ISSN: 2088-9658

LAMPIRAN B 3

# Prosiding KNTIA 2011

## KONFERENSI NASIONAL
Teknologi Informasi dan Aplikasinya

Fakultas Ilmu Komputer UNSRI
Palembang, 21-22 Oktober 2011

**Diterbitkan oleh:**
Jurusan Sistem Informasi Fakultas Ilmu Komputer
Universitas Sriwijaya, PALEMBANG 2011

## KOMITE PROGRAM

Prof. Dr. Ir. Aniati Murni (Universitas Indonesia)
Prof. T. Basaruddin, Ph.D (Universitas Indonesia)
Zainal A. Hasibuan, Ph.D (Universitas Indonesia)
Dr. Ir. Eko K. Budiardjo (Universitas Indonesia)
Prof. Dr. Suyoto, M.Cs (Universitas Atma Jaya Jogyakarta)
Dr. Djuniadi, MT (Universitas Negeri Semarang)
Retantyo Wardoyo, M.Sc, Ph.D (Universitas Gajah Mada)
Kridanto Surendro, Ph.D (Institut Teknologi Bandung)
Dr. Ir. Arry Akhmad Arman (Institut Teknologi Bandung)
Muhammad Faidzul Nasrudin, Ph.D (Universitas Kebangsaan Malaysia)
Dr. Yusuf Hartono (Universitas Sriwijaya)
Dr. Darmawijoyo (Universitas Sriwijaya)
Dr. Ir. Siti Nurmaini, M.T (Universitas Sriwijaya)
Jaidan Jauhari, M.T (Univsersitas Sriwijaya)

## TIM EDITORIAL

### PENANGGUNG JAWAB
Ali Ibrahim, M.T.

### KETUA PENYUNTING
Jaidan Jauhari, M.T.

### SEKRETARIS PEYUNTING
Ken Ditha Tania, M.Kom

### PENYUNTING PELAKSANA
Abdiansyah, M.Cs
Ahmad Rifai, M.T
Mgs. Afriyan Firdaus, M.IT
Apriansyah Putra, M.Kom
Ari Wedhasmara, M.TI
Bayu Adhi Tama, M.TI
Endang Lestari, M.T
Fathoni, MMSI
Julian Supardi, M.T
Mira Afrina, M.Sc
Rifkie Primartha, M.T
Rizka Dhini Kurnia, M.Sc
Sutarno, M.T
Yadi Utama, M.Kom
Yudha Pratomo, M.Sc

### ALAMAT REDAKSI
Kampus Program Diploma Komputer (PDK) Universitas Sriwijaya
Jl. Srijaya Negara Bukit Besar Palembang 30139
Telp (0711) 379249 Fax (0711)379248
E-mail : kntia11@kntia.org dan kntia2011@gmail.com

### PENERBIT
Jurusan Sistem Informasi
Fakultas Ilmu Komputer
Universitas Sriwijaya

## DAFTAR ISI


**Artificial Intelligence**

1 Peramalan Beban Listrik Jangka Pendek Melalui Pendekatan Statistik dan Soft Computing (Aide Gafar Abdullah, Yadi Mulyadi) A1

2 Dasar Pemodelan dan Navigasi *Flocking Mobile Robot* dengan Aplikasi Sensor Jarak (Tresna Dewi) A6

3 Aplikasi Robot Pengaman Rumah dengan Klasifikasi Citra Wajah (Pandapotan Siagian) A14

4 Aksesibilitas Game Komputer Untuk Visual Disability User (Yudi Prayudi) A18

5 Perangkat Lunak Penyusunan Jadwal Perkuliahan Menggunakan Algoritma Genetika (Murnawan, Rido Saputra) A23

6 Analisis Sistem Pendukung Keputusan Dalam Memilih Program Studi Menggunakan Logika Fuzzy (Hetty Rohayani. AH) A33

7 Co-Occurence Matrix dan Jaringan Syaraf Tiruan Propagasi Balik untuk Deteksi Manusia (Indah Agustien Siradjuddin, Fitri Damayanti) A38

8 Penerapan SMS Gateway Generator Menggunakan Metode *Breadth-First Search* (Ken Ditha Tania) A42

9 Perancangan *Problem-Solving Environment* Menggunakan*Case-Based Reasoning* (Jaidan Jauhari, Abdiansah) A46

10 Pelacakan Jarak Untuk Navigasi Gerak Robot (Ahyar Supani) A50

11 Penyelesaian Game Puzzle Hashiwokakero dengan Teknik Solving Hashi dan Depth First Search (Eriska Amrina Pratiwi, Rusdi Efendi, Reza Firsandaya Malik) A54

12 Analisis Translasi Kata Makna Ganda pada Mesin Translasi Berbasis Statistik (Ekawati Prihatini) A58

13 Navigasi Perangkat Lunak eBook Reader Berbasis Mobile Menggunakan Fitur Face Tracking (Rikza Azriyan, Saparudin, Reza Firsandaya Malik) A66

14 Otomatisasi Palang Pintu Rel Kereta Api Menggunakan Sensor Strain Gauge (Sutarno dan Meicilia Grennatha) A71

15 Implementasi Mobile Computing dalam Pengembangan Sistem Pakar untuk Mengukur Tingkat Kecemasan Karyawan (Ali Ibrahim) A77

16 Identifikasi Personal Berdasarkan Tulisan Tangan Menggunakan Pengenalan Pola Biomimetik (Samsuryadi) A83

**Software Engineering**

1 Evaluasi Kualitas Website Pemerintah Daerah dengan Menggunakan Webqual (Studi Kasus Pada Kabupaten Ogan Ilir) (Candra Irawan) B1

2 Sebuah Kajian Tentang *Requirements Recovery* Pada Area Riset *Reverse Engineering* (Elviawaty Muisa Zamzami, Eko Kuswardono Budiardjo) B9

3 Perancangan *Mobile Learning* Berbasis J2ME untuk Penyediaan Bahan Ajar Perkuliahan Program Studi Pendidikan Ekonomi (Yenni Anggrayni, S.Pd) B13

