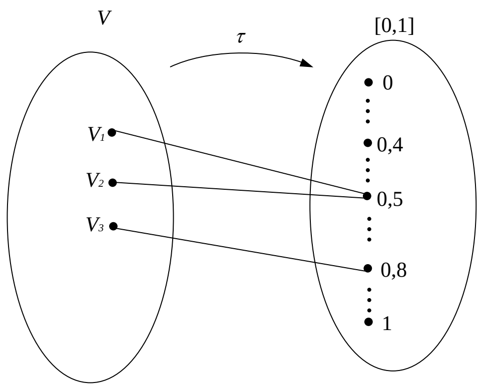
$\tau(V_1) = 0{,}5$; $\tau(V_2) = 0{,}5$; $\tau(V_3) = 0{,}8$

dan derajat keanggotaan himpunan sisinya adalah

ρ (V_1, V_2) = 0,3 ; ρ (V_2, V_3) = 0,5

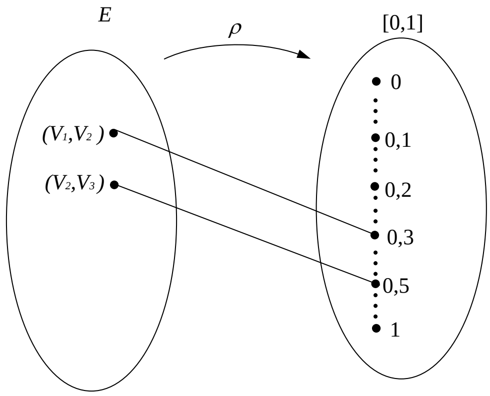
1. Relasi τ dan ρ memenuhi syarat fungsi sebagai berikut:

Relasi $\tau : V \rightarrow [0,1]$



Relasi τ mengawankan setiap himpunan simpul secara tunggal dengan himpunan derajat keanggotaan simpul maka relasi τ merupakan fungsi.

Relasi $\rho : E \rightarrow [0,1]$



Relasi ρ mengawankan setiap himpunan sisi secara tunggal dengan himpunan derajat keanggotaan sisi maka relasi ρ merupakan fungsi.

2. Memenuhi $\rho(u, v) \leq \min\{\tau(u),\ \tau(v)\}$, $\forall\ u, v \in V$

   a. $\rho\ (V_{1,} V_2)\ ) \leq \min\{\tau(V_1),\ \tau(V_2)\}$

      $0{,}3 \leq 0{,}5$

b. $\rho\ (V_{2,}V_3)\ ) \leq \min\{\tau(V_2),\ \tau(V_3)\}$

$0{,}5\ \leq 0{,}5$

3. Memenuhi $\tau(u) \leq \mu(u)$ untuk setiap $u \in V$

a. $\tau(V_1) \leq\ \mu(V_1)$

$0{,}5 \leq 0{,}5$

b. $\tau(V_2) \leq\ \mu(V_2)$

$0{,}5 \leq 0{,}5$

c. $\tau(V_3) \leq\ \mu(V_3)$

0,8 ≤ 0,8

4. Memenuhi $\rho(u, v) \leq \sigma(u, v)$ untuk setiap u,$v \in V$

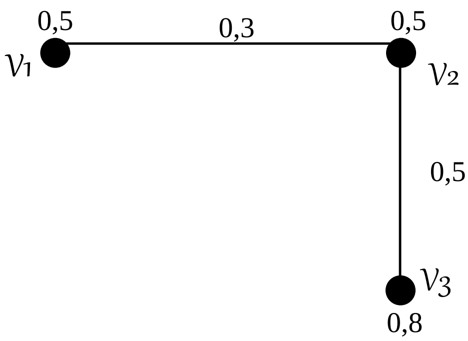
a. $\rho\ (V_1, V_2)\ \leq\ \sigma\ (V_1, V_2)$

$0{,}3 \leq 0{,}3$

b. $\rho\ (V_2, V_3) \leq\ \sigma\ (V_2, V_3)$

$0{,}5 \leq 0{,}5$

Subgraph fuzzy seperti pada Gambar 2.16 berikut



Gambar 2.16 *Subgraph Fuzzy*

e. Aljabar *Max-Plus*

1. Definisi Aljabar *Max-Plus*

Operasi dasar aljabar *max-plus* adalah maksimum yang direpresentasikan oleh $\oplus$ dan penjumplahan yang direpresentasikan oleh $\otimes$. Berikut diberikan definisi selengkapnya untuk aljabar *max-plus.*

Definisi 2.4. (Rudhito, 2016: 13) *Himpunan* $\mathbb{R} \cup \{-\infty\}$ *dengan* $\mathbb{R}$ *merupakan himpunan semua bilangan riil yang dilengkapi dengan operasi maksimum, dinotasikan dengan* $\oplus$ *dan operasi penjumlahan yang dinotasikan dengan* $\otimes$. *Diberikan* $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$ *dengan* $\varepsilon := -\infty$ *dan* $e := 0$.

