

# PROSIDING SEMMAU 2016

**SEMINAR NASIONAL & KONFERENSI SISTEM INFORMASI,  
INFORMATIKA & KOMUNIKASI**

**TEMA: E-GOVERNMENT SEBAGAI DAYA DUKUNG  
TATA KELOLA PEMERINTAHAN**

**Kupang, 17 September 2016**

**BUKU 2**

**ISBN: 978-602-73628-0-3**



**STIKOM UYELINDO KUPANG**

# PROSIDING SEMMAU 2016

---

## KOMITE

### Penulis :

Pemakalah Seminar Nasional & Konferensi Sistem Informasi, Informatika & Komunikasi (SEMMAU 2016)

**ISBN : 978-602-73628-0-3**

### Komite Program :

Dr. Ir. Rila Mandala, M.Eng. (ITB)  
Dr. Achmad Nizar, S.Kom., M.Kom. (UI)  
Ir. Dana Indra Sensuse, M.Lis., Ph.D. (UI)  
Prof. Daniel Herman Fredy Manongga, M.Sc., Ph.D. (UKSW)  
Prof. Mustafid (UNDIP)  
Prof. Dr. Ir. Kuswara Setiawan, M.T. (UPH)  
Prof. Suyoto, P.hd

### Penyunting :

Max ABR. Soleman Lenggu. S.Kom., M.T.  
Marinus I.J. Lamabelawa, S.Kom., M.Cs  
Fransiska S.E. Atonis  
Floriany M. Owa  
Marmi Y. Taek  
Adirwan Tajudin  
Maystri R.A. Ta'eko  
Ahmad Suhendra

### Desain Sampul :

Max Lenggu

### Redaksi :

#### Dapur Semmau

Lembaga Penelitian, Publikasi dan Pengembangan pada Masyarakat  
Jl. Perintis Kemerdekaan 1, Kayu Putih, Kupang, NTT, Indonesia.  
Telp. (0380)8554501, Fax (0380) 8554501  
Email : [semmau@uyelindo.ac.id](mailto:semmau@uyelindo.ac.id)  
<http://www.semmau.uyelindo.ac.id>.

### Penerbit :

Sekolah Tinggi Manajemen Informatika & Komputer (STIKOM) Uyelindo Kupang.  
Jl. Perintis Kemerdekaan 1, Kayu Putih, Kupang, NTT, Indonesia.  
Telp. (0380)8554501, Fax (0380) 8554501  
Email : [stikom@uyelindo.ac.id](mailto:stikom@uyelindo.ac.id)  
<http://www.uyelindo.ac.id>.

*Cetakan kedua September 2016*

*Hak Cipta di Lindungi Undang-undang*

*Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin tertulis dari penerbit.*

## PROSIDING SEMMAU 2016

---

Segala puji dan syukur selayaknya tercurah kehadirat Allah Yang Maha Kuasa yang tanpa henti mengucurkan rahmat dan karuniaNya, baik kurunia sehat, rejeki, kecerdasan, kemauan, dan bahkan juga karunia dalam bentuk kesadaran dan kemampuan bersyukur kepadaNya, dan dengan ijinnya Prosiding Seminar Nasional dan Konferensi Sistem Informasi, Teknik Informatika, dan Komunikasi (SEMMAU) tahun 2016 dengan Tema “E-GOVERNMENT SEBAGAI DAYA DUKUNG TATA KELOLA PEMERINTAHAN”. dapat kami terbitkan.

Buku Prosiding ini berisi sekumpulan *Paper* dari hasil penelitian ilmiah yang telah diseleksi, untuk dipresentasikan dalam kegiatan Seminar Nasional dan Konferensi Sistem Informasi, Teknik Informatika, dan Komunikasi (SEMMAU) tahun 2016 dan bertempat di *Ballroom* Sotis Hotel Kupang Nusa Tenggara Timur pada tanggal 17 September 2016, kegiatan ini diikuti oleh peserta pemakalah yang berasal dari berbagai perguruan tinggi yang tersebar di kawasan Nusa Tenggara Timur (NTT), maupun di luar NTT, yang terdiri dari 26 makalah dari para peserta pemakalah.

Seminar Nasional yang bertemakan “E-GOVERNMENT SEBAGAI DAYA DUKUNG TATA KELOLA PEMERINTAHAN”. ini menghadirkan pembicara utama berkelas nasional yakni Prof. Dr. Ir. Ricardus Eko Indrajit, M.Sc., M.B.A.

Ucapan terima kasih kami sampaikan kepada Reviewer Paper dan pihak-pihak yang telah membantu penyelenggaraan Seminar Nasional dan Konferensi Sistem Informasi, Teknik Informatika, dan Komunikasi (SEMMAU) tahun 2016 ini. Semoga prosiding ini dapat bermanfaat dan dapat digunakan dengan sebaik-baiknya.

Akhir kata, jika ada yang kurang berkenan selama penyelenggaraan kegiatan seminar maupun dalam penerbitan buku prosiding ini mohon dimaafkan. Semoga apa yang telah kita lakukan ini bermanfaat bagi kemajuan bangsa dan negara dimasa depan. Amin.

