PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK
MIXED-STRATEGY TWO-PERSON ZERO-SUM GAME
DENGAN
METODE SIMPLEKS

Disusun oleh :

DANI HERNIWAN

NRP. 2691.100.040

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

1996

PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK
MIXED-STRATEGY TWO-PERSON ZERO-SUM GAME
DENGAN
METODE SIMPLEKS

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
S u r a b a y a

Mengetahui / Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
(Dr. Ir. SUPENO DJANALI)	(Ir. ESTHER HANAYA, MSc.)
NIP. 130 368 610	NIP. 130 816 212

S U R A B A Y A
JULI, 1996

TUGAS AKHIR

ABSTRAK

ABSTRAK

Mixed-Strategy Two-Person Zero-Sum Game adalah permainan yang dimainkan oleh dua pemain, dengan jumlah keuntungan dan kerugian yang diterima oleh kedua pemain tersebut berjumlah nol dengan posisi pilihan terbaik bagi setiap pemain dicapai dengan melakukan pencampuran terhadap strategi-strategi yang berbeda.

Proses pengambilan keputusan pada dua pemain yang bersaing dimana keduanya menggunakan strategi yang banyak (lebih dari dua) dapat dipandang sebagai persoalan Programa Linier. Satu strategi dipandang sebagai satu variabel keputusan. Pembayaran dari setiap strategi dipandang sebagai koefisien variabel keputusan.

Metode yang digunakan disini adalah Metode Simpleks, yang dapat diuraikan sebagai berikut : Langkah pertama (inisialisasi), menetapkan titik ekstrem awal (solusi awal) bagi persoalan programa linier yang sedang dihadapi. Langkah iterasi , menguji optimalitas dari suatu titik ekstrem dan bergerak ke titik ekstrem lainnya yang berdekatan apabila titik ekstrem tersebut belum optimal. Langkah penghentian, memberhentikan langkah iterasi apabila telah sampai pada titik ekstrem yang terbaik (optimum).

TUGAS AKHIR

KATA PENGANTAR

KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadirat Allah SWT, karena berkat dan rahmat-Nya-lah akhirnya Tugas Akhir yang merupakan kewajiban bagi setiap mahasiswa dalam menempuh program sarjananya, di Jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya, dapat penulis selesaikan tepat pada waktunya.

Tugas Akhir yang berjudul **"Perancangan Dan Pembuatan Mixed-Strategy Two-Person Zero-Sum Game Dengan Metode Simpleks"** adalah untuk menyelesaikan masalah Teori Permainan.

Dengan terwujudnya Tugas Akhir ini, tak lupa penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr.Ir. Supeno Djanali, selaku dosen pembimbing yang telah memberikan banyak bimbingan kepada penulis dalam menyelesaikan Tugas Akhir ini.
2. Ibu Ir. Esther Hanaya, MSc, selaku dosen pembimbing dan dosen wali yang telah memberikan banyak bimbingan kepada penulis dalam menyelesaikan Tugas Akhir ini dan dalam kegiatan perkuliahan.
3. Ibu Dr.Ir. Handayani Tjandrasa, selaku ketua jurusan Teknik Informatika.
4. Staf dosen di jurusan Teknik Informatika.

5. Putu Gede, Wayan, Kadek, Paulus, Sri Bayu, Widodo, Probo, Arga, Mujib, Hardana, Baju yang banyak memberikan bantuan dalam menyelesaikan Tugas Akhir ini.
6. Segenap pengurus Lab Komisi serta semua pihak yang banyak memberikan dorongan kepada penulis.

Akhirnya besar harapan penulis semoga Tugas Akhir ini bermanfaat bagi semua pihak.

Surabaya, Juli 1996

Penulis

TUGAS AKHIR

DAFTAR ISI

DAFTAR ISI

TUGAS AKHIR

DAFTAR TABEL

DAFTAR TABEL

Halaman

TUGAS AKHIR

DAFTAR GAMBAR

DAFTAR GAMBAR

Halaman

TUGAS AKHIR

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Persaingan antar perusahaan tidak dapat dihindarkan, karena ini menyangkut kemampuan untuk bertahan dan berkembang. Usaha untuk memenangkan persaingan ini dinyatakan dalam bentuk strategi-strategi perusahaan. Pemilihan strategi yang akan dipergunakan memerlukan banyak pertimbangan, antara lain adalah jumlah keuntungan yang akan diperoleh, besarnya resiko yang akan didapat, jenis strategi yang dipergunakan lawan dan sebagainya. Bila masing-masing pihak dapat mengetahui strategi yang dipergunakan lawan, maka strategi yang dipilih dapat dioptimasi untuk memperoleh hasil yang maksimal, yaitu hasil yang memberikan keuntungan yang paling optimum dan menimbulkan resiko yang paling kecil.

Proses pengambilan keputusan antara dua aktivitas yang bersaing, dalam Teori Optimasi, dapat dipandang sebagai "permainan", sedangkan kedua aktivitas tersebut dipandang sebagai "pemain".

Jika terdapat 2 pemain dengan jumah kerugian dan keuntungan dari permainan ini nol, maka permainan ini disebut **Permainan Dua Pemain Dengan Jumlah Nol** (*Two-Person Zero-Sum Game).*

Ada dua jenis persoalan Permainan Dua Pemain Dengan Jumlah Nol yang biasa ditemui. Jenis pertama adalah permainan yang posisi pilihan terbaik bagi setiap pemain dapat dicapai dengan memilih satu strategi tunggal , yang biasa disebut dengan **Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Tunggal** (*Pure-Strategy Two-Person Zero-Sum Game*). Jenis yang kedua adalah permainan yang kedua pemainnya melakukan pencampuran terhadap strategi-strategi yang berbeda dengan maksud untuk mencapai posisi pilihan terbaik. Hal ini disebabkan nilai maksimin dan nilai minimaks yang diperoleh sama, sehingga penyelesaiannya harus menggunakan prosedur tersendiri. Jenis permainan ini biasa disebut dengan **Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran** (*Mixed-Strategy Two-Person Zero-Sum Game*).

Apabila strategi-strategi yang dipakai dalam Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran oleh setiap pemain lebih dari dua, akan sulit untuk dipecahkan dengan aljabar biasa, tetapi akan lebih mudah jika strategi-strategi tersebut dipandang sebagai persoalan programa linier.

Setelah model matematika dari programa linier diketahui, maka selanjutnya harus ditentukan prosedur yang paling efisien untuk mendapatkan penyelesaian dari programa linier tersebut. Untuk itu telah tersedia prosedur penyelesaian yang sangat efisien, yang dinamakan **Metode Simpleks**.

1.2 PERMASALAHAN

Dalam Tugas Akhir ini akan dibuat suatu perangkat lunak untuk menyelesaikan persoalan Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran. Strategi-strategi yang digunakan oleh kedua pemain dalam Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran bersifat saling bergantungan karena adanya pembatas berupa nilai maksimin dan minimaks yang tidak boleh sama. Selain daripada itu, penyelesaian persoalannya tidak mudah untuk ditentukan. Untuk itu diperlukan algoritma yang baik untuk mendapatkan penyelesaian yang optimum dari Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran. Prosedur penyelesaian yang digunakan dalam Tugas Akhir ini adalah dengan menggunakan Metode Simpleks.

1.3 TUJUAN

Merancang dan membuat perangkat lunak yang dapat menyelesaikan persoalan Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran dengan menggunakan Metode Simpleks.

Hasil penelitian ini diharapkan dapat membantu memecahkan proses pengambilan keputusan yang tebaik bagi dua aktivitas yang bersaing, dimana kedua aktivitas tersebut menggunakan sumber yang sama.

1.4 METODOLOGI

Dalam merancang dan membuat perangkat lunak untuk menyelesaikan masalah Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran dilakukan beberapa langkah sebagai berikut :

- Studi literatur, mempelajari beberapa kepustakaan yang membahas tentang Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran dengan menggunakan Metode Simpleks.
- Perancangan dan pembuatan perangkat lunak, secara garis besar tahap awal dari pembuatan perangkat lunak adalah membuat algoritma Metode Simpleks berdasarkan teori yang telah dipelajari dan algoritma Nilai Permainan. Dan selanjutnya algoritma-algoritma ini dibuat aplikasinya dengan menggunakan compiler Borland C++ 4.5 for Windows.

1.5 RELEVANSI

Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran dipergunakan untuk mendapatkan strategi optimum dari dua aktivitas yang bersaing dalam menggunakan sumber yang sama, seperti bagaimana memilih strategi penjualan bagi dua perusahaan yang menjual produk yang sama agar mendapatkan keuntungan yang optimum bagi kedua perusahaan tersebut, dengan terlebih dahulu diasumsikan bahwa kedua perusahaan tersebut telah mengetahui strategi-strategi yang digunakan oleh pihak lawannya..

1.6 SISTEMATIKA

BAB I : Memuat pendahuluan yang terdiri dari latar belakang, permasalahan, tujuan, metodologi, relevansi dan sistematika.

BAB II : Memuat uraian mengenai programa linier, yang menjadi dasar bagi Metode Simpleks dan Teori Permainan.

BAB III : Memuat teori yang menunjang proses pembuatan perangkat lunak yang meliputi penjelasan singkat dari Metode Simpleks.

BAB IV : Memuat teori mengenai Teori Permainan yang merupakan inti dari penyusunan Tugas Akhir ini.

BAB V : Memuat konstruksi perangkat lunak, yang meliputi penjelasan fungsional dari algoritma fungsi utama beserta data yang mendukung.

BAB VI : Memuat penjelasan hasil evaluasi dari perangkat lunak.

BAB VII : Memuat kesimpulan dan saran.

Pada bagian akhir akan disajikan daftar pustaka dan lampiran mengenai cara penggunaan perangkat lunak yang dibuat.

TUGAS AKHIR

BAB II

PROGRAMA LINIER

BAB II

PROGRAMA LINIER

2.1 PENGERTIAN UMUM

Programa linier adalah suatu cara untuk menyelesaikan persoalan pengalokasian sumber-sumber yang terbatas yang digunakan oleh beberapa aktivitas yang bersaing, dengan cara yang terbaik yang mungkin dilaksanakan. Persoalan pengalokasian ini akan muncul manakala seseorang harus memilih tingkat aktivitas-aktivitas tertentu yang bersaing dalam hal penggunaan sumber daya langka yang dibutuhkan untuk melaksanakan aktivitas-aktivitas tersebut.

Programa linier menggunakan model matematis untuk menjelaskan persoalan yang dihadapinya. Sifat "linier" disini memberi arti bahwa seluruh fungsi matematis yang ada merupakan fungsi linier.

Definisi : " *Suatu fungsi* $f(x_1, x_2, \ldots, x_n)$ *dari* $x_1, x_2, \ldots, x_n$ *adalah fungsi linier, jika dan hanya jika untuk sejumlah himpunan konstanta* $c_1, c_2, \ldots, c_n$ *berlaku hubungan :*

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$ "[1]

---

[1] Tjutju Tarliah Dimyati, Ahmad Dimyati., *Operation Research: Model-model Pengambilan Keputusan*, edisi ke-3; Sinar Baru Algensindo PT.,Bandung, 1994, hal. 23.

Sedangkan kata "programa" merupakan sinonim untuk perencanaan. Dengan demikian, programa linier adalah perencanaan aktivitas-aktivitas untuk memperoleh suatu hasil yang optimum, yaitu suatu hasil yang mencapai tujuan terbaik diantara seluruh alternatif yang memungkinkan.

Programa linier penggunaannya sangat luas, baik dalam perusahaan-perusahaan atau dalam kegiatan masyarakat sehari-hari. Aplikasi yang dapat diselesaikan dengan programa linier diantaranya alokasi fasilitas produksi, persoalan pengalokasian sumber daya nasional untuk berbagai kebutuhan, penjadwalan produksi, pemilihan pola pengiriman (*shipping*), desain terapi radiasi, solusi permainan (*game*) dan sebagainya.

2.2 MODEL PROGRAMA LINIER

Untuk menggambarkan persoalan programa linier, akan lebih mudah untuk dipahami apabila data untuk persoalan tersebut dikonversikan kedalam suatu tabel (tabel data model programa linier)[2] sebagai berikut :

---

2 ibid, hal. 24

Tabel 2.1 Data model programa linier

Aktivitas	Penggunaan Sumber per Unit				Banyaknya Sumber yang Dapat Digunakan
Sumber	1	2	...	n	
1	a_{11}	a_{12}	...	a_{1n}	b_1
2	a_{21}	a_{22}	...	$_{2n}$	b_2
. . .	. . .	. . .	. . .	. . .	. . .
m	a_{m1}	a_{m2}	...	a_{mn}	b_m
z per unit	c_1	c_2	...	c_n	
Tingkat	x_1	x_2	...	x_n	

Adapun arti dari setiap simbol pada tabel diatas adalah sebagai berikut :

1. Terdapat sejumlah **m** buah sumber yang terbatas yang akan dialokasikan di antara sejumlah **n** buah aktivitas yang bersaing, dengan ketentuan m ≤ n.
2. $\mathbf{x_j}$ (j = 1, 2, ... , n) menyatakan tingkat aktivitas ke *j* (variabel keputusan ke *j*)
3. **z** menyatakan nilai dari fungsi tujuan atau ukuran keefektifan yang terpilih.
4. $\mathbf{c_j}$ (j = 1, 2, ... , n) menyatakan koefisien variabel keputusan ke *j* pada fungsi tujuan atau dengan kata lain koefisien keuntungan (ongkos) per unit.
5. $\mathbf{b_i}$ (i = 1, 2, ... , m) menyatakan ruas kanan suatu pembatas atau dengan kata lain menyatakan banyaknya sumber ke *i* yang dapat digunakan dalam pengalokasian.
6. $\mathbf{a_{ij}}$ (i = 1, 2, ... , m ; dan j = 1, 2, ... , n) menyatakan koefisien dari variabel keputusan ke *j* pada pembatas ke *i* atau dengan kata lain menyatakan banyaknya sumber ke *i* yang dapat digunakan oleh unit aktivitas ke *j*.

Dari tabel diatas dapat dibuat suatu formulasi model matematis untuk persoalan programa linier, yaitu :

$$
\begin{aligned}
\text{Maksimum} &: z = c_1x_1 + c_2x_2 + \dots + c_nx_n \\
\text{Pembatas} &: a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \ \text{rop}\ b_1 \\
&\quad a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \ \text{rop}\ b_2 \\
&\quad \vdots \\
&\quad a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \ \text{rop}\ b_m \\
&\quad x_j \geq 0 \quad , j = 1, 2, \dots, n
\end{aligned}
$$

(rop berbentuk $\leq$, $\geq$ atau $=$)

Setiap situasi yang formulasi matematisnya memenuhi model diatas adalah persoalan programa linier. Apabila rop berbentuk (=), maka dinamakan **bentuk standar dari persoalan programa linier**.

2.3 PROSEDUR PENYELESAIAN PROGRAMA LINIER

Pada dasarnya, metode-metode yang dikembangkan untuk memecahkan persoalan programa linier ditujukan untuk mencari solusi dari beberapa alternatif solusi yang ada, yang dibentuk oleh persamaan-persamaan pembatas sehingga diperoleh nilai fungsi tujuan yang optimum. Ada 2 cara yang umum digunakan untuk menyelesaikan persoalan programa linier ini, yaitu dengan cara grafis dan dengan metode simpleks.[3]

---

[3] ibid, hal. 38

Cara grafis dapat dipergunakan apabila persoalan programa linier yang diselesaikan itu hanya mempunyai 2 buah variabel keputusan.

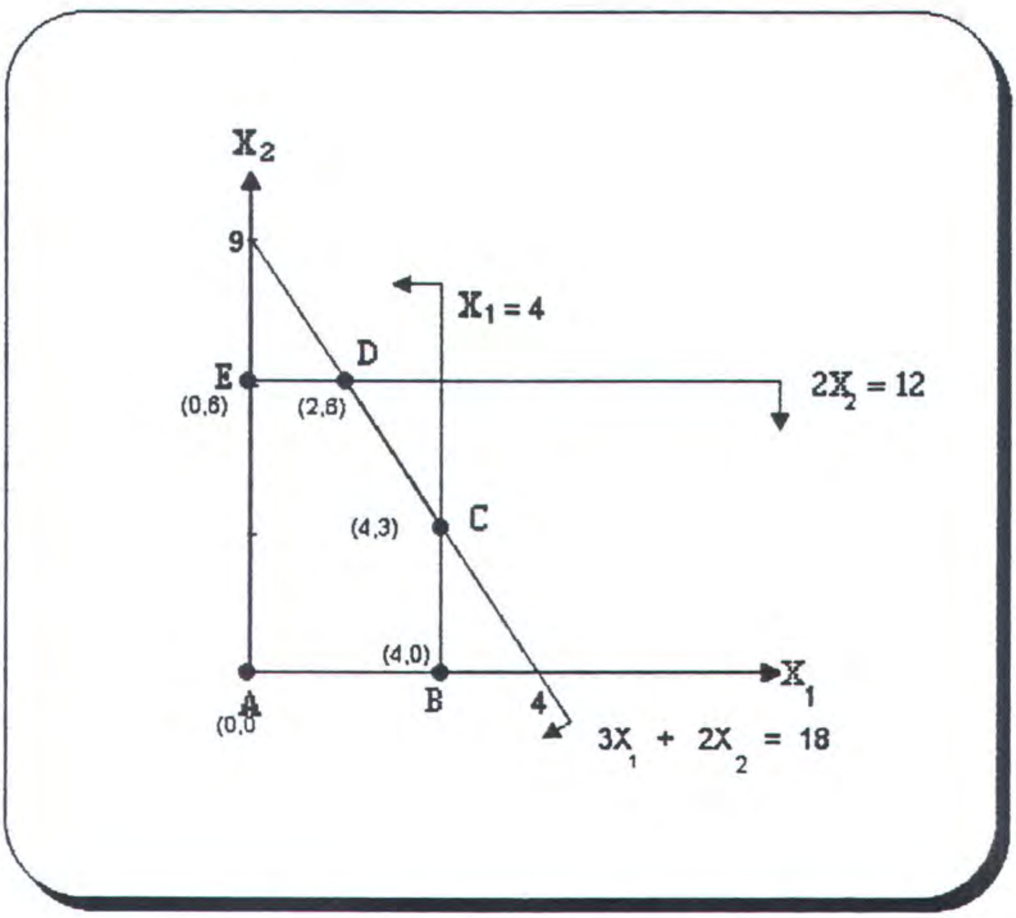
Sebagai gambaran berikut ini akan disajikan suatu contoh persoalan programa linier yang akan diselesaikan dengan menggunakan cara grafis.

