

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

**PENGUNAAN PROGRAM *MAPLE* UNTUK MEDUKUNG
PEMBELAJARAN PROGRAM LINEAR DUA VARIABEL**

Skripsi

Diajukan untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Pendidikan

Program Studi Pendidikan Matematika



Oleh:

Teddy Suparyanto

NIM : 001414018



**PROGRAM STUDI PENDIDIKAN M^A TEMATIKA
JURUSAN PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2005**

HALAMAN PERSETUJUAN

SKRIPSI

**PENGGUNAAN PROGRAM *MAPLE* UNTUK MEDUKUNG
PEMBELAJARAN PROGRAM LINEAR DUA VARIABEL**

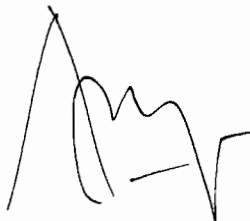
Oleh:

Teddy Suparyanto

NIM: 001414018

Telah disetujui oleh:

Pembimbing,



M. Andy Rudhito, S.Pd., M.Si.

Tanggal 31/3/05

HALAMAN PENGESAHAN

SKRIPSI

**PENGGUNAAN PROGRAM *MAPLE* UNTUK MEDUKUNG
PEMBELAJARAN PROGRAM LINEAR DUA VARIABEL**

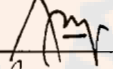

Dipersiapkan dan disusun oleh:

TEDDY SUPARYANTO

NIM: 001414018

Telah dipertahankan di depan Panitia Penguji
pada tanggal 22 Maret 2005 dan dinyatakan telah memenuhi syarat.

Susunan Panitia Penguji

	Nama Lengkap	Tanda Tangan
Ketua	: Drs. A. Atmadi, M.Si.	
Sekretaris	: Drs. Th. Sugiarto, M.T.	
Anggota	: M. Andy Rudhito, S.Pd., M.Si.	
	Drs. Th. Sugiarto, M.T.	
	Drs. A. Marjono	

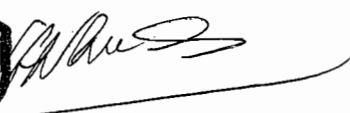
Yogyakarta, 22 Maret 2005

Fakultas Keguruan dan Ilmu Pendidikan

Universitas Sanata Dharma

Dekan,



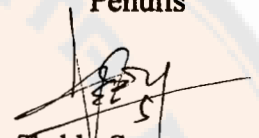

Dr. Annet Soewandi, M.Pd.

PERNYATAAN KEASLIAN KARYA

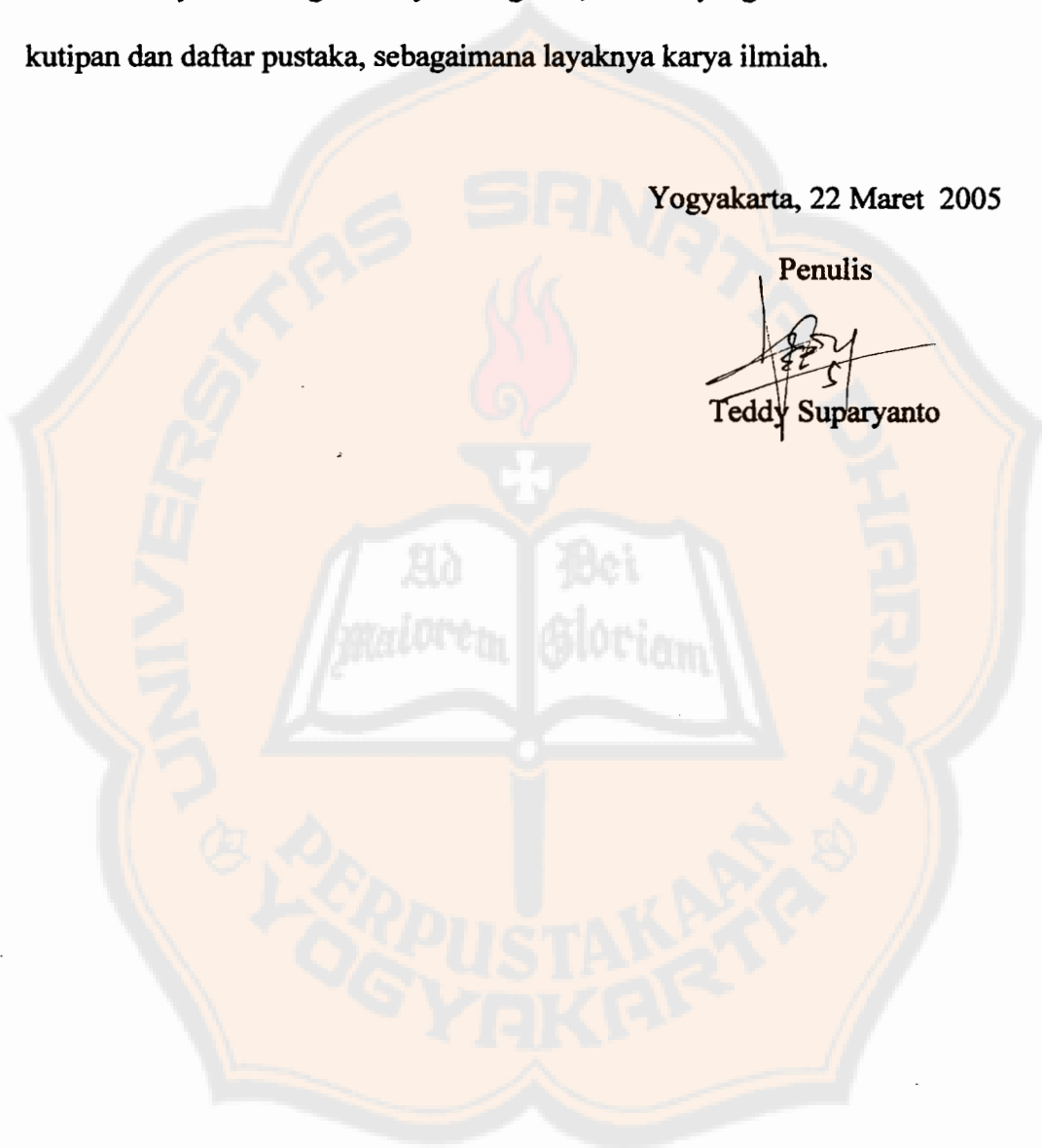
Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah.

Yogyakarta, 22 Maret 2005

Penulis



Teddy Suparyanto



ABSTRAK

Tujuan penulisan skripsi ini adalah mengeksplorasi fasilitas-fasilitas yang dimiliki oleh *Maple8* untuk mendukung pembelajaran Program Linear dua variabel, menyusun sebuah *interface* pembelajaran Program Linear dua variabel dan *handout* yang mendukung pembelajarannya.

Metode yang digunakan dalam penulisan skripsi ini adalah deskriptif eksploratif dan studi pustaka. Dieksplorasi dan dipaparkan secara deskriptif fasilitas-fasilitas dari *Maple8* yang terkait dengan materi Program Linear dua variabel dan kebutuhan untuk penyusunan sebuah *interface*. Eksplorasi dititikberatkan pada dukungan *Maple8* dalam penyelesaian masalah Program Linear dua variabel dengan menggunakan metode grafik. Dari studi pustaka yang dilakukan didapat panduan penyusunan *handout* dan materi-materi pembelajaran Program Linear dua variabel. *Handout* berisi cara pemanfaatan program dan panduan langkah-langkah eksplorasi program yang bisa dilakukan oleh siswa. Penyusunan *handout* dan *interface* dilakukan secara beriringan, agar terjadi keserasian isi antara *interface* dan *handout*.

Program *Maple8* dapat dimanfaatkan untuk membantu pembelajaran matematika pada topik penyelesaian masalah Program Linear dua variabel dengan Metode Grafik. *Interface* pembelajaran Program Linear dua variabel yang telah disusun, dirancang dengan menggunakan fasilitas paket dari *Maple8* yaitu *Maplet*. Kemampuan-kemampuan yang dimiliki *Maple8* adalah mampu memvisualkan masalah program linear dengan menggambarkan grafiknya dengan paket *Plot* meliputi perintah *plot* dan *inequal*. Dengan paket *Simplex*, *Maple8* mampu melakukan perhitungan untuk mencari nilai optimum masalah Program Linear dua variabel yaitu dengan perintah *maximize* dan *minimize*. Selain itu dengan paket *Maplet*, *Maple8* dapat digunakan untuk membuat *interface*, meliputi *windows*, *slider* yang memungkinkan pengguna untuk melakukan eksplorasi dengan cara menggesernya, dan *plotter* untuk menampilkan hasil *plot*. Dalam skripsi ini berhasil disusun sebuah *interface Maplet* sebagai *tool* dalam pembelajaran berbantuan komputer dan *handout*-nya yang dapat mendukung pembelajaran Program Linear dua variabel di SMU klas III IPA dan Analisis Sensitivitas untuk mahasiswa.

ABSTRACT

The aims of this graduating project are: to explore some *Maple8*'s facilities which is support to Two Variables Linear Programming Learning, to create a Two Variables Linear Programming Learning *interface* and to design its *handout*.

The methods that are used in this project are descriptive explorative and literature learning. It is explored and described all *Maple8*'s facilities which is related with Two Variables Linear Programming materi and all needed to arrange an *interface*. The exploration is focus at *Maple8*'s on solving Two Variable Linear Programming problems by graphic methods. By literature study, the writer get a guide to arrange a *handout*. The *handout*'s contents are the way to use the program and exploration guide. The *handout* was written while the *interface* was done.

Maple8 program can be used to solve Two Variable Linear Programming problems by graphic methods. The *interface* was designed by *Maple8*'s package, that is *Maplet*. Its abilities is visualized Linear Programming problem by plotting its graphic using Plot package, including *plot* and *inequal*. By simplex package, *Maple8* able to find the optimum value by *maximize* and *minimize* command. While, by *Maplet*, *Maple8* can be used to make an *interface*, including: *windows*, *slider*, and *plotter*.

The result of this project is a *Maplet interface* as a *tool* on computer assist learning and its *handout*, to support Two Variables Linear Programming Learning at IPA Third Class on Middle School and Sensitivita Analys for higher grade.

KATA PENGANTAR

Puji dan syukur kepada Allah di Surga, atas rahmat dan kasih-nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penggunaan Program *Maple* untuk Mendukung Pembelajaran Program Linear Dua Variabel”. Skripsi dapat terwujud karena bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini perkenankanlah penulis menyampaikan terima kasih kepada :

1. Bapak M. Andy Rudhito S.Pd., M.Si. selaku dosen pembimbing atas segala nasihat dan kritiknya sehingga skripsi ini dapat terwujud.
2. Bapak Drs. Th. Sugiarto, MT selaku Kaprodi Pendidikan Matematika atas dorongan yang diberikan dalam penyelesaian skripsi.
3. Segenap dosen JPMIPA, khususnya Program Studi Pendidikan Matematika atas pengetahuan yang penulis dapatkan.
4. Keluarga dan teman-teman yang telah memberi dukungan kepada penulis.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu saran dan kritik dari berbagai pihak yang bersifat membangun sangat penulis harapkan.

Penulis.

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN KEASLIAN KARYA.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Perumusan Masalah.....	2
C. Tujuan Penulisan.....	3
D. Manfaat Penulisan.....	3
E. Metode Penulisan.....	4
F. Sistematika Penulisan.....	5
BAB II LANDASAN TEORI.....	7
A. Komputer Sebagai Media dalam Pembelajaran Matematika.....	7



PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

B.	Penyusunan Perangkat Lunak Pembelajaran.....	12
C.	Program Linear Dua Variabel	16
D.	Sekilas Tentang <i>Maple8</i> dan <i>Maplet</i>	37
BAB III EKSPLORASI FASILITAS <i>MAPLE</i> YANG MEMBANTU		
	PEMBELAJARAN PROGRAM LINEAR DUA VARIABEL	41
A.	Eksplorasi Fasilitas <i>Maple</i> untuk Program Linear	41
B.	Eksplorasi Fasilitas <i>Maple</i> untuk Mendukung Penyusunan <i>Interface Maplet</i>	45
C.	Pemrograman dengan <i>Maple</i>	54
D.	Eksplorasi Fasilitas <i>Maplet</i>	58
BAB IV PENYUSUNAN <i>INTERFACE</i> UNTUK MENDUKUNG		
	PEMBELAJARAN PROGRAM LINEAR DUA VARIABEL	71
A.	Perencanaan Awal	71
B.	Penyiapan Materi.....	74
C.	Menyusun <i>Interface</i> dengan <i>Maplet</i>	76
D.	Validasi <i>Interface</i>	118
BAB V PENGGUNAAN <i>INTERFACE MAPLET</i> DALAM		
	PEMBELAJARAN PROGRAM LINEAR.....	119
A.	<i>Handout</i> untuk Pembelajaran Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear	119
B.	<i>Handout</i> untuk Pembelajaran Program Linear.....	143
C.	<i>Handout</i> untuk Pembelajaran Masalah Sensitivitas pada Program Linear.....	155

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB VI KESIMPULAN DAN SARAN.....	161
A. Kesimpulan.....	161
B. Saran.....	163
DAFTAR PUSTAKA	165
LAMPIRAN	



DAFTAR GAMBAR

Gambar 2.3.1	Garis putus-putus yang menggambarkan $2x - y = 4$	22
Gambar 2.3.2	Daerah penyelesaian pertidaksamaan $2x - y < 4$	23
Gambar 2.3.2	Garis $x + y = 4$	23
Gambar 2.3.4	Daerah penyelesaian pertidaksamaan $x + y \geq 4$	24
Gambar 2.3.5	(a) grafik $x - y \geq -3$, (b) grafik $2x+3y \leq 28$, (c) grafik $y \geq 2$	26
Gambar 2.3.6	Daerah penyelesaian dari SPL $\{x - y \geq -3, 2x + 3y \leq 28, y \geq 2\}$	26
Gambar 2.3.7	Daerah penyelesaian dari kendala $x \geq 0, y \geq 0, x + y \leq 120, x + 4y \leq 240$	29
Gambar 2.3.8	Daerah penyelesaian dari kendala $x \geq 0, y \geq 0, x + y \geq 12, x + 4y \geq 240$	30
Gambar 2.3.9	Garis senilai dengan $f = 10.000$ dan $f = 20.000$	32
Gambar 2.3.10	Kemungkinan nilai q dari contoh 2.3.9	35
Gambar 2.3.11	Perubahan gradien fungsi sasaran	37
Gambar 3.4.1	Contoh sebuah <i>Window</i>	59
Gambar 3.4.2	Contoh sebuah <i>Window</i> dengan beberapa elemen <i>window</i>	60
Gambar 3.4.3	<i>Button</i> (tombol)	61
Gambar 3.4.4	<i>RadioButton</i>	61
Gambar 3.4.5	<i>Label</i>	62
Gambar 3.4.6	<i>Slider</i>	62
Gambar 3.4.7	<i>Plotter</i>	64
Gambar 3.4.8	<i>TextField</i>	64
Gambar 3.4.9	Menu.....	65
Gambar 3.4.10	<i>Window</i> dengan tombol perintah <i>Close Window</i>	65
Gambar 3.4.11	<i>Window</i> hasil eksekusi Contoh 3.4.2.....	68
Gambar 3.4.12	<i>Window</i> hasil eksekusi Contoh 3.4.5.....	69
Gambar 4.2.1	Urutan materi yang akan disajikan.....	74
Gambar 4.3.1	<i>Flowchart</i> untuk prosedur Koordplotteks	84
Gambar 4.3.2	<i>Flowchart</i> untuk prosedur Pertidaksamaan	85
Gambar 4.3.3a	<i>Flowchart</i> untuk prosedur geser_kendala bagian a	87

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4.3.3b	<i>Flowchart</i> untuk prosedur geser_kendala bagian b	88
Gambar 4.3.4	<i>Flowchart</i> untuk prosedur geser_PLUS	88
Gambar 4.3.5	<i>Flowchart</i> untuk prosedur Gambar_PTL	90
Gambar 4.3.6	<i>Flowchart</i> untuk prosedur TeksSlider	91
Gambar 4.3.7	<i>Flowchart</i> untuk prosedur TeksSasaran	92
Gambar 4.3.8a	<i>Flowchart</i> untuk prosedur Uji_ttk (tahap 1)	93
Gambar 4.3.8b	<i>Flowchart</i> untuk prosedur Uji_ttk (tahap 2)	94
Gambar 4.3.8c	<i>Flowchart</i> untuk prosedur Uji_ttk (tahap 3)	95
Gambar 4.3.9	<i>Flowchart</i> untuk prosedur ntgradien	95
Gambar 4.3.10	<i>Flowchart</i> untuk prosedur tgradien	96
Gambar 4.3.11a	<i>Flowchart</i> untuk prosedur mntgradien (Tahap 1)	97
Gambar 4.3.11b	<i>Flowchart</i> untuk prosedur mntgradien (Tahap 2)	98
Gambar 4.3.11c	<i>Flowchart</i> untuk prosedur mntgradien (Tahap 3)	99
Gambar 4.3.11d	<i>Flowchart</i> untuk prosedur mntgradien (Tahap 4)	100
Gambar 4.3.12	Tahap ketiga dari prosedur gradient	100
Gambar 4.3.13	Tahap keempat dari prosedur Gambar_Grafik	101
Gambar 4.3.14	Tahap ketiga dari prosedur plotdititik	102
Gambar 4.3.15	<i>Flowchart</i> prosedur peringatan	102
Gambar 4.3.16	Tahap kedua prosedur Maksimum	103
Gambar 4.3.17	Tahap ketiga dari prosedur MaksimumKoord	104
Gambar 4.3.18	Tahap kedua prosedur Minimum	104
Gambar 4.3.19	Tahap ketiga dari prosedur MinimumKoord	105
Gambar 4.3.20	<i>Flowchart</i> prosedur pindah	105
Gambar 4.3.21	<i>Flowchart</i> prosedur tambah	105
Gambar 4.3.22	<i>Flowchart</i> prosedur kurang	106
Gambar 4.3.23	<i>Flowchart</i> prosedur gradien	106
Gambar 4.3.24	Layout <i>window</i> W1	107
Gambar 4.3.25	Rancangan layout <i>window</i> W2	108
Gambar 4.3.26	<i>Window</i> W2	109
Gambar 4.3.27	Rancangan layout <i>window</i> W3	110
Gambar 4.3.28	<i>Window</i> W3	112
Gambar 4.3.29	Rancangan layout <i>window</i> W4	113

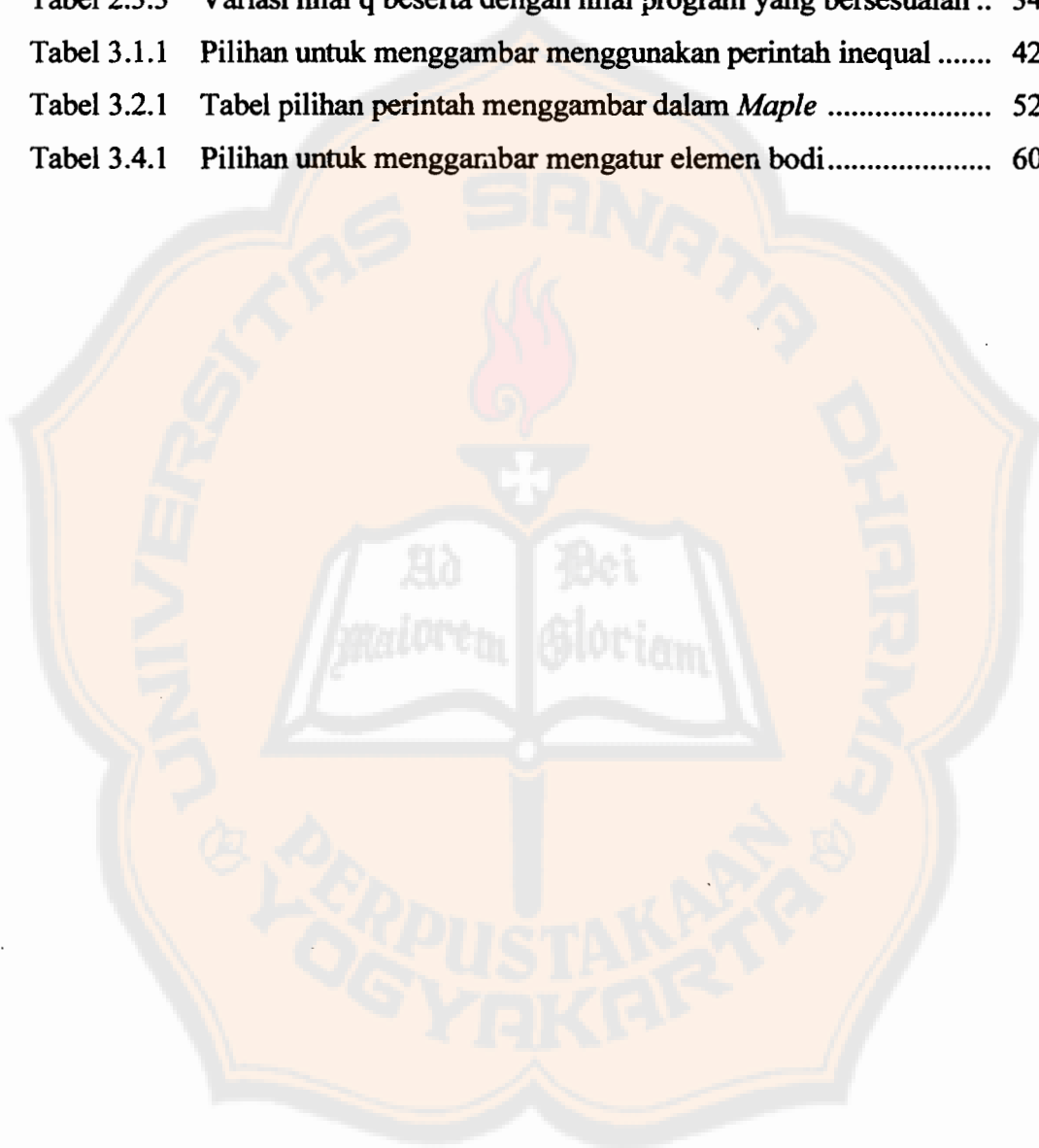
PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

Gambar 4.3.30 <i>Window W4</i>	114
Gambar 4.3.31 Rancangan layout <i>window W4</i>	115
Gambar 4.3.32 <i>Window W4</i>	117



DAFTAR TABEL

Tabel 2.3.1	Kurikulum 2004 untuk Materi Program Linear	19
Tabel 2.3.2	Tabel nilai program dari semua titik pojok daerah OABC pada Gambar 2.3.7	30
Tabel 2.3.3	Variasi nilai q beserta dengan nilai program yang bersesuaian ..	34
Tabel 3.1.1	Pilihan untuk menggambar menggunakan perintah <i>inequal</i>	42
Tabel 3.2.1	Tabel pilihan perintah menggambar dalam <i>Maple</i>	52
Tabel 3.4.1	Pilihan untuk menggambar mengatur elemen bodi.....	60



PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Kemajuan teknologi komputer yang pesat memberikan dampak yang positif terhadap perkembangan ilmu pengetahuan. Hal ini tampak dengan digunakannya komputer hampir di semua bidang ilmu pengetahuan. Banyak perangkat lunak (*software*) disusun untuk mempermudah kerja manusia dalam membantu menyelesaikan masalah, misalkan *MYOB* untuk bidang akuntansi, *CorelDraw* untuk bidang desain grafis, *Ulead* untuk bidang video, dan sebagainya.

Penggunaan komputer dalam membantu pembelajaran matematika sudah banyak dikembangkan dalam bentuk perangkat lunak pembelajaran, misalnya *Cabri*, *Winggeom*, *Math Advantage*, *Maple8*, dan sebagainya. Perangkat lunak pembelajaran matematika yang beredar saat ini masih banyak yang berbahasa asing. Baik bagi guru maupun bagi siswa, hal ini seringkali menjadi kendala yang menimbulkan keengganan untuk memanfaatkannya karena keterbatasan kemampuan berbahasa asing. Perangkat lunak yang berbahasa Indonesia diharapkan akan lebih mudah dikenal dan mudah digunakan oleh guru dan siswa.

Kondisi yang ideal adalah bila seorang guru mampu merancang dan menyusun sendiri perangkat lunak pembelajaran yang dapat dimanfaatkan di kelas. Dengan demikian guru dapat melakukan penyesuaian antara materi, kemampuan siswa dan penyajian. Karena keterbatasan waktu dan keterbatasan

pengetahuan dasar tentang pemrograman dan penggunaan komputer, seringkali guru merasa tidak mampu untuk melakukannya.

Sekarang ini mulai berkembang program aplikasi komputer yang sangat mudah pengoperasiannya dan memiliki kemampuan matematis yang bagus. Salah satunya adalah *Maple8*. *Maple8* mampu melakukan operasi aljabar tingkat tinggi dan memplot grafik. *Maple8* memiliki paket yang disebut dengan *Maplet*, yang dapat digunakan untuk membuat suatu *jendela (Window)*, tombol (*Button*), dan berbagai *interface* visual lain dari *Maple8*. Dengan menggunakan *Maplet*, *Maple8* dapat dijadikan salah satu alternatif untuk membuat *interface* pembelajaran matematika.

Program Linear dua variabel merupakan salah satu pokok bahasan dalam pelajaran matematika di tingkat SMA dan Universitas. Program Linear dua variabel dapat diselesaikan dengan metode grafik. Di tingkat sekolah menengah, siswa seringkali mengalami kesulitan dalam menggambar grafik daerah layak dari persoalan yang diminta. Sedangkan di tingkat universitas, seringkali mahasiswa mengalami kesulitan pada subpokok bahasan analisa sensitivitas.

Melihat kemampuan *Maple*, akan dirancang suatu perangkat lunak yang terdiri dari program dan *interface* dalam *Maplet* beserta *handoutnya* yang dapat digunakan sebagai *tool* dalam pembelajaran Program Linear dua variabel berbantuan komputer.

B. Perumusan Masalah

Adapun perumusan masalah yang akan dibahas dalam penulisan skripsi ini

adalah:

1. Fasilitas apa saja yang dimiliki *Maple8* untuk membantu pembelajaran Program Linear dua variabel.
2. Bagaimana menyusun *interface* dengan menggunakan *Maplet* untuk membantu pembelajaran Program Linear dua variabel.
3. Bagaimana penggunaan *interface Maplet* untuk membantu siswa maupun mahasiswa dalam pembelajaran Program Linear dua variabel.

C. Tujuan Penulisan

Penyusunan skripsi ini bertujuan untuk:

1. Mengetahui fasilitas-fasilitas yang dimiliki *Maple8* untuk membantu pembelajaran Program Linear dua variabel.
2. Mengetahui langkah-langkah penyusunan sebuah *interface* dengan *Maplet* untuk membantu pembelajaran materi Program Linear dua variabel.
3. Mengetahui cara-cara penggunaan *interface* untuk membantu siswa maupun mahasiswa dalam mempelajari materi Program Linear dua variabel dengan menggunakan *interface Maplet*, yang disusun dalam *handout*.

D. Manfaat Penulisan

Manfaat penulisan skripsi ini:

1. Bagi Universitas Sanata Dharma

Skripsi ini diharapkan dapat menambah referensi keputakaan dan dapat berguna bagi pihak-pihak yang memerlukan.

2. Bagi Dunia Kependidikan

Skripsi ini diharapkan dapat menambah alat bantu pembelajaran matematika, khususnya dalam pembelajaran Program Linear berbantuan komputer.

3. Bagi Penulis

Sebagai seorang calon guru, skripsi ini merupakan ajang latihan bagi penulis untuk membuat sebuah perangkat lunak pembelajaran. Dengan demikian diharapkan saat penulis menjadi guru kelak dan akan melakukan pembelajaran dengan berbantuan komputer, penulis dapat merancang dan menyusun sendiri perangkat lunak pembelajaran yang sesuai dengan tuntutan materi dan kemampuan siswa.

E. Metode Penulisan

Penulisan skripsi ini bersifat studi pustaka dan deskriptif eksploratif. Akan dikaji dan dieksplorasi bagaimana *Maple* dapat membantu menggambarkan daerah layak yang memenuhi kendala-kendala dalam masalah Program Linear dua variabel dan menentukan nilai optimum dari fungsi sasaran sebagai penyelesaian dari masalah Program Linear dua variabel dengan Metode Grafik. Kemudian disusun *interface* dengan *Maplet* beserta handout untuk membantu pembelajaran pembelajaran Program Linear dua variabel.

F. Sistematika Penulisan

Bab I skripsi ini merupakan bab pendahuluan yang berisi latar belakang pemilihan topik dan masalah - masalah yang disoroti. Selain itu dalam Bab I ini dikemukakan pula mengenai manfaat penulisan, metode penulisan beserta sistematika pembahasannya.

Dalam Bab II dari skripsi ini akan dituliskan teori-teori yang akan digunakan sebagai landasan dalam penyusunan skripsi ini. Teori-teori yang akan dituliskan dalam bab ini adalah mengenai komputer dalam pembelajaran matematika, Program Linear dua variabel, dan sekilas tentang *Maple8* dan *Maplet*.

Pada Bab III, dieksplorasi fasilitas *Maple* dan *Maplet* yang mendukung pembelajaran Program Linear dua variabel dan fasilitas yang digunakan dalam menyusun *interface*. Untuk itu, dalam Bab III ini disajikan hasil eksplorasi fasilitas *Maple* yang membantu pembelajaran Program Linear dua variabel.

Setelah melakukan eksplorasi, disusun *interface* yang akan dipaparkan dalam Bab IV. Penyajian dalam Bab IV ini meliputi perencanaan awal, pendalaman materi, penyusunan, dan validasi.

Implementasi rancangan ini dalam pembelajaran matematika akan dibahas dalam Bab V. Bab ini akan dibagi menjadi tiga subbab, yaitu: penggunaan *interface* untuk membantu pembelajaran materi pokok bahasan Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear, Program Linear kelas 3 IPA Sekolah Menengah Umum serta penggunaan *interface* untuk membantu mahasiswa mempelajari masalah sensitivitas pada mata kuliah Program Linear. Ketiga

subbab ini berisi rancangan *handout* untuk pembelajaran Program Linear dengan *interface* Maplet yang dirancang di Bab III.

Bab terakhir, yaitu Bab VI, berisi kesimpulan dan saran mengenai hasil rancangan ini.



BAB II

LANDASAN TEORI

A. Komputer sebagai Media dalam Pembelajaran Matematika

Penggunaan komputer sebagai media pembelajaran sudah mulai dikembangkan dan digunakan di dunia pendidikan kita. Hal ini ditandai dengan diujarkannya hal tersebut dalam rambu-rambu No. 7 Kurikulum 2004 bidang studi Matematika, yang berbunyi: "Sekolah dapat menggunakan teknologi seperti kalkulator, komputer, alat peraga, atau media lainnya untuk semakin meningkatkan efektifitas pembelajaran". Pembelajaran Berbantuan Komputer atau sering disingkat dengan PBK adalah segala hal yang berkaitan dengan situasi pembelajaran di mana dalam pembelajaran tersebut, kegiatan dan bahan pelajaran disampaikan melalui komputer.

Menurut Hannafin & Peck, 1998 (dalam <http://www.kota-makassar.net/umum/SLTP>), kegiatan dalam PBK dapat digolongkan menjadi empat kategori, yaitu:

1. Latihan

Dalam kegiatan ini, komputer memberikan soal-soal yang harus dipecahkan oleh siswa dan memberikan umpan balik terhadap jawaban-jawaban yang diberikan oleh siswa.

2. Tutorial

Dalam kegiatan tutorial, komputer memberikan berbagai informasi mengenai materi pembelajaran.

3. Permainan

Kegiatan permainan di sini ditujukan untuk memberi penguatan terhadap suatu konsep yang sudah dimiliki siswa atau juga dapat ditujukan untuk membantu siswa membentuk gambaran mengenai konsep baru yang akan dibangun.

4. Simulasi atau Pemodelan

Dalam simulasi atau pemodelan ini, komputer akan memberikan simulasi atau model dari suatu konsep. Siswa akan memberi masukan dan komputer akan memberikan respon terhadap masukan siswa sesuai dengan kondisi nyata dari konsep bila diberi respon tersebut.

Dari berbagai penelitian (bdk. <http://www.kota-makassar.net/umum/SLTP>) yang sudah pernah dilakukan, manfaat PBK dalam pembelajaran adalah:

1. Meningkatkan interaksi siswa dalam pembelajaran melalui pengelolaan tanggapan siswa dan umpan balik berdasarkan tanggapan tersebut.
2. Individualisasi belajar yang memperhatikan kemampuan awal dan kecepatan belajar siswa.
3. Meningkatkan motivasi siswa dalam belajar karena siswa dapat mengendalikan pembelajaran dan mendapat umpan balik yang segera.

Dalam pemanfaatan komputer sebagai alat bantu belajar, diperlukan adanya suatu paket pembelajaran yang biasa disebut dengan perangkat lunak pembelajaran. Paket pembelajaran ini terdiri dari program dan *handout*. Program yang dimaksud adalah sekumpulan perintah dalam bahasa komputer untuk melakukan suatu hal sesuai keinginan pembuat. Sedangkan *handout* berisi

petunjuk pemakaian program dan langkah-langkah pembelajaran beserta dengan evaluasi. Agar sebuah program dapat digunakan dengan baik, maka harus dirancang sebuah tampilan yang memungkinkan pengguna berinteraksi dengan program tersebut. Tampilan inilah yang disebut dengan *interface* (bdk. Ian Sommerville, 2003). Menurut Einsberg, 1986 (dalam <http://www.kota-makassar.net/umum/SLTP>), suatu perangkat lunak pembelajaran disebut baik, bila memenuhi karakteristik PBK berikut.

1. Siswa dimungkinkan untuk belajar kapan saja.
2. Terdapat respon segera terhadap setiap aksi yang diberikan oleh siswa.
3. Memungkinkan siswa berperan serta dalam proses pembelajaran.

Ada dua macam *tool* yang dapat dipakai untuk membuat suatu perangkat lunak pembelajaran, yaitu Program Aplikasi dan Bahasa Program.

Ada empat jenis Program Aplikasi yang sering dimanfaatkan, yaitu: paket program pengolah kata (misalnya *WS*, *MsWord*, *MsPowerPoint*), paket program pengolah angka yang berupa model lembar kerja atau dikenal dengan *electronics spreadsheet* (misalnya *MsExcell*), paket program grafik (misalnya *CorelDraw*), dan paket program pengolah data (misalnya *MsAccess*).

Dalam menyusun suatu perangkat lunak dengan menggunakan program aplikasi, tidak diperlukan pemahaman bahasa program, namun diperlukan kemampuan untuk mengatur dan mengkombinasikan tampilan-tampilan yang ada. Dengan demikian, biasanya suatu perangkat lunak yang disusun dengan menggunakan program aplikasi cenderung bersifat sederhana dan satu arah, dari komputer ke siswa. Memang dimungkinkan adanya respon dari komputer, tapi

sangat terbatas. Pengguna tidak dapat mengajukan permasalahan kepada komputer.

Bahasa program adalah himpunan atau susunan kata-kata dari berbagai simbol tertulis yang memungkinkan pemrogram atau pengguna komputer berkomunikasi dengan komputer (Shelly, Cashman dan Waggoner, 1990 dalam <http://www.kota-makassar.net/umum/SLTP>). Paket program dengan bahasa program memerlukan kemampuan penguasaan bahasa program. Dibandingkan dengan perangkat lunak yang disusun dengan menggunakan program aplikasi, suatu perangkat lunak yang disusun dengan menggunakan bahasa program bersifat lebih interaktif. Penggunaan bahasa program dalam suatu perangkat lunak pembelajaran memberikan peluang bagi pemrogram untuk dapat memvariasikan variabel matematika secara langsung dengan menggunakan bahasa tersebut. Hal ini memungkinkan pengguna mengajukan permasalahan kepada komputer.

Menurut Tamir, 1996 (dalam <http://www.kota-makassar.net/umum/SLTP>), dari jenisnya, suatu perangkat lunak pembelajaran dapat dibedakan menjadi tiga macam, yaitu:

1. Komputer sebagai *tool*

Sebagai *tool*, komputer dapat dimanfaatkan untuk:

- a. Kalkulasi biasa, misalkan menggunakan *Microsoft Excell* untuk menghitung jumlah, rata-rata dan mean suatu data.
- b. Menyajikan informasi yang dapat digunakan berkali-kali atau diulang-ulang sesuai dengan keperluan perespon (siswa).

- c. Menggambar atau membuat grafik, sehingga siswa dapat melihat representasi suatu fungsi, melihat hubungan antara berbagai variabel/ simbol dalam suatu grafik, melihat proses matematika tertentu, serta membantu siswa dalam mengatasi kesulitan visual (misalnya dalam pokok bahasan geometri dimensi tiga).
- d. Demonstrasi suatu pembelajaran oleh guru melalui suatu paket yang sudah disiapkan, misalnya dengan menggunakan *Microsoft Power Point*.
- e. Simulasi dan Permainan. Dengan menggunakan komputer, guru dapat mensimulasikan eksperimen dalam bentuk model matematika, sehingga siswa dapat memanipulasi dan mengeksplorasi hubungan antar parameter atau menemukan hukum atau rumus. Selain itu, komputer juga dapat membantu simulasi laboratorium untuk mengurangi bahaya dan menghemat waktu.

2. Komputer sebagai tutor

Sebagai tutor, komputer dapat digunakan untuk menampilkan dan menjelaskan suatu konsep. Dalam pemanfaatan komputer sebagai tutor, proses interaksi yang terjadi antara komputer dan siswa adalah sebagai berikut:

- a. Komputer menampilkan suatu informasi.
- b. Siswa menjawab masalah yang diajukan komputer sesuai dengan informasi yang diberikan.
- c. Komputer mengevaluasi jawaban siswa.
- d. Komputer menentukan langkah siswa selanjutnya berdasarkan hasil evaluasi dari jawaban siswa tersebut.

Sebagai tutor, komputer mempunyai beberapa manfaat berikut.

- a. Sebagai latihan sekaligus memberikan umpan balik bagi siswa.
- b. Mereview pencapaian hasil belajar dan menyelenggarakan program remedial serta melakukan dialog instruksional yang berisi tanya jawab yang dapat mengarahkan siswa untuk mencapai suatu kesimpulan.
- c. Melakukan tes yang hasilnya dapat langsung diketahui oleh siswa. Siswa juga dapat mengetahui letak kesalahannya dan dapat mengetahui jawaban yang benar. Tes semacam ini bersifat individual dan dapat disesuaikan dengan kecepatan masing-masing siswa.

3. Komputer sebagai *tutee*

Sebagai *tutee*, komputer dapat digunakan untuk:

- a. Mengajar suatu program aplikasi atau bahasa program komputer, misalnya: *Pascal*, *Maple*, dan lain-lain.
- b. Mempelajari hal-hal yang terkait dengan komputer, baik perangkat keras maupun perangkat lunak.

Dengan demikian, dalam peranannya sebagai *tutee*, komputer merupakan objek dalam suatu pembelajaran.

B. Penyusunan Perangkat Lunak Pembelajaran

Penyusunan suatu perangkat lunak pembelajaran perlu dilakukan dengan terencana. Langkah-langkah yang diambil adalah sebagai berikut (bdk. <http://www.kota-makassar.net/umum/SLTP>):

1. Perencanaan Awal

Dalam perencanaan awal, hal-hal yang perlu dilakukan adalah:

- a. Identifikasi tujuan, kebutuhan belajar dan masalah yang muncul dalam pembelajaran.
- b. Analisis karakteristik dan lingkungan siswa yang akan menggunakan dan materi yang akan dikembangkan.

Dari hasil identifikasi dan analisis di atas, seorang perancang perangkat lunak pembelajaran akan mendapat informasi mengenai: (1) tingkat kemampuan siswa, (2) pemanfaatan program: apakah akan digunakan secara klasikal atau digunakan secara mandiri, (3) tujuan-tujuan khusus yang ingin dicapai, misalkan: program akan digunakan sebagai perangkuman pelajaran, latihan, *problem solving*, atau yang lainnya, (4) jenis perangkat lunak yang diinginkan: sebagai *tool* atau tutor, atau gabungan keduanya.

2. Menyiapkan Materi untuk Perangkat lunak PBK

Langkah-langkah yang ditempuh dalam penyiapan materi untuk perangkat lunak PBK adalah:

- a. Pemilihan materi yang sesuai

Dalam memilih materi yang sesuai perlu diperhatikan rambu-rambu sebagai berikut:

- 1) Materi harus relevan dengan tujuan atau kompetensi-kompetensi yang ingin dicapai melalui kegiatan pembelajaran dengan menggunakan komputer tersebut. Perlu dibedakan antara materi untuk penguasaan

suatu konsep dengan materi untuk ketrampilan menggunakan konsep tersebut.

- 2) Materi harus cocok atau dapat disajikan melalui simbol-simbol dan gambar-gambar yang dapat ditampilkan oleh komputer.
- 3) Materi yang dipilih hendaknya materi yang ada di kurikulum.
- 4) Materi yang dipilih merupakan materi yang dibutuhkan orang banyak.

b. Penentuan lingkup pembelajaran

Lingkup materi perlu dipertimbangkan agar materi tersebut dapat dipelajari dalam tempo yang wajar.

c. Bahan evaluasi

Perlu dipertimbangkan apakah dalam perangkat lunak yang akan dibuat akan disertakan bahan evaluasi atau tidak.

d. Pemilihan sarana komunikasi

1) Menggunakan bahasa program

Berbagai bahasa program dapat digunakan, misalnya: *PASCAL* dan *BASIC*.

2) Menggunakan paket program aplikasi umum

Berbagai paket program aplikasi umum dapat digunakan, misalnya: *MsWord*, *Ms PowerPoint* atau *Ms Excell*.

3) Menggunakan paket program aplikasi khusus matematika

Berbagai paket program aplikasi khusus matematika dapat digunakan, misalnya: *Geometri Sketchpad*, *Cabri Geometry*, *Maple8*, dan *WinPlot*.

e. Buku pedoman penggunaan (*manual user guide*) atau *handout*

Agar perangkat lunak yang dirancang dapat digunakan secara optimal, perlu disusun buku panduan yang memuat antara lain:

- 1) Isi materi dan sasaran (pengguna).
- 2) Tujuan pembelajaran atau kompetensi pengguna yang ingin dicapai dengan menggunakan perangkat lunak tersebut.
- 3) Petunjuk teknis yang memuat deskripsi tentang karakteristik atau spesifikasi serta kemampuan minimal PC (*Personal Computer*) yang diperlukan untuk menjalankan perangkat lunak tersebut.
- 4) Langkah-langkah penggunaan perangkat lunak yang ditulis secara sederhana sehingga pengguna yang tidak terlalu menguasai komputer pun merasa tertarik dan dapat menggunakan.
- 5) Pedoman bagi guru, misalnya pertanyaan-pertanyaan kunci yang harus disampaikan guru dan berbagai hal yang terkait dengan optimalisasi untuk meningkatkan efisiensi dan efektifitas pembelajaran.

3. Mendesain Perangkat Lunak PBK

Ketrampilan yang perlu dimiliki oleh perancang perangkat lunak PBK:

- a. Penguasaan materi dan metodologi pengajaran.
- b. Penguasaan prosedur pengembangan media.
- c. Penguasaan teknis pemrograman komputer serta penguasaan komputer.
- d. Pengetahuan mengenai keterbatasan komputer sehingga perangkat lunak yang disusun tetap komprehensif dan bermanfaat.

Rancangan perangkat lunak PBK perlu disesuaikan dengan fungsi pembelajarannya, misalkan: apakah untuk memperkenalkan materi baru, apakah untuk melengkapi atau menguatkan pelajaran yang telah berlangsung.

4. Memvalidasi Perangkat Lunak PBK

Memvalidasi perangkat lunak adalah membuktikan validasinya secara empirik melalui uji lapangan terhadap perangkat lunak yang dirancang. Menurut Jerman (1969, dalam <http://www.kota-makassar.net/umum/SLTP>), validasi perangkat lunak perlu memperhatikan:

- a. Kebenaran bahan ajar.
- b. Kecepatan antara program dan populasi pengguna.
- c. Kemudahan penggunaannya.
- d. Efisiensi penggunaannya.
- e. "Realibilitas"-nya.

Dalam skripsi ini, perangkat lunak terdiri dari program yang ditampilkan dalam bentuk *interface* dan *handout*. Program dibuat dengan menggunakan bahasa pemrograman dalam *Maple8*, sedangkan *interface*-nya disusun dengan menggunakan fasilitas paket dari *Maple8*, yaitu *Maplet*. *Handout* akan dibahas secara lengkap pada Bab VI.

C. Program Linear Dua Variabel

Pada mulanya, Program Linear adalah suatu analisis dari metode masukan dan keluaran (*input-output method*) yang dikembangkan oleh seorang ahli ekonomi bernama *W.W. Leontief*. Saat itu Program Linear seringkali digunakan

untuk pengambilan keputusan dalam hal-hal yang menyangkut masalah ekonomi, seperti: manajemen, industri, penentuan alokasi barang, ataupun distribusi barang. Pada perkembangan berikutnya, masalah Program Linear ini secara lebih luas diterapkan juga untuk masalah transportasi oleh *Hitchcock* (1941) dan *Koopmans* (1947). Pada tahun 1945, *Stigler* mempergunakan Program Linear untuk memecahkan masalah diet. Kini Program Linear banyak diterapkan dalam berbagai bidang yang menyangkut masalah mencari solusi optimal.

Program Linear yang sederhana adalah Program Linear yang hanya melibatkan dua buah variabel, sehingga dengan mudah dapat diselesaikan dengan menggunakan Metode Grafik. Akan tetapi, mulailah timbul masalah saat problem yang dihadapi melibatkan lebih dari dua variabel. Karena itu pada tahun 1948, *G.B. Dantzig* mulai memperkenalkan Metode Simpleks yang mampu mengatasi permasalahan diatas. Metode ini kemudian dikembangkan oleh Angkatan Udara Amerika Serikat (*U.S. Air Force*) untuk menyelesaikan Problem Transportasi Udara.

Pada skripsi ini tidak akan dibicarakan masalah Program Linear yang menyangkut lebih dari dua variabel. Metode penyelesaian yang akan dibahas hanyalah Metode Grafik. Karena itu sebelum masuk ke pembahasan Program Linear, akan dibahas masalah Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear terlebih dahulu.

Dalam Kurikulum Matematika SMA, pokok bahasan Program Linear diajarkan pada siswa di kelas III. Kompetensi dasar yang diharapkan dapat dicapai adalah pemahaman siswa dan ketrampilan siswa dalam membuat sistem

pertidaksamaan linear untuk menyelesaikan soal-soal Program Linear. Materi pelajarannya dikelompokkan menjadi subpokok bahasan, yaitu subpokok bahasan sistem pertidaksamaan linear dan subpokok bahasan nilai optimum suatu bentuk objektif.

Dalam subpokok bahasan sistem pertidaksamaan linear, siswa diharapkan dapat mengingat kembali cara menentukan daerah penyelesaian suatu sistem pertidaksamaan linear dua peubah, memahami pengertian Program Linear dan model matematika, serta dapat mengubah soal cerita menjadi model matematika. Dalam subpokok bahasan yang kedua, diharapkan siswa mulai mengenal bentuk objektif $ax + by$ dan menentukan nilai optimum bentuk tersebut. Diharapkan pula siswa mulai mengenal pengertian garis selidik yang berbentuk $ax + by = k$, sehingga selain dapat menyelesaikan persoalan Program Linear dengan menggunakan metode uji titik sudut daerah penyelesaian, siswa dapat menentukan nilai optimum dengan menggunakan garis selidik.

Di Program Studi Pendidikan Matematika Universitas Sanata Dharma, Program Linear juga menjadi salah satu matakuliah yang wajib ditempuh oleh mahasiswa. Selain pokok bahasan diatas, salah satu subpokok bahasan yang ada dalam matakuliah tersebut dan seringkali menjadi kendala bagi mahasiswa adalah masalah sensitivitas pada Program Linear.

Aljabar		
Standar Kompetensi:		
<ul style="list-style-type: none"> Memahami, merancang dan menggunakan model matematika Program Linear dalam pemecahan masalah nilai optimal 		
KOMPETENSI DASAR	INDIKATOR	MATERI POKOK
<p>17. Menyelesaikan sistem pertidaksamaan linear dua variabel dan menyelesaikan masalah program linear</p> <p><i>Perwujudan kompetensi di atas ditunjukkan dengan hasil belajar berikut.</i></p> <p>17.1. Merumuskan masalah nyata ke dalam model matematika sistem pertidaksamaan linear, menyelesaikan, dan menafsirkan hasil yang diperoleh</p>	<p>Siswa dapat</p> <ul style="list-style-type: none"> Menjelaskan dan menentukan penyelesaian sistem pertidaksamaan linear dua variabel Menentukan fungsi tujuan beserta kendala yang harus dipenuhi dalam masalah program linear Menggambarkan kendala sebagai daerah di bidang yang memenuhi sistem pertidaksamaan linear Menentukan nilai optimum dari fungsi tujuan sebagai penyelesaian dari program linear Menafsirkan nilai optimum yang diperoleh sebagai penyelesaian masalah program linear 	<p>Program Linear</p>

Tabel 2.3.1. Kurikulum 2004 untuk Materi Program Linear

Dalam uraian dibawah ini, dijabarkan beberapa teori dasar dalam Program Linear yang diajarkan di SMA dan di tingkat Universitas.

1. Pertidaksamaan Linear dalam Dua Variabel

Definisi 2.3.1 (Pertidaksamaan Linear dengan Dua Variabel, Sartono, 2000 p.105)

Pertidaksamaan linear dengan dua variabel adalah suatu kalimat matematika yang mengandung salah satu tanda $<$, $>$, \leq , atau \geq , dan melibatkan dua buah variabel dan masing-masing variabel tersebut berderajat satu.