4 Konsep Penerapan E-Pemilu Berintegrasi Dengan E-KTP di Kota Palembang (Heri Wijaya, M Yusuf, Yoga Prima Hendar) B19

5 Pengembangan Bahan Ajar Kewarganegaraan dengan Menggunakan Media Pembelajaran Berbasis WEBBLOG (Studi Kasus : Sekolah Menengah Atas Arinda Palembang) (Susi Oktaria) B23

6 Ontologi OWL untuk merepresentasikan Framework GSP pada GORE (Imam M. Shofi, Eko K. Budiardjo) B27

7 Training Evaluation on Website Construction as a Public Information Media for Government High School in Palembang (Bambang Tutuko, Siti Nurmaini, Rossi Passarella, Puspa Kurniasari, Candra Setiawan) B36

8 Penyandian SMS pada Telpon Selular dengan Hybrid Cryptosystem (Mega Mulia, M.T, Erwin, M.Si, Anton Sujarwo, S.Kom) B41

Information System

1 Implementasi Hibernate Pada Aplikasi Penentuan Kualitas Jasa Dengan Metode Quality Function Deployment (Jasman Pardede) C1

2 Penggunaan Jejaring Sosial Dalam Mendukung Jaringan Antar Perguruan Tinggi (M. Miftha Farid, Dedi Rianto) C7

3 Aplikasi Zelio Soft 2 Pada Sistem Keamanan Smart Room Menggunakan Smart Relay *(Nyayu Latifah Husni, Ade Silvia Handayani, Rani Utami)* C13

4 Usulan Peningkatan Kualitas Pada Proses Pelayanan Order Taksi di PT. Blue Bird Group Bandung dengan Menggunakan Metode Six Sigma *(Budi Ari Setianto, Amelia Kurniawati ST., MT., Pratya Poeri Suryadhini ST., MT)* C20

5 Analisis dan Perancangan Sistem Informasi Penjadwalan dan Manajemen Training pada Software Laboratory Center Universitas Bina Nusantara (Meiliana, Dwi Handri Kurniawan) C29

6 Sistem Pendukung Keputusan Menggunakan Metode *Technique for Order by Similarity to Ideal Solution* (TOPSIS) (Murnawan, Akhmad Fadjar Siddiq) C33

7 Rancangan Aplikasi *Customer Service pada PT. Lancar Makmur Bersama* (Suryasari , Astrid Callista, Juwita Sari) C39

8 Pengembangan Multimedia Pembelajaran Interaktif pada Mata Kuliah Komputer Akuntansi Menggunakan Program MYOB (Dra. Siti Fatimah, M.Si) C43

9 Pengaruh *BlackBerry* dalam Komunikasi sosial di Kalangan Mahasiswa (Tamin Soim[1], Wandri Ramadhoni) C49

10 Analisa Perbandingan Kualitas Belajar - Mengajar Antara Metode Face to Face dan Video Conference (Yoice Silitonga, Eminency D V) C57

11 Analisis Keputusan Pelanggan pada Sistem E-Ticket Maskapai Penerbangan Garuda Indonesia (Venera Genia, Alvin Pranata) C62

12 Pengembangan Media Pembelajaran Berbasis Komputer Untuk Meningkatkan Kualitas Pembelajaran di SMA Patra Mandiri 1 Plaju (Sri Anggraini Dewi, Widda Liyana) C69

13 Pengaruh Penggunaan Multimedia Berbasis Komputer terhadap Minat Belajar Mahasiswa dalam Pembelajaran Statistika Dasar Jurusan Sistem Informasi FASILKOM UNSRI (Oktober 2011) (Dwi Citra Hartini , Sela Taramita) C76

| | | |
|---|---|---|
| [illegible] | Analisis Usability Pada Aplikasi Berbasis Web Dengan Menggunakan Model Kepuasan Pengguna (Studi Kasus Fakultas Ilmu Komputer Unsri) (Fenty Tri Hidayati , Mamik Susanti) | C80 |
| [illegible] | Pengembangan Model Perpustakaan Berbasis Teknologi Informasi untuk Meningkatkan Kinerja Layanan Perpustakaan dan mewujudkan perpustakaan ideal berbasis digital di Ilmu Komputer Unsri (Ali Ibrahim, Mira Afrina) | C87 |
| [illegible] | Pengembangan Model Sistem Informasi *Integrated Laboratory* pada Perguruan Tinggi (Ali Ibrahim1, Endang Lestari) | C95 |
| [illegible] | Pengembangan Model Transaksi *Cunsomer* to *Cunsomer* dalam membangun *E-Marketing* untuk membantu proses perluasan segmen pasar (Apriyansyah Putra, Alvibianto Sidik) | C101 |
| [illegible] | Solusi Pengembangan Sistem Monitoring Perkuliahan Berbasis Web untuk Optimalisasi Proses Monitoring Perkuliahan di Fakultas Ilmu Komputer Universitas Sriwijaya (Mgs. Afriyan Firdaus, Firdaus, Ali Bardadi) | C110 |
| [illegible] | Usulan Pengembangan Sistem Informasi Klinis pada Pelayanan Pasien Rawat Jalan Rumah Sakit Bhayangkara Palembang (Desi Pibriana, Mgs. Afriyan Firdaus, Rizka Dhini Kurnia) | C115 |
| [illegible] | Analisis Sistem Informasi Akademik pada SMA Negeri 1 Palembang (Eka Puspita Sari, Mgs. Afriyan Firdaus, M.IT, Yudha Pratomo, S.T., M.Sc) | C122 |
| [illegible] | Usulan Pengembangan Sistem Informasi Pemetaan Blok Pemakaman pada Tempat Pemakaman Umum Kota Palembang Berbasis Web (Siska Irawan, Mgs. Afriyan Firdaus, Dwi Rosa Indah) | C130 |
| [illegible] | Pengembangan Aplikasi Multimedia Sebagai Media Alternatif Penyampaian Informasi Akademik (Ahmad Rifai) | C136 |
| [illegible] | Rancang Bangun Data *warehouse* untuk Menunjang Evaluasi Akademik Fakultas (Mukhlis Febriady, Bayu Adhi Tama) | C140 |
| 24 | IS/IT Strategy Planning : A Case Study of A Faculty of State University (Ari Wedhasmara, Bayu Adhi Tama, Apriansyah Putra) | C144 |
| 25 | Effectiveness of E-Learning Implementation At UTM (Rizka Dhini Kurnia, Azizah Abdul Rahman) | C149 |
| 26 | Penerapan Metode Weighted Product Model untuk Seleksi Calon Karyawan (Sri Lestari) | C152 |
| 26 | Analisis Dan Perancangan Aplikasi Customer Relationship Management Untuk Modul Penanganan Jaminan Purna Jual Terminal Smartphone Pada PT. X (Aldhi Jaya dkk) | C156 |
| 27 | Aplikasi Metode Tabu Search Pada Penyelesaian Assignment Problem (Rusdi Efendi, Ernawati) | C160 |
| 28 | Perancangan Aplikasi Material Requirement Planning (MRP) Untuk Produk Makanan Dengan Batas Kadaluarsa ( Studi Kasus di PT. H )(Putri W, dkk) | C166 |