Maksimum : $z = 3x_1 + 5x_2$

Pembatas :

$$\begin{aligned} 3x_1 + 2x_2 &\leq 18 \\ x_1 &\leq 4 \\ 2x_2 &\leq 12 \\ x_1 , x_2 &\geq 0 \end{aligned}$$

Penyelesaian :



Gambar 2.1 Contoh penyelesaian dengan Metode Grafis

Daerah yang dibatasi oleh ABCDE merupakan daerah yang fisibel untuk persoalan ini. Untuk mencari nilai dari z (fungsi tujuan), kita hanya perlu memperhatikan titik ekstrem (titik sudut) pada ruang solusi atau daerah fisibel.

Sehingga untuk setiap titik sudut dapat diperoleh nilai dari z yaitu :

$z(A) = 3*0 + 5*0 = 0$

$z(B) = 3*4 + 5*0 = 12$

$z(C) = 3*4 + 5*3 = 27$

$z(D) = 3*2 + 5*6 = 36$

$z(E) = 3*0 + 4*6 = 24$

Karena persoalan ini merupakan persoalan memaksimumkan, maka solusi optimum adalah titik dengan nilai fungsi tujuan terbesar yaitu titik D (2 , 6), sehingga nilai dari setiap variabel keputusan adalah $x_1 = 2$; $x_2 = 6$ serta nilai $z = 36$.

Metode Simpleks merupakan teknik yang paling berhasil dikembangkan untuk memecahkan persoalan programa linier yang mempunyai variabel keputusan dan pembatas dalam jumlah yang besar. Algoritma simpleks ini diterangkan dengan menggunakan logika secara aljabar matriks, sedemikian sehingga operasi perhitungan dapat dibuat lebih efisien.

TUGAS AKHIR

BAB III

METODE SIMPLEKS

BAB III

METODE SIMPLEKS

Metode Simpleks pertama kali diajukan pada tahun 1947 oleh George Dantzig[4], dan telah terbukti dapat menyelesaikan masalah programa linier dengan sangat efisien.

3.1 PENGERTIAN UMUM

Metode Simpleks merupakan metode untuk menyelesaikan persoalan programa linier dengan menggunakan prosedur aljabar yang bergerak selangkah demi selangkah, dimulai dari suatu titik ekstrem pada daerah fisibel (ruang solusi) menuju titik ekstrem yang optimum.[5]

Formulasi model matematis untuk persoalan programa linier adalah :

Maksimum atau Minimum : $z = c_1x_1 + c_2x_2 + \dots + c_nx_n$

Pembatas :

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \\
x_j \geq 0, \quad j &= 1, 2, \dots, n
\end{aligned}
$$

[4] Siagian, P., *Riset Operasional Teori dan Praktek*, UI-PRESS, 1987, hal. 81.
[5] Taha, Hamdy A., *Operation Research: An Introduction*, edisi ke-4; Macmillan Publishing Co., Inc., New York, 1987, hal. 68.

Pembatas di atas dapat dituliskan kedalam bentuk persamaan AX = b, di mana :

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ : & : & : & : \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ : \\ x_n \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ : \\ b_m \end{bmatrix}$$

Dalam Metode Simpleks terdapat beberapa istilah yang penting yang berkaitan dengan sistem persamaan AX = b, yaitu :

1. **Solusi basis** (*Basic Solution*)

   Solusi basis untuk AX = b adalah solusi di mana terdapat paling sedikit (n - m) variabel bernilai nol. Untuk mendapatkan solusi basis dari AX = b maka sebanyak (n - m) variabel harus diberi nilai nol. Variabel-variabel yang diberi nilai nol tersebut dinamakan **variabel nonbasis** (*Non Basic Variable*). Sedangkan variabel lainnya sebanyak m buah yang tidak bernilai nol dinamakan **variabel basis** (*Basic Variable*).

2. **Solusi basis fisibel** (*Basic Feasible Solution*)

Solusi basis fisibel adalah suatu solusi basis yang seluruh variabelnya bernilai nonnegatif.

3. **Solusi fisibel titik ekstrem**

Solusi fisibel titik ekstrem adalah solusi fisibel yang tidak terletak pada suatu segmen garis yang menghubungkan dua solusi fisibel lainnya.

Secara garis besar, prosedur penyelesaian persoalan programa linier dengan menggunakan Metode Simpleks terdiri dari 3 langkah utama[6] sebagai berikut :

1. **Langkah inisialisasi**

Pada langkah inisialisasi yang perlu dilakukan adalah menetapkan titik ekstrem awal (solusi awal) bagi persoalan programa linier yang sedang dihadapi.

2. **Langkah iteratif**

Pada langkah iteratif yang perlu dilakukan adalah menguji optimalitas dari suatu titik ekstrem dan bergerak ke titik ekstrem lainnya yang berdekatan apabila titik ekstrem tersebut belum optimal.

3. **Langkah penghentian**

Pada langkah penghentian yang perlu dilakukan adalah memberhentikan langkah ke 2 apabila telah sampai pada titik ekstrem yang terbaik (optimum).

---

[6] Tjutju Tarliah Dimyati, Ahmad Dimyati., *Operation Research: Model-model Pengambilan Keputusan*, edisi ke-3; Sinar Baru Algensindo PT.,Bandung, 1994., hal. 49-50.

3.2 BENTUK STANDAR PROGRAMA LINIER

Dalam menyelesaikan persoalan programa linier dengan menggunakan Metode Simpleks, bentuk programa linier yang digunakan haruslah bentuk standar. Secara matematis bentuk standar programa linier dinyatakan sebagai berikut :

Maksimum atau Minimum : $z = c_1x_1 + c_2x_2 + \dots + c_nx_n$

Pembatas :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \\ x_j \geq 0, \quad j = 1, 2, \dots, n \end{aligned}$$

Dalam persoalan yang sebenarnya, ternyata banyak diketemukan bentuk dari programa linier yang tidak sesuai dengan bentuk standar di atas. Bentuk programa linier yang tidak standar memiliki sifat-sifat sebagai berikut :

1. Paling sedikit satu pembatas berbentuk ketidaksamaan (bertanda $\leq$ atau $\geq$).
2. Paling sedikit satu ruas kanan bernilai negatif.
3. Paling sedikit satu variabel keputusan berupa variabel yang tidak terbatas dalam tanda (*unrestricted in sign*).

Untuk mengubah suatu bentuk program linier yang tidak standar ke dalam bentuk yang standar dilakukan dengan cara-cara sebagai berikut :

1. Tidak standar dalam pembatas
   a. Pembatas yang berbentuk ketidaksamaan (bertanda $\leq$), dapat dijadikan bentuk standar yaitu persamaan, dengan menambahkan **variabel slack** (variabel tambahan) pada ruas kiri pembatas tersebut.

Secara matematis dapat dinyatakan sebagai :

Pembatas ke i : $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$

dinyatakan dalam bentuk standar menjadi :

Pembatas ke i : $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + S_i = b_i$

b. Pembatas yang berbentuk ketidaksamaan (bertanda $\geq$), dapat dijadikan bentuk standar yaitu persamaan, dengan mengurangkan variabel slack dari ruas kiri pembatas tersebut.

Secara matematis dapat dinyatakan sebagai :

Pembatas ke i : $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i$

dinyatakan dalam bentuk standar menjadi :

Pembatas ke i : $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - S_i = b_i$

2. Tidak standar dalam ruas kanan

Ruas kanan yang negatif dapat dijadikan bentuk standar, yaitu ruas kanan yang nonnegatif, dengan cara mangalikan kedua ruas dengan -1.

Secara matematis dapat dinyatakan sebagai berikut :

Pembatas ke i : $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = -b_i$

dinyatakan dalam bentuk standar menjadi :

Pembatas ke i : $-a_{i1}x_1 - a_{i2}x_2 - \dots - a_{in}x_n = b_i$

3. Tidak standar dalam variabel keputusan

Variabel keputusan yang tidak terbatas dalam tanda dapat dijadikan bentuk yang standar, yaitu variabel keputusan yang nonnegatif, dengan cara mensubstitusikan variabel keputusan tersebut menjadi dua buah variabel yang nonnegatif pada seluruh pembatas dan fungsi tujuannya.

Substitusi tersebut berbentuk : $x_i = x_i' - x_i''$, di mana x_i berupa variabel keputusan yang tidak terbatas dalam tanda dan $x_i', x_i'' \geq 0$.

Selain perubahan yang dapat dilakukan seperti di atas, kadangkala lebih memudahkan menyelesaikan persoalan maksimasi daripada persoalan minimasi. Oleh karena itu ada suatu ketentuan yang menyatakan bahwa maksimasi suatu fungsi sama dengan minimasi dari negatip fungsi tersebut atau kebalikan dari fungsi tersebut.[7]

Secara matematis dapat dinyatakan sebagai berikut :

Maksimum : $z = c_1x_1 + c_2x_2 + ... + c_nx_n$

ekuivalen dengan

Minimum : $(-z) = -c_1x_1 - c_2x_2 - ... - c_nx_n$, atau

Minimum : $z = 1 / (c_1x_1 + c_2x_2 + ... + c_nx_n)$

[7] ibid, hal. 66

3.3 PENYELESAIAN PERSOALAN MAKSIMASI

Suatu persoalan programa linier disebut persoalan maksimasi jika dan hanya jika fungsi tujuannya berupa fungsi yang memaksimumkan.

Sesuai dengan 3 langkah dasar penyelesaian dengan Metode Simpleks, maka persoalan maksimasi dengan menggunakan Metode Simpleks juga dilakukan dengan 3 langkah tersebut, yaitu :

1. Langkah inisialisasi
   a. Mengkonversikan formulasi persoalan kedalam bentuk yang standar.
   b. Mencari solusi basis fisibel awal.
2. Langkah iteratif
   a. Menetukan optimalitas persoalan dengan melihat seluruh variabel nonbasis pada baris fungsi tujuan (baris persamaan z yang biasa juga disebut baris 0), seperti yang tercantum pada langkah 3.
   b. Menentukan **variabel yang masuk basis** (*Entering Variable*) dengan cara memilih variabel nonbasis yang mempunyai koefisien paling negatif pada baris fungsi tujuan. Jika terdapat lebih dari satu koefisien paling negatif yang sama, pilihlah salah satu diantaranya, hal ini tidak akan mempengaruhi proses berikutnya.
   c. Menetukan rasio dari (ruas kanan) / (koefisien variabel yang masuk basis) pada setiap baris pembatas yang mempunyai koefisien variabel yang masuk basis positif.
   d. Menentukan **variabel yang meninggalkan basis** (*Leaving Variable*) dengan cara memilih variabel basis yang ada pada pembatas yang

memiliki rasio (langkah 2.c) terkecil. Jika terdapat lebih dari satu rasio yang sama, pilihlah salah satu diantaranya, hal ini tidak akan mempengaruhi proses berikutnya.

e. Melakukan operasi baris elementer dengan Metode Eliminasi Gauss-Jordan untuk membuat koefisien variabel yang masuk basis pada baris pembatas dengan rasio positif terkecil ini menjadi bernilai 1 dan bernilai 0 pada baris-baris lainnya.

f. Mencari solusi basis fisibel berikutnya dengan cara menukar status antara variabel yang masuk basis dengan variabel yang meninggalkan basis kemudian kembali ke langkah 2.a.

3. Langkah penghentian

Apabila seluruh variabel nonbasis pada baris fungsi tujuan mempunyai koefisien nonnegatif maka solusi basis fisibel sudah optimum, sedangkan apabila masih terdapat koefisien yang bernilai negatif maka solusi basis fisibel belum optimum dan harus dilakukan langkah-langkah iteratif berikutnya.

3.4 DUAL SIMPLEKS

3.4.1 TEORI DUALITAS

Ide dasar yang melatarbelakangi teori dualitas adalah bahwa setiap persoalan programa linier mempunyai suatu programa linier lain yang saling berkaitan yang disebut "dual", sedemikian sehingga solusi pada persoalan semula (yang disebut "primal") juga memberikan solusi pada dualnya.

Pendefinisian dual ini akan bergantung pada jenis pembatas, tanda-tanda variabel, dan bentuk optimasi dari persoalan primalnya. Akan tetapi, karena setiap persoalan programa linier harus dibuat dalam bentuk standar lebih dahulu sebelum modelnya dipecahkan, maka pendefinisian di bawah ini akan secara otomatis meliputi ketiga hal di atas.

Bentuk umum dari masalah primal-dual adalah sebagai berikut :

Primal :

Maksimum : $z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$

Pembatas :

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \\
x_j \geq 0, \quad j &= 1, 2, \ldots, n
\end{aligned}
$$

Dual :

Minimum : $w = b_1y_1 + b_2y_2 + \dots + b_my_m$

Pembatas :

$$a_{11}y_1 + a_{21}y_2 + \dots + a_{m1}y_m = c_1$$
$$a_{12}y_1 + a_{22}y_2 + \dots + a_{m2}y_m = c_2$$
$$\vdots$$
$$a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m = c_n$$
$$y_i \geq 0, \quad i = 1, 2, \dots, m$$

Korespondensi antara primal dengan dual adalah sebagai berikut :

1. Koefisien fungsi tujuan primal menjadi konstanta ruas kanan bagi dual, sedangkan konstanta ruas kanan primal menjadi koefisien fungsi tujuan bagi dual.
2. Untuk setiap pembatas primal terdapat satu variabel dual, dan untuk setiap variabel primal terdapat satu pembatas dual.
3. Tanda ketidaksamaan pada pembatas akan bergantung pada fungsi tujuannya.
4. Fungsi tujuan berubah bentuk (maksimasi menjadi minimasi dan minimasi menjadi maksimasi).
5. Setiap kolom pada primal berkorespondensi dengan baris (pembatas) pada dual.
6. Setiap baris (pembatas) pada primal berkorespondensi dengan kolom pada dual.
7. Dual dari dual adalah primal.

3.4.2 HUBUNGAN PRIMAL-DUAL

Untuk menyelesaikan persoalan maksimasi dan minimasi dari suatu programa linier, hanya diperlukan satu penyelesaian (penyelesaian untuk maksimasi atau minimasi) sedangkan penyelesaian yang lainnya (penyelesaian untuk minimasi atau maksimasi) dapat diperoleh dari penyelesaian yang pertama melalui hubungan antara primal dengan dual.

Hubungan antara penyelesaian primal dengan dual adalah sebagai berikut :

1. Solusi fisibel persoalan minimasi adalah batas atas dari solusi fisibel persoalan maksimasi
2. Jika kedua persoalan sudah mencapai solusi optimum, maka maks z = min w
3. Nilai optimum variabel-variabel solusi awal pada primal sama dengan nilai optimum variabel- variabel dual yang berkorespondensi dengan persamaan pembatas pada primal.

TUGAS AKHIR

BAB IV

TEORI PERMAINAN

BAB IV

TEORI PERMAINAN

4.1 PENGERTIAN UMUM

Teori Permainan (*Game Theory*) adalah bagian dari ilmu pengetahuan mengenai optimasi yang mempelajari hal-hal yang berkaitan dengan pembuatan keputusan pada saat dua pihak atau lebih berada dalam kondisi persaingan. Pihak-pihak yang bersaing ini mempunyai kepentingan yang sama, seperti mencari keuntungan dari hasil penjualan produk yang sama, kampanye untuk pemilu dan sebagainya. Selain itu pihak-pihak yang bersaing ini diasumsikan juga bersifat rasional dan cerdas, yang berarti masing-masing pihak akan melakukan strategi tindakan yang rasional untuk memenangkan persaingan itu, dan masing-masing pihak juga mengetahui strategi dari pihak lawannya. Ide pembuatan keputusan pada situasi persaingan ini merupakan inti dari keputusan-keputusan manajerial. Dalam Teori Permainan, pihak-pihak yang bersaing ini disebut pemain.

Teori Permainan dapat diklasifikasikan dalam beberapa jenis, tergantung pada faktor-faktor sebagai berikut [9]:

---

[9] ibid, hal . 255

1. Banyaknya pemain

Berdasarkan banyaknya pemain yang terlibat dalam persaingan, model Teori Permainan dapat diklasifikasikan menjadi 2 jenis, yaitu :

a. **Permainan Dua Pemain** (*Two-Person Game*)

Apabila pemain yang terlibat dalam persaingan berjumlah 2, maka game tersebut dinamakan Permainan Dua Pemain.