Contoh 2.3.1.

a. Berikut ini contoh pertidaksamaan linear dua variabel

- $x - 3y < 5$
- $2x + y \leq 4$
- $x - y > -3$
- $2x + 5y \geq 10$

b. Berikut contoh bukan pertidaksamaan linear dua variabel

- $x + y = 3$
- $x^2 + 6y \leq 7$

Untuk mempersingkat penulisan selanjutnya, dalam skripsi ini pertidaksamaan linear dalam dua variabel akan ditulis sebagai pertidaksamaan.

Penyelesaian dari suatu pertidaksamaan linear adalah semua anggota himpunan pasangan-pasangan koordinat-koordinat (x, y) yang memenuhi pernyataan pertidaksamaan linear tersebut. Dengan demikian, penyelesaian dari pertidaksamaan linear tidak tunggal. Himpunan penyelesaian dari suatu pertidaksamaan linear dengan dua variabel biasanya ditampilkan dalam bentuk

grafik yang digambarkan pada sebuah bidang Cartesius. Himpunan penyelesaian ini sering disebut sebagai *daerah penyelesaian* atau *daerah layak*.

Andaikan diketahui suatu pertidaksamaan $ax + by * c$, dimana $a, b, c \in \mathbb{R}$ dan $*$ merupakan salah satu dari $<, >, \leq$ atau \geq . Penyelesaian pertidaksamaan linear dua variabel didapat dari langkah-langkah berikut.

1. Gambar garis $ax + by = c$. Garis ini kita namakan batas himpunan penyelesaian.
 - a. Apabila $*$ merupakan salah satu dari $<$ atau $>$, maka garis $ax + by = c$ digambar sebagai garis putus-putus.
 - b. Apabila $*$ merupakan salah satu dari \leq atau \geq , maka garis $ax + by = c$ digambar sebagai garis mulus.

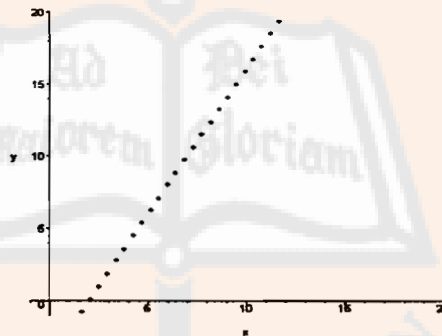
Garis $ax + by = c$ ini akan membagi bidang Cartesius menjadi dua setengah bidang Cartesius. Setengah bidang itu akan disebut sebagai satu bagian bidang.

2. Ambil sembarang titik uji (u, v) yang tidak berada pada garis $ax + by = c$. Substitusi titik uji (u, v) ke dalam pertidaksamaan $ax + by * c$. Apabila pertidaksamaan $au + bv * c$ bernilai benar, maka satu bagian bidang Cartesius yang mengandung (u, v) merupakan daerah penyelesaian. Sebaliknya, apabila $au + bv * c$ bernilai salah, maka satu bagian bidang Cartesius yang mengandung titik (u, v) bukan merupakan daerah penyelesaian. Apabila satu bagian bidang Cartesius bukan merupakan daerah penyelesaian, maka satu bagian yang lain merupakan daerah penyelesaian.

3. Langkah terakhir adalah mengarsir bagian bidang yang merupakan daerah penyelesaian untuk menunjukkan daerah penyelesaian.

Contoh 2.3.2.

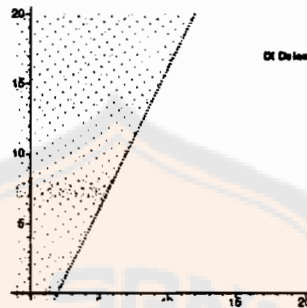
1. Himpunan penyelesaian dari pertidaksamaan $2x - y < 4$ dapat ditentukan dengan langkah berikut.
 - a. Digambar garis $2x - y = 4$ pada bidang Cartesius. Garis ini adalah garis yang memotong sumbu x dan y di titik $(2, 0)$ dan $(0, -4)$. Karena tanda pertidaksamaan yang diminta adalah " $<$ " maka garis $2x - y = 4$ digambar sebagai garis putus-putus.



Gambar 2.3.1. Garis putus-putus yang menggambarkan $2x - y = 4$

- b. Diambil titik uji $(0, 0)$. Titik $(0, 0)$ ini dapat diambil sebagai titik uji sebab $(0, 0)$ tidak terletak pada garis $2x - y = 4$. Titik uji $(0, 0)$ dimasukkan ke pertidaksamaan $2x - y < 4$, dan dihasilkan $0 < 4$. Kalimat matematika ini bernilai benar. Maka bagian bidang yang memuat titik $(0, 0)$ merupakan daerah penyelesaian.
 - c. Diarsir bagian bidang yang merupakan daerah penyelesaian.

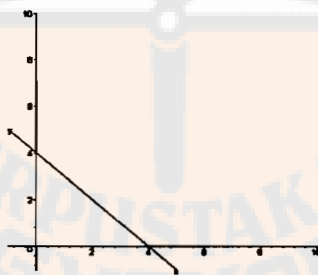
Dengan demikian himpunan penyelesaian dari $2x - y < 4$ dapat digambarkan oleh grafik berikut (Gambar 2.3.2).



Gambar 2.3.2. Daerah penyelesaian pertidaksamaan $2x - y < 4$

2. Himpunan penyelesaian dari pertidaksamaan $x + y \geq 4$ dapat ditentukan dengan langkah berikut.

a. Digambar garis $x + y = 4$ pada bidang Cartesius. Garis ini adalah garis yang memotong sumbu x dan y di titik $(4, 0)$ dan $(0, 4)$. Karena tanda pertidaksamaan yang diminta adalah " \geq " maka garis $x + y = 4$ digambar sebagai garis mulus.

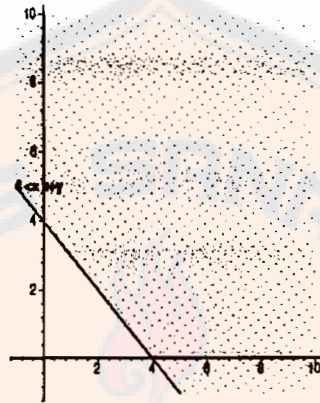


Gambar 2.3.3. Garis $x + y = 4$

b. Diambil titik uji $(0, 0)$. Titik $(0, 0)$ ini dapat diambil sebagai titik uji sebab $(0, 0)$ tidak terletak pada garis $x + y = 4$. Titik uji $(0, 0)$ dimasukkan ke pertidaksamaan $x + y \geq 4$, dan dihasilkan $0 \geq 4$. Kalimat matematika ini bernilai salah. Maka bagian bidang yang memuat titik $(0, 0)$ bukan merupakan daerah penyelesaian.

- c. Diarsir bagian bidang yang merupakan daerah penyelesaian, yaitu bagian bidang yang tidak memuat titik $(0, 0)$.

Dengan demikian himpunan penyelesaian dari $2x - y < 4$ dapat digambarkan oleh grafik berikut.



Gambar 2.3.4. Daerah penyelesaian pertidaksamaan $x + y \geq 4$

2. Sistem Pertidaksamaan Linear Dua Variabel

Definisi 2.3.2 (Sistem Pertidaksamaan Linear Dua Variabel, Sartono, 2000, p.114)

Sistem Pertidaksamaan Linear Dua Variabel terbentuk dari dua atau lebih pertidaksamaan linear dua variabel dimana antara satu pertidaksamaan dengan pertidaksamaan lain mempunyai variabel yang sama.

Contoh 2.3.3.

Sistem $\begin{cases} x - y > 0 \\ 2x + 3y \leq 7 \\ y \geq 0 \end{cases}$ merupakan sistem pertidaksamaan linear dua variabel yang

terdiri dari tiga pertidaksamaan. Sedangkan sistem $\begin{cases} x - y > 0 \\ 2a + 3b \leq 7 \\ t \geq 0 \end{cases}$ bukan

merupakan sistem pertidaksamaan linear dua variabel sebab pertidaksamaan-pertidaksamaan yang menyusunnya memiliki variabel yang berbeda-beda.

Untuk mempersingkat penulisan selanjutnya, dalam skripsi ini sistem pertidaksamaan linear dua variabel akan ditulis sebagai sistem pertidaksamaan linear, atau akan disingkat sebagai SPL.

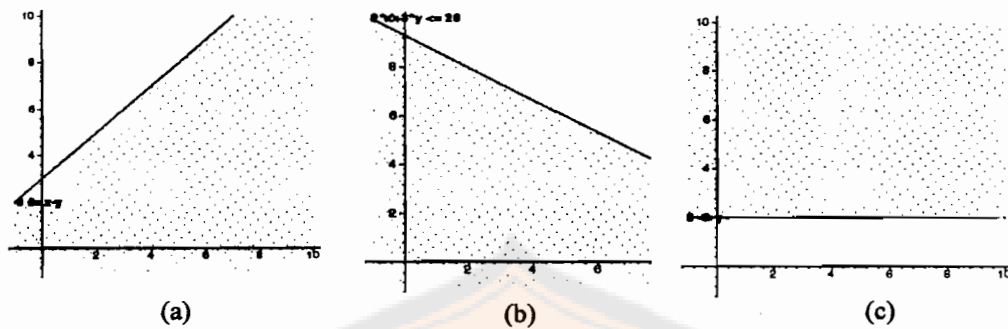
Himpunan penyelesaian dari SPL merupakan irisan dari himpunan penyelesaian masing-masing pertidaksamaan yang menyusun SPL tersebut. Dengan demikian daerah penyelesaian dari suatu SPL merupakan irisan dari daerah penyelesaian pertidaksamaan yang menyusunnya.

Contoh 2.3.4.

SPL $\begin{cases} x - y \geq -3 \\ 2x + 3y \leq 28 \\ y \geq 2 \end{cases}$ tersusun dari tiga buah pertidaksamaan, yaitu

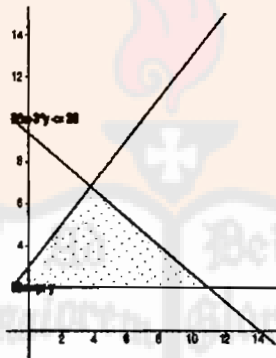
pertidaksamaan-pertidaksamaan $x - y \geq -3$, $2x + 3y \leq 28$ dan $y \geq 2$. Daerah penyelesaian dari masing-masing pertidaksamaan tersebut digambarkan dalam grafik-grafik pada gambar 3.2.5 berikut.





Gambar 2.3.5. (a) grafik $x - y \geq -3$, (b) grafik $2x + 3y \leq 28$, (c) grafik $y \geq 2$

Sehingga daerah penyelesaian dari SPL tersebut merupakan irisan dari ketiga grafik di atas dan dapat digambarkan sebagai berikut.



Gambar 2.3.6. Daerah penyelesaian dari SPL $\{x - y \geq 3, 2x + 3y \leq 28, y \geq 2\}$

3. Program Linear

Definisi 2.3.3. (Masalah Program Linear, Susanta, 1994, p.6)

Secara umum, masalah program linear dapat dirumuskan sebagai berikut.

Mencari :

$$x_j, j = 1, 2, \dots, n \tag{2.1}$$

yang mengoptimalkan (memaksimumkan atau meminimumkan)

$$f = \sum_j c_j x_j \tag{2.2}$$

dengan kendala

$$\sum_{j=1}^n a_{ij}x_j (\leq, =, \geq) b_i \quad i = 1, 2, \dots, m \quad (2.3)$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n \quad (2.4)$$

Bagian (2.1) menyatakan variabel yang akan dicari. Dalam skripsi ini, masalah program linear dibatasi pada masalah program linear dua variabel. Fungsi (2.2) disebut sebagai fungsi sasaran (*objective function*) yang akan dioptimalkan. Dalam masalah program linear, fungsi sasaran selalu merupakan fungsi linear. Kendala (2.3) disebut sebagai kendala utama yang membatasi proses pengoptimalan fungsi sasaran. Sedangkan kendala (2.4) disebut sebagai kendala tambahan. Kendala ini sebetulnya tidak mutlak harus disertakan, namun ternyata pada penerapannya dalam masalah nyata, masalah yang dihadapi memuat syarat tak negatif bagi peubah-peubahnya (berhubungan dengan masalah kuantitas di dunia nyata). Kendala-kendala dalam masalah program linear dapat berupa persamaan-persamaan atau pertidaksamaan-pertidaksamaan linear. Dalam skripsi ini, dikhususkan pada masalah program linear dengan kendala-kendala berupa pertidaksamaan linear dengan tanda ' \leq ' atau ' \geq '.

Contoh 2.3.5.

Dalam model matematika, suatu masalah dapat dirumuskan sebagai berikut.

Mencari x dan y tak negatif yang memenuhi $\begin{cases} x + y \leq 8 \\ 3x - y \geq 0 \end{cases}$ dan memaksimalkan

$$f(x, y) = 50x + 100y.$$

Masalah tersebut merupakan contoh dari masalah program linear sebab memenuhi ketentuan yang disyaratkan dalam Definisi 2.3.3.

Sedangkan permasalahan yang dalam model matematikanya dirumuskan sebagai:

Mencari x dan y tak negatif yang memenuhi $\begin{cases} x - 3y \leq 27 \\ 3x + y \geq 0 \end{cases}$ dan

memaksimalkan $f(x, y) = xy - 10y$.

bukan merupakan masalah program linear sebab fungsi sasarannya bukan merupakan fungsi yang linear.

Ada dua cara menyelesaikan masalah program linear, yaitu dengan menggunakan Metode Grafik atau dengan menggunakan Metode Simpleks. Metode Grafik digunakan bila masalah program linear yang dihadapi melibatkan dua variabel. Sedangkan Metode Simpleks dapat dipakai pada masalah program linear yang melibatkan lebih dari dua variabel. Dalam Metode Grafik ada dua macam cara yang dapat ditempuh untuk mencari penyelesaian optimum, yaitu dengan Metode Uji Titik Pojok dan Metode Garis Selidik.

Dalam skripsi ini akan diangkat cara penyelesaian masalah program linear dua variabel dengan menggunakan Metode Grafik, baik yang Metode Uji Titik Pojok maupun Metode Garis Selidik.

a. Metode Uji Titik Pojok

Langkah-langkah dalam menyelesaikan masalah program linear dengan Metode Uji Titik Pojok adalah:

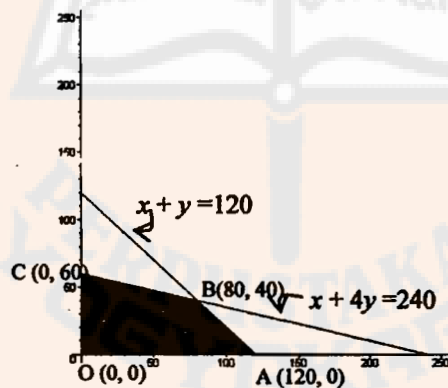
- 1) Menggambar daerah penyelesaian dari semua kendala yang diketahui.

- 2) Mencari koordinat-koordinat titik pojok dari daerah penyelesaian yang telah didapat dari langkah pertama.
- 3) Mensubstitusi setiap pasangan koordinat titik pojok ke dalam fungsi sasaran f .
- 4) Membandingkan hasil substitusi dari langkah ketiga. Nilai f terbesar yang didapat merupakan nilai maksimum dan nilai f terkecil merupakan nilai minimum.

Contoh 2.3.6.

Akan dimaksimumkan nilai fungsi sasaran $f = 250x + 500y$ dengan kendala $x \geq 0, y \geq 0, x + 4y \leq 240$, dan $x + y \leq 120$.

Dari kendala-kendala $x \geq 0, y \geq 0, x + 4y \leq 240$, dan $x + y \leq 120$ didapat daerah penyelesaian seperti pada Gambar 2.3.7 berikut.



Gambar 2.3.7. Daerah penyelesaian dari kendala $x \geq 0, y \geq 0, x + y \leq 120, x + 4y \leq 240$

Koordinat-koordinat titik pojok dari daerah penyelesaian tersebut adalah $O(0,0), A(120,0), B(80,40)$ dan $C(0,60)$. Langkah selanjutnya, substitusikan keempat pasangan koordinat tersebut ke dalam fungsi sasaran. Hasilnya tampak pada Tabel 2.3.1 berikut.

Titik pojok (x,y)	$f = 250x + 500y$
O (0, 0)	0
A (120, 0)	30.000
B (80, 40)	40.000
C (0, 60)	30.000

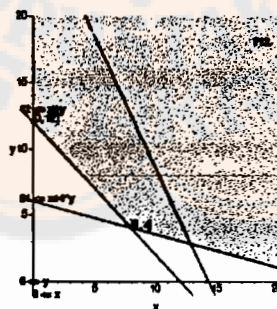
Tabel 2.3.2. Tabel nilai program dari semua titik pojok daerah OABC pada Gambar 2.3.7

Dari Tabel 2.3.2 tersebut tampak bahwa substitusi titik B(80, 40) ke fungsi sasaran $f = 250x + 500y$ menghasilkan nilai maksimal. Dengan demikian, titik B adalah titik maksimal dengan nilai $f_{maksimal} = 40.000$.

Metode uji titik pojok ini tidak dapat dipakai saat gambar daerah layak dari permasalahan Program Linear yang dihadapi berupa daerah terbuka, sehingga tidak semua titik pojok dari daerah layak dapat ditemukan. Berikut ini adalah contoh dari masalah Program Linear yang tidak dapat diselesaikan dengan metode uji titik pojok.

Contoh 3.2.7

Akan dimaksimumkan nilai fungsi sasaran $f = 50x - 25y$ dengan kendala $x \geq 0, y \geq 0, x + 4y \geq 24$, dan $x + y \geq 12$. Dari kendala-kendala $x \geq 0, y \geq 0, x + 4y \geq 24$, dan $x + y \geq 12$ didapat daerah penyelesaian seperti pada Gambar 2.3.8 berikut.



Gambar 2.3.8. Daerah penyelesaian dari kendala $x \geq 0, y \geq 0, x + y \geq 12, x + 4y \geq 24$

Dari gambar 3.2.8 terlihat bahwa daerah layak merupakan daerah tak terbatas. Dengan demikian ada titik pojok dari daerah layak ini yang tidak dapat ditentukan. Sehingga persoalan Program Linear semacam ini tidak dapat diselesaikan dengan metode uji titik pojok.

b. Metode Garis Selidik

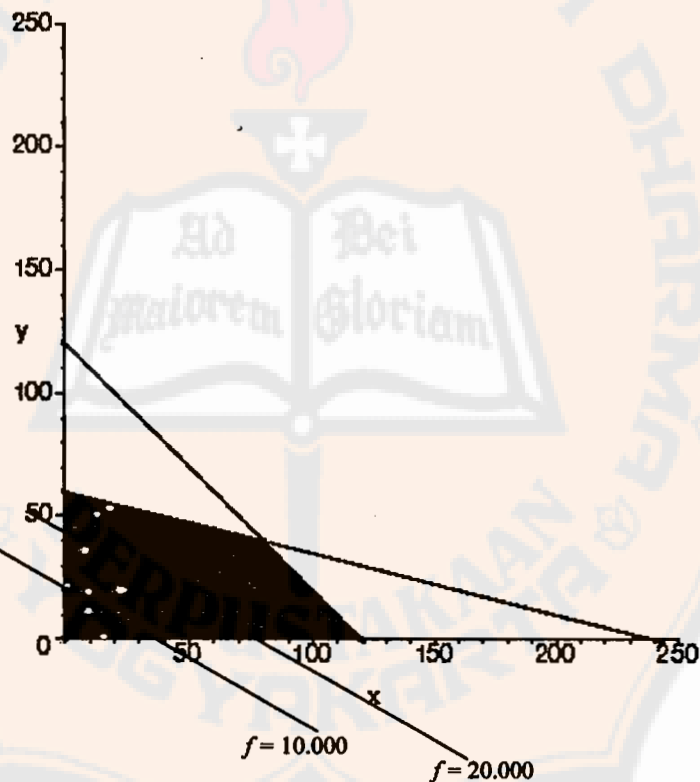
Langkah-langkah dalam menyelesaikan program linear dengan metode garis selidik adalah:

- 1) Gambar daerah penyelesaian dari semua kendala yang diketahui.
- 2) Gambar dua garis senilai (*isoprofit lines*) dengan cara memberi sembarang nilai konstan k pada fungsi sasaran.
- 3) Bandingkan nilai fungsi sasaran dan gambar grafiknya. Dari sini dapat diamati arah pergeseran garis selidik yang akan dibuat.
- 4) Menggambar sebuah garis yang sejajar dengan garis senilai. Garis ini digeser sejajar dengan garis senilai sesuai arah pergeseran untuk mendapatkan titik optimum. Jika akan memaksimumkan fungsi sasaran maka garis selidik digeser ke arah yang menghasilkan nilai fungsi sasaran membesar. Demikian juga sebaliknya. Jika akan meminimumkan fungsi sasaran maka garis selidik digeser ke arah yang menghasilkan nilai fungsi sasaran mengecil.

Contoh 2.3.8.

Persoalan program linear dari Contoh 2.3.6 akan diselesaikan dengan menggunakan metode garis selidik.

Gambar daerah layaknya sama dengan gambar 2.3.7. Untuk membuat garis senilai, diambil nilai $k = 10.000$ dan $k = 20.000$. Sehingga persamaan masing-masing garis senilai adalah $250x + 500y = 10.000$ dan $250x + 500y = 20.000$. Sampai dengan langkah ini penyelesaian dapat ditampilkan pada gambar 2.3.9 berikut.



Gambar 2.3.9. Garis senilai dengan $f = 10.000$ dan $f = 20.000$

Dari Gambar 3.2.9 tersebut tampak bahwa semakin kekanan garis senilai mempunyai nilai semakin besar. Dengan demikian untuk mendapatkan

titik optimum, harus dibuat sebuah garis yang sejajar dengan garis-garis senilai tersebut dan menggesernya ke kanan sampai titik terakhir dari daerah layak. Titik inilah yang menjadi titik optimum. Didapat titik optimum (80,40) dengan nilai program $f = 40.000$.

4. Masalah Sensitivitas

Masalah sensitivitas adalah masalah kepekaan perubahan nilai optimum saat terjadi perubahan batasan atau sumber serta ongkos dalam permasalahan yang dimodelkan. Ada dua macam masalah sensitivitas yang akan dibahas dalam skripsi ini, yaitu:

- a. Masalah sensitivitas dengan perubahan pada konstanta salah satu kendala utama.
- b. Masalah sensitivitas dengan perubahan pada koefisien salah satu variabel dalam fungsi sasaran.

Masalah sensitivitas dengan perubahan pada konstanta salah satu kendala utama muncul akibat adanya perubahan batas dari suatu kendala. Perubahan ini mengakibatkan perubahan daerah layak, yang memungkinkan perubahan titik optimum.

Sedangkan masalah sensitivitas dengan perubahan pada koefisien salah satu variabel dalam fungsi sasaran diakibatkan oleh adanya perubahan koefisien ongkos (harga) di pasar. Pada masalah sensitivitas dengan perubahan pada koefisien salah satu variabel dalam fungsi sasaran ini titik optimum tetap dipertahankan. Dengan demikian, masalahnya adalah mencari sejauh mana

koefisien ongkos itu boleh berubah asalkan titik optimum tidak berubah. Perubahan koefisien ongkos ini akan mempengaruhi perubahan gradien dari fungsi sasaran.

Contoh 2.3.9

Diketahui suatu masalah program linear:

Mencari x, y tak negatif yang memenuhi $x + y \leq 8, 3x - y \geq 0$ dan $x \geq q$, yang memaksimalkan $f = 10x + 20y$.

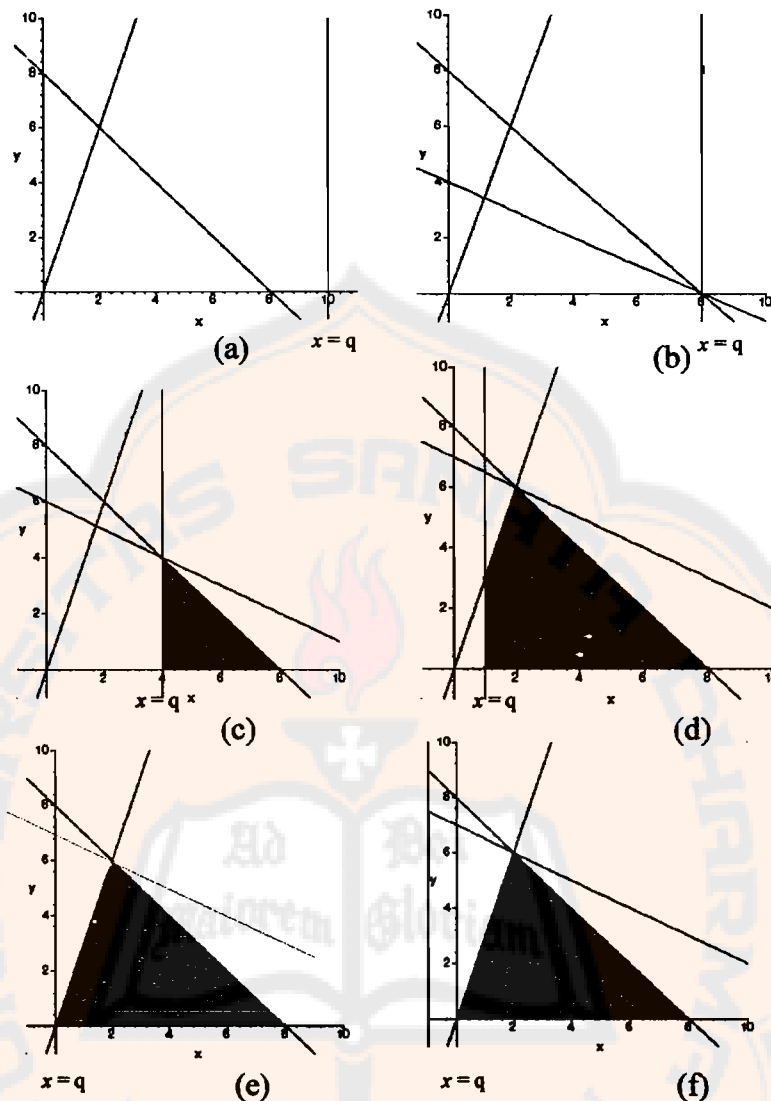
Ingin diketahui untuk nilai q berapa sajakah soal program linear ini mempunyai penyelesaian optimum serta berapakah nilai optimum yang sesuai. Semua kemungkinan nilai q beserta nilai optimum yang bersesuaian dengan q tersebut dapat dilihat pada Tabel 2.3.3 berikut.

No.	Nilai q	Bentuk Daerah Layak	Titik Optimum	Nilai Program
1.	$q > 8$	Tidak ada daerah penyelesaian (lihat gambar 2.9 (a))	-	-
2.	$q = 8$	Titik A (lihat gambar 2.9 (b))	(8, 0)	80
3.	$2 \leq q < 8$	Segitiga PAR (lihat gambar 2.9 (c))	(q, 8 - q)	$160 - 10q$
4.	$0 < q < 2$	Segiempat KABL (lihat gambar 2.9 (d))	(2, 6)	140
5.	$q = 0$	Segitiga OAB (lihat gambar 2.9 (e))	(2, 6)	140
6.	$q < 0$	Segitiga OAB (lihat gambar 2.9 (f)). Pada keadaan ini, kendala $x > q$ merupakan kendala yang berlebih.	(2, 6)	140

Tabel 2.3.3. Variasi nilai q beserta dengan nilai program yang bersesuaian

Dengan demikian, agar soal program linear tersebut mempunyai penyelesaian optimum, maka nilai q yang memenuhi adalah $q \leq 8$.

Gambar daerah layak dari masing-masing kemungkinan nilai q adalah sebagai berikut (Gambar 2.3.10).



Gambar 2.3.10. Kemungkinan nilai q dari contoh 2.3.9.

Contoh 2.3.10.

Dari contoh 2.3.8. telah diperoleh penyelesaian optimum soal program linear yang diketahui, yaitu $B(80,40)$ dengan nilai program $f = 40.000$. Apabila koefisien ongkos dari variabel x berubah menjadi p , harus ditentukan daerah nilai

agar penyelesaian optimum tidak berubah, meskipun nilai program mungkin berubah, agar tidak perlu menyusun perancangan yang baru.

Untuk menyelesaikan permasalahan ini, fungsi sasaran ditulis sebagai $f = px + 500y$. Karena koefisien lain tidak berubah maka daerah layak tidak berubah. Yang terpengaruh perubahan adalah gradien dari fungsi sasaran. Dalam soal semula $p = 250$, sehingga $m_{awal} = -\frac{1}{2}$. Nilai ini tercapai oleh batas kendala dengan $m_3 = -1$ dan kendala dengan $m_4 = -\frac{1}{4}$. Supaya penyelesaian optimum tidak berubah maka syarat ini harus tetap dipenuhi oleh permasalahan yang baru, sehingga:

$$m_3 \leq \frac{-P}{500} \leq m_4$$

Dengan demikian agar nilai optimum tidak berubah, maka nilai p haruslah memenuhi syarat:

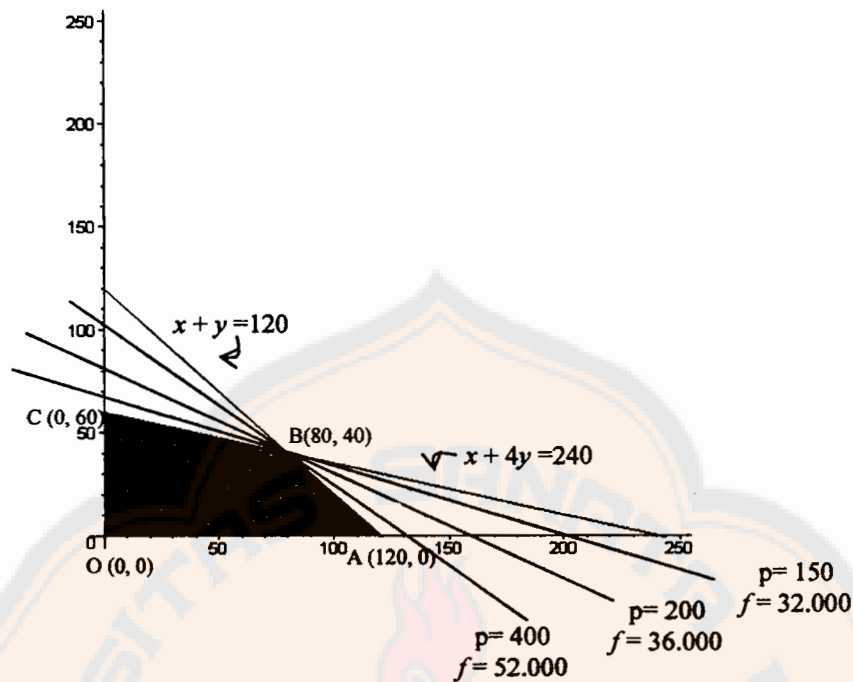
$$125 \leq p \leq 500$$

(saat $p = 125$ atau $p = 500$ akan timbul pilihan penyelesaian, akan tetapi titik B(80,40) tetap menjadi salah satu penyelesaian).

Dengan nilai program menjadi:

$$f = 80p + 20.000$$

Berikut ini disajikan gambar perubahan gradien fungsi yang mungkin terjadi (Gambar 2.3.11).



Gambar 2.3.11. Perubahan gradien fungsi sasaran

D. Sekilas Tentang Maple8 dan Maplet

Maplet adalah sebuah fasilitas *interface* grafis dari *Maple8* yang memungkinkan pengguna perangkat lunak *Maple8* untuk mengkombinasikan paket-paket dan prosedur-prosedur dalam *Maple8* dengan window dan dialog interaktif.

Sebuah *interface Maplet* dapat memfasilitasi pengguna untuk melakukan kegiatan interaktif tanpa menggunakan *syntax Maple8* yang belum tentu mudah dimengerti oleh setiap orang. Pengguna *Maplet*, misalnya siswa, dapat meng-klik tombol, menggeser slider, memasukkan nilai, memilih item menu, dan sebagainya, untuk melakukan modifikasi atau eksplorasi terhadap suatu paket *Maplet*. Dibanding hanya dengan menggunakan *Maple8* untuk membantu suatu proses pembelajaran, dengan menggunakan *Maplet*, seorang siswa mendapat

peluang lebih untuk berkonsentrasi pada pemahaman matematikanya daripada sibuk mengingat simbol-simbol dan bahasa *Maple8*. Dengan demikian sangat dimungkinkan pemanfaatan *Maplet* untuk membuat suatu paket pembelajaran, terutama paket pembelajaran yang berhubungan dengan ranah numerik.

Maplet ditulis dengan menggunakan *Maple8*. *Maplet* dapat dieksekusi dalam *worksheet Maple8* dengan *MapletViewer*. *MapletViewer* juga dapat digunakan secara langsung untuk menjalankan *Maplet* dari desktop.

Worksheet Maple8 (file *.mws) merupakan format termudah yang dapat digunakan untuk membuat *Maplet*. Modifikasi yang diperlukan untuk mengubah *Maple* menjadi *Maplet* adalah cukup dengan menyimpan *worksheet Maple8* sebagai file dalam type *Maplet* (file *.maplet).

Sebelum memahami mengenai *Maplet* lebih lanjut, berikut ini adalah beberapa terminologi yang sebaiknya dimengerti dulu oleh pembaca.

1. *Maplet*

Maplet adalah sekumpulan elemen, seperti elemen-elemen *window*, elemen-elemen *layout*, elemen-elemen dialog dan elemen-elemen perintah. Sebuah *Maplet* mengandung *window-window* dan *dialog-dialog*.

2. Penulis *Maplet* (*Maplet Author*)

Penulis *Maplet* adalah programmer yang menggunakan kode-kode *Maple* untuk membuat *Maplet*.

3. Pengguna *Maplet* (*Maplet User*)

Pengguna *Maplet* adalah seseorang yang berinteraksi dengan *Maplet*.

4. *Layout*

Layout adalah tampilan visual dari susunan elemen-elemen dalam *Maplet*.

5. *Window*

Window tidak sama dengan *Maplet*. *Window* adalah elemen tingkat teratas (*top level element*) dalam *Maplet*.

6. Elemen *Window* (*Window Elements*)

Elemen *window* adalah bagian-bagian yang membentuk suatu *window* dalam *Maplet*. Suatu *Maplet* dapat terdiri lebih dari satu *window*. Setiap *window* dapat mengandung beberapa elemen yang mengatur layout dan fungsi dari *window*.

7. Elemen *Body Window* (*Body Window Elements*)

Elemen *body window* adalah suatu kategori dari elemen yang berhubungan dengan elemen-elemen yang tampak dalam suatu *window*, kecuali *menubar* dan *toolbar*.

8. Paket *Maplet* (*Maple Package*)

Paket *Maplet* terdiri dari tiga subpaket: Elemen-elemen (*Elements*), Contoh-contoh (*Examples*), dan Tool (*Tools*); dan sebuah fungsi tingkat atas (*top level function*): Tampilan (*Display*).

9. Elemen (*Elements*)

Elemen adalah komponen individual yang digunakan untuk membuat sebuah *Maplet*. Macam-macam elemen, antara lain: *window*, tombol (*button*), *checkbox*. Setiap elemen memiliki halaman bantuan yang mendeskripsikan berbagai macam fungsi dari tiap-tiap elemen.

10. Tool

Tool adalah alat bantu bagi pengguna perangkat lunak *Maple8* untuk membuat sebuah *Maplet*.

11. Tampilan (*Display*)

Fungsi tampilan (*display*) digunakan untuk menampilkan (menjalankan) sebuah *Maplet*.



BAB III

EKSPLORASI FASILITAS *MAPLE* YANG MEMBANTU PEMBELAJARAN PROGRAM LINEAR DUA VARIABEL

Seperti telah diuraikan pada Bab I mengenai metode penulisan, skripsi ini bersifat studi pustaka dan eksplorasi. Oleh karena itu pada bab ini akan dikaji dan dieksplorasi bagaimana *Maple* dapat membantu pembelajaran program linear dua variabel dengan metode grafik.

A. Eksplorasi Fasilitas *Maple* untuk Program Linear

Berikut ini akan disajikan hasil eksplorasi perintah *Maple* yang dapat digunakan untuk memecahkan masalah Program Linear.

1. Perintah untuk Menggambar Daerah Penyelesaian dari Suatu Pertidaksamaan

Untuk menggambar daerah penyelesaian dari suatu pertidaksamaan digunakan perintah *inequal*. Sebelum menggunakan perintah untuk menggambar dalam *Maple*, terlebih dahulu harus dipanggil perintah "**with(Plots) :**".

Cara memanggil perintah menggambar *inequal* ini adalah *inequal(ineqs, xspec, yspec, options)*. Parameter *ineqs* menyatakan sebuah pertidaksamaan atau suatu sistem pertidaksamaan yang akan digambar, *xspec* menyatakan interval nilai x yang menjadi daerah gambar, dan *yspec* menyatakan interval y yang menjadi daerah gambar. *Options* dapat diisi dengan pilihan-pilihan berikut (Tabel 3.1.1)

Pilihan	Deskripsi
feasible region	daerah yang memenuhi sistem pertidaksamaan
excluded regions	daerah yang tidak memenuhi sistem pertidaksamaan
open lines	garis lurus yang menggambarkan batas dari suatu pertidaksamaan yang bertanda '<' atau '>'
closed lines	garis lurus yang menggambarkan batas dari suatu pertidaksamaan yang bertanda '≤' atau '≥'

Tabel 3.1.1. Pilihan untuk menggambar menggunakan perintah `inequal`

Selain beberapa pilihan dalam Tabel 3.1.1 di atas *Maple* juga menyediakan pilihan *color* untuk warna dan *thickness* untuk ketebalan garis.

Contoh 3.1.1:

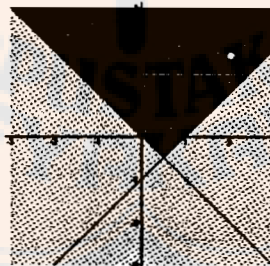
Berikut ini adalah contoh untuk menggambar daerah penyelesaian sistem pertidaksamaan linear $\{x + y > 0, x - y \leq 1\}$.

`> with(plots);`

Warning, the name `changecoords` has been redefined

`[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, scmilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot]`

`> inequal({ x+y>0, x-y<=1}, x=-3..3,y=-3..3, optionsexcluded=(color=blue, thickness= 2));`



2. Perintah Perhitungan untuk Program Linear

Ada dua macam perintah perhitungan dalam *Maple* yang akan dipakai dalam skripsi ini, yaitu perintah *maximize* dan *minimize*. Perintah *maximize*

digunakan untuk mencari nilai maksimum dari suatu masalah Program Linear. Sedangkan perintah *minimize* digunakan untuk mencari nilai minimum dari suatu masalah program linear. Untuk dapat menggunakan perintah ini, harus dipanggil terlebih dahulu perintah "**with(simplex);**".

Cara memanggil perintah *maximize* adalah *maximize(f,C)*, sedangkan cara memanggil untuk perintah *minimize* adalah *minimize(f,C)*. Dalam perintah ini *f* adalah fungsi sasaran yang akan dioptimalkan, dan *C* adalah kendala-kendala dari masalah program linear tersebut.

Contoh 3.1.2:

Berikut ini adalah contoh untuk memaksimalkan fungsi sasaran $f: x + y$, dengan kendala-kendala: $4x + 3y \leq 5$ dan $3x + 4y \leq 4$.

```
> with(simplex);
Warning, the protected names maximize and minimize have been redefined
and unprotected
```

```
[basis, convexhull, cterm, define_zero, display, dual, feasible, maximize, minimize, pivot, pivoteqn,
pivotvar, ratio, setup, standardize]
```

```
>maximize(x+y, {4*x+3*y<=5, 3*x+4*y<=4});
{x = 8/7, y = 1/7}
```

Meminimalkan fungsi sasaran $f: x + y$, dengan kendala-kendala: $4x + 3y \leq 5$, $3x + 4y \leq 4$, $x \geq 0$ dan $y \geq 0$

```
>minimize(x+y, {4*x+3*y<=5, 3*x+4*y<=4, x>=0, y>=0});
{x = 0, y = 0}
```

Dengan *subs*, *Maple* mampu mensubsitusikan titik optimum ke fungsi sasaran sehingga didapatkan nilai optimum. Cara pemanggilan perintah ini adalah

$subs(s1, \dots, sn, expr)$, dimana $s1, \dots, sn$ adalah variabel beserta dengan nilai yang akan disubstitusi, dan $expr$ adalah pernyataan aljabar dengan n variabel.

Contoh 3.1.3:

Berikut ini adalah contoh untuk mensubstitusikan $x = 1$ dan $y = 2$ ke dalam pernyataan $x + y$.

```
> subs (x=1, y=2, x+y);
3
```

Contoh 3.1.4:

Menyelesaikan Contoh 2.3.6 (halaman 29) dengan *Maple*.

Menggambar daerah penyelesaian dari kendala-kendala: $x \geq 0$, $y \geq 0$, $x + 4y \leq 240$, dan $x + y \leq 120$

```
> with (plots) :
Warning, the name changecoords has been redefined
> inequal( {x}>=0, y>=0, x+4*y<=240, x+y<=120},
x=-1..150, y=-1..150, optionexcluded=(color=yellow,
thickness= 2)):
```



Mencari titik optimum yang memaksimalkan fungsi sasaran $f = 250x + 500y$

```
> with(simplex):
> maximize(250*x + 500*y, {x>=0, y>=0, x + 4*y <=240,
  x + y<=120} );
      {x = 80, y = 40}
```

Mencari nilai maksimum:

```
> subs({x= 80, y= 40}, 250*x + 500*y);
      40000
```

Dengan demikian, titik (80, 40) adalah titik maksimal dengan nilai

$f_{\text{maksimal}} = 40.000$.

B. Eksplorasi Fasilitas *Maple* untuk Mendukung Penyusunan *Interface Maplet*

Dalam pembuatan *interface* ini diperlukan beberapa perintah-perintah dan dasar-dasar pemrograman dalam *Maple*. Perintah-perintah yang digunakan untuk mendukung pembuatan *interface*, meliputi:

1. Perintah Untuk Menyelesaikan Suatu Persamaan atau Sistem Persamaan

Untuk menyelesaikan suatu persamaan atau suatu sistem persamaan digunakan perintah *solve*. Untuk memanggil perintah ini, digunakan *solve(eqn, var)*. Pada bagian *eqn* dituliskan persamaan yang akan diselesaikan. Sedangkan pada bagian *var* dituliskan variabel yang dikehendaki.

Contoh 3.2.1:

Berikut ini adalah contoh untuk menyelesaikan persamaan $x + y = 5$ dalam variabel x .

```
> solve(x+y=5, x);
```

$$-y + 5$$

2. Perintah untuk Mengubah Bilangan ke Bentuk Desimal dengan Menentukan Banyak Tampilan Digit

Perintah *evalf* digunakan untuk menampilkan suatu bilangan dalam bentuk desimal dan mengatur banyaknya tampilan digit dari bilangan desimal tersebut. Ada dua macam cara memanggil perintah *evalf*, yaitu *evalf(expr, n)* dan *evalf[n](expr)*. Parameter *n* di sini berupa integer yang menunjukkan banyaknya digit yang ingin ditampilkan, dan *expr* adalah bilangan yang ingin diubah. Kedua macam bentuk urutan pemanggilan perintah ini memberikan keluaran yang sama.

Contoh 3.2.2:

Berikut ini adalah contoh untuk menampilkan bilangan *phi* dalam tiga digit.

```
> evalf(Pi, 3);
3.15
> evalf[3](Pi);
3.15
```

3. Perintah Untuk Mengevaluasi Sebuah Pernyataan Boolean

Untuk mengevaluasi sebuah pernyataan boolean digunakan perintah *evalb*. Hasil dari eksekusi perintah ini berupa nilai *true* (benar), *false* (salah) dan *failed* (gagal). Cara memanggil perintah *evalb* adalah *evalb(x)*, di mana *x* adalah pernyataan yang melibatkan hubungan antar operator (*=*, *<*, *>*, *<=*, *>=*, atau *<>*).

Contoh 3.2.3:

Berikut ini adalah contoh untuk memeriksa pernyataan $x=x$.

```
> evalb (x=x) ;
                                true
```

4. Perintah untuk Mencari Nilai Mutlak Dari Suatu Bilangan

Untuk mencari nilai mutlak dari suatu bilangan digunakan perintah *abs*. Cara memanggil perintah *abs* adalah *abs(x)*, di mana *x* adalah bilangan yang akan dicari nilai absolutnya.

Contoh 3.2.4:

Berikut ini adalah contoh untuk mencari nilai absolut dari -11.

```
> abs (-11) ;
                                11
```

5. Perintah untuk Menguji Jenis Pernyataan

Perintah *type* digunakan untuk menguji jenis pernyataan. Cara memanggil perintah *type* adalah *type(e, t)*, dimana *e* adalah pernyataan yang akan diuji, dan *t* adalah jenis pernyataan yang diujikan. Pernyataan akan diuji, apakah termasuk dalam jenis ' \leq ', '<', atau '*constant*'.

Contoh 3.2.5:

Penyelidikan pernyataan $a < b$, apakah pernyataan tersebut dapat berbentuk pernyataan ' \leq '.

Contoh 3.2.3:

Berikut ini adalah contoh untuk memeriksa pernyataan $x=x$.

```
> evalb(x=x) ;
true
```

4. Perintah untuk Mencari Nilai Mutlak Dari Suatu Bilangan

Untuk mencari nilai mutlak dari suatu bilangan digunakan perintah *abs*. Cara memanggil perintah *abs* adalah *abs(x)*, di mana *x* adalah bilangan yang akan dicari nilai absolutnya.

Contoh 3.2.4:

Berikut ini adalah contoh untuk mencari nilai absolut dari -11.

```
> abs(-11) ;
11
```

5. Perintah untuk Menguji Jenis Pernyataan

Perintah *type* digunakan untuk menguji jenis pernyataan. Cara memanggil perintah *type* adalah *type(e, t)*, dimana *e* adalah pernyataan yang akan diuji, dan *t* adalah jenis pernyataan yang diujikan. Pernyataan akan diuji, apakah termasuk dalam jenis ' \leq ', '<', atau '*constant*'.

Contoh 3.2.5:

Penyelidikan pernyataan $a < b$, apakah pernyataan tersebut dapat berbentuk pernyataan ' \leq '.

```
> type( a < b, `<=` );  
false
```

6. Perintah untuk Menampilkan Koefisien Suatu Variabel dalam Suatu Pernyataan Aljabar

Perintah *coeff* digunakan untuk menampilkan koefisien suatu variabel dalam suatu pernyataan aljabar. Cara memanggil perintah ini adalah *coeff(p,x)*. Parameter *p* menunjukkan polinomial dalam *x*, sedangkan *x* menunjukkan variabel dari sebuah pernyataan aljabar.

Contoh 3.2.6:

Menampilkan koefisien *x* dari pernyataan $2x + y$.

```
> coeff(2*x+y, x);
```

2

7. Perintah untuk Mengubah Suatu Relasi Pernyataan Menjadi Relasi Pernyataan Bentuk Lain

Untuk mengubah suatu relasi pernyataan menjadi relasi lain yang diinginkan digunakan perintah *convert*. Cara memanggil perintah ini adalah *convert(expr, form)*. Parameter *expr* adalah relasi pernyataan yang akan diubah, sedangkan parameter *form* adalah bentuk relasi yang dimaksud. Ada tiga macam pilihan yang dapat diisikan dalam parameter *form*, yaitu: *equality* (digunakan untuk mengubah ke bentuk relasi persamaan), *lessthan* (digunakan untuk

mengubah ke bentuk relasi kurang dari), dan *lessequal* (digunakan untuk mengubah ke bentuk relasi kurang dari sama dengan).

Contoh 3.2.7:

Berikut ini adalah contoh untuk mengubah suatu pertidaksamaan $x + y \leq 10$ menjadi persamaan $x + y = 10$.

```
> convert(x+y<=10,equality);
      x + y = 10
```

8. Perintah untuk Menampilkan Ruas Kiri atau Ruas Kanan Suatu Persamaan atau Pertidaksamaan

Perintah *lhs* digunakan untuk menampilkan ruas kiri suatu persamaan atau pertidaksamaan, sedangkan *rhs* digunakan untuk mencari ruas kanan suatu persamaan atau pertidaksamaan. Cara memanggil perintah *lhs* adalah *lhs(expr)*, sedangkan cara memanggil perintah *rhs* adalah *rhs(expr)*. Parameter *expr* pada kedua perintah tersebut menyatakan persamaan atau pertidaksamaan yang akan ditampilkan ruas kiri atau ruas kanannya.

Contoh 3.2.8:

Dimiliki suatu persamaan $y = ax^2 + b$. Persamaan ini diberi nama *e*.

```
> e := y = a*x^2 + b;
      e := y = a x^2 + b
```

Berikut ini adalah contoh untuk menampilkan ruas kiri dari *e*.

```
> lhs(e);
      y
```


Berikut ini adalah contoh untuk menampilkan ruas kanan dari e
> rhs (e) ;

$$ax^2 + b$$

9. Perintah untuk Mencari Gradien Suatu Persamaan Garis

Untuk mencari gradien suatu persamaan garis digunakan perintah *slope*. Untuk dapat menggunakan perintah ini, harus didefinisikan terlebih dahulu perintah “with(student):”. Cara memanggil perintah ini adalah *slope(equation)*. Parameter *equation* adalah persamaan garis dalam $y=f(x)$ yang akan dicari gradiennya.

Contoh 3.2.9:

Berikut ini adalah contoh untuk mencari gradien dari persamaan garis $y = 3x + 5$

> with(student) ;

[D, Diff, Doubleint, Int, Limit, Lineint, Product, Sum, Tripleint, changevar, completesquare, distance, equate, integrand, intercept, inparts, leftbox, leftsum, makeproc, middlebox, middlesum, midpoint, powsubs, rightbox, righisum, showtangent, simpson, slope, summand, trapezoid]

> slope(y = 3*x+5) ;

3

10. Perintah untuk Melakukan Operasi Antar Himpunan

Perintah *union* digunakan untuk menggabungkan dua buah himpunan. Sedangkan untuk mengecek apakah suatu himpunan merupakan himpunan bagian dari himpunan lain digunakan perintah *subset*.