**Computer Based Learning**


| | | |
|---|---|---|
| 1 | Implementasi Algoritma Dijkstra Dalam Aplikasi untuk menentukan Lintasan Terpendek Jalan Darat Antar Kota di Sumatera Bagian Selatan (1 Fitria, 2 Apri Triansyah) | D1 |
| 2 | Optimasi Tata Letak Fasilitas Departemen Spinning 3 Pt. Grandtex Dan Perancangan Aplikasi Untuk Modifikasi Algoritma Craft Dan Planet (1 Kriswardani Saptaningtias, 2 Dr Luciana Andrawina) | D8 |
| 3 | Pembelajaran Bahasa Inggris untuk Anak-anak Berbasis Multimedia (Ken Ditha Tania, Ali Ibrahim) | D13 |

# Aplikasi Metode Tabu Search Pada Penyelesaian Assignment Problem

**Rusdi Efendi, Ernawati, Safta Hastini**
*Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu*
*Jl. W.R Supratman Kandang Limun Bengkulu*
*Bengkulu 38371 A*
*Telp. (0736) 344087, 21170 - 227*

*Tujuan dari penelitian ini adalah untuk membangun metode Tabu Search pada penyelesaian assignment dan untuk mengetahui sejauh mana pengaruh parameter n, yaitu banyaknya sumber daya dan terhadap waktu. Assignment problem merupakan aplikasi sumber daya yang berkaitan dengan pemilihan daya yang tepat dalam menyelesaikan tugas tertentu, hasil yang didapat*
*Oleh karena itu, aplikasi ini diharapkan dapat bagi para pelaku industri untuk mengambil kebijakan dan memilih sumber daya yang terbatas untuk tugas dengan satu sumber daya hanya satu tugas dan satu tugas hanya dikerjakan oleh daya. Pada penelitian ini, sistem yang akan menggunakan salah satu metode heuristik, yaitu Search, bahasa pemrograman Delphi 7, metode sistem waterfall, dan perancangan sistem DFD. pengujian menunjukkan bahwa ada pengaruh jumlah sumber daya, jumlah tugas dan maksimum terhadap waktu.*

***Kunci: Assignment Problem, Metode Tabu Search, Waterfall, DFD***

## I. PENDAHULUAN

Perkembangan ilmu pengetahuan saat ini, menuntut para industri untuk dapat mengambil keputusan dan kebijaksanaan dan tindakan secara ilmiah atau dengan IPTEK bagaimana merancang dan menjalankan manusia-mesin secara terbaik walaupun menghadapi pengalokasian sumber daya yang terbatas. Masalah muncul bilamana harus dipilih tingkat pekerjaan yang bersaing dengan sumber daya manusia yang untuk melaksanakan pekerjaanpekerjaan tersebut hasil yang didapat akan optimal.

Umumnya tingkat keterampilan, pengalaman kerja, latihan pendidikan, dan latihan setiap pekerja berbeda. Sehingga dalam waktu penyelesaian yang sama itu berbeda juga. Masalah manajemen bagaimana menetapkan pemberian sejumlah tugas pekerja atas dasar satu-satu dengan cara yang satu tugas dikerjakan oleh satu pekerja), dimana pekerja mempunyai tingkat efisiensi yang berbeda-beda tugas yang berbeda-beda pula dan pekerja tingkat efisiensi yang sama dalam satu tugas berbeda untuk tugas yang lain.

Operasi (*Operations Research)* merupakan suatu pemecahan masalah yang menerapkan metode-ilmiah terhadap masalah-masalah rumit yang muncul dalam pengarahan dan pengelolaan dari suatu sistem yang terdiri dari manusia, mesin, bahan dan uang dalam industri, bisnis, pemerintahan dan pertahanan …"[6]. Pada Riset Operasi terdapat permasalahan dalam pemilihan personal yang tepat dalam mengerjakan suatu tugas tentu yang berpengaruh untuk suksesnya suatu usaha. Masalah ini disebut *assignment problem* atau masalah penugasan yang merupakan suatu kasus khusus dari masalah *linear programming* tentang pembagian sumber daya dan tugas, dimana sejumlah *m* sumber daya ditugaskan kepada sejumlah *n* tugas. Sumber daya adalah pekerjaan (pekerja/agen) sedangkan yang dimaksud tugas adalah mesin-mesin. Jadi, *assignment problem* akan mencakup sejumlah *m* sumber daya yang mempunyai *n* tugas dengan *Cij* biaya penugasan yang bertujuan menghemat biaya/waktu pada kasus minimasi atau mendapatkan keuntungan maksimum pada kasus maksimasi. Dalam hal ini, jumlah *m* sumber daya sama dengan jumlah *n* tugas.

Dari sudut pandang masalah optimasi, penyelesaian *assignment problem* bertujuan untuk memetakan *m* sumber daya terhadap *n* tugas berdasarkan penugasan satu-ke-satu, karena secara umum satu orang hanya bisa mengerjakan satu tugas dalam waktu bersamaan. Walaupun untuk menyelesaikan masalah ini bisa menggunakan metode umum yaitu dengan cara permutasi dari sejumlah *n* sumber daya dengan sejumlah *n* tugas, sehingga akan diperoleh *n! (n faktorial)* alternatif. Metode ini mudah dilakukan kalau *n* kecil, tetapi kalau sudah menyangkut untuk *n* yang besar cara ini kurang efisien, karena harus mencari alternatif dari *n!* buah kemungkinan yang harus dipilih. Sehingga besarnya sejumlah *n* yang harus dipetakan akan berdampak terhadap waktu yang digunakan untuk proses pengalokasian tugas tersebut.