Persaingan antara 2 perusahaan yang menjual produk yang sama untuk mendapatkan keuntungan yang optimum merupakan salah satu contoh dari Permainan Dua Pemain yang paling sering dijumpai dalam kehidupan sehari-hari.

b. **Permainan N Pemain** (*N-Person Game*)

Apabila pemain yang terlibat dalam persaingan berjumlah lebih dari 2, maka game tersebut dinamakan Permainan N Pemain.

Kampanye peserta Pemilihan Umum yang akan segera dilaksanakan untuk mendapatkan suara pemilih yang paling banyak, merupakan salah satu contoh dari Permainan N Pemain.

2. Jumlah keuntungan dan kerugian yang diperoleh

Yang dimaksudkan dengan jumlah keuntungan dan kerugian yang diperoleh dalam suatu permainan adalah jumlah dari seluruh keuntungan dan kerugian yang didapatkan oleh seluruh pemain yang terlibat dalam persaingan tersebut.Berdasarkan jumlah keuntungan dan kerugian yang diperoleh dalam suatu persaingan, model Teori Permainan dapat diklasifikasikan menjadi 2 jenis, yaitu:

a. **Permainan Jumlah Nol** (*Zero-Sum Game)*

Apabila jumlah keuntungan dan kerugian yang diperoleh oleh seluruh pemain yang terlibat dalam persaingan tersebut berjumlah nol, maka permainan tersebut dinamakan Permainan Jumlah Nol atau Permainan Jumlah Tetap (*Constant-Sum Game)*.

Sebelum pertandingan dimulai, 2 kesebelasan sepak bola mengadakan kesepakatan untuk membayar sebesar Rp.100.000,00 kepada kesebelasan yang menang dari kesebelasan yang kalah. Kejadian tersebut merupakan salah satu contoh dari Permainan Jumlah Nol, karena jumlah keuntungan dan kerugian yang diperoleh oleh semua pemain (2 pemain, yaitu kedua kesebelasan yang bertanding) berjumlah nol (Rp.100.000,00 - Rp.100.000,00 = 0).

b. **Permainan Jumlah Tidak Nol** (*Non-Zero-Sum Game)*

Apabila jumlah keuntungan dan kerugian yang diperoleh oleh seluruh pemain yang terlibat dalam persaingan tersebut berjumlah selain nol, maka permainan tersebut dinamakan Permainan Jumlah Tidak Nol.

Situasi ketika sejumlah perusahaan melakukan kampanye advertensi yang intensif untuk mendapatkan daerah pemasaran yang lebih besar, merupakan salah satu contoh dari Permainan Jumlah Tidak Nol karena ukuran total pasar biasanya akan meningkat sebagai akibat dari advertensi yang intensif. Demikian pula jumlah kerugian dan keuntungannya akan positif.

Sampai saat ini pengembangan dari model-model permainan hanya terbatas pada persaingan antara dua pemain dengan jumlah keuntungan dan kerugian nol.

4.2 PERMAINAN DUA PEMAIN DENGAN JUMLAH NOL

4.2.1 ELEMEN-ELEMEN DASAR TEORI PERMAINAN

Secara umum, persoalan **Permainan Dua Pemain Dengan Jumlah Nol** (*Two-Person Zero-Sum Game*) dapat dinyatakan dalam bentuk tabel untuk menyederhanakan pengertian sebagai berikut :

Tabel 4.1 Permainan Dua Pemain Dengan Jumlah Nol

		Pemain B			
		B1	B2	...	Bn
Pemain A	A1	a_{11}	a_{12}	...	a_{1n}
	A2	a_{21}	a_{22}	...	a_{2n}
	.	.	.	.	.
	.	.	.	.	.
	.	.	.	.	.
	Am	a_{m1}	a_{m2}	...	a_{mn}

Beberapa pengertian dari tabel di atas adalah sebagai berikut :

1. Tabel tersebut dinamakan juga Matriks Pembayaran (*payoff matrix*).
2. Strategi adalah tindakan pilihan. Dalam hal ini diasumsikan bahwa strategi ini tidak dapat dibolak-balik oleh para pemain. Sebagai contoh, pada tabel di atas Pemain A mempunyai m jenis strategi, sementara Pemain B memiliki n jenis strategi.

3. Aturan Permainan menjelaskan tentang bagaimana cara para pemain memilih strategi-strategi mereka. Diasumsikan bahwa para pemain itu harus memiliki strategi mereka secara serentak, dan bahwa permainannya dilakukan secara berulang-ulang.
4. Bilangan-bilangan yang ada di dalam Matriks Pembayaran (a_{ij} , untuk $i = 1,2, \ldots ,m$ dan $j = 1,2, \ldots ,n$) menyatakan pembayaran (*outcome atau payoff*) dari strategi permainan yang berbeda. Pembayaran ini diartikan sebagai ukuran keefektifan seperti uang, persentase daerah pemasaran, atau utilitas. Bilangan positif menyatakan perolehan (keuntungan) bagi pemain yang ditulis pada baris sebagai **pemain yang akan memaksimumkan**, dan sekaligus sebagai merupakan kerugian bagi pemain yang ditulis pada kolom sebagai **pemain yang akan meminimumkan**.

   Sebagai contoh, jika Pemain A melakukan strategi Ai dan Pemain B memilih strategi Bj, maka Pemain A akan memperoleh keuntungan (kerugian) sebesar a_{ij} jika a_{ij} merupakan bilangan positif (negatif), sedangkan Pemain B akan memperoleh kerugian (keuntungan) sebesar a_{ij} jika a_{ij} merupakan bilangan positif (negatif).
5. Suatu strategi dinyatakan dominan apabila setiap pembayaran yang ada pada suatu strategi bersifat superior (paling tinggi) dibandingkan dengan setiap pembayaran pada strategi lainnya. Sebagai contoh, strategi A2 mendominasi strategi A4 jika dan hanya jika $a_{2j} > a_{4j}$, untuk $j = 1,2, \ldots, n$. Sehingga strategi A4 dapat dihilangkan dalam perhitungan selanjutnya. Aturan

dominansi ini dapat digunakan untuk mengurangi ukuran Matriks Pembayaran dan menyederhanakan perhitungan.

6. Strategi Optimum adalah strategi yang menjadikan seorang pemain berada pada posisi pilihan terbaik, tanpa memperhatikan tindakan-tindakan pemain lawannya. Arti posisi pilihan terbaik ini adalah bahwa setiap penyimpangan dari Strategi Optimum ini akan menyebabkan turunnya pembayaran.
7. Nilai Permainan menyatakan ekspektasi pembayaran per Permainan jika kedua pemain melakukan strategi terbaik (strategi optimum) mereka. Suatu Permainan dikatakan adil (*fair*) jika Nilai Permainannya nol, dan dinyatakan tidak adil jika Nilai Permainannya bukan nol.
8. Tujuan model Permainan adalah untuk mengidentifikasi Strategi Optimum bagi setiap pemain.

Dalam melakukan pemilihan strategi yang tepat untuk mendapatkan keuntungan yang optimum, setiap pemain melakukan berbagai cara untuk mendapatkannya. Secara umum, persoalan Permainan Dua Pemain Dengan Jumlah Nol terbagi menjadi 2 jenis berdasarkan cara pemilihan strategi untuk mendapatkan posisi pilihan terbaik, yaitu :

a. **Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Tunggal** (*Pure-Strategy Two-Person Zero-Sum Game*)
b. **Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran** (*Mixed-Strategy Two-Person Zero-Sum Game*)

4.2.2 PERMAINAN DUA PEMAIN DENGAN JUMLAH NOL MENGGUNAKAN STRATEGI TUNGGAL

Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Tunggal adalah permainan yang posisi pilihan terbaiknya bagi setiap pemain dicapai dengan memilih satu strategi tunggal.

Pemain yang akan memaksimumkan (pada tabel 4.1 adalah Pemain A) akan mengidentifikasi strategi optimumnya dengan menggunakan **kriteria maksimin**, sedangkan pemain yang akan meminimumkan (pada tabel 4.1 adalah pemain B) akan mengidentifikasi strategi optimumnya dengan menggunakan **kriteria minimaks**.

Kriteria maksimin adalah suatu cara untuk mendapatkan nilai maksimin yaitu nomor baris tempat strategi optimum berada. Algoritma untuk mencari nilai maksimin adalah sebagai berikut :

1. Dapatkan nilai minimum dari masing-masing baris.
2. Nilai terbesar (nilai maksimum) dari nilai-nilai yang diperoleh pada langkah (1) adalah nilai maksimin.

Kriteria minimaks adalah suatu cara untuk mendapatkan nilai minimaks yaitu nomor kolom tempat strategi optimum berada. Algoritma untuk mencari nilai minimaks adalah sebagai berikut :

1. Dapatkan nilai maksimum dari masing-masing kolom.
2. Nilai terkecil (nilai minimum) dari nilai-nilai yang diperoleh pada langkah (1) adalah nilai minimaks.

Jika nilai maksimin sama dengan nilai minimaks, maka permainan telah terpecahkan. (Untuk menguji hal ini, nilai tersebut harus merupakan nilai maksimum bagi kolom yang bersangkutan dan sekaligus merupakan nilai minimum bagi baris yang bersangkutan). Dalam kasus seperti ini, maka telah tercapai **titik keseimbangan** (*saddle point*).

Jika nilai maksimin tidak sama dengan nilai minimaks, maka titik keseimbangan tidak akan tercapai. Hal ini berarti titik keseimbangannya tidak ada dan permainan ini tidak dapat diselesaikan dengan strategi tunggal. Akibatnya, suatu permainan yang tidak mempunyai titik keseimbangan harus diselesaikan dengan menggunakan strategi campuran.

Sebagai gambaran, di bawah ini disajikan suatu contoh mengenai persoalan Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Tunggal.

Matriks Pembayaran :

Tabel 4.2 Contoh Permainan Dua Pemain
Dengan Jumlah Nol Menggunakan Strategi Tunggal

		Pemain B			
		B1	B2	B3	B4
Pemain A	A1	3	4	2	4
	A2	4	7	5	3
	A3	7	8	9	10
	A4	4	3	4	4

Minimum baris :	Maksimum kolom :
- Baris 1 : 2	- Kolom 1 : 7
- Baris 2 : 3	- Kolom 2 : 8
- Baris 3 : 7	- Kolom 3 : 9
- Baris 4 : 3	- Kolom 4 : 10
Nilai Maksimin : 7 (baris 3)	Nilai Minimaks : 7 (kolom 1)

Karena nilai maksimin sama dengan nilai minimaks, maka persoalan sudah terpecahkan, yaitu Pemain A akan menggunakan strategi 3 dan Pemain B akan menggunakan strategi 1.

4.3 PERMAINAN DUA PEMAIN DENGAN JUMLAH NOL MENGGUNAKAN STRATEGI CAMPURAN

Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran adalah permainan yang posisi pilihan terbaiknya bagi setiap pemain dicapai dengan melakukan pencampuran terhadap strategi-strategi yang berbeda.

Seperti yang telah dikemukakan pada bagian sebelumnya, pada permainan yang tidak mempunyai titik keseimbangan, penyelesaiannya harus dilakukan dengan menggunakan strategi campuran.

Sebagai gambaran, di bawah ini akan disajikan suatu contoh permainan yang tidak mempunyai titik keseimbangan, sehingga penyelesaiannya harus menggunakan strategi campuran.

Tabel 4.3 Contoh Permainan Dua Pemain

Dengan Jumlah Nol yang tidak memiliki titik keseimbangan

		Pemain B			Minimum Baris
		1	2	3	
Pemain A	1	0	-2	2	-2
	2	5	4	-3	-3
	3	2	3	-4	-4
Maksimum kolom		5	4	2	

Pada permainan di atas nilai maksimin (-2) yaitu untuk strategi A1 dan nilai minimaks (2) yaitu untuk strategi B3. Karena nilai maksimin tidak sama dengan nilai minimaks, maka persoalan di atas tidak memiliki titik keseimbangan. Pada permainan ini, jika A memilih strategi 1 maka B akan memilih strategi 2. Sedangkan jika B memilih strategi 2 maka A akan memilih strategi 2. Jika A memilih strategi 2 maka B akan memilih strategi 3. Jika B memilih strategi 3 maka A akan memilih strategi 1. Kemudian jika A memilih strategi 1 maka B akan memilih strategi 2, yang menyebabkan pemilihan strategi akan kembali seperti semula tanpa ada kesepakatan diantara keduanya. Sehingga permainan yang tidak memiliki titik keseimbangan disebut **permainan yang tidak stabil** (*unstable game*).

Berbeda dengan permainan strategi tunggal, pada permainan yang tidak mempunyai titik keseimbangan ini para pemain dapat memainkan seluruh strateginya sesuai dengan set probabilitas yang telah ditetapkan.

Jika :

x_i = probabilitas Pemain A memilih strategi i (i = 1, 2, ... , m)

y_j = probabilitas Pemain B memilih strategi j (j = 1, 2, ... , n)

dengan m dan n adalah banyaknya strategi yang dapat digunakan dan berlaku pula ketentuan bahwa : $\sum_{i=1}^{m} x_i = \sum_{j=1}^{n} y_j = 1$, dengan x_i, $y_j \geq 0$ untuk setiap i dan j.

Dengan demikian, matriks pembayarannya dapat digambarkan sebagai berikut :

Tabel 4.4 Matriks Pembayaran untuk Permainan Dua Pemain

Dengan Jumlah Nol Menggunakan Strategi Campuran

		Pemain B			
		y_1	y_2	...	y_n
P	x_1	a_{11}	a_{12}	...	a_{1n}
e m	x_2	a_{21}	a_{22}	...	a_{2n}
a	.	.	.	.	.
i	.	.	.	.	.
n	.	.	.	.	.
A	x_m	a_{m1}	a_{m2}	...	a_{mn}

Solusi persoalan strategi campuran ini masih didasarkan pada kriteria maksimin dan minimaks. Perbedaannya adalah bahwa A akan memilih x_i yang memaksimumkan ekspektasi pembayaran terkecil pada suatu kolom, sedangkan B memilih y_j yang dapat meminimumkan ekspektasi pembayaran terbesar pada suatu baris.

Secara matematis dapat dinyatakan sebagai berikut :

- Pemain A akan memilih x_i ($x_i \geq 0$, $\sum_{i=1}^{m} x_i = 1$) yang dapat menghasilkan

$$\underline{v} = \underset{x_i}{\text{maks}} \left\{ \min \left(\sum_{i=1}^{m} a_{i1} x_i , \sum_{i=1}^{m} a_{i2} x_i , \ldots , \sum_{i=1}^{m} a_{in} x_i \right) \right\}$$

- Pemain B akan memilih y_j ($y_j \geq 0$, $\sum_{j=1}^{n} y_j = 1$) yang dapat menghasilkan

$$\bar{v} = \underset{y_j}{\min} \left\{ \text{maks} \left(\sum_{j=1}^{n} a_{1j} y_j , \sum_{j=1}^{n} a_{2j} y_j , \ldots , \sum_{j=1}^{n} a_{mj} y_j \right) \right\}$$

Nilai-nilai di atas adalah nilai-nilai maksimin dan minimaks dari yang ekspektasi pembayaran. Jika x_i dan y_j berkorespondensi dengan solusi optimum, maka $\bar{v} = \underline{v}$ di mana nilai yang diperoleh akan sama dengan nilai ekspektasi optimum dari permainan.

Jika x_i^* dan y_j^* adalah solusi optimum bagi kedua pemain, maka setiap elemen pembayaran a_{ij} akan dihubungkan dengan probabilitas (x_i^* , y_j^*).
Dengan demikian, maka nilai ekspektasi optimum (Nilai Permainan) dari permainannya adalah :

$$v^* = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_i^* y_j^*$$

Ada beberapa metode yang dapat digunakan untuk menyelesaikan permainan jenis ini, di antaranya dengan cara grafis dan dengan menggunakan programa linier.