Untuk setiap $a, b \in \mathbb{R}_{\max}$, *didefinisikan operasi* $\oplus$ *dan* $\otimes$ *sebagai berikut:*

$$a \oplus b := \max\{a, b\}$$
$$a \otimes b := a + b$$

2. Sifat-sifat Dasar Aljabar *Max-Plus*

Aljabar *max-plus* sudah banyak diterapkan dalam penyelesaian masalah-masalah di berbagai bidang seperti teori graf, teori antrian, dan teori sistem. Berdasarkan Definisi 2.4. $\max\{a, -\infty\} = \max\{-\infty, a\} = a$ dan $a + (-\infty) = (-\infty) + a = -\infty$, untuk suatu $a \in \mathbb{R}_{max}$, maka $a \oplus \varepsilon = a = \varepsilon \oplus a$ dan $a \otimes e = a = e \otimes a$ (Ariyanti, 2011: 33).

Contoh 2.7

Misal diberikan ε, e, 5, 10 $\in \mathbb{R}_{max}$, berikut merupakan contoh penerapan operasi $\oplus$ dan $\otimes$ pada aljabar *max-plus.*

$$10 \oplus 5 = \max\{10,5\} = 10$$

$$10 \oplus \varepsilon = \max\{10, -\infty\} = 10$$

$$10 \otimes \varepsilon = 10 + (-\infty) = \varepsilon$$

$$10 \oplus e = \max\{10,0\} = 10$$

$$10 \otimes 5 = 10 + 5 = 15$$

Ada analogi antara operasi $\oplus$ dan $\otimes$ dalam aljabar *max-plus* dengan operasi × dan + dalam aljabar linier (konvensional), yaitu operasi *plus* ($\otimes$) seperti × dalam aljabar linier mempunyai prioritas (didahulukan) daripada operasi *max* ($\oplus$) yang seperti + dalam aljabar linier.

Contoh 2.8

Misal diberikan -3, 2, 4, 5, 7, 10 ∈ $\mathbb{R}_{max}$

a. $10 \otimes 5 \oplus 7 \otimes 4$

   Cara penyelesaianya ialah sebagai berikut

   $10 \otimes 5 \oplus 7 \otimes 4 = (10 \otimes 5) \oplus (7 \otimes 4)$

   $= 15 \oplus 11 = \max\{15,11\} = 15$

b. $10 \otimes -3 \oplus 5 \otimes 2$

   Cara penyelesaianya ialah sebagai berikut

   $10 \otimes -3 \oplus 5 \otimes 2 = (10 \otimes -3) \oplus (5 \otimes 2)$

   $= 7 \oplus 7 = \max\{7,7\} = 7$

Operasi *max* ($\oplus$) dan *plus* ($\otimes$) juga mempunyai sifat aljabar, Menurut Subiono (2015: 3) sifat aljabar dari aljabar *max-plus* adalah $\forall\, x, y, z \in \mathbb{R}_{max}$ berlaku

a. Asosiatif

   $(x \oplus y) \oplus z = \max\{x, y\} \oplus z$

   $= \max\{max\,\{x, y\}, z\}$

   $= max\,\{x, y, z\} = \max\{x, max\,\{x, y\}\} = x \oplus (y \oplus z)$

dan

$x \otimes (y \otimes z) = x + (y + z) = x + y + z = (x + y) + z = (x \otimes y) \otimes z$

b. Komutatif

$x \oplus y = max\ \{x, y\} = max\ \{y, x\} = y \oplus x$

dan

$x \otimes y = x + y = y + x = y \otimes x$

c. Distributif

$$
\begin{aligned}
x \otimes (y \oplus z) &= x \otimes \max\{y, z\} \\
&= x + \max\{y, z\} \\
&= max\{x + y, x + z\} = (x \otimes y) \oplus (x \otimes z)
\end{aligned}
$$

d. Adanya elemen nol

$x \oplus \varepsilon = \max\{\mathrm{x}, -\infty\} = \max\{-\infty, x\} = \varepsilon \oplus x = x$

