

**PROSIDING**

**S E T I S I**

**2 0 1 1**

**Seminar Teknik Informatika & Sistem Informasi**

**Bandung, 24 September 2011**

**Seminar Teknik Informatika & Sistem Informasi  
Fakultas Teknologi Informasi  
Universitas Kristen Maranatha**

---

[setisi.itmaranatha.org](http://setisi.itmaranatha.org)

---

**PROSIDING**  
**SETISI 2011 Seminar Teknik Informatika & Sistem Informasi**

Penyusun:  
Fakultas Teknologi Informasi – Universitas Kristen Maranatha

Penerbit:  
Informatika Bandung

Cetakan Pertama:  
September 2011

ISBN:  
978-602-8758-37-6

**Copyright © 2011**

---

**DAFTAR ISI**

KATA PENGANTAR.....	iii
SAMBUTAN DEKAN.....	iv
DAFTAR ISI .....	v

**I****KELOMPOK JARINGAN KOMPUTER DAN KEAMANAN SISTEM**

<b>Pemodelan Aplikasi Remote Desktop Menggunakan Protokol UDP dan MD5 Checksum Serta Mime Base 64 Encoding Sebagai Keamanan Komunikasi</b> <i>Wiwin Sulisty, Dian W. Chandra, Heru Setyawan .....</i>	3
<b>Modul Enkripsi/Dekripsi RUFT Sebagai Alternatif Pengamanan Informasi Pada Media Komunikasi GPRS</b> <i>Sandromedo Christa Nugroho, Wahyu Indah Rahmawati, Neo Fajar Bawana Konta Dewa Danu .....</i>	11
<b>Pengembangan Aplikasi Microblogging Twitter Berbasis Desktop</b> <i>Brajaseta Yamaditya, Wilfridus Bambang Triadi Handaya .....</i>	19
<b>Kajian Manajemen Antrian Pada Jaringan Multiprotocol Label Switching</b> <i>Timotius Witono .....</i>	22

**II****KELOMPOK KOMPUTASI CERDAS**

<b>Implementasi Algoritma OKAPI BM25 dan K-MEANS Untuk Mencari Relevansi Artikel Pada Beberapa Situs Berita</b> <i>Danny Sebastian, Antonius Rachmat, Willy Sudiarto Raharjo .....</i>	31
<b>Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan dengan Studi Kasus Pada SMP X</b> <i>Radiant Victor Imbar, Jayanti .....</i>	36
<b>Pemanfaatan Fuzzy Logic Untuk Memprediksi Prestasi Mahasiswa Berdasarkan Nilai Ujian Nasional (Studi Kasus: STMIK IBBI)</b> <i>Hartono .....</i>	42
<b>Pemeringkatan Program Studi Perguruan Tinggi Berbasis Data EPSBED dan Webometrics Dengan Metode Pembelajaran Berbasis Kasus</b> <i>Marzuki, Ahmad Cucus .....</i>	48
<b>Sistem Deteksi Kelainan Jantung Menggunakan Sinyal Elektrokardiogram dengan Metode Empirical Mode Decomposition</b> <i>Rosita Dewi, Bambang Hidayat, Achmad Rizal .....</i>	54

<b>Implementasi Metode Fuzzy Rule Base Pada Kasus Job-Shop dengan Penjadwalan Adaptif</b> <i>Ahmad Saikhu, Husnul Hakim</i> .....	59
<b>Pendekatan Penyelesaian Masalah Knapsack Dalam Pembuatan Menu Makanan Sehat</b> <i>Oscar Wongso, Mewati Ayub</i> .....	66
<b>Sistem Pakar Diagnosa Penyakit Paru-paru pada Anak dengan Metode Dempster-Shafer</b> <i>Noor Latifah, Eko Darmanto</i> .....	74
<b>Rancangan Sistem Rekomendasi Pencarian Kerja dengan Pendekatan Content-Based Filtering</b> <i>Herastia Maharani</i> .....	78
<b>Pembentukan Pola Genetik Kalimat Tanya Pada Sistem Tanya Jawab Melalui Pembelajaran Analogi</b> <i>Hapnes Toba</i> .....	84
<b>Penerapan Algoritma Jaringan Saraf Tiruan <i>Self Organizing Map</i> untuk Pengenalan Pola Pada Tulisan Tangan</b> <i>Tjatur Kandaga, Ardy Gunawan</i> .....	92
<b>Pengenalan Bahasa Isyarat Tangan Menjadi Huruf Alfabet Dengan Jaringan Syaraf Tiruan</b> <i>Sofyan, Arnold Aribowo, Reza Adrianus</i> .....	99

### III

#### KELOMPOK PENGOLAHAN CITRA DAN GAME

<b>Sistem Perolehan Citra Mammogram Berdasarkan Ciri Tekstur Menggunakan Gray Level Co-Occurrence Matrix</b> <i>Shofwatul 'Uyun, Agus Harjoko</i> .....	107
<b>Komputasi Simbolik Representasi Titik Dengan Metode Region-Based Mx Quadtree Untuk Pengenalan Daerah Sawah Rawan Banjir</b> <i>Eko Darmanto, Noor Latifah</i> .....	113
<b>Metode Adaptive Background Extraction-Gaussian Mixture Models Untuk Aplikasi Penghitung Kendaraan Berbasis Video</b> <i>Resmana Lim, Raymond Sutjiadi, Endang Setyati</i> .....	118
<b>Game Bergenre Simulation-Action Untuk Pembelajaran Keamanan Jaringan Komputer Dengan Pendekatan Discovery</b> <i>Anteng Widodo, Supriyono</i> .....	124
<b>Pembangunan Perangkat Lunak Game Maker X dengan Mengimplementasikan Teknik Reflection Dan Object Reuse</b> <i>Oscar Karnalim</i> .....	129
<b>Pemanfaatan Fitur Warna Pada Aplikasi Penghitung Sel Kanker Interaktif</b> <i>Roy Artha Perdana, Binti Solihah</i> .....	135