4.4 PROSEDUR PENYELESAIAN DENGAN PROGRAMA LINIER

Persoalan Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran dapat dipandang sebagai persoalan programa linier. Seperti yang telah dikemukakan sebelumnya, pemain A menggunakan kriteria maksimin yang dapat diformulasikan secara matematis sebagai berikut :

$$\max_{x_i} \left\{ \min \left(\sum_{i=1}^{m} a_{i1}x_i, \sum_{i=1}^{m} a_{i2}x_i, \ldots, \sum_{i=1}^{m} a_{in}x_i \right) \right\}$$

di mana $\sum_{i=1}^{m} x = 1$ dan $x_i \geq 0$ untuk $i = 1, 2, \ldots, m$

Jika $v = \min \left(\sum_{i=1}^{m} a_{i1}x_i, \sum_{i=1}^{m} a_{i2}x_i, \ldots, \sum_{i=1}^{m} a_{in}x_i \right)$

maka formulasi matematika untuk kriteria maksimin dapat dinyatakan sebagai berikut :

Maks : $x_0 = v$

Pembatas :

$$\sum_{i=1}^{m} a_{ij}x_i \geq v \quad , \text{untuk } j = 1, 2, \ldots, n \qquad (1)$$

$$\sum_{i=1}^{m} x = 1 \qquad (2)$$

$$x_i \geq 0 \quad , \text{untuk } i = 1, 2, \ldots, m \qquad (3)$$

Karena Nilai Permainan diusahakan agar tidak dapat bernilai negatif atau nol, maka nilai maksimin juga harus lebih besar dari nol ($v > 0$), sehingga pembatas (1) dapat diformulasikan kembali menjadi :

$$\sum_{i=1}^{m} a_{i} \frac{x_i}{v} \geq 1 \qquad \text{, untuk } j = 1, 2, \ldots, n$$

Pembatas (2), yaitu $\sum_{i=1}^{m} x = 1$, bila dikembangkan akan memiliki formulasi sebagai berikut :

$$x_1 + x_2 + \ldots + x_m = 1$$

Karena apabila kedua ruas dibagi dengan suatu variabel (variabel v) akan tetap menghasilkan suatu persamaan yang benar, maka persamaan di atas dapat ditulis menjadi :

$$\frac{x_1 + x_2 + \ldots + x_m}{v} = \frac{1}{v}$$

apabila $\frac{x_i}{v}$ dinotasikan sebagai X_i (i = 1, 2, ... , m) , maka persamaan di atas dapat dituliskan menjadi :

$$X_1 + X_2 + \ldots + X_n = \frac{1}{v}$$

Pada pembahasan sebelumnya, telah diketahui bahwa maksimasi suatu fungsi ekuivalen dengan minimasi (-fungsi) atau minimasi (1 / (fungsi)). Karena itu, fungsi tujuan dapat diganti, yaitu :

$$\text{Min}: x_0 = X_1 + X_2 + \ldots + X_m$$

Sehingga formulasi selengkapnya untuk kriteria maksimin adalah :

$$\text{Min}: x_0 = X_1 + X_2 + \ldots + X_m$$

$$\begin{aligned}
\text{Pembatas}: a_{11} X_1 + a_{21} X_2 + \ldots + a_{m1} X_m &\geq 1 \\
a_{12} X_1 + a_{22} X_2 + \ldots + a_{m2} X_m &\geq 1 \\
&\vdots \\
a_{1n} X_1 + a_{2n} X_2 + \ldots + a_{mn} X_m &\geq 1 \\
X_1 , X_2 , \ldots , X_m &\geq 0
\end{aligned}$$

Seperti yang telah dikemukakan sebelumnya, pemain B menggunakan kriteria minimaks yang dapat diformulasikan secara matematis sebagai berikut :

$$\min_{y_j} \left\{ \text{maks} \left(\sum_{j=1}^{n} a_{1j} y_j , \sum_{j=1}^{n} a_{2j} y_j , \ldots , \sum_{j=1}^{n} a_{mj} y_j \right) \right\}$$

di mana $\sum_{j=1}^{n} y_j = 1$ dan $y_j \geq 0$ untuk $j = 1, 2, \ldots , n$

Dengan cara yang sama seperti pada formulasi matematika untuk kriteria maksimin, maka formulasi matematika untuk kriteria minimaks dapat dinyatakan sebagai berikut :

$$\text{Maks}: y_0 = Y_1 + Y_2 + \ldots + Y_n$$

$$\begin{aligned}
\text{Pembatas}: a_{11} Y_1 + a_{12} Y_2 + \ldots + a_{1n} Y_n &\leq 1 \\
a_{21} Y_1 + a_{22} Y_2 + \ldots + a_{2n} Y_n &\leq 1 \\
&\vdots \\
a_{m1} Y_1 + a_{m2} Y_2 + \ldots + a_{mn} Y_n &\leq 1 \\
Y_1 , Y_2 , \ldots , Y_n &\geq 0
\end{aligned}$$

Ternyata persoalan Pemain B ini merupakan dual dari persoalan Pemain A, sehingga solusi optimum dari salah satu persoalan secara otomatis dapat menghasilkan solusi optimum bagi keduanya.

Berikut ini akan disajikan beberapa contoh mengenai persoalan Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran beserta penyelesaiannya.

Matriks pembayaran :

Tabel 4.5 Matriks Pembayaran untuk Contoh Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran

		Pemain B		
		B1	B2	B3
Pemain A	A1	3	-1	-3
	A2	-3	3	-1
	A3	-4	-3	3

Dari tabel di atas dapat diketahui nilai-nilai minimum untuk setiap baris dan nilai-nilai maksimum untuk setiap kolom.

Nilai minimum :

Baris 1 : -3

Baris 2 : -3

Baris 3 : -4

Nilai maksimum :

Kolom 1 : 3

Kolom 2 : 3

Kolom 3 : 3

Berdasarkan nilai-nilai minimum yang ada, kita dapat menghitung nilai maksiminnya yaitu nilai terbesar dari nilai-nilai minimum tersebut. Untuk persoalan ini nilai maksimin = -3.

Berdasarkan nilai-nilai maksimum yang ada, kita dapat menghitung nilai minimaksnya yaitu nilai terkecil dari nilai-nilai maksimum tersebut. Untuk persoalan ini nilai minimaks = 3.

Karena nilai maksimin tidak sama dengan nilai minimaks, maka persoalan tersebut memenuhi syarat untuk dapat disebut persoalan menggunakan strategi campuran. Karena nilai maksimin lebih kecil dari nol maka nilai dari permainan dapat bernilai negatif atau nol. Oleh karena itu perlu ditambahkan suatu konstanta terhadap setiap elemen pada matriks pembayaran. Besarnya nilai konstanta tidak akan mempengaruhi terhadap proses selanjutnya, dengan syarat konstanta tersebut harus bernilai minimal sama dengan nilai maksimin. Misalkan digunakan konstanta bernilai 5, maka matriks pembayaran berubah menjadi :

Tabel 4.6 Matriks Pembayaran setelah ditambah suatu konstanta

		Pemain B		
		B1	B2	B3
Pemain A	A1	8	4	2
	A2	2	8	4
	A3	1	2	8

Apabila digunakan pemain B sebagai persoalan primal, maka formulasi persoalannya menjadi :

$$\text{Maks} : w = Y_1 + Y_2 + Y_3$$

$$\begin{aligned} \text{Pembatas} : 8Y_1 + 4Y_2 + 2Y_3 &\leq 1 \\ 2Y_1 + 8Y_2 + 4Y_3 &\leq 1 \\ Y_1 + 2Y_2 + 8Y_3 &\leq 1 \\ Y_1 , Y_2 , Y_3 &\quad 0 \end{aligned}$$

Kemudian akan dilakukan proses penyelesaian dengan Metode Simpleks seperti dibawah ini :

Tabel 4.7 Iterasi 0 Metode Simpleks

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 0	w	-1	-1	-1	0	0	0	0
	S_1	8	4	2	1	0	0	1
	S_2	2	8	4	0	1	0	1
	S_3	1	2	8	0	0	1	1

Tabel 4.8 Iterasi 1 Metode Simpleks

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 1	w	0	-0.5	-0.75	0.13	0	0	0.13
	Y_1	1	0.5	0.25	0.13	0	0	0.13
	S_2	0	7	3.5	-0.13	1	0	0.75
	S_3	0	1.5	7.75	-0.13	0	1	0.88

Tabel 4.9 Iterasi 2 Metode Simpleks

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 2	w	0	-0.35	0	0.11	0	0.1	0.21
	Y_1	1	0.45	0	0.13	0	-0.03	0.1
	S_2	0	6.32	0	-0.19	1	-0.45	0.35
	Y_3	0	0.19	1	-0.02	0	0.13	0.11

Tabel 4.10 Iterasi 3 Metode Simpleks

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 3	w	0	0	0	0.1	0.06	0.07	0.23
	Y_1	1	0	0	0.14	-0.07	0	0.07
	Y_2	0	1	0	-0.03	0.16	-0.07	0.06
	Y_3	0	0	1	-0.01	-0.03	0.14	0.1

Tabel 4.10 merupakan iterasi optimum dari Metode Simpleks, karena seluruh variabel nonbasis (S_1 , S_2 , S_3) pada baris fungsi tujuan sudah bernilai positif. Oleh karena itu, nilai-nilai dari A_i* ,B_j* dan v* dapat dihitung berdasarkan tabel 4.10, yaitu :

$$A_1^* = S_1 / w = 0.10 / 0.23 = 0.43$$

$$A_2^* = S_2 / w = 0.06 / 0.23 = 0.26$$

$$A_3^* = S_3 / w = 0.07 / 0.23 = 0.30$$

$B_1^* = Y_1 / w = 0.07 / 0.23 = 0.30$

$B_2^* = Y_2 / w = 0.06 / 0.23 = 0.26$

$B_3^* = Y_3 / w = 0.10 / 0.23 = 0.43$

$$
\begin{aligned}
V^* &= \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} \cdot A_i^* \cdot B_j^* \\
&= (3*0.43*0.30) + (-1*0.43*0.26) + (-3*0.43*0.43) + \\
&\quad (-3*0.26*0.30) + (3*0.26*0.26) + (-1*0.26*0.43) + \\
&\quad (-4*0.30*0.30) + (-3*0.30*0.26) + (3*0.30*0.43) \\
&= -0.64
\end{aligned}
$$

Dari hasil di atas mengandung pengertian bahwa Pemain A untuk mendapatkan hasil yang optimum harus menggunakan pencampuran terhadap strategi-strateginya, dengan komposisi sebagai berikut :

Probabalitas menggunakan strategi 1 sebesar 0.43

Probabalitas menggunakan strategi 2 sebesar 0.26

Probabalitas menggunakan strategi 3 sebesar 0.30

Sedangkan Pemain B untuk mendapatkan hasil yang optimum harus menggunakan pencampuran terhadap strategi-strateginya, dengan komposisi sebagai berikut :

Probabalitas menggunakan strategi 1 sebesar 0.30

Probabalitas menggunakan strategi 2 sebesar 0.26

Probabalitas menggunakan strategi 3 sebesar 0.43

Sedangkan Nilai Permainan yang diperoleh apabila kedua pemain menggunakan strategi optimumnya sebesar -0.64, yaitu berarti Pemain A akan menderita kerugian terkecil sebesar 0.64 apabila menggunakan strategi optimumnya, dan Pemain B mendapat keuntungan sebesar 0.64 apabila menggunakan strategi optimumnya. Permainan ini berlangsung tidak adil karena Nilai Permainan yang diperoleh tidak sama dengan nol.

Sedangkan contoh mengenai permainan yang berlangsung adil, dapat dilihat pada contoh berikut :

Matriks Pembayaran :

Tabel 4.11 Matriks Pembayaran untuk permainan yang adil

		Pemain B	
		B1	B2
Pemain A	A1	5	-3
	A2	-3	1.8

Dari tabel di atas dapat diketahui nilai-nilai minimum untuk setiap baris dan nilai-nilai maksimum untuk setiap kolom.

Nilai minimum :

Baris 1 : -3

Baris 2 : -3

Nilai maksimum :

Kolom 1 : 5

Kolom 2 : 1.8

Berdasarkan nilai-nilai minimum yang ada, kita dapat menghitung nilai maksiminnya yaitu nilai terbesar dari nilai-nilai minimum tersebut. Untuk persoalan ini nilai maksimin = -3.

Berdasarkan nilai-nilai maksimum yang ada, kita dapat menghitung nilai minimaksnya yaitu nilai terkecil dari nilai-nilai maksimum tersebut. Untuk persoalan ini nilai minimaks = 1.8.

Karena nilai maksimin tidak sama dengan nilai minimaks, maka persoalan tersebut memenuhi syarat untuk dapat disebut persoalan menggunakan strategi campuran. Karena nilai maksimin lebih kecil dari nol maka nilai dari permainan dapat bernilai negatif atau nol. Oleh karena itu perlu ditambahkan suatu konstanta terhadap setiap elemen pada matriks pembayaran. Besarnya nilai konstanta tidak akan mempengaruhi terhadap proses selanjutnya, dengan syarat konstanta tersebut harus bernilai minimal sama dengan nilai maksimin. Misalkan digunakan konstanta bernilai 5, maka matriks pembayaran berubah menjadi :

Tabel 4.12 Matriks Pembayaran setelah ditambah konstanta
untuk permainan yang adil

		Pemain B	
		B1	B2
Pemain A	A1	10	2
	A2	2	6.8

Kemudian akan dilakukan proses penyelesaian dengan Metode Simpleks seperti dibawah ini :

Tabel 4.13 Iterasi 0 Metode Simpleks untuk permainan yang adil

	Basis	Y_1	Y_2	S_1	S_2	Solusi
Iterasi 0	w	-1	-1	0	0	0
	S_1	10	2	1	0	1
	S_2	2	6.8	0	1	1

Tabel 4.14 Iterasi 1 Metode Simpleks untuk permainan yang adil

	Basis	Y_1	Y_2	S_1	S_2	Solusi
Iterasi 1	w	0	-0.8	0.1	0	0.1
	Y_1	1	0.2	0.1	0	0.1
	S_2	0	6.4	-0.2	1	0.8

Tabel 4.15 Iterasi 2 Metode Simpleks untuk permainan yang adil

	Basis	Y_1	Y_2	S_1	S_2	Solusi
Iterasi 2	w	0	0	0.07	0.12	0.2
	Y_1	1	0	0.11	-0.03	0.12
	Y_2	0	1	-0.03	0.16	0.075

Tabel 4.15 merupakan iterasi optimum dari Metode Simpleks, karena seluruh variabel nonbasis (S_1 , S_2) pada baris fungsi tujuan sudah bernilai positif. Oleh karena itu, nilai-nilai dari A_i^* ,B_j^* dan v^* dapat dihitung berdasarkan tabel 4.15, yaitu :

$A_1^* = S_1 / w = 0.075 / 0.2 = 0.375$

$A_2^* = S_2 / w = 0.125 / 0.2 = 0.625$

$B_1^* = Y_1 / w = 0.125 / 0.2 = 0.625$

$B_2^* = Y_2 / w = 0.075 / 0.2 = 0.375$

$$V^* = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} \cdot A_i^* \cdot B_j^*$$

$$= (5*0.375*0.625) + (-3*0.375*0.375) + (-3*0.625*0.625) + (1.8*0.625*0.375)$$

$$= 0$$

Dari hasil di atas mengandung pengertian bahwa Pemain A untuk mendapatkan hasil yang optimum harus menggunakan pencampuran terhadap strategi-strateginya, dengan komposisi sebagai berikut :

Probabalitas menggunakan strategi 1 sebesar 0.375

Probabalitas menggunakan strategi 2 sebesar 0.625

Sedangkan Pemain B untuk mendapatkan hasil yang optimum harus menggunakan pencampuran terhadap strategi-strateginya, dengan komposisi sebagai berikut :

Probabalitas menggunakan strategi 1 sebesar 0.375

Probabalitas menggunakan strategi 2 sebesar 0.625

Sedangkan Nilai Permainan yang diperoleh apabila kedua pemain menggunakan strategi optimumnya sebesar 0, yaitu berarti kedua pemain tidak mengalami kerugian dan keuntungan apabila strategi optimum telah dipergunakan oleh kedua pemain. Permainan ini berlangsung adil karena Nilai Permainan yang diperoleh sama dengan nol.

TUGAS AKHIR

BAB V

KONSTRUKSI PERANGKAT LUNAK

BAB V

KONSTRUKSI PERANGKAT LUNAK

Dalam Tugas Akhir ini, perangkat lunak yang dibuat terdiri dari 3 fungsi utama, yaitu fungsi Maksmin, fungsi Metode Simpleks, dan fungsi Nilai Permainan.

5.1 Fungsi Maksmin

Tahap pertama yang perlu dilakukan dalam Tugas Akhir ini adalah mencari nilai maksimin dan nilai minimaks. Hal ini dimaksudkan untuk mengetahui apakah persoalan yang dimaksud termasuk dalam ruang lingkup Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran atau tidak. Proses dalam fungsi Maksmin meliputi proses pencarian nilai maksimin dan proses pencarian nilai minimaks.

Data masukannya berupa Matriks Pembayaran. Bentuk matriks tersebut berupa :

$$\text{Old Matrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

dimana :

- a_{ij} menyatakan koefisien strategi Pemain A ke *i* bila bertemu dengan strategi pemain B ke *j*
- i = 1, 2, ... , m (m = jumlah strategi pemain A)
- j = 1, 2, ... , n (n = jumlah strategi pemain B)

Contoh :

Misalkan jumlah strategi Pemain A = 4, dan jumlah strategi Pemain B = 5, maka matriks yang dialokasikan untuk menyimpan data tersebut adalah 4x5.

5.1.1 Proses Pencarian Nilai Maksimin

Struktur data yang diperlukan dalam proses pencarian nilai maksimin adalah:

```
double  *TempMinRow
double   v_maximin
double   add_matrix
```

dimana :

- TempMinRow[i] menyatakan nilai minimum untuk setiap baris (strategi Pemain A)
- v_maximin menyatakan nilai maksimin

- add_matrix menyatakan konstanta yang perlu ditambahkan kepada setiap elemen OldMatrix apabila nilai maksimin yang didapatkan lebih kecil atau sama dengan nol.

Adapun langkah-langkah yang dilakukan dalam proses ini adalah :

```
for (i = 0; i < m; ++i) {
    TempMinRow[i] = OldMatrix[i][0];
     for (j = 1; j < n; ++j)
          if (TempMinRow[i] > OldMatrix[i][j])
               TempMinRow[i] = OldMatrix[i][j];
}
v_maximin = TempMinRow[0];
for (i = 1; i < m; ++i)
     if (v_maximin < TempMinRow[i])
          v_maximin = TempMinRow[i];
if (v_maximin > 0)
    add_matrix = 0;
else
     add_matriks = -v_maximin + 2;
```

5.1.1 Proses Pencarian Nilai Minimaks

Struktur data yang diperlukan dalam proses pencarian nilai minimaks adalah:

```
double *TempMaxCol
double  v_minimax
```

dimana :

- TempMaxCol[j] menyatakan nilai maksimum untuk setiap kolom (strategi Pemain B)
- v_minimax menyatakan nilai minimaks

Adapun langkah-langkah yang dilakukan dalam proses ini adalah :

```
for (j = 0; j < n; ++j) {
    TempMaxCol[j] = OldMatrix[0][j];
    for (i = 1; i < m; ++i)
        if (TempMaxCol[j] < OldMatrix[i][j])
            TempMaxCol[j] = OldMatrix[i][j];
}
v_minimax = TempMaxCol[0];
for (j = 1; j < n; ++j)
    if (v_minimax > TempMaxCol[j])
        v_minimax = TempMaxCol[j];
```

5.2 Fungsi Metode Simpleks

Proses dalam fungsi Metode Simpleks meliputi proses pembentukan tabel simpleks dan proses iterasi simpleks.