e. Adanya e = elemen satuan (*unit*) terhadap $\otimes$

$x \otimes e = x + 0 = x$

f. Adanya sifat penyerapan oleh ε = elemen nol terhadap $\otimes$

$x \otimes \varepsilon = x + (-\infty) = \varepsilon$

g. Sifat idempotent dari $\oplus$

$x \oplus x = \max\{x, x\} = x$

Menurut Subiono (2015: 5) pangkat dalam aljabar *max-plus* diperkenalkan dengan menggunakan sifat assosiatif. Untuk $\mathbb{N}$ ialah himpunan bilangan asli, didefinisikan untuk $x \in \mathbb{R}_{max}$ sebagai berikut:

$$x^{\otimes n} \stackrel{\text{def}}{=} \underbrace{x \otimes x \otimes \ldots \otimes x}_{n}, \qquad \forall n \in \mathbb{N}\ dengan\ n \neq 0$$

$$x^{\otimes 0} \stackrel{\text{def}}{=} e = 0 \quad , \qquad untuk\ n = 0$$

Jika dianalogikan dengan aljabar linier, tampak bahwa untuk setiap $n \in \mathbb{N}$,

$$x^{\otimes n} \stackrel{\text{def}}{=} \underbrace{x \otimes x \otimes \ldots \otimes x}_{n} \stackrel{\text{def}}{=} \underbrace{x + x + \cdots + x}_{n} = n \times x$$

Contoh 2.9

a. Misal akan dicari $10^{\otimes 3}$, dengan 10 ∈ $\mathbb{R}_{max}$ dan 3 ϵ $\mathbb{N}$

cara penyelesaianya ialah sebagai berikut

$$10^{\otimes 3} = 10 + 10 + 10 = 3 \times 10 = 30$$

b. Misal akan dicari $5^{\otimes 2}$, dengan 5 ∈ $\mathbb{R}_{max}$ dan 2 ϵ $\mathbb{N}$

cara penyelesaianya ialah sebagai berikut

$$5^{\otimes 2} = 5 + 5 = 2 \times 5 = 10$$

Pangkat aljabar *max-plus* mempunyai prioritas tertinggi dibandingkan operasi ⊕ dan ⊗ dalam hal urutan pengoperasian.

3. SemiRing

Suatu semi Ring (**S**, +,×) ialah himpunan tak kosong **S** disertai dengan dua operasi biner + dan ×, yang memenuhi aksioma berikut (Rudhito, 2004: 132-133):

a. (**S**, +) adalah struktur aljabar dengan satu operasi biner (semigrup) yang komutatif dengan elemen netral 0, yakni $\forall a, b, c \in \boldsymbol{S}$ memenuhi

1) komutatif, yakni $a + b = b + a$

2) assosiatif, yakni $(a + b) + c = a + (b + c)$

3) elemen netral, yakni $a + 0 = 0 + a = a$

b. (**S**, $\times$) adalah struktur aljabar dengan satu operasi biner (semigrup) dengan elemen satuan 1, yakni $\forall a, b, c \in \boldsymbol{S}$ memenuhi

1) Assosiatif terhadap perkalian, yakni $(a \times b) \times c = a \times (b \times c)$

2) Elemen satuan 1, yakni $a \times 1 = 1 \times a = a$

c. Sifat penyerapan elemen netral 0 terhadap operasi $\times$, yakni $\forall a \in \boldsymbol{S}$ memenuhi

1) $a \times 0 = 0 \times a = 0$

d. Operasi $\times$ distributif terhadap +, yakni $\forall a, b, c \in \boldsymbol{S}$ memenuhi

1) $(a + b) \times c = (a \times c) + (b \times c)$

2) $a \times (b + c) = (a \times b) + (a \times c)$

Suatu semi Ring (**S**, +,$\times$) dikatakan komutatif jika operasi $\times$ bersifat komutatif, yakni $\forall a, b \in \boldsymbol{S}$ berlaku $a \times b = b \times a$

Kemudian suatu semi Ring (**S**, +,$\times$) mempunyai sifat idempoten terhadap operasi + berlaku $a \times a = a,\ \forall a \in \boldsymbol{S}$

Aljabar *max-plus* merupakan semi Ring komutatif dan idempoten. Karena untuk setiap $\forall\ x, y, z\ \in\ \mathbb{R}_{max}$ berlaku:

a. $\oplus$ ialah asosiatif dan komutatif dengan elemen nol yakni ε:

1) $(x \oplus y) \oplus z$ $= \max\{\mathrm{x}, \mathrm{y}\} \oplus z$

$=\max\{\max\{\mathrm{x}, \mathrm{y}\}, z\}$

$= \max\{\mathrm{x}, \mathrm{y}, \mathrm{z}\} = \max\{x, \max\{\mathrm{y}, \mathrm{z}\}\} = x \oplus (y \oplus z)$

2) $x \oplus y = \max\{\mathrm{x}, \mathrm{y}\} = \max\{\mathrm{y}, \mathrm{x}\} = y \oplus x$

3) $x \oplus \varepsilon = \max\{x, -\infty\} = x$

b. $\otimes$ ialah asosiatif dan distributif terhadap $\oplus$ dan mempunyai elemen satuan yakni e:

1) $x \otimes (y \otimes z)$=$x + (y + z)$

$$= x + y + z = (x + y) + z = (x \otimes y) \otimes z$$

2) $(x \oplus y) \otimes z \ = \ \max\{\mathrm{x}, \mathrm{y}\} + z$

$$= \max\{x + z, y + z\} = (x \otimes z) \oplus (y \otimes z)$$

3) $x \otimes (y \oplus z) \ = \ x + \max\{\mathrm{y}, \mathrm{z}\}$

$$= \max\{x + y, x + z\} = (x \otimes y) \oplus (x \otimes z)$$

4) $x \otimes e = x + 0 = 0 + x = e \otimes x = x$

c. ε merupakan elemen penyerapan terhadap $\otimes$

1) $x \otimes \varepsilon = \ x + (-\infty) = (-\infty) + x = \varepsilon \ \oplus x = \varepsilon$

d. Komutatif dan indempoten, yakni

1) $x \otimes y = x + y = y + x = y \otimes x$

2) $x \oplus x = \max\{x, x\} = x$

4. Matriks atas Aljabar *Max-Plus*

Operasi $\oplus$ dan $\otimes$ dalam aljabar *max-plus* dapat diperluas untuk operasi matriks atas aljabar *max-plus* dengan himpunan matriks ukuran $n \times m$ dalam aljabar *max-plus* dinotasikan dengan $\mathbb{R}_{max}^{n\times m}$, untuk $n, m \in \mathbb{N}$ dengan $n, m \neq 0$ didefinisikan $\underline{n} \overset{\text{def}}{=} \{1,2,3,...,n\}$ dan $\underline{m} \overset{\text{def}}{=} \{1,2,3,...,m\}$. Elemen baris ke-$i$ kolom ke-j dari $A \in \mathbb{R}_{max}^{n\times m}$ dinotasikan oleh $a_{i,j}$ untuk $i \in \underline{n}$ dan $j \in \underline{m}$ (Ariyanti,2011: 43). Kemudian menurut Subiono (2015: 8) matriks A ditulis sebagai:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,m} \end{bmatrix} \quad , \quad \begin{matrix} n\ baris \\ m\ kolom \end{matrix}$$

Elemen $a_{i,j}$ juga dinotasikan sebagai $[A]_{i,j}, i \in \underline{n}, j \in \underline{m}$

Menurut Subiono (2015: 8-10) operasi yang melibatkan matriks dalam aljabar *max-plus* ialah sebagai berikut:

a. Penjumlahan matriks

Operasi penjumlahan matriks $A, B \epsilon \mathbb{R}_{max}^{n\times m}$ dinotasikan oleh $A \oplus B$ didefinisikan sebagai berikut

$$[A\oplus B]_{i,j} = a_{i,j} \oplus b_{i,j} = \max\{a_{i,j}, b_{i,j}\}\ , untuk\ , i \in \underline{n}\ dan\ j \in \underline{m}$$

Contoh 2.10

Misal diberi matriks A, B $\epsilon \mathbb{R}_{max}^{2\times 2}$ sebagai berikut

$A = \begin{bmatrix} e & 3 \\ 5 & 7 \end{bmatrix}$dan $B = \begin{bmatrix} 5 & -3 \\ 10 & \varepsilon \end{bmatrix}$, akan dicari $A\oplus B$

