

**IMPLEMENTASI ALGORITMA A* DAN FLOYD WARSHALL
DENGAN PRUNING UNTUK PENYELESAIAN PERMAINAN
SOKOBAN**

Skripsi



oleh
JOSEF SEPTA SETIAWAN
22094640

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2014

**IMPLEMENTASI ALGORITMA A* DAN FLOYD WARSHALL
DENGAN PRUNING UNTUK PENYELESAIAN PERMAINAN
SOKOBAN**

Skripsi



Diajukan kepada Program Studi Teknik Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar
Sarjana Komputer

Disusun oleh

JOSEF SEPTA SETIAWAN
22094640

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2014

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

IMPLEMENTASI ALGORITMA A* DAN FLOYD WARSHALL DENGAN PRUNING UNTUK PENYELESAIAN PERMAINAN SOKOBAN

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi keserjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar keserjanaan saya.

Yogyakarta, 23 Mei 2014



JOSEF SEPTA SETIAWAN
22094640

HALAMAN PERSETUJUAN

Judul Skripsi : IMPLEMENTASI ALGORITMA A* DAN FLOYD
WARSHALL DENGAN PRUNING UNTUK
PENYELESAIAN PERMAINAN SOKOBAN

Nama Mahasiswa : JOSEF SEPTA SETIAWAN

N I M : 22094640

Matakuliah : Skripsi (Tugas Akhir)

Kode : TIW276

Semester : Genap

Tahun Akademik : 2013/2014

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 23 Mei 2014

Dosen Pembimbing I

Rosa Delina, S.Kom., M.Kom.

Dosen Pembimbing II

Antonius Rachmat C., SKom., M.Cs

HALAMAN PENGESAHAN

IMPLEMENTASI ALGORITMA A* DAN FLOYD WARSHALL DENGAN PRUNING UNTUK PENYELESAIAN PERMAINAN SOKOBAN

Oleh: JOSEF SEPTA SETIAWAN / 22094640

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Teknik Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 9 Juni 2014

Yogyakarta, 12 Juni 2014
Mengesahkan,

Dewan Penguji:

1. Rosa Delima, S.Kom., M.Kom.
2. Antonius Rachmat C., SKom, M.Cs
3. Gloria Virginia, S.Kom., MAI, Ph.D.
4. Ignatia Dhan E K R, S.Kom



Dekan

(Drs. Wimmie Hardiwidjojo, M.IT.)

Ketua Program Studi

(Nugroho Agus Haryono, M.Si)

INTISARI

Sokoban adalah permainan yang mudah dimainkan dan memiliki aturan main yang mudah dipahami. Namun Sokoban dapat memiliki tingkat kerumitan yang tinggi sehingga membuat orang-orang merasa tertantang untuk menyelesaikannya. Dengan algoritma A^* dan *Floyd Warshall* dengan *Prunning* penulis mencoba untuk membuat sistem cerdas yang mampu menyelesaikan permainan Sokoban. A^* digunakan untuk proses *path finding*, sedangkan *Floyd Warshall* yang akan menentukan solusi terbaik. Lalu *Prunning* bekerja membantu kinerja *Floyd Warshall* agar dapat melewati proses-proses yang tidak diperlukan.

Penelitian dimulai dengan analisa aturan main Sokoban, pergerakan objek yang terdapat pada Sokoban, dan permasalahan yang biasa muncul pada permainan ini. Dilanjutkan dengan pengimplementasian A^* dan *Floyd Warshall* dengan *Prunning*. Terakhir dilakukan pengujian pada agen untuk menyelesaikan kasus-kasus dari permainan sokoban.

Hasil analisis data pengujian menunjukkan kesimpulan dari tugas akhir ini, yaitu : (1) Algoritma dianggap sukses diterapkan dan berhasil membuat sistem yang dapat menyelesaikan permainan Sokoban, namun ditemukan juga karakteristik dari beberapa contoh kasus yang tidak dapat diselesaikan oleh sistem (2) Sistem mampu memberikan solusi dengan nilai *cost* yang lebih efektif apabila dibandingkan dengan *user* awam rata-rata. Hasil nilai *cost* dari sistem tidak lebih efektif apabila dibandingkan dengan *user* yang telah terbiasa memainkan permainan Sokoban.