<b>Deteksi Gerak Satu Obyek Pada Video AVI</b> <i>Arif Setiawan, Pratomo Setiaji</i> .....	141
---	-----

## IV

### KELOMPOK SISTEM INFORMASI DAN SISTEM ENTERPRISE

<b>Membangun Aplikasi Perbankan Sebagai Kontrol Optimal Internal Pada Operasional Bank "XYZ"</b> <i>Johannes Petrus, Suwirno Mawlan</i> .....	147
<b>Strategi Integrasi Data dan Aplikasi Enterprais Menggunakan Enterprise Application Integration Studi Kasus: DSS BAPPENAS</b> <i>Gede Karya</i> .....	152
<b>Analisis Efektivitas Penerapan E-Government Terhadap Peningkatan Pelayanan Masyarakat Studi Kasus: Pemerintahan Kabupaten Banyuwasin</b> <i>Mulyati, Desy Iba Ricoida</i> .....	158
<b>Integrasi Model Tata Kelola Teknologi Informasi Antara ISO 38500, RISK IT, dan VAL IT</b> <i>Meliana Christianti Johan, Kridanto Surendro</i> .....	164
<b>Model Konseptual E-Health Pada Departemen Ilmu Kesehatan Anak di Indonesia</b> <i>Oktri Mohammad Firdaus, Eki Rakhmah Zakiyyah</i> .....	174
<b>Taint Analysis dan Eksploitasi Perangkat Lunak</b> <i>Suryo Bramasto</i> .....	179
<b>Sistem Informasi Geografis Tata Letak Pengairan Sungai dan Irigasi Pada Balai PSDA SELUNA</b> <i>Pratomo Setiaji, Arif Setiawan</i> .....	186
<b>Studi Empiris Model Pemrediksian Popularitas Situs Jejaring Sosial</b> <i>Henri Agustin</i> .....	191
<b>Sistem Informasi Pemetaan Penanganan Ibu Hamil dan Neonatal</b> <i>Supriyono, Anteng Widodo</i> .....	197
<b>Studi Pengaruh Aktivitas Twitter Terhadap Tingkat Pengaruh Seseorang Kepada Lingkungan Pada Media Sosial Digital</b> <i>Rahmat Izwan Heroza</i> .....	201
<b>Pengembangan Arsitektur Aplikasi Enterprise dengan IOSDP (Studi Kasus: Pengembangan ERP Menggunakan Mysql 2005 dan Codeigniter 2.0.2)</b> <i>Gerald Kevin Suoth</i> .....	204
<b>Sistem Penunjang Kegiatan Operasional Klinik Bahtera Medika</b> <i>Ririn Ikana Desanti, Arnold Aribowo, Agustinus Chandra</i> .....	209
<b>Pembuatan Aplikasi Sistem Informasi Berbasis C#.Net Pada Toko Besi Baja Mulia Dalam Manajemen Transaksi dan Manajemen Persediaan</b> <i>Wenny Franciska Senjaya, Nico Budi Darmawan</i> .....	215

<b>Sistem Pengelolaan Surat dan Paket Compassion® Indonesia</b> <i>Riz Sony Saksono Harso S., Tiur Gantini</i> .....	222
<b>Perilaku Wanita Dalam Pemanfaatan Teknologi Informasi (Identifikasi Faktor-faktor Motivasional Wanita Karir Dalam Penggunaan Internet Sebagai Penunjang Kesuksesan Kerja)</b> <i>Endang Raino Wirjono</i> .....	229
<b>Sistem Informasi Manajemen Dokumen Mutu (SIDOKU) Sebagai Pendukung Implementasi Sistem Penjaminan Mutu di Laboratorium Terpadu Teknik Informatika UII</b> <i>Lizda Iswari, Rio Kiswandar, Rahadian Faiz Kurniawan, Akhyar Amarullah</i> .....	239
<b>Perancangan dan Pembuatan Website Paid To Click</b> <i>Andy Sentosa, Djoni Setiawan K</i> .....	245
<b>Website Penjualan dan Lelang Meubel Online Pada Toko Meubel "X"</b> <i>Rudy Susanto, Tanti Kristanti</i> .....	251
<b>Pembuatan Website Pariwisata Jawa Tengah</b> <i>Wijayanti Santoso Gandhi, Robby Tan</i> .....	256
<b>Pembuatan Sistem Informasi Akademik Sekolah Menengah Atas Swasta XYZ Menggunakan Java Enterprise Edition</b> <i>Niko Ibrahim, Sedy Ferdian Sujadi</i> .....	260
<b>Pembangunan Sistem Manajemen Ujian "Chiron" Dengan Menerapkan Teknologi LINQ dan Windows Presentation Foundation</b> <i>Erico Darmawan Handoyo, Sulaeman Santoso</i> .....	265
<b>Model Perencanaan Bisnis Adopsi Cloud Computing</b> <i>Andy Prasetyo Utomo, Kridanto Surendro</i> .....	269
<b>Website E-Commerce dan Inventory Management Perusahaan Cutting Tools</b> <i>Diana Trivena Yulianti, Andrianus</i> .....	274
<b>Website Penjualan dan Lelang Spare Part Komputer Dengan Teknologi PHP dan SMS Gateway</b> <i>Teddy Marcus Zakaria, Kevin Tedja</i> .....	279
<b>Analisis Information Economic Untuk Implementasi SAP di Universitas Kristen Maranatha (UKM)</b> <i>Jessica Susanto, Saron Kurniawati Yefta</i> .....	285
<b>Aplikasi Piranti Lunak Pembelian, Penjualan Dan Stok Barang Pada Toko Suka Sari</b> <i>Ririn Ikana Desanti, Arnold Aribowo, Tomy Oscar</i> .....	289
<b>Pengawasan Implementasi Monitoring Center (Studi Kasus: Kejaksaaan Agung RI)</b> <i>Maniah</i> .....	292