Maka

$$[A\oplus B]_{1,1} = e \oplus 5 = \max\{e, 5\} = 5$$

$$[A\oplus B]_{1,2} = 3 \oplus -3 = \max\{3, -3\} = 3$$

$$[A\oplus B]_{2,1} = 5 \oplus 10 = \max\{5,10\} = 10$$

$$[A\oplus B]_{2,2} = 7 \oplus \varepsilon = \max\{7, -\infty\} = 7$$

Jadi, $A\oplus B$ = $\begin{bmatrix} 5 & 3 \\ 10 & 7 \end{bmatrix}$

Dalam operasi penjumlahan di aljabar *max-plus* berlaku sifat komutatif, yakni $A\oplus B = B \oplus A$ karena

$$[A\oplus B]_{i,j} = \max\{a_{i,j}, b_{i,j}\} = \max\{b_{i,j}, a_{i,j}\} = [B\oplus A]_{i,j}, \forall\ i \in \underline{n}\ dan\ j \in \underline{m}$$

b. Perkalian Skalar dan Matriks

Untuk $A \epsilon \mathbb{R}_{max}^{n\times m}$, dan $a \epsilon \mathbb{R}_{max}$ perkalian skalar $a\otimes A$ didenisikan sebagai berikut

$$[a\otimes A]_{i,j} \overset{\text{def}}{=} a \otimes a_{i,j} \ , untuk \ , i \in \underline{n} \ dan \ j \in \underline{m}$$

Contoh 2.11

Misal diberikan konstanta a $\epsilon \ \mathbb{R}_{max}$ dan matriks A $\epsilon \ \mathbb{R}_{max}^{2\times 2}$ sebagai berikut

$a = 5$ dan $A = \begin{bmatrix} e & 10 \\ 5 & \varepsilon \end{bmatrix}$, akan dicari $a\oplus A$

$$[a\otimes A]_{1,1} = 5 \otimes e \ = \ 5 + e \ = 5 + 0 = 5$$

$$[a\otimes A]_{1,2} = 5 \otimes 10 \ = \ 5 + 10 \ = 5 + 10 = 15$$

$$[a\otimes A]_{2,1} = 5 \otimes 5 \ = \ 5 + 5 \ = 5 + 5 = 10$$

$$[a\otimes A]_{2,2} = 5 \otimes \varepsilon \ = \ 5 + \ \varepsilon = 5 + (-\infty) = \varepsilon$$

Jadi, $a\otimes A = \begin{bmatrix} 5 & 15 \\ 10 & \varepsilon \end{bmatrix}$

c. Perkalian antar Matriks

Untuk matriks $A \epsilon \mathbb{R}_{max}^{n\times p}$ dan $B \epsilon \mathbb{R}_{max}^{p\times m}$, perkalian matriks $A\otimes B$ didefinisikan sebagai berikut:

$$[A\otimes B]_{i,j} = \bigoplus_{k=1}^{p} a_{i,k}\otimes b_{k,j} = \max_{k\epsilon \underline{p}}\{a_{i,k} + b_{k,j}\}, untuk \ , i \in \underline{n} \ dan \ j \in \underline{m}$$

Contoh 2.12

Misal diberikan matriks A, B $\epsilon \ \mathbb{R}_{max}^{2\times 2}$ sebagai berikut

$A = \begin{bmatrix} e & 10 \\ 5 & \varepsilon \end{bmatrix}$ dan $B = \begin{bmatrix} 1 & -4 \\ 5 & 2 \end{bmatrix}$, akan dicari $A \otimes B$

$[A \otimes B]_{1,1} = (e \otimes 1) \oplus (10 \otimes 5) = \max\{e + 1, 10 + 5\}$

$= \max\{0 + 1, 10 + 5\} = 15$

$[A \otimes B]_{1,2} = (e \otimes (-4)) \oplus (10 \otimes 2) = \max\{e + (-4), 10 + 2\}$

$= \max\{0 + (-4), 10 + 2\} =$

12

$[A \otimes B]_{2,1} = (5 \otimes 1) \oplus (\varepsilon \otimes 5) = \max\{5 + 1, \varepsilon + 5\}$

$= \max\{5 + 1, (-\infty) + 5\} = 6$

$[A \otimes B]_{2,2} = (5 \otimes (-4)) \oplus (\varepsilon \otimes 2) = \max\{5 + (-4), \varepsilon + 2\} =$

$= \max\{5 + (-4), (-\infty) + 2\} =$

1

Jadi, $A \otimes B = \begin{bmatrix} 15 & 12 \\ 6 & 1 \end{bmatrix}$

Perkalian matriks ini serupa dengan perkalian matriks aljabar linier dengan + diganti dengan max dan × diganti dengan +. Dalam aljabar linier perkalian matriks bersifat tidak komutatif sama halnya operasi perkalian antar matriks dalam aljabar *max-plus* juga tidak bersifat komutatif (Nurwan, 2013: 367)