Kupang, September 2016  
Panitia,

Yampi R. Kaesmetan

# PROSIDING SEMMAU 2016

---

## DAFTAR ISI

	Halaman
<b>BERBAGI PENGETAHUAN MENGGUNAKAN EDMODO BERBASIS SOCIALIZATION MODEL SECI (Studi Kasus : SMK Negeri 1 Boyolali).</b> <i>Dwi Kristiani, Eko Sedyono, Ade Iriani</i>	206 - 214
<b>ANALISIS TOPIK-TOPIK YANG MEMPENGARUHI TERJADINYA SENTIMEN TERKAIT KENAIKAN HARGA BAHAN BAKAR MINYAK (BBM) PADA MEDIA ONLINE"</b> <i>Bobby Christian Sandy, Danny Manongga, Ade Iriani.</i>	215 - 224
<b>IMPLEMENTASI E-GOVERNMENT DI INDONESIA.</b> <i>Dien Novita.</i>	225 - 229
<b>ADLER-32 INTEGRITY VALIDATION IN 24 BIT COLOR IMAGE.</b> <i>Andysah Putera Utama Siahaan.</i>	230 -235
<b>DESAIN UNTUK RANCANG BANGUN FITUR KEAMANAN.</b> <i>Wawan Nurmansyah, Masayu Jamilah.</i>	236 - 242
<b>PENERAPAN E-BISNIS UNTUK MENANGANI PROSES PENJUALAN PRODUK AGUAMOR BERBASIS WEB.</b> <i>Dewi Anggreini</i>	243 - 247
<b>KLASIFIKASI BELIMBING MENGGUNAKAN K-NEAREST NEIGHBORS (KNN) BERDASARKAN CITRA RED-GREEN-BLUE (RGB)</b> <i>Kana Saputra S, Fuzy Yustika Manik.</i>	248 - 251
<b>IMPLEMENTASI PEMECAHAN SLIDING TILE PUZZLE MENGGUNAKAN METODE HEURISTIK (ALGORITMA A*, IDA* DAN BDA*)</b> <i>Sabastianus A.S.Mola.</i>	252 - 259
<b>ANALISIS SISTEM ANTRIAN DENGAN METODE NEXT EVENT TIME ADVANCED MECHANISM (Studi Kasus: PT. ASDP Persero Cabang Kupang)</b> <i>Ardianus Wattileo, Marianus I.J. Lamabelawa</i>	260 - 264
<b>MODEL PENGUKUR BERAT BADAN TERNAK SAPI TIMOR BERBASIS CITRA</b> <i>Deddy B. Lasfet, Markus Daud Letik</i>	265 - 271
<b>PENERAPAN ELECTRONIC CUSTOMER RELATIONSHIP MANAGEMENT UNTUK PRODUK INDUSTRI RUMAH TANGGA (Studi Kasus: Kabupaten Rote Ndao)</b> <i>Wemmy A. Taka, Max ABR Soleman Lenggu.</i>	272 - 278

## PROSIDING SEMMAU 2016

---

<b>PENCARIAN LEMBAGA KURSUS PENDIDIKAN DI KOTA KUPANG BERBASIS LOKASI <i>PENIDAS</i>.</b>	279 - 283
<i>Nyongri E. Akulas, Edwin Malahina, Fransiskus Tjiptabudi.</i>	
<b>SORTASI TEKSTUR BIJI JAGUNG SEBAGAI BENIH TANAM MENGGUNAKAN SELF ORGANIZING MAP (Studi Kasus: Desa Bismarak Kabupaten Kupang Timur Provinsi Nusa Tenggara Timur).</b>	284 - 288
<i>Marlinda Vasty Overbeek</i>	
<b>MODEL SISTEM MONITORING DAN EVALUASI AKADEMIK MAHASISWA BERBASIS WEB (STUDI KASUS STIKOM UYELINDO KUPANG).</b>	289 - 294
<i>Rafliana Natalia da Silva, Marinus I.J. Lamabelawa, Semlinda Juszandri Bulan.</i>	
<b>PEMETAAN HASIL LAUT WILAYAH KABUPATEN ALOR DENGAN ANALISIS KELOMPOK.</b>	295 - 304
<i>Kristian Martiul Malbiyeti Tnunay, Remerta Noni Naatonis, Marlinda V. Overbeek.</i>	
<b>SISTEM PENDUKUNG KEPUTUSAN PENENTUAN CALON PENERIMA RUMAH BANTUAN MENGGUNAKAN METODE PROMOTHEE.</b>	305 - 308
<i>Dony M Sitohang</i>	
<b>SISTEM INFORMASI GOEGRAFIS POLA PENYEBARAN UMAT BERAGAMA DI KABUPATEN TIMOR TENGAH UTARA MENGGUNAKAN METODE K-MEANS CLUSTERING.</b>	309 - 317
<i>Yovita Hilda Helly, Yampi Kaesmetan, Mardhalia Saitakela.</i>	
<b>PERBANDINGAN PENGUKURAN JARAK DALAM PENENTUAN KUALITAS BENIH JAGUNG PULAU TIMOR DENGAN K-NEAREST NEIGHBOR.</b>	318 - 323
<i>Dessy Leonarti Pollo, Marlinda Vasty Overbeek, Franki Yusuf Bisilin</i>	
<b>APLIKASI EVALUASI TENAGA AHLI PESERTA SELEKSI NASIONAL MENGGUNAKAN METODE TOPSIS (Studi Kasus: Satker P2JN Provinsi Nusa Tenggara Timur).</b>	324 - 330
<i>Albert Adrian Bayu Mila1, Menhya Snae2, Franki Yusuf Bisilisin.</i>	
<b>LELANG ONLINE BERBASIS WEBSITE PADA PEGADAIAN CABANG OESAO</b>	331 - 340
<i>Adalberto Guterres, Benyamin Jago Belalawe, Mardhalia Saitakela</i>	
<b>EVALUASI KINERJA DOSEN DAN KARYAWAN DI STIKOM UYELINDO KUPANG BERBASIS WEBSITE.</b>	341 - 349
<i>Lukas H.J.E. Babu, Emanuel Safirman Bata, Marlinda Vasty Overbeek</i>	
<b>SISTEM INFORMASI GEOGRAFIS PEMETAAN KAWASAN KONSERVASI PERAIRAN DI PERBATASAN LAUT SAWU NUSA TENGGARA TIMUR.</b>	350 - 358
<i>M. Nurhudah, Yampi R. Kaesmetan, Remerta Noni Naatonis</i>	
<b>APLIKASI TUNTUNAN DOA SEHARI-HARI DIZIKIR DAN SUNNAH RASUL BAGI UMAT ISLAM BERBASIS ANDROID.</b>	359 - 364
<i>Mastura Masan, Emanuel Safirman Bata, Edwin A.U. Malahina</i>	