## PENDEKATAN PENYELESAIAN MASALAH KNAPSACK DALAM PEMBUATAN MENU MAKANAN SEHAT

Oscar Wongso, Mewati Ayub

Jurusan Teknik Informatika Fakultas Teknologi Informasi  
Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri 65 Bandung  
lonebloody@yahoo.com, mewati.ayub@itmaranatha.org

### ABSTRAK

Nutrisi dalam makanan yang dikonsumsi oleh seseorang seringkali tidak terkendali, bahkan menjadi berlebih, terutama dalam jumlah kalori makanan yang dikonsumsi. Untuk mengatur jumlah kalori yang dikonsumsi sesuai dengan takaran nutrisi yang dianjurkan setiap harinya, perlu dibuat sebuah aplikasi yang mampu membuat menu makanan sehat dan sesuai dengan takaran nutrisi yang dibutuhkan oleh tubuh. Pembuatan menu makanan tersebut dilakukan dengan menggunakan pendekatan penyelesaian masalah Knapsack, yaitu dengan program dinamis dan algoritma Greedy. Aplikasi tersebut diharapkan dapat memberikan menu makanan yang disarankan sesuai dengan kebutuhan yang dianjurkan perharinya baik dalam jumlah kalori maupun dalam jumlah nutrisi. Implementasi aplikasi tersebut menggunakan lingkungan pemrograman .Net, dengan Microsoft SQL Server sebagai tempat penyimpanan data makanan dan rekomendasi nutrisi. Untuk membuktikan bahwa aplikasi menu makanan tersebut dapat beroperasi dengan baik, maka telah dilakukan pengujian terhadap aplikasi tersebut yang menunjukkan hasil yang sesuai dengan kebutuhan nutrisi dan kalori seseorang.

**Keywords:** nutrisi, kalori, program dinamis, algoritma Greedy

### 1. PENDAHULUAN

Setiap harinya, masyarakat Indonesia dianjurkan untuk mengonsumsi makanan secara rutin, yaitu tiga kali sehari [8]. Dalam pengonsumsiannya, terdapat batasan nutrisi minimal yang dibutuhkan oleh tubuh, dan batasan maksimal yang boleh dikonsumsi per harinya. Angka Kecukupan Gizi (AKG) dapat digunakan sebagai acuan perencanaan makanan dan penilaian tingkat konsumsi makanan individu atau masyarakat [7].

Salah satu batasan minimal nutrisi yang diperlukan oleh tubuh dalam melakukan aktivitas setiap harinya, adalah kalori dalam makanan [5]. Setiap orang memiliki kebutuhan jumlah kalori yang berbeda – beda tergantung dari banyaknya aktivitas yang dilakukan, gender, umur, tinggi serta berat badan yang dimilikinya [4][6].

Karena itu, diperlukan sebuah aplikasi yang dapat membantu penggunaannya, untuk menjaga jumlah nutrisi yang dibutuhkan, dengan mengikuti menu makanan sehat yang dianjurkan sesuai dengan AKG dan aktivitas yang dilakukan, khususnya untuk kalangan remaja dan dewasa.

Salah satu solusi yang dapat digunakan dalam pembuatan aplikasi tersebut adalah dengan menggunakan pendekatan penyelesaian masalah Knapsack. Cara pencarian solusi dengan pendekatan ini diharapkan dapat memberikan hasil optimal untuk pembuatan menu makanan sehat. Terdapat beberapa cara dalam pencarian

solusi untuk menyelesaikan masalah Knapsack tersebut, diantaranya adalah dengan menggunakan program dinamis dan algoritma Greedy.[1]

### 2. LANDASAN TEORI

#### 2.1 Masalah Knapsack

Masalah Knapsack dapat digambarkan dengan sebuah kasus di mana seorang pencuri ingin memasukkan perhiasan ke dalam sebuah kantung yang memiliki kapasitas maksimal berat barang yang dapat ditampungnya. Setiap barang yang dimasukkan ke dalam kantung tersebut memiliki berat dan *profit* yang berbeda-beda. Tujuan utama dalam permasalahan Knapsack ini adalah mencari solusi terbaik, yaitu barang apa saja yang dapat dimasukkan ke dalam kantung, sehingga jumlah profit dari barang tersebut mencapai nilai optimal, dengan jumlah berat barang tidak melebihi dari kapasitas maksimal kantung. Jika masalah Knapsack ini dihubungkan dengan pemilihan menu makanan, maka berat barang dapat disetarakan dengan jumlah kalori makanan, dan *profit* dari barang tersebut adalah nutrisi dari makanan yang dikonsumsi [1].

Masalah Knapsack dibedakan menjadi Knapsack 0/1 dan *Fractional Knapsack*. Knapsack 0/1 diselesaikan dengan memasukkan barang secara utuh (1) atau tidak dimasukkan sama sekali (0), sedangkan *Fractional Knapsack* memungkinkan barang dimasukkan hanya sebagian saja dalam bentuk pecahan ( $0 \leq x \leq 1$ ).

**2.2 Program Dinamis untuk Knapsack 0/1**

Permasalahan Knapsack 0/1 dapat diselesaikan dengan menggunakan program dinamis. Cara kerja program dinamis ini adalah dengan membagi permasalahan menjadi beberapa tahap, kemudian mencari solusi optimal dari setiap tahap. Hasil dari suatu tahap dapat digunakan untuk membangun solusi pada tahap berikutnya. Dalam permasalahan Knapsack 0/1, banyaknya barang dimasukkan oleh banyaknya barang yang dapat dimasukkan ke dalam Knapsack. Pada Tabel 1, setiap tahap digambarkan sebagai suatu baris dalam Matriks V.