5. Nilai Eigen dan Vektor Eigen dalam Aljabar Max-Plus

Menurut Subiono (2015: 32) pengertian nilai eigen dan vektor eigen yang bersesuaian dari suatu matriks persegi A berukuran $n \times n$ sebagaimana dijumpai dalam aljabar linear, juga dijumpai dalam aljabar *max-plus*, yakni bila diberikan suatu persamaan sebagai berikut:

$$A \otimes \bar{x} = \lambda \otimes \bar{x}\,, \qquad \forall\, \bar{x}$$

dalam hal ini masing-masing vektor $x \in \mathbb{R}^n_{max}$ merupakan vektor eigen dan $\lambda \in \mathbb{R}$ merupakan nilai eigen dari matriks A dengan vektor $\bar{x} \neq (\varepsilon, \varepsilon, \dots, \varepsilon)^T$. Suatu algoritma untuk menentukan nilai eigen dan vektor eigen dari matriks $A \in \mathbb{R}^{n \times n}_{max}$ dilakukan secara berulang dari bentuk persamaan linear

$$\bar{x}(k+1) = A \otimes \bar{x}(k), k = 0,1,2, \dots$$

Relasi berulang dari persamaan tersebut untuk matriks A yang taktereduksi maupun yang tereduksi erat kaitanya dengan vektor waktu sikel yang didefinisikan sebagai berikut:

$$\lim_{k \to \infty} \frac{\bar{x}(k)}{k}$$

Limit ini ada untuk setiap keadaan awal $\bar{x} \neq (\varepsilon, \varepsilon, \dots, \varepsilon)^T$ seperti pada teorema yang menjelaskan bentuk nilai eigen dari matriks persegi A berikut (Subiono, 2015: 40):

Bila untuk sebarang keadaan awal $\bar{x}(0) \neq \varepsilon$ sistem persamaan $\bar{x}(k+1) = A \otimes \bar{x}(k)$ memenuhi $\bar{x}(p) = c \otimes \bar{x}(q)$ untuk beberapa bilangan bulat p dan q dengan $p > q \geq 0$ dan beberapa bilangan *real* c dan $\bar{x}(q) \in \mathbb{R}^n$

maka $\lim_{k \to \infty} \frac{\bar{x}(k)}{k} = \begin{bmatrix} \lambda \\ \lambda \\ \vdots \\ \lambda \end{bmatrix}$

Dengan $\lambda = \frac{c}{p-q}$. Selanjutnya λ ialah suatu nilai eigen dari matriks A dengan vektor karakteristik diberikan oleh

$$\bar{v} = \bigoplus_{i=1}^{p-q} (\lambda^{\otimes(p-q-i)} \otimes x(q+i-1)$$

Berdasarkan hasil teorema tersebut, menurut Subiono (2015: 42) algoritma power dapat digunakan untuk menentukan nilai eigen dan vektor eigen dari suatu matriks persegi *A*. Algoritma power pertama kali dikenalkan oleh Olsder dan diberikan contohnya tetapi pada saat itu belum ada teori pengembanganya. Pengembangan teori algoritma power berturut-turut dilakukan oleh Braker dan Subiono. Algoritma power ialah sebagai berikut:

a. Mulai dari sebarang vektor awal $\bar{x}(0) \neq \varepsilon$
b. Iterasi persamaan $\bar{x}(k+1) = A \otimes \bar{x}(k)$ hingga ada bilangan bulat $p > q \geq 0$ dan bilangan real *c* sehingga suatu perilaku periodik terjadi, yakni $\bar{x}(p) = c \otimes \bar{x}(q)$
c. Hitung nilai eigen $\lambda = \frac{c}{p-q}$
d. Hitung vektor eigen $\bar{v} = \bigoplus_{i=1}^{p-q} (\lambda^{\otimes(p-q-i)} \otimes \bar{x}(q+i-1)$

Contoh 2. 13

Diberikan matriks $A=\begin{bmatrix} 3 & 2 & 1 \\ 0 & 1 & 4 \\ 0 & 2 & 1 \end{bmatrix}$ akan dicari nilai eigen dan vektor eigen

Keadaan awal

$\bar{x}$ (0)=$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$, *k*=0,1,2,...

selanjutnya, iterasi $\bar{x}$ (*k*+1)=*A* ⊗ $\bar{x}$ (*k*), *k*=0,1,2,...

