METODE *STEEPEST DESCENT* UNTUK MENYELESAIKAN MASALAH OPTIMISASI FUNGSI MULTIVARIABEL TANPA KENDALA

SKRIPSI

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Pendidikan
Program Studi Pendidikan Matematika

Oleh :

Bibiana Kristiani
NIM : 981414012

PROGRAM STUDI PENDIDIKAN MATEMATIKA
JURUSAN PENDIDIKAN MATEMATIKA DAN IPA
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS SANATA DHARMA
YOGYAKARTA

2003

SKRIPSI

METODE *STEEPEST DESCENT*
UNTUK MENYELESAIKAN MASALAH OPTIMISASI FUNGSI MULTIVARIABEL TANPA KENDALA

Oleh:
Bibiana kristiani
NIM: 981414012

Telah disetujui oleh:

Pembimbing

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

Tanggal 5 Desember 2003

SKRIPSI

METODE *STEEPEST DESCENT*
UNTUK MENYELESAIKAN MASALAH OPTIMISASI FUNGSI MULTIVARIABEL TANPA KENDALA

Dipersiapkan dan ditulis oleh:
Bibiana kristiani
NIM: 981414012

Telah dipertahankan di depan Panitia Penguji
pada tanggal 22 Desember 2003
dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

	Nama lengkap	Tanda tangan
Ketua	Drs. A. Atmadi, M. Si.	....
Sekretaris	Drs. Th. Sugiarto, MT.	....
Anggota	M. Andy Rudhito, S.Pd., M.Si.	....
Anggota	Drs. A. Mardjono.	....
Anggota	Wanty Widjaja, S.Pd., M.Ed	....

Yogyakarta, 22 Desember 2003
Fakultas Keguruan dan Ilmu Pendidikan
Universitas Sanata Dharma
Dekan,

Dr. A. M. Slamet Soewandi, M. Pd.

PERSEMBAHAN

"Bukan hal – hal besar, tapi Cintalah yang membuat segalanya menjadi besar".

(St. Theresia Lisieux)

Dengan Penuh Syukur
Kupersembahkan Skripsi ini bagi
Keluarga besar Kongregasi
Biarawati Abdi Kristus tercinta.

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 Desember 2003

Penulis

Bibiana Kristiani

ABSTRAK

Optimisasi fungsi lebih dari dua variabel, yaitu fungsi $f : \Re^n \rightarrow \Re$ sangat sukar diselesaikan secara analitis . Kendati demikian, tidak berarti bahwa masalah tersebut tidak dapat diselesaikan. Metode numeris dapat mengatasi permasalahan tersebut, yakni dengan menggunakan metode *Steepest Descent*. Dalam metode *Steepest descent* ini, penyelesaian masalah optimisasi fungsi multivariabel dibawa ke fungsi satu variabel, yaitu fungsi alpha ($f(\alpha)$). Perhitungan α_k pada metode tersebut bertujuan untuk memilih pencapaian jumlah maksimum berkurangnya nilai fungsi obyektif, yakni $\alpha_k = \min f(\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}))$, $\alpha \geq 0$. Perhitungan dalam menyelesaikan masalah tersebut akan menjadi lebih efektif dan efisien apabila dilakukan dengan menggunakan alat bantu komputer, yaitu program MATLAB untuk metode *Steepest Descent*. Pemanggilan program dan penyelesaiannya dikerjakan dalam *Command Windows*. Dalam skripsi ini, penerapan metode *Steepest Descent* digunakan untuk menentukan kuantitas suatu jenis TV Warna yang harus diproduksi oleh suatu perusahaan agar laba yang diperoleh maksimum.

ABSTRACT

The optimization function more than two variables, the function $f: \Re^n \rightarrow \Re$ is very difficult to solved by analysis. But, it doesn't mean that the problem couldn't solved. The numerical methods could solve the problem, using Steepest Descent method. The solution of optimization problem for multivariables function change into the one variable function in method, that is alpha function ($f(\alpha)$). On the step size α_k in the method to choose the achieve of maximum amount of the objective function decrease, that is $\alpha_k = \min f(\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}))$, $\alpha \geq 0$. Calculation the solving problem will effective and efficient if using the computer, the MATLAB program to Steepest Descent method. Starting and finishing work into the Command Windows. Application into the script of Steepest Descent method used to decide of the colour TV quantity that have to produced by company to get maximum profit.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Kasih atas rahmat yang senantiasa dilimpahkan kepada penulis sehingga skripsi ini dapat terselesaikan dengan baik.

Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Strata I (SI) dan meraih gelar Sarjana Pendidikan pada Program Studi Pendidikan Matematika di Universitas Sanata Dharma Yogyakarta.

Dalam proses penulisan skripsi ini, penulis menyadari banyak memperoleh bimbingan, bantuan dan dukungan dari berbagai pihak. Oleh karena itu dengan segala kerendahan hati penulis mengucapkan banyak terima kasih kepada:

1. M. Andy Rudhito. S.Pd., M.Si, selaku Dosen Pembimbing yang dengan penuh kesabaran, ketulusan dan kesetiaan memberikan dorongan, bimbingan, kritikan dan saran-saran kepada penulis selama proses penulisan skripsi.
2. Drs. Th. Sugiarto, MT, selaku Kaprodi Pendidikan Matematika yang telah memberikan dorongan kepada penulis untuk menyelesaikan skripsi.
3. Drs. Al. Haryono selaku Dosen Pembimbing Akademik.
4. Drs. St. Susento, M.Si, atas segala motivasinya sehingga penulis merasa terdorong untuk segera menyusun skripsi ini.
5. Para Dosen atas segala bimbingan yang diberikan selama penulis belajar di Universitas Sanata Dharma.
6. Sekretariat JPMIPA atas segala bantuannya selama penulis belajar di Universitas Sanata Dharma.
7. Segenap Staf Perpustakaan USD Unit Paingan, atas segala perhatian, keramahan dan pelayanan yang tulus.
8. Sr. M. Goretti Sugiarti. AK, selaku Pimpinan Umum Kongregasi Biarawati Abdi Kristus beserta staf Dewan yang telah memberi kesempatan dan kepercayaan kepada penulis untuk studi di Universitas Sanata Dharma, terima

kasih atas segala dukungan, doa-doa dan fasilitas yang penulis terima selama studi dan menyelesaikan skripsi ini.

9. Sr. Didima. AK dan Sr. Mathea AK, selaku Pimpinan Komunitas serta Para suster AK Komunitas Condronegaran Yogyakarta dan Komunitas Postulat, terima kasih atas doa-doa, kebersamaan, dan dukungan yang diberikan selama penulis studi di USD.
10. Para Suster AK dimanapun berada yang telah turut mendukung penulis dengan segala perhatian dan doa-doanya.
11. Rm. J. Setyakarjana. SJ, terima kasih atas segala dukungan, perhatian dan bantuannya termasuk meng'in-stall' program MATLAB.
12. Orangtua dan saudara-saudaraku, terima kasih atas segala dukungan, perhatian dan doa-doa yang diberikan kepada penulis dalam meniti peziarahan Panggilan dan melaksanakan tugas studi yang dipercayakan Kongregasi.
13. Mbak Kanti. P dan Sr. Olinda. Fdcc, terima kasih atas segala persahabatan, persaudaran, pengalaman saling berbagi suka dan duka selama studi di USD.
14. Br. Thomas FIC dan teman-teman PMAT'98 khususnya Okta, Sylvi dan Nanin, terima kasih atas dukungan dan kebersamaan kita selama studi di USD.
15. Semua pihak yang tak dapat disebut satu persatu.

Penulis menyadari adanya kekurangan dan keterbatasan dalam penyusunan skripsi ini, maka saran dan kritik yang sifatnya membangun akan diterima dengan hati terbuka.

Akhirnya, penulis berharap agar skripsi ini dapat bermanfaat baik bagi penulis sendiri maupun para pembaca.

Penulis

DAFTAR ISI

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Salah satu masalah optimisasi dalam matematika adalah masalah mencari nilai ekstrim suatu fungsi, yakni mencari nilai maksimum atau minimum suatu fungsi. Optimisasi fungsi satu variabel atau dua variabel merupakan masalah optimisasi yang masih sederhana dan relatif mudah diselesaikan secara analitis, sedangkan masalah optimisasi fungsi lebih dari dua variabel, yakni fungsi $f: \Re^n \rightarrow \Re$ lebih kompleks dan sulit diselesaikan secara analitis. Untuk itu telah dikembangkan suatu metode numeris yang dapat mengatasi permasalahan tersebut, meskipun hasil yang diperoleh hanya pendekatannya saja.

Perhitungan secara numeris memang kurang praktis dan tidak efisien apabila dilakukan secara manual, karena perhitungannya panjang dan berulang-ulang. Untuk itu komputer dapat digunakan sebagai alat bantu untuk melakukan perhitungan secara cepat. Oleh karena itulah dalam skripsi ini selain membahas metode *Steepest Descent* dan penerapannya, juga akan dibahas program MATLAB untuk metode *Steepest Descent*.

B. Perumusan Masalah

Pokok perumusan masalah yang akan dibahas di sini adalah :

1. Apakah metode *Steepest Descent* itu dan bagaimana metode *Steepest Descent* tersebut digunakan untuk menyelesaikan masalah optimisasi fungsi multivariabel tanpa kendala?
2. Bagaimanakah program MATLAB untuk metode *Steepest Descent* ?
3. Bagaimana penerapan metode *Steepest Descent* dalam kehidupan sehari-hari?

C. Tujuan Penulisan

Tujuan penulisan skripsi ini adalah untuk mengetahui tentang :

1. Metode *Steepest Descent* dan penggunaan metode *Steepest Descent* tersebut dalam menyelesaikan masalah optimisasi fungsi multivariabel tanpa kendala
2. Bagaimana program MATLAB dan penggunaan metode *Steepest Descent*
3. Bagaimana penerapan metode *Steepest Descent* dalam kehidupan sehari-hari

D. Manfaat Penulisan

Dengan pembahasan dan penulisan skripsi tentang Metode *Steepest Descent* Untuk Menyelesaikan Masalah Optimisasi Fungsi Multivariabel Tanpa Kendala ini diharapkan bermanfaat bagi beberapa pihak terutama :

1. Bagi Universitas Sanata Dharma

Diharapkan dapat menambah referensi kepustakaan dan dapat digunakan sebagai bahan acuan bagi pihak-pihak yang membutuhkan

2. Bagi Penulis

Penulis memperoleh pengetahuan dan pengalaman berharga serta wawasan yang luas tentang metode *Steepest Descent* dan bagaimana cara pemrogramannya dengan menggunakan MATLAB.

E. Metode Penulisan

Metode yang digunakan dalam penulisan skripsi ini adalah metode studi pustaka, yaitu dengan mempelajari buku-buku pendukung yang berkaitan dengan metode *Steepest Descent* dan menuliskan programnya dengan program MATLAB.

F. Sistematika Pembahasan

Pembahasan skripsi ini terbagi dalam lima bab, Bab I merupakan pendahuluan yang berisi latar belakang masalah, perumusan masalah, manfaat penulisan, metode penulisan, sistematika pembahasan dan materi prasyarat.

Dalam Bab II, dibahas tentang materi-materi prasyarat yang akan mendasari pembahasan-pembahasan selanjutnya. Pada sub bab pertama akan dibahas tentang ruang *Euclides* berdimensi-n ($\Re^n$). Pada sub bab kedua akan dibahas tentang kalkulus diferensial yang meliputi: keterdiferensialan, vektor gradien, matriks *Hessian*, aturan rantai dan deret *Taylor*. Akhirnya pada sub bab ketiga akan dibahas tentang optimisasi fungsi tanpa kendala

Dalam Bab III dibahas tentang metode *Steepest Descent* yang merupakan materi pokok. Sebelum pembahasan tentang materi pokok tersebut, terlebih dahulu pada sub bab pertama dan kedua berturut-turut akan dibahas tentang metode *Newton* dan metode *Secant*, untuk selanjutnya dibahas tentang metode *Steepest Descent*.