Untuk perintah *union*, cara memanggil perintahnya adalah *s1 union s2*. Parameter *s1* adalah himpunan pertama yang akan digabungkan dan parameter *s2* adalah himpunan kedua yang akan digabungkan.

Contoh 3.2.10:

Berikut ini adalah contoh untuk menggabungkan himpunan {a,b} dan {b,c}.

```
> {a,b} union {b,c};
      {a,b,c}
```

Untuk perintah *subset*, cara memanggil perintahnya adalah *s1 subset s2*. Di sini, parameter *s1* adalah himpunan yang akan diperiksa keanggotaannya terhadap himpunan *s2*.

Contoh 3.2.11:

Berikut ini adalah contoh untuk memeriksa apakah himpunan {a,b} merupakan himpunan bagian dari himpunan {a,b,c,d}.

```
> {a,b} subset {a,c,b,d};
      true
```

11. Perintah untuk Menggambar

Sebelum menggunakan perintah-perintah untuk menggambar dalam *Maple*, terlebih dahulu harus didefinisikan perintah “**with(Plots):**”. Ada empat macam perintah untuk menggambar dalam *Maple* yang akan dibahas dalam skripsi ini, yaitu: perintah *plot*, perintah *textplot*, perintah *pointplot*, dan perintah *inequal*. Dalam perintah-perintah tersebut terdapat beberapa jenis pilihan yang

dapat digunakan untuk menampilkan gambar sesuai dengan keinginan perancang.

Beberapa pilihan yang dapat dipakai ditampilkan dalam Tabel 3.2.1 berikut.

Pilihan	Manfaat
<i>Color</i>	menentukan warna dari kurva yang akan digambar
<i>Thickness</i>	menentukan ketebalan garis kurva yang digambar
<i>Symbol</i>	Menentukan jenis poin dalam gambar: BOX, CROSS, CIRCLE, POINT, dan DIAMOND
<i>SymbolSize</i>	Menentukan ukuran simbol yang digunakan dalam gambar
<i>Font</i>	Menentukan jenis huruf untuk objek-objek teks dalam gambar.

Tabel 3.2.1. Tabel pilihan perintah menggambar dalam *Maple*

Berikut ini akan dibahas manfaat dan cara pemanggilan perintah dari masing-masing perintah untuk menggambar dalam *Maple*.

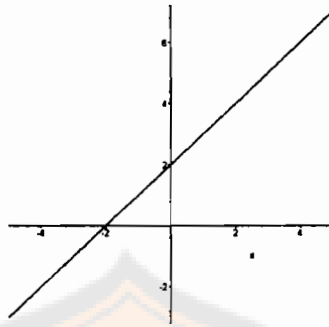
a) Perintah untuk menggambar grafik suatu fungsi

Perintah *plot* digunakan untuk menggambar grafik suatu fungsi. Cara untuk memanggil perintah *plot* adalah *plot(f, h, v, opt1, opt2,...)*. Di mana *f* menunjukkan fungsi yang akan digambar, *h* menyatakan interval dari sumbu *x*, *v* menyatakan interval dari sumbu *y* dan *opt1, opt2, ...,* dapat diisi dengan pilihan-pilihan seperti dalam Tabel 3.2.1.

Contoh 3.2.12:

Berikut ini adalah contoh untuk menggambar grafik dari $y = x + 2$, dalam interval $x = [-5, 5]$. Grafik yang tergambar diinginkan berwarna merah dengan ketebalan 2.

```
> plot(x+2, x=-5..5, color=red, thickness=2);
```



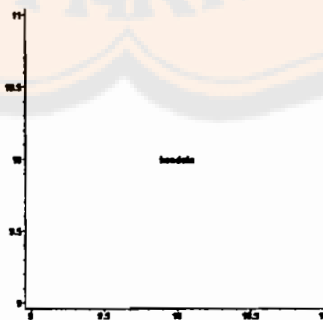
b) Perintah untuk menggambar teks pada bidang gambar

Untuk menggambar teks pada bidang gambar digunakan perintah *textplot*. Untuk perintah ini, cara memanggil perintahnya adalah *textplot(L,options)*. Dalam perintah ini, *L* merupakan teks yang akan ditampilkan dan dinyatakan sebagai urutan atau himpunan, sedangkan *options* pilihan pengaturan seperti tercantum dalam Tabel 3.2.1.

Contoh 3.2.13:

Berikut ini adalah contoh untuk meletakkan teks 'kendala' pada bidang xy, dengan koordinat (10,10).

```
> textplot([10,10,`kendala`],font=[HELVETICA,
BOLD,10]);
```



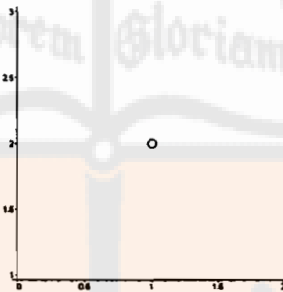
c) Perintah untuk membuat titik pada bidang gambar 2 dimensi

Perintah *pointplot* digunakan untuk membuat titik pada bidang gambar 2 dimensi. Cara memanggil perintah *pointplot* adalah *pointplot(L, options)*. Dalam perintah ini, *L* menyatakan koordinat titik yang akan digambar dan dinyatakan sebagai urutan, sedangkan *options* adalah pilihan pengaturan seperti dalam Tabel 3.2.1.

Contoh 3.2.14:

Berikut ini adalah contoh untuk menggambar titik (1,2) pada bidang xy

```
> pointplot([[1,2]], symbol=CIRCLE, color=red,  
symbolsize=30);
```

**C. Pemrograman dengan Maple**

Berikut ini akan disajikan beberapa pemahaman yang dibutuhkan dalam melakukan pemrograman dalam Maple.

1. Prosedur (Procedure)

Prosedur dalam Maple adalah suatu ekspresi yang valid dan dapat diberi nama. Dengan nama tersebut prosedur dapat dieksekusi. Cara untuk memanggil

sebuah prosedur adalah *proc (argseq) local nseq; global nseq; statseq end proc*. Parameter *argseq* menyatakan nama parameter masukan, *nseq* menyatakan nama dari variabel lokal maupun global, dan *statseq* menyatakan isi prosedur. Variabel yang berlaku untuk beberapa prosedur dalam *Maple* dalam satu *worksheet* disebut *variabel global*. Variabel yang didefinisikan dalam sebuah prosedur dan dikenali oleh *Maple* hanya dalam prosedur tersebut disebut *variabel lokal*.

Contoh 3.3.1:

Berikut ini contoh prosedur untuk mencari nilai maksimum masalah Program Linear menggunakan paket *simplex* dengan nama 'Penyelesaian_PL'. Masukannya adalah a, b, c, dan d sebagai batas bidang gambar. Isi prosedur adalah mencari nilai maksimum serta menggambar daerah layaknya.

```
> with(simplex):
with(plots):

Warning, the name display has been redefined
Warning, the name display has been redefined

> Penyelesaian_PL:=proc(a,b,c,d)
  local maksimum, nilai;
maksimum:=maximize(sasaran,kendala);
nilai:=subs(maksimum,sasaran);
print(`Daerah Layaknya:`);
print(inequal(kendala,x=a..b,y=c..d,
optionsfeasible=(color=red),optionsexcluded=(color=white),
optionsclosed=(color=coral),thickness=2));
print(`Titik maksimumnya`);
print(maksimum);
print(`nilai maksimumnya`);
print(nilai)
end;
```

```
Penyelesaian_PL := proc (a, b, c, d)
  local maksimum, nilai;
```

```

maksimum := maximize(sasaran,kendala); print('Daerah Layaknya:');
nilai:=subs(maksimum, sasaran)
print(inequal(kendala,x = a .. b,y = c .. d,optionsfeasible = (color =
red),optionsexcluded = (color = white),optionsclosed = (color = coral),thickness =
2));
print('Titik maksimumnya')
print(maksimum)
print('nilai maksimumnya')
print(nilai)
end proc

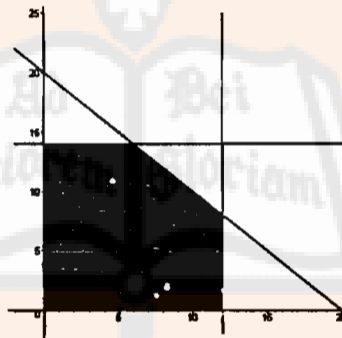
```

Berikut ini contoh penggunaan prosedur Penyelesaian_PL, untuk $a = -2$, $b = 20$, $c = -2$, dan $d = 25$,

```

> kendala := {x+y<=20, x>=0, x<=12, y>=0, y<=14};
kendala := {x + y ≤ 20, 0 ≤ x, x ≤ 12, 0 ≤ y, y ≤ 14}
> sasaran := 2*x+10*y+11;
sasaran := 2x + 10y + 11
> Penyelesaian_PL(-2,20,-2,25);
Daerah Layaknya:

```



Titik maksimumnya
 $\{y = 14, x = 6\}$
 nilai maksimumnya
 163

2. Pernyataan Keadaan

Suatu pernyataan keadaan memiliki bentuk:

```

If      keadaan
Then statement 1
Else statement 2
End if;

```

Statement di sini adalah suatu pernyataan. Pernyataan keadaan harus ditutup dengan '**end if**'.

3. Ekspresi Boolean

Beberapa operator relasi dalam *Maple* yang dibutuhkan dalam skripsi ini adalah < (kurang dari), > (lebih dari), <= (kurang dari sama dengan), >= (lebih dari sama dengan), dan <> (tidak sama dengan)

Selain operator tersebut, dibutuhkan juga operator boolean 'and' (dan) dan 'or' (atau).

4. Perulangan (*Loop*)

Perulangan digunakan untuk melakukan perhitungan yang sama atau mirip selama sekian waktu. Ada dua macam perulangan, yaitu: perulangan "**for..to..do..**" dan perulangan "**while..do..**".

a) Perulangan *for.. to .. do ..*

Statemen perulangan *for.. to .. do ..* berbentuk:

```
For   var   from num1 to num2 do statment  
End do;
```

Parameter *var* merupakan variabel yang didefinisikan dalam perintah tersebut, *num1* dan *num2* merupakan batas interval perintah tersebut dilakukan, dan *statmen* adalah perintah yang akan dilakukan selama syarat perulangan dipenuhi.

b) Perulangan *while .. do ..*

Sebuah pernyataan perulangan *while .. do ..* mempunyai bentuk:


```
While kondisi do statetment  
End do;
```

Parameter *kondisi* adalah syarat di mana perintah dijalankan, sedangkan *statement* adalah perintah yang akan dilakukan selama kondisi terpenuhi.

Dalam perulangan *while .. do ..*, *Maple* akan menguji keadaan secara berulang dan berhenti saat syarat kondisi tidak terpenuhi lagi.

D. Eksplorasi Fasilitas *Maplet*

Dalam subbab berikut akan disajikan hasil eksplorasi dari beberapa fasilitas *Maplet* yang digunakan dalam pembuatan *interface*. Namun sebelumnya, sebagai gambaran umum, akan disajikan beberapa aturan (kesepakatan) yang akan dipakai untuk menulis *Maplet*. Aturan-aturannya adalah sebagai berikut.

1. Gunakan satu definisi elemen per baris. Jika definisinya pendek, dalam satu baris dapat dipakai untuk mendefinisikan lebih dari satu elemen.
2. Tulis menjorok kekanan setiap sub-urutan elemen atau daftar bersarang.
3. Letakkan kurung tutup, kurung dari elemen, atau daftar bersarang pada indenasi yang sama sebagai pembuka.
4. Letakkan tanda kutip di sekitar simbol-simbol atau nama yang digunakan sebagai nama pilihan atau referensi. Jika suatu simbol atau nama telah dipakai, maka perintah tidak dapat terdefinisi lagi.
5. Dalam prosedur, gunakan statemen *use* atau statemen *with* menjalankan kode. Statemen *use* akan berlaku untuk satu prosedur tertentu. Sedangkan statemen *with* akan berlaku untuk semua prosedur dalam satu *worksheet* *Maple*.

Untuk memulai menulis Maplet, perintah-perintah awal yang harus diketikkan adalah:

```
> restart;  
> with (Maplets [Elements] );
```

Contoh 3.4.1:

Berikut ini adalah contoh Maplet yang menampilkan sebuah window berisi teks 'Teddy Suparyanto'.

```
> restart;  
> with (Maplets [Elements] ) :  
> Maplets [Display] ( Maplet ( ["Teddy Suparyanto"] ) ) ;
```

Tampilan yang akan muncul adalah sebagai berikut (Gambar 3.4.1).



Gambar 3.4.1. Contoh sebuah window

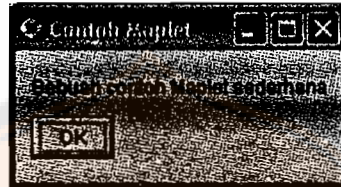
Saat [X] (ikon close) pada pojok window di-klik, maplet akan tertutup dan tampilan akan kembali ke *worksheet Maple*. Ikon close ini dapat divariasikan dengan pengatur window, misalkan menggunakan elemen tombol.

Berikut ini akan disajikan beberapa fasilitas dalam *Maplet* yang dipakai dalam pembuatan *interface* dalam skripsi ini.

1. Elemen *Window*

Sebuah *window* tidak sama dengan *Maplet*, tetapi merupakan salah satu bagian dalam *Maplet*. Suatu *Maplet* dapat mengandung lebih dari satu *window*. Setiap *window* dapat memuat banyak elemen yang dapat digunakan untuk

mengatur layout dan fungsi dari *window*, yang disebut Elemen *Window*. Sebuah *window Maplet* bisa terdiri dari *title bar*, *optional menu bar*, *optional toolbar*, dan *bodi*.



Gambar 3.4.2 sebuah *window* dengan beberapa elemen *bodi window*

Adapun cara untuk memanggil perintah ini adalah `window[refID](opts)`. Parameter *opts* merupakan pilihan yang dapat diisi dengan *defaultbutton*, *height*, *layout*, *menubar*, *reference*, *resizable*, *title*, *toolbar*, *width*, *xcoord*, atau *ycoord*, dan *refID* adalah nama atau string yang memberikan referensi pada suatu elemen.

Tabel 3.4.1 berikut ini merupakan penjelasan dari masing-masing pilihan.

Pilihan	Deskripsi
<i>defaultbutton</i>	tombol yang diaktifkan saat ENTER ditekan. Jika pilihan <i>defaultbutton</i> tidak ditentukan secara khusus, maka tombol pertama pada <i>window</i> adalah tombol setingan awal.
<i>height</i>	merupakan ukuran ketinggian <i>window</i> dalam satuan piksel
<i>layout</i>	elemen <i>layout</i> atau <i>reference</i> (<i>name</i> atau <i>string</i>), yaitu <i>layout window</i> atau referensi dari <i>layout</i> .
<i>menubar</i>	elemen <i>MenuBar</i> atau referensi elemen <i>MenuBar</i> (<i>name</i> atau <i>string</i>), yaitu sebuah bar menu atau referensi dari bar menu dalam <i>window</i>
<i>reference</i>	yaitu sebuah referensi elemen <i>Window</i>
<i>resizable</i>	menentukan apakah ukuran <i>window</i> dapat diubah-ubah. Pengaturan awalnya bernilai benar
<i>title</i>	<i>string</i> atau <i>symbol</i> , yaitu judul (<i>title</i>) yang terlihat pada titel bar <i>window</i> . Judul <i>window</i> menurut pengaturan awalnya adalah <i>Maplet</i> .
<i>toolbar</i>	elemen <i>ToolBar</i> atau referensi untuk elemen <i>ToolBar</i> (<i>name</i> atau <i>string</i>), merupakan sebuah referensi untuk <i>toolbar</i>
<i>width</i>	menunjukkan ukuran lebar dari <i>window</i> dalam satuan piksel
<i>xcoord</i>	merupakan pengaturan letak <i>window</i> dari sisi kiri layar dalam satuan piksel. Pengaturan awalnya adalah berada di tengah
<i>ycoord</i>	merupakan pengaturan letak <i>window</i> dari sisi atas layar dalam satuan piksel. Pengaturan awalnya adalah berada di tengah

Tabel 3.4.1. Pilihan untuk menggambar mengatur elemen *bodi*

2. Elemen Body Window

Berikut akan disajikan beberapa elemen bodi *window* yang akan dipakai dalam pembuatan *iterface* program linear ini.

a) Tombol (*Button*)

Tombol didefinisikan dengan menggunakan elemen *Button* yang mengandung dua informasi:

- (i) Teks yang tampil pada tombol.
- (ii) Aksi yang terjadi saat tombol di-klik



Gambar 3.4.3 *Button* (tombol)

Cara untuk memanggil perintah ini adalah *Button[refID](opts)*. Parameter *opts* menyatakan pilihan *background* (warna tombol), *caption* (tulisan dalam tombol), *enabled* (dapat tidaknya sebuah tombol diklik), *font*, *foreground* (warna teks dalam tombol), *onclick* (aksi yang diberikan pada tombol setelah diklik), dan *refID* yang menyatakan nama dari tombol.

b) *RadioButton*

RadioButton digunakan untuk memberikan beberapa alternatif pilihan dalam *Maplet* kepada pengguna. Pengguna hanya dapat memilih satu diantara sekian pilihan yang diberikan (hanya ada satu nilai benar untuk semua pilihan).

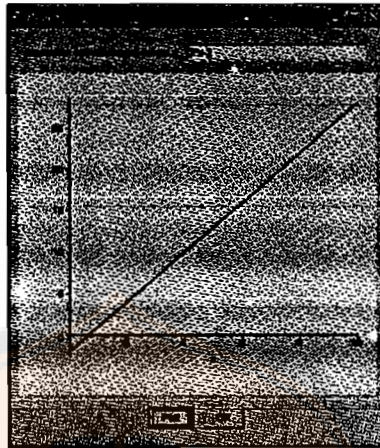


Gambar 3.4.4 *RadioButton*

Pilihan *filled* digunakan untuk menentukan jejak slider diperlihatkan atau tidak. Pilihan *lower* digunakan untuk menetapkan nilai terendah dari slider, dan pilihan *upper* digunakan untuk menetapkan nilai tertinggi dari slider, sedangkan untuk menetapkan nilai awal dari slider digunakan pilihan *value*. Pilihan *majorticks* mendefinisikan skala yang memuat angka, dan *minorticks* mendefinisikan skala antara yang tidak berangka. Sebuah elemen *Slider* dapat mengandung elemen aksi atau perintah untuk menentukan pilihan *onchange*. Pilihan *orientation* menunjukkan arah slider. Pilihan *showlabel* digunakan untuk menentukan apakah nilai majorstick akan ditampilkan pada slider, pilihan *showtick* digunakan untuk menentukan apakah minorstick akan ditampilkan pada slider, sedangkan pilihan *snaptick* digunakan untuk menetapkan nilai minorstick terdekat yang akan dituju saat mouse dilepas.

e) **Plotter**

Elemen plotter dapat menampilkan plot dua dimensi ataupun plot tiga dimensi. Cara untuk memanggil perintah plotter adalah *Plotter[refID](opts)*. Di mana *RefID* merupakan nama dari plotter dan *opts* merupakan pilihan dalam plotter. Pilihan-pilihan dalam plotter meliputi *background*, *reference*, *height*, *value*, dan *width*.

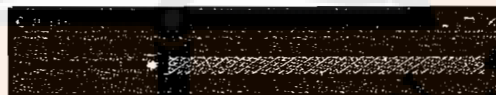


Gambar 3.4.7 Plotter

Pilihan *height* digunakan untuk menunjukkan ketinggian plot dalam satuan piksel, sedangkan pilihan *width* digunakan untuk menetapkan lebar daerah plot dalam satuan piksel. Pilihan *value* merupakan hasil plot dari *Maple* yang akan ditampilkan dalam plotter.

f) *TextField*

Sebuah *TextField* dapat digunakan untuk menampilkan informasi kepada pengguna maupun meminta informasi dari pengguna.

Gambar 3.4.8 *TextField*

Cara untuk memanggil perintah *TextField* adalah *Textfield[refID](opts)*. Di mana *RefID* merupakan nama dari *TextField* dan *opts* merupakan pilihan dalam *TextField*. Pilihan-pilihan dalam *TextField* adalah *background*, *caption*, *font*, dan *foreground*.

3. Elemen Menu

Sebuah *MenuBar* dapat memuat beberapa elemen *Menu*. Cara untuk memanggil perintah elemen menu adalah *MenuBar[refID](opts, element_content)*. Di mana *RefID* merupakan nama dari elemen menu, *element_content* merupakan sembarang jumlah elemen *Menu* dan *opts* merupakan pilihan dalam menu.



Gambar 3.4.9 Menu

4. Elemen Perintah

Berikut ini merupakan elemen perintah dalam penyusunan *Interface* dengan *Maplet*.

a) *Close Window*

Perintah *Close Window* digunakan untuk menutup suatu *window*. Perintah ini tidak dapat memuat perintah lain. Cara untuk memanggil perintah *Close Window* adalah *CloseWindow(opts)*. Perintah ini biasanya dieksekusi dengan menggunakan bantuan tombol. Saat sebuah tombol yang diberi perintah *Close Window* di-klik, maka *window* akan tertutup.



Gambar 3.4.10 Window dengan tombol perintah *Close Window*

b) Evaluate

Perintah *evaluate* digunakan untuk melakukan perhitungan aljabar dalam *Maple*. Cara untuk memanggil perintah *evaluate* adalah *Evaluate(opts,args)* . Parameter *opts* merupakan pilihan yang dapat diisi dengan salah satu dari *option* (pilihan dari tujuan evaluasi), *target* (tujuan dari evaluasi), atau *waitforresult* (pilihan apakah hasil akan segera muncul atau tidak). Sama halnya dengan perintah *Close Window* perintah ini dieksekusi dengan menggunakan bantuan tombol.

c) Run Window

Perintah *RunWindow* digunakan untuk menampilkan elemen-elemen *window*. Cara untuk memanggil perintah *Run Window* adalah *RunWindow(opts)*.

d) Set Option

Perintah *SetOption* digunakan untuk mengatur pilihan nilai dari *Maplet*. Cara untuk memanggil perintah *SetOption* adalah *SetOption(opts, argument)*.

e) Shut Down

Perintah *ShutDown* digunakan untuk mengakhiri *Maplet*. Cara memanggil perintah ini adalah *ShutDown(opts)*. Parameter *opts* diisi dengan pilihan-pilihan untuk menentukan aksi yang terjadi saat *Maplet* diakhiri. Jika pilihan ini tidak ditentukan secara khusus, maka setelah *Maplet* diakhiri, tampilan akan kembali ke *worksheet Maple*.

f) *Display*

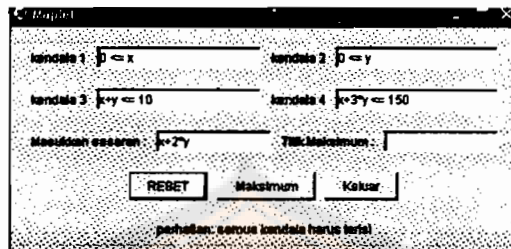
Perintah *display* digunakan untuk menampilkan (menjalankan) sebuah *Maplet*. Urutan pemanggilan perintahnya adalah *Display(maplet)*.

Contoh 3.4.2:

Berikut ini contoh penggunaan *display*, *evaluated*, dan *setoption* dalam *Maplet* untuk mencari titik maksimum suatu masalah Program Linear.

```
> with(simplex);
with(Maplets[Elements]):
Maplets[Display](
Maplet(
[
["kendala
1",TextField['B1'](10,value='x>=0'),"kendala
2",TextField['B2'](10,value='y>=0')],
["kendala
3",TextField['B3'](10,value='x+y<=10'),"kendala
4",TextField['B4'](10,value='x+3*y<=150')],
["Masukkan                               sasaran
:",TextField['B5'](10,value='x+2*y'),"Titik   Maksimum
:",TextField['B6'](10,editable=false)],
[
Button("RESET", 'onclick' = 'A1'),
Button("Maksimum", Evaluate('B6' = 'maximize(B5,
{B1,B2,B3,B4 }'))),
Button("Keluar", Shutdown(['B2']))
],
["perhatian: semua kendala harus terisi"]
]
,
Action[A1](SetOption('target' = 'B1', 'value' =
""),SetOption('target'='B2', 'value'=
""),SetOption('target'='B3', 'value'=
""),SetOption('target'='B4', 'value'=
""),SetOption('target'='B5', 'value'=
""),SetOption('target' = 'B6', 'value' = ""))
));
```

Hasil eksekusi:



Gambar 3.4.11 Window hasil eksekusi Contoh 3.4.2

5. Elemen Layout

Beberapa elemen *Layout* yang dipakai dalam pembuatan *interface* program linear ini adalah:

a) **Daftar Bersarang (*Nested List*)**

Sebuah *Maplet* dapat didefinisikan dengan menggunakan struktur layout boks. Struktur ini didasarkan pada konsep *Maple* tentang daftar bersarang (*nested lists - list of lists*). Dalam *Maple*, suatu daftar adalah barisan terurut dari ekspresi yang diapit dengan kurung siku ([]).

Contoh 3.4.3:

```
> List := [1, 5, 7];
List := [1, 5, 7]
```

Sedangkan, sebuah daftar bersarang adalah sebuah barisan terurut dari suatu ekspresi yang diapit dengan kurung siku, di mana ekspresi tersebut berupa daftar.

Contoh 3.4.4:

```
NestedList := [1, [2,3], [4, 5,6], 7, 8, [9,10]];  
NestedList := [1, [2, 3], [4, 5, 6], 7, 8, [9, 10]]
```

b) Layout Boks (*Box Layout*)

Suatu *Box Layout* menyusun objek-objek dalam kolom dan baris.

Kolom dan baris tersebut dapat disusun secara bersarang.

Contoh 3.4.5:

```
Maplets[Display]( Maplet(  
    Window(["A", [{"B", "C"}, "D", [{"E", "F", "G"}],  
    "H", "I"], "J"], Button("OK", Shutdown()))  
) );
```

Hasil eksekusi:



Gambar 3.4.12 *Window* hasil eksekusi Contoh 3.4.5

Selain *BoxLayout*, masih ada jenis elemen layout yang lain, yaitu: *GridLayout*.

Namun karena *GridLayout* tidak dipakai dalam skripsi ini, maka tidak akan dibahas.

c) *BoxRow*

Elemen *BoxRow* digunakan untuk pengaturan horisontal dalam sebuah *BoxLayout*. Cara untuk memanggil perintah *BoxRow* adalah *BoxRow(opts)*. Parameter *opts* dapat diisi dari salah satu pilihan dalam *BoxRow*, yaitu *halign*, *hscroll*, *valign*, *value*, atau *vscroll*.

Pilihan *halign* digunakan untuk pengaturan layout horisontal. Pengaturan awalnya adalah *center*. Sedangkan pilihan *hscroll* digunakan untuk menentukan kapan suatu scroll bar horisontal tampak pada *BoxRow*. Pengaturan awalnya adalah *never*. Pilihan *valign* digunakan untuk menentukan aligmen vertikal dari isi. Pengaturan awalnya adalah *center*. Pilihan *vscroll* digunakan untuk menentukan kapan sebuah scroll bar vertikal tampak pada *BoxRow*. Pengaturan awalnya adalah *never*.

d) *Box Coloumn*

Untuk menentukan pengaturan vertikal sebuah *BoxLayout*, digunakan elemen *BoxColumn*. Cara untuk memanggil perintah *BoxColoumn* adalah *BoxColumn(opts, element_content)*. Parameter *opts* dapat berupa pilihan untuk *background*, *border*, *caption*, *halign*, *inset*, *spacing*, *valign*, *visible*, dan *vscroll*. Sedangkan parameter *element_content* merupakan sembarang bilangan dari elemen *BoxColumn*.

Pilihan *border* digunakan untuk menentukan apakah *BoxColoumn* akan diberi border atau tidak. Defaultnya bernilai *false*.

BAB IV

PENYUSUNAN *INTERFACE* UNTUK Mendukung PEMBELAJARAN PROGRAM LINEAR DUA VARIABEL

Seperti yang telah dipaparkan pada Bab I, tujuan penulisan skripsi ini adalah menyusun sebuah *interface* dengan *Maplet* untuk membantu pembelajaran pokok bahasan Program Linear dua variabel. *Interface* yang dirancang diharapkan dapat berperan sebagai *tool* bagi pengguna. Agar tujuan penyusunan *interface* ini dapat tercapai, maka penyusunannya didasarkan pada prinsip-prinsip pembelajaran, dengan urutan langkah seperti yang telah diuraikan pada Bab II. Berikut ini uraian dari masing-masing langkah yang dilakukan dalam penyusunan *interface* ini.

A. Perencanaan Awal

Dalam perencanaan awal, telah dilakukan dua hal, yaitu indentifikasi tujuan, kebutuhan belajar dan masalah yang muncul dalam pembelajaran, serta analisis karakteristik dan lingkungan siswa yang akan menggunakan dan materi yang akan dikembangkan.

Dalam mempelajari pokok bahasan Program Linear dua variabel, seorang siswa atau mahasiswa membutuhkan kemampuan untuk melakukan analisa masalah, kemampuan untuk menggambar grafik, dan kemampuan visual untuk membayangkan bentuk daerah penyelesaian dari masalah Program Linear yang dihadapi. Oleh karena itu, dalam membantu siswa atau mahasiswa mempelajari

pokok bahasan Program Linear ini, seorang guru atau dosen diharapkan dapat memberikan banyak contoh penyelesaian masalah. Untuk keperluan itu, seorang guru atau dosen perlu membuat banyak gambar-gambar daerah penyelesaian di papan tulis maupun dengan menggunakan transparansi. Hal ini tidaklah mudah, sebab selain membutuhkan banyak waktu untuk menggambar, kadangkala ketepatan dalam menggambar juga menjadi kendala. Gambar yang kurang tepat akan membuat siswa atau mahasiswa kurang dapat memahami konsep yang diberikan.

Dari segi siswa dan mahasiswa, hal ini juga terjadi. Untuk dapat memahami konsep dengan lebih mendalam, mereka perlu melakukan banyak latihan. Dalam latihan tersebut, mereka juga dituntut untuk dapat menyelesaikan masalah program linear sampai dengan analisa grafik penyelesaian. Namun karena waktu yang dibutuhkan untuk menggambar grafik cukup lama, maka mereka seringkali hanya terfokus pada penggambaran grafik dan kurang memahami konsep penyelesaian masalah program linear itu sendiri.

Hal ini semakin terasa di jenjang perguruan tinggi, khususnya pada subpokok bahasan masalah sensitivitas. Untuk mempelajari masalah sensitivitas pergeseran kendala, seorang mahasiswa harus mampu membayangkan bagaimana bentuk daerah layak dan bagaimana keadaan titik optimum yang baru ketika ada kendala yang berubah pada interval tertentu. Begitu juga pada masalah sensitivitas pergeseran fungsi sasaran. Mahasiswa harus dapat membayangkan posisi-posisi garis senilai yang memenuhi syarat saat terjadi perubahan nilai. Biasanya untuk membantu pemahaman dan memperoleh bayangan keadaan, maka digambarlah

grafik secara berulang-ulang dalam berbagai macam kondisi. Ketika hal ini dilakukan, kendala yang dihadapi pun masih sama, yaitu masalah waktu dan ketepatan gambar.

Dilihat bahwa ada satu faktor siswa dan guru yang dapat membantu untuk mengatasi masalah ini, yaitu mulai berkembangnya pembelajaran berbantuan komputer di sekolah dan meningkatnya kemampuan penggunaan komputer di masyarakat. Terlebih seperti telah dikemukakan pada Bab II (halaman 7), rambu-rambu No. 7 Kurikulum 2004 bidang studi Matematika memperbolehkan dan menganjurkan sekolah untuk menggunakan teknologi seperti kalkulator, komputer, alat peraga, atau media lainnya untuk semakin meningkatkan efektifitas pembelajaran. Karena itu mulai dipikirkan perancangan sebuah *interface* yang dapat membantu pembelajaran program linear dua variabel.

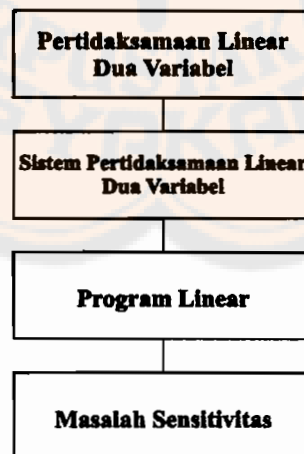
Sudah ada *interface* maupun fasilitas yang dirancang untuk membantu pembelajaran program linear ini, namun *interface* tersebut hanya menampilkan hasil penyelesaian tanpa menampilkan grafik penyelesaian, seperti *interface* QSB ataupun fasilitas *solve* pada *MsExcell*. Dijumpai sebuah *interface* program linear yang dapat menampilkan grafik penyelesaian, namun berbahasa asing dan aturan pemanfaatannya cenderung kompleks karena *interface* ini tidak dirancang untuk pembelajaran namun untuk bidang ekonomi, yaitu *ExcellQM*. *Interface* ataupun fasilitas semacam ini tidak dapat memberikan kesempatan kepada siswa untuk melakukan eksplorasi untuk pemahaman konsep, sehingga proses pembelajaran menjadi kurang baik.

Oleh karena itu *interface* yang akan dirancang akan berbahasa Indonesia, mampu menampilkan proses, dan mampu memberikan kesempatan pada siswa untuk melakukan eksplorasi. *Interface* ini akan dirancang sebagai *tool* bagi siswa dan guru. Sehingga dapat digunakan secara klasikal maupun secara mandiri. *Handout* dari *interface* ini pun dirancang untuk dapat digunakan oleh siswa dan guru.

Interface ini tidak akan disusun sebagai alat hitung, namun hanya untuk membantu pemahaman konsep bagi siswa dan mahasiswa. Hal ini dimaksudkan untuk mengurangi ketergantungan siswa dan mahasiswa dalam pemakaian alat bantu. Untuk itu dalam *interface* ini ada pembatasan jumlah kendala yang dapat dioperasikan, yaitu enam.

B. Penyiapan Materi

Setelah perancangan awal, langkah berikutnya adalah penyiapan materi. Materi dipilih mulai dari pertidaksamaan linear sampai masalah sensitivitas, dengan urutan sebagai berikut.



Gambar 4.2.1 Urutan materi yang akan disajikan

Alasan pemilihan materi tersebut dipaparkan dalam uraian berikut. Untuk menyelesaikan masalah Program Linear dua variabel dengan menggunakan metode grafik, diperlukan kemampuan untuk menggambar daerah layak yang tidak lain adalah penyelesaian dari sebuah sistem pertidaksamaan linear, mengingat kendala-kendala dalam masalah program linear berupa pertidaksamaan linear. Karena itu sebelum menyajikan materi program linear disajikan terlebih dahulu sistem pertidaksamaan linear. Sistem pertidaksamaan linear ini didahului oleh materi pertidaksamaan linear yang diharapkan dapat membantu pemahaman konsep daerah penyelesaian dan bukan daerah penyelesaian. Setelah disajikan materi Program Linear, untuk tingkat perguruan tinggi disajikan masalah sensitivitas. Untuk penggunaan *interface* di tingkat sekolah menengah, materi terakhir belum perlu dipelajari.

Interface akan dirancang dengan menggunakan *Maplet* yang merupakan paket dari *Maple8*. Seperti telah dikemukakan pada Bab I (halaman 2), *Maple8* mampu melakukan operasi aljabar tingkat tinggi dan menggambar grafik. *Maple8* memiliki paket yang disebut dengan *Maplet*, yang dapat digunakan untuk membuat suatu *window*, *dialog*, dan berbagai *interface* visual lain dari *Maple*.

Agar *interface* yang dirancang dapat digunakan secara optimal, disusun pula sebuah *handout* yang dijabarkan dalam Bab V skripsi ini. *Handout* ini dirancang seperti poin-poin yang telah dijabarkan dalam Bab II (halaman 15).



C. Menyusun *Interface* dengan *Maplet*

Setelah melakukan penyiapan materi seperti pada Subbab B di atas, maka akan disusun sebuah *interface Maplet* yang mempunyai kemampuan untuk membantu siswa memahami konsep dan prinsip yang ada dalam pokok bahasan pertidaksamaan linear, memahami konsep dan prinsip dalam sistem pertidaksamaan linear, memahami konsep masalah Program Linear dan penyelesaian masalah Program Linear dengan metode grafik, serta membantu mahasiswa untuk memahami masalah sensitivitas dalam masalah Program Linear.

Agar materi yang ingin disajikan tetap utuh dan prinsip individualisasi dalam belajar yang sesuai dengan manfaat PBK dalam Bab II (halaman 8) tetap terpenuhi, maka *interface* ini akan dirancang dalam beberapa *window*. Setiap *window* akan menyajikan satu pokok bahasan. Dengan demikian akan ada empat *window* utama dalam *interface Maplet* ini.

Dirancang pula agar pengguna *interface* dapat memilih pokok bahasan mana yang akan dipelajarinya. Untuk itu perlu ditambahkan sebuah *window* sebagai pembuka yang berisi menu pokok bahasan dan aturan-aturan pemanfaatan *interface* ini. Disusun pula menu bantuan pada *window – window* bagian awal. Menu bantuan ini berisi cara penggunaan *window* yang bersangkutan. Pada bagian akhir, menu bantuan tidak lagi diperlukan, karena diharapkan pengguna sudah memahami *interface* secara menyeluruh.

1. Perencanaan

a. Window Untuk Menyajikan Pokok Bahasan Pertidaksamaan Linear

Dalam penyajian pokok bahasan Pertidaksamaan Linear ini, ada dua tujuan yang ingin dicapai, yaitu siswa dapat menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y < b$, $m x \pm n y \leq b$, $m x \pm n y > b$, atau $m x \pm n y \geq b$ dimana m , n , b anggota bilangan real; serta siswa dapat menentukan keanggotaan suatu titik terhadap sebuah himpunan penyelesaian suatu pertidaksamaan linear. Oleh karena itu, *window* ini dengan bagian-bagian sebagai berikut.

1. Bagian untuk menuliskan pertidaksamaan yang akan digambar

Bagian ini dirancang berupa sebuah kolom yang dapat diisi oleh pengguna. Pada kolom tersebut, pengguna dapat menuliskan pertidaksamaan yang akan digambar. Di sebelah kolom tersebut dirancang sebuah tombol untuk mengeksekusi perintah menggambar fungsi yang telah dimasukkan oleh pengguna.

2. Daerah gambar

Daerah gambar dirancang sebagai tempat grafik hasil eksekusi. Pengguna dapat mengamati hasil gambar dan hasil eksplorasi dari fungsi yang telah ditulis pada kolom yang tersedia. Pada daerah gambar ini juga diberi tempat untuk keterangan keanggotaan titik uji terhadap daerah penyelesaian.

3. Bagian untuk menentukan batas koordinat bidang gambar

Batas koordinat bidang dirancang dapat diatur oleh pengguna. Pengguna dapat memasukkan nilai maksimum dan minimum dari absis dan ordinat.

4. Bagian untuk menuliskan titik yang akan diuji

Sesuai dengan tujuan yang kedua, yaitu agar siswa dapat menentukan keanggotaan suatu titik terhadap daerah penyelesaian, maka dirancanglah bagian ini. Siswa tinggal mengetikkan absis dan ordinat dari titik yang akan diuji keanggotaannya. Disediakan pula sebuah tombol untuk mengeksekusi perintah uji.

5. Bagian untuk dapat menggeser pertidaksamaan

Agar pemahaman siswa mengenai daerah penyelesaian dari suatu pertidaksamaan dan keanggotaan suatu titik terhadap daerah penyelesaian semakin dalam, maka dirancanglah bagian ini. Bagian ini dapat merubah konstanta dari pertidaksamaan yang telah ditulis. Perubahan konstanta tersebut berakibat pada perubahan grafik penyelesaian. Perubahan grafik penyelesaian akan langsung tampak saat terjadi perubahan. Dengan demikian siswa dapat mengamati pengaruh perubahan konstanta terhadap daerah layak dan titik uji.

b. Window Untuk Menyajikan Pokok Bahasan Sistem Pertidaksamaan Linear

Pada *window* ini, ada tiga tujuan yang akan dicapai, yaitu siswa dapat menentukan penyelesaian suatu sistem pertidaksamaan linear, menentukan

batas-batas penyelesaian suatu sistem pertidaksamaan linear, serta menentukan macam-macam bentuk penyelesaian sistem pertidaksamaan linear. Dengan demikian, *window* ini dirancang dengan bagian-bagian sebagai berikut.

1. Bagian untuk menuliskan pertidaksamaan-pertidaksamaan

Pada *window* ini, kolom disediakan sebanyak enam buah. Pada setiap kolom tersebut, pengguna dapat menuliskan pertidaksamaan yang akan digambar.

2. Bagian untuk menggambar

Sama halnya dengan *window* sebelumnya, daerah gambar dirancang sebagai tempat grafik hasil eksekusi. Pengguna dapat mengamati hasil gambar dan hasil eksplorasi dari fungsi yang telah ditulis pada kolom yang tersedia.

3. Bagian untuk menggeser pertidaksamaan

Bagian ini dirancang agar pemahaman siswa mengenai daerah penyelesaian dari suatu sistem pertidaksamaan semakin dalam. Bagian ini dapat merubah konstanta dari pertidaksamaan yang telah ditulis dan diletakkan di sebelah masing-masing kolom. Sehingga apabila dikehendaki untuk menggeser sebuah pertidaksamaan, maka yang diubah adalah bagian yang terletak di sebelah kolom yang bersangkutan. Sama halnya dengan *window* di atas, perubahan konstanta tersebut berakibat pada perubahan grafik penyelesaian. Perubahan grafik penyelesaian akan langsung tampak saat terjadi perubahan. Dengan demikian siswa dapat

mengamati pengaruh perubahan konstanta pertidaksamaan terhadap daerah layak.

6. Bagian untuk menentukan batas koordinat bidang gambar

Batas koordinat bidang dirancang dapat diatur oleh pengguna. Pengguna dapat memasukkan nilai maksimum dan minimum dari absis dan ordinat.

7. Tombol Eksekusi

Tombol ini akan dirancang untuk mengeksekusi perintah menggambar daerah layak setelah semua kendala dituliskan.

c. Window Untuk Menyajikan Pokok Bahasan Program Linear

Dua buah tujuan yang ingin dicapai dalam *window* ini adalah siswa dapat menyelesaikan masalah program linear dengan metode uji titik pojok dan dapat menyelesaikan masalah program linear dengan metode garis selidik. Dengan demikian, *window* ini dirancang dengan bagian-bagian sebagai berikut.

1. Bagian untuk menuliskan kendala dari masalah program linear

Sama seperti *window* sebelumnya, pada *window* ini kolom disediakan sebanyak enam buah. Pada setiap kolom tersebut, pengguna dapat menuliskan pertidaksamaan yang merupakan kendala dari masalah program linear yang akan diselesaikan.

2. Bagian untuk menuliskan fungsi sasaran yang akan dioptimumkan

Bagian ini terdiri dari sebuah kolom untuk menuliskan fungsi sasaran yang akan dioptimumkan

3. Bagian untuk menggambar

Sama halnya dengan *window* sebelumnya, daerah gambar dirancang sebagai tempat grafik hasil eksekusi. Pengguna dapat mengamati hasil gambar dan hasil eksplorasi dari fungsi yang telah ditulis pada kolom yang tersedia.

4. Bagian untuk menentukan koordinat batas daerah gambar

Batas koordinat bidang dirancang dapat diatur oleh pengguna. Pengguna dapat memasukkan nilai maksimum dan minimum dari absis dan ordinat.

5. Bagian untuk menggeser fungsi sasaran

Agar siswa dapat bereksplorasi untuk memahami penyelesaian masalah program linear dengan metode garis selidik, maka dirancang adanya bagian ini.

6. Bagian untuk melakukan uji titik pojok

Bagian ini dirancang agar siswa dapat bereksplorasi untuk memahami penyelesaian masalah program linear dengan metode uji titik pojok.

7. Bagian yang menampilkan hasil optimum

Nilai maksimum maupun nilai minimum dari masalah program linear yang diselesaikan akan ditampilkan pada bagian ini agar pengguna dapat membandingkan hasil eksplorasinya dengan nilai optimum yang sesungguhnya.

8. Tombol eksekusi

Tombol ini akan dirancang untuk mengeksekusi perintah menggambar daerah layak setelah semua kendala dituliskan.

d. Window untuk menyajikan pokok bahasan masalah sensitivitas

Pada pokok bahasan sensitivitas, ada dua tujuan yang ingin dicapai dengan bantuan *window* ini, yaitu menyelesaikan masalah sensitivitas perubahan fungsi sasaran, dan menyelesaikan masalah sensitivitas perubahan kendala. *Window* ini dirancang berkaitan dengan *window* untuk masalah Program Linear. Kendala-kendala dan fungsi sasaran dimasukkan pada *window* sebelumnya, sehingga pada *window* ini tidak disediakan tempat untuk melakukan pemasukan tersebut, namun hanya tampilan dari *window* sebelumnya. Bagian-bagian *window* ini adalah:

1. Bagian untuk menggambar

Daerah gambar pada *window* ini dirancang untuk menampilkan kembali grafik dari masalah Program Linear yang sudah dieksekusi pada *window* sebelumnya.

2. Bagian untuk menampilkan fungsi sasaran dan kendala sesuai dengan masukan pada *window* sebelumnya

3. Bagian menampilkan nilai optimum masalah awal sesuai dengan masukan pada *window* sebelumnya

4. Bagian untuk menggeser kendala

Bagian ini digunakan untuk eksplorasi masalah sensitivitas perubahan kendala

5. Bagian untuk mengubah gradien fungsi sasaran

Bagian ini digunakan untuk eksplorasi masalah sensitivitas perubahan fungsi sasaran

6. Bagian yang menampilkan nilai optimum baru hasil dari pergeseran

2. Prosedur-prosedur yang Dibuat

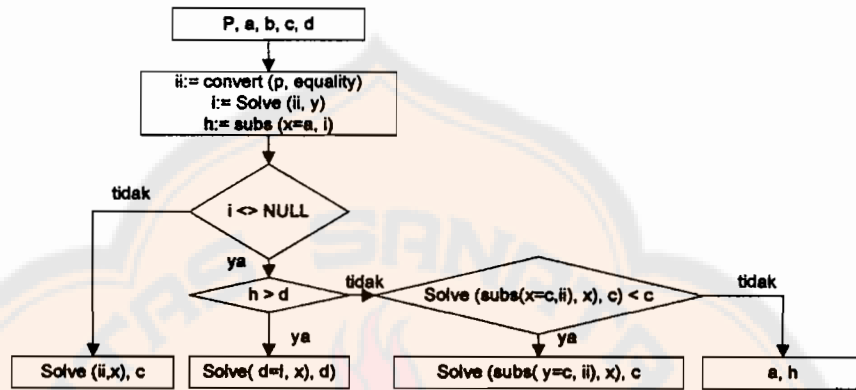
Interface ini akan disusun dengan menggunakan *Maplet*. Untuk membuat sebuah *Maplet* diperlukan bahasa pemrograman yang disusun dalam prosedur-prosedur untuk menjalankan suatu perintah. Dalam perancangan *interface* ini digunakan 23 prosedur. Berikut ini adalah paparan dari masing-masing prosedur tersebut.

a. Prosedur “Koordplotteks”

Prosedur ini berfungsi untuk menentukan koordinat label untuk suatu kendala pada daerah plot. Label kendala akan ditampilkan di tepi daerah plot dan dekat dengan garis kendala. Hal ini dimaksudkan agar pengguna tidak mengalami kesulitan dalam mengidentifikasi suatu kendala dan proses pengamatan tidak terganggu karena label berada di tepi daerah plot. Keluaran dari prosedur ini berupa koordinat dalam (x,y) .

Dalam prosedur ini dibutuhkan input berupa sebuah pertidaksamaan yang merupakan suatu kendala (dimisalkan p), nilai terkecil untuk sumbu x pada daerah plot (dimisalkan a), nilai terbesar untuk sumbu x pada daerah plot (dimisalkan b), nilai terkecil untuk sumbu y pada daerah plot (dimisalkan c), dan nilai terbesar untuk sumbu y pada daerah plot (dimisalkan d). Prinsip

kerja dari prosedur ini adalah mencari koordinat label dengan posisi berada di bawah, di atas, atau disamping kanan dalam daerah plot. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.1 berikut.



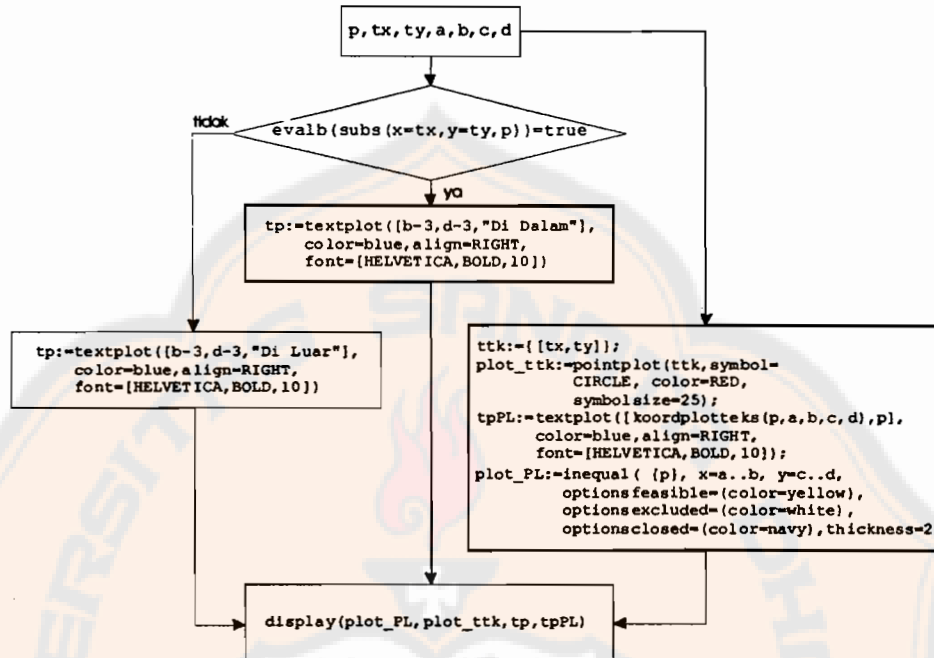
Gambar 4.3.1 flowchart untuk prosedur Koordplotteks

b. Prosedur “Pertidaksamaan”

Prosedur ini berfungsi untuk menampilkan gambar himpunan penyelesaian dari suatu pertidaksamaan dan menguji keanggotaan sebuah titik terhadap himpunan penyelesaian tersebut.