Iterasi pertama, *k* = 0 → *x*(0+1)=*A* ⊗ *x*(0)

$$\bar{x}\,(1)=\begin{pmatrix}3 & 2 & 1\\ 0 & 1 & 4\\ 0 & 2 & 1\end{pmatrix}\otimes\begin{pmatrix}0\\0\\0\end{pmatrix}=\begin{pmatrix}\max\{3,2,1\}\}\\ \max\{0,1,4\}\\ \max\{0,2,1\}\end{pmatrix}=\begin{pmatrix}3\\4\\2\end{pmatrix}$$

Iterasi Kedua, $k = 1 \rightarrow \bar{x}\,(1+1)=A \otimes \bar{x}\,(1)$

$$\bar{x}\,(2)=\begin{pmatrix}3 & 2 & 1\\ 0 & 1 & 4\\ 0 & 2 & 1\end{pmatrix}\otimes\begin{pmatrix}3\\4\\2\end{pmatrix}=\begin{pmatrix}6\\6\\6\end{pmatrix}$$

Iterasi Ketiga, $k = 2 \rightarrow \bar{x}\,(2+1)=A \otimes \bar{x}\,(2)$

$$\bar{x}\,(3)=\begin{pmatrix}3 & 2 & 1\\ 0 & 1 & 4\\ 0 & 2 & 1\end{pmatrix}\otimes\begin{pmatrix}6\\6\\6\end{pmatrix}=\begin{pmatrix}9\\10\\8\end{pmatrix}$$

Iterasi Keempat, $k = 3 \rightarrow x(3+1)=A \otimes x(3)$

$$\bar{x}\,(4)=\begin{pmatrix}3 & 2 & 1\\ 0 & 1 & 4\\ 0 & 2 & 1\end{pmatrix}\otimes\begin{pmatrix}9\\10\\8\end{pmatrix}=\begin{pmatrix}12\\12\\12\end{pmatrix}$$

didapat,

$$\bar{x}\,(0)=\begin{pmatrix}0\\0\\0\end{pmatrix},\ \bar{x}\,(1)=\begin{pmatrix}3\\4\\2\end{pmatrix},\ \bar{x}\,(2)=\begin{pmatrix}6\\6\\6\end{pmatrix},\ \bar{x}\,(3)=\begin{pmatrix}9\\10\\8\end{pmatrix},\ \bar{x}\,(4)=\begin{pmatrix}12\\12\\12\end{pmatrix},\ \ldots$$

Terlihat bahwa suatu periodesisasi terjadi, $\bar{x}\,(2)=6\otimes\bar{x}\,(0)$. Didapatkan nilai c=6, p=2, dan q=0. Nilai eigen dari matriks A diperoleh:

$$\lambda = \frac{c}{p-q} = \frac{6}{2-0} = 3$$

Dengan vektor eigen

$$\bar{v} = \bigoplus_{i=1}^{p-q} (\lambda^{\otimes(p-q-i)} \otimes \bar{x}(q+i-1)$$

$$\bar{v} = \bigoplus_{i=1}^{2-0} (3^{\otimes(2-0-i)} \otimes \bar{x}(0+i-1)$$

$$\bar{v} = \bigoplus_{i=1}^{2} (3^{\otimes(2-i)} \otimes \bar{x}(i-1)$$

$$\bar{v} = (3^{\otimes(1)} \otimes \bar{x}(0)) \oplus ((3^{\otimes(0)} \otimes \bar{x}(1))$$

$$\bar{v} = \max\left(3^{\otimes(1)} \otimes \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, 3^{\otimes(0)} \otimes \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}\right)$$

$$v = \max\left(\begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}\right) = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$$

Selanjutnya dicek hasilnya

$$A \otimes \bar{v} \ = \ \lambda \otimes \bar{v}$$

$$\begin{pmatrix} 3 & 2 & 1 \\ 0 & 1 & 4 \\ 0 & 2 & 1 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix} = 3 \otimes \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$$

$$\begin{pmatrix} 6 \\ 7 \\ 6 \end{pmatrix} = \begin{pmatrix} 6 \\ 7 \\ 6 \end{pmatrix} \ ,$$

Jadi didapatkan nilai eigen $\lambda = 3$ dan vektor eigen $\bar{v} = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$