Pencarian solusi pada program dinamis dapat ditunjukkan dengan menggunakan empat macam barang yang memiliki berat (variabel w) dan profit (variabel p), dengan berat maksimal kantung yang digunakan adalah 10, seperti pada Tabel 1.

Pertama-tama, kita gunakan kapasitas berat kantung sebesar 0. Dikarenakan tidak ada barang yang dapat dimasukkan ke dalam kantung, maka nilai dengan berat 0 berisikan angka nol mulai pada baris i=0 hingga 4. Hal ini terus berlanjut hingga kapasitas kantung mencapai 3. Pada kapasitas berat kantung sebesar 3, terdapat perbandingan profit maksimal yang didapatkan pada hasil sebelumnya, dengan membandingkan jumlah profit yang didapatkan jika mengganti barang. Akan tetapi, dikarenakan profit maksimal sebelumnya sama dengan 0, maka barang ke-4 dimasukkan ke dalam kantung. Sama halnya pada berat kapasitas maksimal kantung sebesar 10, terdapat perbandingan antara solusi sebelumnya dengan mengganti barang, hingga akhirnya tercapai profit optimal sebesar 90.

Tabel 1. Contoh Perhitungan dengan Program Dinamis [2]

(Barang ke -)	i	1	2	3	4
(Profit barang)	v <sub>i</sub>	10	40	30	50
(Berat barang)	w <sub>i</sub>	5	4	6	3

V[i, w]	0	1	2	3	4	5	6	7	8	9	10
i = 0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	10	10	10	10	10	10
2	0	0	0	0	40	40	40	40	40	50	50
3	0	0	0	0	40	40	40	40	40	50	70
4	0	0	0	50	50	50	50	90	90	90	90

Berikut ini adalah rumus 1 yang digunakan untuk perhitungan profit dalam Program Dinamis untuk persoalan Knapsack 0/1 [2]:

$$V[i, w] = \max(V[i - 1, w], v_i + V[i - 1, w - w_i])$$

Cara kerja program dinamis tersebut dapat dituliskan menjadi sebuah fungsi seperti di bawah ini:

```

Function Knapsack0_1(input p,w: array
[1...n] of integer; n,W:integer)→ integer
{p adalah array profit barang, w array berat
barang, n banyak barang, dan W berat
maksimal kantung}
Deklarasi Data:
i, b: integer
V: array [0...n][0...W] of integer
Keep: array [0...n][0...W] of integer

Algoritma:
b traversal [0...W]
V[0, b] = 0
end traversal

i traversal [1...n]
b traversal [0...W]
if((w[i] ≤ b) and
(p[i] + V[i-1,b-w[i]]>V[i-1,b]))
then {barang diambil}
V[i,b]← p[i]+V[i-1, b-w[i]]
Keep[i,b] ← 1
else {barang tidak diambil}
V[i, b] ← V[i-1, b]
Keep[i,b] ← 0
end if
end traversal
end traversal
return (V[n, W])
    
```

Algoritma 1. Fungsi Program Dinamis untuk Knapsack 0/1 [2]

**2.3 Algoritma Greedy untuk Fractional Knapsack**

Algoritma Greedy bekerja sesuai dengan namanya, yaitu dengan mengambil sebanyak-banyaknya barang hingga berat maksimal tercapai, yang dapat dituliskan dengan rumus 2. Algoritma Greedy seringkali gagal untuk mendapatkan solusi optimal untuk menyelesaikan masalah Knapsack 0/1 [3]. Tetapi, jika pencarian solusi dengan algoritma Greedy dilakukan untuk Fractional Knapsack yang memungkinkan barang dimasukkan sebagian (berbentuk pecahan), maka algoritma Greedy akan memberikan hasil optimal dengan melihat perbedaan density atau pembagian profit dengan berat dari nilai yang terbesar hingga terkecil, yang dapat kita lihat pada tabel 2.

Berikut ini adalah rumus 2 yang digunakan untuk perhitungan profit dalam algoritma Greedy untuk persoalan Fractional Knapsack [3]:



Maksimasi 
$$F = \sum_{i=1}^n p_i x_i$$
  
dengan kendala 
$$\sum_{i=1}^n w_i x_i \leq K$$
  
dalam hal ini,  $0 \leq x_i \leq 1, i = 1, 2, \dots, n$

Tabel 2. Contoh Pemakaian Algoritma Greedy untuk Fractional Knapsack [3]

Properti objek				Greedy by		
i	w <sub>i</sub>	p <sub>i</sub>	p <sub>i</sub> /w <sub>i</sub>	profit	weight	density
1	18	25	1,4	1	0	0
2	15	24	1,6	2/15	2/3	1
3	10	15	1,5	0	1	1/2
Total bobot				20	20	20
Total keuntungan				28,2	31,0	31,5

Cara kerja algoritma Greedy tersebut dapat dituliskan menjadi sebuah fungsi seperti di bawah ini:

```

Function Fractional_Knapsack (input C: objek;
n, W: integer) → objek
{objek C terdiri dari berat barang, profit barang,
dan density. Objek tersebut telah terurut
berdasarkan density dari terbesar hingga terkecil}
Deklarasi Data:
i, sum: integer
muat: boolean
x: array [1...n] of integer

Algoritma:
i traversal [1...n]
    x[i] ← 0
end traversal

i ← 0; sum ← 0; muat ← true
while (i ≤ n) and (muat) do
    i ← i + 1
    if (sum + C.w[i] ≤ W)
    then x[i] ← 1
        sum ← sum + C.w[i]
    else muat ← false
        x[i] ← (W - sum) / C.w[i]
    end if
end while
return x
    
```

Algoritma 2. Fungsi Algoritma Greedy untuk Fractional Knapsack [3]