Data masukannya berupa Matriks Pembayaran yang telah diperbaharui dan diperluas. Yang dimaksud dengan diperbaharui adalah setiap elemen dari matriks tersebut telah ditambah dengan konstanta (add_matrix). Sedangkan yang dimaksud dengan diperluas adalah mengalami penambahan jumlah baris dan kolom. Jumlah baris yang ditambahkan sebanyak 1 baris, yang berfungsi untuk menghitung nilai dari fungsi tujuan. Sedanglan jumlah kolom yang ditambahkan sebanyak (m + 1) kolom, yang berfungsi untuk menghitung nilai dari variabel tambahan (slack) dan nilai dari solusi setiap baris.

Bentuk matriks tersebut selengkapnya adalah :

$$\text{NewMatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} & S_{11} & S_{12} & \cdots & S_{1m} & b_1 \\ A_{21} & A_{22} & \cdots & A_{2n} & S_{21} & S_{22} & \cdots & S_{2m} & b_2 \\ : & : & : & : & : & : & : & : & : \\ A_{m1} & A_{m2} & \cdots & A_{mn} & S_{m1} & S_{m2} & \cdots & S_{mm} & b_m \\ A_{01} & A_{02} & \cdots & A_{0n} & S_{01} & S_{02} & \cdots & S_{0m} & b_0 \end{bmatrix}$$

dimana :

- A_{ij} ($i = 1, 2, \ldots, m$ dan $j = 1, 2, \ldots, n$) menyatakan Matriks Pembayaran yang telah diperbaharui.
- A_{0j} ($j = 1, 2, \ldots, n$) meyatakan koefisien fungsi tujuan untuk Matriks Pembayaran yang telah diperbaharui.
- S_{ij} ($i = 1, 2, \ldots, m$ dan $j = 1, 2, \ldots, m$) menyatakan keofisien variabel tambahan untuk Matriks Pembayaran yang telah diperbaharui.
- S_{0j} ($j = 1, 2, \ldots, n$) meyatakan koefisien fungsi tujuan pada variabel tambahan untuk Matriks Pembayaran yang telah diperbaharui.
- b_i ($i = 1, 2, \ldots, m$) menyatakan keofisien solusi untuk Matriks Pembayaran yang telah diperbaharui.
- $i = 1, 2, \ldots, m$ (m = jumlah strategi pemain A)
- $j = 1, 2, \ldots, n$ (n = jumlah strategi pemain B)

Contoh :

Misalkan jumlah strategi Pemain A = 4, dan jumlah strategi Pemain B = 5, maka matriks yang dialokasikan untuk menyimpan data tersebut adalah $(4 + 1) \times (5 + 4 + 1) = 5 \times 10$.

5.2.1 Proses Pembentukan Tabel Simpleks

Untuk memudahkan dalam proses selanjutnya, data matriks dibagi menjadi beberapa bagian, yaitu :

- matriks (m x n+m) untuk menyimpan data matriks pembayaran yang diperluas
- matriks (m x 1) untuk menyimpan data variabel basis
- matriks (n x 1) untuk menyimpan data variabel nonbasis
- matriks (m+1 x n) untuk menyimpan data solusi
- matriks (1 x n+m) untuk menyimpan data fungsi tujuan

Sebagai inisialisasi, tabel simpleks diisi dengan data-data sebagai berikut :

$$\text{NewMatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} & 1 & 0 & \cdots & 0 & 1 \\ A_{21} & A_{22} & \cdots & A_{2n} & 0 & 1 & \cdots & 0 & 1 \\ : & : & : & : & : & : & : & : & : \\ A_{m1} & A_{m2} & \cdots & A_{mn} & 0 & 0 & \cdots & 1 & 1 \\ -1 & -1 & \cdots & -1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

dimana :

- A_{ij} ($i = 1, 2, \ldots, m$ dan $j = 1, 2, \ldots, n$) menyatakan Matriks Pembayaran yang telah diperbaharui.
- A_{0j} ($j = 1, 2, \ldots, n$) yang menyatakan koefisien fungsi tujuan untuk Matriks Pembayaran yang diperbaharui diinisialisasi dengan nilai -1.

- S_{ij} (i = 1, 2, ... ,m dan j = 1, 2, ... , m) yang menyatakan keofisien variabel tambahan untuk Matriks Pembayaran yang diperbaharui diinisialisasi dengan matriks identitas.
- S_{0j} (j = 1, 2, ... , n) meyatakan koefisien fungsi tujuan pada variabel tambahan untuk Matriks Pembayaran yang diperbaharui diinisialisasi dengan nilai 0.
- b_i (i = 1, 2, ... ,m) menyatakan keofisien solusi untuk Matriks Pembayaran yang diperbaharui diinisialisasi dengan nilai 1.
- i = 1, 2, ... , m (m = jumlah strategi pemain A)
- j = 1, 2, ... , n (n = jumlah strategi pemain B)

Sedangkan struktur data yang diperlukan dalam proses pembentukan tabel simpleks adalah :

```
struct v_nonbasis {     //data untuk variabel nonbasis
    double nilai;       //sebanyak n
    int    kolom;
};
struct v_basis {        //data untuk variabel nonbasis
    double nilai;       //sebanyak m
    int    baris;
};
struct v_rasio {        //data untuk menghitung rasio solusi
    double nilai;       // dengan koefisien EV
    int    baris;
};
```

Adapun langkah-langkah yang dilakukan dalam proses ini adalah :

```
for (i = 0; i < m+1; ++i)
    for (j = 0; j < m+n+1; ++j)
        if ((i < m) && (j < n))
            NewMatrix[i][j] = OldMatrix[i][j] +
        else if ((i < m) && (j >= n))
            if ((j == i+n) || (j == m+n))
                NewMatrix[i][j] = 1;
            else
                NewMatrix[i][j] = 0;
        else if ((i >= m) && (j < n))
                NewMatrix[i][j] = -1;
        else
                NewMatrix[i][j] = 0;
```

5.2.2 Proses Iterasi Simpleks

Setelah tabel simpleks selesai dibentuk, maka proses selanjutnya untuk Metode Simpleks adalah iterasi simpleks yang merupakan proses utama untuk penyelesaian optimum dari masalah Programa Linier.

Pada proses iterasi ini, setiap iterasi mempunyai tahapan-tahapan proses, yaitu proses pengujian nilai variabel nonbasis pada fungsi tujuan, proses pencarian variabel yang masuk basis, proses pencarian variabel yang meninggalkan basis, dan proses eliminasi Gauss-Jordan.

Tahap pertama yang dilaksanakan adalah proses pengujian nilai variabel nonbasis pada fungsi tujuan. Jika seluruh variabel nonbasis tersebut bernilai nonnegatif, maka iterasi berakhir untuk kemudian mencari Nilai Permainan. Sedangkan apabila terdapat minimal satu variabel nonbasis yang bernilai negatif, maka akan dilanjutkan dengan proses selanjutnya.

Adapun langkah-langkah yang diperlukan seperti yang terlihat di bawah ini :

```
for (j = 0; j < n; ++j)
    if (NBV[j].nilai < 0)
        counter_NBV ++;
if (counter_NBV == 0)
    Proses_Nilai_Permainan();
else
    Tahap_Selanjutnya();
```

Tahap selanjutnya yang dilaksanakan adalah proses pencarian variabel yang masuk basis untuk iterasi selanjutnya. Langkah-langkah yang dilakukan adalah mencari variabel nonbasis yang bernilai paling negatif. Variabel nonbasis ini pada iterasi selanjutnya akan berubah status menjadi variabel basis.

Adapun langkah-langkah yang diperlukan seperti yang terlihat di bawah ini :

```
counter  : jumlah NBV yang negatif (counter_NBV)
EV = NBV[0].nilai;
col_EV = NBV[0].kolom;
for (i = 1; i < counter; ++i)
    if (EV > NBV[i].nilai) {
        EV = NBV[i].nilai;
        col_EV = NBV[i].kolom;
    }
```

Tahap berikutnya yang dilaksanakan adalah proses pencarian variabel yang meninggalkan basis untuk iterasi selanjutnya. Langkah-langkah yang dilakukan adalah mencari rasio pada seluruh baris (kecuali baris fungsi tujuan) dari solusi dengan koefisien variabel yang masuk basis jika koefisien variabel yang masuk basis pada baris tersebut bernilai positif. Kemudian dari rasio-rasio tersebut dicari rasio positif yang terkecil. Variabel basis yang berkorespondensi dengan baris tempat rasio positif terkecil akan berbah status menjadi variabel nonbasis pada iterasi selanjutnya.

Adapun langkah-langkah yang diperlukan seperti yang terlihat di bawah ini :

```
col    : kolom tempat varibel yang masuk basis (col_EV)
counter = 0;
for (i = 0; i < m; ++i)
    if (A[i][col] > 0) {
        rasio[counter].nilai = b_i  / A[i][col];
        rasio[counter].baris = i;
        counter++;
    }
if (counter == 0)
    Error; // unbounded solution
else {
    row_LV = rasio[0].baris;
    temp = rasio[0].nilai;
    for (i = 1; i < counter; ++i)
        if (temp > rasio[i].nilai) {
            row_LV = rasio[i].baris;
            temp = rasio[i].nilai;
}       }
```

Tahap berikutnya yang dilaksanakan adalah proses eliminasi Gauss-Jordan yang bertujuan untuk membuat koefisien variabel masuk basis pada setiap baris bernilai 0 kecuali pada baris tempat variabel yang meninggalkan basis bernilai 1. Adapun langkah-langkah yang diperlukan seperti yang terlihat di bawah ini :

```
row_pivot : baris tempat variabel yang meninggalkan basis (row_LV)
col_pivot  : kolom tempat varibel yang masuk basis (col_EV)
pivot      : elemen pivot ( A[row_LV][col_EV]
for (i = 0; i < m+1; ++i)
    for (j = 0; j < n+m+1; ++j)
        if ( i == row_pivot)
            Temp[i][j] = A[i][j] / pivot;
        else
            Temp[i][j] = A[i][j] - (A[i][col_EV] / pivot)*A[row_LV][j];
for (i = 0; i < m+1; ++i)
    for (j = 0; j < n+m+1; ++j)
        A[i][j] = Temp[i][j];
```

5.3 Fungsi Nilai Permainan

Proses dalam fungsi Nilai Permainan meliputi proses pencarian nilai strategi optimum Pemain A, proses pencarian nilai strategi optimum Pemain B , dan proses pencarian Nilai Permainan.

Data masukannya berupa matriks, yang elemen-elemennya merupakan hasil dari iterasi optimum Metode Simpleks dan variabel-variabel basis serta nonbasis. Bentuk matriks tersebut berupa :

$$
\text{MatrixOpt} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} & A_{1n+1} & A_{1n+2} & \cdots & A_{1n+m} & b_1 \\ A_{21} & A_{22} & \cdots & A_{2n} & A_{2n+1} & A_{2n+2} & \cdots & A_{2n+m} & b_2 \\ : & : & : & : & : & : & : & : & : \\ A_{m1} & A_{m2} & \cdots & A_{mn} & A_{mn+1} & A_{mn+2} & \cdots & A_{mn+} & b_m \\ A_{01} & A_{02} & \cdots & A_{0n} & A_{0n+1} & A_{0n+2} & \cdots & A_{0n+m} & b_{m+1} \end{bmatrix}
$$

dimana :

- A_{ij} (i = 1, 2, ... ,m dan j = 1, 2, ... , n+m) menyatakan matriks yang diperoleh dari iterasi optimum Metode Simpleks.
- A_{0j} (j = 1, 2, ... , n+m) meyatakan koefisien fungsi tujuan untuk matriks yang diperoleh dari iterasi optimum Metode Simpleks.
- b_i (i = 1, 2, ... ,m+1) menyatakan keofisien solusi untuk matriks yang diperoleh dari iterasi optimum Metode Simpleks.
- m = jumlah strategi pemain A
- n = jumlah strategi pemain B

Dalam penentuan strategi optimum untuk Pemain A dan Pemain B, akan mengikuti aturan sebagai berikut :

1. Jika A_i (i = 1, 2, ... , m) pada iterasi terakhir berupa variabel nonbasis maka A_i berkorespondensi dengan elemen pada baris fungsi tujuan di kolom variabel tambahan ke i (MatrixOpt [m+1][n+i]).

2. Jika A_i (i = 1, 2, ... , m) pada iterasi terakhir berupa variabel basis maka A_i tidak memiliki pengaruh dalam permainan tersebut, dengan kata lain nilai dari $A_i = 0$.
3. Jika B_j (j = 1, 2, ... , n) pada iterasi terakhir berupa variabel basis maka B_i berkorespondensi dengan elemen pada baris tempat variabel basis tersebut berada di kolom solusi (MatrixOpt [j][n+m+1]).
4. Jika B_j (j = 1, 2, ... , n) pada iterasi terakhir berupa variabel nonbasis maka B_j tidak memiliki pengaruh dalam permainan tersebut, dengan kata lain nilai dari $B_j = 0$.

Sedangkan strategi-strategi optimum dari Pemain A dan Pemain B memenuhi ketentuan :

$A_i{}^* = A_i$ / MatrixOpt[m+1][n+m+1]

$B_j{}^* = B_j$ / MatrixOpt[m+1][n+m+1]

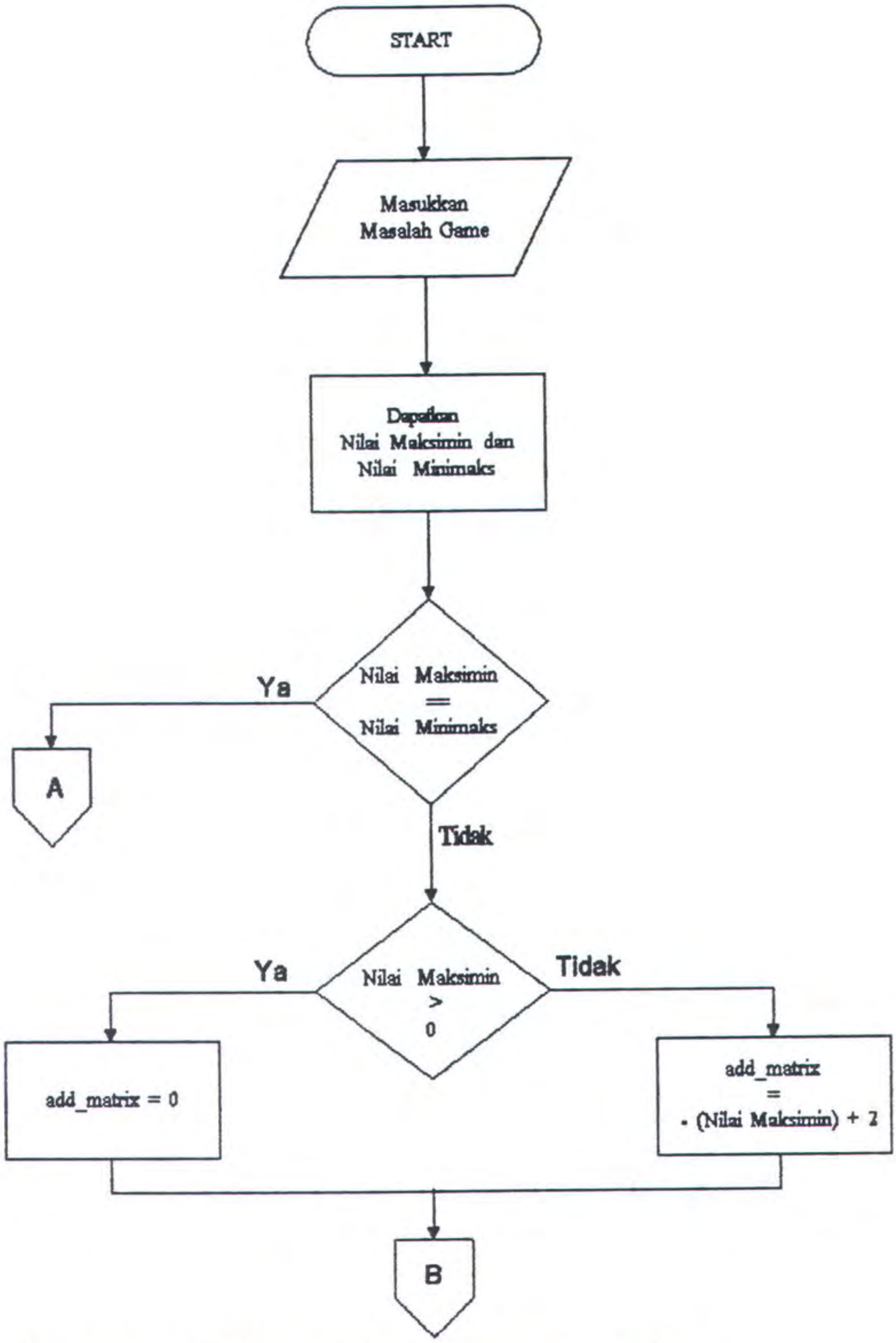
Untuk menghitung Nilai Permainan diperlukan data Matriks Pembayaran yang pertama, yaitu matriks yang tidak diperbaharui dan diperluas.