Pembahasan Bab III di atas dilanjutkan dengan Bab IV yang berisi tentang contoh penerapan metode *Steepest Descent*. Pemecahannya dilakukan dengan bantuan komputer, yakni program MATLAB untuk metode *Steepest Descent*.

Bab terakhir, yakni Bab V, berisi tentang beberapa kesimpulan dan saran.

G. Materi Prasyarat

Dalam skripsi ini diasumsikan pembaca telah mengenal konsep-konsep dasar aljabar linear dan kalkulus.

BAB II

LANDASAN TEORI

Pada bab ini akan dibahas tentang : ruang *Euclides* berdimensi-n, keterdifernsialan fungsi, vektor gradien, matriks *Hessian*, aturan rantai, deret *Taylor*, optimisasi fungsi tanpa kendala, pembuat minimum lokal dan global, arah layak, syarat perlu tingkat pertama dan kedua, dan syarat cukup tingkat kedua, yang merupakan materi prasyarat untuk pembahasan bab-bab selanjutnya. Definisi, teorema serta konsep-konsep mengacu pada daftar pustaka.

A. Ruang Euclides Berdimensi-n ($\Re^n$)

Definisi 2.1

Ruang Euclides berdimensi-n dinotasikan dengan $\Re^n$ *adalah himpunan semua matriks berorde nx1 dengan elemen-elemen bilangan real.*

Matriks berorde nx1 dapat juga dinyatakan sebagai *transpose* dari matriks berorde nx1, yakni matriks berorde 1x n. Anggota dari $\Re^n$ disebut vektor.

Definisi 2.2

Vektor $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$ *dan vektor* $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$ *dalam* $\Re^n$ *dikatakan sama jika* $u_i = v_i$ *, untuk setiap i.*

Selanjutnya penjumlahan dua vektor **u** dan **v** dinyatakan dengan **u** + **v** didefinisikan oleh :

$$\mathbf{u} + \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = [u_1 + v_1,\ u_2 + v_2,\ \ldots,\ u_n + v_n\,]^T.$$

Definisi 2.3

Suatu vektor dalam $\Re^n$ *yang elemennya sama dengan nol disebut vektor nol dan dinyatakan dengan :* $\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ *atau* $\mathbf{0} = [0,\ 0,\ \ldots,\ 0]^T$.

Definisi 2.4

Andaikan vektor $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$ *atau dinyatakan juga dengan* $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$ *sebarang vektor dalam* $\Re^n$ *maka vektor* $-\mathbf{u} = [-u_1, -u_2, \ldots, -u_n]^T$ *disebut* <u>*invers*</u> *terhadap operasi penjumlahan dari* $\mathbf{u}$.

Selanjutnya <u>operasi pengurangan</u> dalam $\Re^n$ didefinisikan sebagai :

$$\mathbf{u} - \mathbf{v} = \mathbf{u} + (-\mathbf{v}) = [u_1 - v_1, u_2 - v_2, \ldots, u_n - v_n]^T,$$

untuk setiap $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$ dan $\mathbf{v} = [v_1, v_2, \ldots, v_n]^T$.

Definisi 2.5

Dua vektor **u** dan **v** *dikatakan saling ortogonal jika* $\mathbf{u} \cdot \mathbf{v} = 0$, *dan dilambangkan dengan* $\mathbf{u} \perp \mathbf{v}$.

Sifat-sifat operasi penjumlahan dan perkalian skalar vektor pada $\Re^n$ dinyatakan dalam teorema berikut :

Teorema 2.1

Jika $\mathbf{u} = (u_1, u_2, \ldots, u_n)^T$, $\mathbf{v} = (v_1, v_2, \ldots, v_n)^T$ *dan* $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$ *adalah vektor-vektor pada* $\Re^n$ *dan k serta l adalah skalar dalam* $\Re$, *maka berlaku :*

a. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$

b. $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$

c. $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$

d. $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$, yaitu $\mathbf{u} - \mathbf{u} = \mathbf{0}$

e. $k(l\mathbf{u}) = (kl)\mathbf{u}$

f. $k(\mathbf{u} + \mathbf{v}) = k\mathbf{u} + k\mathbf{v}$

g. $(k + l)\mathbf{u} = k\mathbf{u} + l\mathbf{u}$

h. $1\mathbf{u} = \mathbf{u}$

Bukti :

a. $\mathbf{u} + \mathbf{v} = (u_1, u_2, \ldots, u_n)^T + (v_1, v_2, \ldots, v_n)^T$

$= (u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n)^T$

$= (v_1 + u_1, v_2 + u_2, \ldots, v_n + u_n)^T$

$= (v_1, v_2, \ldots, v_n)^T + (u_1, u_2, \ldots, u_n)^T$

$= \mathbf{v} + \mathbf{u}.$

b. $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (u_1, u_2, \ldots, u_n)^T + ((v_1, v_2, \ldots, v_n)^T + (w_1, w_2, \ldots, w_n)^T)$

$= (u_1, u_2, \ldots, u_n)^T + (v_1 + w_1, v_2 + w_2, \ldots, v_n + w_n)^T$

$= (u_1 + (v_1 + w_1), u_2 + (v_2 + w_2), \ldots, u_n + (v_n + w_n))^T$

$= ((u_1 + v_1) + w_1), (u_2 + v_2) + w_2), \ldots, (u_n + v_n) + w_n)^T$

$= ((u_1 + v_1), (u_2 + v_2), \ldots, (u_n + v_n))^T + (w_1, w_2, \ldots, w_n)^T)$

$= ((u_1, u_2, \ldots, u_n)^T + (v_1, v_2, \ldots, v_n)^T + (w_1, w_2, \ldots, w_n)^T)$

$= (\mathbf{u} + \mathbf{v}) + \mathbf{w}.$

c. $\mathbf{u} + \mathbf{0} = (u_1, u_2, \ldots, u_n)^T + (0, 0, \ldots, 0)^T$

$= (u_1 + 0, u_2 + 0, \ldots, u_n + 0)^T$

$= (u_1, u_2, \ldots, u_n)^T$

$= \mathbf{u}$.

d. $\mathbf{u} + (-\mathbf{u}) = (u_1, u_2, \ldots, u_n)^T + (-u_1, - u_2, \ldots, -u_n)^T$

$= (u_1 - u_1, u_2 - u_2, \ldots, u_n - u_n)^T$

$= (0, 0, \ldots, 0)^T = \mathbf{0}$.

e. $k\,(l\mathbf{u}) = k\,(l\,(u_1, u_2, \ldots, u_n)^T)$

$= k\,(l\,u_1, l\,u_2, \ldots, l\,u_n)^T$

$= (k\,(l\,u_1), (k\,l\,u_2), \ldots, (k\,l\,u_n))^T$

$=(((kl)\,u_1), ((kl)\,u_2), \ldots,((kl)\,u_n))^T$

$= ((kl)\,(u_1, u_2, \ldots, u_n)^T)$

$= (kl)\,\mathbf{u}$.

f. $k\,(\,\mathbf{u} + \mathbf{v}\,) = k\,((u_1, u_2, \ldots, u_n)^T) + (v_1, v_2, \ldots, v_n)^T$

$= k\,((u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n)^T)$

$= (k\,(u_1 + v_1), k\,(u_2 + v_2), \ldots, k\,(u_n + v_n))^T$

$= (k\,u_1 + k\,v_1, k\,u_2 + k\,v_2, \ldots, ku_n + k\,v_n)^T$

$= (k\,u_1, k\,u_2, \ldots, ku_n)^T + (k\,v_1, k\,v_2, \ldots, k\,v_n)^T$

$= k\,(u_1, u_2, \ldots, u_n)^T + k\,(v_1, v_2, \ldots, v_n)^T$

$= k\mathbf{u} + k\mathbf{v}$.

g. $(k + l)\,\mathbf{u} = (k+l)\,(u_1, u_2, \ldots, u_n)^T$

$= ((k+l)\, u_1, (k+l)u_2, \ldots, (k+l)\, u_n)^T$

$= (ku_1+lu_1, ku_2+lu_2, \ldots, ku_n+lu_n)^T$

$= (ku_1, ku_2, \ldots, ku_n)^T + (lu_1, lu_2, \ldots, lu_n)^T$

$= (k(u_1, u_2, \ldots, u_n)^T + l(u_1, u_2, \ldots, u_n)^T)$

$= k\mathbf{u}+l\mathbf{u}.$

h. $1\mathbf{u} = 1(u_1, u_2, \ldots, u_n)^T$

$= (1u_1, 1\, u_2, \ldots, 1\, u_n)^T$

$= (u_1, u_2, \ldots, u_n)^T$

$= \mathbf{u}.$ ■

Definisi 2.6

Jika $\mathbf{u} = (u_1, u_2, \ldots, u_n)^T$ *dan* $\mathbf{v} = (v_1, v_2, \ldots, v_n)^T$ *adalah sebarang vektor pada* $\Re^n$ *maka* hasil kali dalam Euclides *vektor* **u** *dan* **v** *didefinisikan sebagai :* $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v}$.

Contoh 1 :

Bila $\mathbf{u} = (1, 3, 2, 6, -1)^T$ dan $\mathbf{v} = (0, 0, 2, 4, 1)^T$, *hasil kali dalam Euclides* dari **u** dan **v** adalah :

$$\mathbf{u} \cdot \mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 6 \\ -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 2 & 4 & 1 \end{bmatrix} = 1(0) + 3(0) + 2(2) + 6(4) + (-1)(1) = 27.$$

Sifat-sifat *hasil kali dalam Euclides* dinyatakan dalam teorema berikut :

Teorema 2.2

Jika **u**, **v**, *dan* **w** *adalah vektor dalam* $\Re^n$ *dan k adalah sebarang skalar, maka berlaku :*

a. $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$

b. $(\mathbf{u} + \mathbf{v}) \cdot \mathbf{w} = \mathbf{u} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{w}$

c. $(k\mathbf{u}) \cdot \mathbf{v} = k(\mathbf{u} \cdot \mathbf{v})$

d. $\mathbf{v} \cdot \mathbf{v} \geq 0$. dan $\mathbf{v} \cdot \mathbf{v} = 0$ bila dan hanya bila $\mathbf{v} = 0$.

Bukti :

a. $\mathbf{u} \cdot \mathbf{v} = (u_1, u_2, \ldots, u_n)^T \cdot (v_1, v_2, \ldots, v_n)^T$

$= (u_1.v_1, u_2. v_2, \ldots, u_n . v_n)^T$

$= (v_1. u_1, v_2. u_2, \ldots, v_n . u_n)^T$

$= . (v_1, v_2, \ldots, v_n)^T . (u_1, u_2, \ldots, u_n)^T$

$= \mathbf{v} \cdot \mathbf{u}$

b. $(\mathbf{u} + \mathbf{v}) \cdot \mathbf{w} = (u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n)^T . (w_1, w_2, \ldots, w_n)^T$

$= \{(u_1 + v_1) . w_1 + (u_2 + v_2) . w_2 + (u_n + v_n)\}$

$= (u_1.w_1 , u_2.w_2, \ldots, u_n.w_n)^T + (v_1.w_1, v_2.w_2, \ldots, v_n.w_n)^T$

$= \mathbf{u} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{w}$

c. $(k\mathbf{u}) \cdot \mathbf{v} = (ku_1, ku_2, \ldots, ku_n)^T . (v_1, v_2, \ldots, v_n)^T$

$= k((u_1, u_2, \ldots, u_n)^T . (v_1, v_2, \ldots, v_n)^T)$

$= k\,(u_1.v_1, u_2.\, v_2, \ldots, u_n\,.\, v_n)^T$

$= k\,(\,\mathbf{u}\,.\,\mathbf{v}\,)$

d. Di perhatikan $\mathbf{v}\,.\,\mathbf{v} = v_1^{\;2} + v_2^{\;2} + \ldots + v_n^{\;2} \geq 0$, karena $v_i^2 \geq 0\ \forall_i$. Selanjutnya kesamaan tersebut benar jika dan hanya jika $v_1 = v_2 = \ldots = v_n = 0$, yakni $\mathbf{v} = 0$. ■

Definisi 2.7

Norma Euclides vektor $\mathbf{u} = (\,u_1, u_2, \ldots, u_n\,)^T$ *dalam* $\Re^n$ *didefinisikan sebagai :*

$$\|\mathbf{u}\| = (\,\mathbf{u}\,.\,\mathbf{u}\,)^{1/2} = \sqrt{u_1^2 + u_2^2 + \ldots u_n^2}$$

Contoh 2 :

Andaikan $\mathbf{v} = (\,-1, 1, 1, 3, 6\,)^T$ maka norma *Euclides* dari vektor **v** adalah :

$$\|\mathbf{u}\| = \sqrt{-1^2 + 1^2 + 1^2 + 3^2 + 6^2}$$

$$\|\mathbf{u}\| = \sqrt{1 + 1 + 1 + 9 + 36}$$

$$= \sqrt{48}$$

Definisi 2.8

Jika $\mathbf{u} = (\,u_1, u_2, \ldots, u_n\,)^T$ *dan* $\mathbf{v} = (\,v_1, v_2, \ldots, v_n\,)^T$, *jarak antara* **u** *dan* **v** *pada* $\Re^n$ *didefinisikan oleh :*

$$d\,(\,\mathbf{u}, \mathbf{v}\,) = \|\mathbf{u} - \mathbf{v}\| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \ldots + (u_n - v_n)^2}$$

B. Kalkulus Diferensial

1. Keterdiferensialan

Suatu fungsi real dalam satu variabel dikatakan kontinu pada $x = a$ jika:

a. $f(x)$ didefinisikan, yakni ada $x = a$

b. $\lim_{x \to a} f(x)$ ada, dan

c. $\lim_{x \to a} f(x) = f(a)$.