Dalam prosedur ini dibutuhkan input berupa sebuah pertidaksamaan yang merupakan suatu kendala (dimisalkan p), absis titik uji (dimisalkan tx), ordinat titik uji (dimisalkan ty), nilai terkecil untuk sumbu x pada daerah plot (dimisalkan a), nilai terbesar untuk sumbu x pada daerah plot (dimisalkan b), nilai terkecil untuk sumbu y pada daerah plot (dimisalkan c), dan nilai terbesar untuk sumbu y pada daerah plot (dimisalkan d). Prinsip kerja dari prosedur ini terdiri dari 2 tahap. Tahap yang pertama adalah melakukan pengujian apakah titik uji termasuk dalam daerah himpunan penyelesaian dari pertidaksamaan, kemudian menampilkan hasil dari pengujian tersebut dalam daerah plot. Tahap yang kedua adalah menggambar titik uji, menampilkan label dari

kendala, dan menggambar daerah himpunan penyelesaian dari pertidaksamaan p . Flowchart dari prosedur ini tampak pada Gambar 4.3.2 berikut.



Gambar 4.3.2 flowchart untuk prosedur pertidaksamaan

c. Prosedur “geser_kendala”

Prosedur ini berfungsi untuk mengubah nilai konstanta suatu pertidaksamaan berdasarkan nilai dari suatu *slider*. Hal ini dimaksudkan agar jika nilai *slider* diubah dengan cara menggeser, maka nilai konstanta suatu pertidaksamaan akan berubah.

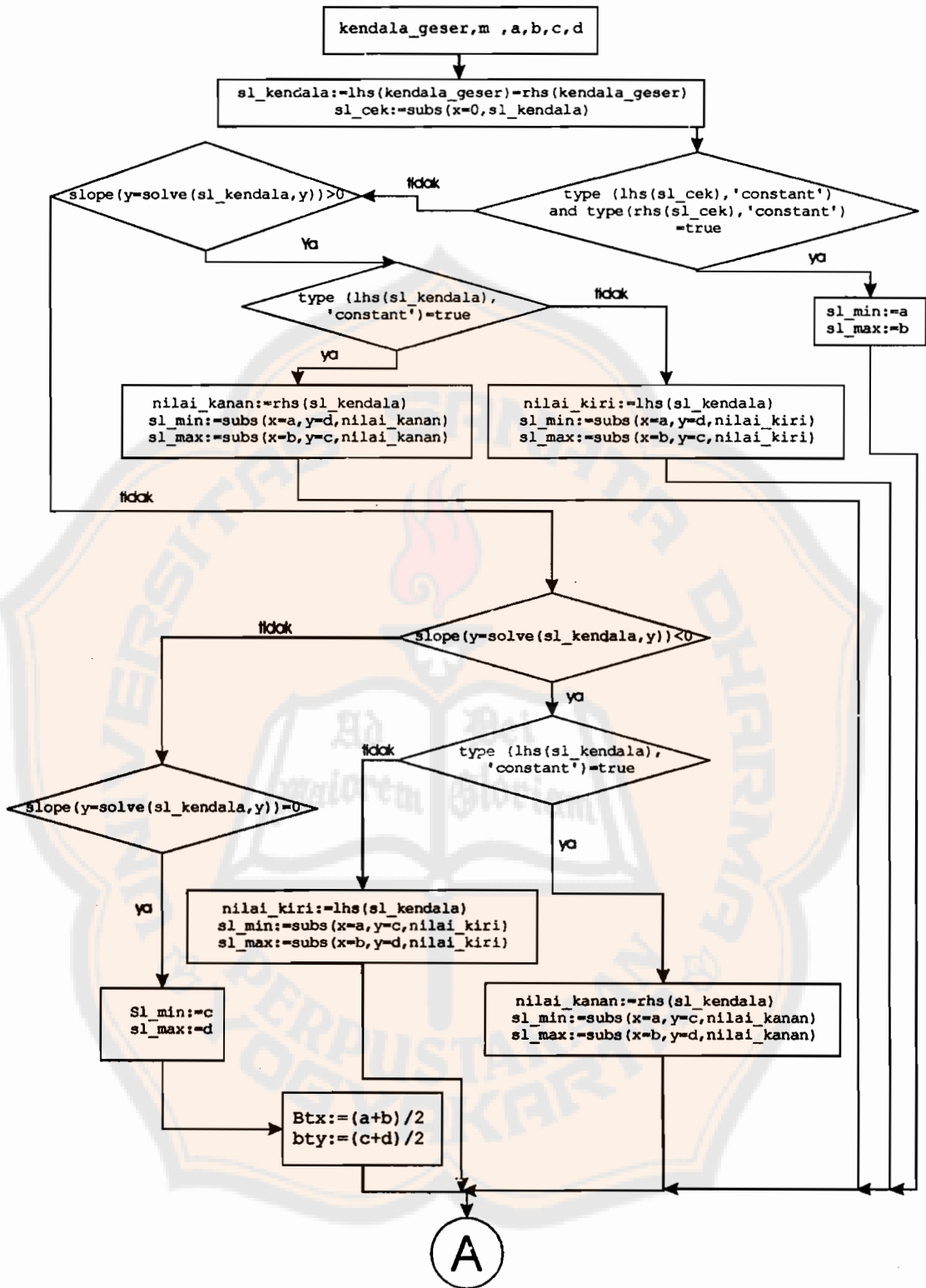
Dalam prosedur ini dibutuhkan input berupa sebuah pertidaksamaan yang merupakan suatu kendala (dimisalkan *kendala_geser*), nilai dari suatu slider (dimisalkan *m*), nilai terkecil untuk sumbu x pada daerah plot (dimisalkan *a*), nilai terbesar untuk sumbu x pada daerah plot (dimisalkan *b*), nilai terkecil untuk sumbu y pada daerah plot (dimisalkan *c*), dan nilai terbesar

untuk sumbu y pada daerah plot (dimisalkan d). Prinsip kerja dari prosedur ini adalah sebagai berikut. Langkah yang pertama adalah menentukan nilai minimum dan maksimum dari nilai konstanta pertidaksamaan berdasarkan nilai dari slider, hal ini dimaksudkan agar pertidaksamaan dengan nilai konstanta baru selalu tampak di bidang gambar. Langkah selanjutnya adalah menampilkan pertidaksamaan dengan nilai konstanta baru. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.3a dan Gambar 4.3.3b.

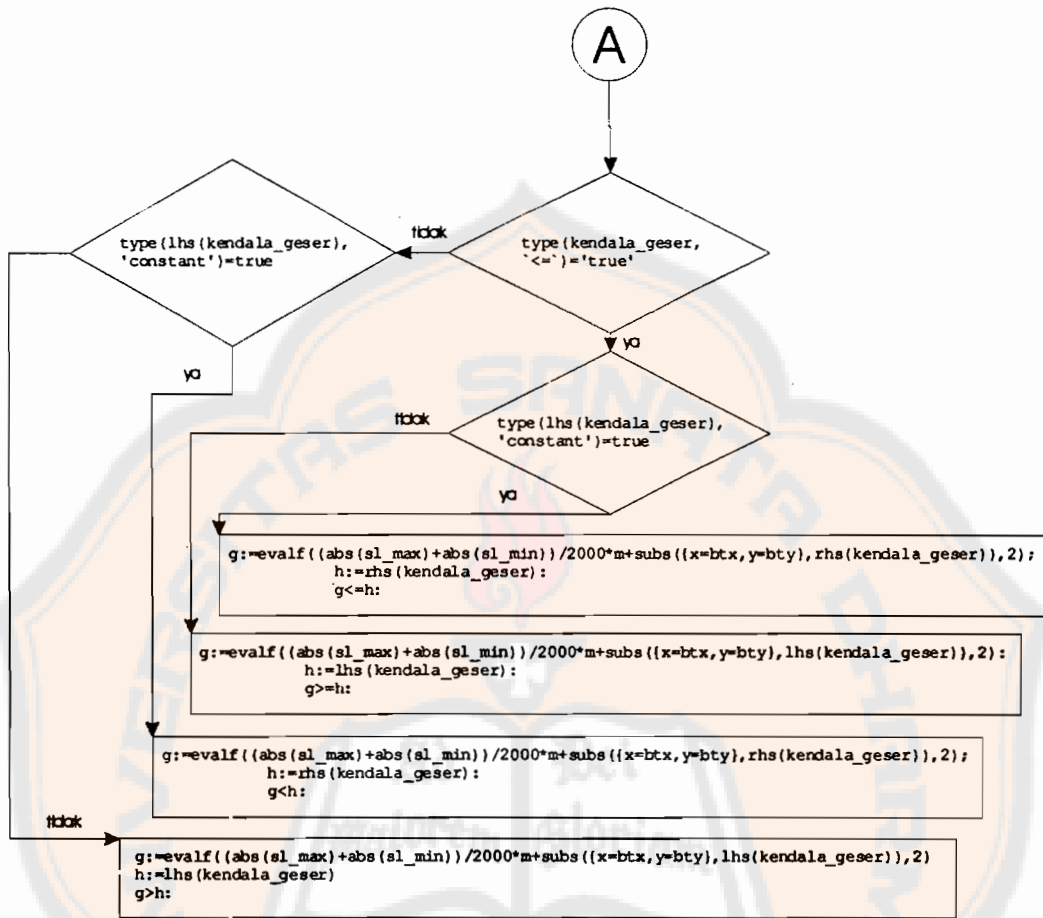
d. Prosedur “geser_PLUS”

Prosedur ini berfungsi untuk mengubah nilai konstanta suatu pertidaksamaan, perubahan sesuai dengan nilai yang diinginkan. Fungsi dari prosedur sama dengan prosedur `geser_kendala`, namun cara menggunakan berbeda. Prosedur `geser_PLUS` digunakan untuk action sebuah tombol, sedangkan prosedur `geser_kendala` digunakan untuk action pada slider.

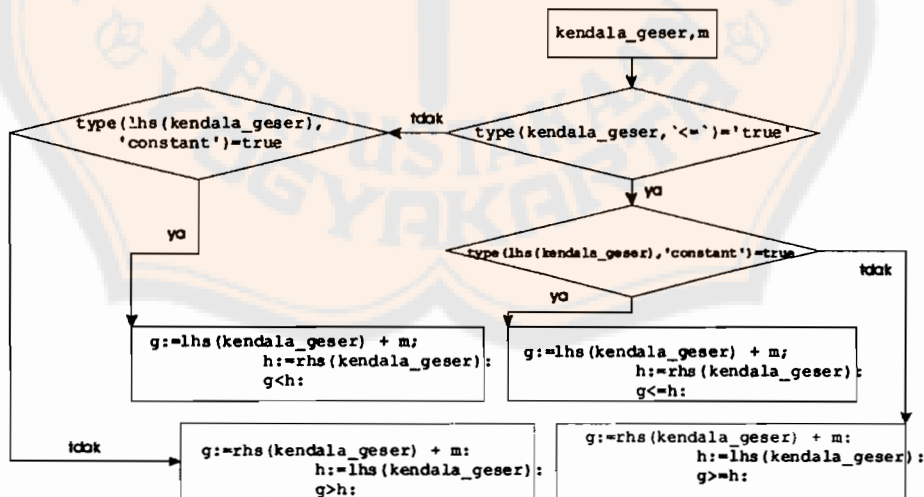
Dalam prosedur ini dibutuhkan input berupa sebuah pertidaksamaan yang merupakan suatu kendala (dimisalkan *kendala_geser*), nilai pergeseran (dimisalkan m). Prinsip kerja dari prosedur ini adalah sebagai berikut. Langkah yang pertama adalah mengidentifikasi jenis relasi dari pertidaksamaan, setelah itu mengubah konstanta pertidaksamaan, kemudian menampilkan pertidaksamaan dengan nilai konstanta baru. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.4.



Gambar 4.3.3a flowchart untuk prosedur geser_kendala bagian a



Gambar 4.3.3b flowchart untuk prosedur gese_kendala bagian b



Gambar 4.3.4 flowchart untuk prosedur gese_PLUS

e. Prosedur “Gambar_PTL”

Prosedur ini berfungsi untuk menampilkan gambar dari daerah penyelesaian suatu sistem pertidaksamaan linear.

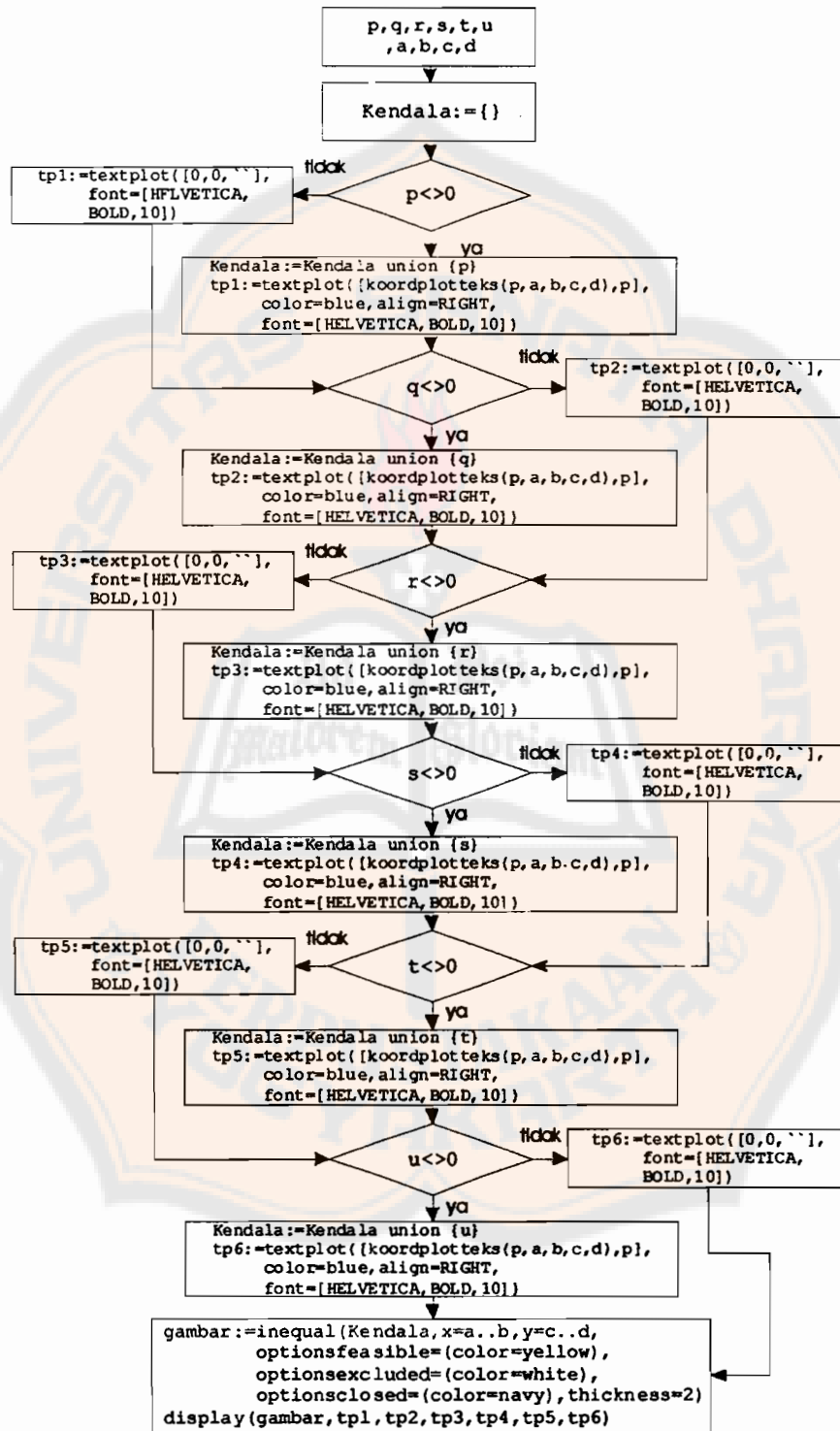
Dalam prosedur ini dibutuhkan input berupa enam buah pertidaksamaan (dimisalkan p, q, r, s, t, u), nilai terkecil untuk sumbu x pada daerah plot (dimisalkan a), nilai terbesar untuk sumbu x pada daerah plot (dimisalkan b), nilai terkecil untuk sumbu y pada daerah plot (dimisalkan c), dan nilai terbesar untuk sumbu y pada daerah plot (dimisalkan d). Prinsip kerja dari prosedur ini terdiri atas 2 tahap. Tahap yang pertama adalah inisialisasi input sistem pertidaksamaan linear dan menampilkan label dari masing-masing pertidaksamaan. Tahap yang kedua adalah Menggambar daerah himpunan penyelesaian dari sistem pertidaksamaan linear. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.5.

f. Prosedur “TeksSlider”

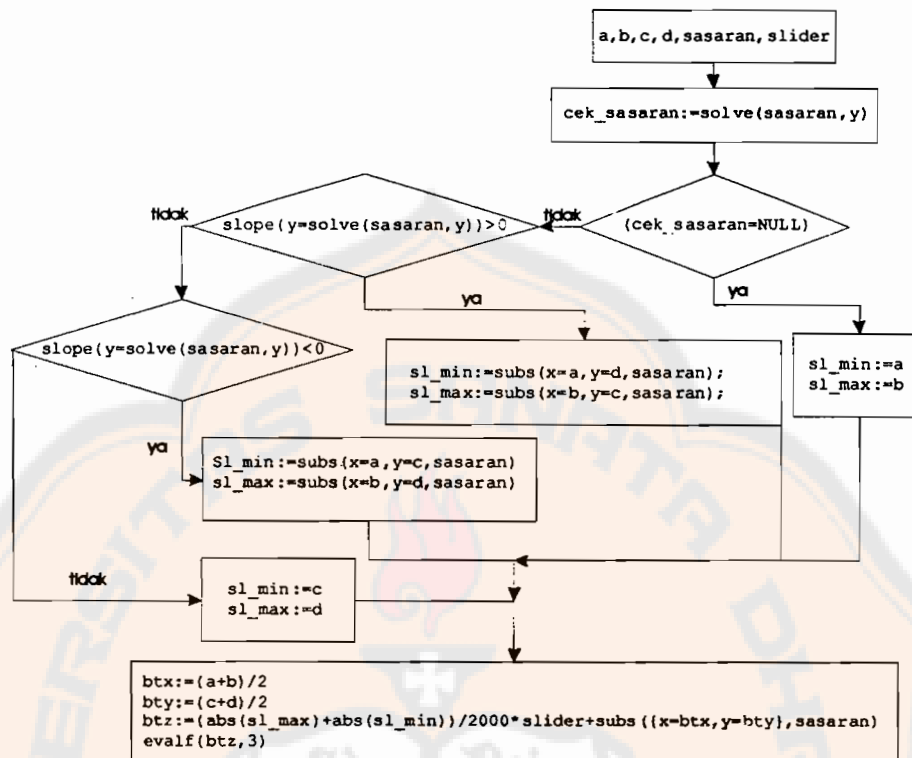
Prosedur ini berfungsi untuk mendapatkan nilai fungsi sasaran yang didapat dari nilai suatu slider.

Dalam prosedur ini dibutuhkan input berupa nilai terkecil untuk sumbu x pada daerah plot (dimisalkan a), nilai terbesar untuk sumbu x pada daerah plot (dimisalkan b), nilai terkecil untuk sumbu y pada daerah plot (dimisalkan c), nilai terbesar untuk sumbu y pada daerah plot (dimisalkan d), fungsi sasaran (dimisalkan f), dan nilai suatu slider (dimisalkan $slider$). Prinsip

nilai konstanta pertidaksamaan berdasarkan nilai dari slider. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.6.



Gambar 4.3.5 flowchart untuk prosedur Gambar_PTL

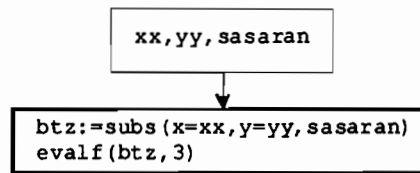


Gambar 4.3.6 flowchart untuk prosedur TeksSlider

g. Prosedur “TeksSasaran”

Prosedur ini berfungsi untuk menghitung nilai fungsi sasaran pada titik tertentu yang diberikan *user*.

Dalam prosedur ini dibutuhkan input berupa absis suatu titik (dimisalkan *xx*), ordinat suatu titik (dimisalkan *yy*), dan fungsi sasaran (dimisalkan *sasaran*). Prinsip kerja dari prosedur ini adalah menghitung nilai fungsi sasaran dititik yang ditentukan. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.7.



Gambar 4.3.7 flowchart untuk prosedur **TeksSasaran**

h. Prosedur “Uji_ttk”

Prosedur ini berfungsi untuk menampilkan gambar dari suatu masalah program linear dimana nilai dari fungsi sasaran ditentukan oleh suatu titik uji.

Dalam prosedur ini dibutuhkan input berupa absis titik uji (dimisalkan Utx), ordinat titik uji (dimisalkan Uty), enam buah pertidaksamaan (dimisalkan p, q, r, s, t, u), nilai terkecil untuk sumbu x pada daerah plot (dimisalkan a), nilai terbesar untuk sumbu x pada daerah plot (dimisalkan b), nilai terkecil untuk sumbu y pada daerah plot (dimisalkan c), dan nilai terbesar untuk sumbu y pada daerah plot (dimisalkan d). Prinsip kerja dari prosedur ini terdiri atas 3 tahap. Tahap yang pertama adalah inisialisi input kendala dan menampilkan label dari masing-masing kendala. Tahap yang kedua adalah mencari titik potong antar kendala dan titik ujung dari himpunan penyelesaian. Tahap yang ketiga adalah menghitung nilai fungsi sasaran dan menggambar daerah himpunan penyelesaian dari sistem pertidaksamaan linear. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.8a, 4.3.8b, dan 4.3.8c.

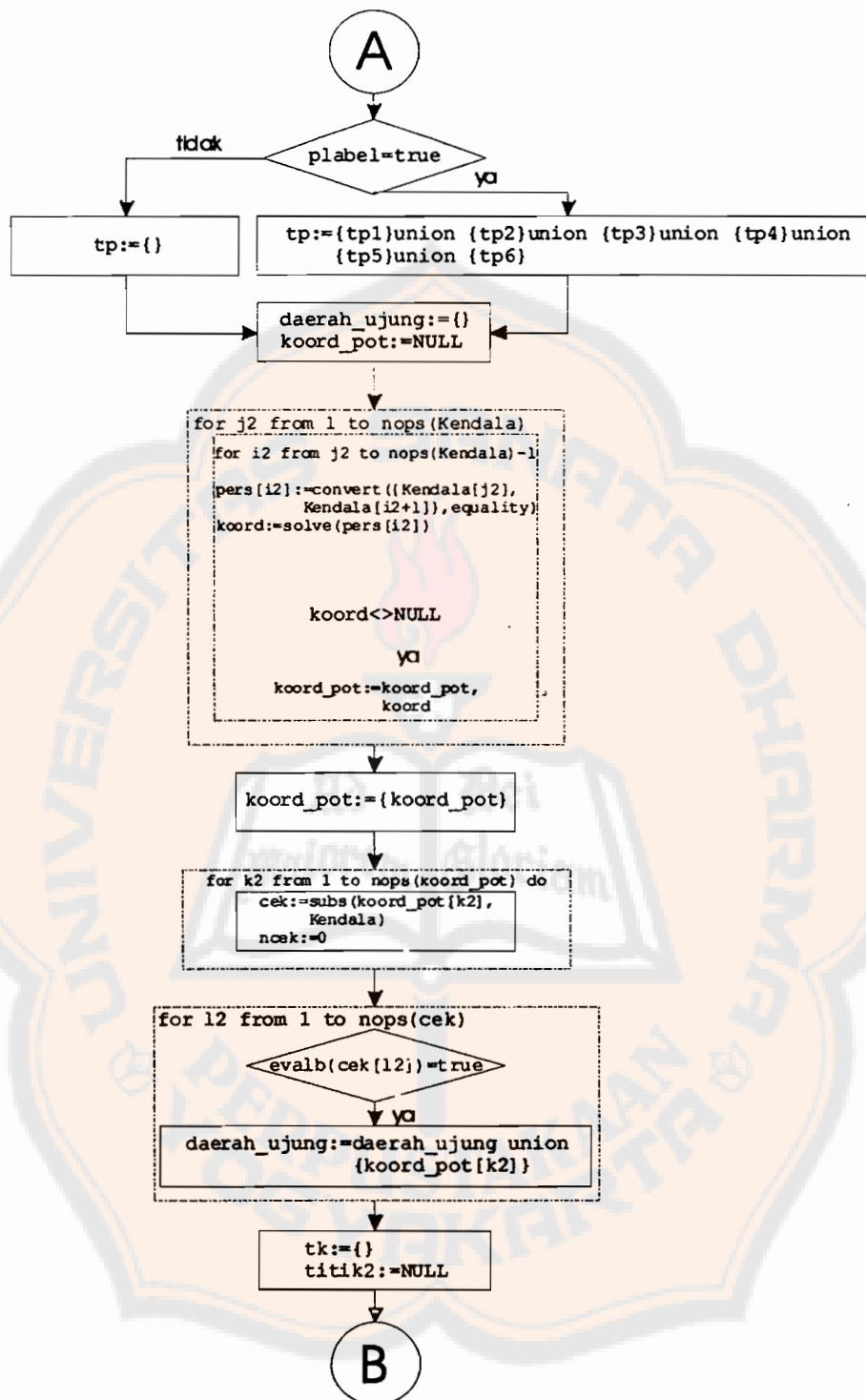
i. Prosedur “ntgradien”

Prosedur ini berfungsi untuk mencari nilai fungsi sasaran baru pada saat gradiennya diubah untuk melakukan analisa sensitivitas fungsi sasaran.

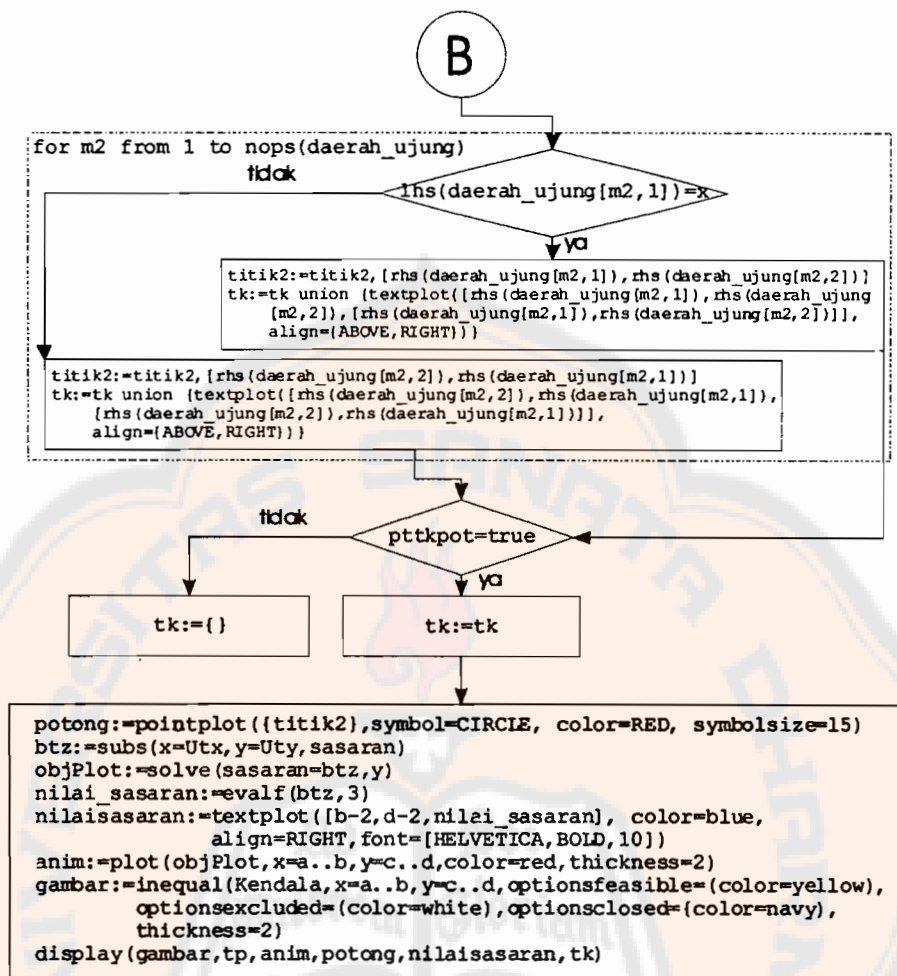
Dalam prosedur ini dibutuhkan input berupa keterangan dimana analisa sensitivitas akan dilakukan, yaitu titik minimum atau maksimum (dimisalkan min_max), titik minimum (dimisalkan Pot_min), titik maksimum (dimisalkan pot_max), fungsi sasaran (dimisalkan Sb). Prinsip kerja dari prosedur adalah menguji di titik mana analisa sensitivitas dilakukan kemudian mencari nilai fungsi sasaran di titik tersebut. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.9.



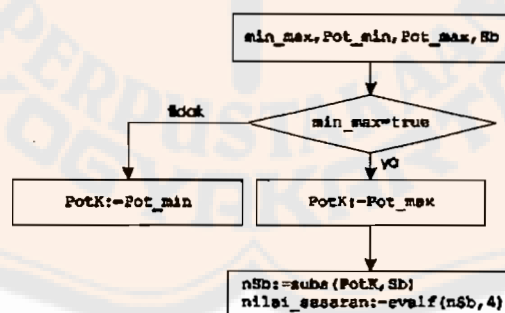
Gambar 4.3.8a flowchart untuk prosedur Uji_ttk (tahap 1)



Gambar 4.3.8b flowchart untuk prosedur Uji_ttk (tahap 2)



Gambar 4.3.8c flowchart untuk prosedur Uji_ttk (lanjutan tahap 2 dan tahap 3)

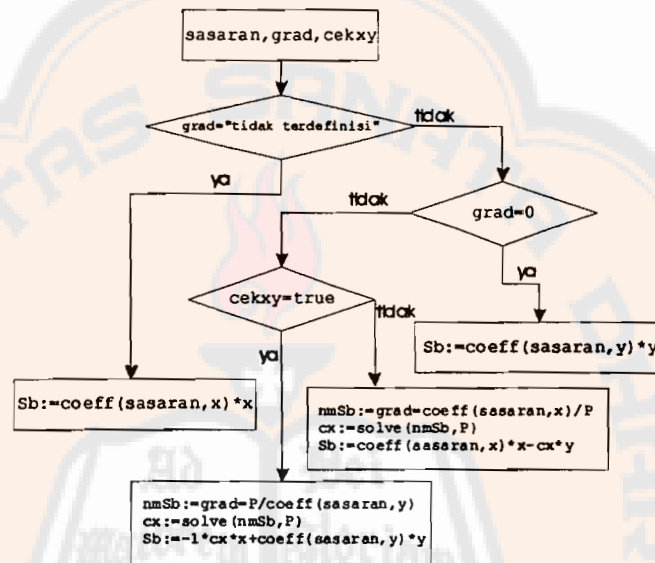


Gambar 4.3.9 flowchart untuk prosedur ntgradien

j. Prosedur “tgradien”

Prosedur ini berfungsi untuk menyusun fungsi sasaran baru pada saat gradien diubah.

Dalam prosedur ini dibutuhkan input berupa fungsi sasaran baru (dimisalkan *sasaran*), gradien fungsi baru (dimisalkan *grad*), dan keterangan variabel yang akan diubah (dimisalkan *cekxy*). Prinsip kerja dari prosedur adalah menyusun fungsi sasaran baru berdasarkan gradien yang diberikan. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.10.



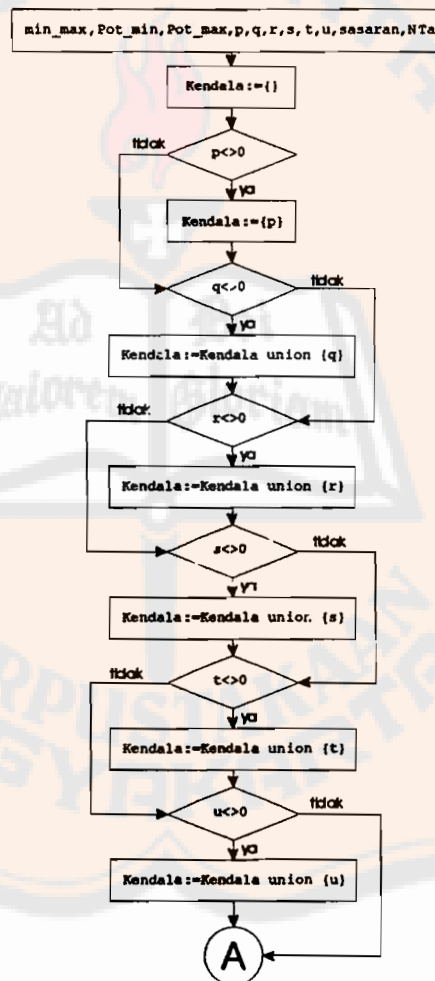
Gambar 4.3.10 flowchart untuk prosedur tgradien

k. Prosedur “mntgradien”

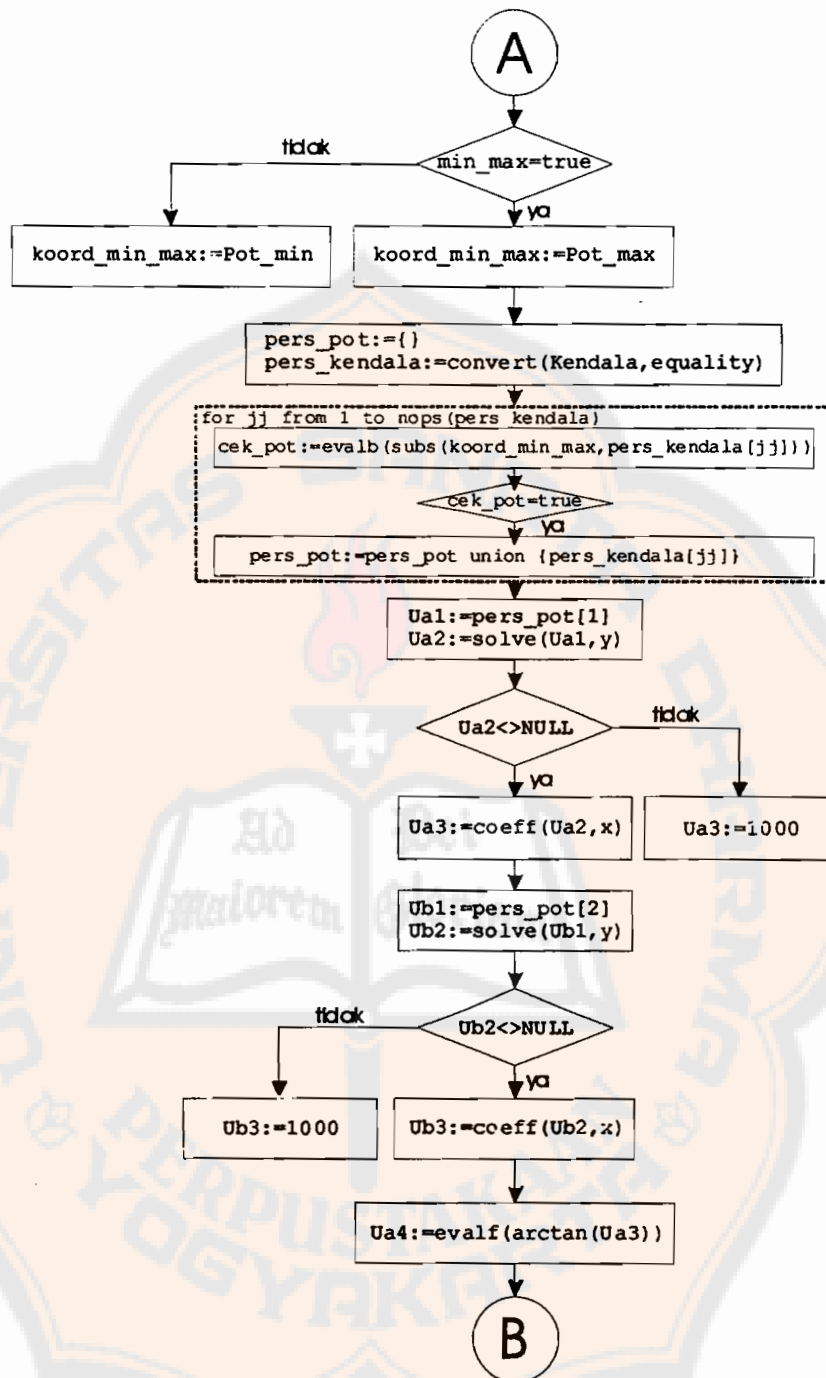
Prosedur ini berfungsi untuk mencari nilai gradien untuk fungsi baru pada saat slider digeser.

Dalam prosedur ini dibutuhkan input berupa keterangan dimana analisa sensitivitas akan dilakukan, yaitu titik minimum atau maksimum (dimisalkan *min_max*), titik minimum (dimisalkan *Pot_min*), titik maksimum (dimisalkan *pot_max*), pertidaksamaan yang menjadi kendala (dimisalkan *p, q, r, s, t, u*), nilai slider (dimisalkan *NTa*), dan fungsi sasaran (dimisalkan *sasaran*). Prinsip kerja dari prosedur terdiri dari 4 tahap. Tahap yang pertama

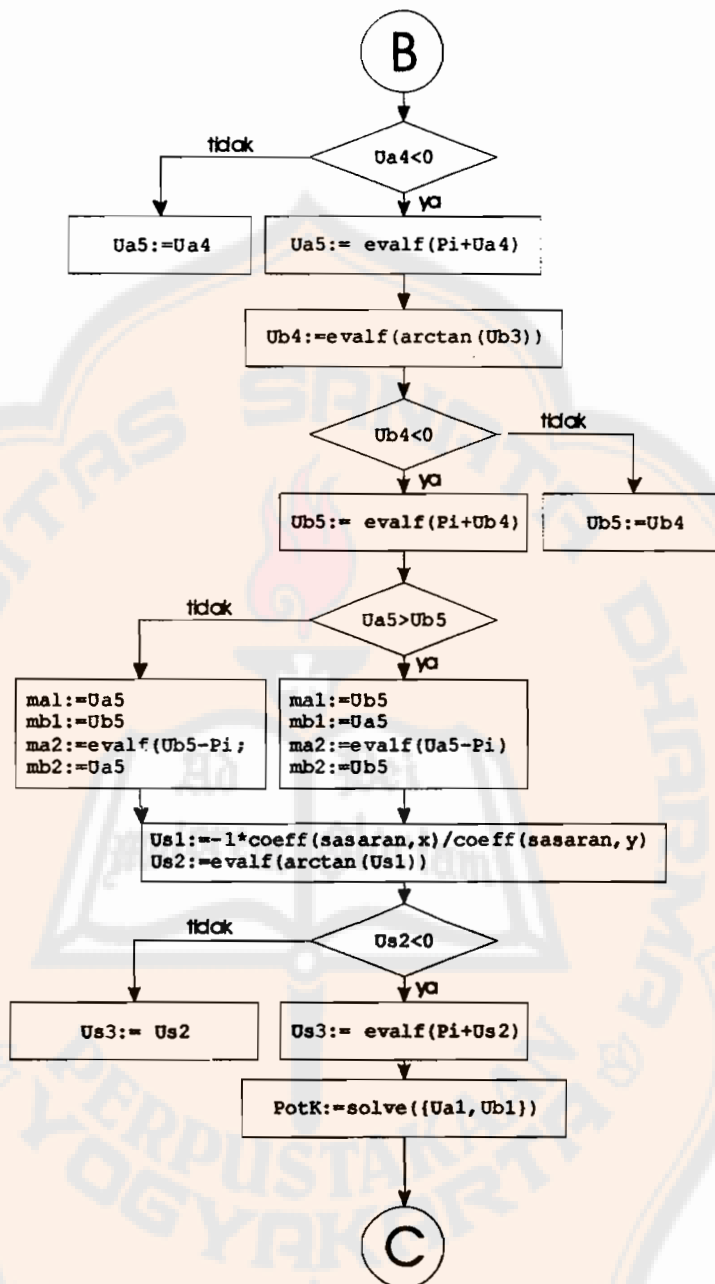
adalah inisialisi input kendala dan menampilkan label dari masing-masing kendala. Tahap yang kedua adalah mencari titik potong antar kendala dan titik ujung dari himpunan penyelesaian. Tahap yang ketiga adalah mencari batas minimum dan maksimum perubahan gradien yang bisa dilakukan. Tahap yang terakhir adalah menyesuaikan nilai gradien dengan nilai pada slider. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.11a, 4.3.11b, 1.3.11c dan 4.3.11d.



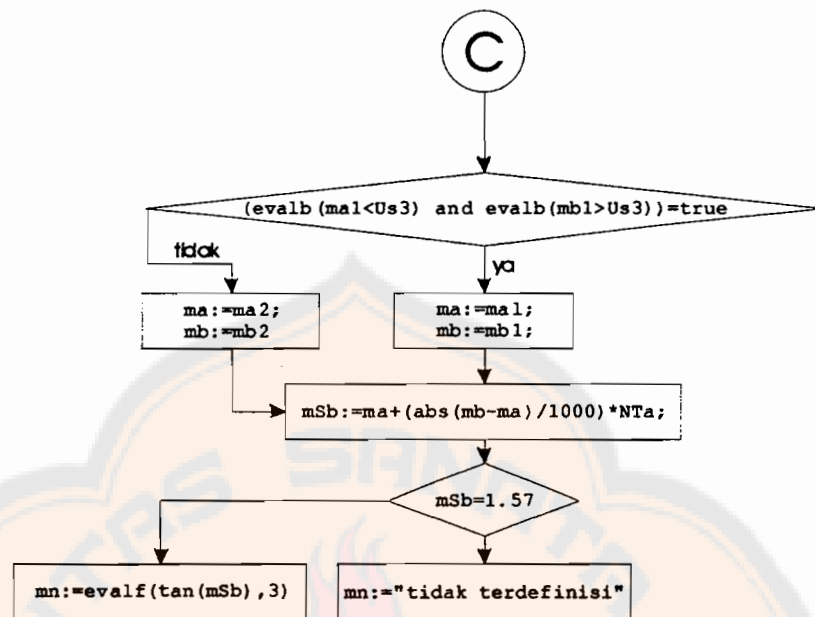
Gambar 4.3.11a flowchart untuk prosedur *mtgradien* (tahap 1)



Gambar 4.3.11b flowchart untuk prosedur mntgradien (tahap 2)



Gambar 4.3.11c flowchart untuk prosedur mntgradien (tahap 3)



Gambar 4.3.11d flowchart untuk prosedur mntgradien (tahap 4)

l. Prosedur “gradien”

Prosedur ini digunakan untuk menampilkan gambar dari suatu masalah program linear jika gradien diubah.

Prosedur ini mirip dengan prosedur Uji_ttk, perbedaan terletak pada tahap terakhir (tahap ke-3) yaitu tahap menghitung nilai fungsi sasaran dan menggambar daerah himpunan penyelesaian dari sistem pertidaksamaan linear.

Flowchart dari prosedur ini tampak pada Gambar 4.3.12.

```

potong:=pointplot({titik2},symbol=CIRCLE,color=RED,symbolsize=15)
nilai_sasaran:=evalf(nSb,3)
nilaisasaran:=textplot({b-3,3-2,{nilai_sasaran}},color=blue,
align=RIGHT,font=[HELVETICA,BOLD,10])
objPlot:=solve(Sb=nSb,y)
anim:=plot(objPlot,x=a..b,y=c..d,color=red,thickness=2)
gambar:=inequal(Kendala,x=a..b,y=c..d,optionsfeasible=(color=yellow),
optionexcluded=(color=white),optionsclosed=(color=navy),
thickness=2)
display(gambar,tp,anim,potong,nilaisasaran,tk)
    
```

Gambar 4.3.12 tahap ke-3 dari prosedur gradient



m. Prosedur “Gambar_Grafik”

Prosedur ini digunakan untuk menampilkan gambar dari suatu masalah program linear dimana nilai dari fungsi sasaran ditentukan oleh suatu slider.

Prinsip kerja dari prosedur ini terdiri atas 4 tahap. Tahap pertama dan kedua mirip dengan tahap pertama dan kedua pada prinsip kerja pada prosedur Uji_ttk. Tahap yang ketiga mirip dengan prinsip kerja pada prosedur TeksSlider. Sedangkan tahap keempat adalah menghitung nilai fungsi sasaran dan menggambar daerah himpunan penyelesaian dari sistem pertidaksamaan linear.

Flowchart dari prosedur ini tampak pada Gambar 4.3.13 berikut.

```

btx:=(a+b)/2
bty:=(c+d)/2
btz:=(abs(sl_max)+abs(sl_min))/2000*slider+subs({x=btx,y=bty},sasaran)
objPlot:=solve(sasaran=btz,y)
nilai_sasaran:=evalf(btz,3)
nilaisasaran:=textplot([b-2,d-2,nilai_sasaran],color=blue,align=RIGHT,
font=[HELVETICA,BOLD,10])
anim:=plot(objPlot,x=a..b,y=c..d,color=red,thickness=2)
gambar:=inequal(Kendala,x=a..b,y=c..d,optionsfeasible=(color=yellow),
optionsexcluded=(color=white),optionsclosed=(color=navy),
thickness=2)
display(gambar,tp,anim,potong,nilaisasaran,tk)
    
```

Gambar 4.3.13 tahap keempat dari prosedur Gambar_Grafik

n. Prosedur “plotdititik”

Prosedur ini digunakan untuk menampilkan gambar dari suatu masalah program linear dimana nilai dari fungsi sasaran ditentukan oleh pengguna.

Prinsip kerja dari prosedur ini terdiri atas 4 tahap. Tahap pertama dan kedua mirip dengan tahap pertama dan kedua pada prinsip kerja pada prosedur Uji_ttk. Tahap yang ketiga adalah menghitung nilai fungsi sasaran dan menggambar daerah himpunan penyelesaian dari sistem pertidaksamaan linear.

Flowchart dari prosedur ini tampak pada Gambar 4.3.14.

```

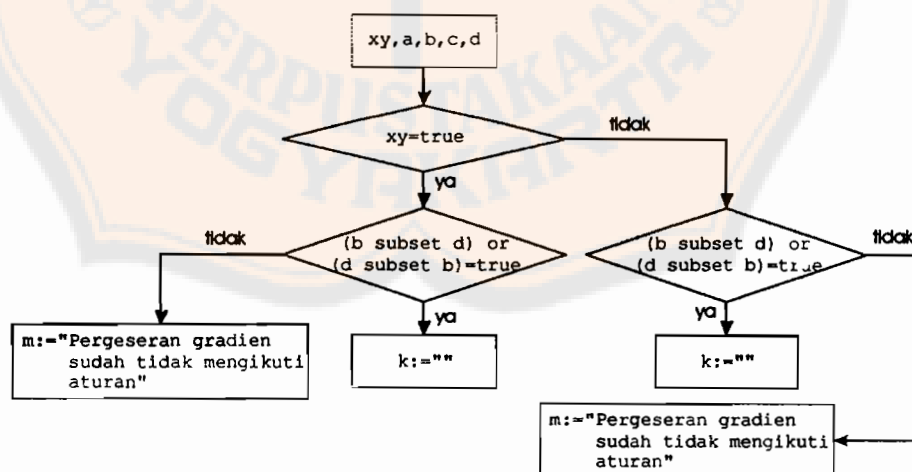
potong:=pointplot({titik2},symbol=CIRCLE, color=RED, symbolsize=15)
objPlot:=solve(sasaran=teknilai,y)
nilaisasaran:=textplot({b-2,d-2,teknilai}, color=blue,
    align=RIGHT,font={HELVETICA,BOLD,10})
anim:=plot(objPlot,x=a..b,y=c..d,color=red,thickness=2)
gambar:=inequal(Kendala,x=a..b,y=c..d,optionsfeasible=(color=yellow),
    optionsexcluded=(color=white),optionsclosed=(color=navy),
    thickness=2)
display(gambar,tp,anim,potong,nilaisasaran,tk)
    
```

Gambar 4.3.14 tahap ketiga dari prosedur **plotteditik**

o. Prosedur “peringatan”

Prosedur ini digunakan untuk menampilkan keterangan jika pergeseran gradien fungsi sasaran sudah melanggar ketentuan.

Dalam prosedur ini dibutuhkan input berupa keterangan konstanta variabel yang akan diubah (dimisalkan xy), nilai terkecil untuk sumbu x pada daerah plot (dimisalkan a), nilai terbesar untuk sumbu x pada daerah plot (dimisalkan b), nilai terkecil untuk sumbu y pada daerah plot (dimisalkan c), dan nilai terbesar untuk sumbu y pada daerah plot (dimisalkan d). *Flowchart* dari prosedur ini tampak pada Gambar 4.3.15.