### 3. PENERAPAN PROGRAM DINAMIS DAN ALGORITMA GREEDY DALAM PEMBUATAN MENU MAKANAN

#### 3.1 Penerapan Program Dinamis dalam Pembuatan Menu Makanan

Untuk memudahkan dalam pencarian solusi menu makanan, data makanan akan dibagi berdasarkan waktu santap, yang diurut berdasarkan jumlah kalori dan rata-rata nutrisi. Data makanan ini akan dimasukkan ke dalam sebuah list data makanan dan diurut berdasarkan jumlah rata-rata nutrisi dan kalori yang terkandung di dalamnya, dengan urutan data makanan yang memiliki jumlah rata-rata nutrisi dan kalori dari jumlah terkecil hingga terbesar. Jika terdapat makanan dengan jumlah kalori yang sama, maka makanan dengan jumlah rata-rata nutrisi yang lebih kecil akan diprioritaskan terlebih dahulu. Sebagai contoh, pada tabel 3 terdapat data makanan yang sudah diurutkan berdasarkan jumlah kalori dan jumlah rata-rata nutrisi.

Tabel 3 Contoh Data Makanan yang Sudah Diurutkan

Nama Makanan	Jenis Makanan	Jumlah Kalori (kal)	Rata-rata nutrisi	...
Pepaya	Buah	10	10	...
Apel	Buah	10	15	...
Sayur Asam	Sayuran	12	20	...
Ikan Asin	Daging	14	20	...
Nasi Merah	Pokok	23	25	...
Nasi Putih	Pokok	27	30	...

Setelah data makanan dibagi berdasarkan waktu santap dan diurutkan pada list, maka pencarian solusi menu makanan akan dilakukan dengan menggunakan dua tabel, yaitu tabel penyimpanan (*keep(i,w)*), dan tabel jumlah nutrisi yang didapatkan (*V(i,w)*). Jumlah baris dan kolom pada tabel *V(i,w)* dan *keep(i,w)* bergantung kepada jumlah kalori (*w*) dan banyaknya data pada list makanan (*i*) seperti yang terlihat pada tabel 4, dengan jumlah kalori sebagai kolom tabel, dan banyaknya data makanan sebagai baris tabel.

Tabel 4. Contoh Tabel dalam Program Dinamis

V(i,w)	0	1	...	50	...	b
1	0	0	...	30	...	0
...	...	...	...	...	...	...
n	0	0	...	30	...	100

Keep(i,w)	0	1	...	50	...	b
1	0	0	...	1	...	0
...	...	...	...	...	...	...
n	0	0	...	0	...	1

Tabel *keep(i,w)* akan diisi dengan angka nol dan satu, dengan angka satu menandakan data

makanan ke- $i$  pada list dimasukkan ke dalam menu makanan, sedangkan angka nol menunjukkan data makanan ke- $i$  pada list tidak dimasukkan ke dalam menu makanan. Tabel  $V(i,w)$  dan diisi dengan jumlah rata-rata nutrisi yang didapatkan dari penjumlahan data-data makanan, yang diambil dari tabel  $Keep(i,w)$ .

Menjadi solusi menu makanan ini dimulai dari data makanan pertama hingga ke- $n$  dengan jumlah kalori sebesar 0, dan dilanjutkan hingga jumlah kalori maksimal atau hingga data pada tabel solusi saat terisi semua.

Untuk mengisi kolom-kolom pada tabel tersebut, dilakukan perbandingan hasil data makanan yang didapat pada kolom-kolom sebelumnya dengan melihat jumlah rata-rata nutrisi makanan. Jika jumlah rata-rata nutrisi lebih besar dibandingkan kolom sebelumnya, dan kalori makanan tidak lebih besar dari batas maksimal yang tertera pada kolom, maka data makanan tersebut akan ditandai dan diberi angka satu. Jika ternyata jumlah rata-rata nutrisi yang didapatkan lebih kecil dengan nilai pada kolom sebelumnya setelah data makanan diganti, maka data makanan tersebut akan ditandai dan diberi angka nol.

Setelah tabel  $Keep$  terisi, maka pemilihan data makanan ke dalam menu pun akan dilakukan. Pemilihan data makanan ini akan melihat dari angka nol dan satu dilihat dari tabel  $Keep$  dimulai dari data makanan terakhir pada kolom kalori terbesar. Jika angka pada tabel berisikan satu, maka data makanan akan dimasukkan ke dalam menu, kemudian melakukan pengecekan apakah jumlah kalori maksimal dikurangi kalori makanan tersebut sudah mencapai angka nol atau tidak. Jika tidak, maka pemilihan makanan akan terus dilanjutkan hingga mencapai data makanan pertama. Pemasukan data makanan di atas berpegang kepada prinsip empat sehat lima sempurna yang dapat kita lihat pada tabel 5 dan 6.

Tabel 5. Contoh Pengerjaan Solusi dengan Program Dinamis

Keep(i,w)	0	...	50	...
Pepaya	0	...	1	...
Apel	0	...	1	...
Sayur Asam	0	...	0	...
Ikan Asin	0	...	0	...
Nasi Merah	0	...	0	...
Nasi Putih	0	...	1	...

V(i,w)	0	...	50	...
Pepaya	0	...	25	...
Apel	0	...	15	...

V(i,w)	0	...	50	...
Sayur Asam	0	...	15	...
Ikan Asin	0	...	15	...
Nasi Merah	0	...	15	...
Nasi Putih	0	...	15	...

Tabel 6. Contoh Pemasukan Data Menu Makanan Untuk Program Dinamis

Jenis Makanan (4 sehat 5 sempurna)	Nama Makanan	Pemasukan	Nama Makanan
Makanan Pokok	-	-->	Nasi Putih
Sayuran	-		-
Daging	-		-
Buah	-		-

### 3.2 Penerapan Algoritma Greedy untuk Pembuatan Menu Makanan

Berbeda dengan teknik pencarian pada algoritma sebelumnya, pada algoritma *Greedy*, pencarian solusi menu makanan bergantung kepada jumlah profit atau keuntungan yang akan kita dapatkan. Jumlah profit tersebut kita dapatkan dari pembagian rata-rata nutrisi dengan kalori pada tiap makanan. Sehingga diharapkan jumlah nutrisi yang kita dapatkan dalam menu adalah optimal.