## PROSIDING SEMMAU 2016

---

<b>ANALISI PEMASARAN RUMPUT LAUT KECAMATAN SULAMU MENGUNAKAN METODE PERBANDINGAN EKSPONENSIAL (MPE).</b> <i>Melkianus Babis, Max ABR Soleman Lenggu.</i>	<b>365 - 369</b>
<b>PERBANDINGAN EKSTRAKSI TEKSTUR CITRA DENGAN METODE STATISTIK ORDE I DAN STATISTIK ORDE II UNTUK PEMELIHAN BENIH JAGUNG.</b> <i>Antonius Yosef Tampani, Petrus Katemba.</i>	<b>370 - 380</b>
<b>MANAJEMEN KINERJA KEPALA SEKOLAH DAN GURU DALAM PENINGKATAN MUTU PAUD.</b> <i>Hasibun Asikin</i>	<b>381 - 387</b>
<b>KAJIAN SITUS WEB RESMI PEMERINTAHAN KABUPATEN/KOTA NTT SEBAGAI WUJUD IMPLEMENTASI E-GOVERNMENT.</b> <i>Maria Yenita Soru, Yohanes Payong</i>	<b>388 - 393</b>

IMPLEMENTASI PEMECAHAN *SLIDING TILE PUZZLE* MENGGUNAKAN  
METODE HEURISTIK (ALGORITMA A\*, IDA\* DAN BDA\*)

Sebastianus A.S.Mola<sup>1</sup>

<sup>1</sup>Fakultas Sains Dan Teknik, Universitas Nusa Cendana  
<sup>1</sup>adie\_s\_mola@yahoo.co.id

Abstrak

*Sliding tile puzzle* adalah sebuah permainan acak angka, huruf atau gambar yang harus disusun ke dalam pola yang benar. Permainan ini biasanya terdiri dari 9 kotak angka, satu kotak kosong dan delapan kotak teracak. Tujuan permainan ini adalah menyelesaikan kotak yang sudah teracak menjadi sebuah susunan yang benar. Cara menyelesaikan puzzle ini dengan menggeser kotak – kotak yang ada sehingga mencapai puzzle kembali ke pola dengan angka berurut. Metode Heuristik (*Heuristik searching*) merupakan salah satu metode yang dapat digunakan untuk menyelesaikan permainan *slide tile puzzle* secara selektif, dengan memberikan solusi berupa jalur terpendek dan waktu yang efisien, sehingga memungkinkan pengguna dapat menyelesaikan permainan *puzzle* ini dengan baik, cepat dan relevan. Beberapa algoritma yang menggunakan fungsi heuristik adalah Algoritma A\* (*A Star*), Algoritma *Iterative Deepening A\** (IDA\*) dan Algoritma *Bi-directional A\** (BDA\*). Fungsi heuristik yang digunakan adalah dengan melihat banyaknya kotak pada posisi salah dan total keseluruhan jarak dari kotak yang berada di tempat yang salah untuk mencapai posisi yang benar. Tujuan dari pembuatan proyek akhir ini untuk mengimplementasikan pemecahan *Sliding Tile Puzzle* Menggunakan Metode Heuristik (Algoritma A\*, IDA\* dan BDA\*). Dari aplikasi ini dapat dilihat bahwa Algoritma A\*, IDA\* dan BDA\* mampu memberikan jalur dalam penyelesaian *Sliding Tile Puzzle*. Setelah pengujian berulang yang dilakukan dengan menggunakan 200 dataset, dapat dilihat perbandingan hasil antara Algoritma A\* dengan Algoritma IDA\* dan Algoritma BDA\* dimana Algoritma BDA\* menghasilkan jalur yang lebih pendek dalam menyelesaikan *Sliding Tile Puzzle*.

**Kata kunci:** *Sliding Tile Puzzle, Metode Heuristik, Algoritma A\*, Algoritma IDA, Algoritma BDA\**.

1. PENDAHULUAN

Program permainan (*game*) merupakan salah satu implementasi dari bidang ilmu komputer. Inti dari sebuah aplikasi permainan (*game*) adalah mengembangkan kemampuan otak untuk mengatur strategi, kecepatan, dan ketepatan dalam mencapai tujuan akhir. Perkembangan permainan pada masa kini sudah sangat pesat dengan jenis yang bermacam-macam dan tampilan yang menarik sehingga diminati oleh berbagai kalangan. Salah satu contoh permainan yang sering dimainkan adalah *sliding tile puzzle* atau yang biasa kita kenal dengan *game puzzle*. *Puzzle* terdiri dari beberapa jenis, ada yang menggunakan angka, huruf, dan gambar. Dalam permainan (*game*) *puzzle*, pemain diharapkan dapat mencapai tujuan akhir untuk membentuk sebuah *puzzle* menjadi sebuah angka atau pola yang benar dengan waktu yang cepat.

*Sliding tile puzzle* adalah sebuah permainan acak angka yang harus disusun ke dalam pola yang benar. Permainan ini biasanya terdiri dari kotak 3 x 3, satu kotak kosong dan delapan kotak teracak. Cara menyelesaikan *puzzle* ini dengan menggeser kotak-kotak yang ada sehingga mencapai *puzzle* kembali ke pola dengan angka berurut. Umumnya orang yang memainkan *puzzle* ini membutuhkan waktu yang cukup lama. Hal ini disebabkan karena

pada *slide puzzle* tidak ada informasi tambahan yang dimiliki untuk membantu melakukan pencarian solusi, untuk membentuk susunan *puzzle* seperti semula pada saat proses penyusunan. Untuk menyelesaikan permasalahan ini dibutuhkan suatu algoritma pencarian efektif yang dapat diterapkan.