Selanjutnya suatu fungsi dikatakan diferensiabel pada suatu titik jika derivatif pada titik tersebut ada.

Kalkulus diferensial didasarkan pada gagasan aproksimasi suatu fungsi yang berubah-ubah oleh suatu fungsi *affine* . Suatu fungsi $\mathcal{A}: \Re^n \to \Re^m$ dikatakan *affine* jika terdapat suatu fungsi linear $£: \Re^n \to \Re^m$ dan suatu vektor $\mathbf{y} \in \Re^m$ sedemikian sehingga

$$\mathcal{A}(\mathbf{x}) = £(\mathbf{x}) + \mathbf{y} \text{ untuk setiap } \mathbf{x} \in \Re^n$$

Untuk menemukan suatu fungsi *affine* yang aproksimasi f nya dekat x_0, terlebih dahulu ditentukan suatu kondisi awal $\mathcal{A}(x_0) = f(x_0)$

$$\mathcal{A}(\mathbf{x}) = £(\mathbf{x}) + \mathbf{y}$$

$$\mathcal{A}(\mathbf{x}_0) = £(\mathbf{x}_0) + \mathbf{y}$$

$$f(\mathbf{x}_0) = £(\mathbf{x}_0) + \mathbf{y}$$

$$\mathbf{y} = f(\mathbf{x}_0) - £(\mathbf{x}_0)$$

$$\begin{aligned}
\pounds(\mathbf{x}) + \mathbf{y} &= \pounds(\mathbf{x}) + f(\mathbf{x}_0) - \pounds(\mathbf{x}_0) \\
&= \pounds(\mathbf{x}) - \pounds(\mathbf{x}_0) + f(\mathbf{x}_0) \\
&= \pounds\,(\mathbf{x} - \mathbf{x}_0) + f(\mathbf{x}_0)
\end{aligned}$$

$$\therefore \mathscr{A}(\mathbf{x}) = \pounds\,(\mathbf{x} - \mathbf{x}_0) + f(\mathbf{x}_0)$$

Selanjutnya $\mathscr{A}(\mathbf{x})$ menghampiri $f(\mathbf{x})$ lebih cepat dari pada $\mathbf{x}$ menghampiri x_0, yaitu $\lim\limits_{x \to x_0 \in \Omega} \dfrac{\| f(x) - \mathscr{A}(x) \|}{\| x - x_0 \|} = 0$. Kondisi tersebut menjamin bahwa $\mathscr{A}$ menghampiri f dekat $\mathbf{x}_0$ dalam arti bahwa kesalahan dalam aproksimasi pada suatu titik adalah 'kecil' dibandingkan dengan jarak titik dari $\mathbf{x}_0$.

Secara ringkas, dalam fungsi $f : \Omega \to \Re^m$, $\Omega \subset \Re^n$ dikatakan diferensiabel pada $x_0 \in \Omega$ jika terdapat suatu fungsi *affine* yang menghampiri f dekat $\mathbf{x}_0$, yaitu terdapat suatu fungsi linear $\pounds : \Re^n \to \Re^m$ sedemikian sehingga

$$\lim_{x \to x_0 \in \Omega} \frac{\| f(x) - \pounds(x - x_0) + f(x_0) \|}{\| x - x_0 \|} = 0$$

Fungsi linear $\pounds$ di atas ditentukan oleh f dan $\mathbf{x}_0$, dan dinamakan derivatif f terhadap $\mathbf{x}_0$ .Suatu fungsi f dikatakan diferensiabel pada Ω jika f diferensiabel di setiap titik dari domain Ω.

Dalam $\Re$, suatu fungsi *affine* mempunyai bentuk ax + b, di mana $a,b \in \Re$. Oleh karena itu suatu fungsi $f(x)$ bernilai real dari variabel real x yaitu, diferensiabel pada x_0 dapat diaproksimasikan dekat x_0 oleh suatu fungsi

$\mathcal{A}(x) = ax + b.$

Karena $f(x_0) = \mathcal{A}(x_0) = ax_0 + b$, maka:

$\mathcal{A}(x) = ax + b$

$\mathcal{A}(x_0) = ax_0 + b$

$f(x_0) = ax_0 + b$

$b = f(x_0) - ax_0$

$\mathcal{A}(x) = ax + f(x_0) - ax_0$

$= ax - ax_0 + f(x_0)$

$= a(x - x_0) + f(x_0)$

$\therefore \mathcal{A}(x) = ax + b = a(x - x_0) + f(x_0)$

Bagian linear dari $\mathcal{A}(x)$, lebih mudah dinyatakan oleh £(x), hanya dalam kasus ax. Bentuk dari suatu bilangan real yaitu nilai mutlak , juga mendefinisikan keterdiferensialan menjadi $\lim\limits_{x \to 0} \frac{|f(x) - (a(x - x_0) + f(x_0))|}{|x - x_0|} = 0$

yang ekuivalen dengan $\lim\limits_{x \to 0} \frac{|f(x) - f(x_0)|}{|x - x_0|} = a$. Bilangan *a* biasanya dinyatakan dengan $f'(x_0)$ dan dinamakan derivatif f terhadap x_0. Oleh karena itu fungsi *affine* diberikan oleh : $\mathcal{A}(x) = f(x_0) + f'(x_0)(x - x_0)$.

2. Vektor Gradien

Definisi 2.9

Diberikan suatu fungsi $f: \Re^n \rightarrow \Re$. *Gradien dari* f *pada* $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ *dinyatakan dengan* $\nabla f(\mathbf{x})$ *didefinisikan sebagai :*

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \ldots, \frac{\partial f}{\partial x_n}(x)\right)^T.$$ *Atau dinyatakan dengan vektor kolom*

sebagai : $$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

Contoh 4 :

Diberikan suatu fungsi $f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 1)^2$ dan vektor $\mathbf{x} = (1.8, 1.6)$.

Vektor gradien dari $f(\mathbf{x})$ pada vektor (1.8, 1.6) dihitung sebagai berikut:

Nilai fungsi f pada $\mathbf{x}$ yang diberikan adalah:

$$f(1.8, 1.6) = (1.8 - 1)^2 + (1.6 - 1)^2$$
$$= 0.64 + 0.76$$
$$= 1.40$$

$$\frac{\partial x}{\partial x1}(1.8, 1.6) = 2(x_1 - 1)$$
$$= 2(1.8 - 1)$$
$= 1,6$

$$\frac{\partial x}{\partial x_2}(1,8, 1,6) = 2(x_2 - 1)$$
$$= 2(1,6 - 1)$$
$$= 1,2$$

$$\nabla f(x) = \begin{bmatrix} 1,6 \\ 1,2 \end{bmatrix}$$

$= 1.6$

$\frac{\partial x}{\partial x_2}(1.8, 1.6) = 2\,(x_2 - 1)$

$= 2\,(1.6 - 1)$

$= 1.2$

Jadi $\nabla f(\mathbf{x}) = \begin{bmatrix} 1.6 \\ 1.2 \end{bmatrix}$

3. Matriks *Hessian*

Definisi 2.10

Diberikan suatu fungsi f: $\Re^n \to \Re$. Matriks Hessian dari f merupakan suatu matriks yang diperoleh dari derivatif parsial kedua untuk fungsi f ($\mathbf{x}$).

Matriks *Hessian* tersebut dinyatakan sebagai:

$$D^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Contoh 5:

Diberikan suatu fungsi $f(\mathbf{x}) = x_1^3 + x_2^3 + 2x_1^2 + 3x_2^2 - x_1x_2 + 2x_1 + 4x_2$ dan vektor (1, 2).

Vektor gradien dan matriks *Hessian* dari fungsi tersebut dihitung sebagai berikut:

Derivatif parsial pertama dari fungsi adalah :

$$\frac{\partial f}{\partial x_1} = 3x_1^2 + 4x_1 - x_2 + 2$$

$$\frac{\partial f}{\partial x_2} = 3x_2^2 + 6x_2 - x_1 + 4$$

Disubstitusikan $x_1 = 1$, $x_2 = 2$, diperoleh:

$$\frac{\partial f}{\partial x_1} = 3(1)^2 + 4(1) - 2 + 2$$

$$= 3 + 4 = 7$$

$$\frac{\partial f}{\partial x_2} = 3(2)^2 + 6(2) - 1 + 4$$

$$= 12 + 12 + 3$$

$$= 27.$$

Jadi vektor gradien dari fungsi tersebut adalah: $\nabla f(1,2) = \begin{bmatrix} 7 \\ 27 \end{bmatrix}$

Derivatif parsial kedua dari fungsi adalah:

$$\frac{\partial^2 f}{\partial x_1^2} = 6x_1 + 4 = 6(1) + 4 = 10$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = -1$$

$$\frac{\partial^2 f}{\partial x_2 \partial x_1} = -1$$

$$\frac{\partial^2 f}{\partial x_2^{\,2}} = 6x_2 + 6 = 6(2) + 6 = 18.$$

Oleh karena itu, matriks *Hessian* pada vektor (1,2) yang diberikan adalah:

$$\nabla^2 f(1,2) = \begin{bmatrix} 10 & -1 \\ -1 & 18 \end{bmatrix}$$

4. Aturan Rantai

Suatu fungsi $f : \Omega \to \Re^m$, $\Omega \subset \Re^n$, dikatakan diferensiabel kontinu pada Ω jika fungsi tersebut diferensiabel pada Ω, dan $Df\ \Omega \to \Re^{m \times n}$ kontinu, yakni komponen dari f mempunyai derivatif parsial kontinu.

Teorema 2.3

Andaikan $g : \Psi \to \Re$ diferensiabel kontinu pada suatu himpunan terbuka $\Psi \subset \Re^n$, dan andaikan $f : (a,b) \to \Psi$ diferensiabel pada (a,b). Fungsi komposit $F : (a,b) \to \Re$ yang diberikan oleh $F(t) = g(f(t))$ diferensiabel pada (a,b) dan

$$F'(t) = Dg(f(t))\, Df(t) = \nabla g(f(t))^T \begin{bmatrix} f_1'(t) \\ \vdots \\ f_n'(t) \end{bmatrix}$$

Bukti :

Dengan definisi: $F'(t) = \lim\limits_{h \to 0} \dfrac{F(t+h) - F(t)}{h} = \lim\limits_{h \to 0} \dfrac{g(f(t+h)) - g(f(t))}{h}$

jika limitnya ada.