Kata Kunci : A^* , *Dynamic Programming Floyd Warshall*, *Prunning*, *Sokoban*.

DAFTAR ISI

HALAMAN JUDUL.....	
PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
UCAPAN TERIMA KASIH.....	vi
INTISARI.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Metodologi Penelitian.....	3
BAB II LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.1.1 Algoritma Pencarian dan Optimalisasi.....	5
2.2 Landasan Teori.....	6
2.2.1 Sokoban.....	6
2.2.1.1 Aturan Main Sokoban.....	6
2.2.1.2 Kompleksitas Sokoban.....	7

2.2.1.3 Deadlocks	8
2.2.2 A*	9
2.2.3 Floyd Warshall.....	10
2.2.4 Pruning.....	12
BAB III ANALISIS DAN PERANCANGAN SISTEM	13
3.1 Kebutuhan Sistem.....	13
3.2 Rancangan Sistem	13
3.2.1 Floyd Warshall.....	13
3.2.2 Pruning.....	17
3.2.3 A*	21
3.3 Aturan Permainan.....	23
3.4 Struktur Data	25
3.5 Perancangan Basis Data	27
3.6 Perancangan Antar Muka	28
3.7 Perancangan Pengujian Sistem.....	29
BAB IV IMPLEMENTASI DAN ANALISIS SISTEM	31
4.1 Implementasi Sistem	31
4.1.1 Implementasi Rancangan Tampilan	31
4.1.1.1 Tampilan Menu Utama.....	31
4.1.1.2 Tampilan Permainan.....	34
4.1.1.3 Tampilan Sistem melakukan Solving.....	38
4.1.1.4 Tampilan Pembuatan Map.....	40
4.1.1.5 Tampilan Skor Pemain	43
4.1.1.6 Tampilan Hint/Bantuan	44
4.1.2 Implementasi Rancangan Sistem dan Algoritma.....	45
4.1.2.1 Implementasi Algoritma A*	45
4.1.1.1 Implementasi Algoritma Floyd Warshall dengan Pruning	50
4.2 Analisis Sistem	54

4.2.1 Hasil Pengujian.....	54
4.2.1.1 Kemampuan Solving Sistem.....	55
4.2.1.1 Kemampuan Sistem Mendapatkan Rute Efektif.....	64
BAB V KESIMPULAN DAN SARAN.....	73
5.1 Kesimpulan.....	73
5.2 Saran	74
DAFTAR PUSTAKA	75
LAMPIRAN.....	

©UKDW

DAFTAR TABEL

Tabel 4.1 Kemampuan Sistem Dalam Menyelesaikan Permainan Sokoban	55
Tabel 4.2 Hasil Cost yang diberikan User Kategori-1	65
Tabel 4.3 Hasil Cost yang didapat oleh Sistem dibandingkan dengan MIN dari User Kategori-1	66
Tabel 4.4 Hasil Cost yang didapat oleh Sistem dibandingkan dengan Average dari User Kategori-1	68
Tabel 4.5 Hasil Cost yang diberikan User Kategori-2	69
Tabel 4.6 Hasil Cost yang didapat oleh Sistem dibandingkan dengan MIN dari User Kategori-2	70
Tabel 4.7 Hasil Cost yang didapat oleh Sistem dibandingkan dengan Average dari User Kategori-2	71