Untuk melakukan suatu tugas, sumber daya harus dipilih berdasarkan *skill* yang dimiliki, yaitu dipilih sesuai dengan probabilitas sumber daya yang tinggi dimana banyaknya sumber daya yang harus dipilih untuk mengerjakan tugas tertentu berpengaruh terhadap waktu pengalokasian tugas tersebut. Karena itu, permasalahan ini merupakan permasalahan optimasi kombinatorial (*combinatorial optimamization*) yang kompleks atau permasalahan *np-hard*, yaitu suatu permasalahan yang pencarian solusinya (waktu komputasinya) akan naik seiring dengan naiknya skala parameter *n* secara linear. Oleh karena itu, pada penelitian ini digunakan metode heuristik untuk pemecahan *assignment problem*. Metode heuristik adalah metode pencarian solusi permasalahan optimasi kombinatorial yang cukup sulit dan berskala besar dengan cara mencari *good solution* yang dapat mencapai semua kriteria dengan waktu yang relatif kecil[2].

Salah satu metode heuristik yang dapat digunakan adalah metode *Tabu Search*. "Metode *Tabu Search* merupakan suatu metode optimasi yang menggunakan *short-term memory* untuk menjaga agar proses pencarian tidak terjebak pada nilai *optimum local"*[5]. Metode ini bertujuan untuk mengefektifkan proses pencarian solusi terbaik dari suatu permasalahan optimasi kombinatorial yang berskala besar. Untuk menjaga agar solusi terbaik tidak hilang, maka metode *Tabu Search* menyimpan solusi terbaik ke dalam *memory* yang disebut *tabu list,* serta terus mencari solusi terbaik dari setiap iterasi, kemudian dilakukan perbandingan antara solusi terbaik yang baru dievaluasi dengan solusi terbaik sebelumnya. Selain itu, metode ini mengingat atau melakukan
pengecekan solusi yang pernah ditemui pada *memory* (*tabu list*) dan melarang untuk menggunakan solusi yang telah dievaluasi untuk menghindari pengulangan yang sia-sia. Hal ini yang membuat metode *Tabu Search* menjadi lebih efisien dalam hal waktu[8].

Penelitian ini bertujuan untuk menerapkan metode *Tabu Search* dalam menyelesaikan *assignment problem* yang bisa menangani sejumlah *n* dengan waktu yang relatif singkat. Oleh karena itu, untuk mencapai tujuan tersebut, maka dibangun aplikasi metode *Tabu Search* pada penyelesaian *assignment problem.*

## II. Landasan Teori

### 2.1. Assigment Problem

Assignment problem atau masalah penugasan pertama kali dikenalkan oleh seorang ahli matematika dari Hongaria yang bernama D Konig pada tahun 1916. "Assignment problem adalah jenis khusus linear programming dimana sumber-sumber dialokasilan kepada kegiatan-kegiatan atas dasar satu-satu (one-toone-basis)…"[3].

*Assignment problem* merupakan masalah pengalokasian sejumlah *m* sumber daya yang ditugaskan kepada sejumlah *n* tugas (satu sumber daya untuk satu tugas) sehingga diperoleh biaya/waktu total yang minimum atau keuntungan total yang maksimum. Dalam hal ini, sumber daya adalah pekerjaan (pekerja/agen), sedangkan yang dimaksud dengan tugas adalah mesin-mesin. Jadi, ada *m* sumber daya yang ditugaskan kepada *n* tugas, dimana sumber daya *i (i = 1, 2, … , m)* ditugaskan kepada tugas *j (j = 1, 2, … , n)* dengan *Cij* biaya penugasan.

Pada *assignment problem* satu sumber daya hanya ditugaskan kepada satu tugas, maka *ai* kapasitas penawaran pada setiap sumber daya yang tersedia hanya 1. Demikian pula halnya dengan tugas, karena satu tugas hanya dikerjakan oleh satu sumber daya, maka *bj* kapasitas permintaan pada setiap tugas yang tersedia hanya 1. Dengan biaya pengalokasian *m* sumber daya ke *n* tugas sebesar *Cij.* sehingga variabel keputusan *Xij* bernilai 0 atau 1 untuk sumber daya *i= 1, 2, …, m* yang akan ditugaskan ke *j = 1, 2, …., n* [3]. Secara matematis dapat dinyatakan sebagai berikut:

$$X_{ij} = \begin{cases} 0, \text{jika sumber daya ke} - i \text{ tidak melakukan tugas ke} - j \\ 1, \text{jika sumber daya ke} - i \text{ melakukan tugas ke} - j \end{cases}$$

Berikut ini gambaran umum penyelesaian *assignment problem* dapat dilihat pada Tabel 1 di bawah ini:

**Tabel 1.** Gambaran Umum Penyelesaian *Assignment Problem*

| Sumber daya | Tugas | | | |
|---|---|---|---|---|
| | 1 | 2 | … | n |
| 1 | $C_{11}$ $X_{11}$ | $C_{12}$ $X_{12}$ | … | $C_{1n}$ $X_{1n}$ |
| 2 | $C_{21}$ $X_{21}$ | $C_{22}$ $X_{22}$ | … | $C_{2n}$ $X_{2n}$ |
| … | … | … | … | |
| m | $C_{m1}$ $X_{m1}$ | $C_{m2}$ $X_{m2}$ | … | $C_{mn}$ $X_{mn}$ |
| $b_j$ | 1 | 1 | … | 1 |

Dengan model *assignment problem* sebagai berikut:
Fungsi tujuan (maksimum/minimum)

$$Z = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} X_{ij}$$

Batasan-batasan

$$\sum_{i=1}^{m} x_{ij} = 1; j = 1,2,\ldots,m$$

$$\sum_{i=1}^{n} x_{ij} = 1; j = 1,2,\ldots,n$$

dan $x_{ij} = 0$ atau 1untuk *i =1, 2, …, m* dan *j=1, 2, …, n*
Keterangan dari model *assignment problem* di atas:

1. Fungsi tujuan adalah *Z* sebagai total [illegible] penugasan yang menyatakan apakah [illegible] *problem* ini mencari nilai minimum atau maksimum.
2. Variabel keputusan adalah *Xij.* Nilai dari *Xij* [illegible] untuk *i = 1, 2, …, m* dan *j = 1, 2, …, n.* [illegible] apabila *i* sumber daya ditugaskan untuk *j* [illegible] bernilai 0, apabila *i* sumber daya tidak [illegible] untuk *j* tugas.
3. Tetapan biaya yang telah ditentukan adalah [illegible] sumber daya yang ditugaskan kepada *n* tugas.
4. Jumlah sumber daya adalah *m.*
5. Jumlah tugas yang akan diselesaikan adalah *n.*

Variabel keputusan *Xij* bernilai 0 atau 1. Dalam [illegible] untuk menyelesaikan *assignment problem* dicari [illegible] yang bernilai 1 dimana variabel *Xij* bernilai 1 [illegible] sumber daya ditugaskan ke *n* tugas yang [illegible] berdasarkan keputusan saat *Xij* terpilih menjadi [illegible] penugasan pada setiap pencarian solusi dengan [illegible] penugasan, sehingga angka tersebut dikalikan dengan [illegible] terpilih dari sel tersebut (*Xij* Cij).* Sedangkan [illegible] yang tidak terpilih maka nilai *Xij* bernilai 0 karena [illegible] daya tidak terpilih untuk melakukan *n* tugas. [illegible] fungsi tujuan *Z* dapat digambaran sebagai berikut:

$$Z = |(C_{11}.X_{11}) + (C_{12}.X_{12}) + \cdots + (C_{1n}.X_{1n}) \\ |(C_{21}.X_{21}) + (C_{22}.X_{22}) + \cdots + (C_{2n}.X_{2n}) \\ |(C_{m1}.X_{m1}) + (C_{m2}.X_{m2}) + \\ (C_{mn}.X_{mn})|$$

Berdasarkan Persamaan (4), misalnya [illegible] kolom pertama terpilih dengan *C11* biaya penugasan [illegible] variabel keputusan *X11* bernilai 1 sedangkan *X1j* [illegible] pertama atau kolom pertama lainnya bernilai 0 [illegible] kapasitas penawaran setiap sumber daya dan [illegible] permintaan setiap tugas hanya bernilai 1 karena [illegible]

satu tugas dan sebaliknya satu tugas untuk satu ya. Demikian juga untuk baris kedua kolom kedua nya. Sehingga nilai $Z$ menjadi:

$$\cdot X_{11}) + (0) + \cdots + (0)| + |(0) + \cdot X_{22}) + \cdots + (0)| + |(0) + (0) + \cdots + \cdot X_{mn})| \quad (5)$$

arkan penjelasan di atas, dapat disimpulkan signment *problem* adalah mencari $Z$ total biaya m/minimum) dari $Cij$ biaya yang telah ditentukan tiap $m$ sumber daya yang akan ditugaskan kepada $n$ ngan $ai$ kapasitas penawaran pada setiap sumber $bj$ kapasitas permintaan pada setiap tugas bernilai ga variabel keputusan $Xij$ bernilai 0 atau 1. Dalam icari $Cij$ yang terpilih dengan variabel keputusan bernilai 1 yaitu sumber daya yang mengerjakan leh karena itu, $ai$ kapasitas penawaran pada setiap nya dan $bj$ kapasitas permintaan pada setiap tugas iabel keputusan $Xij$ tidak perlu dicantumkan. tabel gambaran umum penyelesaian *assignment* dapat disederhanakan menjadi tabel ongkos berikut

2. Tabel Ongkos pada *Assignment Problem*

| | 1 | 2 | ... | n |
|---|---|---|---|---|
| 1 | $C_{11}$ | $C_{12}$ | ... | $C_{1n}$ |
| 2 | $C_{21}$ | $C_{22}$ | ... | $C_{2n}$ |
| . | . | . | ... | . |
| . | . | . | ... | . |
| . | . | . | ... | . |
| m | $C_{m1}$ | $C_{m2}$ | ... | $C_{mn}$ |

**de *Tabu Search***

de *Tabu Search* adalah suatu metode optimasi yang termasuk ke dalam kelas *local search*. *abu Search* memperbaiki performansi *local search* emanfaatkan struktur *memory*[8]. Definisi lainnya an bahwa metode *Tabu Search* adalah suatu

yang menggunakan *short-term memory* untuk agar proses pencarian tidak terjebak pada nilai *local"*[4].

dasar penyelesaian metode *Tabu Search* adalah ifan proses pencarian solusi dengan cara mencari *ion* pada setiap tahap pelacakan. Pada beberapa ucakan dapat dikategorikan sebagai langkah *tabu* karena akan menghasilkan *local optimal* dan juga an mengakibatkan langkah pengulangan kembali ke solusi yang pernah ditemukan sebelumnya *nt)*. Langkah-langkah ini kemudian dimasukkan daftar yang disebut dengan *tabu list*. *Tabu list* tempat untuk menyimpan sekumpulan solusi saja dievaluasi. Proses pencariannya dilakukan cara menentukan solusi awal dan kemudian gerakan *(move)* ke solusi-solusi berikutnya dan enti sampai kriteria penghentian *(stopping* tercapai. Kriteria penghentian metode *Tabu* isalnya sejumlah iterasi yang ditentukan oleh *user*, waktu CPU tertentu, atau sejumlah iterasi tanpa peningkatan nilai fungsi objektif terbaik, ainya[7]. Metode *Tabu Search* bekerja secara enggunakan algoritma *local search* pada setiap k mencari solusi diantara sebagian tetangga dari solusi terbaik saat ini. Pada setiap iterasi, algoritma *local search* memilih solusi tetangga dengan total biaya/waktu yang minimal pada kasus minimasi atau total keuntungan maksimum pada kasus maksimasi tergantung dari kasus yang dihadapi.