Metode Heuristik (*Heuristik searching*) merupakan salah satu metode yang dapat digunakan untuk menyelesaikan permainan *slide tile puzzle* secara selektif, dengan memberikan solusi berupa jalur terpendek dan waktu yang efisien, sehingga memungkinkan pengguna dapat menyelesaikan permainan *puzzle* ini dengan baik, cepat dan relevan. Sehingga akan diterapkan 3 metode pencarian heuristik berupa pencarian satu arah algoritma A\* (*A Star*) dan algoritma *Iterative Deepening A\** (IDA\*), serta pencarian dua arah algoritma *Bi-directional A\** (BDA\*).

2. KAJIAN LITERATUR DAN  
PEGEMBANGAN HIPOTESIS

2.1 Kecerdasan Buatan

*Artificial Intelligence* (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Pernyataan tersebut juga dapat dijadikan definisi dari AI.

Definisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada prinsip-prinsip aplikasi dari bidangnya [5].

**2.2 Permainan (game) Puzzle**

*Puzzle* adalah representasi permainan teka-teki yang dapat diselesaikan dengan mengurutkan atau menyusun komponen-komponen pembentuknya sesuai dengan kondisi yang berurut. Komponen pada *Puzzle* adalah berupa kotak-kotak bernomor atau bergambar yang dapat diacak sedemikian hingga menjadi suatu pola *random* yang dapat dicari jalan penyelesaiannya [3].

**2.3 Heuristik**

Kata heuristik berasal dari sebuah kata kerja bahasa Yunani, *heuriskein*, yang berarti mencari atau memasukkan. Dalam dunia pemrograman, sebagian orang menggunakan kata heuristik sebagai lawan kata dari algoritmik, dimana kata heuristik ini diartikan sebagai suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak menjamin bahwa solusi yang dicari selalu dapat ditemukan [6].

**2.3.1 Fungsi Heuristik**

Di dalam metode-metode yang termasuk *heuristic search*, fungsi heuristik memainkan peranan yang sangat menentukan. Suatu fungsi dapat diterima sebagai fungsi heuristik jika biaya perkiraan yang dihasilkan tidak melebihi dari biaya sebenarnya [7].

**2.3.2 Manhattan Distance**

*Manhattan distance* atau sering disebut *Taxicab Geometry* atau *city block distance*, diperkenalkan oleh Hermann Minkowski pada abad ke-19. *Manhattan distance* merupakan heuristik standar [7].

Pada kasus *puzzle*, fungsi heuristik *Manhattan distance* adalah sebagai berikut :

$$h(n) = (abs(n,x-tujuan,x) + abs(n,y-tujuan,y)).....(1)$$

Dengan,

- h(n) = nilai heuristik untuk simpul n
- n,x = nilai koordinat x dari simpul n
- n,y = nilai koordinat y dari simpul n
- tujuan,x = nilai koordinat x dari simpul tujuan
- tujuan,y = nilai koordinat y dari simpul tujuan

**2.3.3 Plateau**

*Plateau* adalah kondisi ketika ada dua (2) atau lebih *evaluation state* yang mempunyai nilai *heuristic* sama besar dan juga merupakan nilai terbaik. Untuk *plateau* cukup dapat diatasi dengan menerapkan aturan prioritas pergerakan kotak kosong. Misalnya, pergerakan kotak kosong ke atas adalah lebih diprioritaskan daripada pergerakan ke kiri, pergerakan kotak ke kiri lebih diprioritaskan daripada pergerakan ke kanan, dan seterusnya [8].

**2.4 Teknik-Teknik Dasar dalam Pencarian**

Pencarian atau pelacakan merupakan salah satu teknik untuk menyelesaikan permasalahan *Artificial Intelligence* (AI). Keberhasilan suatu sistem salah satunya ditentukan oleh kesuksesan dalam pencarian dan pencocokan.

**2.5 Algoritma Pencarian**

Terdapat banyak metode pencarian yang telah diusulkan. Berbagai metode yang ada dapat dibedakan ke dalam dua jenis, yaitu Pencarian Buta (*blind search*) dan Pencarian Terbimbing (*heuristic search*) [8]. Dalam paper ini digunakan metode pencarian terbimbing.

**2.6 Algoritma A\* (A star)**

Dalam kasus *puzzle* ini, algoritma A\* membandingkan 2 posisi *puzzle* yaitu posisi *puzzle* awal (*state* awal) dengan posisi *puzzle* yang terurut dengan benar (*state* akhir).

Rumus yang digunakan oleh algoritma A\* yaitu [8]:

$$F(n) = g(n) + h(n).....(2)$$

dengan:

- F(n) = total biaya
- g(n) = Jumlah kotak yang berada di posisi salah
- h(n) = Jumlah langkah untuk mencapai posisi benar

**2.7 Algoritma IDA\* (Iterative Deepening A\*)**

Merupakan salah satu metode pencarian yang merupakan pengembangan dari algoritma *Depth-first search*, yang menjadi nilai batasan dari setiap penelusuran, berupa nilai yang paling minimum (gambar 2.1)

**2.8 Algoritma BDA\* (Bi-directional A\*)**

Berbeda dengan A\* dimana pencarian dilakukan pada satu arah, pada *Bidirectional A\** pencarian dilakukan pada dua arah, yaitu dari simpul asal dan simpul tujuan (gambar 2.2)

**3. METODE PENELITIAN**

**3.1 Unified Modeling Language**

*Unified modeling language* (UML) merupakan sebuah bahasa pemodelan standar atau kumpulan teknik-teknik pemodelan yang dipakai untuk menspesifikasi (gambar 2.3)

**3.2 Use Case Diagram**

*Use case diagram* adalah sebuah penggambaran dari sekelompok urutan kegiatan yang yang dikerjakan oleh sistem untuk menghasilkan keluaran yang dapat dilihat oleh pengguna (*actor/user*) [2].