Sebelum data makanan diurutkan, data pada tabel makanan akan dibagi menjadi list-list yang lebih kecil berdasarkan waktu santap makanan tersebut. Setelah itu, data makanan pada list tersebut akan diurutkan berdasarkan *density* dengan urutan dari angka yang terbesar hingga terkecil. *Density* adalah pembagian rata-rata nutrisi dengan jumlah kalori yang dapat kita lihat pada tabel 7. Jika terdapat jumlah *density* yang sama, maka jumlah kalori yang lebih kecil akan diprioritaskan terlebih dahulu.

Tabel 7 Contoh Data Makanan yang Diurutkan Berdasarkan Density

Nama Makanan	Jenis Makanan	Jumlah Kalori (c)	Jumlah rata-rata nutrisi (n)	Density (n/c)
Apel	Buah	1	15	15
Nasi Merah	Pokok	2	25	12.5
Pepaya	Buah	1	10	10
Sayur Asam	Sayuran	2	20	10
Ikan Asin	Daging	2	20	10
Nasi Putih	Pokok	3	30	10

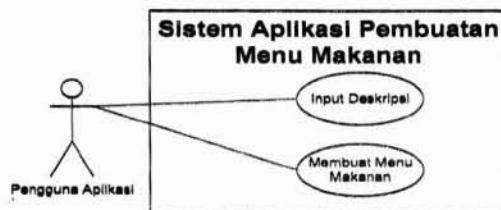
Setelah data makanan diurutkan pada tiap list makanan, data makanan tersebut akan dimasukkan ke dalam solusi menu makanan dengan melihat nilai *density* makanan tersebut. Cara kerja algoritma Greedy adalah mengambil semua data makanan yang bisa didapatkan hingga jumlah kalori maksimal tercapai. Jika jumlah kalori maksimal belum tercapai dan jumlah kalori makanan masih lebih besar daripada kapasitas jumlah kalori, maka akan diambil data makanan berikutnya dengan membagi porsi makanan tersebut, sehingga total kalori pada menu makanan yang dicapai, tidak melebihi kalori maksimal pada menu makanan yang harus dikonsumsi. Seperti pada tabel 8, di mana porsi ikan asin dikurangi menjadi setengahnya.

Tabel 8. Contoh Solusi Menu Makanan dengan Algoritma Greedy

Jenis Makanan (4 sehat 5 sempurna)	Porsi Makanan (g)	Nama Makanan	Jumlah Kalori Makanan(kal)
Makanan Pokok	1	Nasi Merah	25
Sayuran	1	Sayur Asam	20
Daging	1 -> ½	Ikan Asin	½ * 20 = 10
Buah	1	Apel	15
Susu	-	-	-

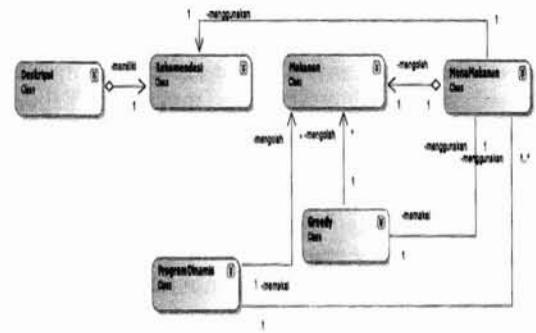
#### 4. DISAIN PERANGKAT LUNAK

Pada gambar 1 ditampilkan diagram *Use Case* dari aplikasi pembuatan menu makanan, sedangkan pada gambar 2 ditampilkan diagram *class* dari aplikasi.



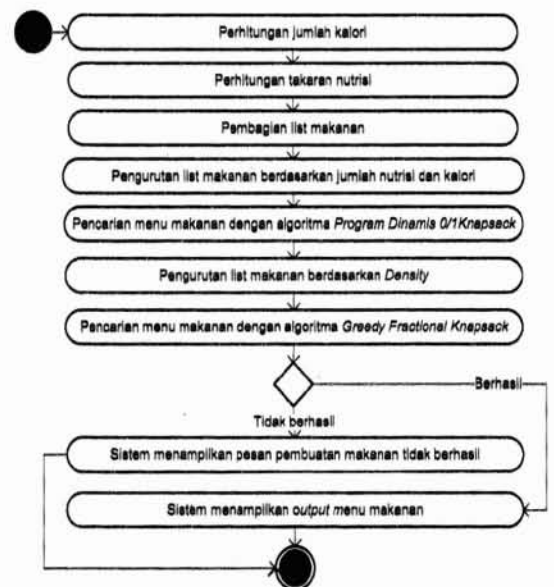
Gambar 1. Diagram Use Case

Fitur yang ada pada aplikasi ini adalah memasukkan deskripsi data makanan dan membuat menu makanan. Data makanan yang digunakan berasal dari Daftar Komposisi Bahan Makanan Indonesia, sedangkan untuk takaran nutrisi yang digunakan berasal dari tabel Angka Kecukupan Gizi 2004.