Untuk efesiensi *memory* dan waktu proses, *tabu list* hanya menyimpan *move* yang merupakan kebalikan dari langkah yang telah digunakan pada iterasi-iterasi sebelumnya dengan panjang *list* yang dibatasi oleh *user*. Metode *Tabu Search* mempunyai struktur *memory* yang menyimpan solusi terbaik ke dalam *tabu list*. Selain itu, metode ini mengingat solusi yang ada dengan cara melakukan pengecekan pada *tabu list* dan melarang menggunakan solusi yang pernah ditemui untuk menghindari pengulangan solusi yang sudah ada. Oleh karena itu, dengan menggunakan *tabu list*, metode *Tabu Search* dapat menerima solusi yang tidak memberikan peningkatan kualitas, sehingga metode *Tabu Search* bisa keluar dari *optimum local*. Akan tetapi, terdapat pengecualian dalam metode *Tabu Search*. Jika terdapat *move* yang sudah berada dalam *tabu list* (terlarang untuk dipilih) tetapi memberikan solusi yang lebih baik dibandingkan semua solusi terbaik yang pernah dievaluasi, maka *move* tersebut bisa diterima dan *move* tersebut harus dikeluarkan dari *tabu list* (dibebaskan dari larangan). Hal ini merupakan prioritas khusus pada *tabu list* yang disebut kriteria aspirasi atau kondisi aspirasi *(aspiration conditions)*. Algoritma *Tabu Search* secara garis besar dapat ditulis sebagai berikut[5]:

```
Langkah 0. Tetapkan:
           X=Matriks input berukuran mxn.
           MaxIter=maksimum iterasi.
Langkah 1. S=bangkitkan solusi secara random
Langkah 2. GlobalMin=FCost(S)
Langkah 3. Best=S
Langkah 4. TabuList=[ ]
Langkah 5. Kerjakan dari k=1  sampai
           MaxIter:
Langkah 6.  BestSoFar=FCost(S)
Langkah 7.  BestMove=S
Langkah 8.  Kerjakan dari i=1 sampai(n-1):
Langkah 9.   Kerjakan dari j=i sampai n:
Langkah 10.    L=Tukar(S[i],S[j]).
Langkah 11.    Cost=FCost(L)
Langkah 12.    Jika(L∉TabuList)atau
               (Cost<GlobalMin),
                 kerjakan:
Langkah 13.        Jika(Cost<BestSoFar),
                    kerjakan
Langkah 14.        BestSoFar=Cost.
Langkah 15.        BestMove=L.
Langkah 16. S=BestMove.
Langkah 17. Tambahkan S ke TabuList
Langkah 18. Jika BestSoFar<GlobalMin,
            kerjakan:
Langkah 19.   GlobalMin=BestSoFar.
Langkah 20.   Best=BestMove.

Solusi  akhir  adalah  Best,  dengan  cost
sebesar GlobalMin.
```

### 2.3. Pengukuran Waktu Kerja Sistem

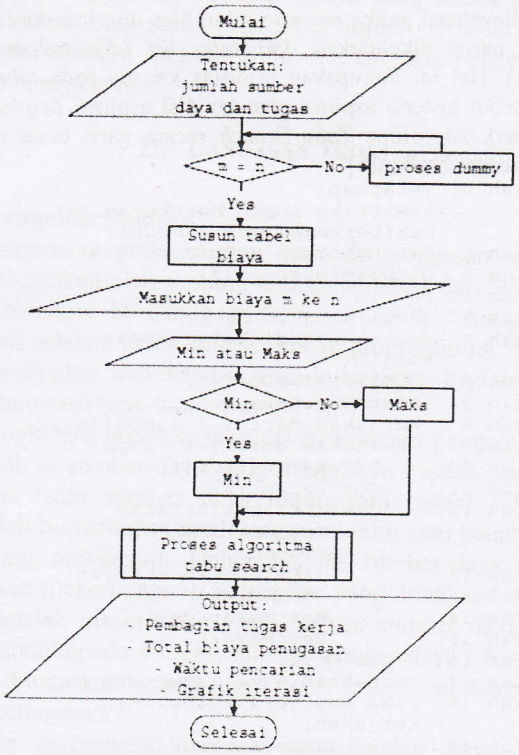
Pengukuran kerja sistem dilakukan untuk mendapatkan ukuran-ukuran tentang kinerja dan beban yang berlaku pada suatu sistem kerja. Kinerja adalah kecepatan atau pengukuran waktu kerja sistem yang akan dibuat (ukuran kuantitas untuk kualitas tertentu), sedangkan beban adalah yang dialami sumber daya (beban fisik dan beban psikososiologik)[7].

Pada penelitian ini, pengukuran waktu kerja sistem merupakan pengukuran waktu sejauh mana terdapat pengaruh parameter $n$, yaitu banyaknya sumber daya dan tugas, terhadap waktu pencarian yang digunakan untuk proses pengalokasian tugas kepada sumber daya dengan tujuan untuk mengetahui kinerja dari sistem yang telah dibangun. Selain itu, akan dilakukan analisis beban kerja.

Dalam penelitian ini, beban kerja adalah banyaknya sumber daya yang akan dialokasikan kepada tugas untuk mendapatkan solusi optimal. Waktu akan dicatat dengan menggunakan *clock* fisik yang terdapat pada komputer sebagai alat pengukur waktu dalam penyelesaian suatu aktivitas yang diamati. *Clock* adalah alat elektronik yang menghitung osilasi yang terjadi pada frekuensi tertentu, dan menyimpannya dalam *counter register*. Sistem operasi membaca *clock* fisik tersebut dan menerjemahkannya ke *software clock*. *Software clock* tidak selalu akurat sehingga perhitungan waktu *hardware* dan *software* memiliki perbedaan walaupun sangat kecil. Namun, *software clock* tetap menjadi acuan pencatatan waktu setiap kejadian proses[1].