**3.3 Software yang Digunakan**

Perangkat lunak (*software*) yang digunakan dalam pembuatan aplikasi ini adalah sistem operasi Windows 7, Program aplikasi Borland Delphi 7.0 dan Adobe Photoshop 7.

```

1. OPEN = node asal
   CLOSE array kosong
    $g = 0$ 
    $f' = h'$ 

2. Ulangi sampai node tujuan ditemukan
   If OPEN kosong then
     Gagal
   Else
     BestNode = node yang ada di OPEN dengan  $f'$  minimal
     Pindahkan node terbaik tersebut dari OPEN ke CLOSE
     If BestNode = goal then
       Sukses
     Else
       Bangkitkan semua suksesor BestNode tapi jangan buat pointer
       Untuk setiap suksesor kerjakan:
       Hitung  $g(\text{suksesor}) = g(\text{BestNode}) + \text{actual cost}(\text{dari BestNode ke suksesor})$ 
       {Periksa suksesor}
       If suksesor ada di OPEN then {sudah pernah digenerate tapi belum diproses}
         OLD = isi OPEN tersebut
         Tambahkan OLD sebagai suksesor BestNode
         Buat pointer dari OLD ke BestNode
         Bandingkan nilai  $g(\text{OLD})$  dengan  $g(\text{isi OPEN})$ 
         If  $g(\text{OLD})$  lebih baik then
           Ubah parent isi OPEN ke BestNode
           Ubah nilai  $g$  dan  $f'$  pada isi OPEN
         End
       Else
         If suksesor ada di CLOSE then {sudah pernah digenerate dan sudah diproses}
           OLD = isi CLOSE
           Tambahkan OLD sebagai suksesor BestNode
           Bandingkan nilai  $g(\text{OLD})$  dengan  $g(\text{isi CLOSE})$ 
           If  $g(\text{OLD})$  lebih baik then
             Ubah parent isi CLOSE ke BestNode
             Ubah nilai  $g$  dan  $f'$  pada isi CLOSE
             Propagasi untuk semua suksesor OLD dengan penelusuran DFS dengan aturan:
             Ulangi sampai node suksesor tidak ada di OPEN atau node tidak punya suksesor
           If suksesor ada di OPEN then
             Propagasi diteruskan
           Else
             If nilai  $g$  via suksesor lebih baik then
               Propagasi diteruskan
             Else
               Propagasi dihentikan
             End
           End
         End
       Else {suksesor tidak ada di OPEN maupun CLOSE}
         Masukkan suksesor ke OPEN
         Tambahkan suksesor tersebut sebagai suksesor BestNode
         Hitung  $f' = g(\text{suksesor}) + h'(\text{suksesor})$ 
       End
     End
   End

```

Gambar 2.1 pseudocode dari Algoritma A\* [8]

```

function IDA*(problem) returns a solution sequence
  inputs: problem, a problem
  local variables: f-limit, the current f-Cost limit
                  root, a node

  root ← MAKE-NODE(INITIAL-STATE[problem])
  f-limit ← f-Cost(root)
  loop do
    solution, f-limit ← DFS-CONTOUR(root, f-limit)
    if solution is non-null then return solution
    if f-limit is INFINITE then return failure
  end

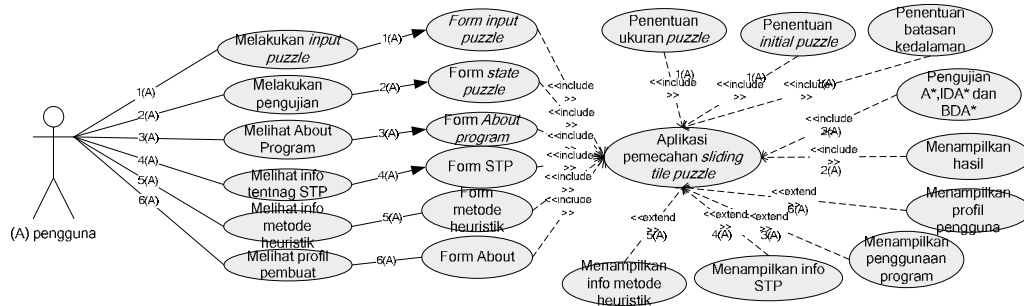
function DFS-CONTOUR(node, f-limit) returns a solution sequence and a new f-Cost limit
  inputs: node, a node
         f-limit, the current f-Cost limit
  local variables: next-f, the f-Cost limit for the next contour, initially INFINITE

  if f-Cost[node] > f-limit then return null, f-Cost[node]
  if GOAL-TEST[problem](STATE[node]) then return node, f-limit
  for each node s in SUCCESSOR(node) do
    solution, new-f ← DFS-CONTOUR(s, f-limit)
    if solution is non-null then return solution, f-limit
    next-f ← MIN(next-f, new-f)
  end
  return null, next-f
    
```

Gambar 2.2 pseudocode dari Algoritma IDA\* [8]

1. Himpunan  $S, T \in \emptyset$ ; nilai  $s$  dan  $d$  tidak boleh  $+\infty$ ; tentukan  $p_s(s) = 0$  dan  $p_d(d) = 0$ .
2. Cari node  $u_s$  yang memiliki nilai  $p_s(v_s) + h_s(v_s)$  terkecil di  $S$  dan tambahkan  $u_s$  ke  $S$ . Jika  $u_s$  di  $D$ , lakukan langkah 7.
3. Untuk semua node  $v_s$  dimana sisi  $(u_s, v_s)$  di dalam  $E$ , jika  $p_s(u_s) + l(u_s, v_s)$  lebih kecil dari  $p_s(v_s)$ : ganti rute  $(s, v_s)$  dengan rute  $(s, u_s) +$  sisi  $(u_s, v_s)$  dan ganti nilai  $p_s(v_s) = p_s(u_s) + l(u_s, v_s)$  dan hapus  $v_s$  dari  $S$  jika  $v_s$  di dalam  $S$ .
4. Cari node  $u_d$  yang memiliki nilai  $p_d(v_d) + h_d(v_d)$  terkecil di  $D$  dan tambahkan  $u_d$  ke  $D$ . Jika  $u_d$  di  $S$ , lakukan langkah 7.
5. Untuk semua node  $v_d$  dimana sisi  $(v_d, u_d)$  di dalam  $E$ , jika  $l(v_d, u_d) + p_d(u_d)$  lebih kecil dari  $p_d(v_d)$ : ganti rute  $(v_d, d)$  dengan sisi  $(v_d, u_d) +$  rute  $(u_d, d)$  dan ganti nilai  $p_d(v_d) = l(v_d, u_d) + p_d(u_d)$  dan hapus  $v_d$  dari  $D$  jika  $v_d$  di dalam  $D$ .
6. Kembali ke langkah 2.
7. Cari sisi  $(u, v)$  dengan meminimalisasi  $p_s(u) + l(u, v) + p_d(v)$  dimana  $u$  di dalam  $S$  dan  $v$  di dalam  $D$ . Jika  $p_s(u) + l(u, v) + p_d(v) < p_s(u_s) + p_d(u_d)$  maka rute terpendek  $(s, d) =$  rute  $(s, u) +$  sisi  $(u, v) +$  rute  $(v, d)$  jika tidak, maka rute terpendek  $(s, d) =$  rute  $(s, u_s) +$  rute  $(u_d, d)$ .