Adapun proses perhitungan untuk Nilai Permainan adalah :

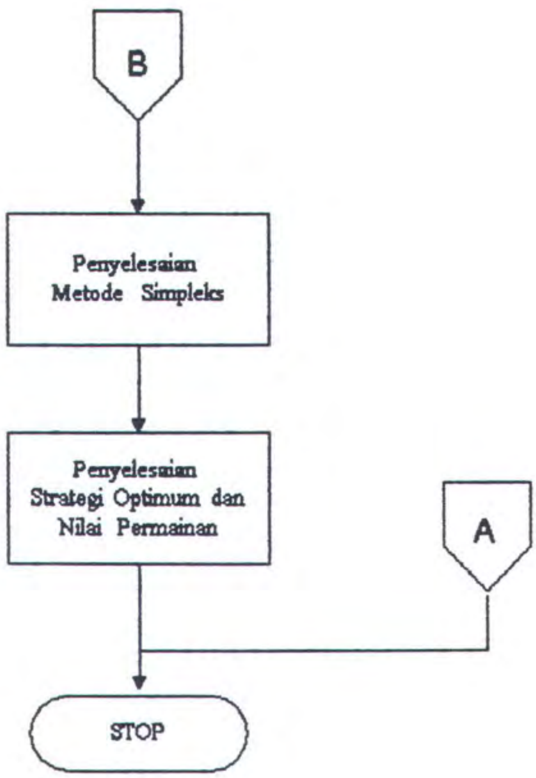
$$V^* = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} \cdot A_i{}^* \cdot B_j{}^*$$

Adapun langkah-langkah yang diperlukan seperti yang terlihat di bawah ini :

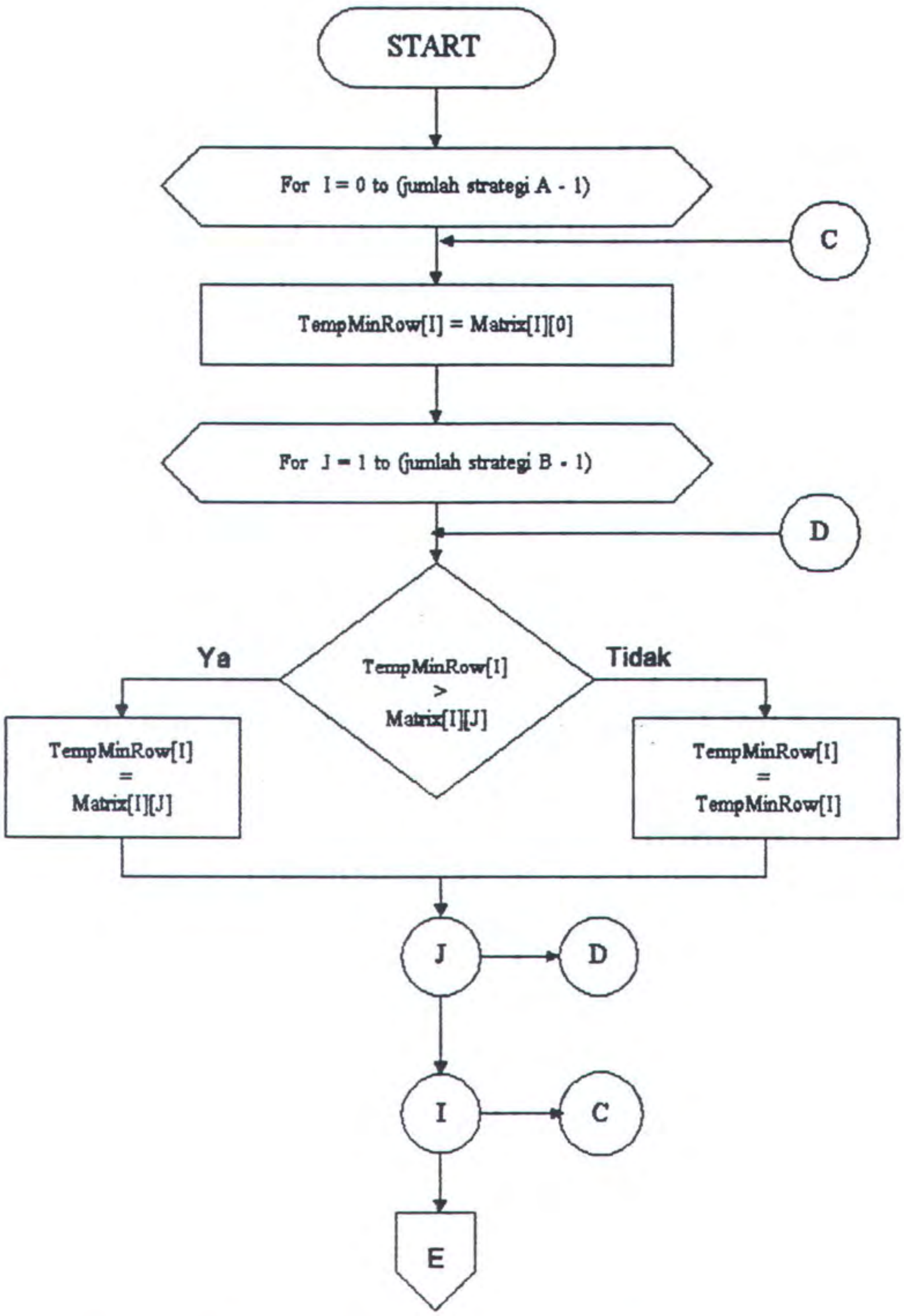
```
for (i = 0; i < m; ++i)          //Untuk mencari nilai strategi optimum
    if ( Ai berstatus 'basis')    // Pemain A
        Ai * = MatrixOpt [m+1][n+i] / MatrixOpt[m+1][n+m+1];
    else
        Ai * = 0;
for (j = 0; j < n; ++j)          //Untuk mencari nilai strategi optimum for
    if ( Bj berstatus 'nonbasis')    // Pemain B
      Bj * = MatrixOpt [j][n+m+1] / MatrixOpt[m+1][n+m+1];
    else
      Bj * =  0;
for (i = 0; i < m; ++i)          //Untuk mencari Nilai Permainan optimum
    for (j = 0; j < n; ++j)
       V* += OldMatrix[i][j] . Ai * . Bj *
```



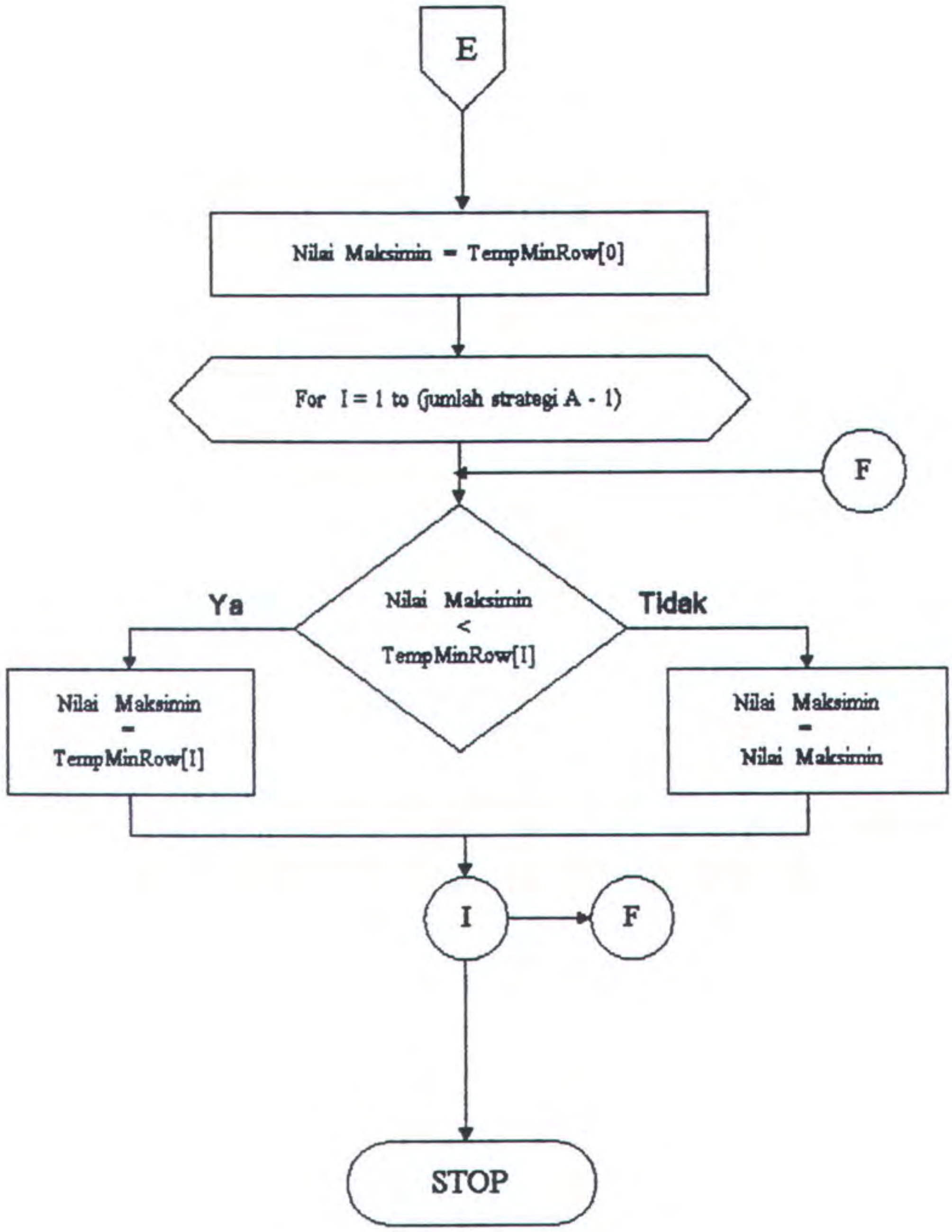
Gambar 5.1 Flow Chart Penyelesaian Persoalan Permainan



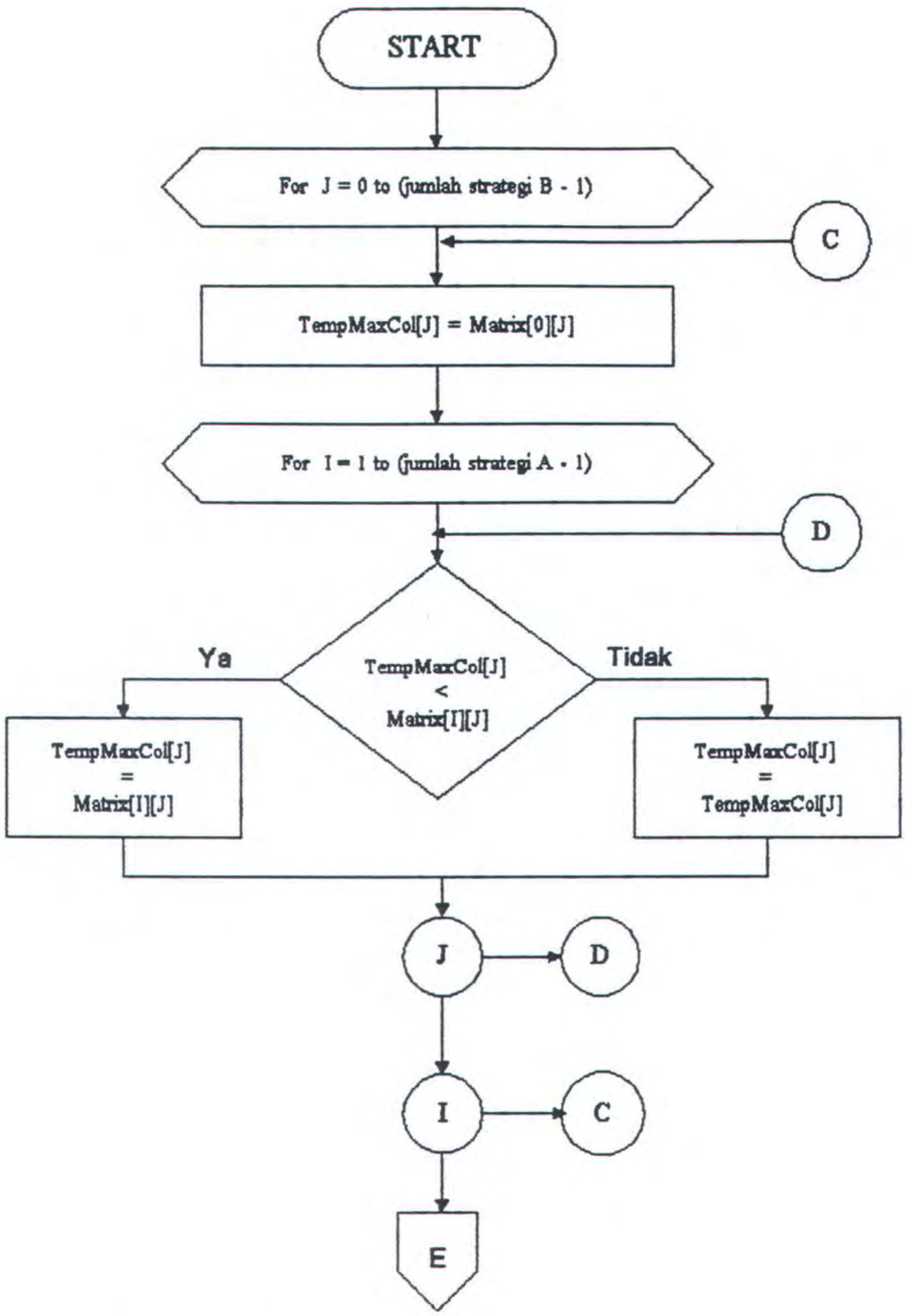
Gambar 5.2 Flow Chart Penyelesaian Persoalan Permainan (lanjutan)



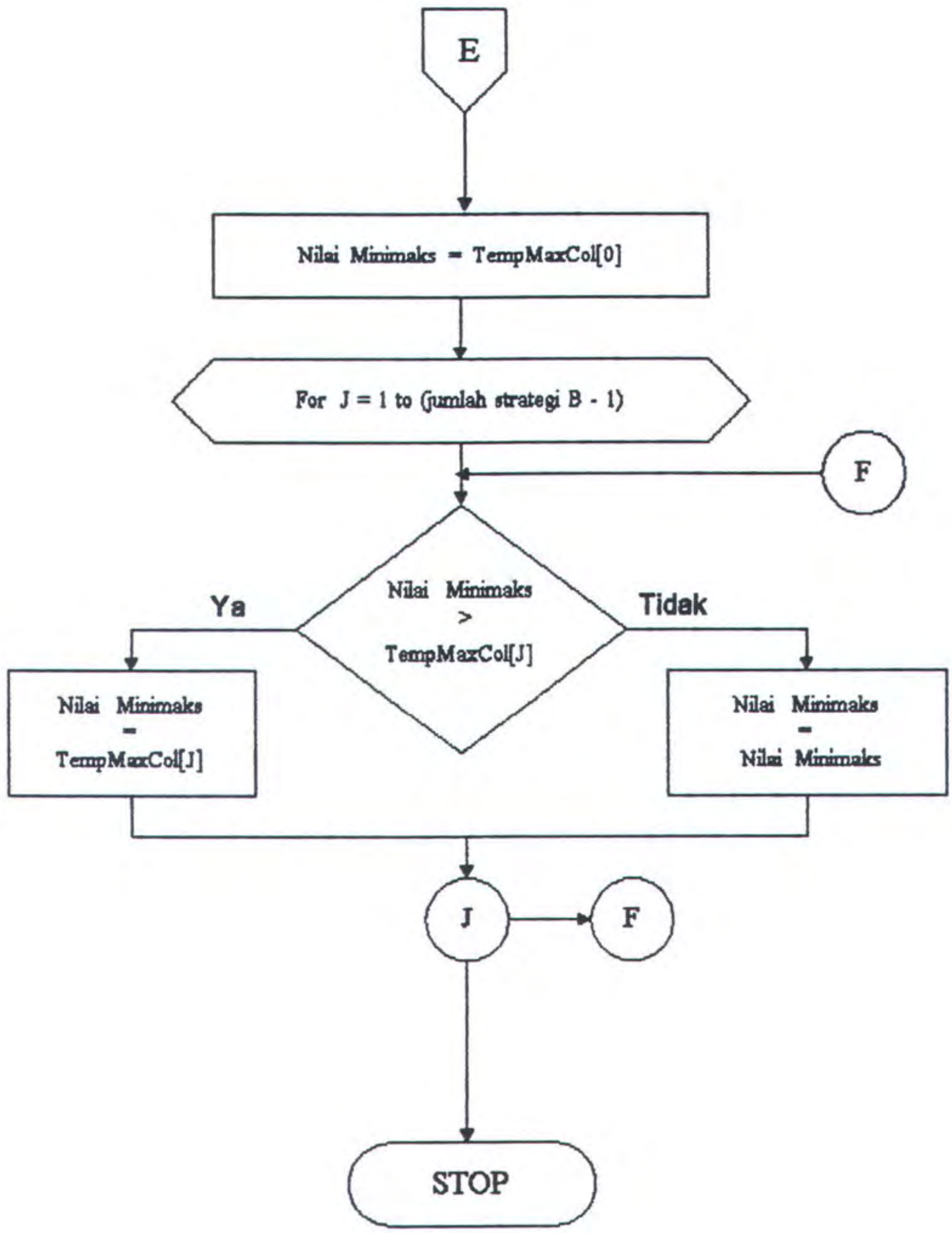
Gambar 5.3 Flow Chart Penyelesaian Nilai Maksimin