## III. Perancangan FlowChart System

Diagram alir (*flowchart*) adalah suatu bagan (*chart*) yang menunjukkan aliran (*flow*) dalam program atau prosedur sistem secara logika[4]. *Flowchart* sistem aplikasi secara umum dapat dilihat pada Gambar 1 di bawah ini:




Gambar 1. Flowchart sistem secara umum

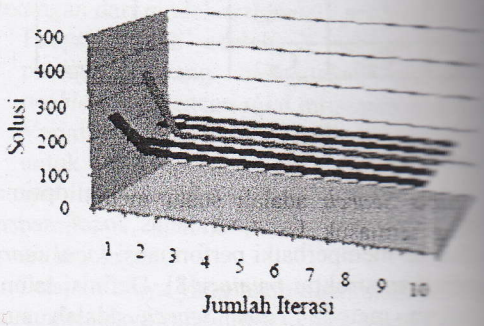
Pada gambar di atas penyelesaian *assignment problem* dimulai dari proses *input* jumlah *m* sumber daya dan jumlah *n* tugas. Setelah itu dilakukan pengecekan jumlah sumber daya *m* dan tugas *n*. Jika *m=n*, maka lakukan langkah selanjutnya, yaitu penyusunan tabel biaya dan peng-*input*-an biaya penugasan *m* ke *n*. Jika tidak (*m<n* atau *m>n*), maka lakukan proses *dummy* (proses penambahan *m* atau *n* sehingga menjadi sama dengan biaya penugasan *dummy* sama dengan nol (*Cij=0*). Kemudian, lakukan langkah seperti pada kasus *m=n* (penyusunan tabel biaya dan peng-*input*-an biaya penugasan Pilih kasus yang diselesaikan, kasus minimasi untuk meminimumkan biaya/waktu atau kasus maksimasi untuk memaksimumkan keuntungan. Dari data yang di-*input*, dilakukan pemrosesan dengan algoritma *Tabu Search* sampai mendapatkan *output*. *output* berupa penugasan kerja, total biaya penugasan (*Cij*) dan waktu pencarian dalam sekon tahap pemrosesan selesai dilakukan.

## IV. Pembahasan

Aplikasi ini dapat digunakan [illegible] seperti penugasan pekerja terhadap [illegible] dengan pekerjaan dan kasus umum [illegible] Dengan hasil dari pengujian aplikasi [illegible] pada penyelesaian *assignment* [illegible] *pengujian* jumlah sumber daya dan [illegible] bertambah untuk mengetahui solusi [illegible] pengujian jumlah ordo matriks, j [illegible] jumlah tugas untuk mengetahui [illegible] dibutuhkan sampai proses berhenti. [illegible] dilakukan sebanyak 5 kali.

### 4.1. Analisis Pengaruh Jumlah [illegible] yang Dihasilkan

Analisis ini dilakukan untuk melihat [illegible] iterasi terhadap solusi yang dihasilkan [illegible] 5 pada kasus minimasi pengujian [illegible] dengan jumlah iterasi 1, solusi yang [illegible] ke 266. Pada jumlah iterasi 3 dan seterusnya [illegible] yang dihasilkan semakin baik, yaitu [illegible] disimpulkan bahwa jumlah iterasi [illegible] kualitas solusi yang dihasilkan. Hal ini [illegible] kesempatan yang lebih banyak [illegible] pembangkitan solusi. Hasil pengujian [illegible] Grafik 1.

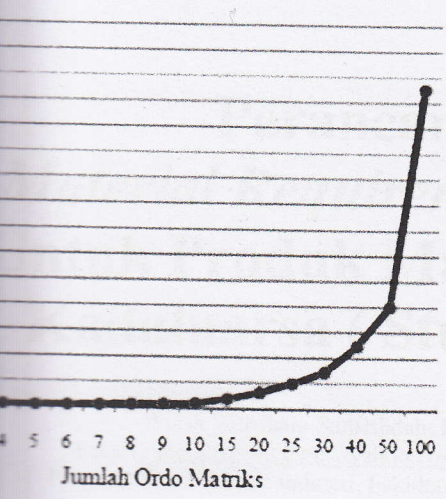



Grafik 1. Hasil Pengujian Pengaruh [illegible] terhadap Solusi yang Dihasilkan

### 4.2. Analisis Pengaruh Jumlah Ordo Matriks [illegible] Waktu Pencarian

Analisis ini dilakukan untuk melihat [illegible] ordo matriks dari 1 sampai 100 terhadap waktu [illegible] yang dibutuhkan untuk mendapatkan solusi. [illegible] dilakukan dengan 1 iterasi.

Berdasarkan Grafik 2 dapat dilihat bahwa [illegible] matriks berpengaruh terhadap waktu pencarian [illegible] dibutuhkan untuk mendapatkan solusi. Semakin [illegible] jumlah ordo matriks, maka semakin meningkat [illegible] pencarian yang dibutuhkan. Peningkatan tersebut [illegible] pada ordo matriks 1, rata-rata waktu pencarian [illegible] dibutuhkan adalah 0,038 sekon. Ordo matriks [illegible] pencarian yang dibutuhkan adalah 0,0468 sekon, dan [illegible] jumlah ordo matriks ditambah secara linear menjadi [illegible] dengan waktu pencarian yang dibutuhkan [illegible] 12,5054 sekon. Waktu pencarian terus meningkat secara linear yang di[illegible] untuk melakukan pembangkitan solusi dalam satu [illegible] menjadi bertambah.

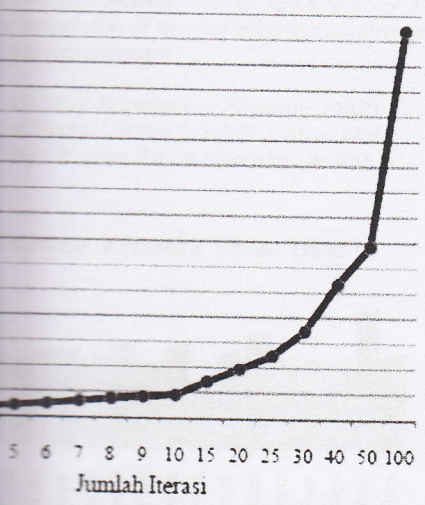



il Pengujian Pengaruh Jumlah Ordo
lap Waktu Pencarian untuk 1 Iterasi

**garuh jumlah iterasi terhadap waktu**

akukan dengan melihat pengaruh jumlah aktu pencarian yang dibutuhkan sampai si. Dalam *pengujian ini data* yang rdo matriks 10.

afik 3 dapat dilihat bahwa jumlah iterasi dap waktu pencarian yang dibutuhkan solusi. Pada pengujian dengan jumlah waktu pencarian adalah 0,1774 sekon, ata pencarian adalah 0,359 sekon, dan umlah iterasi ditambah menjadi 100, rata 30,7602 sekon. Sehingga dari data impulkan bahwa bertambahnya jumlah katkan waktu pencarian yang dibutuhkan solusi. Hal ini dikarenakan, jumlah an semakin banyak.