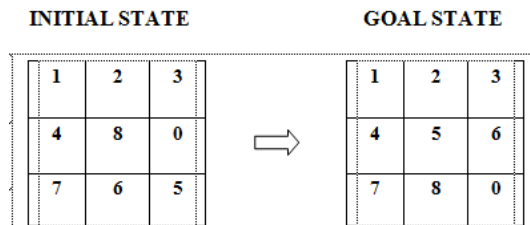
Gambar 2.3 pseudocode dari Algoritma BDA\* [8]



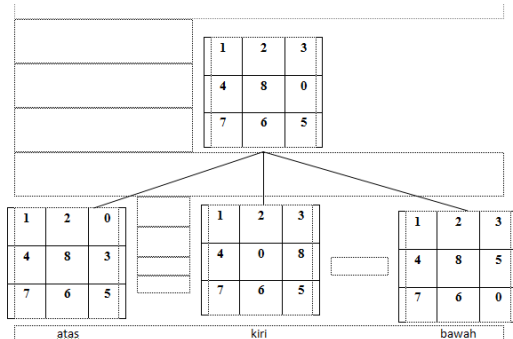
Gambar 3.1 Use case diagram sistem [1]

3.4 Ilustrasi Perhitungan biaya (f(n))

Gambar 3.2 menunjukkan contoh data set dengan ukuran *puzzle* 3x3. Setiap kotak mempresentasikan simpul (*node*). Setiap kotak terhubung ke delapan kotak yang paling dekat, artinya setiap simpul (*node*) terhubung ke simpul lain yang berada di sebelah kanan, kiri, atas-kanan, bawah-kanan, bawah-kiri, dan atas-kiri dari simpul tersebut. Tujuan dari *puzzle* ini adalah melakukan pergerakan dari keadaan awal untuk mencapai konfigurasi nomor ubin tertentu.



Gambar 3.2 Contoh dataset 8 Puzzle ukuran 3 x 3 Misalkan pada langkah pertama terjadi pembukaan node:



Gambar 3.3 Ilustrasi langkah pertama untuk dataset pada contoh 3.2

Pada conoth pembukaan cabang atas (gambar 3.3) angka 1,2,4, dan 7 membutuhkan 0 langkah karena sudah berada pada posisi yang benar. Sedangkan angka 3,5,6, dan 8 berada di posisi yang salah jadi  $g(n) = 4$ . Jumlah langkah yang diperlukan masing-masing kotak yang salah untuk menuju posisi yang benar di *goal state* adalah angka 3 dan 8 membutuhkan 1 langkah sedangkan angka 5 dan 6 membutuhkan 2 langkah. Sehingga  $h(n) =$

$0+0+0+1+1+0+2+2 = 6$ , jadi nilai keseluruhan  $f(n)=g(n)+h(n)$  adalah  $f(n) = 4+6 = 10$  (angka 0 tidak dihitung karena merupakan posisi kotak yang kosong).

4. HASIL DAN PEMBAHASAN

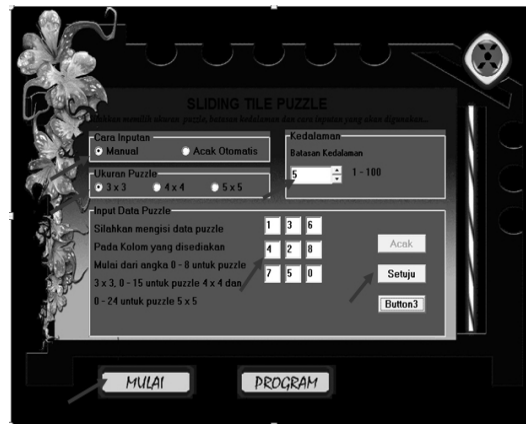
4.1 Hasil

Sesuai dengan tujuan dari penelitian ini maka dapat diketahui hasil unjuk kerja dari metode heuristik (Algoritma A\*, IDA\* dan BDA\*) dalam aplikasi *Sliding Tile Puzzle*, sehingga dapat dibuktikan bahwa ketiga algoritma tersebut dapat digunakan dalam penyelesaian *puzzle* dan dapat melihat perbandingan antara algoritma A\*, algoritma IDA\* dan algoritma BDA\* [1].