Karena f kontinu dan diferensiabel, dengan menerapkan Teorema Nilai Rata-rata untuk g diperoleh : $g(y) - g(x) = Dg(x_0)(y-x)$, di mana x_0 adalah beberapa titik pada segmen x dan y. Misalnya $x = f(t)$ dan $y = f(t+h)$ maka :

$$\frac{F(t+h)-F(t)}{h} = \frac{Dg(x_0)f(t+h)-f(t)}{h}$$

Oleh kekontinuan dari Dg, $Dg(x_0) \to Dg(f(t))$ seperti $h \to 0$.

Jadi $F'(t) = \lim_{h\to 0} Dg(x_0) \dfrac{f(t+h)-f(t)}{h} = Dg(f(t))Df(t)$. ■

5. Deret Taylor

Teorema 2.4

Andaikan suatu fungsi $f : \Re \to \Re$ *terdeferensial ke-m dan kontinu, yakni* $f \in \wp^m$ *pada interval [a,b], maka*

$$f(b) = f(a) + \frac{(b-a)}{1!}f'(a) + \frac{(b-a)^2}{2!}f''(a) + \ldots + \frac{(b-a)^{m-1}}{(m-1)!}f^{(m-1)}(a) + R_m$$

dan $R_m = \dfrac{(b-a)^m(1-\theta)^{m-1}}{(m-1)!}f^{(m)}(a+\theta(b-a)) = \dfrac{(b-a)^m}{m!}f^{(m)}(a+\theta'(b-a))$

di mana $\theta, \theta' \in (0,1)$.

Bukti: Perhatikan bahwa

$$R_m = f(a) - f(b) - \frac{(b-a)}{1!}f'(a) - \frac{(b-a)^2}{2!}f''(a) - \ldots - \frac{(b-a)^{m-1}}{m-1!}f^{(m-1)}(a).$$

Andaikan $R_m = g_m$ (x) dengan x menggantikan a, maka :

$$g_m(x) = f(x) - f(b) \frac{(b-x)}{1!} f'(x) - \frac{(b-x)^2}{2!} f''(x) - \ldots - \frac{(b-x)^{m-1}}{(m-1)!} f^{(m-1)}(x).$$

$$g_m(b) = f(b) - f(b) \frac{(b-b)}{1!} f'(b) - \frac{(b-b)^2}{2!} f''(b) - \ldots - \frac{(b-b)^{m-1}}{(m-1)!} f^{(m-1)}(b).$$

$$g_m(b) = 0$$

$$g_m(a) = f(a) - f(b) - \frac{(b-a)}{1!} f'(a) - \frac{(b-a)^2}{2!} f''(a) - \ldots - \frac{(b-a)^{m-1}}{m-1!} f^{(m-1)}(a) = R_m.$$

Dengan menerapkan Nilai Rata-rata, maka terdapat paling sedikit satu bilangan c dalam (a,b) di mana $\frac{f(b)-f(a)}{b-a} = f'(c)$ maka diperoleh:

$\frac{g_m(b)-g_m(a)}{b-a} = g'_m(a+\theta(b-a))$ di mana $\theta \in (a,b)$. Yang ekuivalen dengan

$$\frac{-R_m}{b-a} = -\frac{(b-a-\theta(b-a))^{(m-1)}}{(m-1)!} f^{(m)}(a+\theta(b-a))$$

$$-R_m = \frac{-(b-a)^{m-1}(1-\theta)^{m-1}}{(m-1)!} f^{(m)}(a+\theta(b-a))$$

Jadi $\quad R_m = \frac{(b-a)^m(1-\theta)^{m-1}}{(m-1)!} f^m(a+\theta(b-a))$ ■

C. Optimisasi Fungsi Tanpa Kendala

Optimisasi merupakan suatu proses penentuan penyelesaian yang terbaik dari suatu masalah. Dalam masalah optimisasi fungsi, tujuannya adalah mengoptimumkan (memaksimumkan / meminimumkan) nilai suatu fungsi. Optimisasi fungsi tanpa kendala berarti mencari nilai x untuk mengoptimalkan fungsi tujuan atau fungsi obyektif tanpa adanya pembatas yang membatasi pencapaian nilai optimal dari fungsi itu. Menentukan Penyelesaian masalah optimisasi tanpa kendala untuk bentuk maksimum fungsi, ekuivalen dengan menentukan minimum dari negatif f $(-f)$ fungsi. Oleh karena itu, pada pembahasan selanjutnya akan selalu dianggap masalah optimisasi sebagai masalah minimisasi. Fungsi yang akan diminimumkan disebut fungsi tujuan atau fungsi obyektif, yakni fungsi $f: \Re^n \rightarrow \Re$ dengan nilai fungsi suatu bilangan real. Perumusan masalah optimisasinya adalah sebagai berikut :

Optimumkan $f(\mathbf{x})$

di mana $\mathbf{x} \in \Omega$.

Dalam hal ini, vektor $\mathbf{x}$ adalah suatu vektor berdimensi-n dari variabel-variabel bebas, yaitu $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ dengan $x_1, x_2, x_3, \ldots, x_n$ merupakan variabel-variabel keputusan. Himpunan Ω merupakan subset dari $\Re^n$ yang disebut himpunan kendala (*feasible set*), Masalah optimisasi di atas dapat dipandang sebagai suatu masalah keputusan yang meliputi penemuan vektor $\mathbf{x}$ 'terbaik' dari semua variabel yang mungkin dalam Ω. Vektor 'terbaik ' dimaksudkan sebagai suatu vektor yang membuat nilai fungsi obyektifnya optimum. Vektor demikian dinamakan pembuat minimum dari fungsi *f*.

Definisi 2.11 Pembuat minimum lokal

Andaikan $f:\Omega\to\Re$ *dengan* $\Omega\subset\Re^n$ *adalah suatu fungsi dengan nilai real yang didefinisikan pada suatu himpunan* $\Omega\subset\Re^n$ *.vektor* $\mathbf{x}^*\in\Omega$ *adalah pembuat minimum lokal f atas* Ω *jika terdapat* $\varepsilon>0$ *sedemikian sehingga* $f(\mathbf{x})\geq f(\mathbf{x}^*)$ *untuk setiap* $\mathbf{x}\in\Omega\setminus\{\mathbf{x}^*\}$ *dan* $\|\mathbf{x}-\mathbf{x}^*\|<\varepsilon$

Definisi 2.12 Pembuat minimum global

Suatu vektor $\mathbf{x}^*\in\Omega$ *adalah pembuat minimum global f atas* Ω *jika* $f(\mathbf{x})\geq f(\mathbf{x}^*)$ *untuk setiap* $\mathbf{x}\in\Omega\setminus\{\mathbf{x}^*\}$.

Untuk mendapatkan suatu vektor $\mathbf{x}^*$ pembuat minimum lokal digunakan derivatif fungsi $f:\Re^n\to\Re$. Derivatif pertama dari *f* dinyatakan oleh :

$$Df(\mathbf{x}) = \left[\frac{\delta f}{\delta x_1}(\mathrm{x}), \frac{\delta f}{\delta x_2}(\mathrm{x}), \ldots \frac{\delta f}{\delta x_n}(\mathrm{x})\right],$$

dan derivatif kedua dari f adalah :

$$\mathrm{D}^2 f(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Selanjutnya untuk menentukan bahwa suatu pembuat minimum fungsi terletak di sekitar Ω, diperlukan definisi berikut :

Definisi 2.13 Arah layak (Feasible direction)

Suatu vektor $\mathbf{d} \in \Re^n$, $\mathbf{d} \neq 0$, *adalah* arah layak *pada* $\mathbf{x} \in \Omega$ *jika terdapat* $\alpha_0 > 0$ *sedemikian sehingga* $\mathbf{x} + \alpha \mathbf{d} \in \Omega$ *untuk setiap* $\alpha \in [0, \alpha_0]$.

Beberapa teorema yang penting untuk kondisi minimum lokal, diberikan oleh teorema berikut :

Teorema 2.5 Syarat Perlu Tingkat Pertama (SPTP)

Andaikan $\Omega \subset \Re^n$ *dan* $f \subset \wp^1$ *suatu fungsi bernilai real pada* Ω. *Jika* $\mathbf{x}^*$ merupakan *pembuat minimum lokal* f *atas* Ω, *maka ada arah layak* $\mathbf{d}$ *pada* $\mathbf{x}^*$ sedemikian *sehingga* $\mathbf{d}^T \nabla f(\mathbf{x}^*) \geq 0$.

Bukti :

Perhatikan bahwa $\mathbf{x}(\alpha) = \mathbf{x}^* + \alpha\mathbf{d} \in \Omega$ (Definisi).

Untuk $\alpha = 0$, maka $\mathbf{x}(\alpha) = \mathbf{x}^* + 0\mathbf{d} = \mathbf{x}^*$.

Kemudian dibuat suatu fungsi $\phi(\alpha) = f(\mathbf{x}(\alpha))$. Dengan menggunakan teorema deret *Taylor* diperoleh :

$f(\mathbf{x}^* + \alpha\mathbf{d}) - f(\mathbf{x}^*) = \phi(\alpha) - \phi(0) = \phi'(0)\,\alpha + 0(\alpha) = \alpha\mathbf{d}^T\nabla f(\mathbf{x}(0)) + 0(\alpha)$,

dimana $\alpha \geq 0$. Jadi jika $\phi(\alpha) \geq 0$, maka $\mathbf{d}^T\nabla f(\mathbf{x}^*) \geq 0$. ■

Corollary 2.1 Masalah Interior

Andaikan $\Omega \subset \Re^n$ dan $f \subset \wp^1$ suatu fungsi bernilai real pada Ω. Jika $\mathbf{x}^*$ merupakan *pembuat minimum lokal f atas Ω dan jika* $\mathbf{x}^*$ *adalah suatu titik interior pada Ω, maka* $\nabla f(\mathbf{x}^*) = \mathbf{0}$.

Bukti :

Andaikan f mempunyai pembuat minimum lokal $\mathbf{x}^*$ yang merupakan titik interior pada Ω. Karena $\mathbf{x}^*$ titik interior pada Ω, himpunan arah layak pada $\mathbf{x}^*$ adalah seluruh $\Re^n$. Jadi terdapat $\mathbf{d} \in \Re^n$, $\mathbf{d}^T\nabla f(\mathbf{x}^*) \geq 0$ dan $-\mathbf{d}^T\nabla f(\mathbf{x}^*) \geq 0$. Sehingga $\nabla f(\mathbf{x}^*) = \mathbf{0}$. ■

Teorema 2.6 Syarat Perlu Tingkat Kedua (SPTK) *Masalah Interior*

Andaikan $\Omega \in \Re^n$, $f \in \wp^2$ *suatu fungsi pada* Ω, $\mathbf{x}^*$ *adalah pembuat minimum lokal pada* f *atas* Ω, *dan* $\mathbf{d}$ *suatu arah layak pada* $\mathbf{x}^*$. *Jika* $\mathbf{d}^T \nabla f(\mathbf{x}^*) = 0$, *maka*

$\mathbf{d}^T \mathbf{F}(\mathbf{x}^*)\, \mathbf{d} \geq 0$, *dimana F adalah suatu matriks Hessian pada f.*

Bukti:

Andaikan terdapat suatu arah layak $\mathbf{d}$ pada $\mathbf{x}^*$ sedemikian sehingga $\mathbf{d}^T \nabla f(\mathbf{x}^*) = 0$ dan $\mathbf{d}^T \mathbf{F}(\mathbf{x}^*)\, \mathbf{d} < 0$.