Gambar 4.3.15 flowchart prosedur **peringatan**

p. Prosedur “Maksimum”

Prosedur ini digunakan untuk menghitung nilai maksimum dari suatu masalah program linear.

Dalam prosedur ini dibutuhkan input berupa pertidaksamaan yang menjadi kendala (dimisalkan p, q, r, s, t, u dan fungsi sasaran (dimisalkan $sasaran$). Prinsip kerja prosedur ini terdiri dari 2 tahap. Tahap yang pertama sama dengan tahap inialisasi input kendala pada prosedur **mntgradien**. Sedangkan tahap yang kedua adalah perhitungan nilai maksimum. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.16.

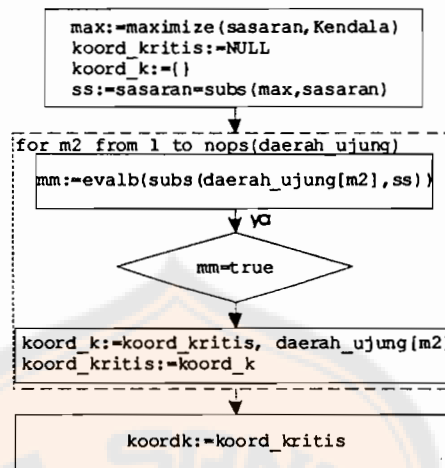
```
max:=maximize(sasaran,Kendala)
evalf:=(subs(max,sasaran),4)
```

Gambar 4.3.16 Tahap kedua dari prosedur **Maksimum**

q. Prosedur “MaksimumKoord”

Prosedur ini digunakan untuk mencari koordinat dari nilai maksimum dari suatu masalah program linear.

Dalam prosedur ini dibutuhkan input berupa pertidaksamaan yang menjadi kendala (dimisalkan p, q, r, s, t, u dan fungsi sasaran (dimisalkan $sasaran$). Prinsip kerja prosedur ini terdiri dari 3 tahap. Tahap yang pertama dan kedua sama dengan tahap pertama dan kedua pada prosedur **mntgradien**. Sedangkan tahap yang ketiga adalah perhitungan nilai maksimum dan mencari koordinat titik maksimum. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.17.

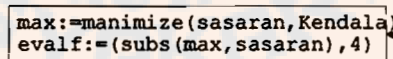


Gambar 4.3.17 Tahap ketiga dari prosedur MaksimumKoord

r. Prosedur “Minimum”

Prosedur ini digunakan untuk menghitung nilai minimum dari suatu masalah program linear.

Dalam prosedur ini dibutuhkan input berupa pertidaksamaan yang menjadi kendala (dimisalkan p, q, r, s, t, u dan fungsi sasaran (dimisalkan $sasaran$). Prinsip kerja prosedur ini terdiri dari 2 tahap. Tahap yang pertama sama dengan tahap inisialisasi input kendala pada prosedur `mntgradien`. Sedangkan tahap yang kedua adalah perhitungan nilai minimum. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.18.

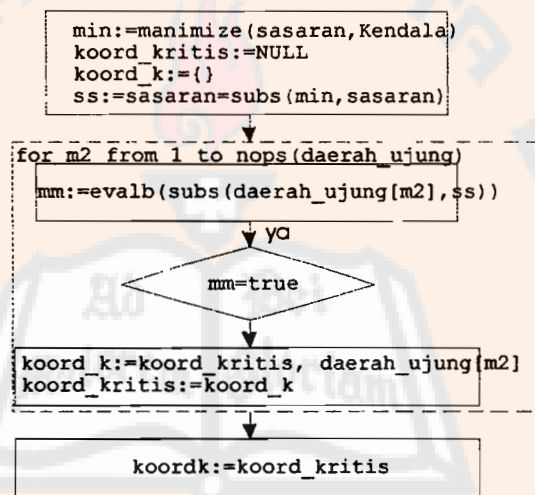


Gambar 4.3.18 tahap ketiga dari prosedur Minimum

s. Prosedur “MinimumKoord”

Prosedur ini digunakan untuk mencari koordinat dari nilai minimum dari suatu masalah program linear.

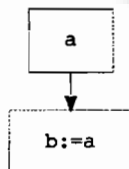
Dalam prosedur ini dibutuhkan input berupa pertidaksamaan yang menjadi kendala (dimisalkan p, q, r, s, t, u dan fungsi sasaran (dimisalkan $sasaran$). Prinsip kerja prosedur ini terdiri dari 3 tahap. Tahap yang pertama dan kedua sama dengan tahap pertama dan kedua pada prosedur **mntgradien**. Sedangkan tahap yang ketiga adalah perhitungan nilai minimum dan mencari koordinat titik minimum. *Flowchart* dari prosedur ini tampak pada Gambar 4.3.19 berikut.



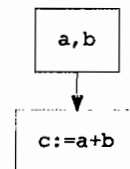
Gambar 4.3.19 tahap ketiga dari prosedur **MinimumKoord**

t. Prosedur Lainnya

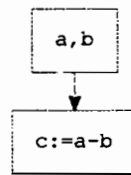
Prosedur-prosedur sederhana yang digunakan meliputi prosedur “pindah”, prosedur “tambah”, prosedur “kurang”, dan prosedur “gradien”.



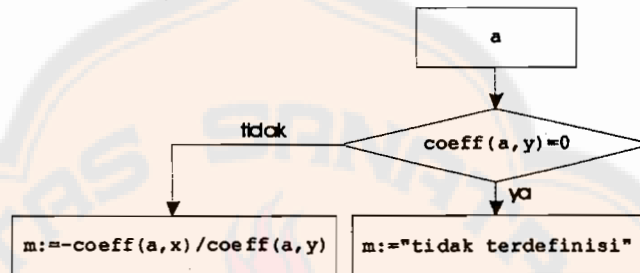
Gambar 4.3.20 flowchart prosedur pindah



Gambar 4.3.21 flowchart prosedur tambah



Gambar 4.3.22 flowchart prosedur kurang



Gambar 4.3.23 flowchart prosedur gradien

3. Penyusunan Maplet

Interface Maplet yang disusun dalam skripsi ini terdiri dari sebelas *window* yang akan disajikan di bawah ini. Adapun *source code* dari masing-masing *window* dapat dilihat pada lampiran.

a. Window W1: Menu_Pembuka

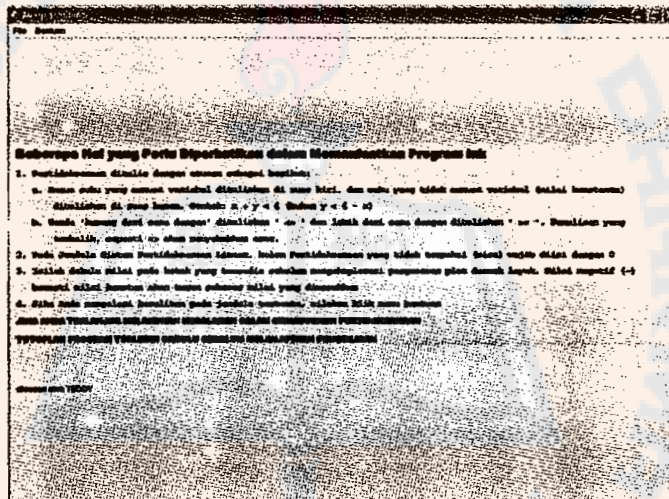
Window W1 adalah *window* yang pertama kali muncul saat Maplet dieksekusi. *Window* ini terdiri dari beberapa bagian, yaitu: Elemen *Window*, Elemen Body, Elemen Menu dan Elemen Layout.

Elemen *Window* memuat title bar berisi judul *window*, yaitu: Menu_Pembuka dan ukuran *window* yaitu 1024 x 768 pixel.

Elemen Body *Window* W1 memuat label yang memaparkan aturan-aturan dalam pemakaian *interface* ini.

Elemen Menu memuat Menu Bar yang berisi dua pilihan menu utama, yaitu: File dan Bantuan. Menu File berisi menu Buka dan menu Close. Menu Close digunakan untuk menutup *window*. Sedangkan menu Buka digunakan untuk membuka *window* yang lain, yaitu *window* Pertidaksamaan Linear, *window* Sistem Pertidaksamaan Linear dan *window* Program Linear. Menu utama Bantuan terdiri dari dua macam pilihan, yaitu bantuan untuk Persamaan Linear dan bantuan untuk Sistem Persamaan Linear.

Layout window sebagai berikut:



gambar 4.3.24. *Layout* untuk *window* W1

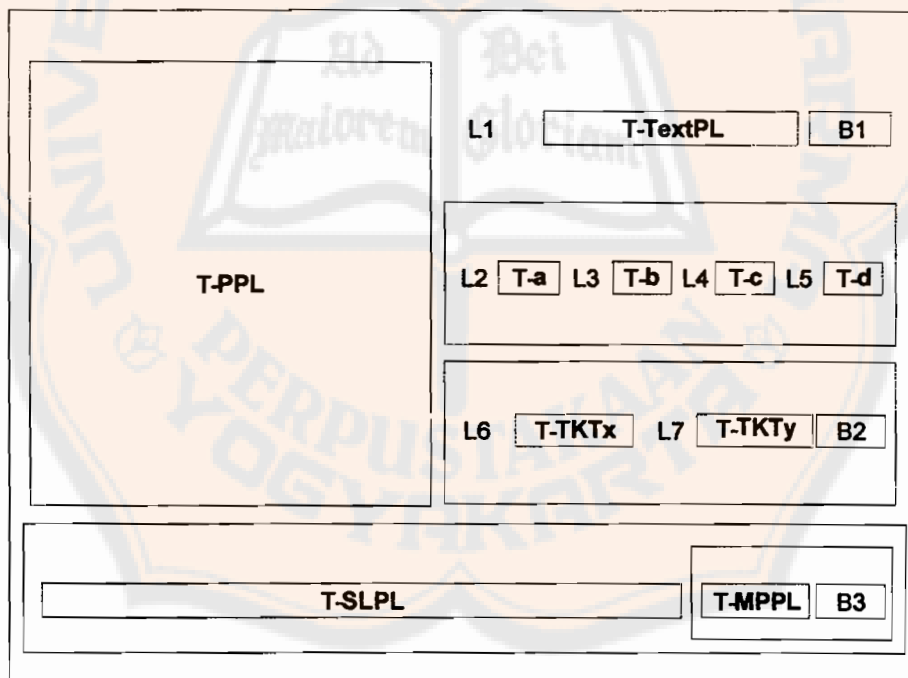
b. Window W2: Pertidaksamaan Linear

Window ini dirancang dapat dibuka dari menu: file/buka/pertidaksamaan_linear yang terdapat pada *window* W1, W3, dan W4. *Window* ini terdiri dari beberapa bagian, yaitu: Elemen *Window*, Elemen Body, Elemen Menu dan Elemen *Layout*.

Elemen *Window* memuat title bar berisi judul *window*, yaitu: *Pertidaksamaan_Linear* dan ukuran *window* yaitu 1024 x 768 pixel.

Elemen Menu memuat Menu Bar yang berisi dua pilihan menu utama, yaitu: File dan Bantuan. Menu File berisi menu Buka, *Reset* dan *Close*. Menu *Close* digunakan untuk menutup *window*. Sedangkan menu Buka digunakan untuk membuka *window* yang lain, yaitu *window* Menu Pembuka, *window* Sistem *Pertidaksamaan Linear* dan *window* Program *Linear*. Menu utama Bantuan terdiri dari empat macam pilihan, yaitu bantuan untuk Plot daerah layak, Pergerakan *pertidaksamaan*, Batas daerah plot dan bantuan untuk Titik uji.

Rancangan *layout window* W2 seperti Gambar 4.3.25 berikut.



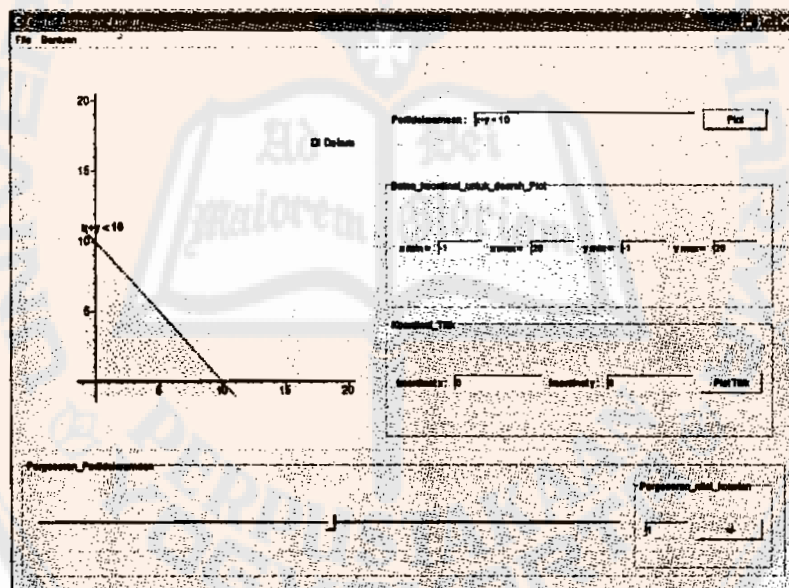
gambar 4.3.25. Rancangan *layout* untuk *window* W2

Keterangan :

- L1 : Label, caption = "Pertidaksamaan"
- L2 : Label, caption = "x min"
- L3 : Label, caption = "x max"

- L4 : Label, caption = "y min"
- L5 : Label, caption = "y max"
- L6 : Label, caption = "Koordinat x"
- L7 : Label, caption = "Koordinat y"
- B1 : Button, caption = "Plot", Action= PTtk
- B2 : Button, caption = "Plot Titik ", Action= PTtk
- B3 : Button, caption = "+/- ", Action= PLPLUS
- S-SLPL : Slider SLPL, Action= ASPTL
- T-TextPL : TextField TextPL, menerima input: pertidaksamaan
- T-a : TextField a, menerima input: nilai x plot min
- T-b : TextField b, menerima input: nilai x plot max
- T-c : TextField c, menerima input: nilai y plot min
- T-d : TextField d, menerima input: nilai y plot max
- T-TKTx : TextField TKTx, menerima input: Koordinat x titik uji
- T-TKTy : TextField TKTy, menerima input: Koordinat y titik uji
- T- MPPL : TextField MPPL, menerima input: konstanta pergeseran dengan tombol B3
- P-PPL : Plotter PPL

Layout window W2 yang dihasilkan tampak seperti pada Gambar 4.3.26.



gambar 4.3.26 Window W2

c. Window W3 : Sistem_Pertidaksamaan_Linear

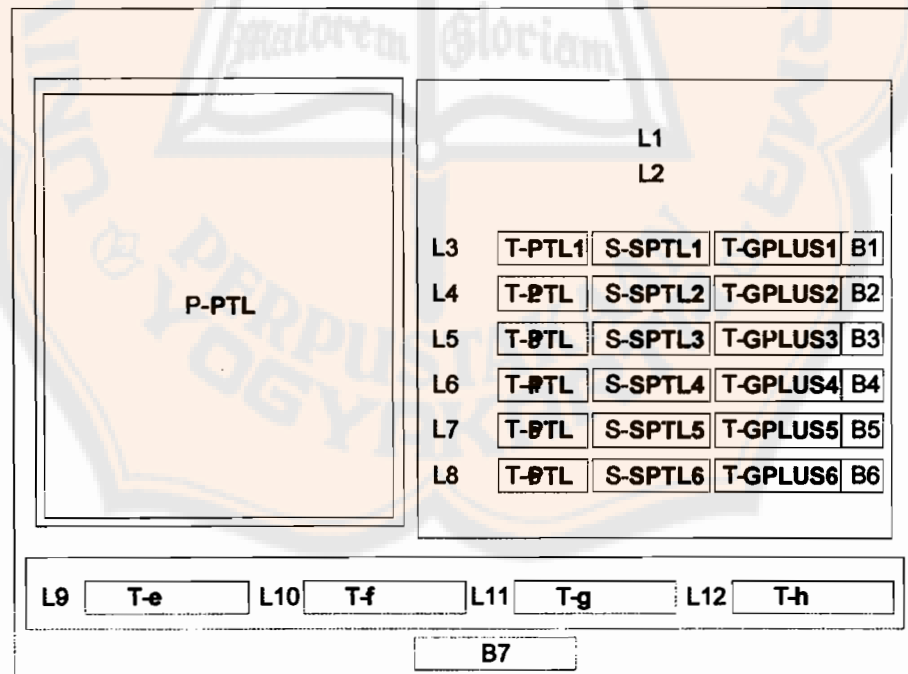
Window ini dapat dibuka dari menu: file/buka/ sistem_pertidaksamaan_linear yang terdapat pada *window* W1, W2, dan W4.

Window ini terdiri dari beberapa bagian, yaitu: Elemen *Window*, Elemen *Body*, Elemen *Menu* dan Elemen *Layout*.

Elemen *Window* memuat *title bar* berisi judul *window*, yaitu: *Sistem_Pertidaksamaan_Linear* dan ukuran *window* yaitu 1024 x 768 pixel.

Elemen *Menu* memuat *Menu Bar* yang berisi dua pilihan menu utama, yaitu: *File* dan *Bantuan*. *Menu File* berisi menu *Buka*, *Reset* dan menu *Close*. *Menu Close* digunakan untuk menutup *window*. Sedangkan menu *Buka* digunakan untuk membuka *window* yang lain, yaitu *window Menu_Pembuka*, *window Pertidaksamaan Linear*, dan *window Program Linear*. *Menu utama Bantuan* terdiri dari tiga macam pilihan, yaitu bantuan untuk *Plot daerah layak*, *Pergerakan pertidaksamaan*, dan bantuan untuk *Batas daerah plot*.

Rancangan *layout window W3* seperti *Gambar 4.3.27* berikut.

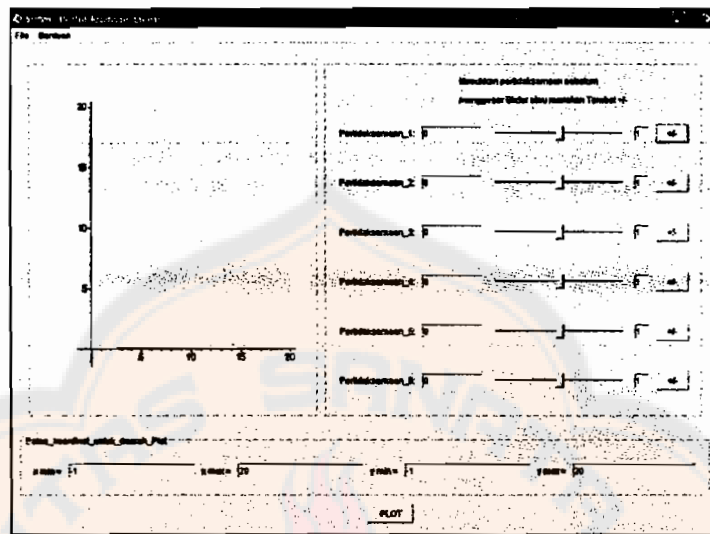


gambar 4.3.27 Rancangan layout window W3

Keterangan:

- L1 : Label, caption = "Masukkan pertidaksamaan sebelum"
- L2 : Label, caption = "Menggeser slider atau menekan tombol +/-"
- L3 : Label, caption = "Pertidaksamaan 1"
- L4 : Label, caption = "Pertidaksamaan 2"
- L5 : Label, caption = "Pertidaksamaan 3"
- L6 : Label, caption = "Pertidaksamaan 4"
- L7 : Label, caption = "Pertidaksamaan 5"
- L8 : Label, caption = "Pertidaksamaan 6"
- L9 : Label, caption = "x min"
- L10 : Label, caption = "x max"
- L11 : Label, caption = "y min"
- L12 : Label, caption = "y max"
- B1 : Button, caption = "+/-", Action= AGPLUS1
- B2 : Button, caption = "+/-", Action= AGPLUS2
- B3 : Button, caption = "+/-", Action= AGPLUS3
- B4 : Button, caption = "+/-", Action= AGPLUS4
- B5 : Button, caption = "+/-", Action= AGPLUS5
- B6 : Button, caption = "+/-", Action= AGPLUS6
- B7 : Button, caption = "PLOT", Action= BPTL
- S-SPTL1 : Slider SPTL1, Action= ASPTL1
- S-SPTL2 : Slider SPTL2, Action= ASPTL2
- S-SPTL3 : Slider SPTL3, Action= ASPTL3
- S-SPTL4 : Slider SPTL4, Action= ASPTL4
- S-SPTL5 : Slider SPTL5, Action= ASPTL5
- S-SPTL6 : Slider SPTL6, Action= ASPTL6
- T- PTL1 : TextField PTL1, menerima input: Pertidaksamaan 1
- T- PTL 2 : TextField PTL2, menerima input: Pertidaksamaan 2
- T- PTL 3 : TextField PTL3, menerima input: Pertidaksamaan 3
- T- PTL 4 : TextField PTL4, menerima input: Pertidaksamaan 4
- T- PTL 5 : TextField PTL5, menerima input: Pertidaksamaan 5
- T- PTL 6 : TextField PTL6, menerima input: Pertidaksamaan 6
- T-e : TextField e, menerima input: nilai x plot min
- T-f : TextField f, menerima input: nilai x plot max
- T-g : TextField g, menerima input: nilai y plot min
- T-h : TextField h, menerima input: nilai y plot max
- T- GPLUS1 : TextField GPLUS1, menerima input: konstanta pergeseran dengan tombol B1
- T- GPLUS2 : TextField GPLUS2, menerima input: konstanta pergeseran dengan tombol B2
- T- GPLUS3 : TextField GPLUS3, menerima input: konstanta pergeseran dengan tombol : B3
- T- GPLUS4 : TextField GPLUS4, menerima input: konstanta pergeseran dengan tombol B4
- T- GPLUS5 : TextField GPLUS5, menerima input: konstanta pergeseran dengan tombol : B5
- T- GPLUS6 : TextField GPLUS6, menerima input: konstanta pergeseran dengan tombol B6
- P-PTL : Plotter PTL

Layout window W3 yang dihasilkan tampak seperti Gambar 4.3.28 berikut.



gambar 4.3.28 Window W3

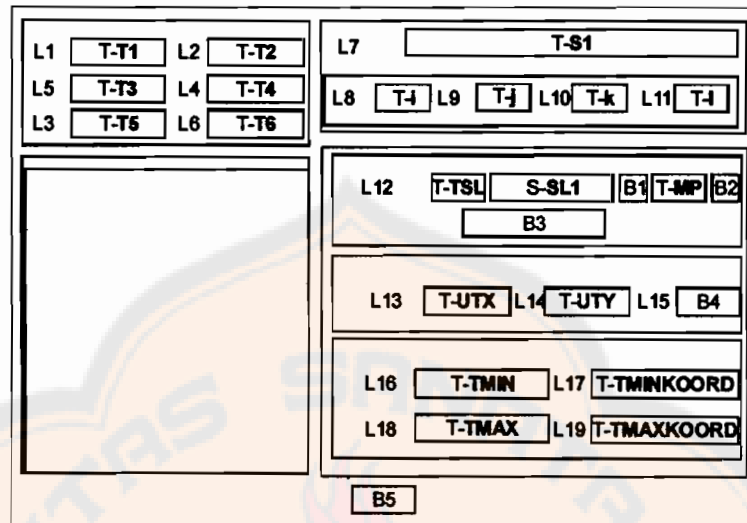
d. Window W4 : Program_Linear

Window ini dapat dibuka dari menu: file/buka/pertidaksamaan_linear yang terdapat pada *window* W1, W2, dan W3. *Window* ini terdiri dari beberapa bagian, yaitu: Elemen *Window*, Elemen Body, Elemen Menu dan Elemen Layout.

Elemen *Window* memuat title bar berisi judul *window*, yaitu: Program_Linear dan ukuran *window* yaitu 1024 x 768 pixel.

Elemen Menu memuat Menu Bar yang berisi dua pilihan menu utama, yaitu: File dan Pengaturan. Menu File berisi menu Buka, Reset dan menu Close. Menu Close digunakan untuk menutup *window*. Sedangkan menu Buka digunakan untuk membuka *window* yang lain, yaitu *window* pembuka, *window* Pertidaksamaan Linear, *window* Sistem Pertidaksamaan Linear dan *window* Sensitifitas. Menu utama Pengaturan terdiri dari dua macam pilihan, yaitu pilihan Perlihatkan Titik Potong dan Perlihatkan Kendala.

Rancangan *layout window* W4 seperti Gambar 4.3.29 berikut.



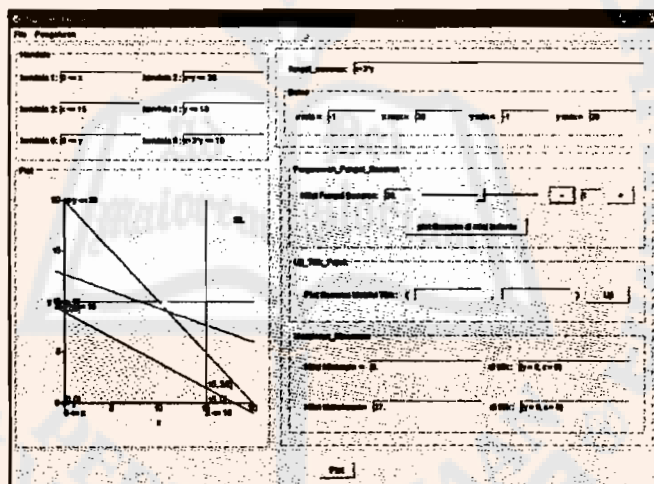
gambar 4.3.29. Rancangan *layout* untuk *window* W4

Keterangan :

- L1 : Label, caption = "Kendala 1"
- L2 : Label, caption = "Kendala 2"
- L3 : Label, caption = "Kendala 3"
- L4 : Label, caption = "Kendala 4"
- L5 : Label, caption = "Kendala 5"
- L6 : Label, caption = "Kendala 6"
- L7 : Label, caption = "Fungsi Sasaran"
- L8 : Label, caption = "x min"
- L9 : Label, caption = "x max"
- L10 : Label, caption = "y min"
- L11 : Label, caption = "y max"
- L12 : Label, caption = "Nilai fungsi sasaran"
- L13 : Label, caption = "Plot sasaran melalui titik :("
- L14 : Label, caption = ","
- L15 : Label, caption = ")"
- L16 : Label, caption = "Nilai minimum :"
- L17 : Label, caption = "di titik :"
- L18 : Label, caption = "Nilai maksimum"
- L19 : Label, caption = "di titik"
- B1 : Button, caption = "-", Action= Minus
- B2 : Button, caption = "+", Action= PLUS
- B3 : Button, caption = "Plot sasaran di nilai tertentu", Action= PT
- B4 : Button, caption = "Uji", Action= UTP
- B5 : Button, caption = "Plot", Action= PLOT
- S-SL1 : Slider SL1, Action= ASL1
- T-T1 : TextField T1, menerima input: kendala 1
- T-T2 : TextField T2, menerima input: kendala 2
- T-T3 : TextField T3, menerima input: kendala 3
- T-T4 : TextField T4, menerima input: kendala 4
- T-T5 : TextField T5, menerima input: kendala 5
- T-T6 : TextField T6, menerima input: kendala 6

- T-i : TextField i, menerima input: nilai x plot min
- T-j : TextField j, menerima input: nilai x plot max
- T-k : TextField k, menerima input: nilai y plot min
- T-l : TextField l, menerima input: nilai y plot max
- T-S1 : TextField S1, menerima input: fungsi sasaran
- T-TSL : TextField TMIN, menampilkan: nilai fungsi sasaran berdasarkan nilai slider SL1
- T-MP : TextField MP, menerima input: konstanta pergeseran dengan tombol B1 atau B2
- T-UTX : TextField UTX, menerima input: ordinat titik uji
- T-UTY : TextField UTY, menerima input: absis titik uji
- T-TMIN : TextField TMIN, menampilkan: nilai minimum penyelesaian program linear
- T-TMINKOORD : TextField TMINKOORD, menampilkan: koordinat nilai minimum penyelesaian program linear
- T-TMAX : TextField TMAX, menampilkan: nilai maksimum penyelesaian program linear
- T-TMAXKOORD : TextField TMAXKOORD, menampilkan: koordinat nilai maksimum penyelesaian program linear

Layout window W4 yang dihasilkan tampak seperti Gambar 4.3.30 berikut.



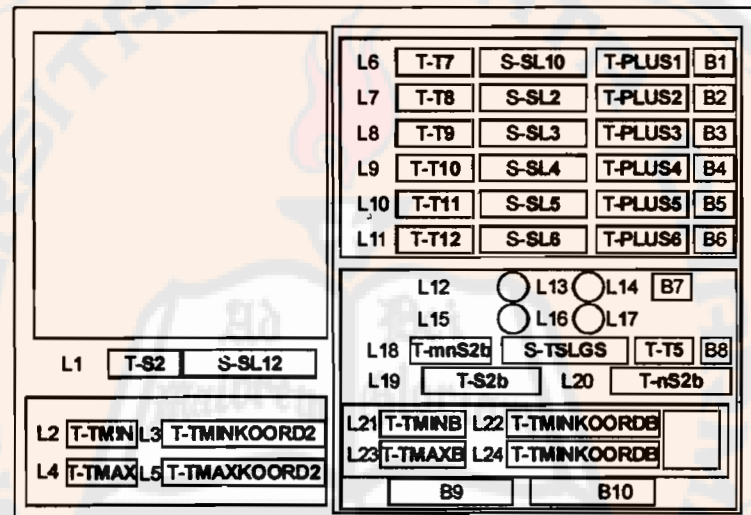
gambar 4.3.30 Window W4

e. Window W5 : Sensitivitas

Window ini dapat dibuka dari menu: file/buka/sensitivitas yang terdapat pada window W4. Window ini terdiri dari beberapa bagian, yaitu: Elemen Window, Elemen Body, Elemen Menu dan Elemen Layout. Elemen Window memuat title bar berisi judul window, yaitu: Sensitivitas dan ukuran window yaitu 1024 x 768 pixel.

Elemen Menu memuat Menu Bar yang berisi dua pilihan menu utama, yaitu: File dan Pengaturan. Menu File berisi menu Close dan menu Kembali. Menu Close digunakan untuk menutup *window*. Sedangkan menu Kembali digunakan untuk membuka *window* Program Linear. Menu utama Pengaturan terdiri dari dua macam pilihan, yaitu pilihan Perlihatkan Titik Potong dan Perlihatkan Kendala.

Rancangan *layout window* W5 seperti Gambar 4.3.31 berikut.



Gambar 4.3.31. Rancangan *layout* untuk *window* W2

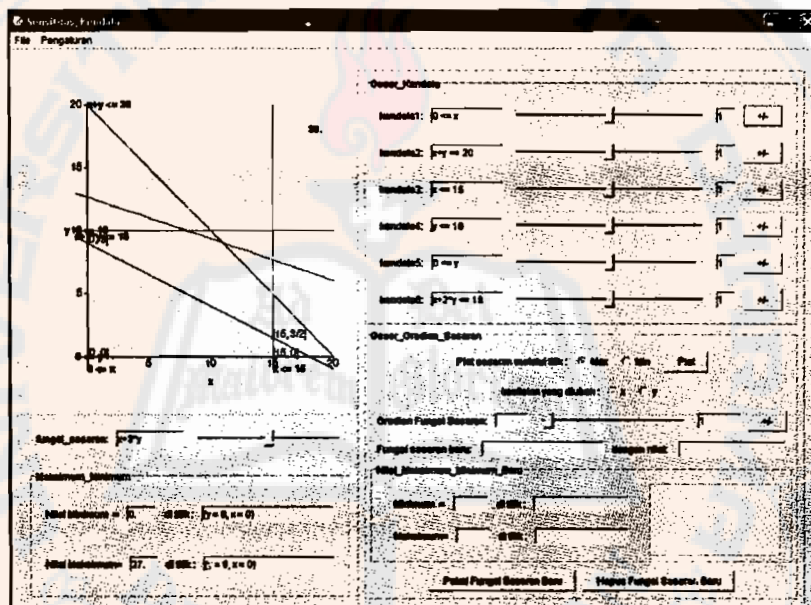
Keterangan:

- L1 : Label, caption = "Fungsi Sasaran"
- L2 : Label, caption = "Nilai Minimum"
- L3 : Label, caption = "di titik"
- L4 : Label, caption = "Nilai Maksimum"
- L5 : Label, caption = "di titik"
- L6 : Label, caption = "Kendala 1"
- L7 : Label, caption = "Kendala 2"
- L8 : Label, caption = "Kendala 3"
- L9 : Label, caption = "Kendala 4"
- L10 : Label, caption = "Kendala 5"
- L11 : Label, caption = "Kendala 6"
- L12 : Label, caption = "Plot sasaran melalui titik :"
- L13 : Label, caption = "max"
- L14 : Label, caption = "min"

L15	: Label, caption = "koefisien ang diubah"
L16	: Label, caption = "x"
L17	: Label, caption = "y"
L18	: Label, caption = "Gradien Fungsi Sasaran"
L19	: Label, caption = "Fungsi Sasaran Baru"
L20	: Label, caption = "dengan nilai :"
L21	: Label, caption = "Minimum"
L22	: Label, caption = "di titik"
L23	: Label, caption = "Maksimum"
L24	: Label, caption = "di titik"
B1	: Button, caption = "+/-", Action= AGPLUS7
B2	: Button, caption = "+/-", Action= AGPLUS8
B3	: Button, caption = "+/-", Action= AGPLUS9
B4	: Button, caption = "+/-", Action= AGPLUS10
B5	: Button, caption = "+/-", Action= AGPLUS11
B6	: Button, caption = "+/-", Action= AGPLUS12
B7	: Button, caption = "Plot", Action= ASLGS1
B8	: Button, caption = "+/-", Action= AGPLUSG
B9	: Button, caption = "Pakai Fungsi Sasaran Baru", Action= Pakai
B10	: Button, caption = "Hapus Fungsi Sasaran Baru", Action= TdkPakai
S-SL12	: Slider SL12, Action= A12
S-SL10	: Slider SL10, Action= A6
S-SL2	: Slider SL2, Action= A7
S-SL3	: Slider SL3, Action= A8
S-SL4	: Slider SL4, Action= A9
S-SL5	: Slider SL5, Action= A10
S-SL6	: Slider SL6, Action= A11
S-SLGS	: Slider SLGS, Action= ASLGS
T-T7	: TextField T1, menampilkan: kendala 1
T-T8	: TextField T2, menampilkan: kendala 2
T-T9	: TextField T3, menampilkan: kendala 3
T-T10	: TextField T4, menampilkan: kendala 4
T-T11	: TextField T5, menampilkan: kendala 5
T-T13	: TextField T6, menampilkan: kendala 6
T-PLUS1	: TextField PLUS1, menerima input: konstanta pergeseran dengan tombol B1
T-PLUS2	: TextField PLUS2, menerima input: konstanta pergeseran dengan tombol B2
T-PLUS3	: TextField PLUS3, menerima input: konstanta pergeseran dengan tombol B3
T-PLUS4	: TextField PLUS4, menerima input: konstanta pergeseran dengan tombol B4
T-PLUS5	: TextField PLUS5, menerima input: konstanta pergeseran dengan tombol B5
T-PLUS6	: TextField PLUS6, menerima input: konstanta pergeseran dengan tombol B6
T-S2	: TextField S2, menampilkan: fungsi sasaran
T-TMIN2	: TextField TMIN2, menampilkan: nilai minimum penyelesaian program linear
T-TMINKOORD2	: TextField TMINKOORD2, menampilkan: koordinat nilai minimum penyelesaian program linear
T-TMAX2	: TextField TMAX2, menampilkan: nilai maksimum penyelesaian program linear
T-TMAXKOORD2	: TextField TMAXKOORD2, menampilkan: koordinat nilai maksimum penyelesaian program linear

- T-mnS2b : TextField mnS2b, menampilkan: Gradien fungsi sasaran baru
 - T-PLUSG : TextField PLUSG, menerima input: konstanta pergeseran dengan tombol B8
 - T-S2b : TextField S2b, menampilkan: fungsi sasaran baru
 - T-nS2b : TextField nS2, menampilkan: nilai fungsi sasaran baru
 - T-TMINB : TextField TMINB, menampilkan: nilai minimum penyelesaian program linear baru
 - T-TMINKOORDB: TextField TMINKOORDB, menampilkan: koordinat nilai minimum penyelesaian program linear baru
 - T-TMAXB : TextField TMAXB, menampilkan: nilai maksimum penyelesaian program linear baru
 - T-TMAXKOORDB: TextField TMAXKOORDB, menampilkan: koordinat nilai maksimum penyelesaian program linear baru
- Tabel

Layout window W5 yang dihasilkan tampak seperti Gambar 4.3.32 berikut.



Gambar 4.3.32. Window W2

f. Window Lainnya

Window lainnya meliputi : Window H1, Window H2, Window H3, Window H4, Window H5, dan Window H6. Window-window memuat label yang berisi keterangan bantuan.

D. Validasi *Interface*

Secara keseluruhan *interface* Program Linear ini mampu membantu siswa untuk melakukan eksplorasi dalam memahami masalah Program Linear. *Interface* ini mampu memplot grafik yang cukup interaktif dengan siswa. Siswa dapat melakukan uji coba mengenai macam-macam kemungkinan daerah penyelesaian dengan menekan tombol atau menggunakan slider.

Ada hal - hal yang perlu diperhatikan demi perbaikan *interface*. Salah satunya adalah mengenai antisipasi kesalahan yang dilakukan oleh siswa. Saat seorang siswa melakukan kesalahan dalam menuliskan sebuah pertidaksamaan, program akan terhenti. Sebenarnya adanya kemungkinan masalah ini sudah diantisipasi dengan penyediaan tombol *restart*. Namun proses eksekusi tombol ini membutuhkan waktu yang cukup lama, sehingga akan menghambat proses pembelajaran.

Interface ini dapat berjalan dengan lancar jika digunakan pada komputer dengan spesifikasi minimal PentiumIII 500MB, RAM 128MB dan dengan Sistem Operasi Windows98 serta tersedia software *Maple8 FullVersion*.

BAB V

PENGUNAAN *INTERFACE MAPLET*

DALAM PEMBELAJARAN PROGRAM LINEAR

Pada bab ini akan disajikan penggunaan *Interface Maplet* untuk pembelajaran Program Linear dalam bentuk *handout*. *Handout* ini disusun menjadi tiga bagian.

Bagian yang pertama adalah *handout* untuk pembelajaran Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear. *Handout* Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear ini nanti akan membimbing siswa untuk menggambar grafik dari pertidaksamaan linear dan sistem pertidaksamaan linear. Di dalam *handout* ini siswa dapat bereksplorasi secara bebas, hingga siswa paham konsep pertidaksamaan linear dan sistem pertidaksamaan linear beserta dengan sifat-sifatnya. Pemahaman mengenai pertidaksamaan linear dan sistem pertidaksamaan linear ini nantinya akan sangat membantu siswa dalam mempelajari Program Linear.

Bagian kedua adalah *handout* untuk pembelajaran Program Linear. *Handout* bagian kedua ini dibagi menjadi dua subbagian, yaitu: Penyelesaian Program Linear dengan uji titik pojok dan Penyelesaian Program Linear dengan Metode Garis selidik.

Bagian terakhir adalah *handout* untuk pembelajaran Masalah Sensitivitas dalam Program Linear.

A. Handout untuk Pembelajaran Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear

LEMBAR KERJA SISWA

Petunjuk Umum

Sebelum menggunakan *Interface* Maplet ini, terlebih dahulu ingatlah mengenai konsep-konsep dalam persamaan linear, cara menggambar grafik persamaan linear beserta dengan sifat-sifatnya dan penyelesaian dari sistem persamaan linear.

Petunjuk Khusus

1. Pokok bahasan : Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear
2. Kelas : III SMA
3. Waktu : 2 x 45 menit
4. Tujuan :

Setelah melakukan kegiatan ini, diharapkan siswa dapat:

- a. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y > b$.
- b. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y < b$.
- c. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y \geq b$.
- d. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y \leq b$.
- e. menentukan keanggotaan suatu titik terhadap sebuah himpunan penyelesaian suatu pertidaksamaan linear.
- f. menentukan penyelesaian suatu sistem pertidaksamaan linear.
- g. menentukan batas-batas penyelesaian suatu sistem pertidaksamaan linear.


h. menentukan macam-macam bentuk penyelesaian sistem pertidaksamaan linear.

5. Alat : Komputer

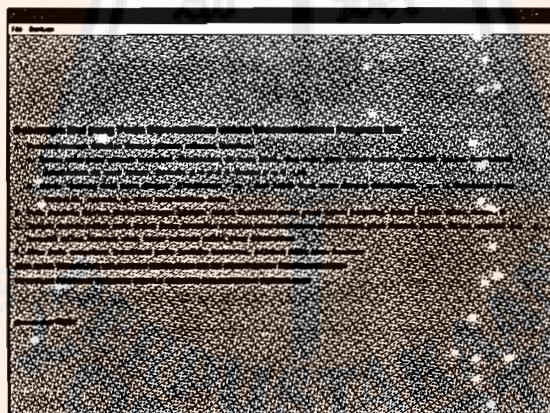
6. Sumber : Buku Matematika

Keterangan:

Ada dua cara untuk memanggil *Interface* Maplet ini, yaitu:

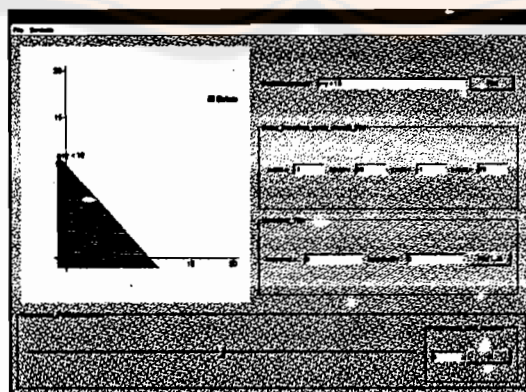
- o Bukalah windows explorer, lalu pilih mydocument. Kemudian bukalah file: Prolin.maplet dengan cara men-*double* klik pada nama file tersebut.
- o Carilah ikon  pada desktop. Setelah itu *double* klik pada ikon tersebut.

Setelah kalian melakukan salah satu prosedur di atas akan tampil *window* seperti ini:



Pada menubar dari window pertama, pilih File/ Buka/ Pertidaksamaan Linear.

Akan tampil window seperti berikut:



Bagian 1 : Memahami Pertidaksamaan Linear Dua Variabel dengan Tanda '<' dan '>'

Buatlah grafik dari pertidaksamaan $-4x + y > 2$, dengan cara mengetikkan pertidaksamaan tersebut pada kolom pertidaksamaan, kemudian klik tombol plot.

Ingat: Kita harus menambahkan tanda '*' di antara konstanta dan variabel.

Contoh: $4x$ diketik $4*x$

Tentukan batas daerah plotnya:

$$x \text{ min} = -5 \quad x \text{ maks} = 10 \quad y \text{ min} = -5 \quad y \text{ maks} = 10$$

• **Pertanyaan 1.a.**

Ketikkan pasangan koordinat $(-1,3)$ pada kolom **titik uji**, kemudian klik tombol uji. Amati keterangan pada bidang plot. Apakah titik ini berada di dalam daerah penyelesaian pertidaksamaan $-4x + y > 2$? Apa artinya?

• **Pertanyaan 1.b.**

Ketikkan pasangan koordinat $(0,2)$ pada kolom **titik uji**, kemudian klik tombol uji. Amati keterangan pada bidang plot. Apakah titik ini berada di dalam himpunan penyelesaian pertidaksamaan $-4x + y > 2$? Apa artinya?

• **Pertanyaan 1.c.**

Ketikkan pasangan koordinat $(2,-1)$ pada kolom **titik uji**, kemudian klik tombol uji. Amati keterangan pada bidang plot. Apakah titik ini berada di dalam daerah penyelesaian pertidaksamaan $-4x + y > 2$? Apa artinya?

- **Pertanyaan 1.d.**

Berdasarkan kegiatan yang telah kamu lakukan pada pertanyaan 1.a. – 1.c. diatas, bagaimana cara menentukan daerah penyelesaian dari suatu pertidaksamaan linear dengan dua variabel.

- **Pertanyaan 1.e.**

Ketikkan angka '1' pada kotak di kolom pergeseran pertidaksamaan. Lalu klik tombol ' +/- '. Amati apa yang terjadi!

Ulangi kegiatan untuk angka '-1'. Amati apa yang terjadi!

Berapakah nilai konstanta c agar titik (-1,3) bukan penyelesaian pertidaksamaan $-4x + y > 2$?

- **Pertanyaan 1.f.**

Buatlah grafik dari pertidaksamaan $-4x + y < 2$, dengan cara seperti di atas.

Tentukan batas daerah plotnya:

$$x \text{ min} = -5 \qquad x \text{ maks} = 10 \qquad y \text{ min} = -5 \qquad y \text{ maks} = 10$$

Bandingkan grafik dari pertidaksamaan $-4x + y < 2$ dengan grafik dari pertidaksamaan $-4x + y > 2$ tadi. Apakah perbedaannya?

Dugaan apa yang dapat kamu katakan tentang hubungan antara himpunan penyelesaian dari pertidaksamaan yang berbentuk $mx + ny > b$ dengan himpunan penyelesaian dari pertidaksamaan yang berbentuk $mx + ny < b$?

- **Pertanyaan 1.g.**

Lengkapilah tabel di bawah ini, dengan terlebih dahulu menggambar grafik yang disebut dengan menggunakan bantuan maplet!

No.	Pertidaksamaan	Kemiringan Gradien (Kanan/ Kiri)	Letak Daerah penyelesaian (Kanan/ Kiri)	Titik Uji (0,0) (Dalam/ Luar)
1.	$2x + 3y < 5$			
2.	$-2x + 3y < 5$			
3.	$2x - 3y < 5$			
4.	$-2x - 3y < 5$			
5.	$2x + 3y < -5$			
6.	$-2x + 3y < -5$			
7.	$2x - 3y < -5$			
8.	$-2x - 3y < -5$			
9.	$2x + 3y > 5$			
10.	$-2x + 3y > 5$			
11.	$2x - 3y > 5$			
12.	$-2x - 3y > 5$			
13.	$2x + 3y > -5$			
14.	$-2x + 3y > -5$			
15.	$2x - 3y > -5$			
16.	$-2x - 3y > -5$			

Bagian 2: Memahami Pertidaksamaan Linear Dua Variabel dengan tanda ‘ \leq ’ dan ‘ \geq ’

• **Pertanyaan**

Gambarlah grafik dari pertidaksamaan $6x + y \leq -4$, dengan cara seperti kegiatan pada bagian pertama.

Batas daerah plotnya:

$$x \text{ min} = -5 \qquad x \text{ maks} = 10 \qquad y \text{ min} = -5 \qquad y \text{ maks} = 10$$

Apakah titik (2,8), (-1,1) dan (1,-10) merupakan penyelesaian dari pertidaksamaan linear $6x + y \leq -4$? Bagaimana caramu mengetahuinya?

Bagian 3: Memahami Sistem Pertidaksamaan Linear Dua Variabel

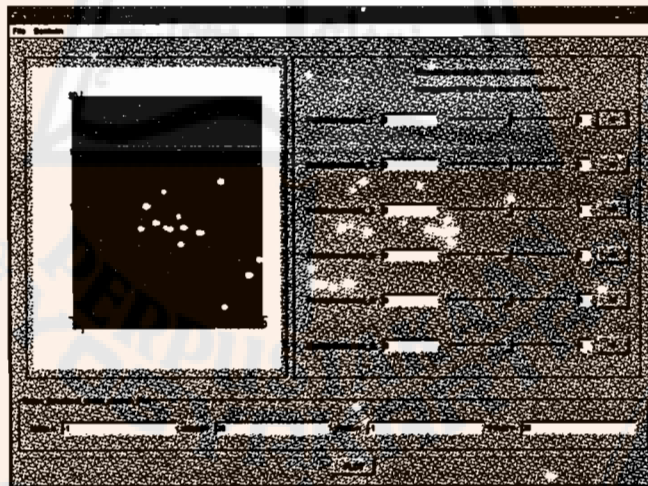
▪ **Pertanyaan 3.a.**

Lengkapilah tabel berikut!

No.	Pertidaksamaan	Titik Uji (di dalam/ di luar)	
		(0, 0)	(3, 3)
1.	$x + y \leq 6$		
2.	$-x + 2y \leq 2$		

▪ **Pertanyaan 3.b.**

Pada menu bar, pilih: File/ Buka/ Sistem Pertidaksamaan Linear. Akan tampil window seperti berikut:



Gambarlah himpunan penyelesaian dari sistem pertidaksamaan linear

$$\begin{cases} x + y \leq 6 \\ -x + 2y \leq 2 \end{cases} \text{ dengan cara memasukkan masing-masing pertidaksamaan}$$

dalam kolom pertidaksamaan yang tersedia. Pada kolom yang tidak dipakai



ketikkan ' 0 '. Cara pengetikan masih sama dengan kegiatan di atas. Setelah itu klik tombol 'Plot'.

Tentukan batas daerah plotnya:

$$x \text{ min} = -1 \quad x \text{ maks} = 10 \quad y \text{ min} = -1 \quad y \text{ maks} = 10$$

Apakah titik (0,0) merupakan penyelesaian dari sistem pertidaksamaan linear tersebut?

Apakah titik (3,3) merupakan penyelesaian dari sistem pertidaksamaan linear tersebut?

Berbentuk apakah himpunan penyelesaian sistem pertidaksamaan linear tersebut? Sebutkan koordinat-koordinat titik sudut-titik sudutnya!

Apa yang bisa kita simpulkan dari hal ini?

▪ **Pertanyaan 3.c.**

Ketikkan ' 1 ' pada kotak di sebelah kanan pertidaksamaan kedua, kemudian klik tombol '+/-'. Apa yang terjadi?

Berbentuk apakah daerah penyelesaiannya sekarang? Tentukan koordinat-koordinat titik sudut-titik sudutnya!