Gambar 2. Diagram Class

Aktivitas membuat menu makanan seperti tampak pada gambar 3, dapat dijelaskan dengan tahapan sebagai berikut. Aplikasi memproses data yang diberikan dari aktivitas Input Deskripsi. Kemudian aplikasi akan menghitung jumlah kalori yang dibutuhkan oleh pengguna per harinya. Sebelum mencari solusi menu makanan, aplikasi menghitung takaran nutrisi yang dibutuhkan seperti jumlah maksimal vitamin dan lainnya berdasarkan data yang telah dimasukkan sebelumnya. Setelah mendapatkan takaran nutrisi dan kalori yang dibutuhkan, aplikasi akan membuat menu makanan dengan menggunakan algoritma Greedy dan program dinamis. Pembuatan menu makanan dimulai dengan pembagian dan pengurutan list makanan. Kemudian sistem akan mencari solusi menu makanan dengan menggunakan kedua algoritma tersebut di atas. Setelah pencarian menu makanan selesai, aplikasi akan menampilkan *output* berupa menu makanan kepada pengguna.



gambar 3. Diagram Aktivitas

Penyimpanan data menggunakan dua tabel, yaitu tabel Data Makanan dan tabel Data Rekomendasi. Sedangkan hasil proses pembuatan menu makanan disimpan dalam struktur list dari objek kelas sesuai diagram *class* pada gambar 2.



**5. IMPLEMENTASI**

Implementasi program dinamis dan algoritma Greedy dalam penyelesaian masalah *Knapsack* memiliki kecenderungan untuk selalu memberikan hasil menu makanan yang sama, dengan jumlah takaran nutrisi maksimal yang diberikan. Salah satu cara yang dapat digunakan untuk memberikan menu makanan yang bervariasi adalah dengan melakukan pengacakan pada data makanan, yang dapat dilihat pada tabel 9.

Tabel 9. Implementasi Pemilihan Makanan Secara Acak

Nama Makanan		Nama Makanan
Nasi Putih	Pengacakan →	Nasi Putih
Nasi Merah		Sayur Asin
Sayur Asam		Sayur Asin
Sayur Asin		Apel
Apel		

Kecenderungan lain yang dihasilkan dalam pengimplementasian terletak pada jumlah nutrisi solusi menu makanan yang dihasilkan. Dikarenakan sifat algoritma ini selalu memberikan hasil optimal, yaitu takaran nutrisi makanan yang optimal dengan bertolak ukur pada batasan kalori maksimal. Akibatnya hasil jumlah nutrisi pada menu makanan menjadi berlebih di atas batasan takaran yang dianjurkan. Cara yang digunakan untuk mengatasi masalah tersebut adalah dengan memotong porsi menu makanan sehingga jumlah nutrisi pun dapat disesuaikan dengan takaran yang dianjurkan. Pengimplementasian solusi ini dapat dilihat pada tabel 10, yang menunjukkan pemotongan porsi makanan dengan takaran nutrisi yang dianjurkan untuk vitamin A per harinya adalah 500 µg.

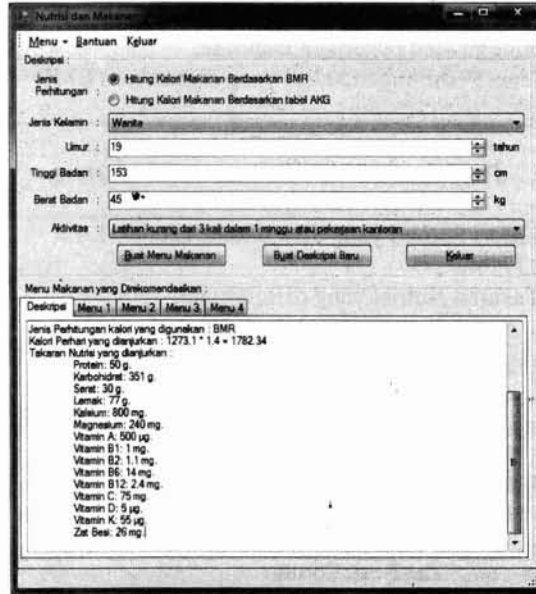
Tabel 10. Pemotongan Porsi Menu Makanan

Nama Makanan	Porsi Makanan	Kandungan Vitamin A (µg)
Nasi goreng+telur	145.82 gram	433
Sawi	8.7 gram	646
Ayam goreng	10 gram	81
Semangka	46 gram	590
Cake coklat (bolu)	10 gram	7,5
<b>Total Kandungan Vitamin A</b>		<b>1757,5</b>

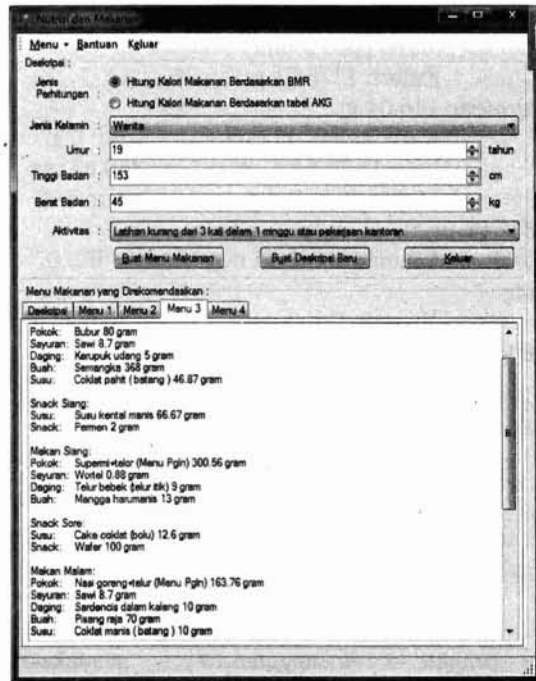
Nama Makanan	Porsi Makanan	Kandungan Vitamin A (µg)
Nasi goreng+telur	41.49 gram	123.18
Sawi	2.47 gram	183.78
Ayam goreng	2.84 gram	23
Semangka	13.08 gram	167.85

Cake coklat (bolu)	2.84 gram	2.13
<b>Total Kandungan Vitamin A</b>		<b>499.94</b>

Gambar 4 menunjukkan form utama yang ada pada aplikasi pembuatan menu makanan. Pada form ini, pengguna aplikasi dapat membuat menu makanan, melihat menu bantuan, ataupun pindah ke form data makanan dan rekomendasi. gambar 5 menampilkan rekomendasi menu makanan.