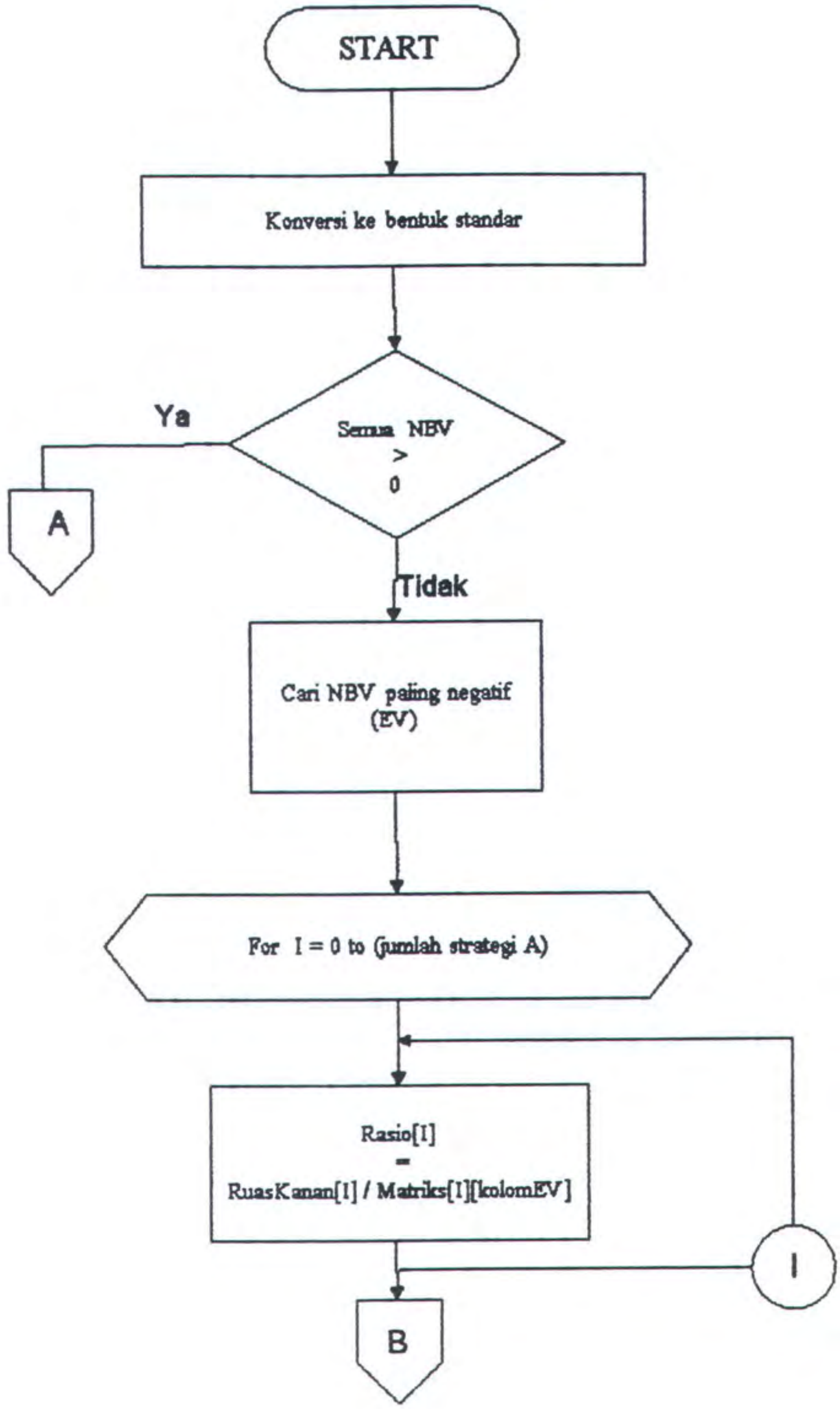
Gambar 5.4 Flow Chart Penyelesaian Nilai Maksimin (lanjutan)



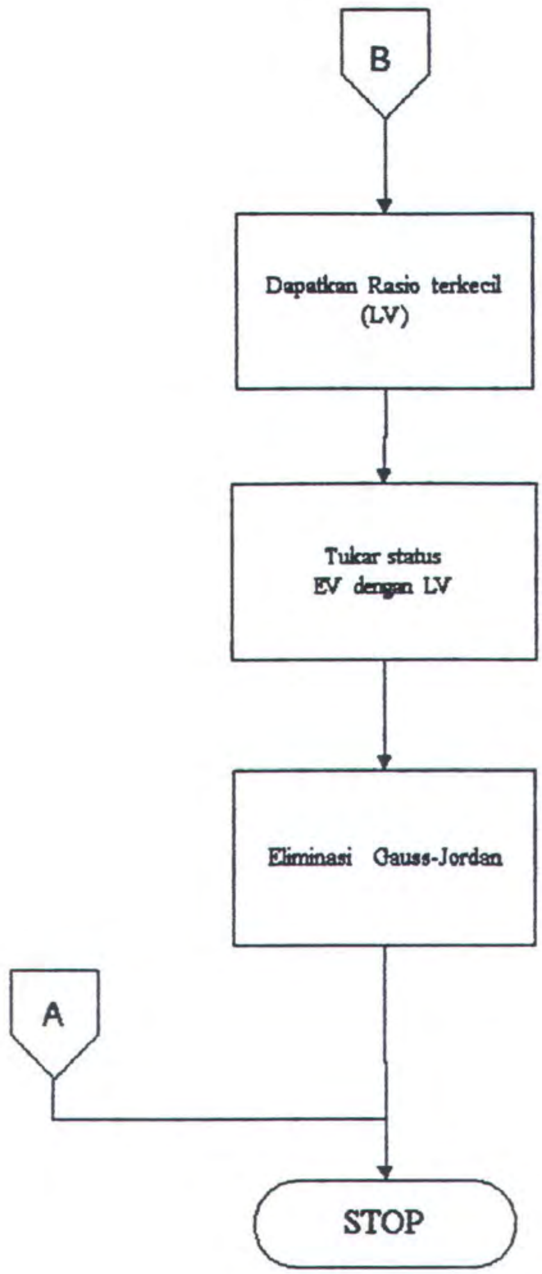
Gambar 5.5 Flow Chart Penyelesaian Nilai Minimaks



Gambar 5.6 Flow Chart Penyelesaian Nilai Minimaks (lanjutan)



Gambar 5.7 Flow Chart Penyelesaian Metode Simpleks



Gambar 5.8 Flow Chart Penyelesaian Metode Simpleks (lanjutan)

TUGAS AKHIR

BAB VI

EVALUASI PERANGKAT LUNAK

BAB VI

EVALUASI PERANGKAT LUNAK

Pada bab ini akan diuraikan mengenai hasil evaluasi perangkat lunak yang dibuat. Evaluasi dilakukan berdasarkan faktor yang menimbulkan kelambatan dan kesulitan dalam menentukan penyelesaian dari masalah Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran .

Masalah Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran akan menjadi sulit bila jumlah strategi Pemain A lebih kecil daripada jumlah strategi Pemain B. Selain itu karena digunakan perhitungan dengan bilangan rasional, maka kesalahan akibat pembulatan nilai suatu bilangan sangat mungkin terjadi. Dalam perhitungan dengan menggunakan komputer, apabila suatu bilangan bernilai sangat kecil (kurang dari 10e-12), bilangan tersebut akan menimbulkan suatu kesulitan jika digunakan sebagai bilangan pembagi. Sedangkan waktu penyelesaiannya akan menjadi lama apabila jumlah strategi kedua pemain besar dan jumlah iterasi yang diperlukan sama dengan jumlah iterasi maksimum penyelesaian Metode Simpleks.

6.1 Faktor Jumlah Strategi

Sesuai dengan bentuk Programa Linier yang menyatakan bahwa m ≥ n, maka apabila jumlah strategi Pemain A lebih kecil dibandingkan dengan jumlah strategi Pemain B akan menimbulkan kesulitan dalam penyelesaiannya. Hal ini disebabkan jumlah variabel basis dalam suatu Programa Linier harus lebih besar atau sama dengan jumlah variabel nonbasis.

Apabila jumlah variabel basis lebih kecil, maka satu hal yang sudah pasti yaitu tidak semua variabel basis mempunyai nilai. Karena dalam penyelesaian suatu persoalan Programa Linier, yang diharapkan adalah nilai dari variabel-variabel keputusan (pada iterasi terakhir diharapkan semuanya berbentuk variabel basis), maka hal ini akan menimbulkan suatu kesalahan. Kesalahan ini terjadi apabila kenaikan atau penurunan fungsi tujuan dikendalikan oleh variabel yang bukan termasuk variabel basis, yang apabila formulasi persoalannya sudah benar (jumlah strategi Pemain A lebih besar atau sama dengan jumlah strategi Peman B), hal tersebut tidak akan terjadi.

Penyelesaian dari masalah ini adalah dengan membalik kedua strategi, sehingga Pemain B akan menempati posisi Pemain A, sedangkan Pemain A akan menempati posisi Pemain B.

Contoh :

Jumlah strategi Pemain A = 4

Jumlah strategi Pemain B = 6

Tabel 6.1 Permainan yang strateginya terbalik

		Pemain B					
		B1	B2	B3	B4	B5	B6
Pemain A	A1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}
	A2	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}
	A3	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}
	A4	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}

Apabila persoalan tersebut diselesaikan, jumlah variabel basis yang ada hanya berjumlah 4, sedangkan perubahan nilai dari fungsi tujuan (sebaiknya) dipengaruhi oleh 6 buah variabel (variabel B1 , B2, ... , B6). Sehingga nilai fungsi tujuan yang diperoleh belum optimum, karena hanya didapatkan nilai dari 4 buah variabel.

Sebagai penyelesaiannya, kedua strategi harus dibalik, menjadi :

Jumlah strategi Pemain A = 6

Jumlah strategi Pemain B = 4

Tabel 6.2 Permainan yang strateginya sudah diperbaiki

		Pemain A			
		A1	A2	A3	A4
Pemain B	B1	$-a_{11}$	$-a_{21}$	$-a_{31}$	$-a_{41}$
	B2	$-a_{12}$	$-a_{22}$	$-a_{32}$	$-a_{42}$
	B3	$-a_{13}$	$-a_{23}$	$-a_{33}$	$-a_{43}$
	B4	$-a_{14}$	$-a_{24}$	$-a_{34}$	$-a_{44}$
	B5	$-a_{15}$	$-a_{25}$	$-a_{35}$	$-a_{45}$
	B6	$-a_{16}$	$-a_{26}$	$-a_{36}$	$-a_{46}$

Yang dibalik diatas bukan strategi kedua pemain, melainkan posisi kedua pemain, yaitu yang menempati tempat dalam matrik untuk Pemain A adalah Pemain B sedangkan tempat untuk Pemain B dalam matriks oleh Pemain A, selain itu koefisien a_{ij} berlawanan dengan a_{ij} semula, sehingga penyelesaian yang optimum tetap untuk kedua pemain tersebut.

6.2 Faktor Jumlah Iterasi

Dalam Perangkat lunak yang dibuat, kecepatan proses selain ditentukan oleh faktor luar (jenis mikroprosesor, besarnya memori), juga ditentukan oleh banyaknya iterasi yang diperlukan untuk mencapai solusi optimum.

Sedangkan iterasi itu sendiri dipengaruhi oleh banyaknya strategi setiap pemain dan komposisi dari strategi-strategi itu sendiri. Jika jumlah strategi suatu pemain sebanyak 4, tentu akan berlainan waktu prosesnya apabila jumlah strateginya 5, karena dibutuhkan waktu tambahan untuk menghitung kelebihan strategi tersebut. Sedangkan yang dimaksud dengan komposisi dari strategi-strategi yaitu apabila koefisien dari suatu variabel yang masuk basis berbeda nilainya. Misalnya, jika variabel yang masuk basis terletak pada kolom 2. Maka koefisien lainnya pada kolom 2 tersebut akan mempengaruhi waktu proses dalam perhitungan rasio. Apabila hanya terdapat 1 koefisien yang bernilai positif, tentu akan berlainan dengan apabila mempunyai 2 koefisien yang bernilai positif.

Contoh :

Dalam suatu iterasi didapatkan nilai sebagai berikut :

Tabel 6.3 Contoh 1 untuk faktor jumlah iterasi

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 2	w	0	-0.5	0	0.1	0.06	0.07	0.23
	Y_1	1	-1	0	0.14	-0.07	0	0.07
	S_2	0	1	0	-0.03	0.16	-0.07	0.06
	Y_3	0	2	1	-0.01	-0.03	0.14	0.1

Pada tabel di atas variabel yang masuk basis adalah variabel Y_2. Untuk menghitung rasio setiap baris, diperlukan waktu proses sebesar 2 kali perhitungan yaitu untuk baris S_2 dan Y_3. Waktu proses akan berbeda untuk kejadian seperti di bawah ini, dimana waktu yang diperlukan hanya untuk menghitung 1 kali yaitu untuk baris S_2.

Tabel 6.4 Contoh 2 untuk faktor jumlah iterasi

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 2	w	0	-0.5	0	0.1	0.06	0.07	0.23
	Y_1	1	-1	0	0.14	-0.07	0	0.07
	S_2	0	1	0	-0.03	0.16	-0.07	0.06
	Y_3	0	-2	1	-0.01	-0.03	0.14	0.1

Banyaknya iterasi yang tejadi paling sedikit 1 iterasi, sedangkan iterasi maksimum berjumlah ${}^{n}C_{m}$ = n! / (n - m)! m! yaitu kombinasi variabel basis yang mungkin dari seluruh variabel yang ada.

Iterasi tidak mungkin tidak terjadi (jika persoalan sudah memenuhi syarat) karena seluruh variabel nonbasis diinisialisasi dengan nilai negatif (-1). Sehingga iterasi minimal terjadi 1 kali. Sedangkan iterasi maksimum terjadi apabila kombinasi dari seluruh variabel (X_i dan Y_j) untuk membentuk variabel basis terjadi.

Contoh :

Jumlah strategi Pemain A = 2

Jumlah strategi Pemain B = 2

Iterasi minimum terjadi apabila setelah iterasi pertama, seluruh variabel nonbasis bernilai nonnegatif.

Iterasi maksimum terjadi sebanyak 6 kali, yaitu :

1. Iterasi pertama, yang menjadi variabel basis adalah variabel x_1 dan x_2
2. Iterasi kedua sampai iterasi keenam, yang menjadi variabel basis adalah x_1 dan y_1 , x_1 dan y_2 , x_2 dan y_1 , x_2 dan y_2 , y_1 dan y_2.

6.3 Faktor Pembulatan Nilai

Pada contoh penyelesaian Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran dalam Tabel 4.10 didapatkan hasil sebagai berikut :

$A_1^* = S_1 / w = 0.10 / 0.23 = 0.43$

$A_2^* = S_2 / w = 0.06 / 0.23 = 0.26$

$A_3^* = S_3 / w = 0.07 / 0.23 = 0.30$

$B_1^* = Y_1 / w = 0.07 / 0.23 = 0.30$

$B_2^* = Y_2 / w = 0.06 / 0.23 = 0.26$

$B_3^* = Y_3 / w = 0.10 / 0.23 = 0.43$

Ternyata $A_1 + A_2 + A_3$ tidak sama dengan 1. Begitu juga dengan $B_1 + B_2 + B_3$ tidak sama dengan 1. Hasil yang diperoleh tidak memenuhi syarat untuk pernyelesaian persoalan ini. Kesalahan yang terjadi ini disebabkan karena adanya pembulatan nilai pada bilangan rasional. Tetapi karena pembulatan dilakukan pada digit yang ke 16 (tipe variabel double), sebenarnya hal ini tidak terlalu mutlak untuk dapat disebut suatu kesalahan, karena kesalahan yang terjadi relatif kecil (untuk persoalan ini kesalahan hanya sebesar 1%).

Apabila dilakukan perhitungan dengan menggunakan bilangan pecahan, proses perhitungan dengan Metode Simpleks akan seperti berikut :

Tabel 6.5 Iterasi 0 menggunakan bilangan pecahan

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 0	w	-1	-1	-1	0	0	0	0
	S_1	8	4	2	1	0	0	1
	S_2	2	8	4	0	1	0	1
	S_3	1	2	8	0	0	1	1

Tabel 6.6 Iterasi 1 menggunakan bilangan pecahan

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 1	w	0	-1/2	-3/4	1/8	0	0	1/8
	Y_1	1	1/2	1/4	1/8	0	0	1/8
	S_2	0	7	7/2	-1/8	1	0	3/4
	S_3	0	3/2	31/4	-1/8	0	1	7/8

Tabel 6.7 Iterasi 2 menggunakan bilangan pecahan

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 2	w	0	-7/20	0	9/80	0	1/10	17/80
	Y_1	1	9/20	0	29/240	0	-1/30	23/240
	S_2	0	63/10	0	-23/120	1	-28/60	41/120
	Y_3	0	1/5	1	-1/60	0	8/60	7/60

Tabel 6.8 Iterasi 3 menggunakan bilangan pecahan

	Basis	Y_1	Y_2	Y_3	S_1	S_2	S_3	Solusi
Iterasi 3	w	0	0	0	5/49	11/196	1/14	45/196
	Y_1	1	0	0	1/7	-1/14	0	1/14
	Y_2	0	1	0	-3/98	31/196	-1/14	11/196
	Y_3	0	0	1	-1/98	-3/98	1/7	5/49

Sehingga hasil yang didapat adalah :

$A_1^* = S_1 / w = (5/49) / (45/196) = 20/45$

$A_2^* = S_2 / w = (11/196) / (45/196) = 11/45$

$A_3^* = S_3 / w = (1/14) / (45/196) = 14/45$

$B_1^* = Y_1 / w = (1/14) / (45/196) = 14/45$

$B_2^* = Y_2 / w = (11/196) / (45/196) = 11/45$

$B_3^* = Y_3 / w = (5/49) / (45/196) = 20/45$

Ternyata $A_1 + A_2 + A_3 = 1$. Begitu juga dengan $B_1 + B_2 + B_3 = 1$. Sehingga hasil yang didapat sesuai dengan syarat optimum yaitu $\sum_{i=1}^{m} x_i = 1$.