Jika kita klik sekali lagi, apakah daerah penyelesaiannya berubah? Mengapa?

Ketikkan '-1' pada kotak di sebelah kanan pertidaksamaan kedua, kemudian klik tombol '+/-' sebanyak dua kali. Daerah penyelesaian akan kembali ke bentuk semula.

• **Pertanyaan 3.d.**

Ketikkan '- 0.5 ' pada kotak di sebelah kanan pertidaksamaan $x + y \leq 3$. Klik tombol '+/-' sebanyak duabelas kali. Amati perubahan yang terjadi!

Jika kita klik sekali lagi pada tombol '+/-' pertidaksamaan ketiga tersebut, apa yang terjadi? Mengapa?

• **Pertanyaan 3.e.**

Gambarlah SPL $\begin{cases} x \geq 0 \\ y \geq 0 \\ x + y \geq 3 \\ 2x + y \geq 4 \end{cases}$. Dapatkah kita menentukan koordinat-koordinat

titik sudut-titik sudutnya ? Mengapa?

PETUNJUK UNTUK GURU

- Handout : Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear
- Topik : Pertidaksamaan Linear dan Sistem Pertidaksamaan Linear
- Kelas : III SMA
- Waktu : 2 x 45 menit

Petunjuk Umum

Handout ini akan digunakan untuk mempelajari cara menentukan penyelesaian pertidaksamaan linear dan sistem pertidaksamaan linear beserta sifat-sifatnya. Kemampuan minimal yang diperlukan oleh siswa sebelum menggunakan *Interface* ini adalah pemahaman konsep persamaan linear beserta sifat-sifatnya. Terlebih dulu guru dapat menjelaskan secara singkat mengenai konsep penyelesaian pertidaksamaan linear secara manual. Selain itu guru perlu

menjelaskan kepada siswa tentang cara penulisan pertidaksamaan dalam *Interface* Maplet.

Petunjuk Khusus

1. Topik : Pertidaksamaan Linear dua Variabel dan Sistem Pertidaksamaan Linear Dua Variabel

2. Kelas : III SMA

3. Waktu : 2 x 45 menit

4. Tujuan :

Setelah menyelesaikan handout ini diharapkan siswa dapat

- a. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y > b$.
- b. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y < b$.
- c. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y \geq b$.
- d. menentukan penyelesaian pertidaksamaan berbentuk $m x \pm n y \leq b$.
- e. menentukan keanggotaan suatu titik terhadap sebuah himpunan penyelesaian suatu pertidaksamaan linear.
- f. menentukan penyelesaian suatu sistem pertidaksamaan linear.
- g. menentukan batas-batas penyelesaian suatu sistem pertidaksamaan linear.
- h. menentukan macam-macam bentuk penyelesaian sistem pertidaksamaan linear.

5. Pokok-pokok Pelajaran:

- a. Pertidaksamaan linear berbentuk $m x \pm n y > b$
- b. Pertidaksamaan linear berbentuk $m x \pm n y < b$

- c. Pertidaksamaan linear berbentuk $m x \pm n y \geq b$
- d. Pertidaksamaan linear berbentuk $m x \pm n y \leq b$
- e. Sistem pertidaksamaan linear

6. Prosedur Pengajaran:

a. Tugas Guru:

- Sebelum menggunakan *Interface* Maplet ini, guru menjelaskan konsep penyelesaian pertidaksamaan linear secara manual.
- Membimbing dan membantu siswa pada saat penggunaan *Interface* Maplet untuk memahami penyelesaian pertidaksamaan linear dan sistem pertidaksamaan linear.
- Menilai ketercapaian tujuan belajar melalui hasil kerja siswa pada lembar kerja dan lembar evaluasi.

b. Tugas Siswa:

- Memahami tujuan pelajaran.
- Melakukan kegiatan sesuai dengan urutan kegiatan dan petunjuknya.
- Melakukan pengamatan dan menyimpulkan hasil kegiatan.
- Mengerjakan soal latihan pada lembar kerja.
- Mengerjakan tes yang terdapat pada lembar evaluasi

c. Alat dan sumber yang diperlukan:

- Alat:
 - Komputer yang sudah terinstall Maple8.
 - File *interface* Maplet untuk pembelajaran Program Linear (Prolin.maplet)

- Sumber:
 - Buku matematika

7. Evaluasi

a. Prosedur:

- Pertanyaan-pertanyaan lisan dari guru yang membimbing siswa untuk sampai pada proses penemuan.
- Pengisian lembar kerja selama kegiatan berlangsung.

b. Alat Evaluasi: Lembar Kerja

PANDUAN PENGISIAN LEMBAR KERJA SISWA BAGI GURU

Lembar kerja ini terdiri dari tiga bagian. Bagian yang pertama (pertanyaan 1.a. – 1.g.) berisi kegiatan-kegiatan dan pertanyaan-pertanyaan yang membimbing siswa untuk memahami konsep-konsep dan prinsip-prinsip pertidaksamaan linear dua variabel dengan tanda ' $<$ ' dan ' $>$ '. Sedangkan bagian kedua berisi kegiatan-kegiatan dan pertanyaan-pertanyaan yang diarahkan untuk membimbing siswa memahami konsep-konsep dan prinsip-prinsip pertidaksamaan linear dua variabel dengan tanda ' \leq ' dan ' \geq '. Bagian ketiga berisi kegiatan-kegiatan dan pertanyaan-pertanyaan yang diarahkan untuk membimbing siswa memahami konsep-konsep dan prinsip-prinsip sistem pertidaksamaan linear dua variabel.

Guru tidak harus memaksa siswa untuk menghasilkan jawaban seperti pada panduan ini. Diharapkan guru dapat menghargai pendapat siswa dan menanyakan alasan jawaban siswa. Dengan demikian proses yang dilakukan dapat

benar-benar membimbing siswa untuk memahami konsep-konsep dan prinsip-prinsip pertidaksamaan linear dua variabel.

Bagian 1 : Memahami Pertidaksamaan Linear Dua Variabel dengan Tanda '<' dan '>'

- Siswa diminta untuk membuka Window 2: Pertidaksamaan_Linear.
- Siswa diminta untuk membuat grafik dari pertidaksamaan $-4x + y > 2$, dengan cara menyetikkan pertidaksamaan tersebut pada kolom pertidaksamaan, kemudian klik tombol plot. Daerah batas plotnya adalah:
 $x \text{ min} = -5$ $x \text{ maks} = 10$ $y \text{ min} = -5$ $y \text{ maks} = 10$
- Ingatkan siswa untuk menambahkan tanda '*' di antara konstanta dan variabel.
Contoh: $4x$ diketik $4*x$
- Sebagai pembuka, setelah siswa berhasil menggambar pertidaksamaan tersebut dengan menggunakan *Interface* ini, guru dapat menanyakan perbedaan antara grafik dari persamaan linear dua variabel dengan grafik dari pertidaksamaan linear dua variabel.
- Diharapkan siswa dapat melihat bahwa perbedaannya terletak pada daerah penyelesaian. Pada persamaan linear dua variabel, grafik berupa garis lurus. Sehingga penyelesaiannya adalah titik-titik yang berada pada garis tersebut. Sedangkan pada pertidaksamaan linear dua variabel dengan tanda '>' grafik berupa garis putus-putus dengan daerah arsiran (blok berwarna kuning) yang terletak di sebelah kanan garis putus-putus tersebut.

- Pertanyaan 1.a. – 1.d. berikut diharapkan dapat membimbing siswa untuk memahami konsep daerah penyelesaian dan anggota himpunan penyelesaian serta dapat mengerti prinsip untuk menentukan daerah penyelesaian dengan menggunakan titik uji.

- **Pertanyaan 1.a.**

Permasalahan siswa:

Ketikkan pasangan koordinat $(-1,3)$ pada kolom **titik uji**, kemudian klik tombol uji. Amati keterangan pada bidang plot. Apakah titik ini berada di dalam daerah penyelesaian pertidaksamaan $-4x + y > 2$? Apa artinya?

Petunjuk guru:

Diharapkan siswa dapat menjawab bahwa titik ini berada di dalam daerah penyelesaian pertidaksamaan $-4x + y > 2$. Hal ini berarti bahwa titik $(-1,3)$ merupakan salah satu penyelesaian dari pertidaksamaan tersebut. (anggota dari himpunan penyelesaian pertidaksamaan $-4x + y > 2$)

- **Pertanyaan 1.b.**

Permasalahan siswa:

Ketikkan pasangan koordinat $(0,2)$ pada kolom **titik uji**, kemudian klik tombol uji. Amati keterangan pada bidang plot. Apakah titik ini berada di dalam himpunan penyelesaian pertidaksamaan $-4x + y > 2$? Apa artinya?

Petunjuk guru:

Titik ini berapa tepat pada garis putus-putus dari daerah penyelesaian pertidaksamaan $-4x + y > 2$, namun titik ini berada di luar daerah penyelesaian. Diharapkan siswa dapat mengemukakan alasan mengapa titik

tersebut bukan merupakan penyelesaian dari pertidaksamaan $-4x + y > 2$. Salah satu alasan yang dapat diharapkan dapat dikemukakan siswa adalah bahwa jika titik $(0,2)$ disubstitusikan ke pertidaksamaan $-4x + y > 2$ akan menghasilkan pernyataan $2 > 2$, dan pernyataan tersebut adalah salah. Dengan demikian diharapkan pula siswa dapat menemukan arti garis putus-putus pada penyelesaian pertidaksamaan $-4x + y > 2$.

- **Pertanyaan 1.c.**

Permasalahan siswa:

Ketikkan pasangan koordinat $(2,-1)$ pada kolom **titik uji**, kemudian klik tombol uji. Amati keterangan pada bidang plot. Apakah titik ini berada di dalam daerah penyelesaian pertidaksamaan $-4x + y > 2$? Apa artinya?

Petunjuk guru:

Diharapkan siswa dapat menjawab bahwa titik ini berada di luar daerah penyelesaian pertidaksamaan $-4x + y > 2$. Hal ini berarti bahwa titik $(2,-1)$ bukan penyelesaian dari pertidaksamaan tersebut.

- **Pertanyaan 1.d.**

Permasalahan siswa:

Berdasarkan kegiatan yang telah kamu lakukan pada pertanyaan 1.a. – 1.c. diatas, bagaimana cara menentukan daerah penyelesaian dari suatu pertidaksamaan linear dengan dua variabel.

Petunjuk guru:

Pertanyaan ini bertujuan agar siswa dapat menyimpulkan prinsip menggambar penyelesaian pertidaksamaan linear dua variabel dengan bantuan tiga permasalahan sebelumnya.

- **Pertanyaan 1.e.**

Permasalahan siswa:

Ketikkan angka '1' pada kotak di kolom pergeseran pertidaksamaan. Lalu klik tombol '+/-' . Amati apa yang terjadi!

Ulangi kegiatan untuk angka '-1'. Amati apa yang terjadi!

Berapakah nilai konstanta c agar titik $(-1,3)$ bukan penyelesaian pertidaksamaan $-4x + y > 2$?

Petunjuk guru:

Kegiatan ini bertujuan agar siswa dapat mengerti bahwa perbedaan nilai c dari pertidaksamaan $mx + ny < c$ akan menyebabkan perbedaan daerah penyelesaian. Diharapkan siswa dapat menjawab dan mengemukakan alasan bahwa nilai konstanta c pertama yang menyebabkan titik $(-1, 3)$ bukan merupakan penyelesaian dari pertidaksamaan $-4x + y > 2$ adalah 7. Hal ini disebabkan karena jika pertidaksamaan tersebut diubah menjadi $-4x + y > 7$, maka substitusi titik $(-1, 3)$ akan menghasilkan pernyataan $7 > 7$ yang bernilai salah. Titik $(-1,3)$ akan berada tepat pada garis putus-putus yang membatasi daerah penyelesaian pertidaksamaan $-4x + y > 7$.

• **Pertanyaan 1.f.**

Permasalahan siswa:

Buatlah grafik dari pertidaksamaan $-4x + y < 2$, dengan cara seperti di atas.

Tentukan batas daerah plotnya:

$$x \text{ min} = -5 \qquad x \text{ maks} = 10 \qquad y \text{ min} = -5 \qquad y \text{ maks} = 10$$

Bandingkan grafik dari pertidaksamaan $-4x + y < 2$ dengan grafik dari pertidaksamaan $-4x + y > 2$ tadi. Apakah perbedaannya?

Dugaan apa yang dapat kamu katakan tentang hubungan antara himpunan penyelesaian dari pertidaksamaan yang berbentuk $mx + ny > b$ dengan himpunan penyelesaian dari pertidaksamaan yang berbentuk $mx + ny < b$?

Petunjuk guru:

Tujuan dari kegiatan ini adalah membimbing siswa untuk semakin paham tentang penyelesaian pertidaksamaan linear dua variabel. Diharapkan pula siswa dapat membedakan pertidaksamaan yang berbentuk $mx + ny < b$ dan $mx + ny > b$, dilihat dari arah arsirannya. Arah arsiran dari pertidaksamaan linear dua variabel dengan bentuk $mx + ny < b$ berlawanan dengan arah arsiran pertidaksamaan linear dua variabel berbentuk $mx + ny > b$.

• **Pertanyaan 1.g.**

Permasalahan siswa:

Lengkapilah tabel di bawah ini, dengan terlebih dahulu menggambar grafik yang disebut dengan menggunakan bantuan maplet!

No.	Pertidaksamaan	Kemiringan Gradien (Kanan/ Kiri)	Letak Daerah penyelesaian (Kanan/ Kiri)	Titik Uji (0,0) (Dalam/ Luar)
1.	$2x + 3y < 5$			
2.	$-2x + 3y < 5$			
3.	$2x - 3y < 5$			
4.	$-2x - 3y < 5$			
5.	$2x + 3y < -5$			
6.	$-2x + 3y < -5$			
7.	$2x - 3y < -5$			
8.	$-2x - 3y < -5$			
9.	$2x + 3y > 5$			
10.	$-2x + 3y > 5$			
11.	$2x - 3y > 5$			
12.	$-2x - 3y > 5$			
13.	$2x + 3y > -5$			
14.	$-2x + 3y > -5$			
15.	$2x - 3y > -5$			
16.	$-2x - 3y > -5$			

Petunjuk guru:

Diharapkan pula siswa dapat menyimpulkan bahwa jenis kemiringan yang berbeda akan mengakibatkan perbezaan letak daerah penyelesaian. Sehingga pengujian titik hendaknya selalu dilakukan untuk mengetahui arah daerah arsiaran.

Bagian 2: Memahami Pertidaksamaan Linear Dua Variabel dengan tanda

‘≤’ dan ‘≥’

- Siswa diminta untuk membuat grafik dari pertidaksamaan $6x + y \leq -4$, dengan cara seperti kegiatan pada bagian pertama.

Batas daerah plotnya: $x \text{ min} = -5$; $x \text{ maks} = 10$; $y \text{ min} = -5$; $y \text{ maks} = 10$

- Ingatkan siswa bahwa tanda ‘≤’ diketik ‘<=’, sedangkan tanda ‘≥’ diketik ‘>=’

Contoh: ' $6x + y \leq -4$ ' diketik ' $6*x + y <= -4$ '

• **Pertanyaan**

Permasalahan siswa:

Apakah titik (2,8), (-1,1) dan (1,-10) merupakan penyelesaian dari pertidaksamaan linear $6x + y \leq -4$? Bagaimana caramu mengetahuinya?

Petunjuk guru:

Diharapkan siswa sudah dapat menentukan apakah suatu titik merupakan bagian dari penyelesaian suatu pertidaksamaan dengan menggunakan uji titik.

Titik (2,8) bukan merupakan penyelesaian dari pertidaksamaan $6x + y \leq -4$ sebab titik (2,8) berada di luar daerah arsiran. Titik (-1,1) merupakan penyelesaian dari pertidaksamaan $6x + y \leq -4$ sebab titik (-1,1) berada di dalam daerah arsiran. Titik (1,-10) berada pada batas arsiran, dan merupakan penyelesaian dari pertidaksamaan $6x + y \leq -4$. Hal ini berbeda dengan dua bentuk pertidaksamaan di muka. Guru diharapkan dapat membantu siswa untuk melihat perbedaan ini.

- Guru juga dapat bertanya kepada siswa mengenai arti garis putus-putus dan garis lurus.

Bagian 3: Memahami Sistem Pertidaksamaan Linear Dua Variabel

- Kegiatan dalam bagian ini diharapkan dapat membantu siswa untuk memahami sistem pertidaksamaan linear dua variabel.
- Pertanyaan 3.a. dan 3.b. diarahkan untuk membantu siswa memahami konsep penyelesaian sistem pertidaksamaan linear dua variabel.

- Pertanyaan 3.c. diarahkan untuk membantu siswa mengetahui bahwa perubahan nilai konstan pada suatu pertidaksamaan akan mengakibatkan perubahan daerah layak pertidaksamaan tersebut.
- Sedangkan pertanyaan 3.d. dan 3.e. diarahkan untuk membantu siswa memahami jenis-jenis penyelesaian sistem pertidaksamaan linear dua variabel.
- Untuk pertanyaan 3.a., siswa masih bekerja dalam window 2: Pertidaksamaan_Linear. Sedangkan untuk pertanyaan 3.b. – 3.e. siswa bekerja dengan window 3: Sistem_Persamaan_Linear.
- Guru dapat menunjukkan cara pindah window kepada siswa.
- Pada pertanyaan 3.c dan 3.d. siswa akan diminta untuk melakukan pergeseran pertidaksamaan. Selain menggunakan tombol '+/-' , guru dapat mengenalkan tombol 'slider'.
- **Pertanyaan 3.a.**

Permasalahan siswa:

Lengkapilah tabel berikut!

No.	Pertidaksamaan	Titik Uji (di dalam/ di luar)	
		(0, 0)	(3, 3)
1.	$x + y \leq 6$		
2.	$-x + 2y \leq 2$		

Petunjuk guru:

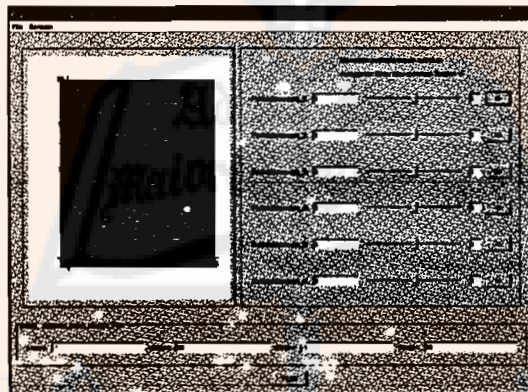
Selain untuk mengetahui kemampuan teknis siswa dalam menentukan penyelesaian pertidaksamaan linear dengan titik uji, pertanyaan ini juga dimaksudkan untuk membantu siswa memahami konsep penyelesaian sistem

pertidaksamaan linear dua variabel. Penyelesaian sistem pertidaksamaan linear dua variabel adalah irisan dari penyelesaian semua pertidaksamaan linear yang menyusunnya. Titik (0,0) berada di dalam daerah penyelesaian kedua pertidaksamaan tersebut. Sedangkan titik (3,3) merupakan anggota penyelesaian pertidaksamaan $x + y \leq 6$ namun bukan anggota penyelesaian pertidaksamaan $-x + 2y \leq 2$.

▪ **Pertanyaan 3.b.**

Permasalahan siswa:

Pada menubar, pilih: File/ Buka/ Sistem Pertidaksamaan Linear. Akan tampil window seperti berikut:



Gambarlah himpunan penyelesaian dari sistem pertidaksamaan linear

$$\begin{cases} x + y \leq 6 \\ -x + 2y \leq 2 \end{cases} \text{ dengan cara memasukkan masing-masing pertidaksamaan}$$

dalam kolom pertidaksamaan yang tersedia. Pada kolom yang tidak dipakai ketikkan '0'. Cara pengetikan masih sama dengan kegiatan di atas. Setelah itu klik tombol 'Plot'.

Tentukan batas daerah plotnya:

$$x \text{ min} = -5 \quad x \text{ maks} = 10 \quad y \text{ min} = -5 \quad y \text{ maks} = 10$$

Apakah titik (0,0) merupakan penyelesaian dari sistem pertidaksamaan linear tersebut?

Apakah titik (3,3) merupakan penyelesaian dari sistem pertidaksamaan linear tersebut?

Berbentuk apakah himpunan penyelesaian sistem pertidaksamaan linear tersebut? Sebutkan koordinat-koordinat titik sudut-titik sudutnya!

Apa yang bisa kita simpulkan dari hal ini?

Petunjuk guru:

Seperti yang telah dikemukakan di muka, mulai pertanyaan 3.b. siswa akan bekerja dengan window 3. Guru membantu siswa untuk pindah ke window 3. Dari pertanyaan ini, diharapkan siswa dapat memahami konsep penyelesaian sistem pertidaksamaan linear dua variabel. Diharapkan siswa dapat memahami bahwa penyelesaian suatu sistem pertidaksamaan linear dua variabel adalah irisan dari penyelesaian semua pertidaksamaan linear dua variabel yang menyusunnya.

Titik (0,0) berada di dalam daerah penyelesaian kedua pertidaksamaan tersebut. Karena itu titik (0,0) merupakan anggota dari irisan penyelesaian pertidaksamaan $x + y \leq 6$ dan $-x + 2y \leq 2$, sehingga titik (0,0) merupakan anggota dari penyelesaian sistem pertidaksamaan linear dua variabel itu. Sedangkan titik (3,3) merupakan anggota penyelesaian pertidaksamaan $x + y \leq 6$ namun bukan anggota penyelesaian pertidaksamaan $-x + 2y \leq 2$. Karena itu titik (3,3) bukan merupakan anggota dari irisan penyelesaian

pertidaksamaan $x + y \leq 6$ dan $-x + 2y \leq 2$, sehingga titik (3,3) bukan anggota dari penyelesaian sistem pertidaksamaan linear dua variabel itu.

▪ **Pertanyaan 3.c.**

Permasalahan siswa:

Ketikkan ' 1 ' pada kotak di sebelah kanan pertidaksamaan kedua, kemudian klik tombol '+/-' . Apa yang terjadi?

Berbentuk apakah daerah penyelesaiannya sekarang? Tentukan koordinat-koordinat titik sudut-titik sudutnya!

Jika kita klik sekali lagi, apakah daerah penyelesaiannya berubah? Mengapa?

Ketikkan '-1' pada kotak di sebelah kanan pertidaksamaan kedua, kemudian klik tombol '+/-' sebanyak dua kali. Daerah penyelesaian akan kembali ke bentuk semula.

Petunjuk guru:

Selain menggunakan tombol '+/-' guru dapat juga mengenalkan slider.

Pertanyaan ini diarahkan untuk membantu siswa mengetahui bahwa perubahan nilai konstan pada suatu pertidaksamaan akan mengakibatkan perubahan daerah layak pertidaksamaan tersebut. Klik yang pertama akan menyebabkan pertidaksamaan $-x + 2y \leq 2$ berubah menjadi $-x + 2y \leq 3$. Hal ini mengakibatkan perubahan daerah penyelesaian pertidaksamaan yang mengakibatkan perubahan penyelesaian sistem pertidaksamaan linear dua variabel.

• **Pertanyaan 3.d.**

Permasalahan siswa:

Tambahkan pertidaksamaan baru dalam sistem tersebut, sedemikian hingga

sistem tersebut berubah menjadi
$$\begin{cases} x + y \leq 6 \\ -x + 2y \leq 2 \\ x \geq 0 \\ y \geq 0 \end{cases}.$$

Kemudian, ubahlah pertidaksamaan $x + y \leq 6$ dengan menggunakan tombol

‘+/-’ atau dengan ‘slider’ menjadi pertidaksamaan $x + y \leq 0$.

Adakah daerah layak nya?

Jika kita klik sekali lagi pada tombol ‘+/-’ di samping pertidaksamaan pertama, apa yang terjadi? Mengapa?

Petunjuk guru:

Penambahan dua pertidaksamaan baru $x \geq 0$ dan $y \geq 0$ pada sistem awal mengakibatkan daerah penyelesaian sistem pertidaksamaan linear dua variabel tersebut dibatasi berada pada kuadran pertama.

Saat pertidaksamaan $x + y \leq 6$ diubah menjadi $x + y \leq 0$, sistem pertidaksamaan linear dua variabel ini memiliki daerah penyelesaian berupa titik (0,0), karena hanya titik itulah yang memenuhi sistem pertidaksamaan linear ini.

Saat tombol ‘+/-’ di-klik sekali lagi, pertidaksamaan $x + y \leq 6$ akan berubah menjadi $x + y \leq -1$. Hal ini mengakibatkan sistem pertidaksamaan linear ini tidak memiliki daerah penyelesaian. Tidak ada titik yang memenuhi sistem

pertidaksamaan ini. Atau dengan kata lain, irisan dari daerah penyelesaian semua pertidaksamaan penyusunnya adalah \emptyset .

• **Pertanyaan 3.e.**

Permasalahan siswa:

Gambarlah SPL $\begin{cases} x \geq 0 \\ y \geq 0 \\ x + y \geq 3 \\ 2x + y \geq 4 \end{cases}$. Dapatkah kita menentukan semua koordinat-

koordinat titik sudut-titik sudutnya ? Mengapa?

Petunjuk guru:

Tidak semua titik sudut ditentukan, karena penyelesaian dari sistem pertidaksamaan ini berupa daerah yang tak terbatas. Guru dapat membantu pemahaman siswa dengan meminta siswa untuk mengubah daerah plot, sehingga siswa benar-benar paham bahwa daerah penyelesaiannya tak terbatas.

B. Handout untuk Pembelajaran Program Linear

LEMBAR KERJA SISWA

Petunjuk Umum

Sebelum menggunakan *Interface* Maplet ini, terlebih dahulu ingatlah mengenai prinsip-prinsip dalam menyelesaikan sistem pertidaksamaan linear dan konsep-konsep dari masalah program linear.

Petunjuk Khusus

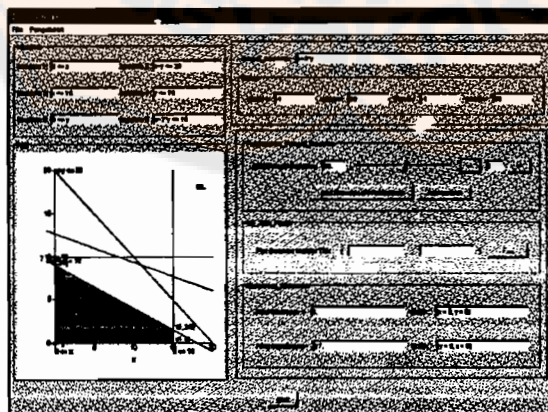
1. Pokok bahasan : Program Linear
2. Kelas : III SMA
3. Waktu : 2 x 45 menit
4. Tujuan :

Setelah melakukan kegiatan ini, diharapkan siswa dapat:

- a. menentukan penyelesaian masalah program linear dengan metode uji titik pojok
 - b. menentukan penyelesaian masalah program linear dengan metode garis selidik
6. Alat : Komputer
 7. Sumber : Buku Matematika

Bagian 1: Penyelesaian Masalah Program Linear dengan Metode Uji Titik Pojok

Dari menu, pilihlah: File/ Buka/ Program Linear. Buka/ Sistem Pertidaksamaan Linear. Akan tampil window seperti berikut:



Diberikan masalah program linear:

Mencari x, y tak negatif yang memenuhi kendala: $x \leq 15, y \leq 10, x + y \leq 20$
 dan memaksimalkan fungsi sasaran $f: 20x + 16y$.

Untuk menyelesaikan masalah tersebut, ketikkan masing-masing kendala pada kolom kendala. Jangan lupa mengetikkan kendala tambahan $x \geq 0$ dan $y \geq 0$ dan memberi "0" pada kolom kendala yang tidak terpakai. Dengan demikian kolom kendala yang terpakai ada lima. (mengapa?)

Ketikkan fungsi sasaran pada kolom fungsi sasaran, dan tentukan batas daerah plotnya: $x \text{ min} = -1$ $x \text{ maks} = 20$ $y \text{ min} = -1$ $y \text{ maks} = 20$
 setelah itu klik tombol 'Plot'.

Amati grafik yang terbentuk. Garis merah adalah gambar fungsi sasaran, dan daerah kuning adalah gambar daerah layak.

Pertanyaan 1.a.

Sebutkan batas-batas dari daerah layak dan tentukan koordinat-koordinat titik pojoknya!

Masukkan masing-masing titik pojok pada kolom uji titik pojok. Klik tombol 'Plot' di sebelahnya dan amati nilai fungsi sasaran yang terlihat di sebelah kanan daerah plot. Catat hasilnya pada tabel berikut.

No.	Koordinat-koordinat Titik Pojok	Nilai fungsi sasaran
1.		
2.		
3.		
4.		

No.	Koordinat-koordinat Titik Pojok	Nilai fungsi sasaran
5.		

Pasangan koordinat-koordinat yang manakah yang menghasilkan nilai fungsi sasaran terbesar?

Pertanyaan 1.b.

Perhatikan posisi grafik fungsi sasaran pada saat nilai sudah optimal dan pada saat nilai belum optimal. Apakah perbedaannya?

Bagian 2: Menentukan Penyelesaian Masalah Program Linear dengan Metode Garis Selidik

Diberikan masalah program linear:

Mencari x, y tak negatif yang memenuhi kendala: $x - y \leq 15, y \leq 10, x + y \leq 20$ dan memaksimalkan fungsi sasaran $f: 20x + 16y - 50$

Untuk menyelesaikan masalah tersebut, ketikkan masing-masing kendala pada kolom kendala. Jangan lupa menyetikkan kendala tambahan $x \geq 0$ dan $y \geq 0$ dan memberi "0" pada kolom kendala yang tidak terpakai. Dengan demikian kolom kendala yang terpakai ada lima. (mengapa?)

Ketikkan fungsi sasaran pada kolom fungsi sasaran, dan tentukan batas daerah plotnya:

$$x \text{ min} = -1 \qquad x \text{ maks} = 20 \qquad y \text{ min} = -1 \qquad y \text{ maks} = 20$$

setelah itu klik tombol 'Plot'.

Amati daerah layak yang terbentuk.

- Ketikkan '0' pada kolom 'Nilai Fungsi Sasaran', lalu klik tombol 'Plot Sasaran di Nilai Tertentu'. Amati letak grafik fungsi sasaran!
- Ketikkan '50' pada kolom 'Nilai Fungsi Sasaran', lalu klik tombol 'Plot Sasaran di Nilai Tertentu'. Amati letak grafik fungsi sasaran!
- Ketikkan '150' pada kolom 'Nilai Fungsi Sasaran', lalu klik tombol 'Plot Sasaran di Nilai Tertentu'. Amati letak grafik fungsi sasaran!

(Grafik fungsi sasaran disini disebut dengan garis selidik)

Apa yang dapat kamu simpulkan dari kegiatan ini?

Semakin nilai fungsi sasaran,
garis selidik bergerak semakin ke-

Dengan demikian, pada masalah ini, jika kita ingin memaksimalkan fungsi sasaran, ke arah mana kita harus menggeser garis selidik?

Berapakah nilai fungsi sasaran yang terbesar?

Pada pasangan koordinat yang manakah nilai tersebut diperoleh?

PETUNJUK UNTUK GURU

- Handout : Program Linear
- Topik : Program Linear
- Subtopik : Menyelesaikan Masalah Program Linear
- Kelas : III SMA
- Waktu : 2 x 45 menit

Petunjuk Umum

Handout ini akan digunakan untuk mempelajari cara menentukan penyelesaian suatu masalah program linear. Kemampuan minimal yang diperlukan oleh siswa sebelum menggunakan *Interface* ini adalah pemahaman konsep pertidaksamaan linear dan sistem pertidaksamaan linear serta penyelesaiannya. Terlebih dulu guru dapat mengingatkan siswa kembali tentang prinsip penyelesaian pertidaksamaan linear secara manual dan menjelaskan secara singkat mengenai beberapa konsep dalam program linear, seperti: kendala, fungsi sasaran, daerah layak, uji titik pojok, garis senilai, dan garis selidik.

Petunjuk Khusus

1. Topik : Program linear
2. Subtopik : Menyelesaikan Masalah Program Linear
3. Kelas : III SMA
4. Waktu : 2 x 45 menit
5. Tujuan :

Setelah menyelesaikan handout ini diharapkan siswa dapat

- a. Menyelesaikan masalah program linear dengan metode uji titik pojok
 - b. Menyelesaikan masalah program linear dengan metode garis selidik
6. Pokok-pokok Pelajaran:
- a. Penyelesaian masalah program linear dengan metode uji titik pojok
 - b. Penyelesaian masalah program linear dengan metode garis selidik

7. Prosedur Pengajaran:

a. Tugas Guru:

- Sebelum menggunakan *Interface* Maplet ini, guru menjelaskan konsep program linear
- Membimbing dan membantu siswa pada saat penggunaan *Interface* Maplet untuk memahami penyelesaian program linear
- Menilai ketercapaian tujuan belajar melalui hasil kerja siswa pada lembar kerja dan lembar evaluasi.

b. Tugas Siswa:

- Memahami tujuan pelajaran.
- Melakukan kegiatan sesuai dengan urutan kegiatan dan petunjuknya.
- Melakukan pengamatan dan menyimpulkan hasil kegiatan.
- Mengerjakan soal latihan pada lembar kerja.
- Mengerjakan tes yang terdapat pada lembar evaluasi

c. Alat dan sumber yang diperlukan:

- Alat:
 - Komputer yang sudah terinstall Maple 8.
 - File *software* Maplet untuk pembelajaran Program Linear (Prolin.maplet)
- Sumber:
 - Buku matematika

8. Evaluasi

a. Prosedur:

- Pertanyaan-pertanyaan lisan dari guru yang membimbing siswa untuk sampai pada proses penemuan.
- Pengisian lembar kerja selama kegiatan berlangsung.

b. Alat Evaluasi:

- Lembar Kerja



PANDUAN PENGISIAN LEMBAR KERJA SISWA BAGI GURU

Lembar kerja ini terdiri dari dua bagian. Lembar kerja bagian pertama diharapkan dapat membantu siswa untuk memahami penyelesaian masalah Program Linear dengan Metode Uji Titik Pojok. Sedangkan bagian kedua diharapkan dapat membantu siswa untuk memahami penyelesaian masalah Program Linear dengan Metode Garis Selidik.

Secara umum, *Interface* Program Linear dan handout ditujukan untuk membantu siswa memahami masalah Program Linear dan metode-metode penyelesaiannya. Karena itu, secara manual, sebaiknya guru menjelaskan dulu beberapa konsep awal dari masalah Program Linear, seperti: Pengertian Masalah Program Linear, Kendala, Fungsi Sasaran, Daerah Layak, dan Garis Selidik. Setelah itu, barulah *Interface* ini dapat dimanfaatkan.

Pada menu bar dalam window ini, terdapat pilihan pengaturan. Dengan pengaturan, plot dari masalah Program Linear dapat diatur dengan pilihan

tampilan nama kendala dan tampilan titik sudut daerah layak. Guru dapat memanfaatkan menu ini untuk melakukan variasi permasalahan.

Dalam lembar kerja ini, siswa akan bekerja dengan menggunakan window 4 dari *Interface* Maplet, yaitu: Program_Linear. Guru dapat membantu siswa untuk membuka window ini.

Bagian 1: Menyelesaikan Masalah Program Linear dengan Metode Uji Titik Pojok

- Permasalahan Siswa:

Diberikan masalah program linear:

Mencari x, y tak negatif yang memenuhi kendala: $x \leq 15, y \leq 10, x + y \leq 20$ dan memaksimalkan fungsi sasaran $f: 20x + 16y$.

Untuk menyelesaikan masalah tersebut, ketikkan masing-masing kendala pada kolom kendala. Jangan lupa mengetikkan kendala tambahan $x \geq 0$ dan $y \geq 0$ dan memberi "0" pada kolom kendala yang tidak terpakai. Dengan demikian kolom kendala yang terpakai ada lima.

Ketikkan fungsi sasaran pada kolom fungsi sasaran, dan tentukan batas daerah

plotnya: $x \text{ min} = -1$ $x \text{ maks} = 20$ $y \text{ min} = -1$ $y \text{ maks} = 20$

setelah itu klik tombol 'Plot'.

Amati grafik yang terbentuk. Garis merah adalah gambar fungsi sasaran, dan daerah kuning adalah gambar daerah layak.

Petunjuk Guru:

Dari kegiatan ini diharapkan siswa semakin paham mengenai konsep-konsep awal masalah Program Linear.

▪ **Pertanyaan 1.a.**

Permasalahan Siswa:

Sebutkan batas-batas dari daerah layak dan tentukan koordinat-koordinat titik pojoknya!

Masukkan masing-masing titik pojok pada kolom uji titik pojok. Klik tombol 'Plot' di sebelahnya dan amati nilai fungsi sasaran yang terlihat di sebelah kanan daerah plot. Catat hasilnya pada tabel berikut.

No.	Koordinat-koordinat Titik Pojok	Nilai fungsi sasaran
1.		
2.		
3.		
4.		
5.		

Pasangan koordinat-koordinat yang manakah yang menghasilkan nilai fungsi sasaran terbesar?

Petunjuk Guru:

Guru dapat melakukan modifikasi kegiatan sebagai berikut. Sebelum siswa meng-klik tombol plot, guru dapat meminta siswa untuk memilih menu pengaturan dan menghilangkan tanda *check* di depan pilihan "tampilkan titik potong". Sehingga siswa harus mencari titik perpotongannya secara manual.

Pasangan koordinat yang memberikan nilai fungsi sasaran inilah yang merupakan titik maksimal dari soal program linear di atas, dan nilai fungsi sasaran yang bersesuaian adalah nilai fungsi maksimum dari soal tersebut.

Kemudian, guru dapat meminta siswa untuk membandingkan hasil yang mereka peroleh dengan hasil yang sudah ditampilkan oleh *Interface* pada kolom “Maximum_Minimum”!

▪ **Pertanyaan 1.b.**

Permasalahan siswa:

Perhatikan posisi grafik fungsi sasaran pada saat nilai sudah optimal dan pada saat nilai belum optimal. Apakah perbedaannya?

Petunjuk Guru:

Pertanyaan ini ditampilkan untuk membantu siswa untuk memahami permasalahan berikutnya: menentukan penyelesaian masalah Program Linear dengan metode garis selidik. Pada keadaan optimal (nilai maksimum ataupun minimum) garis selidik memotong daerah layak hanya di satu titik.

Bagian 2: Menentukan Penyelesaian Masalah Program Linear dengan Metode Garis Selidik

▪ Permasalahan Siswa:

Diberikan masalah program linear:

Mencari x, y tak negatif yang memenuhi kendala: $x - y \leq 15, y \leq 10,$

$x + y \leq 20$ dan memaksimalkan fungsi sasaran $f: 20x + 16y - 50$

Untuk menyelesaikan masalah tersebut, ketikkan masing-masing kendala pada kolom kendala. Jangan lupa mengetikkan kendala tambahan $x \geq 0$ dan $y \geq 0$

dan memberi “0” pada kolom kendala yang tidak terpakai. Dengan demikian kolom kendala yang terpakai ada lima

Ketikkan fungsi sasaran pada kolom fungsi sasaran, dan tentukan batas daerah plotnya:

$$x \text{ min} = -1 \quad x \text{ maks} = 20 \quad y \text{ min} = -1 \quad y \text{ maks} = 20$$

setelah itu klik tombol ‘Plot’.

Amati daerah layak yang terbentuk.

- Ketikkan ‘0’ pada kolom ‘Nilai Fungsi Sasaran’, lalu klik tombol ‘Plot Sasaran di Nilai Tertentu’. Amati letak grafik fungsi sasaran!
- Ketikkan ‘50’ pada kolom ‘Nilai Fungsi Sasaran’, lalu klik tombol ‘Plot Sasaran di Nilai Tertentu’. Amati letak grafik fungsi sasaran!
- Ketikkan ‘150’ pada kolom ‘Nilai Fungsi Sasaran’, lalu klik tombol ‘Plot Sasaran di Nilai Tertentu’. Amati letak grafik fungsi sasaran!

Apa yang dapat kamu simpulkan dari kegiatan ini?

Semakin nilai fungsi sasaran,
garis selidik bergerak semakin ke-

Dengan demikian, pada masalah ini, jika kita ingin memaksimalkan fungsi sasaran, ke arah mana kita harus menggeser garis selidik?

Berapakah nilai fungsi sasaran yang terbesar?

Pada pasangan koordinat yang manakah nilai tersebut diperoleh?

Petunjuk Guru:

Kegiatan memasukkan nilai-nilai tertentu pada kolom ‘Nilai Fungsi Sasaran’ diharapkan akan membantu siswa untuk memahami bahwa perubahan nilai

fungsi sasaran akan membuat fungsi sasaran bergerak sejajar ke arah tertentu.

Mintalah siswa untuk benar-benar mengamati perubahan nilai tersebut.

Selanjutnya, diharapkan siswa dapat menyimpulkan sendiri mengenai pergeseran fungsi sasaran sesuai dengan permintaan soal.

Guru dapat memberikan permasalahan lain yang diharapkan dapat menambah pemahaman siswa mengenai pergeseran garis selidik.

C. *Handout* untuk Pembelajaran Masalah Sensitivitas pada Program Linear

PETUNJUK PENGGUNAAN PROGRAM

Handout	: Program Linear
Topik	: Masalah Sensitivitas dalam Program Linear
Semester	: III / PMat / USD
Waktu	: 2 x 50 menit

Petunjuk Umum

Handout ini akan digunakan untuk mempelajari cara menentukan penyelesaian suatu masalah sensitivitas program linear, terutama untuk mengatasi kesulitan visual. Kemampuan minimal yang diperlukan oleh mahasiswa sebelum menggunakan *Interface* ini adalah pemahaman konsep program linear dan penyelesaiannya. Terlebih dulu dosen dapat menjelaskan secara singkat mengenai konsep sensitivitas.

Petunjuk Khusus

1. Topik : Program linear
2. Subtopik : Masalah Sensitivitas dalam Masalah Program Linear
3. Semester : III / PMat/ USD
4. Waktu : 2 x 50 menit

5. Tujuan :
Setelah menyelesaikan handout ini diharapkan mahasiswa dapat
 - a. Menyelesaikan masalah sensitivitas perubahan fungsi sasaran.
 - b. Menyelesaikan masalah sensitivitas perubahan kendala
6. Pokok-pokok Pelajaran:
 - a. Penyelesaian masalah sensitivitas perubahan fungsi sasaran.
 - b. Penyelesaian masalah sensitivitas perubahan kendala
7. Prosedur Pengajaran:
 - a. Tugas Dosen:
 - Sebelum menggunakan *Interface* Maplet ini, dosen menjelaskan konsep sensitivitas
 - Membimbing dan membantu mahasiswa pada saat penggunaan *Interface* Maplet untuk memahami penyelesaian masalah sensitivitas pada program linear
 - Menilai ketercapaian tujuan belajar melalui hasil kerja siswa pada lembar kerja dan lembar evaluasi.

b. Tugas Mahasiswa:

- Memahami tujuan pelajaran.
- Melakukan kegiatan sesuai dengan urutan kegiatan dan petunjuknya.
- Melakukan pengamatan dan menyimpulkan hasil kegiatan.
- Mengerjakan soal latihan pada lembar kerja.
- Mengerjakan tes yang terdapat pada lembar evaluasi

c. Alat dan sumber yang diperlukan:

- Alat:
 - Komputer yang sudah terinstall Maple 8.
 - File *interface* Maplet untuk pembelajaran Program Linear (Prolin.maplet)
- Sumber:
 - Buku matematika

8. Evaluasi

a. Prosedur:

- Pertanyaan-pertanyaan lisan dari dosen yang membimbing siswa untuk sampai pada proses penemuan.
- Pengisian lembar kerja selama kegiatan berlangsung.

b. Alat Evaluasi:

- Latihan C diktat halaman 77.

LEMBAR KERJA MAHASISWA**Petunjuk Umum**

Sebelum menggunakan *Interface* Maplet ini, terlebih dahulu ingatlah mengenai prinsip-prinsip dalam penyelesaian masalah program linear.

Petunjuk Khusus

1. Topik : Program linear
2. Subtopik : Masalah Sensitivitas dalam Masalah Program Linear
3. Semester : III / PMat/ USD
4. Waktu : 2 x 50 menit
5. Tujuan :
Setelah menyelesaikan handout ini diharapkan mahasiswa dapat
 - a. Menyelesaikan masalah sensitivitas perubahan fungsi sasaran.
 - b. Menyelesaikan masalah sensitivitas perubahan kendala
6. Alat : Komputer
7. Sumber : Diktat Kuliah Program Linear (B. Susanta)

Bagian 1: Masalah Sensitivitas Perubahan Fungsi Sasaran

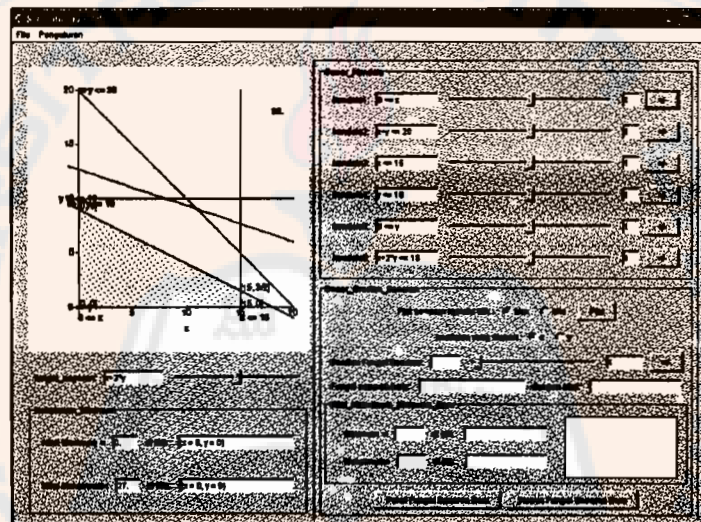
- Bacalah diktat kuliah Program Linear halaman 72 – 75 (Sensitivitas). Berikut ini adalah kegiatan yang dapat membantu Anda memahami contoh C.1. pada halaman 73 dari diktat tersebut.

- Permasalahan:

Mencari x dan y yang memenuhi: $x \geq 0$; $y \geq 0$; $2x + 5y \geq 600$;

$4x + 3y \leq 530$; $2x + y \leq 240$ dan memaksimumkan $f: 32x + 20y$.

- Selesaikan permasalahan ini dengan menggunakan *Interface* Program Linear window keempat: Program_Linear.
- Setelah itu, bukalah window lima: Sensitivitas, dengan cara memilih menu file\buca\sensitivitas_kendala. Muncul window sebagai berikut.



- Bila koefisien ongkos dari dari fungsi sasaran berubah, sejauh manakah perubahan itu boleh terjadi supaya penyelesaian optimum (p.o) tidak berubah?
- Geser slider pada subwindow Geser_Gradien_Sasaran. (Sebelumnya pilih tombol radiobutton pada maksimum dan x)
- Amati perubahan kendala pada daerah plot.
- Catatlah selang nilai gradien kendala yang memenuhi persoalan ini.
- Bukalah halaman 77, kerjakan Latihan C soal no. 1!

- Untuk membuat masalah Program Linear yang baru, pilihlah menu file/kembali.

Bagian 2: Masalah Sensitivitas Perubahan Kendala

- Bacalah diktat kuliah Program Linear halaman 75 – 77 (Sensitivitas). Berikut ini adalah kegiatan yang dapat membantu Anda memahami contoh C.2. pada halaman 75 dari diktat tersebut.
- Permasalahan:
Mencari x dan y yang memenuhi: $x \geq 0$; $y \geq 0$; $x + y \leq 8$; $3x - y \geq 0$; $y \leq q$
dan memaksimumkan $f: 10x + 20y$.
Berapa sajakah nilai q supaya soal di atas mempunyai p.o. dan berapa nilai program yang sesuai?
- Selesaikan permasalahan ini dengan *Interface* Program Linear window keempat. Untuk permulaan, berilah nilai $q = 10$
- Setelah itu, bukalah window kelima.
- Geser slider di sebelah kendala terakhir. Amati perubahan daerah layak!
- Catatlah semua nilai q yang menyebabkan soal tersebut memiliki p.o.!
- Kerjakanlah Latihan C no. 2 dan 3 dari diktat halaman 77!

- Untuk membuat masalah Program Linear yang baru, pilihlah menu file/kembali.

Bagian 2: Masalah Sensitivitas Perubahan Kendala

- Bacalah diktat kuliah Program Linear halaman 75 – 77 (Sensitivitas). Berikut ini adalah kegiatan yang dapat membantu Anda memahami contoh C.2. pada halaman 75 dari diktat tersebut.

- Permasalahan:

Mencari x dan y yang memenuhi: $x \geq 0$; $y \geq 0$; $x + y \leq 8$; $3x - y \geq 0$; $y \leq q$
dan memaksimumkan $f: 10x + 20y$.

Berapa sajakah nilai q supaya soal di atas mempunyai p.o. dan berapa nilai program yang sesuai?

- Selesaikan permasalahan ini dengan *Interface* Program Linear window keempat. Untuk permulaan, berilah nilai $q = 10$
- Setelah itu, bukalah window kelima.
- Geser slider di sebelah kendala terakhir. Amati perubahan daerah layak!
- Catatlah semua nilai q yang menyebabkan soal tersebut memiliki p.o.!
- Kerjakanlah Latihan C no. 2 dan 3 dari diktat halaman 77!

BAB VI

KESIMPULAN DAN SARAN

Dalam Bab ini, diberikan beberapa kesimpulan dari skripsi ini dan beberapa saran yang dapat dimanfaatkan.