Gambar 4. Form Utama Aplikasi



Gambar 5. Rekomendasi Menu Makanan

Untuk mengevaluasi fitur-fitur yang ada pada aplikasi ini, telah dilakukan beberapa pengujian dengan menggunakan teknik *black box* dan *white box testing*. Sebagai contoh pada gambar 6 ditunjukkan hasil pengujian untuk suatu sampel,



yang menunjukkan rekomendasi menu makanan yang diberikan oleh aplikasi mengandung jumlah kalori dan takaran nutrisi sesuai dengan anjuran yang diberikan.

Data Masukan
Jenis Kelamin: Wanita. Umur : 31 tahun. Tinggi Badan: 164 cm. Berat Badan: 48 kg. Aktivitas: Latihan kurang dari 3 kali dalam 1 minggu atau pekerjaan kantor. Jenis Perhitungan kalori yang digunakan: BMR
Keluaran 1: Jumlah kalori dan takaran nutrisi yang dianjurkan
Kalori Perhari yang dianjurkan: $1265.3 \times 1.4 = 1771.42$ Takaran Nutrisi yang dianjurkan: Protein: 50 g, Karbohidrat: 332 g. Serat: 30 g, Lemak: 73 g. Kalsium: 800 mg, Magnesium: 270 mg. Vitamin A: 500 µg, Vitamin B1: 1 mg. Vitamin B2: 1.1 mg, Vitamin B6: 14 mg. Vitamin B12: 2.4 mg, Vitamin C: 0 mg. Vitamin D: 5 µg, Vitamin K: 55 µg. Zat Besi: 26 mg.
Keluaran 2: Jumlah Nutrisi pada Rekomendasi Menu Makanan
Jumlah Nutrisi pada Menu Makanan ini adalah: Kalori: 1776.71 g, Protein: 116.04 g, Karbohidrat: 117.08 g, Serat: 0 g. Lemak: 70.64 g, Kalsium: 840.68 mg. Magnesium: 0 mg, Vitamin A: 835.08 µg. Vitamin B1: 2.35 mg, Vitamin B2: 0 mg. Vitamin B6: 0 mg, Vitamin B12: 0 mg. Vitamin C: 17.49 mg, Vitamin D: 0 µg. Vitamin K: 0 µg, Zat Besi: 21.09 mg.

Gambar 6. Contoh Hasil Pengujian

## 6. KESIMPULAN

Kesimpulan dari hasil analisis, disain dan implementasi aplikasi ini adalah sebagai berikut:

1. Aplikasi pembuatan menu makanan sehat berhasil dibuat dan diimplementasikan dengan menggunakan pendekatan penyelesaian masalah *Knapsack*, yaitu dengan menggunakan *program dinamis* dan algoritma *Greedy*. Aplikasi mampu memberikan rekomendasi menu makanan yang memiliki jumlah kalori dan takaran nutrisi yang sesuai dengan kebutuhan jumlah nutrisi per harinya.

2. Pencarian solusi dengan menggunakan *program dinamis* dan algoritma *Greedy* memiliki kecenderungan untuk selalu memberikan solusi perhitungan yang sama, untuk data makanan dan batasan kalori yang nilainya tidak berubah. Oleh karena itu, pada aplikasi ini ditambahkan pemilihan makanan secara acak, sehingga menu makanan yang dihasilkan dapat bervariasi untuk setiap perhitungannya.
3. Kecenderungan lainnya yang muncul pada pengimplementasian kedua algoritma tersebut adalah memberikan hasil perhitungan yang optimal hanya untuk salah satu variabel saja, yaitu antara jumlah kalori maksimal atau jumlah nutrisi optimal. Agar hasil menu makanan yang diberikan memiliki jumlah nutrisi yang sesuai dengan kebutuhan pengguna, terdapat pembagian porsi menu makanan, sehingga jumlah nutrisi yang dihasilkan tidak melebihi batasan takaran nutrisi per harinya.
4. Pada aplikasi pembuatan menu makanan ini, terdapat batasan nutrisi dalam perhitungan menu makanan dengan menggunakan tabel Angka Kecukupan Gizi tahun 2004, sehingga jumlah nutrisi yang ada pada menu makanan dapat dibatasi, dan jumlah nutrisi pada menu yang ditampilkan tidak berlebih.

## 7. DAFTAR PUSTAKA

- [1] H. Cormen, Thomas et al., 1990. Introduction to Algorithms. McGraw-Hill Book Company. New York.
- [2] Otten, Ralph, Lecture Material The Knapsack Problem, <http://www.es.ele.tue.nl/education/5MC10/Solutions/knapsack.pdf>, Tanggal Akses 22 Oktober 2010.
- [3] Wang, Yusu. Analysis of Algorithms – Fractional Knapsack Problem, <http://www.cse.ohio-state.edu/~yusu/courses/780/knapsack.pdf>, Tanggal akses 22 Oktober 2010.
- [4] Stare, Frederick J. et al. 1981. LIVING NUTRITION 4<sup>th</sup> Edition, John Wiley & Sons, United State of America.
- [5] Hartono, Andry. 1997. Asuhan Nutrisi Rumah Sakit. Penerbit Buku Kedokteran EOC, Jakarta.
- [6] Santoso, Denny. Pengaruh Metabolisme Terhadap Pembakaran Lemak. <http://www.dennysantoso.com/pengaruh-metabolisme-terhadap-pembakaran-lemak.html>, 20 Oktober 2010.
- [7] Soekirman, et al. PUGS-Pesan Untuk Gizi Seimbang. <http://gizi.net/pugs/PUGS-text1.do>, 5 Januari 2011.
- [8] Astriani. Pola Menu 1500 Kalori untuk Diet. <http://id.shvoong.com/medicine-and->

*health/alternative-medicine/1958014-pola-  
menu-500-kalori-untuk/, 21 Oktober 2010.*