Diketahui $\mathbf{x}(\alpha) = \mathbf{x}^* + \alpha \mathbf{d}$ dan $\phi(\alpha) = f(\mathbf{x}^* + \alpha \mathbf{d}) = f(\mathbf{x}(\alpha)$. Dengan bentuk *Taylor* diperoleh :

$\phi(\alpha) = \phi(0) + \phi''(0)\dfrac{\alpha^2}{2} + 0(\alpha^2)$, dengan asumsi bahwa $\phi'(0) = \mathbf{d}^T \nabla f(\mathbf{x}^*) = 0$, dan $\phi''(0) = \mathbf{d}^T \mathbf{F}(\mathbf{x}^*)\, \mathbf{d} < 0$. Untuk nilai yang cukup kecil α,

$\phi(\alpha) - \phi(0) = \phi''(0)\dfrac{\alpha^2}{2} + 0(\alpha^2) < 0$, yaitu $f(\mathbf{x}^* + \alpha \mathbf{d}) < f(\mathbf{x}^*)$, terjadi

kontradiksi dengan asumsi bahwa $\mathbf{x}^*$ adalah pembuat minimum lokal. Jadi $\phi''(0) = \mathbf{d}^T \mathbf{F}(\mathbf{x}^*)\, \mathbf{d} \geq 0$. ■

Corollary 2.2 Masalah Interior

Andaikan $\mathbf{x}^*$ *adalah suatu titik interior pada* $\Omega \subset \Re^n$, *jika* $\mathbf{x}^*$ *adalah pembuat minimum lokal dari* $f : \Omega \to \Re$, $f \in \wp^2$ *maka* $\nabla f(\mathbf{x}^*) = 0$ *dan* $F(\mathbf{x}^*) \geq 0$, *yakni untuk setiap* $\mathbf{d} \in \Re^n$, $\mathbf{d}^T F(\mathbf{x}^*)\, \mathbf{d} \geq 0$.

Bukti:

Jika $\mathbf{x}^*$ adalah suatu titik interior, maka menurut *corollary* 2.1 dan teorema 2.6 semua merupakan arah layak. ∎

Teorema 2.7 Syarat Cukup Tingkat Kedua (SCTK) *Masalah Interior*

Andaikan $f \in \wp^2$ *didefinisikan pada daerah dalam* $\mathbf{x}^*$ *yang merupakan titik interior. Andaikan bahwa* $\nabla f(\mathbf{x}^*) = 0$; *dan* $f(\mathbf{x}^*) > 0$. *Maka* $\mathbf{x}^*$ *adalah pembuat minimum lokal dari* f.

Bukti :

Karena $\wp^2$, $\mathbf{F}(\mathbf{x}^*) = \mathbf{F}^T(\mathbf{x}^*)$. Dengan menggunakan asumsi $f(\mathbf{x}^*) > 0$ dan menurut ketaksamaan *Rayleigh* (Chong & Zak, 1996 : 36) bahwa jika $\mathbf{d} \neq 0$, maka $0 < \lambda_{\min}(\mathbf{F}(\mathbf{x}^*))\|\mathrm{d}\|^2 \leq \mathbf{d}^T\ \mathbf{F}(\mathbf{x}^*)\mathbf{d}$. Dari teorema deret *Taylor* diperoleh :

$$f(\mathbf{x}^* + \mathbf{d}) - f(\mathbf{x}^*) = \frac{1}{2}\mathbf{d}^T\mathbf{F}(\mathbf{x}^*)\mathbf{d} + 0(\|\mathbf{d}\|^2) \geq \frac{\lambda_{mi}(F(x^*))}{2}\|\mathbf{d}\|^2 + 0(\|\mathbf{d}\|^2),$$

sehingga dengan asumsi di atas $\nabla f(\mathbf{x}^*) = 0$. Untuk semua $\mathbf{d}$ sedemikian sehingga $\|\mathbf{d}\|$ adalah cukup kecil, $f(\mathbf{x}^* + \mathbf{d}) > f(\mathbf{x}^*)$. ■

BAB III

METODE *STEEPEST DESCENT*

Pada bab ini sebelum pembahasan tentang metode S*teepest Descent*, terlebih dahulu akan dibahas tentang metode *Newton* dan metode *Secant*.

A. Metode *Newton*

Metode *Newton* merupakan suatu metode yang didasarkan pada pendekatan fungsi $f(x)$ dengan fungsi yang lebih sederhana. Fungsi tersebut dicari minimumnya, kemudian minimum dari fungsi tersebut digunakan sebagai pendekatan dari minimum $f(x)$. Fungsi paling sederhana yang dapat dipakai pendekatan $f(x)$ adalah fungsi kuadrat, karena fungsi kuadrat mempunyai minimum. Fungsi kuadrat mempunyai bentuk: $q(x) = f(x^{(k)}) + f'(x^{(k)})(x-x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x-x^{(k)})^2$.

Maka sebagai pengganti dari minimum f, diminimumkan aproksimasi q. Dengan menggunakan syarat perlu tingkat pertama (SPTP) untuk suatu minimum dari q menghasilkan :

$0 = q'(x) = f'(x^{(k)}) + f''(x^{(k)})(x-x^{(k)})$. Ambil $x = (x^{(k+1)})$ maka:

$0 = q'(x^{(k+1)}) = f'(x^{(k)}) + f''(x^{(k)})(x^{(k+1)}-x^{(k)})$.

$\Leftrightarrow f'(x^{(k)}) + f''(x^{(k)})(x^{(k+1)}-x^{(k)}) = 0$

$f''(x^{(k)})(x^{(k+1)}-x^{(k)}) = -f'(x^{(k)})$

$$x^{(k+1)} - x^{(k)} = \frac{-f'(x^{(k)})}{f''(x^{(k)})}$$

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

Andaikan $x^{(k)}$ adalah pendekatan x^*, maka ekspansi deret *Taylor* fungsi f di sekitar $x^{(k)}$ adalah: $f(x^{(k)} + (x-x^{(k)}) = f(x^{(k)}) + f'(x^{(k)}(x-x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x-x^{(k)})^2 + \ldots$

Minimum fungsi f dapat didekati dengan ekspansi deret *Taylor* tersebut, sehingga diperoleh:

$$f(x^*) = \min_{x} f(x)$$

$$= \min_{x-x^{(k)}} f(x^{(k)} + (x-x^{(k)})$$

$$= \min_{x-x^{(k)}} [f(x^{(k)} + f'(x^{(k)}(x-x^{(k)}) + \tfrac{1}{2} f''(x^{(k)})(x-x^{(k)})^2 + \ldots]$$

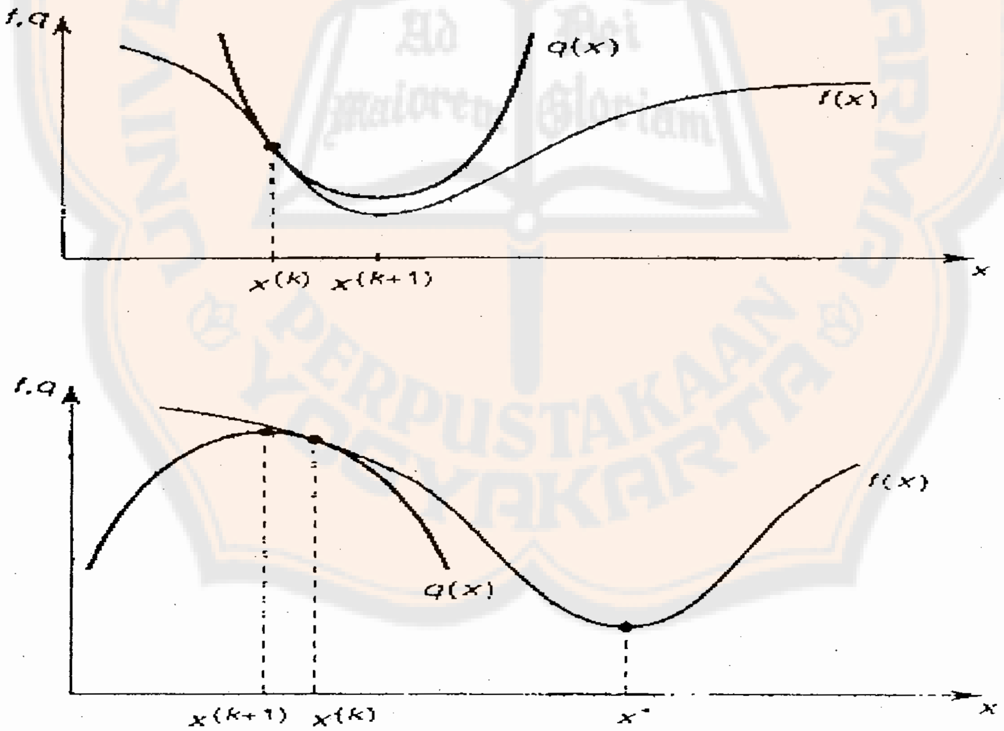
$$\approx \min_{x-x^{(k)}} [f(x^{(k)} + f'(x^{(k)}(x-x^{(k)}) + \tfrac{1}{2} f''(x^{(k)})(x-x^{(k)})^2]$$

Karena $f(x)$ didekati oleh fungsi kuadrat dalam $(x-x^{(k)})$ maka suku-suku berorde lebih tinggi dapat diabaikan. Untuk meminimumkan $f(x^{(k)} + f'(x^{(k)}(x-x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x-x^{(k)})^2$ diturunkan terhadap $(x-x^{(k)})$ diperoleh: $f'(x^{(k)}) + f''(x^{(k)})(x-x^{(k)}) = 0$ atau

$$(x-x^{(k)}) = \frac{-f'(x^{(k)})}{f''(x^{(k)})}$$

$(x-x^{(k)})$ adalah langkah pendekatan yang akan membawa $x^{(k)}$ menuju x^* sehingga $x^* = x^{(k)} + (x-x^{(k)})$. Metode *Newton* menggunakan pendekatan ini sebagai rumus iteratif, sehingga diperoleh rumus iteratif : $x^{(k+1)} = x^{(k)} + (x-x^{(k)}) = x^{(k)} - \dfrac{f'(x^{(k)})}{f''(x^{(k)})}$

Pada metode *Newton*, untuk menjamin bahwa hasil yang diperoleh adalah minimum maka $f''(x) > 0$ untuk semua x (lihat gambar 1 bagian atas). Namun jika $f''(x) \leq 0$ metode *Newton* gagal menemukan pembuat minimum, sebab jika $f''(x) < 0$ maka hasil yang diperoleh adalah maksimum (lihat gambar 1 bagian bawah), sedangkan jika $f''(x) = 0$ berarti f mempunyai ekstrim lebih dari satu.



Gambar 1

Contoh :

Akan digunakan metode *Newton* untuk memperbaiki suatu aproksimasi pertama $x^{(0)}$ = 12, akar dari persamaan $g(x) = x^3 - 12.2x^2 + 7.45x + 42 = 0$.

$g'(x) = 3x^2 - 24.4x + 7.45$.