A. Kesimpulan

Dari pembahasan pada bab-bab diatas dapat ditarik beberapa kesimpulan sebagai berikut:

1. Perintah *inequal* dalam paket *Plot*, membuat *Maple8* mampu digunakan untuk menggambar daerah penyelesaian dari suatu sistem pertidaksamaan linear dua variabel atau menggambarkan daerah layak dari suatu masalah Program Linear dua variabel. Dengan paket *Simplex*, *Maple8* dapat digunakan untuk mencari penyelesaian optimum dari masalah Program Linear dua variabel, meliputi: perintah *maximize* untuk mencari nilai maksimum dan *minimize* untuk mencari nilai minimum dari suatu masalah program linear. Selain itu dengan perintah *subs*, *Maple8* dapat digunakan mencari nilai optimum dari fungsi sasaran jika penyelesaian optimum sudah didapat. Hasil eksplorasi mengenai manfaat dan cara penggunaan dari masing-masing fasilitas ini telah dipaparkan pada Bab III subbab A
2. Perintah-perintah dalam *Maple8* yang dapat digunakan untuk membantu penyusunan prosedur-prosedur dalam *interface Maplet* meliputi: perintah *solve* untuk menyelesaikan suatu persamaan, *evalf* untuk mengubah bilangan

ke bentuk desimal dengan menentukan banyak tampilan digit, *evalb* untuk mengevaluasi sebuah pernyataan boolean, *type* untuk menguji jenis pernyataan, *abs* untuk mencari nilai mutlak dari suatu bilangan, *coeff* untuk menampilkan koefisien suatu variabel dalam suatu pernyataan aljabar, *convert* untuk mengubah suatu relasi pernyataan menjadi relasi pernyataan bentuk lain, *lhs* dan *rhs* untuk menampilkan ruas kiri atau ruas kanan suatu persamaan atau pertidaksamaan, *slope* untuk mencari gradien suatu persamaan garis, *union* dan *subset* untuk menggabungkan dua buah himpunan dan mengecek apakah suatu himpunan merupakan himpunan bagian dari himpunan lain. Selain itu dibutuhkan juga perintah-perintah untuk menggambar yang termasuk dalam paket *Plot*, meliputi: perintah *plot* untuk menggambar grafik suatu fungsi, *textplot* untuk menggambar teks pada bidang gambar, dan *pointplot* untuk membuat titik pada bidang gambar 2 dimensi. Hasil eksplorasi mengenai manfaat dan cara penggunaan dari masing-masing fasilitas ini telah dipaparkan pada Bab III subbab B.

3. Fasilitas-fasilitas *Maplet* yang dipakai untuk penyusunan *interface* dibedakan menjadi lima kategori, yaitu: elemen *window*, elemen bodi *window*, elemen menu, elemen perintah, dan elemen layout. Salah satu elemen bodi *window* yang membuat *interface* yang telah disusun dapat digunakan untuk membantu pembelajaran Program Linear dua variabel adalah elemen *slider*. Dengan elemen slider, siswa sebagai pengguna *interface* dapat melakukan eksplorasi dengan cara menggesernya. Hasil eksplorasi mengenai manfaat dan cara

penggunaan dari masing-masing fasilitas ini telah dipaparkan pada Bab III subbab C.

4. Berhasil disusun sebuah *interface* untuk membantu pembelajaran penyelesaian masalah Program Linear dua variabel dengan Metode Grafik. Penyusunan *interface* Program Linear ini dibagi menjadi dua bagian, yaitu bagian prasyarat yang berisi materi pertidaksamaan linear dan sistem pertidaksamaan linear, dan bagian materi pokok yaitu masalah Program Linear dan masalah sensitivitas. Masing-masing materi di atas disusun dalam *window* tersendiri. Sehingga secara keseluruhan, ada lima *window* dalam rancangan *Interface* ini, termasuk halaman pembuka. Dalam setiap *window*, siswa dituntun untuk dapat bereksplorasi secara bebas, agar dapat semakin memahami konsep. Siswa juga dapat mengukur dan menyesuaikan sendiri kecepatan belajar sesuai dengan kemampuannya. Hasil penyusunan *interface* yang telah dipaparkan pada Bab IV.
5. Pada bagian terakhir dari skripsi ini disusun pula sebuah *handout* yang diharapkan dapat membantu siswa dalam mempelajari Program Linear dua variabel dengan menggunakan *interface Maple* yang telah disusun.

B. Saran

Bagian ini berisi saran yang dapat digunakan untuk menggunakan dan mengembangkan pembelajaran Program Linear dua variabel dengan menggunakan *Maple8*.

1. Rancangan *interface* dan *handout* hasil skripsi ini belum diujicobakan di sekolah. Oleh karena itu sebelum dimanfaatkan untuk pembelajaran nyata, sebaiknya diadakan uji coba terlebih dahulu.
2. Masalah Program Linear tidak hanya terbatas untuk dua variabel. Oleh karena itu eksplorasi ini dapat dikembangkan untuk masalah Program Linear yang melibatkan lebih dari dua variabel.
3. *Interface* yang dibuat dapat digunakan secara klasikal maupun individual. Oleh karena itu, pemanfaatannya perlu memperhatikan ketersediaan komputer di sekolah. Apabila jumlah komputer mencukupi untuk penggunaan secara individual, guru dapat berperan sebagai fasilitator dan pembimbing untuk mendorong siswa melakukan eksplorasi. Namun bila jumlah komputer tidak memenuhi, maka guru dapat memanfaatkan *interface* yang telah ada untuk pembelajaran klasikal dengan bantuan *LCD Projector*. Dalam hal ini diharapkan guru dapat memberikan berbagai macam kasus yang dapat diamati dengan menggunakan *interface* tersebut.

DAFTAR PUSTAKA

- B. Susanta. 1994. Diktat Kuliah Program Linear. Yogyakarta.
- Douglas B. Meade. 2002. *Maple8 and Maplets: A New Use of Computer Algebra for Teaching Mathematics and Science*, University of South Carolina.
<http://www.bmcc.cuny.edu/2002/presentations/meade.html>
- Frank Garvan. 2002. *The MAPLE BOOK*. Washington, D. C.: Press Company
- Greg Gamble & Grant Keady. 2003. *MapleNet and Maplets under Maple 8, MSOR Connections Vol. 3 No. 1 February 2003*.
<http://www.adeptscience.co.uk>.
- http://www.kota-makassar.net/umum/SLTP/Pengembangan%20Software/PENGEMBANGAN%20SOFTWARE%20_April%202003_2.pdf.
- <http://www.ExploreMath.com/LessonPlan/LinearEnequalities.pdf>
- Ian Sommerville. 2003. *Software Engineering (Rekayasa Perangkat Lunak)*. Edisi 6. Jakarta: Erlangga
- 2003. Kurikulum Berbasis Kompetensi Sekolah Menengah Atas Dan Madrasah Aliyah. Jakarta: Departemen Pendidikan Nasional.
- 2001. *Maplet Begginer's Guide*. Canada: Waterloo Maple Inc.
- Michael McCabe. 2004. *Maple goes GUI with Maplets- A Review of Maple 8, MSOR Connections Vol. 2 No. 4 November 2002*.
<http://www.adeptscience.co.uk>.
- P.H. Stikker. 2002. *Procedures and a Maplet for Analyzing Linear Programming Problems* . p.stikker@hsholland.nl.

Sartono Wirodikromo. 2002. Matematika Jilid 5 untuk SMA Kelas XII Semester

I. Jakarta: Penerbit Erlangga,



PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

LAMPIRAN



PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
> restart:
with(student):
with(simplex):
with(plots):
setoptions(scaling=CONSTRAINED):
with(Maplets[Elements]):
Digits:=3;
Warning, the protected names maximize and minimize have been redefined and unprotected
Warning, the names changecoords and display have been redefined
```

Digits := 3

[>

- Prosedur

- Koordinat Plot untuk Teks Kendala

```
> koordplotteks:=proc(p,a,b,c,d)
local ii,i,h;

ii:=lhs(p)=rhs(p):
i:=solve(ii,y):
h:=subs(x=a,i):
if i<>NULL then
if h>d then
solve(d=i,x),d;
else
if solve(subs(x=a,ii),y)<c then
solve(subs(y=c,ii),x),c:
else
a,h:
end if:
end if:
else
solve(ii,x),c:
end if:

end proc;
koordplotteks := proc(p, a, b, c, d)
local ii, i, h;
ii := lhs(p) = rhs(p);
i := solve(ii, y);
h := subs(x = a, i);
if i ≠ NULL then
if d < h then solve(d = i, x), d
else if solve(subs(x = a, ii), y) < c then solve(subs(y = c, ii), x), c else a, h end if
end if
else solve(ii, x), c
end if
end proc
```

- Pertidaksamaan Linear

```
> Pertidaksamaan:=proc(p,tx,ty,a,b,c,d)
local plot_PL,tp,ttk,plot_ttk,tpPL:
plot_PL:=inequal({p},x=a..b,y=c..d,optionsfeasible=(color=yellow),
optionsexcluded=(color=white),optionsclosed=(color=navy),thickness=2);
if evalb(subs(x=tx,y=ty,p))=true then
tp:=textplot([b-3,d-3,"Di Dalam"],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else tp:=textplot([b-3,d-3,"Di Luar"],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
end if;
ttk:=[tx,ty];
plot_ttk:=pointplot(ttk,symbol=CIRCLE,color=RED,symbolsize=25);
tpPL:=textplot([koordplotteks(p,a,b,c,d),p],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
```


PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
display(plot_PL,plot_ttk,tp,tpPL);

end proc;

Pertidaksamaan := proc(p, tx, ty, a, b, c, d)
local plot_PL, tp, ttk, plot_ttk, tpPL;
  plot_PL := inequal({p}, x = a .. b, y = c .. d, optionsfeasible = (color = yellow),
    optionsexcluded = (color = white), optionsclosed = (color = navy), thickness = 2);
  if cvalb(subs(x = tx, y = ty, p)) = true then
    tp := textplot([b - 3, d - 3, "Di Dalam"], color = blue, align = RIGHT, font = [HELVETICA, BOLD, 10])
  else tp := textplot([b - 3, d - 3, "Di Luar"], color = blue, align = RIGHT, font = [HELVETICA, BOLD, 10])
  end if;
  ttk := {[tx, ty]};
  plot_ttk := pointplot(ttk, symbol = CIRCLE, color = RED, symbolsize = 25);
  tpPL :=
    textplot([koordplotteks(p, a, b, c, d), p], color = blue, align = RIGHT, font = [HELVETICA, BOLD, 10]);
  display(plot_PL, plot_ttk, tp, tpPL)
end proc
[ >
```

- Pergeseran Kendala

```
> geser_kendala:=proc (kendala_geser,m,a,b,c,d)
local g,h,l,sl_kendala,sl_cek,sl_max,sl_min,nilai_kanan,nilai_kiri,btx,bty;

sl_kendala:=lhs(kendala_geser)=rhs(kendala_geser);
sl_cek:=subs(x=0,sl_kendala);
if type(lhs(sl_cek),'constant') and type(rhs(sl_cek),'constant')=true then
sl_min:=a;
sl_max:=b;
else
if slope(y=solve(sl_kendala,y))>0 then
if type(lhs(sl_kendala),'constant')=true then
nilai_kanan:=rhs(sl_kendala);
sl_min:=subs(x=a,y=d,nilai_kanan);
sl_max:=subs(x=b,y=c,nilai_kanan);
else
nilai_kiri:=lhs(sl_kendala);
sl_min:=subs(x=a,y=d,nilai_kiri);
sl_max:=subs(x=b,y=c,nilai_kiri);
end if;
else
if slope(y=solve(sl_kendala,y))<0 then
if type(lhs(sl_kendala),'constant')=true then
nilai_kanan:=rhs(sl_kendala);
sl_min:=subs(x=a,y=c,nilai_kanan);
sl_max:=subs(x=b,y=d,nilai_kanan);
else
nilai_kiri:=lhs(sl_kendala);
sl_min:=subs(x=a,y=c,nilai_kiri);
sl_max:=subs(x=b,y=d,nilai_kiri);
end if;
else
if slope(y=solve(sl_kendala,y))=0 then
sl_min:=c;
sl_max:=d;
end if;
end if;
end if;
end if;

btx:=(a+b)/2;
bty:=(c+d)/2;
if type(kendala_geser,'<=')=true then
if type(lhs(kendala_geser),'constant')=true then
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
g:=evalf((abs(sl_max)+abs(sl_min))/2000*m+subs({x=btx,y=pty},rhs(kendala_geser))
,2);
    h:=rhs(kendala_geser):
    g<=h:
else
g:=evalf((abs(sl_max)+abs(sl_min))/2000*m+subs({x=btx,y=pty},lhs(kendala_geser))
,2):
    h:=lhs(kendala_geser):
    g>=h:
end if:
else
if type(lhs(kendala_geser),'constant')=true then

g:=evalf((abs(sl_max)+abs(sl_min))/2000*m+subs({x=btx,y=pty},rhs(kendala_geser))
,2):
    h:=rhs(kendala_geser):
    g<h:
else
g:=evalf((abs(sl_max)+abs(sl_min))/2000*m+subs({x=btx,y=pty},lhs(kendala_geser))
,2):
    h:=lhs(kendala_geser):
    g>h:
end if:
end if:
end proc;
geser_kendala := proc(kendala_geser, m, a, b, c, d)
local g, h, l, sl_kendala, sl_cek, sl_max, sl_min, nilai_kanan, nilai_kiri, btx, bty;
    sl_kendala := lhs(kendala_geser) = rhs(kendala_geser);
    sl_cek := subs(x = 0, sl_kendala);
    if type(lhs(sl_cek), 'constant') and type(rhs(sl_cek), 'constant') = true then sl_min := a; sl_max := b
    else
        if 0 < slope(y = solve(sl_kendala, y)) then
            if type(lhs(sl_kendala), 'constant') = true then
                nilai_kanan := rhs(sl_kendala);
                sl_min := subs(x = a, y = d, nilai_kanan);
                sl_max := subs(x = h, y = c, nilai_kanan)
            else
                nilai_kiri := lhs(sl_kendala);
                sl_min := subs(x = a, y = d, nilai_kiri);
                sl_max := subs(x = h, y = c, nilai_kiri)
            end if
        else
            if slope(y = solve(sl_kendala, y)) < 0 then
                if type(lhs(sl_kendala), 'constant') = true then
                    nilai_kanan := rhs(sl_kendala);
                    sl_min := subs(x = a, y = c, nilai_kanan);
                    sl_max := subs(x = b, y = d, nilai_kanan)
                else
                    nilai_kiri := lhs(sl_kendala);
                    sl_min := subs(x = a, y = c, nilai_kiri);
                    sl_max := subs(x = b, y = d, nilai_kiri)
                end if
            else if slope(y = solve(sl_kendala, y)) = 0 then sl_min := c; sl_max := d end if
            end if
        end if
    end if;
    btx := 1/2*a + 1/2*b;
    bty := 1/2*c + 1/2*d;
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
if type(kendala_geser, '<=')= 'true' then
  if type(lhs(kendala_geser), 'constant')= true then
    g := evalf(1 / 2000*(abs(sl_max) + abs(sl_min))*m + subs({ x = btx, y = bty }, rhs(kendala_geser)),
      2);
    h := rhs(kendala_geser);
    g ≤ h
  else
    g := cvalf(1 / 2000*(abs(sl_max) + abs(sl_min))*m + subs({ x = btx, y = bty }, lhs(kendala_geser)),
      2);
    h := lhs(kendala_geser);
    h ≤ g
  end if
else
  if type(lhs(kendala_geser), 'constant')= true then
    g := evalf(1 / 2000*(abs(sl_max) + abs(sl_min))*m + subs({ x = btx, y = bty }, rhs(kendala_geser)),
      2);
    h := rhs(kendala_geser);
    g < h
  else
    g := cvalf(1 / 2000*(abs(sl_max) + abs(sl_min))*m + subs({ x = btx, y = bty }, lhs(kendala_geser)),
      2);
    h := lhs(kendala_geser);
    h < g
  end if
end if
end proc
[ >
```

- Pergeseran dengan Button PLUS/MINUS

```
> geser_PLUS:=proc (kendala_geser,m)
  local g,h;

  if type(kendala_geser, '<=')= 'true' then
    if type(lhs(kendala_geser), 'constant')=true then
      g:=lhs(kendala_geser) + m;
      h:=rhs(kendala_geser);
      g<=h:
    else g:=rhs(kendala_geser) + m;
      h:=lhs(kendala_geser);
      g>=h:
    end if:
  else
    if type(lhs(kendala_geser), 'constant')=true then
      g:=lhs(kendala_geser) + m;
      h:=rhs(kendala_geser);
      g<h:
    else g:=rhs(kendala_geser) + m;
      h:=lhs(kendala_geser);
      g>h:
    end if:
  end if:
end proc;
geser_PLUS := proc(kendala_geser, m)
  local g, h;
  if type(kendala_geser, '<=')= 'true' then
    if type(lhs(kendala_geser), 'constant')= true then
      g := lhs(kendala_geser) + m; h := rhs(kendala_geser); g ≤ h
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
else g := rhs(kendala_geser) + m; h := lhs(kendala_geser); h ≤ g
end if
else
  if type(lhs(kendala_geser), 'constant') = true then
    g := lhs(kendala_geser) + m; h := rhs(kendala_geser); g < h
  else g := rhs(kendala_geser) + m; h := lhs(kendala_geser); h < g
  end if
end if
end proc
```

- Gambar Sistem Pertidaksamaan Linear

```
> Gambar_PTL:=proc(p, q, r, s, t, u, a, b, c, d)
  local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, gambar:

  Kendala:=():

  if p<>0 then
    Kendala:={p}:
    tp1:=textplot([koordplotteks(p, a, b, c, d), p],
  color=blue, align=RIGHT, font=[HELVETICA, BOLD, 10]);
  else
    tp1:=textplot([0, 0, ``], font=[HELVETICA, BOLD, 10]):
  end if:

  if q<>0 then
    Kendala:= Kendala union {q}:
    tp2:=textplot([koordplotteks(q, a, b, c, d), q],
  color=blue, align=RIGHT, font=[HELVETICA, BOLD, 10]);
  else
    tp2:=textplot([0, 0, ``], font=[HELVETICA, BOLD, 10]):
  end if:

  if r<>0 then
    Kendala:= Kendala union {r}:
    tp3:=textplot([koordplotteks(r, a, b, c, d), r],
  color=blue, align=RIGHT, font=[HELVETICA, BOLD, 10]);
  else
    tp3:=textplot([0, 0, ``], font=[HELVETICA, BOLD, 10]):
  end if:

  if s<>0 then
    Kendala:= Kendala union {s}:
    tp4:=textplot([koordplotteks(s, a, b, c, d), s],
  color=blue, align=RIGHT, font=[HELVETICA, BOLD, 10]);
  else
    tp4:=textplot([0, 0, ``], font=[HELVETICA, BOLD, 10]):
  end if:

  if t<>0 then
    Kendala:= Kendala union {t}:
    tp5:=textplot([koordplotteks(t, a, b, c, d), t],
  color=blue, align=RIGHT, font=[HELVETICA, BOLD, 10]);
  else
    tp5:=textplot([0, 0, ``], font=[HELVETICA, BOLD, 10]):
  end if:

  if u<>0 then
    Kendala:= Kendala union {u}:
    tp6:=textplot([koordplotteks(u, a, b, c, d), u],
  color=blue, align=RIGHT, font=[HELVETICA, BOLD, 10]);
  else
    tp6:=textplot([0, 0, ``], font=[HELVETICA, BOLD, 10]):
  end if:

  gambar:=inequal(Kendala, x=a..b, y=c..d, optionsfeasible=(color=yellow),
  optionsexcluded=(color=white), optionsclosed=(color=navy), thickness=2);
  display(gambar, tp1, tp2, tp3, tp4, tp5, tp6);
end proc;
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
Gambar_PTL := proc(p, q, r, s, t, u, a, b, c, d)
local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, gambar;
  Kendala := { };
  if p ≠ 0 then
    Kendala := { p };
    tp1 := textplot([koordplottcks(p, a, b, c, d), p], color = blue, align = RIGHT,
      font = [HELVETICA, BOLD, 10])
  else tp1 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
  end if;
  if q ≠ 0 then
    Kendala := Kendala union { q };
    tp2 := textplot([koordplottcks(q, a, b, c, d), q], color = blue, align = RIGHT,
      font = [HELVETICA, BOLD, 10])
  else tp2 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
  end if;
  if r ≠ 0 then
    Kendala := Kendala union { r };
    tp3 := textplot([koordplottcks(r, a, b, c, d), r], color = blue, align = RIGHT,
      font = [HELVETICA, BOLD, 10])
  else tp3 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
  end if;
  if s ≠ 0 then
    Kendala := Kendala union { s };
    tp4 := textplot([koordplottcks(s, a, b, c, d), s], color = blue, align = RIGHT,
      font = [HELVETICA, BOLD, 10])
  else tp4 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
  end if;
  if t ≠ 0 then
    Kendala := Kendala union { t };
    tp5 := textplot([koordplottcks(t, a, b, c, d), t], color = blue, align = RIGHT,
      font = [HELVETICA, BOLD, 10])
  else tp5 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
  end if;
  if u ≠ 0 then
    Kendala := Kendala union { u };
    tp6 := textplot([koordplottcks(u, a, b, c, d), u], color = blue, align = RIGHT,
      font = [HELVETICA, BOLD, 10])
  else tp6 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
  end if;
  gambar := incqual(Kendala, x = a .. b, y = c .. d, optionsfeasible = (color = yellow),
    optionsexcluded = (color = white), optionsclosed = (color = navy), thickness = 2);
  display(gambar, tp1, tp2, tp3, tp4, tp5, tp6)
end proc
```

— Text untuk nilai Slider

```
> TeksSlider:=proc(a,b,c,d,sasaran,slider)
local cek_sasaran,sl_min,sl_max,btx,bty,btz:
cek_sasaran:=solve(sasaran,y):
if (cek_sasaran=NULL) then
sl_min:=a;
sl_max:=b;
else
if slope(y=solve(sasaran,y))>0 then
sl_min:=subs(x=a,y=d,sasaran);
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
sl_max:=subs(x=b,y=c,sasaran);
else
  if slope(y=solve(sasaran,y))<0 then
    sl_min:=subs(x=a,y=c,sasaran);
    sl_max:=subs(x=b,y=d,sasaran);
  else
    if slope(y=solve(sasaran,y))=0 then
      sl_min:=c;
      sl_max:=d;
    end if;
  end if;
end if;
end if;
btx:=(a+b)/2;
bty:=(c+d)/2;
btz:=(abs(sl_max)+abs(sl_min))/2000*slider+subs({x=btx,y=bty},sasaran);
evalf(btz,3);
end proc;
```

- Text Nilai Sasaran untuk Plot Titik Uji

```
> TeksSasaran:=proc(xx,yy,sasaran)
  local btz;
  btz:=subs(x=xx,y=yy,sasaran);
  evalf(btz,3);
end proc;
```

- Plot Uji Titik Pojok

```
> Uji_ttk:=proc(Utx,Uty,p,q,r,s,t,u,a,b,c,d,sasaran,pttkpot,plabel)
  local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, tp, gambar, daerah_ujung,
  koord_pot, j2, i2, pers, koord, k2, cek, ncek, l2, tk, titik2, m2, potong, btz,
  objPlot, nilai_sasaran, nilaisasaran, anim;
  Kendala:={};
  if p<>0 then
    Kendala:={p};
    tp1:=textplot([koordplotteks(p,a,b,c,d),p],
    color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
  else
    tp1:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);
  end if;
  if q<>0 then
    Kendala:= Kendala union {q};
    tp2:=textplot([koordplotteks(q,a,b,c,d),q],
    color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
  else
    tp2:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);
  end if;
  if r<>0 then
    Kendala:= Kendala union {r};
    tp3:=textplot([koordplotteks(r,a,b,c,d),r],
    color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
  else
    tp3:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);
  end if;
  if s<>0 then
    Kendala:= Kendala union {s};
    tp4:=textplot([koordplotteks(s,a,b,c,d),s],
    color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
  else
    tp4:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);
  end if;
  if t<>0 then
    Kendala:= Kendala union {t};
    tp5:=textplot([koordplotteks(t,a,b,c,d),t],
    color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
  else
    tp5:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);
  end if;
  if u<>0 then
    Kendala:= Kendala union {u};
    tp6:=textplot([koordplotteks(u,a,b,c,d),u],
    color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
  end if;
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
else
  tp6:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:
if plabel=true then tp:={tp1} union {tp2} union {tp3} union {tp4} union {tp5}
union {tp6}:
else tp:={}:
end if:
daerah_ujung:={}:
koord_pot:=NULL:
for j2 from 1 to nops(Kendala) do
  for i2 from j2 to nops(Kendala)-1 do
    pers[i2]:=convert({Kendala[j2],Kendala[i2+1]},equality):
    koord:=solve(pers[i2]):
    if koord<>NULL then
      koord_pot:=koord_pot,koord:
    end if:
  end do:
end do:
koord_pot:={koord_pot}:
for k2 from 1 to nops(koord_pot) do
cek:=subs(koord_pot[k2],Kendala):
ncek:=0:
for l2 from 1 to nops(cek) do
if evalb(cek[l2])=true then ncek:=ncek+1:
end if:
end do:
if ncek=nops(cek) then
daerah_ujung:=daerah_ujung union {koord_pot[k2]}:
end if:
end do:
tk:={}:
titik2:=NULL:
for m2 from 1 to nops(daerah_ujung) do
if lhs(daerah_ujung[m2,1])=x then
titik2:=titik2,[rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2])]:
tk:=tk union
{textplot([rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2])],[rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2])],align={ABOVE,RIGHT})}:
else
titik2:=titik2,[rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1])]:
tk:=tk union
{textplot([rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1])],[rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1])],align={ABOVE,RIGHT})}:
end if:
end do:
if pttkpot=true then tk:=tk:
else tk:={}:
end if:
potong:=pointplot({titik2},symbol=CIRCLE,color=RED,symbolsize=15):
btz:=subs(x=Utx,y=Uty,sasaran):
objPlot:=solve(sasaran=btz,y):
nilai_sasaran:=evalf(btz,3):
nilaisasaran:=textplot([b-2,d-2,nilai_sasaran],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]):
anim:=plot(objPlot,x=a..b,y=c..d,color=red,thickness=2):
gambar:=inequal(Kendala,x=a..b,y=c..d,optionsfeasible=(color=yellow),
optionsexcluded=(color=white),optionsclosed=(color=navy),thickness=2):
display(gambar,tp,anim,potong,nilaisasaran,tk):
end proc:
```

- Text fungsi sasaran baru untuk Sensitifitas perubahan gradien

```
> tgradien:=proc(sasaran,grad,cekxy)
local mSb, Sb, nmSb, cx, sasaran_baru:
if grad="tidak terdefinisi" then Sb:=coeff(sasaran,x)*x:
else if grad=0 then Sb:=coeff(sasaran,y)*y:
else
if cekxy=true then
nmSb:=grad/P/coeff(sasaran,y):
cx:=solve(nmSb,P):
Sb:=-1*cx*x+coeff(sasaran,y)*y:
else
nmSb:=grad/coeff(sasaran,x)/P:
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
cx:=solve(nmSb,P);
Sb:=coeff(sasaran,x)*x-cx*y;
end if;
end if;
end if;
end proc;
```

- Text nilai fungsi sasaran baru untuk Sensitifitas perubahan gradien

```
> ntgradien:=proc(min_max,Pot_min,Pot_max,Sb)
local Kendala, PotK, nSb, nilai_sasaran;

if min_max=true then
PotK:=Pot_max;
else PotK:=Pot_min;
end if;
nSb:=subs(PotK,Sb);
nilai_sasaran:=evalf(nSb,4);

end proc;
```

- Text nilai gradien fungsi sasaran baru untuk Sensitifitas perubahan gradien

```
> mntgradien:=proc(min_max,Pot_min,Pot_max,p,q,r,s,t,u,sasaran,NTa)

local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, koord_min_max, gambar, NK1,
NK2, Ua1, Ua2, Ua3, Ub1, Ub2, Ub3, Ua4, Ua5, Ub4, Ub5, ma1, mb1, ma2, mb2, Us1,
Us2, Us3, PotK, ma, mb, mSb, mn, pers_pot, pers_kendala, cek_pot, jj;
Kendala:={};

if p<>0 then
Kendala:={p};
end if;

if q<>0 then
Kendala:= Kendala union {q};
end if;

if r<>0 then
Kendala:= Kendala union {r};
end if;

if s<>0 then
Kendala:= Kendala union {s};
end if;

if t<>0 then
Kendala:= Kendala union {t};
end if;

if u<>0 then
Kendala:= Kendala union {u};
end if;

if min_max=true then
koord_min_max:=Pot_max;
else koord_min_max:=Pot_min;
end if;
pers_pot:={};
pers_kendala:=convert(Kendala,equality);
for jj from 1 to nops(pers_kendala) do
cek_pot:=evalb(subs(koord_min_max,pers_kendala[jj]));
if cek_pot=true then
pers_pot:=pers_pot union {pers_kendala[jj]};
end if;
end do;

Ua1:=pers_pot[1];
Ua2:=solve(Ua1,y);
if Ua2<>NULL then Ua3:=coeff(Ua2,x);
else Ua3:=1000;
end if;
```


PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
Ub1:=pers_pot[2];
Ub2:=solve(Ub1,y);
if Ub2<>NULL then Ub3:=coeff(Ub2,x);
else Ub3:=1000;
end if;

Ua4:=evalf(arctan(Ua3));
if Ua4<0 then Ua5:= evalf(Pi+Ua4)
else Ua5:=Ua4;
end if;
Ub4:=evalf(arctan(Ub3));
if Ub4<0 then Ub5:= evalf(Pi+Ub4)
else Ub5:=Ub4;
end if;

if Ua5>Ub5 then
ma1:=Ub5;
mb1:=Ua5;
ma2:=evalf(Ua5-Pi);
mb2:=Ub5;
else
ma1:=Ua5;
mb1:=Ub5;
ma2:=evalf(Ub5-Pi);
mb2:=Ua5;
end if;
Us1:=-1*coeff(sasaran,x)/coeff(sasaran,y);
Us2:=evalf(arctan(Us1));
if Us2<0 then Us3:= evalf(Pi+Us2);
else Us3:= Us2;
end if;
PotK:=solve({Ua1,Ub1});
if (evalb(ma1<Us3) and evalb(mb1>Us3))=true then
ma:=ma1;
mb:=mb1;
else
ma:=ma2;
mb:=mb2;
end if;

mSb:=ma+(abs(mb-ma)/1000)*NTa;
if mSb=1.57 then mn:="tidak terdefinisi";
else
mn:=evalf(tan(mSb),3);
end if;

end proc;
```

- Plot fungsi sasaran baru untuk Sensitifitas perubahan gradien

```
> gradien:=proc(p,q,r,s,t,u,a,b,c,d,nSb,Sb,pttkpot,plabel)

local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, tp, gambar, daerah_ujung,
koord_pot, j2, i2, pers, koord, k2, cek, ncek, l2, tk, titik2, m2, potong,
cek_sasaran, sl_min, sl_max, btx, bty, btz, objPlot, nilai_sasaran,
nilaisasaran, anim;

Kendala:={}:
if p<>0 then
Kendala:={p}:
tp1:=textplot([koordplotteks(p,a,b,c,d),p],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
tp1:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if;

if q<>0 then
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
Kendala:= Kendala union {q}:
tp2:=textplot([koordplotteks(q,a,b,c,d),q],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
tp2:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

if r<>0 then
Kendala:= Kendala union {r}:
tp3:=textplot([koordplotteks(r,a,b,c,d),r],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
tp3:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

if s<>0 then
Kendala:= Kendala union {s}:
tp4:=textplot([koordplotteks(s,a,b,c,d),s],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
tp4:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

if t<>0 then
Kendala:= Kendala union {t}:
tp5:=textplot([koordplotteks(t,a,b,c,d),t],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
tp5:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

if u<>0 then
Kendala:= Kendala union {u}:
tp6:=textplot([koordplotteks(u,a,b,c,d),u],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
tp6:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:
if plabel=true then tp:={tp1} union {tp2} union {tp3} union {tp4} union {tp5}
union {tp6}:
else tp:={}:
end if:
daerah_ujung:={}:
koord_pot:=NULL:
for j2 from 1 to nops(Kendala) do
for i2 from j2 to nops(Kendala)-1 do
pers[i2]:=convert({Kendala[j2],Kendala[i2+1]},equality):
koord:=solve(pers[i2]):
if koord<>NULL then
koord_pot:=koord_pot,koord:
end if:
end do:
koord_pot:={koord_pot}:
for k2 from 1 to nops(koord_pot) do
cek:=subs(koord_pot[k2],Kendala):
ncek:=0:
for l2 from 1 to nops(cek) do
if evalb(cek[l2])=true then ncek:=ncek+1:
end if:
end do:
if ncek=nops(cek) then
daerah_ujung:=daerah_ujung union {koord_pot[k2]}:
end if:
end do:
tk:={}:
titik2:=NULL:
for m2 from 1 to nops(daerah_ujung) do
if lhs(daerah_ujung[m2,1])=x then
titik2:=titik2,[rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2])]:
tk:=tk union
{textplot([rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2]),[rhs(daerah_ujung[m2,
1]),rhs(daerah_ujung[m2,2])]],align={ABOVE,RIGHT})};
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
else
titik2:=titik2,[rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1])]:
tk:=tk union
{textplot([rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1]),[rhs(daerah_ujung[m2,
2]),rhs(daerah_ujung[m2,1])]],align={ABOVE,RIGHT})}:
end if:
end do:
if pttkpot=true then tk:=tk:
else tk:={}:
end if:
potong:=pointplot({titik2},symbol=CIRCLE,color=RED,symbolsize=15):

nilai_sasaran:=evalf(nSb,3):
nilaisasaran:=textplot([b-3,d-3,{nilai_sasaran}],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]):
objPlot:=solve(Sb=nSb,y):
anim:=plot(objPlot,x=a..b,y=c..d,color=blue,thickness=2):
gambar:=inequal(Kendala,x=a..b,y=c..d,optionsfeasible=(color=yellow),
optionsexcluded=(color=white),optionsclosed=(color=navy),thickness=2):
display(gambar,tp,potong,anim,nilaisasaran,tk):
end proc:
```

- Perpindahan Nilai

```
> pindah:=proc(a)
local b:
b:=a:
end proc:
```

pindah := proc(a) local b; b := a end proc

- PLUS

```
> tambah:=proc(a,b)
local c:
c:=a+b:
end proc:
```

tambah := proc(a, b) local c; c := a + b end proc

- Minus

```
> kurang:=proc(a,b)
local c:
c:=a-b:
end proc:
```

kurang := proc(a, b) local c; c := a - b end proc

- Gradien

```
> Grad:=proc(a)
local m;
if coeff(a,y)=0 then m:="tidak terdefinisi";
else m:=-coeff(a,x)/coeff(a,y);
end if;
end proc:
```

Grad :=

proc(a) local m; if coeff(a, y) = 0 then m := "tidak terdefinisi" else m := -coeff(a, x) / coeff(a, y) end if end proc

- Peringatan

```
> peringatan:=proc(xy,a,b,c,d)
local k;
if xy=true then
if ((b subset d) or (d subset b))=true then k:="";
else k:="Titik optimum sudah bergeser";
end if;
else
if ((a subset c) or (c subset a))=true then k:="";
else k:="Titik optimum sudah bergeser";
end if;
end if;
end proc:
```

peringatan := proc(xy, a, b, c, d)

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
local k;
```

```
if xy = true then if (b subset d or d subset b) = true then k := "" else k := "Titik optimum sudah bergeser" end if  
else if (a subset c or c subset a) = true then k := "" else k := "Titik optimum sudah bergeser" end if  
end if
```

```
end proc
```

- Penggambaran Grafik

Prosedur ini digunakan untuk menginisialisasi masukkan kemudian menggambar grafiknya dalam maplot.

```
> Gambar_Grafik:=proc(p,q,r,s,t,u,a,b,c,d,sasaran,slider,pttkpot,plabel)  
local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, tp, gambar, daerah_ujung,  
koord_pot, j2, i2, pers, koord, k2, cek, ncek, l2, tk, titik2, m2, potong,  
cek_sasaran, sl_min, sl_max, btx, bty, btz, objPlot, nilai_sasaran,  
nilaisasaran, anim;  
  
Kendala:={};  
  
if p<>0 then  
  Kendala:={p};  
  tp1:=textplot([koordplotteks(p,a,b,c,d),p],  
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);  
else  
  tp1:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);  
end if;  
  
if q<>0 then  
  Kendala:= Kendala union {q};  
  tp2:=textplot([koordplotteks(q,a,b,c,d),q],  
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);  
else  
  tp2:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);  
end if;  
  
if r<>0 then  
  Kendala:= Kendala union {r};  
  tp3:=textplot([koordplotteks(r,a,b,c,d),r],  
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);  
else  
  tp3:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);  
end if;  
  
if s<>0 then  
  Kendala:= Kendala union {s};  
  tp4:=textplot([koordplotteks(s,a,b,c,d),s],  
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);  
else  
  tp4:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);  
end if;  
  
if t<>0 then  
  Kendala:= Kendala union {t};  
  tp5:=textplot([koordplotteks(t,a,b,c,d),t],  
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);  
else  
  tp5:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);  
end if;  
  
if u<>0 then  
  Kendala:= Kendala union {u};  
  tp6:=textplot([koordplotteks(u,a,b,c,d),u],  
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);  
else  
  tp6:=textplot([0,0,``],font=[HELVETICA,BOLD,10]);  
end if;  
if plabel=true then tp:={tp1} union {tp2} union {tp3} union {tp4} union {tp5}  
union {tp6};  
else tp:={};  
end if;  
daerah_ujung:={};  
koord_pot:=NULL;  
for j2 from 1 to nops(Kendala) do
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
for i2 from j2 to nops(Kendala)-1 do
  pers[i2]:=convert({Kendala[j2],Kendala[i2+1]},equality):
  koord:=solve(pers[i2]):
  if koord<>NULL then
    koord_pot:=koord_pot,koord:
  end if:
end do:
end do:
koord_pot:={koord_pot}:
for k2 from 1 to nops(koord_pot) do
cek:=subs(koord_pot[k2],Kendala):
ncek:=0:
for l2 from 1 to nops(cek) do
if evalb(cek[l2])=true then ncek:=ncek+1:
end if:
end do:
if ncek=nops(cek) then
daerah_ujung:=daerah_ujung union {koord_pot[k2]}:
end if:
end do:
tk:={}:
titik2:=NULL:
for m2 from 1 to nops(daerah_ujung) do
if lhs(daerah_ujung[m2,1])=x then
titik2:=titik2,[rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2])]:
tk:=tk union
{textplot({rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2])},{rhs(daerah_ujung[m2,1]),rhs(daerah_ujung[m2,2])}],align={ABOVE,RIGHT})};
else
titik2:=titik2,[rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1])]:
tk:=tk union
{textplot({rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1])},{rhs(daerah_ujung[m2,2]),rhs(daerah_ujung[m2,1])}],align={ABOVE,RIGHT})};
end if:
end do:
if pttkpot=true then tk:=tk:
else tk:={}:
end if:
potong:=pointplot({titik2},symbol=CIRCLE,color=RED,symbolsize=15):

cek_sasaran:=solve(sasaran,y):
if (cek_sasaran=NULL) then
sl_min:=a:
sl_max:=b:
else
if slope(y=solve(sasaran,y))>0 then
sl_min:=subs(x=a,y=d,sasaran):
sl_max:=subs(x=b,y=c,sasaran):
else
if slope(y=solve(sasaran,y))<0 then
sl_min:=subs(x=a,y=c,sasaran):
sl_max:=subs(x=b,y=d,sasaran):
else
if slope(y=solve(sasaran,y))=0 then
sl_min:=c:
sl_max:=d:
end if:
end if:
end if:
end if:
btx:=(a+b)/2:
bty:=(c+d)/2:
btz:=(abs(sl_max)+abs(sl_min))/2000*slider+subs({x=btx,y=bty},sasaran):
objPlot:=solve(sasaran=btz,y):
nilai_sasaran:=evalf(btz,3):
nilaisasaran:=textplot([b-2,d-2,nilai_sasaran],
color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]):

anim:=plot(objPlot,x=a..b,y=c..d,color=red.thickness=2):
gambar:=inequal(Kendala,x=a..b,y=c..d,optionsfeasible=(color=yellow),
optionsexcluded=(color=white),optionsclosed=(color=navy),thickness=2):
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
display (gambar, tp, anim, potong, nilaisasaran, tk);  
end proc;
```

>

```
Gambar_Grafik := proc(p, q, r, s, t, u, a, b, c, d, sasaran, slider, putpot, plabel)
```

```
local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, tp, gambar, daerah_ujung, koord_pot, j2, i2, pers, koord, k2, cek,  
ncek, l2, tk, titik2, m2, potong, cek_sasaran, sl_min, sl_max, btx, bty, btz, objPlot, nilai_sasaran, nilaisasaran, anim;
```

```
  Kendala := { };
```

```
  if p ≠ 0 then
```

```
    Kendala := { p };
```

```
    tp1 := textplot([koordplottcks(p, a, b, c, d), p], color = blue, align = RIGHT,
```

```
      font = [HELVETICA, BOLD, 10]);
```

```
  else tp1 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
```

```
  end if;
```

```
  if q ≠ 0 then
```

```
    Kendala := Kendala union { q };
```

```
    tp2 := textplot([koordplottcks(q, a, b, c, d), q], color = blue, align = RIGHT,
```

```
      font = [HELVETICA, BOLD, 10])
```

```
  else tp2 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
```

```
  end if;
```

```
  if r ≠ 0 then
```

```
    Kendala := Kendala union { r };
```

```
    tp3 := textplot([koordplottcks(r, a, b, c, d), r], color = blue, align = RIGHT,
```

```
      font = [HELVETICA, BOLD, 10])
```

```
  else tp3 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
```

```
  end if;
```

```
  if s ≠ 0 then
```

```
    Kendala := Kendala union { s };
```

```
    tp4 := textplot([koordplottcks(s, a, b, c, d), s], color = blue, align = RIGHT,
```

```
      font = [HELVETICA, BOLD, 10])
```

```
  else tp4 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
```

```
  end if;
```

```
  if t ≠ 0 then
```

```
    Kendala := Kendala union { t };
```

```
    tp5 := textplot([koordplottcks(t, a, b, c, d), t], color = blue, align = RIGHT,
```

```
      font = [HELVETICA, BOLD, 10])
```

```
  else tp5 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
```

```
  end if;
```

```
  if u ≠ 0 then
```

```
    Kendala := Kendala union { u };
```

```
    tp6 := textplot([koordplottcks(u, a, b, c, d), u], color = blue, align = RIGHT,
```

```
      font = [HELVETICA, BOLD, 10])
```

```
  else tp6 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
```

```
  end if;
```

```
  if plabel = true then tp := (((({tp1} union {tp2}) union {tp3}) union {tp4}) union {tp5}) union {tp6}
```

```
  else tp := { }
```

```
  end if;
```

```
  daerah_ujung := { };
```

```
  koord_pot := NULL;
```

```
  for j2 to nops(Kendala) do for i2 from j2 to nops(Kendala) - 1 do
```

```
    pers[i2] := convert({Kendala[j2], Kendala[i2 + 1]}, equality);
```

```
    koord := solve(pers[i2]);
```

```
    if koord ≠ NULL then koord_pot := koord_pot, koord end if
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```

end do
end do:
koord_pot := { koord_pot };
for k2 to nops(koord_pot) do
cek := subs(koord_pot[k2], Kendala);
ncek := 0;
for l2 to nops(cek) do if evalb(cek[l2]) = true then ncek := ncek + 1 end if end do;
if ncek = nops(cek) then daerah_ujung := daerah_ujung union { koord_pot[k2] } end if
end do:
tk := { };
titik2 := NULL;
for m2 to nops(daerah_ujung) do
if lhs(daerah_ujung[m2, 1]) = x then
titik2 := titik2, [rhs(daerah_ujung[m2, 1]), rhs(daerah_ujung[m2, 2])];
tk := tk union { textplot([rhs(daerah_ujung[m2, 1]), rhs(daerah_ujung[m2, 2])],
[rhs(daerah_ujung[m2, 1]), rhs(daerah_ujung[m2, 2])], align = { ABOVE, RIGHT } ) }
else
titik2 := titik2, [rhs(daerah_ujung[m2, 2]), rhs(daerah_ujung[m2, 1])];
tk := tk union { textplot([rhs(daerah_ujung[m2, 2]), rhs(daerah_ujung[m2, 1])],
[rhs(daerah_ujung[m2, 2]), rhs(daerah_ujung[m2, 1])], align = { ABOVE, RIGHT } ) }
end if
end do:
if ptkpot = true then tk := tk else tk := { } end if;
potong := pointplot( { titik2 }, symbol = CIRCLE, color = RED, symbolsize = 15);
cek_sasaran := solve(sasaran, y);
if cek_sasaran = NULL then sl_min := a; sl_max := b
else
if 0 < slope(y = solve(sasaran, y)) then
sl_min := subs(x = a, y = d, sasaran); sl_max := subs(x = b, y = c, sasaran)
else
if slope(y = solve(sasaran, y)) < 0 then
sl_min := subs(x = a, y = c, sasaran); sl_max := subs(x = b, y = d, sasaran)
else if slope(y = solve(sasaran, y)) = 0 then sl_min := c; sl_max := d end if
end if
end if
end if;
btx := 1 / 2 * a + 1 / 2 * b;
bty := 1 / 2 * c + 1 / 2 * d;
btz := 1 / 2000 * (abs(sl_max) + abs(sl_min)) * slider + subs( { x = btx, y = bty }, sasaran );
objPlot := solve(sasaran = btz, y);
nilai_sasaran := evalf(btz, 3);
nilaisasaran :=
textplot([b - 2, d - 2, nilai_sasaran], color = blue, align = RIGHT, font = [HELVETICA, BOLD, 10]);
anim := plot(objPlot, x = a .. b, y = c .. d, color = red, thickness = 2);
gambar := inequal(Kendala, x = a .. b, y = c .. d, optionsfeasible = (color = yellow),
optionsexcluded = (color = white), optionsclosed = (color = navy), thickness = 2);
display(gambar, tp, anim, potong, nilaisasaran, tk)
end proc

```

plotdititik (sasaran)

```

> plotdititik:=proc(p,q,r,s,t,u,a,b,c,d,sasaran,teknilai,pttkpot,plabel)
local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, tp, gambar, daerah_ujung,

```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
koord_pot, j2, i2, pers, koord, k2, cek, ncek, l2, tk, titik2, m2, potong,
objPlot, nilaisasaran, anim;
```

```
Kendala:={}:

```

```
if p<>0 then
  Kendala:={p}:
  tp1:=textplot([koordplotteks(p,a,b,c,d),p],
  color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
  tp1:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

```

```
if q<>0 then
  Kendala:= Kendala union {q}:
  tp2:=textplot([koordplotteks(q,a,b,c,d),q],
  color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
  tp2:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

```

```
if r<>0 then
  Kendala:= Kendala union {r}:
  tp3:=textplot([koordplotteks(r,a,b,c,d),r],
  color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
  tp3:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

```

```
if s<>0 then
  Kendala:= Kendala union {s}:
  tp4:=textplot([koordplotteks(s,a,b,c,d),s],
  color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
  tp4:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

```

```
if t<>0 then
  Kendala:= Kendala union {t}:
  tp5:=textplot([koordplotteks(t,a,b,c,d),t],
  color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
  tp5:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

```

```
if u<>0 then
  Kendala:= Kendala union {u}:
  tp6:=textplot([koordplotteks(u,a,b,c,d),u],
  color=blue,align=RIGHT,font=[HELVETICA,BOLD,10]);
else
  tp6:=textplot([0,0,``],font=[HELVETICA,BOLD,10]):
end if:

```

```
if plabel=true then tp:={tp1} union {tp2} union {tp3} union {tp4} union {tp5}
union {tp6}:
else tp:={}:
end if:

```

```
daerah_ujung:={}:

```

```
koord_pot:=NULL:

```

```
for j2 from 1 to nops(Kendala) do
  for i2 from j2 to nops(Kendala)-1 do
    pers[i2]:=convert({Kendala[j2],Kendala[i2+1]},equality):
    koord:=solve(pers[i2]):
    if koord<>NULL then
      koord_pot:=koord_pot,koord:
    end if:
  end do:
end do:

```