4.2 Pembahasan

4.2.1 Aplikasi

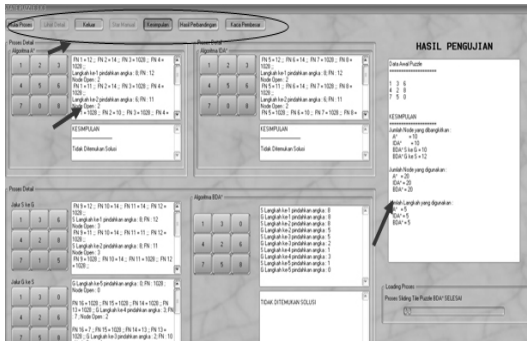
Aplikasi ini merupakan aplikasi pemecahan *puzzle* dimana terdapat *form* menu utama dan *form* untuk *input puzzle*. Pada *form input puzzle* terlebih dahulu kita harus memilih ukuran *puzzle* yang akan diselesaikan dan kemudian dapat memilih batasan kedalaman yang digunakan dalam pemecahan *puzzle* tersebut. Didalam *form* terdapat tiga buah ukuran *puzzle* yang dapat dipilih, yaitu 3x3, 4x4 dan 5x5 [1].



Gambar 3.1 Tampilan *input puzzle* [1]

Setelah *user* mengisi semua dengan benar, maka masing-masing algoritma dapat menampilkan

jalur yang ditemukan, sehingga dari jawaban tersebut dapat dilihat bahwa algoritma mana yang lebih baik.



Gambar 3.2 Hasil pengujian [1]

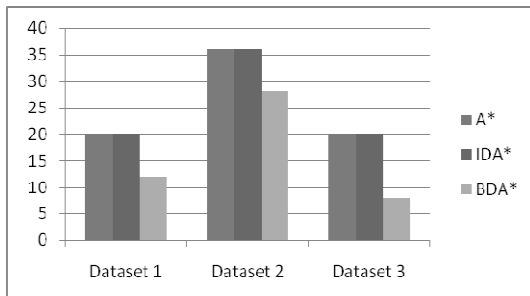
4.3 Pengujian dataset

Pengujian yang akan dilakukan terhadap aplikasi adalah dengan membandingkan ketiga algoritma dengan parameter perbandingan yaitu jumlah *node* yang digunakan, jumlah *node* yang dibangkitkan (*generate node*) dan jumlah langkah yang dibutuhkan untuk mencapai solusi.

Untuk mempermudah melihat perbandingan dari hasil pengujian jumlah *node* yang digunakan, maka dibuat tabel sebagai berikut [1] :

Tabel 1 perbandingan jumlah *node* A\*, IDA\* dan BDA\*

No	Contoh kasus	Algoritma A*	Algoritma IDA*	Algoritma BDA*
1	1 2 3	20 node	20 node	12 node
	4 8 0			
	7 6 5			
2	1 5 2	36 node	36 node	28 node
	7 4 0			
	8 6 3			
3	2 0 3	20 node	20 node	8 node
	1 5 6			
	4 7 8			

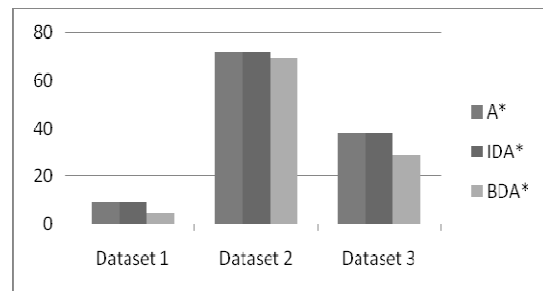


Gambar 3.3 Grafik perbandingan jumlah *node* A\*, IDA\* dan BDA\*

Pada tabel 1 dapat dilihat perbandingan jumlah *node* yang di perlukan oleh Algoritma A\*, Algoritma IDA\* dan Algoritma BDA\*. Dalam penyelesaian kasus, Algoritma BDA\* memiliki jumlah *node* yang sedikit dibandingkan algoritma A\* dan algoritma IDA\*.

Tabel 2 perbandingan jumlah *Generate node* A\*, IDA\* dan BDA\*

No	Contoh kasus	Algoritma A*	Algoritma IDA*	Algoritma BDA*
1	1 2 3	9 node	9 node	5 node
	5 0 6			
	4 7 8			
2	1 2 6	72 node	72 node	69 node
	7 5 8			
	3 0 4			
3	1 2 3	38 node	38 node	29 node
	0 4 5			
	6 7 8			



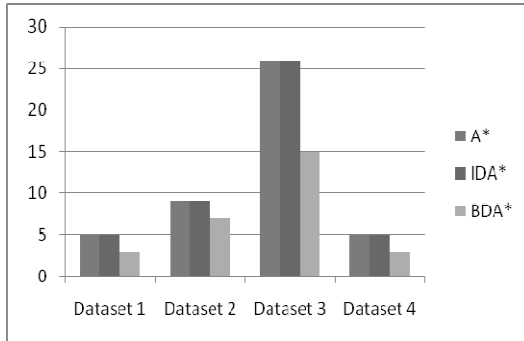
Gambar 3.4 Grafik perbandingan jumlah *generated node* A\*, IDA\* dan BDA\*

Pada tabel 2 dapat dilihat perbandingan *node* yang dibangkitkan oleh Algoritma A\*, Algoritma IDA\* dan Algoritma BDA\*. Dalam penyelesaian kasus, Algoritma BDA\* memiliki jumlah *node* yang dibangkitkan lebih sedikit dibandingkan algoritma A\* dan algoritma IDA\*.

Tabel 3 Perbandingan jumlah langkah A\*, IDA\* dan BDA\*

No	Contoh kasus	Algoritma A*	Algoritma IDA*	Algoritma BDA*
1	1 2 3	5 langkah	5 langkah	3 langkah
	4 8 0			
	7 6 5			
2	1 5 2			

	7	4	0	9	9	7
	8	6	3	langkah	langkah	langkah
3	4	1	3	26	26	15
	2	5	0			
	7	8	6			
4	2	0	3	5	5	3
	1	5	6			
	4	7	8			



Gambar 3.5 Grafik perbandingan jumlah langkah A\*, IDA\* dan BDA\*

Tabel 3 merupakan perbandingan antara Algoritma A\*, Algoritma IDA\* dan Algoritma BDA\* di dalam menyelesaikan *sliding tile puzzle*. Dari tabel 3 dapat dilihat bahwa, Algoritma BDA\* mampu menyelesaikan contoh kasus *puzzle* dengan baik karena memiliki langkah yang lebih pendek dibandingkan algoritma A\* dan algoritma IDA\*.