Dengan melakukan dua iterasi diperoleh:

$$x^{(1)} = 12 - \frac{102.6}{146.65} = 11.33$$

$$x^{(2)} = 11.33 - \frac{14.73}{116.11} = 11.21$$

B. Metode *Secant*

Dari Pembahasan metode *Newton* di atas, dapat diketahui bahwa penentuan minimisasi fungsi f membutuhkan derivatif kedua dari fungsi f tersebut, yakni :

$$x^{(k+1)} = x^{(k)} - \frac{f'\left(x^{(k)}\right)}{f''\left(x^{(k)}\right)}.$$

Jika derivatif kedua dari fungsi tersebut tidak ada, maka dapat diusahakan dengan menghampiri $f''(x^{(k)})$ menggunakan derivatif pertama dari fungsi tersebut. Jadi diusahakan $f''(x^{(k)})$ dihampiri dengan $\dfrac{\left(x^{(k)}\right) - f'\left(x^{(k-1)}\right)}{x^{(k)} - x^{(k-1)}}$ sehingga diperoleh:

$$x^{(k+1)} = x^{(k)} - \frac{f'\left(x^{k}\right)}{\dfrac{f'\left(x^{(k)}\right) - f'\left(x^{(k-1)}\right)}{x^{(k)} - x^{(k-1)}}}$$

$x^{(k+1)} = x^{(k)} - \dfrac{x^{(k)} - x^{(k-1)}}{f'\left(x^{(k)}\right) - f'\left(x^{(k-1)}\right)} f'(x^{(k)})$. Rumus inilah yang dinamakan *rumus metode Secant.*

Algoritma metode *Secant* membutuhkan dua titik awal untuk memulai melakukan iterasi, yang dinyatakan dengan $x^{(-1)}$ dan $x^{(0)}$. Metode *Secant* disajikan dalam bentuk yang ekuivalen, yakni:

$$x^{(k+1)} = \frac{-f'\left(x^{(k)}\right)x^{(k-1)} - f'\left(x^{(k-1)}\right)x^{(k)}}{f'\left(x^{(k)}\right) - f'\left(x^{(k-1)}\right)}$$

Sebagaimana metode *Newton*, metode *Secant* tidak langsung memasukkan nilai-nilai dari $f(x^{(k)})$. Sebagai pengganti diusahakan mengendalikan $f'(x^{(k)})$. Cara pengerjaan menggunakan metode *Newton* tidak berbeda dengan cara pengerjaan menggunakan metode *Secant* dalam hal metode *Secant* diintepretasikan sebagai suatu algoritma untuk menyelesaikan persamaan yang berbentuk $g(x) = 0$. Dengan spesifikasi bahwa algoritma metode *Secant* untuk menemukan suatu akar persamaan $g(x) = 0$ mempunyai bentuk : $x^{(k+1)} = x^{(k)} - \dfrac{(b-a)^m(1-\theta)^{m-1}}{(m-1)!} g(x^{(k)})$, atau ekuivalen dengan :

$$x^{(k+1)} = -\frac{g\left(x^{(k)}\right)x^{(k-1)} - g\left(x^{(k-1)}\right)x^{(k)}}{g\left(x^{(k)}\right) - g\left(x^{(k-1)}\right)}$$

Metode *Newton* menggunakan kemiringan (*slope*) dari g untuk menentukan titik berikutnya, sedangkan metode *Secant* menggunakan titik '*Secant*' di antara titik ke-(k-1) dan titik ke-k untuk menentukan titik ke-(k+1) .

C. Metode *Steepest Descent*

Metode *Steepest Descent* merupakan suatu algoritma gradien dimana setiap langkah bertujuan untuk memilih pencapaian jumlah maksimum berkurangnya nilai fungsi obyektif. Metode tersebut digunakan untuk mencari minimum suatu fungsi, yakni dengan menggunakan nilai negatif dari gradien fungsi di suatu titik. Secara singkat, proses pencarian minimum fungsi f atau $\mathbf{x}^*$ adalah sebagai berikut: mula-mula diberikan titik awal $\mathbf{x}^{(0)} \in \Re^n$, maka dicari α_0 yang meminimumkan fungsi $f(x^{(0)}-\alpha_0\nabla f(x^{(0)})$. Setelah itu dicari $x^{(1)}$ dengan rumus $x^{(1)} = x^{(0)}-\alpha_0\nabla f(x^{(0)})$. Dari $x^{(1)}$ ini kembali dicari α_1 yang meminimumkan fungsi $f(x^{(1)}-\alpha_1\nabla f(x^{(1)}))$ dan dicari $x^{(2)}$ dengan rumus $x^{(2)}=x^{(1)}-\alpha_1\nabla f(x^{(1)})$. Demikian seterusnya sehingga secara umum diperoleh rumus untuk $x^{(k)}$ sebagai berikut: $x^{(k+1)} = x^{(k}-\alpha_k\nabla f(x^{(k)})$ dengan $\alpha_k > 0$ dan memenuhi $f(x^{(k)}-\alpha_k-\nabla f(x^{(k)})) = \min f(x^{(k)}-\alpha_k\nabla f(x^{(k)}))$. Proses pencarian minimum fungsi f ini berhenti jika $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$ untuk suatu ε bilangan positif yang kecil. Algoritma metode *Steepest Descent* adalah sebagai berikut:

i. Tentukan titik awal $\mathbf{x}^{(0)}$, ε dan ambil $k = 0$

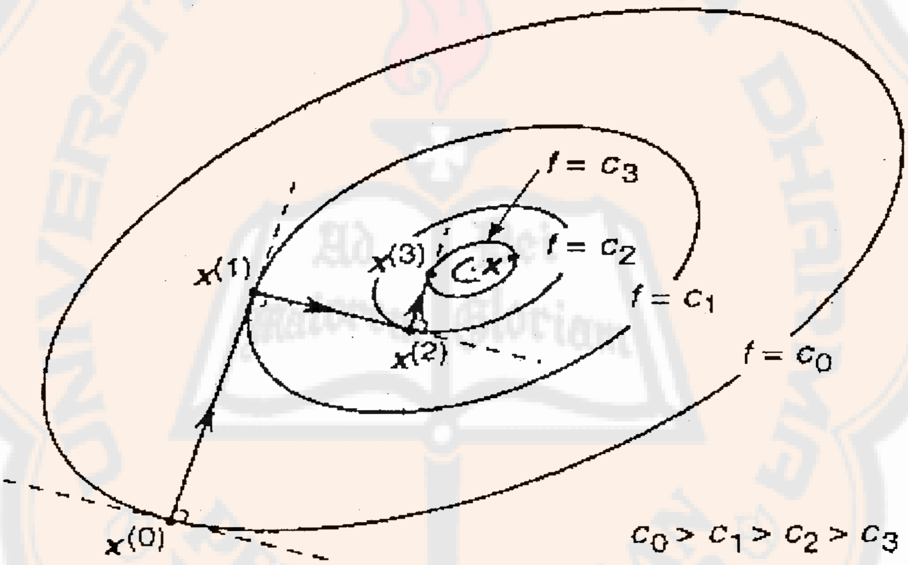
ii. Dihitung gradien dari $f(\mathbf{x})$ pada titik $\mathbf{x}^{(k)}$,yaitu $\nabla f(\mathbf{x}^{(k)})$. Kemudian dihitung $\|\nabla f(\mathbf{x}^{(k)})\|$. Jika $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$, maka iterasi dihentikan dan dengan demikian $\mathbf{x}^* = \mathbf{x}^{(k)}$ sebagai titik minimum. Jika sebaliknya, maka dilanjutkan ke langkah iii.

iii. Dimisalkan arah pencarian pada titik $\mathbf{x}^{(k)}$ adalah $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$.

iv. Dihitung α_k untuk meminimumkan $f(\mathbf{x}^{(k)} + \alpha_k\mathbf{d}^{(k)})$.

v. Dihitung $\mathbf{x}^{(k+1)}$ dimana $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{d}^{(k)}$ kemudian dilanjutkan ke langkah ii dan seterusnya hingga suatu pembuat minimum $\mathbf{x}^{(k+1)}$ ditemukan.

Suatu tipe urutan hasil dari metode *Steepest Descent* diilustrasikan dalam gambar 2 di bawah ini:



Gambar 2

Dari gambar di atas nampak bahwa metode *Steepest Descent* bergerak dalam langkah-langkah orthogonal sebagaimana dinyatakan dalam proposisi di bawah ini.

Proposisi 1

Jika $\left\{x^{(k)}\right\}_{k=0}^{\infty}$ *adalah suatu urutan Steepest Descent dari fungsi* $f : \Re^n \rightarrow \Re$ *yang diberikan, maka untuk setiap k dari vektor* $\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ *adalah ortogonal ke vektor* $\mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)}$.

Bukti :

Dari formula iteratif metode *Steepest Descent*, diketahui bahwa: $\langle \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, \mathbf{x}^{(k+2)} - \mathbf{x}^{(k+1)} \rangle = \alpha_k \alpha_{k+1} \langle \nabla f(\mathbf{x}^{(k)}), \nabla f(\mathbf{x}^{(k+1)}) \rangle$. Untuk melengkapi bukti tersebut cukup ditunjukkan bahwa $\langle \nabla f(\mathbf{x}^{(k)}), \nabla f(\mathbf{x}^{(k+1)}) \rangle = 0$. Perhatikanlah bahwa α_k adalah suatu skalar tak negatif yang meminimumkan $\Phi_k(\alpha) \stackrel{\Delta}{=} f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}))$. Sehingga dengan menggunakan SPTP dan aturan rantai, diperoleh:

$$
\begin{aligned}
0 &= \Phi'_k(\alpha_k) \\
&= \frac{d_k}{d\alpha}(\alpha_k) \\
&= \nabla f(\mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}))^T(-\nabla f(\mathbf{x}^{(k)})) \\
&= -\langle \nabla f(\mathbf{x}^{(k+1)}), \nabla f(\mathbf{x}^{(k)}) \rangle. \blacksquare
\end{aligned}
$$

Proposisi di atas mengimplikasikan bahwa $\nabla f(\mathbf{x}^{(k)})$ sejajar ke bidang singgung untuk level himpunan $\{f(\mathbf{x}) = f(\mathbf{x}^{(k+1)})\}$ pada $\mathbf{x}^{(k+1)}$. Pada saat titik yang baru diteruskan oleh algoritma *Steepest Descent*, nilai korespondensi dalam fungsi berkurang sebagaimana dinyatakan dalam proposisi berikut.

Proposisi 2

Jika $\{x^{(k)}\}_{k=0}^{\infty}$ *adalah suatu urutan Steepest Descent dari fungsi* $f: \Re^n \to \Re$ *dan jika* $\nabla f(\mathbf{x}^{(k)}) \neq 0$ *maka* $(f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$.

Bukti:

Ingat kembali bahwa $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})$, dimana $\alpha_k \geq 0$ adalah pembuat minimum dari $\phi_k(\alpha) = f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}))$ untuk semua $\alpha \geq 0$. Untuk $\alpha \geq 0$ maka $\phi_k(\alpha_k) \leq \phi_k(\alpha)$. Dengan aturan rantai diperoleh:

$$\Phi'_k(0) = \frac{d\phi_k}{d\alpha}(0) = -(\nabla f(\mathbf{x}^{(k)} - 0\nabla f(\mathbf{x}^{(k)}))^T \nabla f(\mathbf{x}^{(k)}) = -\|\nabla f(\mathbf{x}^{(k)})\|^2 < 0,$$

karena diasumsikan bahwa $\nabla f(\mathbf{x}^{(k)}) \neq 0$.

Kemudian $\Phi'_k(0) < 0$, dan ini mengimplikasikan bahwa terdapat suatu $\bar{\alpha} > 0$ sedemikian sehingga $\Phi_k(0) > \Phi_k(\alpha)$ untuk semua $\alpha \in (0, \bar{\alpha}]$ sehingga

$f(\mathbf{x}^{(k+1)}) = \Phi_k(\alpha_k) \leq \Phi_k(\bar{\alpha}) < \Phi_k(0) = f(\mathbf{x}^{(k)})$. ■

Dalam proposisi di atas, digunakan asumsi $\nabla f(\mathbf{x}^{(k)}) \neq 0$ guna membuktikan proses algoritma dari satu sifat *descent*, yaitu $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ jika $\nabla f(\mathbf{x}^{(k)}) \neq 0$. Jika untuk beberapa k, diperoleh $\nabla f(\mathbf{x}^{(k)}) = 0$, maka titik $\mathbf{x}^{(k)}$ memenuhi SPTP. Dalam kasus ini $\mathbf{x}^{(k)} = \mathbf{x}^{(k+1)}$ dapat digunakan sebagai kriteria penghentian dari algoritma.

Kondisi $\nabla f(\mathbf{x}^{(k+1)}) = 0$ kerap kali tidak dengan tepat dapat dipenuhi karena perhitungan secara numeris dari gradien akan jarang tepat sama dengan nol. Dalam kasus demikian maka sebagai kriteria penghentian adalah dengan mengecek norma gradien, yakni jika norma $\| \nabla f(\mathbf{x}^{(k)}) \|$ dari gradien adalah tidak lebih besar dari suatu toleransi yang diberikan maka iterasi dihentikan. Sebagai alternatif dapat juga dilakukan dengan menghitung perbedaan nilai mutlak $| f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) |$ di antara nilai fungsi obyektif untuk setiap dua iterasi berturut-turut, jika perbedaannya tidak lebih besar dari toleransi yang diberikan maka perhitungan dihentikan. Sebagai alternatif lain adalah dengan menghitung norma $\| (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) \|$ dari perbedaan di antara 2 iterasi berturut-turut dan perhitungan dihentikan jika norma lebih kecil dari toleransi yang diberikan.