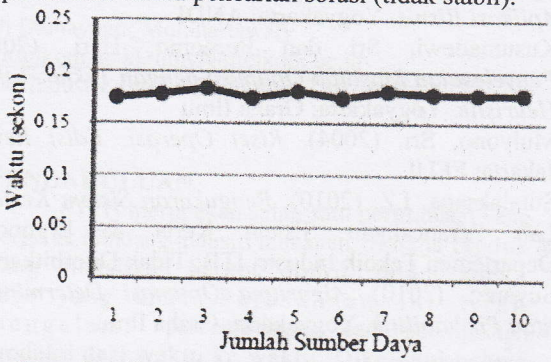



Pengujian Pengaruh Jumlah Iterasi
Pencarian untuk Ordo Matriks 10

**aruh jumlah sumber daya dan tugas u pencarian**

akukan untuk mengetahui pengaruh ugas terhadap waktu pencarian untuk Untuk mewakili analisis ini, maka rafik 4, yaitu analisis pengaruh *dummy* pada sumber daya terhadap waktu pencarian. Dengan jumlah sumber daya bertambah dari 1 sampai 10 dan jumlah tugas tetap, yakni 10 tugas.

Dari Grafik tersebut dapat dilihat bahwa jika jumlah sumber daya bertambah dan tugas tetap, maka akan mempengaruhi waktu pencarian yang dibutuhkan untuk mendapatkan solusi optimal. Semakin bertambah jumlah sumber daya, maka waktu pencarian yang dibutuhkan hampir sama dan tidak stabil. Hal ini terlihat pada jumlah sumber daya 1 sampai jumlah sumber daya 3, jumlah sumber daya 4 sampai 5, jumlah sumber daya 6 sampai 7, dan jumlah sumber daya 7 sampai 10 terjadi peningkatan dan penurunan waktu pencarian solusi (tidak stabil).




**Grafik 4. Hasil Pengujian Jika Jumlah Sumber Daya Bertambah dan Jumlah Tugas Tetap**

Dari seluruh pengujian di atas ditarik kesimpulan bahwa pada analisis solusi diketahui jumlah iterasi mempengaruhi solusi yang dihasilkan. Dengan bertambahnya jumlah iterasi akan memperbaiki kualitas solusi dari algoritma *Tabu Search*. Sedangkan pada analisis kerja sistem disimpulkan bahwa jumlah ordo matriks, jumlah iterasi, jumlah sumber daya, dan jumlah tugas mempengaruhi waktu pencarian untuk mendapatkan solusi optimal. Semakin bertambah jumlah ordo matriks dan jumlah iterasi, maka waktu pencarian yang dibutuhkan akan meningkat. Begitu juga dengan jumlah sumber daya dan jumlah tugas, semakin bertambah jumlah sumber daya atau jumlah tugas, maka waktu pencarian yang dibutuhkan untuk mendapatkan solusi optimal hampir sama atau tidak stabil.

## V. KESIMPULAN

Berdasarkan analisis dari aplikasi metode *Tabu Search* pada penyelesaian *assignment problem* yang telah dibangun, maka terdapat beberapa hal yang dapat disimpulkan, yaitu:

1. Penelitian ini menghasilkan aplikasi metode *Tabu Search* pada penyelesaian *assignment problem* yang cukup fisien dalam menyelesaikan kasus penugasan kerja.
2. Waktu pencarian naik seiring dengan bertambahnya jumlah ordo matriks secara linear.
3. Penyelesaian ordo matriks 1000 atau lebih dengan perangkat uji (komputer) yang digunakan pada penelitian ini belum bisa menyelesaikan *assignment problem*.
4. Kualitas solusi algoritma *Tabu Search* dipengaruhi jumlah iterasi.

## DAFTAR PUSTAKA

[1] Ardani,I, Hamdani,T, dan W Herusetyo,A. (2011). *Time and Global State.* http://te.ugm.ac.id/~risanuri/distributed /ringk/Bab10.pdf. Terakhir diakses: 6 Maret 2011

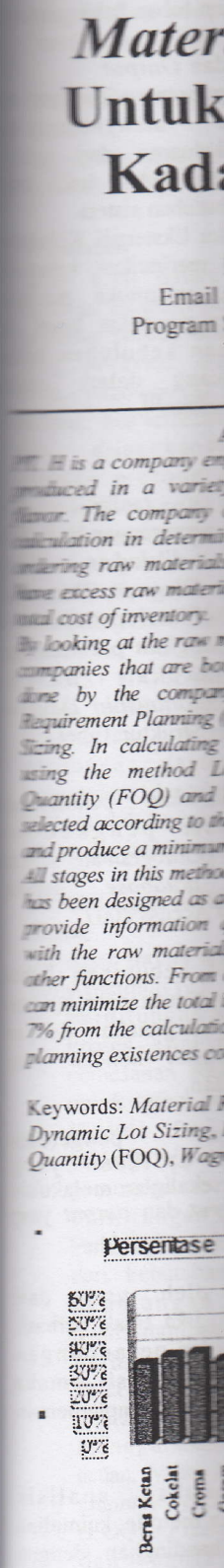
[2] Betrianis dan Aryawan, P. T. (2003). Penerapan Algoritma Tabu Search dalam Penjadwalan Job Shop. *Jurnal Teknologi Departemen Teknik Industri, Fakultas Teknik, Universitas Indonesia.* 7( 3), 107-112

[3] Hillier, F., Lieberman, G.J. (1990). *Introductionto Operations Research, fifth edition.* (alih bahasa Ellen Gunawan) Jakarta: Erlangga

[4] Jogiyanto. (2005). *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis.* Yogyakarta: ANDI

*[5]* Kusumadewi, Sri dan Purnomo, Hari. (2005). *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik.* Yogyakarta: Graha Ilmu

*[6]* Mulyono, Sri. (2004). *Riset Operasi. Edisi Revisi.* Jakarta: FEUI

*[7]* Sutalaksana, I.Z. (2010). *Pengukuran Sistem Kerja 1.* Lab Perancangan Sistem Kerja & Ergonomi. Departemen Teknik Industri ITB: Tidak Diterbitkan

*[8]* Suyanto. (2010). *Algoritma Optimasi Deterministik atau Probabilitik.* Yogyakarta: Graha Ilmu