```
koord_pot:={koord_pot}:

```

```
for k2 from 1 to nops(koord_pot) do
  cek:=subs(koord_pot[k2],Kendala):
  ncek:=0:
  for l2 from 1 to nops(cek) do

```


PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
if evalb(cek[l2])=true then ncek:=ncek+1:
end if:
end do:
if ncek=nops(cek) then
daerah_ujung:=daerah_ujung union {koord_pot[k2]}:
end if:
end do:
tk:={}:
titik2:=NULL:
for m2 from 1 to nops(daerah_ujung) do
if lhs(daerah_ujung[m2,1])=x then
titik2:=titik2, [rhs(daerah_ujung[m2,1]), rhs(daerah_ujung[m2,2])]:
tk:=tk union
{textplot([rhs(daerah_ujung[m2,1]), rhs(daerah_ujung[m2,2]), [rhs(daerah_ujung[m2,
1]), rhs(daerah_ujung[m2,2])]]], align={ABOVE, RIGHT})}:
else
titik2:=titik2, [rhs(daerah_ujung[m2,2]), rhs(daerah_ujung[m2,1])]:
tk:=tk union
{textplot([rhs(daerah_ujung[m2,2]), rhs(daerah_ujung[m2,1]), [rhs(daerah_ujung[m2,
2]), rhs(daerah_ujung[m2,1])]]], align={ABOVE, RIGHT})}:
end if:
end do:
if pttkpot=true then tk:=tk:
else tk:={}:
end if:
potong:=pointplot({titik2}, symbol=CIRCLE, color=RED, symbolsize=15):

objPlot:=solve(sasaran=teknilai, y);
nilaisasaran:=textplot([b-2, d-2, teknilai],
color=blue, align=RIGHT, font=[HELVETICA, BOLD, 10]):

anim:=plot(objPlot, x=a..b, y=c..d, color=red, thickness=2);
gambar:=inequal(Kendala, x=a..b, y=c..d, optionsfeasible=(color=yellow),
optionsexcluded=(color=white), optionsclosed=(color=navy), thickness=2);
display(gambar, tp, anim, potong, nilaisasaran, tk);
end proc;

plotdittitik := proc(p, q, r, s, t, u, a, b, c, d, sasaran, teknilai, pttkpot, plabel)
local Kendala, total, tp1, tp2, tp3, tp4, tp5, tp6, tp, gambar, daerah_ujung, koord_pot, j2, i2, pers, koord, k2, cek,
ncek, l2, tk, titik2, m2, potong, objPlot, nilaisasaran, anim;
Kendala := { };
if p ≠ 0 then
Kendala := {p};
tp1 := textplot([koordplotteks(p, a, b, c, d), p], color = blue, align = RIGHT,
font = [HELVETICA, BOLD, 10])
else tp1 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
end if;
if q ≠ 0 then
Kendala := Kendala union {q};
tp2 := textplot([koordplotteks(q, a, b, c, d), q], color = blue, align = RIGHT,
font = [HELVETICA, BOLD, 10])
else tp2 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
end if;
if r ≠ 0 then
Kendala := Kendala union {r};
tp3 := textplot([koordplotteks(r, a, b, c, d), r], color = blue, align = RIGHT,
font = [HELVETICA, BOLD, 10])
else tp3 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
end if;
if s ≠ 0 then
Kendala := Kendala union {s};
tp4 := textplot([koordplotteks(s, a, b, c, d), s], color = blue, align = RIGHT,
font = [HELVETICA, BOLD, 10])
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```

else tp4 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
end if;
if t ≠ 0 then
    Kendala := Kendala union {t};
    tp5 := textplot([koordplotteks(t, a, b, c, d), t], color = blue, align = RIGHT,
        font = [HELVETICA, BOLD, 10])
else tp5 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
end if;
if u ≠ 0 then
    Kendala := Kendala union {u};
    tp6 := textplot([koordplotteks(u, a, b, c, d), u], color = blue, align = RIGHT,
        font = [HELVETICA, BOLD, 10])
else tp6 := textplot([0, 0, ``], font = [HELVETICA, BOLD, 10])
end if;
if plabel = true then tp := (((({tp1} union {tp2}) union {tp3}) union {tp4}) union {tp5}) union {tp6}
else tp := { }
end if;
daerah_ujung := { };
koord_pot := NULL;
for j2 to nops(Kendala) do for i2 from j2 to nops(Kendala) - 1 do
    pers[i2] := convert({Kendala[j2], Kendala[i2 + 1]}, equality);
    koord := solve(pers[i2]);
    if koord ≠ NULL then koord_pot := koord_pot, koord end if
end do
end do;
koord_pot := { koord_pot };
for k2 to nops(koord_pot) do
    cek := subs(koord_pot[k2], Kendala);
    ncek := 0;
    for l2 to nops(cek) do if eval(cek[l2]) = true then ncek := ncek + 1 end if end do;
    if ncek = nops(cek) then daerah_ujung := daerah_ujung union { koord_pot[k2] } end if
end do;
tk := { };
titik2 := NULL;
for m2 to nops(daerah_ujung) do
    if lhs(daerah_ujung[m2, 1]) = x then
        titik2 := titik2, [rhs(daerah_ujung[m2, 1]), rhs(daerah_ujung[m2, 2])];
        tk := tk union { textplot([rhs(daerah_ujung[m2, 1]), rhs(daerah_ujung[m2, 2]),
            [rhs(daerah_ujung[m2, 1]), rhs(daerah_ujung[m2, 2])], align = {ABOVE, RIGHT})}
    else
        titik2 := titik2, [rhs(daerah_ujung[m2, 2]), rhs(daerah_ujung[m2, 1])];
        tk := tk union { textplot([rhs(daerah_ujung[m2, 2]), rhs(daerah_ujung[m2, 1]),
            [rhs(daerah_ujung[m2, 2]), rhs(daerah_ujung[m2, 1])], align = {ABOVE, RIGHT})}
    end if
end do;
if pitkpot = true then tk := tk else tk := { } end if;
potong := pointplot({ titik2 }, symbol = CIRCLE, color = RED, symbolsize = 15);
objPlot := solve(sasaran = teknilai, y);
nilaisasaran :=
    textplot([b - 2, d - 2, teknilai], color = blue, align = RIGHT, font = [HELVETICA, BOLD, 10]);
anim := plot(objPlot, x = a .. b, y = c .. d, color = red, thickness = 2);

```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
gambar := inequal(Kendala, x = a .. b, y = c .. d, optionsfeasible = (color = yellow),  
optionsexcluded = (color = white), optionsclosed = (color = navy), thickness = 2);  
display(gambar, tp, anim, potong, nilaisasaran, tk)  
end proc
```

- Prosedur Perhitungan

- Maximum

```
> Maksimum:=proc(a,b,c,d,e,f,sasaran)  
local Kendala,max:  
Kendala:={}:  
if a<> 0 then  
Kendala:={a}:  
end if:  
if b<> 0 then  
Kendala:=Kendala union {b}:  
end if:  
if c<> 0 then  
Kendala:=Kendala union {c}:  
end if:  
if d<> 0 then  
Kendala:=Kendala union {d}:  
end if:  
if e<> 0 then  
Kendala:=Kendala union {e}:  
end if:  
if f<> 0 then  
Kendala:=Kendala union {f}:  
end if:  
max:=maximize(sasaran,Kendala):  
evalf(subs(max,sasaran),4):  
end proc:  
> MaksimumKoord:=proc(a,b,c,d,e,f,sasaran)  
local Kendala,max,daerah_ujung, koord_pot, j2, i2, pers, koord, k2, cek, ncek,  
l2,koord_kritis,koord_k,mm,m2,ss,koordk:  
Kendala:={}:  
if a<> 0 then  
Kendala:={a}:  
end if:  
if b<> 0 then  
Kendala:=Kendala union {b}:  
end if:  
if c<> 0 then  
Kendala:=Kendala union {c}:  
end if:  
if d<> 0 then  
Kendala:=Kendala union {d}:  
end if:  
if e<> 0 then  
Kendala:=Kendala union {e}:  
end if:  
if f<> 0 then  
Kendala:=Kendala union {f}:  
end if:  
  
daerah_ujung:={}:  
koord_pot:=NULL:  
for j2 from 1 to nops(Kendala) do  
for i2 from j2 to nops(Kendala)-1 do  
pers[i2]:=convert({Kendala[j2],Kendala[i2+1]},equality):  
koord:=solve(pers[i2]):  
if koord<>NULL then  
koord_pot:=koord_pot,koord:
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
end if:
end do:
end do:
koord_pot:={koord_pot}:
for k2 from 1 to nops(koord_pot) do
cek:=subs(koord_pot[k2],Kendala):
ncek:=0:
for l2 from 1 to nops(cek) do
if evalb(cek[l2])=true then ncek:=ncek+1:
end if:
end do:
if ncek=nops(cek) then
daerah_ujung:=daerah_ujung union {koord_pot[k2]}:
end if:
end do:
max:=maximize(sasaran,Kendala):
koord_kritis:=NULL:
koord_k: {};
ss:=sasaran=subs(max,sasaran):
for m2 from 1 to nops(daerah_ujung) do
mm:=evalb(subs(daerah_ujung[m2],ss)):
if mm=true then
koord_k:=koord_kritis,daerah_ujung[m2]:
koord_kritis:=koord_k:
end if:
end do:
koordk:=koord_kritis:
end proc:
>
```

- Minimum

```
> Minimum:=proc(a,b,c,d,e,f,sasaran)
local Kendala,min:
Kendala:={}:
if a<> 0 then
Kendala:={a}:
end if:
if b<> 0 then
Kendala:=Kendala union {b}:
end if:
if c<> 0 then
Kendala:=Kendala union {c}:
end if:
if d<> 0 then
Kendala:=Kendala union {d}:
end if:
if e<> 0 then
Kendala:=Kendala union {e}:
end if:
if f<> 0 then
Kendala:=Kendala union {f}:
end if:
min:=minimize(sasaran,Kendala):
evalf(subs(min,sasaran),4)
end proc:
> MinimumKoord:=proc(a,b,c,d,e,f,sasaran)
local Kendala,min,daerah_ujung, koord_pot, j2, i2, pers, koord, k2, cek, ncek,
l2,koord_kritis,koord_k,mm,m2,ss,koordk:
Kendala:={}:
if a<> 0 then
Kendala:={a}:
end if:
if b<> 0 then
Kendala:=Kendala union {b}:
end if:
if c<> 0 then
Kendala:=Kendala union {c}:
end if:
if d<> 0 then
Kendala:=Kendala union {d}:
end if:
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
if e<> 0 then
Kendala:=Kendala union {e}:
end if:
if f<> 0 then
Kendala:=Kendala union {f}:
end if:

daerah_ujung:={}:
koord_pot:=NULL:
for j2 from 1 to nops(Kendala) do
  for i2 from j2 to nops(Kendala)-1 do
    pers{i2}:=convert({Kendala[j2],Kendala[i2+1]},equality):
    koord:=solve(pers[i2]):
    if koord<>NULL then
      koord_pot:=koord_pot,koord:
    end if:
  end do:
end do:
koord_pot:={koord_pot}:
for k2 from 1 to nops(koord_pot) do
cek:=subs(koord_pot[k2],Kendala):
ncek:=0:
for l2 from 1 to nops(cek) do
if evalb(cek[l2])=true then ncek:=ncek+1:
end if:
end do:
if ncek=nops(cek) then
daerah_ujung:=daerah_ujung union {koord_pot[k2]}:
end if:
end do:
min:= minimize(sasaran,Kendala):
koord_kritis:=NULL:
koord_k:={}:
ss:= subs(min,sasaran)=sasaran:
for m2 from 1 to nops(daerah_ujung) do
mm:=evalb(subs(daerah_ujung[m2],ss)):
if mm=true then
koord_k:=koord_kritis,daerah_ujung[m2]:
koord_kritis:=koord_k:
end if:
end do:
koordk:=koord_kritis:
end proc:
```

- Maplets

```
> coba:=proc(AA)
local maplet:
maplet := Maplets[Display](Maplet('onstartup'=AA,
Window['W1']('title'='Menu_Pembuka','width'=1024,'height'=768,'resizable'=false,'la
yout' = 'BL1','menubar'='MB1'),
Window['W2']('title'='Pertidaksamaan_Linear','width'=1024,'height'=768,'resizable'=
false,'layout' = 'BL2','menubar'='MB2'),
Window['W3']('title'='Sistem_Pertidaksamaan_Linear','width'=1024,'height'=768,'resi
zable'=false,'layout' = 'BL3','menubar'='MB3'),
Window['W4']('title'='Program_Linear','width'=1024,'height'=768,'resizable'=false,'
layout' = 'BL4','menubar'='MB4'),
Window['W5']('title'='Sensititas','width'=1024,'height'=768,'resizable'=false,'layo
ut' = 'BL5','menubar'='MB5'),
Window['H1']('title'='Bantuan_Pertidaksamaan_Linear','width'=800,'height'=200,'resi
zable'=false,'layout' = 'BLH1'),
Window['H2']('title'='Bantuan_Sistem_Pertidaksamaan_Linear','width'=800,'height'=20
0,'resizable'=false,'layout' = 'BLH2'),
Window['H3']('title'='Bantuan_Plot_Daerah_Layak','width'=800,'height'=400,'resizabl
e'=false,'layout' = 'BLH3'),
Window['H4']('title'='Bantuan_Pergeseran_Pertidaksamaan','width'=800,'height'=300,'
resizable'=false,'layout' = 'BLH4'),
Window['H5']('title'='Bantuan_Batas_Daerah_Plot','width'=800,'height'=200,'resizabl
e'=false,'layout' = 'BLH5'),
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
Window['H6']('title'='Bantuan_Titik_Uji','width'=800,'height'=200,'resizable'=false,'layout' = 'BLH6'),
```

```
BoxLayout['BL1'](BoxColumn(Label('font' = Font("Arial Black",18),"Beberapa Hal yang Perlu Diperhatikan dalam Memanfaatkan Program ini:"),Label('font' = Font("courier",14),"1. Pertidaksamaan ditulis dengan aturan sebagai berikut:"),Label('font' = Font("courier",14)," a. Semua suku yang memuat variabel dituliskan di ruas kiri, dan suku yang tidak memuat variabel (nilai konstanta)'),Label('font' = Font("courier",14)," dituliskan di ruas kanan. Contoh:  $x + y < 4$  (bukan  $y < 4 - x$ ")),Label('font' = Font("courier",14)," b. Tanda 'kurang dari sama dengan' dituliskan ' $\leq$ ' dan lebih dari sama dengan dituliskan ' $\geq$ '. Penulisan yang"),Label('font' = Font("courier",14)," terbalik, seperti  $=>$  akan menyebabkan eror."),Label('font' = Font("courier",14)," c. Perkalian antara konstanta dengan variabel diberi tanda ' $*$ '. Contoh:  $4x$  ditulis  $4*x$ "),Label('font' = Font("courier",14),"2. Pada Jendela Sistem Pertidaksamaan Linear, kolom Pertidaksamaan yang tidak terpakai (sisa) wajib diisi dengan 0"),Label('font' = Font("courier",14),"3. Isilah dahulu nilai pada kotak yang tersedia sebelum mengeksplorasi pergeseran plot daerah layak. Nilai negatif (-)"),Label('font' = Font("courier",14)," berarti nilai konstan akan turun sebesar nilai yang dimasukkan"),Label('font' = Font("courier",14),"4. Jika Anda mengalami kesulitan pada jendela tertentu, silakan klik menu bantuan"),Label('font' = Font("Arial Black",12),"JIKA ANDA TERLANJUR MELAKUKAN KESALAHAN DALAM MEMASUKKAN PERTIDAKSAMAAN"),Label('font' = Font("Arial Black",12),"TUTUPLAH PROGRAM TERLEBIH DAHULU SEBELUM MELANJUTKAN Pengerjaan"),Label('font' = Font("Arial Black",12),"disusun oleh TEDDY" ))),
```

```
BoxLayout['BL2'](BoxColumn(BoxRow(Plotter['PPL']('width'=340,'height'=340),BoxColumn(BoxRow("Pertidaksamaan :",TextField['TextPL'](8,value='x+y<10'),Button("Plot",onclick='PTtk')),BoxColumn(border = true,caption = Batas_koordinat_untuk_daerah_Plot,BoxRow("x min =",TextField['a'](3,value = ('-1')), "x max =",TextField['b'](3,value = ('20')), "y min =",TextField['c'](3,value = ('-1')), "y max =",TextField['d'](3,value = ('20')))),BoxRow(border=true,caption=Koordinat_Titik,"koordinat x :",TextField['TKTx'](3,value=0),"koordinat y :",TextField['TKTy'](3,value=0),Button("Plot Titik",onclick='PTtk')))),BoxRow(border = true,caption = Pergeseran_Pertidaksamaan,Slider['SLPL'](-1000..1000,1,('onchange') = ('ASLPL'),('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = 0,showlabels = false,filled = true),BoxRow(caption=Pergeseran_nilai_konstan,border=true,TextField['MPPL'](2,value = 1),Button(" +/- ",onclick = ('PLPLUS')))),),
```

```
BoxLayout['BL3']({BoxColumn(BoxRow(BoxColumn(border=true,Plotter['PTL']()),BoxColumn(border = true,"Masukkan pertidaksamaan sebelum","menggeser Slider atau menekan Tombol +/-",["Pertidaksamaan 1:", TextField['PTL1'](8,value=0),Slider['SPTL1'](-1000..1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = 0,onchange = ('ASPTL1'),showlabels = false,filled = true),TextField['GPLUS1'](2,value = 1),Button("+/-",onclick='AGPLUS1')],["Pertidaksamaan 2:", TextField['PTL2'](8,value=0),Slider['SPTL2'](-1000..1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('ASPTL2'),showlabels = false,filled = true),TextField['GPLUS2'](2,value = 1),Button("+/-",onclick='AGPLUS2')],["Pertidaksamaan 3:", TextField['PTL3'](8,value=0),Slider['SPTL3'](-1000..1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('ASPTL3'),showlabels = false,filled = true),TextField['GPLUS3'](2,value = 1),Button("+/-",onclick='AGPLUS3')],["Pertidaksamaan 4:", TextField['PTL4'](8,value=0),Slider['SPTL4'](-1000..1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('ASPTL4'),showlabels = false,filled = true),TextField['GPLUS4'](2,value = 1),Button("+/-",onclick='AGPLUS4')],["Pertidaksamaan 5:", TextField['PTL5'](8,value=0),Slider['SPTL5'](-1000..1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('ASPTL5'),showlabels = false,filled = true),TextField['GPLUS5'](2,value = 1),Button("+/-",onclick='AGPLUS5')],["Pertidaksamaan 6:", TextField['PTL6'](8,value=0),Slider['SPTL6'](-1000..1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') =
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
('0'),onchange = ('ASPTL6'),showlabels = false, filled = true),TextField['GPLUS6'](2,value = 1),Button("+/-",onclick='AGPLUS6'))),BoxColumn(border = true,caption = Batas_koordinat_untuk_daerah_Plot,BoxRow("x min =",TextField['e'](3,value = ('-1')), "x max =",TextField['f'](3,value = ('20')), "y min =",TextField['g'](3,value = ('-1')), "y max =",TextField['h'](3,value = ('20'))),Button("PLOT",onclick = ('BPTL'))),
```

```
BoxLayout['BL4'](inset=1,BoxColumn(inset=1,BoxRow(spacing=1,inset=1,BoxColumn(spacing=1,inset=1,BoxRow(spacing=1,inset=1,BoxColumn(border = true,spacing=1,inset=1,caption = Kendala,BoxRow(spacing=1,inset=1,"kendala 1: ",TextField['T1'](5,value = ('0 <= x')), "kendala 2: ", TextField['T2'](5,value = ('x+y <= 20')),BoxRow(spacing=1,inset=1,"kendala 3: ", TextField['T3'](5,value = ('x <= 15')), "kendala 4: ", TextField['T4'](5,value = ('y <= 10')),BoxRow(spacing=1,inset=1,"kendala 5: ", TextField['T5'](5,value = ('0 <= y')), "kendala 6: ", TextField['T6'](5,value = ('x+2*y <= 18')))),BoxRow(border=true,spacing=1,inset=1,caption=Plot,Plotter['P']()),BoxColumn(spacing=1,inset=1,BoxRow(spacing=1,inset=1,BoxColumn(border = true,["fungsi_sasaran:", TextField['S1'](5,value = x+3*y)],BoxColumn(border = true,caption = Batas,BoxRow("x min =",TextField['i'](3,value = ('-1')), "x max =",TextField['j'](3,value = ('20')), "y min =",TextField['k'](3,value = ('-1')), "y max =",TextField['l'](3,value = ('20')))),BoxRow(BoxColumn(border = true,BoxColumn(border = true,caption = Pergeseran_Fungsi_Sasaran,["Nilai Fungsi Sasaran:",TextField['TSL'](4),Slider['SL1'](-1000 .. 1000,1,('onchange') = ('ASL1'),('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = 0,showlabels = false, filled = true),Button("-",onclick='Minus'),TextField['MP'](3,value = 1),Button("+",onclick='PLUS')),[Button("plot Sasaran di nilai tertentu",onclick='PT')]),BoxColumn(border = true,caption = Uji_Titik_Pojok,BoxColumn(["Plot Sasaran Melalui Titik: ", ("",TextField['UTX'](3), " ", " ",TextField['UTY'](3), " ") " ,Button(" Uji ",onclick='UTP'))),BoxColumn(border = true,caption = Maksimum_Minimum,BoxColumn(["Nilai Minimum =", TextField['TMIN'](3), "di titik :", TextField['TMINKOORD'](3)]),BoxColumn(["Nilai Maksimum=", TextField['TMAX'](3), "di titik :", TextField['TMAXKOORD'](3)]))))) ,BoxRow(Button("Plot",onclick='PLOT'))),
```

```
BoxLayout['BL5']([BoxColumn(BoxRow(spacing=1,inset=1,[Plotter['P1'](), ["fungsi_sasaran:", TextField['S2'](8,editable = false), Slider['SL12'](-1000 .. 1000,1,('onchange') = ('A12'),('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = 0,showlabels = false, filled = true)],BoxColumn(border = true,caption = Maksimum_Minimum,["Nilai Minimum =", TextField['TMIN2'](3), "di titik :", TextField['TMINKOORD2'](15)],["Nilai Maksimum=", TextField['TMAX2'](3), "di titik :", TextField['TMAXKOORD2'](15)]),BoxColumn(spacing=1,inset=1,border = true,BoxColumn(border = true,caption='Geser_Kendala',["kendala1:", TextField['T7'](8), Slider['SL10'](-1000 .. 1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = 0,onchange = ('A6'),showlabels = false, filled = true),TextField['PLUS1'](2,'value' = 1),Button("+/-",onclick='AGPLUS7')),["kendala2:", TextField['T8'](8), Slider['SL2'](-1000 .. 1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('A7'),showlabels = false, filled = true),TextField['PLUS2'](2,'value' = 1),Button("+/-",onclick='AGPLUS8')),["kendala3:", TextField['T9'](8), Slider['SL3'](-1000 .. 1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('A8'),showlabels = false, filled = true),TextField['PLUS3'](2,'value' = 1),Button("+/-",onclick='AGPLUS9')),["kendala4:", TextField['T10'](8), Slider['SL4'](-1000 .. 1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('A9'),showlabels = false, filled = true),TextField['PLUS4'](2,'value' = 1),Button("+/-",onclick='AGPLUS10')),["kendala5:", TextField['T11'](8), Slider['SL5'](-1000 .. 1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('A10'),showlabels = false, filled = true),TextField['PLUS5'](2,'value' = 1),Button("+/-",onclick='AGPLUS11')),["kendala6:", TextField['T12'](8), Slider['SL6'](-1000 .. 1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('0'),onchange = ('A11'),showlabels = false, filled = true),TextField['PLUS6'](2,'value' = 1),Button("+/-",onclick='AGPLUS12'))),BoxColumn(border = true,spacing=1,inset=1,caption = Geser_Gradien_Sasaran,["Plot sasaran melalui titik : ",RadioButton['RB1']("Max", 'value'=true, 'group'='BG1'), RadioButton['RB2']("Min", 'value'=false, 'group'='BG1'), Button("Plot",onclick =
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
('ASLGS1'))],[["koefisien yang diubah :",RadioButton['RB3']("x", 'value'=true, 'group'='BG2'), RadioButton['RB4']("y", 'value'=false, 'group'='BG2') ],["Gradien Fungsi Sasaran:",TextField['mns2b'](4), Slider['SLGS'](0 .. 1000,1,('showticks') = false,('majorticks') = 5,('minorticks') = 1,('snapticks') = true,('value') = ('50'),onchange = ('ASLGS'),showlabels = false,filled = true),TextField['PLUSG'](5, 'value' = 1),Button("+/-",onclick='AGPLUSG')],[["Fungsi sasaran baru: ",TextField['S2b'](6),"dengan nilai: ",TextField['ns2b'](4)],BoxRow(border = true,caption = "Nilai_Maksimum_Minimum Baru,[["Minimum =", TextField['TMINB'](3), "di titik :", TextField['TMINKOORDB'](10)],["Maksimum=", TextField['TMAXB'](3), "di titik :", TextField['TMAXKOORDB'](10)]],TextBox['BBB']()),[Button("Pakai Fungsi Sasaran Baru",onclick = ('Pakai')),Button("Hapus Fungsi Sasaran Baru",onclick = ('TdkPakai'))]]))]]),
```

```
BoxLayout['BLH1'](BoxColumn(valign = top,Button(" Tutup Jendela Bantuan ",onclick='Bh1'),"Cara membuka Jendela Pertidaksamaan Linear:","Pada Jendela Menu Pembuka, Klik File-Buka-Pertidaksamaan Linear.")), BoxLayout['BLH2'](BoxColumn(valign = top,Button(" Tutup Jendela Bantuan ",onclick='Bh2'),"Cara membuka Jendela Sistem Pertidaksamaan Linear:","Pada Jendela Menu Pembuka, Klik File-Buka-Sistem Pertidaksamaan Linear.")), BoxLayout['BLH3'](BoxColumn(valign = top,Button(" Tutup Jendela Bantuan ",onclick='Bh3'),"1. Daerah plot. Daerah plot menampilkan hasil dari penggambaran pertidaksamaan linear yang diberikan.", " Selain itu, daerah plot juga menampilkan keterangan dari titik uji yang diberikan.", "2. Kolom Pertidaksamaan.", " Pada kolom inilah pertidaksamaan yang akan digambar dituliskan.", " Ada beberapa aturan penulisan pertidaksamaan dalam program ini, yaitu:", " a. Semua suku yang mengandung variabel diletakkan di ruas kiri, sedangkan suku yang tidak mengandung variabel (konstanta) diletakkan", " di sebelah kanan. Contoh:  $x + y < 5$  (bukan  $y < 5 - x$ , walaupun secara matematis pernyataan tersebut ekuivalen)", " b. Tanda ' ' ditulis ' $\leq$ ', bukan ' $\leq$ ' ! Contoh:  $x - y - 2$  ditulis  $x - y \leq -2$ ", " c. Tanda ' ' ditulis ' $\geq$ ', bukan ' $\geq$ ' ! Contoh:  $x + y \geq 5$  ditulis  $x + y \geq 5$ ", " d. Perkalian antara konstanta dan variabel diberi tanda ' * '. Contoh:  $4x$  ditulis  $4*x$ !", "3. Tombol Plot. Tombol plot adalah tombol eksekusi dari pertidaksamaan yang dimasukkan.", " Gambar pertidaksamaan yang diminta akan muncul saat tombol eksekusi di-klik.")), BoxLayout['BLH4'](BoxColumn(valign = top,Button(" Tutup Jendela Bantuan ",onclick='Bh4'),"Pengatur Pergeseran Konstanta. Pengatur Pergeseran Konstanta digunakan untuk mengamati perubahan daerah layak apabila", "konstanta berubah. Nilai interval perubahan konstanta di masukkan pada kotak disebelah kiri tombol ' +/- '.", "Nilai yang dapat dimasukkan adalah nilai real. Setelah itu, di-klik tombol ' +/- '.", "Contoh:  $-0.5$  berarti konstanta dari pertidaksamaan yang dituliskan akan berkurang setengah setiap kali klik.", "Itu berarti juga bahwa plot dari pertidaksamaan akan turun setengah setiap kali klik.")), BoxLayout['BLH5'](BoxColumn(valign = top,Button(" Tutup Jendela Bantuan ",onclick='Bh5'),"Pengatur Daerah Plot. Daerah plot dapat diatur dari pengatur daerah plot. Batas-batas sumbu-x dan sumbu-y dapat ditentukan.")), BoxLayout['BLH6'](BoxColumn(valign = top,Button(" Tutup Jendela Bantuan ",onclick='Bh6'),"Titik Uji. Dengan memasukkan nilai (x,y) pada kolom masing-masing, dan meng-klik tombol plot, maka dapat diamati", "keanggotaan titik uji tersebut terhadap daerah layak. Keterangan mengenai keanggotaan ini tampil pada sebelah kanan atas daerah plot.")),
```

```
MenuBar['MB1'](Menu("File", Menu("Buka", MenuItem("Pertidaksamaan Linear", onclick='AM1_1'),MenuItem("Sistem Pertidaksamaan Linear",onclick='AM1_2'),MenuItem("Program Linear",onclick='AM1_3')),MenuItem("Close", Shutdown())),Menu("Bantuan",MenuItem("Pertidaksamaan Linear",onclick='MBh1'),MenuItem("Sistem Pertidaksamaan",onclick='MBh2')) ),
```

```
MenuBar['MB2'](Menu("File", Menu("Buka", MenuItem("Menu Pembuka", onclick='AM2_1'),MenuItem("Sistem Pertidaksamaan Linear",onclick='AM2_2'),MenuItem("Program Linear",onclick='AM2_3')),MenuItem("Reset", onclick='R1'),MenuItem("Close", Shutdown())),Menu("Bantuan",MenuItem("Plot Daerah Layak",onclick='MBh3'),MenuItem("Pergeseran Pertidaksamaan",onclick='MBh4'),MenuItem("Batas Daerah Plot",onclick='MBh5'),MenuItem("Titik Uji",onclick='MBh6')) ),
```

```
MenuBar['MB3'](Menu("File", Menu("Buka", MenuItem("Menu Pembuka", onclick='AM3_1'),MenuItem("Pertidaksamaan Linear",onclick='AM3_2'),MenuItem("Program
```


PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
Linear", onclick='AM3_3')), MenuItem("Reset", onclick='R2'), MenuItem("Close",
Shutdown()))), Menu("Bantuan", MenuItem("Plot Daerah
Layak", onclick='MBh3'), MenuItem("Pergeseran
Pertidaksamaan", onclick='MBh4'), MenuItem("Batas Daerah Plot", onclick='MBh5'))
),
MenuBar['MB4'](Menu("File", Menu("Buka", MenuItem("Menu
Pembuka", onclick='AM4_1'), MenuItem("Pertidaksamaan
Linear", onclick='AM4_2'), MenuItem("Sistem Pertidaksamaan
Linear", onclick='AM4_3'), MenuSeparator(), MenuItem("Sensitifitas", onclick='AM4_4')),
MenuItem("Reset", onclick='R3'), MenuItem("Close",
Shutdown()))), Menu("Pengaturan", CheckBoxMenuItem['CBMI1']("Perlihatkan Titik
Potong", value=true), CheckBoxMenuItem['CBMI2']("Perlihatkan Kendala", value=true))
),
MenuBar['MB5'](Menu("File", MenuItem("Close", Shutdown()),
MenuItem("Kembali", onclick =
('A2'))), Menu("Pengaturan", CheckBoxMenuItem['CBMI3']("Perlihatkan Titik
Potong"), CheckBoxMenuItem['CBMI4']("Perlihatkan Kendala"))
),
ButtonGroup['BG1'](),
ButtonGroup['BG2'](),

Action[A](RunWindow('W1')),
Action[B](RunWindow('W2')),
Action[C](RunWindow('W3')),
Action[D](RunWindow('W4')),
Action['AM1_1'](CloseWindow('W1'), RunWindow('W2'), Evaluate('PPL'='Pertidaksamaan(Te
xtPL,TKTx,TKTy,a,b,c,d)')),
Action['AM1_2'](CloseWindow('W1'), RunWindow('W3'), Evaluate('PTL'='Gambar_PTL(PTL1,P
TL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)')),
Action['AM1_3'](CloseWindow('W1'), RunWindow('W4'), Evaluate('TSL'='TeksSlider(i,j,k,
l,S1,SL1)'), Evaluate('TMAX'='Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('TMAXKOORD'=
'MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'),
Evaluate('TMIN'='Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('TMINKOORD'='MinimumKoord
(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P'='Gambar_Grafik(T1,T2,T3,T4,T5,T6,i,j,k,l,S1,S
L1,CBMI1,CBMI2)')),

Action['AM2_1'](CloseWindow('W2'), RunWindow('W1')),
Action['AM2_2'](CloseWindow('W2'), RunWindow('W3'), Evaluate('PTL'='Gambar_PTL(PTL1,P
TL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)')),
Action['AM2_3'](CloseWindow('W2'), RunWindow('W4'), Evaluate('TSL'='TeksSlider(i,j,k,
l,S1,SL1)'), Evaluate('TMAX'='Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('TMAXKOORD'=
'MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'),
Evaluate('TMIN'='Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('TMINKOORD'='MinimumKoord
(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P'='Gambar_Grafik(T1,T2,T3,T4,T5,T6,i,j,k,l,S1,S
L1,CBMI1,CBMI2)')),
Action['R1'](Shutdown("1")),

Action['AM3_1'](CloseWindow('W3'), RunWindow('W1')),
Action['AM3_2'](CloseWindow('W3'), RunWindow('W2'), Evaluate('PPL'='Pertidaksamaan(Te
xtPL,TKTx,TKTy,a,b,c,d)')),
Action['AM3_3'](CloseWindow('W3'), RunWindow('W4'), Evaluate('TSL'='TeksSlider(i,j,k,
l,S1,SL1)'), Evaluate('TMAX'='Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('TMAXKOORD'=
'MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'),
Evaluate('TMIN'='Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('TMINKOORD'='MinimumKoord
(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P'='Gambar_Grafik(T1,T2,T3,T4,T5,T6,i,j,k,l,S1,S
L1,CBMI1,CBMI2)')),
Action['R2'](Shutdown("2")),

Action['AM4_1'](CloseWindow('W4'), RunWindow('W1')),
Action['AM4_2'](CloseWindow('W4'), RunWindow('W2'), Evaluate('PPL'='Pertidaksamaan(Te
xtPL,TKTx,TKTy,a,b,c,d)')),
Action['AM4_3'](CloseWindow('W4'), RunWindow('W3'), Evaluate('PTL'='Gambar_PTL(PTL1,P
TL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)')),
Action['AM4_4'](RunWindow('W5'), SetOption('target' = 'CBMI3',
Argument('CBMI1')), SetOption('target' = 'CBMI4', Argument('CBMI2')),
SetOption('target' = 'SL12', Argument('SL1')), Evaluate('T7'='pindah(T1)'),
Evaluate('T8'='pindah(T2)'),
Evaluate('T9'='pindah(T3)'), Evaluate('T10'='pindah(T4)'), Evaluate('T11'='pindah(T5)
'), Evaluate('T12'='pindah(T6)'), Evaluate('S2'='pindah(S1)'), Evaluate('TMAX2'='Maksi
mum(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)')
), Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
0, T11, T12, i, j, k, l, S2, SL1, CBMI3, CBMI4)'),  
Action['R3'] (Shutdown("3")),
```

```
Action['PTtk'] (Evaluate('PPL'='Pertidaksamaan(TextPL,TKTx,TKTy,a,b,c,d)'),  
Action['ASLPL'] (Evaluate('TextPL'='geser_kendala(TextPL,SLPL,a,b,c,d)'), Evaluate('P  
PL'='Pertidaksamaan(TextPL,TKTx,TKTy,a,b,c,d)'),  
Action['PLPLUS'] (Evaluate('TextPL'='geser_PLUS(TextPL,MPPL)'), Evaluate('PPL'='Perti  
daksamaan(TextPL,TKTx,TKTy,a,b,c,d)'),
```

```
Action['ASPTL1'] (Evaluate('PTL1'='geser_kendala(PTL1,SPTL1,e,f,g,h)'), Evaluate('PTL  
'='Gambar_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['ASPTL2'] (Evaluate('PTL2'='geser_kendala(PTL2,SPTL2,e,f,g,h)'), Evaluate('PTL  
'='Gambar_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['ASPTL3'] (Evaluate('PTL3'='geser_kendala(PTL3,SPTL3,e,f,g,h)'), Evaluate('PTL  
'='Gambar_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['ASPTL4'] (Evaluate('PTL4'='geser_kendala(PTL4,SPTL4,e,f,g,h)'), Evaluate('PTL  
'='Gambar_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['ASPTL5'] (Evaluate('PTL5'='geser_kendala(PTL5,SPTL5,e,f,g,h)'), Evaluate('PTL  
'='Gambar_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['ASPTL6'] (Evaluate('PTL6'='geser_kendala(PTL6,SPTL6,e,f,g,h)'), Evaluate('PTL  
'='Gambar_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['BPTL'] (Evaluate('PTL'='Gambar_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'))
```

```
Action['AGPLUS1'] (Evaluate('PTL1'='geser_PLUS(PTL1,GPLUS1)'), Evaluate('PTL'='Gambar  
_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['AGPLUS2'] (Evaluate('PTL2'='geser_PLUS(PTL2,GPLUS2)'), Evaluate('PTL'='Gambar  
_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['AGPLUS3'] (Evaluate('PTL3'='geser_PLUS(PTL3,GPLUS3)'), Evaluate('PTL'='Gambar  
_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['AGPLUS4'] (Evaluate('PTL4'='geser_PLUS(PTL4,GPLUS4)'), Evaluate('PTL'='Gambar  
_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['AGPLUS5'] (Evaluate('PTL5'='geser_PLUS(PTL5,GPLUS5)'), Evaluate('PTL'='Gambar  
_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),  
Action['AGPLUS6'] (Evaluate('PTL6'='geser_PLUS(PTL6,GPLUS6)'), Evaluate('PTL'='Gambar  
_PTL(PTL1,PTL2,PTL3,PTL4,PTL5,PTL6,e,f,g,h)'),
```

```
Action['Bh1'] (CloseWindow('H1')),  
Action['Bh2'] (CloseWindow('H2')),  
Action['Bh3'] (CloseWindow('H3')),  
Action['Bh4'] (CloseWindow('H4')),  
Action['Bh5'] (CloseWindow('H5')),  
Action['Bh6'] (CloseWindow('H6')),
```

```
Action['MBh1'] (RunWindow('H1')),  
Action['MBh2'] (RunWindow('H2')),  
Action['MBh3'] (RunWindow('H3')),  
Action['MBh4'] (RunWindow('H4')),  
Action['MBh5'] (RunWindow('H5')),  
Action['MBh6'] (RunWindow('H6')),
```

```
Action['PLOT'] (SetOption('target' = 'UTX', 'value' = ''), SetOption('target' =  
'UTY', 'value' = ''), Evaluate(('TSL') =  
(('TeksSlider(i,j,k,l,S1,SL1)'), Evaluate(('TMAX') =  
(('Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMAXKOORD') =  
(('MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMIN') =  
(('Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMINKOORD') =  
(('MinimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P'='Gambar_Grafik(T1,T2,T3,T4,T5,  
T6,i,j,k,l,S1,SL1,CBMI1,CBMI2)'),  
Action['PT'] (SetOption('target' = 'UTX', 'value' = ''), SetOption('target' = 'UTY',  
'value' = ''), Evaluate(('TMAX') =  
(('Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMAXKOORD') =  
(('MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMIN') =  
(('Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMINKOORD') =  
(('MinimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P' =  
'plotdititik(T1,T2,T3,T4,T5,T6,i,j,k,l,S1,TSL,CBMI1,CBMI2)'),  
Action['ASL1'] (SetOption('target' = 'UTX', 'value' = ''), SetOption('target' =  
'UTY', 'value' = ''), Evaluate(('TSL') =  
(('TeksSlider(i,j,k,l,S1,SL1)'), Evaluate(('TMAX') =  
(('Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMAXKOORD') =  
(('MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMIN') =  
(('Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMINKOORD') =
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
('MinimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('P') =
('Gambar_Grafik(T1,T2,T3,T4,T5,T6,i,j,k,l,S1,SL1,CBMI1,CBMI2)')),
Action['PLUS'](SetOption('target' = 'UTX', 'value' = ''), SetOption('target' =
'UTY', 'value' = ''), Evaluate(('TSL') = ('tambah(TSL,MP)'), Evaluate(('TMAX') =
('Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMAXKOORD') =
('MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMIN') =
('Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMINKOORD') =
('MinimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P'='plotdititik(T1,T2,T3,T4,T5,T6
,i,j,k,l,S1,TSL,CBMI1,CBMI2)')),
Action['Minus'](SetOption('target' = 'UTX', 'value' = ''), SetOption('target' =
'UTY', 'value' = ''), Evaluate(('TSL') = ('kurang(TSL,MP)'), Evaluate(('TMAX') =
('Maksimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMAXKOORD') =
('MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMIN') =
('Minimum(T1,T2,T3,T4,T5,T6,S1)'), Evaluate(('TMINKOORD') =
('MinimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P'='plotdititik(T1,T2,T3,T4,T5,T6
,i,j,k,l,S1,TSL,CBMI1,CBMI2)')),
Action['UTP'](Evaluate('TSL'='TeksSasaran(UTX,UTY,S1)'), Evaluate('P'='Uji_ttk(UTX,U
TY,T1,T2,T3,T4,T5,T6,i,j,k,l,S1,CBMI1,CBMI2)')),

Action['A6'](Evaluate('T7'='geser_kendala(T7,SL10,i,j,k,l)'), Evaluate('TMAX2'='Maksi
imum(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)
'), Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
0,T11,T12,i,j,k,l,S2,SL12,CBMI3,CBMI4)')),
Action['A7'](Evaluate('T8'='geser_kendala(T8,SL2,i,j,k,l)'), Evaluate('TMAX2'='Maksi
imum(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)
'), Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
0,T11,T12,i,j,k,l,S2,SL12,CBMI3,CBMI4)')),
Action['A8'](Evaluate('T9'='geser_kendala(T9,SL3,i,j,k,l)'), Evaluate('TMAX2'='Maksi
imum(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)
'), Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
0,T11,T12,i,j,k,l,S2,SL12,CBMI3,CBMI4)')),
Action['A9'](Evaluate('T10'='geser_kendala(T10,SL4,i,j,k,l)'), Evaluate('TMAX2'='Mak
simum(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)
'), Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
0,T11,T12,i,j,k,l,S2,SL12,CBMI3,CBMI4)')),
Action['A10'](Evaluate('T11'='geser_kendala(T11,SL5,i,j,k,l)'), Evaluate('TMAX2'='Ma
ksimum(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)
)'), Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
0,T11,T12,i,j,k,l,S2,SL12,CBMI3,CBMI4)')),
Action['A11'](Evaluate('T12'='geser_kendala(T12,SL6,i,j,k,l)'), Evaluate('TMAX2'='Ma
ksimum(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)
)'), Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
0,T11,T12,i,j,k,l,S2,SL12,CBMI3,CBMI4)')),
Action['A12'](Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T10,T11,T12,i,j,k,l,S2,SL12,CBM
I3,CBMI4)')),
Action['A2'](SetOption('target' = 'CBMI1', Argument('CBMI3')), SetOption('target' =
'CBMI2', Argument('CBMI4')), SetOption('target' = 'SL1', Argument('SL12')),
Evaluate('T1'='pindah(T7)'),
Evaluate('T2'='pindah(T8)'), Evaluate('T3'='pindah(T9)'),
Evaluate('T4'='pindah(T10)'), Evaluate('T5'='pindah(T11)'), Evaluate('T6'='pindah(T12)
)'), Evaluate('S1'='pindah(S2)'), Evaluate('TMAX'='Maksimum(T1,T2,T3,T4,T5,T6,S1)'), E
valuate('TMAXKOORD'='MaksimumKoord(T1,T2,T3,T4,T5,T6,S1)'),
Evaluate('TMIN'='Minimum(T1,T2,T3,T4,T5,T6,S1)'),
Evaluate('TMINKOORD'='MinimumKoord(T1,T2,T3,T4,T5,T6,S1)'), Evaluate('P'='Gambar_Gra
fik(T1,T2,T3,T4,T5,T6,i,j,k,l,S2,SL12,CBMI1,CBMI2)'), CloseWindow('W5')),

Action['AGPLUS7'](Evaluate('T7'='geser_PLUS(T7,PLUS1)'), Evaluate('TMAX2'='Maksimum(
T7,T8,T9,T10,T11,T12,S2)'), Evaluate('TMIN2'='Minimum(T7,T8,T9,T10,T11,T12,S2)'), Eva
luate(('TMAXKOORD2') =
('MaksimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7,T8,T9,T10,T11,T12,S2)'), Evaluate('P1'='Gambar_Grafik(T7,T8,T9,T1
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
0, T11, T12, i, j, k, l, S2, SL12, CBMI3, CBMI4)'),
Action['AGPLUS8'] (Evaluate('T8'='geser_PLUS(T8, PLUS2)'), Evaluate('TMAX2'='Maksimum(
T7, T8, T9, T10, T11, T12, S2)'), Evaluate('TMIN2'='Minimum(T7, T8, T9, T10, T11, T12, S2)'), Eva
luate(('TMAXKOORD2') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('P1'='Gambar_Grafik(T7, T8, T9, T1
0, T11, T12, i, j, k, l, S2, SL12, CBMI3, CBMI4)'),
Action['AGPLUS9'] (Evaluate('T9'='geser_PLUS(T9, PLUS3)'), Evaluate('TMAX2'='Maksimum(
T7, T8, T9, T10, T11, T12, S2)'), Evaluate('TMIN2'='Minimum(T7, T8, T9, T10, T11, T12, S2)'), Eva
luate(('TMAXKOORD2') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('P1'='Gambar_Grafik(T7, T8, T9, T1
0, T11, T12, i, j, k, l, S2, SL12, CBMI3, CBMI4)'),
Action['AGPLUS10'] (Evaluate('T10'='geser_PLUS(T10, PLUS4)'), Evaluate('TMAX2'='Maksim
um(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('TMIN2'='Minimum(T7, T8, T9, T10, T11, T12, S2)'),
Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('P1'='Gambar_Grafik(T7, T8, T9, T1
0, T11, T12, i, j, k, l, S2, SL12, CBMI3, CBMI4)'),
Action['AGPLUS11'] (Evaluate('T11'='geser_PLUS(T11, PLUS5)'), Evaluate('TMAX2'='Maksim
um(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('TMIN2'='Minimum(T7, T8, T9, T10, T11, T12, S2)'),
Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('P1'='Gambar_Grafik(T7, T8, T9, T1
0, T11, T12, i, j, k, l, S2, SL12, CBMI3, CBMI4)'),
Action['AGPLUS12'] (Evaluate('T12'='geser_PLUS(T12, PLUS6)'), Evaluate('TMAX2'='Maksim
um(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('TMIN2'='Minimum(T7, T8, T9, T10, T11, T12, S2)'),
Evaluate(('TMAXKOORD2') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate(('TMINKOORD2') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2)'), Evaluate('P1'='Gambar_Grafik(T7, T8, T9, T1
0, T11, T12, i, j, k, l, S2, SL12, CBMI3, CBMI4)'),

Action['AGPLUSG'] (Evaluate('mns2b'='tambah(PLUSG, mns2b)'), Evaluate('S2b'='tgradien(
S2, mns2b, RB3)'), Evaluate('ns2b'='ntgradien(RB1, TMINKOORD2, TMAXKOORD2, S2b)'), Evaluate
c(('TMAXB') = ('Maksimum(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMAXKOORDB') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMINB') =
('Minimum(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMINKOORDB') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate('P1'='gradien(T7, T8, T9, T10, T11
, T12, i, j, k, l, ns2b, S2b, CBMI3, CBMI4)'), Evaluate('BBB'='peringatan(RB1, TMINKOORD2, TMAX
KOORD2, TMINKOORDB, TMAXKOORDB)'),

Action['ASLGS1'] (Evaluate('mns2b'='Grad(S2)'), Evaluate('S2b'='pindah(S2)'), Evaluate
('ns2b'='ntgradien(RB1, TMINKOORD2, TMAXKOORD2, S2b)'), Evaluate(('TMAXB') =
('Maksimum(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMAXKOORDB') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMINB') =
('Minimum(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMINKOORDB') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate('P1'='gradien(T7, T8, T9, T10, T11
, T12, i, j, k, l, ns2b, S2b, CBMI3, CBMI4)'), Evaluate('BBB'='peringatan(RB1, TMINKOORD2, TMAX
KOORD2, TMINKOORDB, TMAXKOORDB)'),

Action['ASLGS'] (Evaluate('mns2b'='mntgradien(RB1, TMINKOORD2, TMAXKOORD2, T7, T8, T9, T10
, T11, T12, S2, SLGS)'), Evaluate('S2b'='tgradien(S2, mns2b, RB3)'), Evaluate('ns2b'='ntgra
dien(RB1, TMINKOORD2, TMAXKOORD2, S2b)'), Evaluate(('TMAXB') =
('Maksimum(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMAXKOORDB') =
('MaksimumKoord(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMINB') =
('Minimum(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate(('TMINKOORDB') =
('MinimumKoord(T7, T8, T9, T10, T11, T12, S2b)'), Evaluate('P1'='gradien(T7, T8, T9, T10, T11
, T12, i, j, k, l, ns2b, S2b, CBMI3, CBMI4)'), Evaluate('BBB'='peringatan(RB1, TMINKOORD2, TMAX
KOORD2, TMINKOORDB, TMAXKOORDB)'),

Action['Pakai'] (Evaluate('S2'='pindah(S2b)'), Evaluate('TMAX2'='pindah(TMAXB)'), Eval
uate('TMIN2'='pindah(TMINS)'), Evaluate('TMINKOORD2'='pindah(TMINKOORDB)'), Evaluate(
'TMAXKOORD2'='pindah(TMAXKOORDB)'), SetOption('target' = 'TMINS', 'value' =
''), SetOption('target' = 'TMAXB', 'value' = ''), SetOption('target' = 'TMAXKOORDB',
'value' = ''), SetOption('target' = 'TMINKOORDB', 'value' = ''), SetOption('target' =
'S2b', 'value' = ''), SetOption('target' = 'ns2b', 'value' = ''), SetOption('target' =
'mns2b', 'value' = ''), SetOption('target' = 'BBB', 'value' = '')),
Action['TdkPakai'] (SetOption('target' = 'TMINS', 'value' = ''), SetOption('target' =
'TMAXB', 'value' = ''), SetOption('target' = 'TMAXKOORDB', 'value' =
''), SetOption('target' = 'TMINKOORDB', 'value' = ''), SetOption('target' = 'S2b',
'value' = ''), SetOption('target' = 'ns2b', 'value' = ''), SetOption('target' =
'mns2b', 'value' = ''), SetOption('target' = 'BBB', 'value' = ''))
):
```

PLAGIAT MERUPAKAN TINDAKAN TIDAK TERPUJI

```
| end proc:
> Tampil:=proc(A)
local m;
m:=coba(A);
while (m="1" or m="2" or m="3") do
if m="1" then m:=coba(B);
else if m="2" then m:=coba(C);
else if m="3" then m:=coba(D);
end if;
end if;
end if;
end do;
end proc;
Tampil:=proc(A)
local m;
m:=coba(A);
while m="1" or m="2" or m="3" do
if m="1" then m:=coba(B)
else if m="2" then m:=coba(C) else if m="3" then m:=coba(D) end if end if
end if
end do
end proc
> Tampil(A);
Indahiding Java runtime environment.
C >
C >
I >
```