Contoh 1:

Diberikan suatu fungsi $f(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1x_2$ dengan titik awal $[0\ 1]^T$. Dengan menggunakan metode *Steepest Descent* akan dicari pembuat minimum dari fungsi tersebut.

Penyelesaian:

Untuk menyelesaikan masalah di atas, kita ikuti langkah-langkah dari algoritma *Steepest Descent*.

i. $\mathbf{x}^{(0)} = [0\ 1]^T$

ii. $\nabla f(\mathbf{x}^{(0)}) = [2x_1 - 2x_2,\ 2x_2 - 2x_1]^T$

$= [2(0) - 2(1),\ 2(1) - 2(0)]^T$

$= [-2, 2]^T$

$\|\nabla f(\mathbf{x}^{(0)})\| = \sqrt{(-2)^2 + 2^2}$

$= \sqrt{8}$

$= 2\sqrt{2} \neq 0$

iii. $\mathbf{d}^{(0)} = -\nabla f(\mathbf{x}^{(0)}) = -[-2, 2]^T = [2, -2]^T$

iv. Dihitung α_0 untuk meminimumkan $f(\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)})$ di mana :

$\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)} = [0, 1]^T + \alpha_0 [2, -2]^T$

$= [0, 1]^T + [2\alpha_0, -2\alpha_0]^T$

$= [2\alpha_0, 1-2\alpha_0]^T$

$f(\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)}) = (2\alpha_0)^2 + (1-2\alpha_0)^2 - 2(2\alpha_0)(1-2\alpha_0)$

$= 4\alpha_0^2 + 1 - 4\alpha_0 + 4\alpha_0^2 - 4\alpha_0 + 8\alpha_0^2$

$= 16\alpha_0^2 - 8\alpha_0 + 1 \equiv f(\alpha_0)$

Disyaratkan bahwa $\nabla f(\alpha_0) = 0$, sehingga :

$\frac{df(\alpha_0)}{d(\alpha_0)} = 0 \iff 32\alpha_0 - 8 = 0$

$32\alpha_0 = 8$

$\alpha_0 = 0{,}25.$

$\dfrac{d^2(\alpha_0)}{d\alpha_0^2} = 32 > 0$. Oleh karena itu kondisi cukup untuk minimum dipenuhi.

v. Dihitung kembali ($\mathbf{x}^{(0)} + \alpha_0\,\mathbf{d}^{(0)}$), diperoleh:

$x_1^{(1)} = 0 + 0.25\ (2) = 0.5$

$x_2^{(1)} = 1 + 0.25\ (-2) = 0.5$

$\therefore \mathbf{x}^{(1)} = [0.5,\ 0.5]^T$

$\nabla f(\mathbf{x}^{(1)}) = [2x_1 - 2x_2,\ 2x_2 - 2x_1]^T$

$= [2(0.5) - 2(0.5),\ 2(0.5) - 2(0.5)]^T$

$= [1 - 1,\ 1 - 1]^T$

$= [0,\ 0]^T$

$\nabla f(\mathbf{x}^{(1)}) = [0,\ 0]^T$ sudah memenuhi kriteria penghentian dan dengan demikian $\mathbf{x}^{(1)} = [0.5,\ 0.5]^T$ merupakan pembuat minimum dari fungsi yang diberikan.

Contoh 2:

Diberikan suatu fungsi $f(\mathbf{x}) = (x_1-4)^4 + (x_2-3)^2 + 4(x_3+5)^4$. Dengan titik awal $[4,\ 2,\ -1]^T$. Dengan menggunakan metode *Steepest Descent* akan dicari pembuat minimum dari fungsi tersebut.

Penyelesaian:

Iterasi pertama:

i. $\mathbf{x}^{(0)} = [4, 2, -1]^T$

ii. $\nabla f(x^{(0)}) = [4(x_1-4)^3, 2(x_2-3), 16(x_3+5)^3]^T$

$= [4(4 - 4)^3, 2(2-3), 16(-1+5)^3]^T$

$= [0, -2, 1024]^T$

$\|\nabla f(x^{(0)})\| = \sqrt{0^2 + (-2)^2 + 1024^2}$

$= \sqrt{1048580} = 1024.001953 \neq 0$

iii. $\mathbf{d}^{(0)} = -\nabla f(x^{(0)}) = [0, 2, -1024]^T$

iv. Dihitung α_0 untuk meminimumkan $f(\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)})$ di mana :

$\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)} = [4, 2, -1]^T + \alpha_0 [0, 2, -1024]^T$

$= [4, 2, -1]^T + [0, 2\alpha_0, -1024\alpha_0]^T$

$= [4, 2+2\alpha_0, -1-1024\alpha_0]^T$

$f(\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)}) = (4-4)^4 + (2+2\alpha_0-3)^2 + 4(-1-1024\alpha_0+5)^4$

$= (0+(2+2\alpha_0 - 3)^2 + 4(-1-1024\alpha_0 + 5)^4$

$= (2+2\alpha_0 - 3)^2 + 4(-1-1024\alpha_0 + 5)^4 \equiv f(\alpha_0)$

Disyaratkan bahwa $\nabla f(\alpha_0) = 0$, sehingga :

$$\frac{df(\alpha_0)}{d(\alpha_0)} = 0 \quad \Leftrightarrow \quad 2(2+2\alpha_0 - 3)2 + 16(-1-1024\alpha_0 + 5)\,1024 = 0$$

$$8+8\alpha_0 - 12 - 16384 - 16777216\alpha_0 + 81920 = 0$$

$$65532 - 16777208\alpha_0 = 0$$

$$65532 = 16777208\alpha_0$$

$$\alpha_0 = \frac{65532}{16777208}$$

$$\alpha_0 = 3,906.\ 10^{-3}$$

$$\frac{d^2(\alpha_0)}{d\alpha_0^2} = 8 - 16777216 = -16777208 \ < 0.$$

v. Dihitung kembali ($\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)}$), diperoleh:

$x_1^{(1)} = 4 + 3.906.10^{-3}(0) = 4$

$x_2^{(1)} = 2 + 3.906.10^{-3}(2) = 2.008$

$x_3^{(1)} = -1 + 3.906.10^{-3}(-1024) = -4.999$

$\therefore \mathbf{x}^{(1)} = [4, 2.008, -4.999]^T$

Iterasi kedua:

i. $\mathbf{x}^{(1)} = [4, 2.008, -4.999]^T$

ii. $\nabla f(x^{(1)}) = [4(4\text{-}4)^3, 2(2.008\text{-}3), 16(-4.999+5)^3]^T$

$= [0,\ -1,984,\ 0,000000016]^T$

$$\| \nabla f(x^{(1)}) \| = \sqrt{0^2 + (-1.984)^2 + 0.000000016^2}$$

$$= \sqrt{3.936256} = 1.984 \neq 0$$

iii. $\mathbf{d}^{(1)} = -\nabla f(x^{(1)}) = [0, 1.984, -0.000000016]^T$

iv. Dihitung α_1 untuk meminimumkan $f(\mathbf{x}^{(1)} + \alpha_1 \mathbf{d}^{(1)})$ di mana :

$$\mathbf{x}^{(1)} + \alpha_1 \mathbf{d}^{(1)} = [4, 2.008, -4.999]^T + \alpha_1 [0, 1.984, -0.000000016]^T$$

$$= [4, 2.008, -4.999]^T + [0, 1.984\alpha_1, -0.000000016\alpha_1]^T$$

$$= [4, 2.008+1.984\alpha_1 , -4.999-0.000000016]^T \equiv f(\alpha_1)$$

Disyaratkan bahwa $\nabla f(\alpha_1) = 0$, sehingga :

$$\frac{df(\alpha_1)}{d(\alpha_1)} = 0$$

$$\Leftrightarrow 2(2.008+1.984\alpha_1-3)1.984 + 16(-4.999-0.000000016\alpha_1 +5)$$

$$(-0.000000016) = 0$$

$$7.967744 + 7.872512\alpha_1 - 11.904 + 1.279744.10^{-6} +$$

$$4.096.10^{-15}\alpha_1 \; -1.28.10^{-6} = 0$$

$$7.872512\alpha_1 - 3.936256 = 0$$

$$7.872512\alpha_1 = 3.936256$$

$$\alpha_1 = 0.5$$

$$\frac{d^2 f(\alpha_1)}{d\alpha_1^2} = 7.872512 > 0$$

v. Dihitung kembali ($\mathbf{x}^{(1)} + \alpha_1 \mathbf{d}^{(1)}$), diperoleh:

$x_1^{(2)} = 4 + 0.5(0) = 4$

$x_2^{(2)} = 2.008 + 0.5(1.984) = 3$

$x_3^{(2)} = -4.999 + 0.5(-0.000000016) = -4.999$

$\therefore \mathbf{x}^{(2)} = [4, 3, -4.999]^T$

Iterasi ketiga:

i. $\mathbf{x}^{(2)} = [4, 3, -4.999]^T$

ii. $\nabla f(\mathbf{x}^{(2)}) = [4(4-4)^3, 2(3-3), 16(-4.999+5)^3]^T$

$= [0, \; 0, \; 0.000000016]^T$

$$\|\nabla f(x^{(2)})\| = \sqrt{0^2 + 0^2 + 0.000000016^2}$$

$$= \sqrt{0.000000016^2} = 0.000000016$$

iii. $\mathbf{d}^{(2)} = -\nabla f(x^{(2)}) = [0, 0, -0.000000016]^T$

iv. Dihitung α_2 untuk meminimumkan $f(\mathbf{x}^{(2)} + \alpha_2 \mathbf{d}^{(2)})$ di mana :

$$\mathbf{x}^{(2)} + \alpha_2 \mathbf{d}^{(2)} = [4, 3, -4.999]^T + \alpha_2 [0, 0, -0.000000016]^T$$

$$= [4, 3, -4.999]^T + [0, 0, -0.000000016\alpha_2]^T$$

$$= [4, 3, -4.999 - 0.000000016\alpha_2]^T \equiv f(\alpha_2)$$

Disyaratkan bahwa $\nabla f(\alpha_2) = 0$, sehingga :

$$\frac{df(\alpha_2)}{d(\alpha_2)} = 0$$

$$\Leftrightarrow 16(-4.999 - 0.000000016\alpha_2 + 5)(-0.000000016) = 0$$

$$1.279744.10^{-6} + 4.096.10^{-15}\alpha_2 - 1.28.10^{-6} = 0$$

$$-2.56.10^{-10} + 4.096.10^{-15}\alpha_2 = 0$$

$$4.096.10^{-15}\alpha_2 = 2.56.10^{-10}$$

$$\alpha_2 = 6.25.10^{-26}$$

$$\frac{d^2 f(\alpha_2)}{d\alpha_2^2} = 4.096.10^{-15} > 0$$

v. Dihitung kembali $(\mathbf{x}^{(2)} + \alpha_2 \mathbf{d}^{(2)})$, diperoleh:

$$x_1^{(3)} = 4 + 6.25.10^{-26}(0) = 4$$

$$x_2^{(3)} = 3 + 6.25.10^{-26}(0) = 3$$

$x_3^{(3)} = -4.999 + 6.25.10^{-26}(-0.000000016) = -4.999$

$\therefore \mathbf{x}^{(3)} = [4, 3, -4.999]^T$

Ternyata setelah melakukan tiga iterasi diperoleh:

$\mathbf{x}^{(2)} = \mathbf{x}^{(3)} = [4, 3, -4.999]^T$ dan $\nabla f(\mathbf{x}^{(2)}) = \nabla f(\mathbf{x}^{(3)}) = [0,\ 0,\ 0.000000016]^T$

$\approx [0,0,0]^T$. Hal ini memenuhi kriteria penghentian dan dengan demikian $\mathbf{x}^{(3)}=[4,\ 3,\ -4.999]^T$ merupakan pembuat minimum dari fungsi yang diberikan.