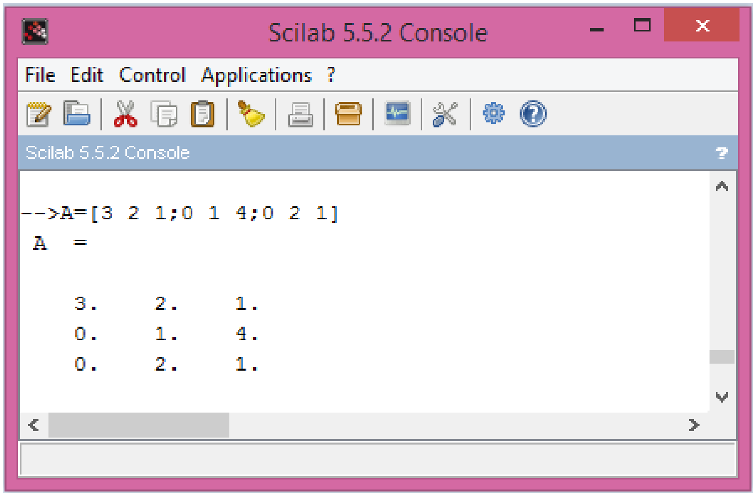
6. Mencari Nilai Eigein dan Vektor Eigen Menggunakan Software Scilab

Software scilab digunakan untuk mempermudah pencarian vektor eigen dan nilai eigen. Scilab merupakan perangkat lunak untuk komputasi numerik dan visualisasi data. Alamat website scilab ialah www.scilab.org, merupakan program *freeware* atau dapat digunakan secara gratis. Kemudian digunakan *toolbox-scilab* khusus untuk aljabar *max-plus* yang didapat dari internet pada alamat www.atoms.scilab.org/toolboxes/maxplus_petrinet /1.1.0

Untuk menjalankan *toolbox* khusus aljabar *max-plus* yakni buka file **Builder** dan **Loader.** Langkahnya File – Execute – Builder – Open kemudian Langkah yang sama untuk file **Loader,** File – Execute – Loader – Open.

Contoh 2.14

Menggunakan matriks pada Contoh 2.12 akan digunakan program scilab dalam perhitungan matriks untuk mencari nilai eigen dan vektor eigen. Berikut pada Gambar 2.17 tampilan perhitungan dalam program scilab:



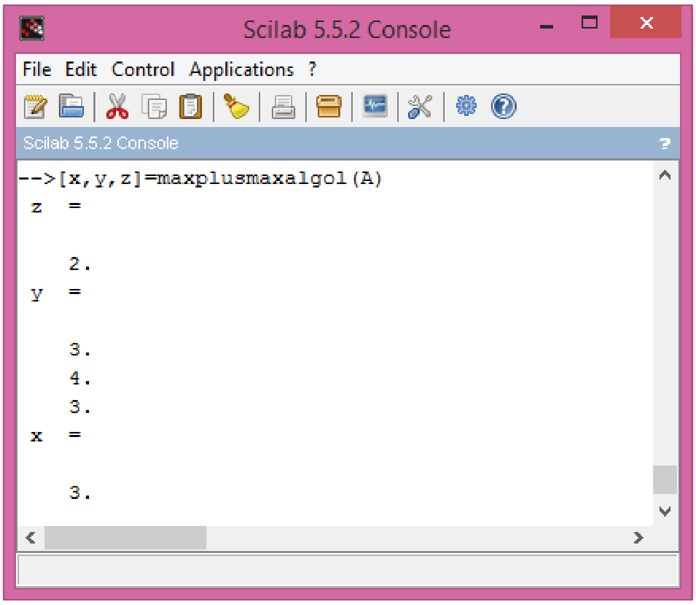
Gambar 2.17 Tampilan Perintah Matriks Dalam Software Scilab

Kemudian perintah yang digunakan untuk mendapatkan nilai eigen dan vektor eigen ialah sebagai berikut:

[x,y,z] = maxplusmaxalgol(A)

dengan x merupakan nilai eigen, y merupakan vektor eigen, dan z merupakan banyak fase dalam satu siklus jika dalam perhitungan manual

$z=p\text{-}q$. Pada Gambar 2.18 berikut merupakan perintah dan hasil dalam program scilab:



Gambar 2.18 Tampilan Hasil Dalam Software Scilab

Dari Gambar 2.18 didapat:

$z = p\text{-}q = 2$,

nilai eigen (λ) = x = 3

vektor eigen(v) = y = $\begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$

Terlihat untuk perhitungan manual dan perhitungan menggunakan program scilab dengan *toolbox-scilab* khusus untuk aljabar *max-plus* didapat hasil nilai eigen dan vektor eigen yang sama. Sehingga perhitungan dengan software scilab cukup efektif untuk digunakan dalam perhitungan nilai eigen dan vektor eigen pada penelitian ini yang menggunakan matriks berukuran lebih besar.