6.4 Faktor Pembagi Kecil

Dalam perhitungan dengan menggunakan komputer, bilangan yang bernilai sangat kecil (kurang dari 10e-12) akan menimbulkan suatu kesalahan apabila bilangan tersebut digunakan sebagai bilangan pembagi. Dalam perhitungan dengan menggunakan Metode Simpleks, terdapat 2 kejadian yang memungkinkan terjadinya kesalahan tersebut, yaitu :

1. Pembagian ruas kanan dengan koefisien pada kolom variabel yang masuk basis.
2. Pembagian koefisien pada kolom variabel yang masuk basis dengan koefisien variabel yang masuk basis (dalam Metode Eliminasi Gauss-Jordan).

Selain daripada itu, dalam perhitungan nilai strategi optimum setiap pemain, terdapat juga perhitungan yang memasukkan operasi pembagian antara strategi optimum dari iterasi simpleks dengan koefisien solusi optimum.

Untuk mencegah kejadian seperti ini, dalam perangkat lunak yang dibuat, ditambahkan pernyataan sebagai berikut :

1. Jika bilangan pembagi bernilai sangat kecil (kurang dari 10e-12), bilangan tersebut dinaikkan nilainya menjadi 10e-12. Hasil pembagian yang didapatkan, dikalikan lagi dengan bilangan sebesar kenaikan yang diperlukan. Hal ini dimaksudkan agar perhitungan yang didapat tetap optimum.
2. Jika bilangan pembagi bernilai lebih besar atau sama dengan 10e-12, bilangan tersebut akan diperlakukan seperti apa adanya.

Setelah pernyataan diatas ditambahkan, kesalahan akibat pembagian dengan bilangan yang sangat kecil akan dapat dihindarkan.

TUGAS AKHIR

BAB VII

PENUTUP

BAB VII

PENUTUP

7.1 KESIMPULAN

Berdasarkan hasil proses terhadap berbagai bentuk masalah Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran, maka dapat ditarik kesimpulan antara lain :

1. Yang mempengaruhi waktu penyelesaian, selain jumlah strategi dari kedua pemain adalah jumlah iterasi yang diperlukan untuk mencapai solusi optimum.
2. Sebaiknya jumlah strategi Pemain A lebih besar atau sama dengan jumlah strategi Pemain B, karena penyelesaian optimum yang didapat apabila jumlah strategi Pemain A lebih kecil dari jumlah strategi Pemain B belum menjamin keoptimuman dari persoalan, sehingga kedua strategi harus dibalik.
3. Pesan-pesan kesalahan yang didapat dalam menjalankan perangkat lunak, diperoleh karena kesalahan dalam memasukkan strategi setiap pemain, seperti :

   a. Nilai maksimin sama dengan nilai minimaks

      Kesalahan yang diakibatkan nilai maksimin sama dengan nilai minimaks, sehingga persoalan tidak sesuai dengan persoalan untuk Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran.

b. Strategi tidak lengkap

Kesalahan yang diakibatkan solusi yang didapat dalam Metode Simpleks tidak terbatas.

4. Permainan yang adil akan didapatkan apabila matriks segitiga atas bersesuaian dengan matriks segitiga bawah dalam matriks pembayaran, dan diagonalnya membentuk koefisien-koefisien yang simetris.

5. Solusi optimum yang diperoleh dapat digunakan sebagai bahan pertimbangan untuk memilih strategi mana yang akan dipilih. Strategi yang akan dipilih tergantung cara berpikir dan bertindak dari pemain yang bersangkutan.

7.2 SARAN

Sebagai bahan pertimbangan, saran-saran yang dapat diberikan bagi para pemakai :

1. Perangkat lunak yang dibuat, dapat dikembangkan menjadi perangkat lunak untuk menyelesaikan persoalan permainan yang lainnya, seperti Permainan Dua Pemain Dengan Jumlah Tidak Nol, dengan beberapa perubahan yang dianggap perlu.
2. Untuk mengefisienkan penggunaan memori, Metode Eliminasi Gauss-Jordan dapat diganti menjadi Metode Dekomposisi LU (*Lower-Upper*).
3. Data untuk masalah Permainan Dua Pemain Dengan Jumlah Nol Menggunakan Strategi Campuran ini menggunakan data tipe real (double), sehingga kemungkinan kesalahan akibat pembulatan angka sangat mungkin terjadi, untuk itu perlu perubahan terhadap data masukannya. Misalnya, dengan menggunakan data yang bertipe bulat, dan setiap bilangan real tersebut direpresentasikan dalam bentuk *record* yang terdiri dari dua field yaitu field untuk menyimpan pembilang dan penyebut dari bilangan real tersebut.

TUGAS AKHIR

DAFTAR PUSTAKA

DAFTAR PUSTAKA

1. Bazaraa, M., and J. Jarvis, ***Linear Programming and Network Flows***, Wiley, New York, 1977

2. Luce, R., and H. Raiffa, ***Games and Decisions***, Wiley, New York, 1957

3. Murty, Katta, ***Linear Programming***, Wiley, New York, 1983

4. Siagian, P., ***Penelitian Operasional Teori dan Praktek***, UI-PRESS, Jakarta, 1987

5. Sullivan, S. Robert., and Yih Long Chang, ***Quantitative Systems for Business Plus***, New Jersey : Prentice-Hall, International Edition, 1989

6. Taha, Hamdy A., ***Operation Research : An Introduction***, fourth edition, New York : MacMillan Publishing Co. Inc., 1987

7. Tjutju Tarliah Dimyati, Ahmad Dimyati., ***Operation Research: Model-model Pengambilan Keputusan***, edisi ke-3; Sinar Baru Algensindo PT., Bandung, 1994

8. Williams, J., ***The Compleat Strategyst***, rev. ed., McGraw-Hill, New York, 1966

TUGAS AKHIR

LAMPIRAN

LAMPIRAN

CARA PENGGUNAAN PROGRAM

Pada saat program dijalankan, tampilan pertamanya adalah seperti yang terlihat pada gambar 1. Pada gambar tersebut terlihat bahwa program terdiri dari 3 menu utama, yaitu File, Proses, dan About.

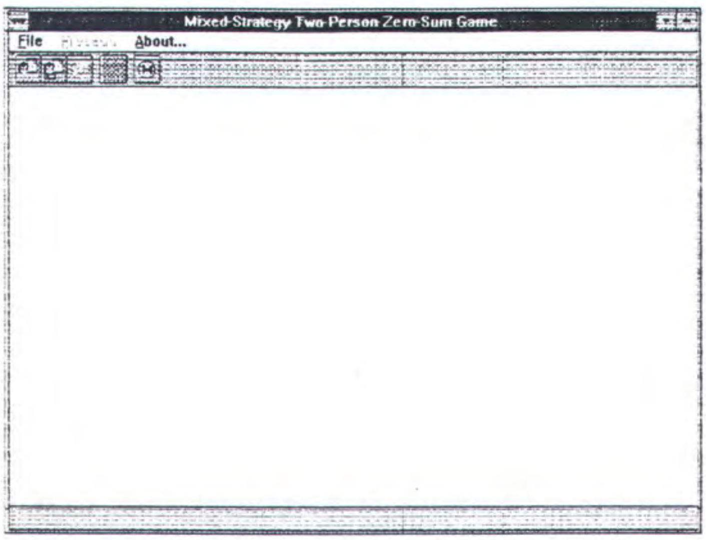
Dalam menu File, terdapat beberapa submenu sebagai berikut :

1. New : digunakan untuk memasukkan suatu data masalah permainan yang baru.
2. Open : digunakan untuk mengambil suatu data masalah permainan dari disk yang berekstension *.SAE.
3. Save : digunakan untuk menyimpan data masalah permainan ke disk, jika data masalah permainan belum terbentuk, maka submenu ini dinonaktifkan.
4. Save As : digunakan untuk menyimpan data masalah permainan ke disk dengan menggunakan nama file lain, jika data masalah permainan belum terbentuk, maka submenu ini dinonaktifkan.
5. Exit : digunakan untuk keluar dari program ini.

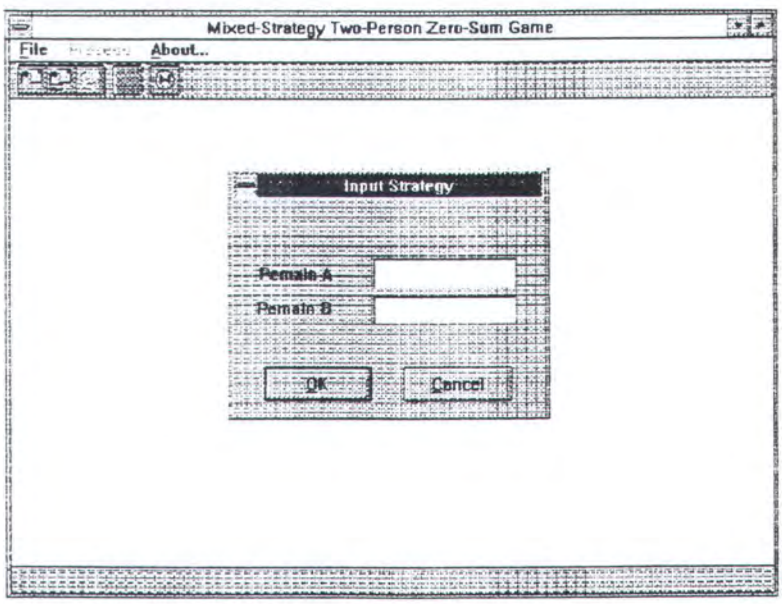
Adapun informasi yang ditampilkan setelah memilih submenu diatas, dapat dilihat pada gambar 2 dan gambar 3 untuk submenu New, gambar 4 untuk submenu Open.

Untuk menu Proses, yang dilakukan adalah melakukan penyelesaian dari data masalah permainan yang tersaji di editor. Jika suatu pemrosesan berhasil dengan baik, maka akan dicetak suatu hasil seperti pada gambar 5.

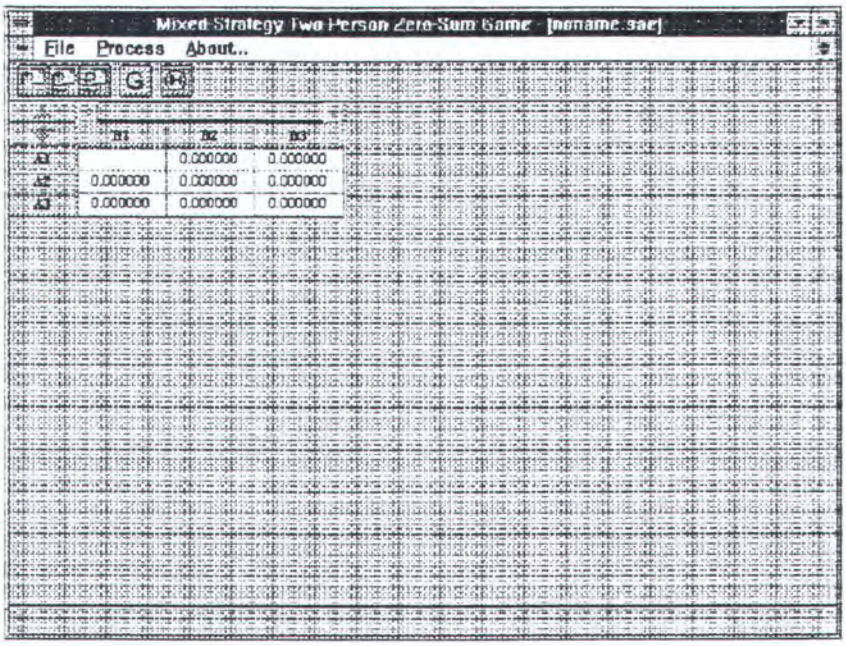
Untuk menu About, yang dilakukan adalah menampilkan suatu kotak dialog mengenai penyusunan program ini, seperti yang terlihat pada gambar 6.



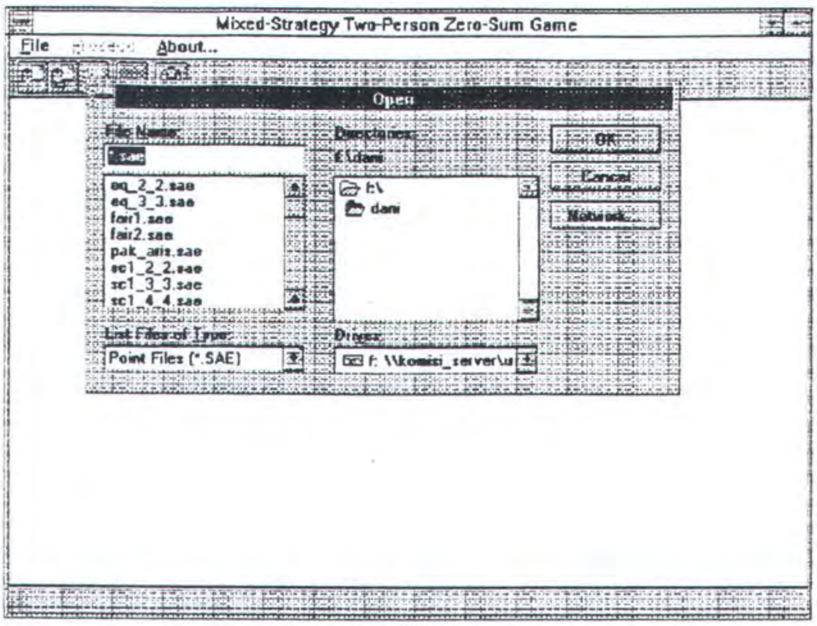
Gambar 1: Tampilan awal dari program



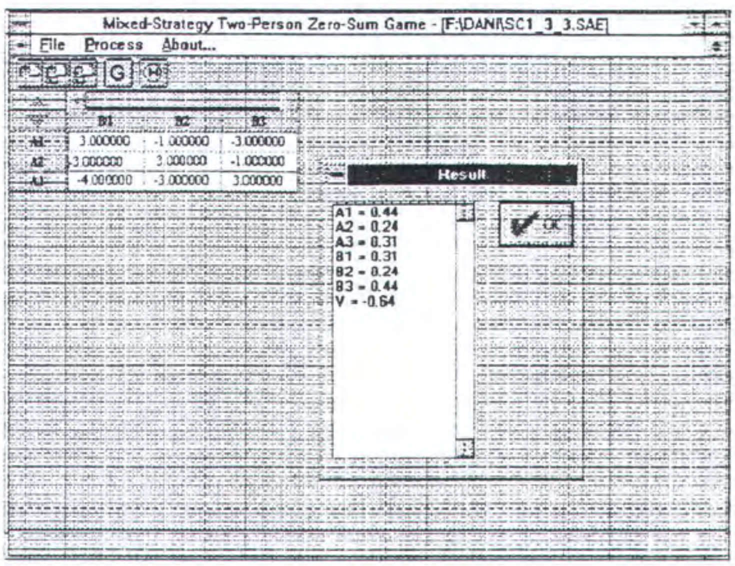
Gambar 2: Tampilan setelah menu New



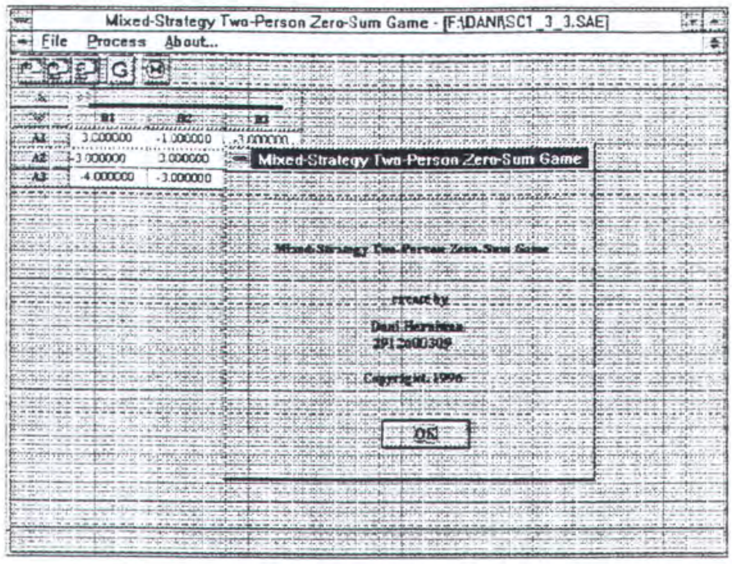
Gambar 3 : Tampilan setelah menu New



Gambar 4 : Tampilan setelah menu Open



Gambar 5 : Tampilan setelah menu Proses



Gambar 6 : Tampilan setelah menu About