D. Program MATLAB Untuk Metode *Steepest Descent*

Program merupakan sekumpulan perintah yang disusun untuk suatu tujuan tertentu. Dalam skripsi ini penyusunan program bertujuan untuk membantu melakukan perhitungan dalam menemukan pembuat minimum fungsi dengan menggunakan metode *Steepest Descent* sehingga perhitungan menjadi lebih efektif dan efisien. Penulisan program MATLAB dapat dilakukan pada *Command Windows* atau pada *windows* lain pada MATLAB yang disebut dengan *Windows M-file* (MATLAB Editor / debugger). Program–program yang sederhana atau beberapa fungsi yang sudah disediakan oleh MATLAB akan lebih mudah diselesaikan pada *Command Windows.* Sedangkan program yang cukup panjang atau melibatkan beberapa sub program akan lebih mudah diselesaikan pada *Windows M-File.* Program dalam MATLAB sering dinamakan *Script file* atau *M-File* dan mempunyai eksistensi - M. Berikut ini diberikan program MATLAB untuk metode *Steepest*

Descent, program ditulis dan disimpan pada *Windows M-File*. Pemanggilan program dilakukan pada *Command Windows*.

```
%Program Utama Untuk Metode Steepest Descent:
function res=mincg(f,derf,ftau,x,tol)
global p1 d1
n=size(x);noiter=0;
%Menghitung gradien awal
df=feval(derf,x);
%Lup utama
while norm(df)>tol
   noiter=noiter+1;
   df=feval(derf,x);d1=df;
   %Lup bagian dalam.
   for inner=1:n
      p1=x;
      tau=fmin(ftau,-2,2,[0.001]);
      %Menghitung x baru
      x1=x-tau*d1;
      %Menyimpan gradien sebelumnya
      dfp=df;
      %Menghitung gradien baru
      df=feval(derf,x1);
      %x dan d terbaru
      d=d1; x=x1;
      %Metode Steepest Descent
      beta=(df'*df)/(dfp'*dfp);
      d1=-df+beta*d;
   end;
end;
res=x1;
disp('Solusi x=');
disp('Gradien:');disp(df);
disp('Banyak iterasi :');disp(noiter);
```

Perhatikan kembali contoh soal di atas untuk contoh 1, jika soal tersebut diselesaikan dengan program MATLAB, maka pemanggilan program dilakukan dalam *Command Windows* dengan masukan sebagai berikut:

x0=[0 1]'

Toleransi = 0.000005

```
function fv=fcth1(x);
fv=x(1)^2+x(2)^2-2*x(1)*x(2);

function pd=fcth1pd(x);
pd=zeros(size(x));
pd(1)=2*x(1)-2*x(2);
pd(2)=2*x(2)-2*x(1);

function ftauv=ftau2cg(tau);
global p1 d1
q1=p1-tau*d1;
ftauv=feval('fcth1',q1);
```

Setelah program tersebut dikerjakan diperoleh hasil :

```
» global p1 d1
» x0=[0 1]';
» x1=mincg('fcth1','fcth1pd','ftau2cg',x0,0.000005)

    Solusi x =
       0.5000
       0.5000

    Gradien:
        0
        0

Banyak iterasi :
        1
```

Sedangkan untuk contoh 2, masukannya adalah sebagai berikut:

x0=[4 2 -1]'
Toleransi = 0.000005

```
function fv=fcth2(x);
fv=(x(1)-4)^4+(x(2)-3)^2+4*(x(3)+5)^4;

function pd=fcth2pd(x);
```

```
pd=zeros(size(x));
pd(1)=4*(x(1)-4)^3;
pd(2)=2*(x(2)-3);
pd(3)=16*(x(3)+5)^3;

function ftauv=ftau2cg(tau);
global p1 d1
q1=p1-tau*d1;
ftauv=feval('fcth2',q1);
```

Setelah program tersebut dikerjakan diperoleh hasil :

```
» global p1 d1
» x0=[4 2 -1]';
» x1=mincg('fcth2','fcth2pd','ftau2cg',x0,0.000005)

Solusi x =
   4.0000
   3.0000
  -5.0067

Gradien:
  1.0e-005 *

        0
   0.1095
  -0.4858

Banyak iterasi :
  343
```

Dengan memperhatikan hasil perhitungan menggunakan kedua cara di atas, yaitu perhitungan secara manual dan perhitungan menggunakan program MATLAB untuk metode *Steepest Descent*, nampak bahwa hasil yang diperoleh adalah sama atau mendekati.

BAB IV

PENERAPAN METODE *STEEPEST DESCENT* DALAM MASALAH PRODUKSI

Pada bab ini akan dibahas tentang contoh penerapan metode *Steepest Descent* dalam kehidupan sehari-hari, yaitu dalam masalah produksi. Perhitungan akan dilakukan dengan menggunakan program MATLAB untuk metode *Steepest Descent*.

Masalah Produksi TV Warna

The ACE Manufacturing Company memproduksi 3 jenis TV warna, masing – masing stereo 24 inch, stereo 23 inch, dan stereo 21 inch .Diketahui biaya total produksi diberikan oleh rumus $C = x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 + 2x_2 - x_3 + 10$

Dimana : x_1 = jumlah TV stereo 24 inch

x_2 = jumlah TV stereo 23 inch

x_3 = jumlah TV stereo 21 inch.

Diketahui juga harga jual TV stereo 24 inch, TV stereo 23 inch, TV stereo 21 inch berturut-turut adalah p_1= \$ 300; p_2 = \$ 150; dan p_3 = \$ 75 yang merupakan pendapatan (omzet) dari penjualan. Masalah produksi yang dihadapi adalah menentukan banyaknya jenis TV stereo 24 inch , TV stereo 23 inch, TV stereo 21 inch yang harus diproduksi, agar perusahaan tersebut memperoleh laba maksimal.

Persoalan optimisasi laba tersebut di atas secara matematis dapat dirumuskan sebagai berikut:

Mencari x_1, x_2 dan x_3

Agar memaksimalkan L = R – C

Dimana :

L = Laba

$R = p_1.x_1 + p_2.x_2 + p_3.x_3$

$= 300x_1 + 150x_2 + 75x_3$ (pendapatan dari penjualan dengan asumsi bahwa semua barang yang diproduksi terjual).

$C = x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 + 2x_2 - x_3 + 10$ (fungsi ongkos)

Dengan demikian laba dari perusahaan adalah:

$$L = R - C$$

$$= 300x_1 + 150x_2 + 75x_3 - (x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 + 2x_2 - x_3 + 10)$$

$$= 300x_1 + 150x_2 + 75x_3 - x_1^2 - 2x_2^2 - x_3^2 + 2x_1x_2 - 2x_2 + x_3 - 10$$

Masalah optimisasi tersebut di atas merupakan masalah memaksimumkan nilai fungsi. Menyelesaikan masalah maksimum fungsi di atas ekuivalen dengan menentukan minimum dari negatif fungsi $-L = -300x_1 - 150x_2 - 75x_3 + x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 + 2x_2 - x_3 + 10$. Selanjutnya masalah tersebut akan diselesaikan menggunakan program MATLAB untuk metode *Steepest Descent* dengan masukan terdiri dari :

x0=[1 2 3]'

Toleransi = 0,000005

```
function fv=fsoal(x);
fv = -300*x(1)-150*x(2)-75*x(3)+x(1)^2+2*x(2)^2+x(3)^2-

     2*x(1)*x(2)+2*x(2)-x(3)+10

function pd=fsoalpd(x);
pd=zeros(size(x));
pd(1)=-300+2*x(1)-2*x(2);
pd(2)=-150+4*x(2)-2*x(1)+2;
pd(3)=-75+2*x(3)-1;

function ftauv=ftau2cg(tau);
global p1 d1
q1=p1-tau*d1;
ftauv=feval('fsoal',q1);
```

Setelah program tersebut dipanggil dan dikerjakan pada *Command Windows* diperoleh hasil sebagai berikut:

```
» global p1 d1
» x0=[1 2 3]';
» x1=mincg('fsoal','fsoalpd','ftau2cg',x0,0.000005)


Solusi x =
 374.0000
 224.0000
  38.0000

Gradien:
 1.0e-005 *

  -0.4421
   0.0776
   0.1421

Banyak iterasi :
  10
```

Dari perhitungan diperoleh nilai-nilai:

$x_1 = 374$

$x_2 = 224$

$x_3 = 38$

Dengan demikian kuantitas masing-masing jenis TV warna stereo yang harus diproduksi oleh *The ACE Manufacturing Company* agar memperoleh laba maksimal:

TV stereo 24 inch = 374 unit

TV stereo 23 inch = 224 unit

TV stereo 21 inch = 38 unit

Dengan demikian laba maksimal : $ 74110

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya, maka dapat ditarik beberapa kesimpulan berikut:

1. Optimisasi fungsi multivariabel dapat diselesaikan dengan menggunakan metode *Steepest Descent* yaitu suatu algoritma gradien di mana langkah ke α_k adalah memilih pencapaian jumlah maksimum berkurangnya nilai fungsi obyektif , yakni $\alpha_k = \min f(\mathbf{x}^{(k)} - \alpha\nabla f(\mathbf{x}^{(k)}), \alpha \geq 0$.
2. Setiap langkah dari metode *Steepest Descent* dimulai dari titik $\mathbf{x}^{(k)}$, untuk k =0, 1, 2, 3, ... Kemudian dihitung gradien dari $f(\mathbf{x})$ pada titik tersebut. Setelah itu dihitung norma dari gradien tersebut. Jika $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$, maka iterasi dihentikan dan dengan demikian $\mathbf{x}^* = \mathbf{x}^{(k)}$ merupakan titik optimum Jika sebaliknya, maka dilanjutkan ke iterasi berikutnya dengan arah pencarian pada titik $\mathbf{x}^{(k)}$ adalah $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$. Kemudian dihitung α_k untuk meminimumkan $f(\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)})$. Selanjutnya dilakukan iterasi berikutnya sesuai dengan langkah-langkah pada iterasi sebelumnya hingga suatu pembuat minimum $\mathbf{x}^{(k+1)}$ ditemukan.
3. Keistimewaan dari metode *Steepest Descent* adalah bahwa dalam masalah optimasi fungsi multivariabel penyelesaiannya dibawa ke fungsi satu variabel

sehingga penyelesaiannya lebih mudah dilakukan. Penyelesaian akan menjadi lebih efektif dan efisien jika perhitungannya tidak dilakukan secara manual, tetapi dengan menggunakan komputer sebagai alat bantu, yakni dengan menggunakan program MATLAB untuk metode *Steepest Descent*.

B. Saran

Berikut diberikan beberapa saran yang dapat dijadikan permasalahan yang dapat dibahas lebih lanjut berkenaan dengan metode *Steepest Descent*:

1. Analisis kekonvergenan metode *Steepest Descent*.
2. Metode *Steepest Descent* untuk menyelesaikan sistem persamaan non linear

DAFTAR PUSTAKA

Anton, Howard. (1987). *Aljabar Linear Elementer*. Edisi kelima. Erlangga, Jakarta.

Avriel, Mordecai. (1976). *Non Linear Programming Analisis and Methods*. Prentice Hall, New Jersey.

Bazaraa, Sherali and Shetty. (1993).*Non Linear Programming Theory and algoritms*, John Wiley & Sons, Inc, Singapore.

Benyamin, F. Plybon, *An Introduction to Applied Numerical Analysis*. Miami University.

Chong, Edwin K.P, and Zak, Stanislaw H. (1996). *An Introduction to Optimization*, John Wiley & Sons.

John. H, Mathews, *Numerical Methods for Computer Science, Engineering, and Mathematics*.

Lindfield. G, and Penny. J. *Numerical Methods Using MATLAB*. Departement of Mechanical Engineering, Aston University.

Meerschaert, Mark M. *Mathematical Modelling*. Academic Press, INC.

Mital, K.V. (1976). *Optimization Methods*. Wiley Eastern Limited.