Berdasarkan dari analisa beberapa faktor di atas, dapat disimpulkan bahwa algoritma BDA\* merupakan algoritma pencarian yang paling efektif dari pada ke-2 algoritma pencarian lainnya. Beberapa alasan yang dapat mendukung pernyataan tersebut adalah sebagai berikut [1] :

- 1) Algoritma BDA\* selalu menemukan solusi dengan waktu yang relatif singkat dari algoritma lainnya. Dan apabila ada solusi lebih dari satu , maka BDA\* akan menemukan solusi yang paling dekat dan membutuhkan waktu yang lebih sedikit.
- 2) Algoritma BDA\* membutuhkan ruang memori yang lebih sedikit dari algoritma yang lain.
- 3) Jumlah *node* yang digunakan dalam pencarian solusi minimal, berbeda dengan ke-2 algoritma pencarian lainnya.
- 4) Apabila solusi terdapat pada level yang dalam, maka algoritma BDA\* akan menemukannya lebih cepat dari algoritma pencarian lainnya.
- 5) Algoritma BDA\* dipastikan dapat memberikan solusi yang diharapkan dan optimal.

## 5. KESIMPULAN

### 5.1 Kesimpulan

Setelah mengetahui hasil perbandingan unjuk kerja metode heuristik (algoritma A\*, IDA\* dan BDA\*) dapat ditarik kesimpulan sebagai berikut :

- a. Dataset yang diuji adalah 200 dataset, hasilnya 160 dataset dapat memecahkan kasus dan memberikan hasil yang *complete* (selalu menemukan solusi jika solusinya ada), sehingga dapat menemukan solusi untuk mencapai *goal* yang diinginkan, sedangkan 40 dataset lainnya tidak dapat diselesaikan karena tidak dapat menemukan solusi untuk mencapai *goal*. Dan tingkat keberhasilan dari pemecahan *sliding tile puzzle* dengan menggunakan metode heuristik (Algoritma A\*, IDA\* dan BDA\* ) adalah 80 % dapat menemukan solusi dan 20 % tidak dapat menemukan solusi dari 200 kali pengujian.
- b. Algoritma BDA\* adalah algoritma terbaik dalam mendapatkan jalur yang lebih pendek yaitu 50 % dari 80 % tingkat keberhasilan dibandingkan dengan Algoritma A\* dan Algoritma IDA\*.
- c. Untuk perhitungan *puzzle* pada ukuran 4x4 dan 5x5, waktu pencarian yang dibutuhkan sedikit lebih lama dari ukuran 3x3 dan belum tentu ditemukan solusi. Ini disebabkan karena angka yang dibutuhkan lebih banyak. Pada beberapa kasus yang telah di uji hingga batasan kedalaman 50 langkah belum ada yang menemukan solusi, tetapi algoritma BDA\* adalah algoritma yang lebih mendekati solusi.

### 5.2 Saran

Beberapa saran yang dapat diberikan untuk membantu dalam pengembangan aplikasi ini lebih lanjut yaitu dapat dibuat jenis *puzzlenya* dalam bentuk gambar dan huruf agar tampilannya lebih menarik dan diharapkan dapat menggunakan metode-metode heuristik lainnya dalam penyelesaian pencarian jalur pada kasus *Sliding Tile Puzzle* dan dapat melakukan pengujian lebih dari 200 kasus yang ada untuk mendapatkan hasil yang lebih optimal.

## REFERENSI

- [1] Asunta, Maria. 2014, Implementasi Pemecahan *Sliding tile puzzle* Menggunakan Metode Heuristik (Algoritma A\*, IDA\* dan BDA\*). Skripsi Ilmu Komputer. Kupang: Universitas Nusa Cendana.

## PROSIDING SEMMAU 2016

---

- [2] Booch, G., Rumbaugh, J. and Jacobson, I. 1999, *The Unified Modeling Language User Guide*. Addison-Wesley, Canada.
- [3] Jokodo, 2011, *Game puzzle number*. Jakarta : Universitas Gunadharma [4] Kusumadewi S., 2003, *Artificial Intelligence : Teknik dan Aplikasinya*, Yogyakarta: Graha Ilmu.
- [4] Kusumadewi S., 2003, *Artificial, Intelligence, Teknik dan Aplikasinya*. Yogyakarta: Graha Ilmu.
- [5] Kusumadewi, Sri, 2007, *Pengantar Kecerdasan Buatan, Teknik Pencarian Heuristik*. Yogyakarta : Universitas Islam Indonesia.
- [6] Suyanto, 2007, *Artificial, Intelligence, Searching, Reasoning, Planning and Learning*, Informatika.
- [7] Suyanto, 2011, *Artificial Intelligence*, Bandung: Penerbit Informatika.
- [8] Yuliana, Ananda, dan Ibnu Surya, 2012, *Implementasi Algoritma A Star dan Greedy Best First Search pada pemecahan puzzle* 8. Teknik Informatika.



# STIKOM UYELINDO KUPANG

Jalan Perintis Kemerdekaan I -Kayu Putih Kupang-NTT  
Telp; 0380-8554500, 85554499, Fax.0380-8554502

Website: <http://www.uyelindo.ac.id>

Website: <http://www.semmau.uyelindo.ac.id>

Email: [stikom@uyelindo.ac.id](mailto:stikom@uyelindo.ac.id), [semmau@uyelindo.ac.id](mailto:semmau@uyelindo.ac.id)

PROGRAM STUDI :

SISTEM INFORMASI (S1) TERAKREDITASI

TEKNIK INFORMATIKA (S1) TERAKREDITASI

TEKNIK INFORMATIKA (D3) TERAKREDITASI

ISBN



978-602-73628-0-3