©UKDW

DAFTAR GAMBAR

Gambar 2.1 Kasus Sokoban Sederhana	7
Gambar 2.2 Kasus Sokoban Rumit	8
Gambar 2.3 Area Instant Deadlock bertanda ‘X’	8
Gambar 2.4 Contoh kasus: pencarian path optimal S ke G.....	9
Gambar 2.5 Contoh Kasus: Mencari jarak terpendek tiap edge.....	11
Gambar 2.6 Matriks Adjacency	11
Gambar 2.7 Contoh pemotongan Tree Prunning	12
Gambar 3.1 Flowchart Floyd Warshall.....	14
Gambar 3.2 Ilustrasi Penerapan Algoritma Floyd Warshall	16
Gambar 3.3 Flowchart Prunning.....	18
Gambar 3.4 Ilustrasi Penerapan Prunning.....	19
Gambar 3.5 Pemotongan Cabang Solusi oleh Prunning	21
Gambar 3.6 Flowchart A*	22
Gambar 3.7 Dua kotak berhimpit.....	23
Gambar 3.8 Contoh Map Sokoban.....	24
Gambar 3.9 Kondisi Game Selesai	25
Gambar 3.10 Struktur Map dalam Array	25
Gambar 3.11 Struktur letak objek dalam Array	26
Gambar 3.12 Penyalinan Array Level	27
Gambar 3.13 Model Basis Data	27
Gambar 3.14 Rancangan Interface Game Sokoban	29
Gambar 4.1 Tampilan Menu Utama.....	32
Gambar 4.2 Memilih Level dari Selection Input	33
Gambar 4.3 Menampilkan Permainan dari Level yang dipilih	34
Gambar 4.4 Player Menggerakkan Obyek Pendorong	35
Gambar 4.5 Warna Box berubah saat menempati TitikTujuan	36
Gambar 4.6 Sistem Memberi Alert Permainan Terselesaikan	37
Gambar 4.7 Prompt Input Nama Player	37

Gambar 4.8 Tombol Solve dapat dioperasikan	38
Gambar 4.9 Informasi Detail Proses Solving	39
Gambar 4.10 Sistem tidak menemukan solusi	40
Gambar 4.11 Membuat Map Permainan	41
Gambar 4.12 Lokasi yang ditandai	41
Gambar 4.13 Icon Pusher hilang	42
Gambar 4.14 Map yang dibuat disimpan pada level 0.....	43
Gambar 4.15 Sistem menampilkan Best Score	44
Gambar 4.16 Tombol “Need Help?” untuk menunjukkan Hint	44
Gambar 4.17 Sistem Memunculkan Hint	45
Gambar 4.18 Menemukan Rute untuk Box	46
Gambar 4.19 Rute Box ditemukan	47
Gambar 4.20 Rute Pendorong	47
Gambar 4.21 Pengujian Map Buatan User	48
Gambar 4.22 Pseudo Code A*	49
Gambar 4.23 Contoh Kondisi 1	50
Gambar 4.24 Solve Box B	51
Gambar 4.25 Solve Box B setelah Solve Box A	51
Gambar 4.26 Ilustrasi Percabangan Solusi Floyd Warshall, kasus dimana terdapat 3 box	52
Gambar 4.27 Proses Pengecekan Tiap Cabang Solusi	52
Gambar 4.28 Pruning Menghentikan Percabangan Solusi	53
Gambar 4.29 Pseudo Code Floyd Warshall dan Pruning	54
Gambar 4.30 Persentase Keberhasilan Sistem menyelesaikan Kasus 1	59
Gambar 4.31 Persentase Keberhasilan Sistem menyelesaikan Kasus 2	60
Gambar 4.32 Persentase Keberhasilan Sistem menyelesaikan Kasus 2 untuk map yang memiliki solusi penyelesaian	61
Gambar 4.33 Map 10 Aplikasi Permainan Sokoban	62
Gambar 4.34 Map 10 Aplikasi Permainan Sokoban	63

Gambar 4.35 Diagram Perbandingan Jumlah Hasil Nilai Terkecil yang didapat Sistem dengan nilai Min dari User Kategori-1	66
Gambar 4.36 Diagram Perbandingan Jumlah Hasil Nilai Terkecil untuk setiap Map yang berhasil diselesaikan Sistem dengan nilai Min dari User Kategori-1	67
Gambar 4.37 Diagram Perbandingan Jumlah Hasil Nilai Terkecil yang didapat Sistem dengan nilai Rata-rata dari User Kategori-1	68
Gambar 4.38 Diagram Perbandingan Jumlah Hasil Nilai Terkecil untuk setiap Map yang berhasil diselesaikan Sistem dengan nilai Rata-rata dari User Kategori-1	69
Gambar 4.39 Diagram Perbandingan Jumlah Hasil Nilai Terkecil untuk setiap Map yang berhasil diselesaikan Sistem dengan nilai MIN dari User Kategori-2	71
Gambar 4.40 Diagram Perbandingan Jumlah Hasil Nilai Terkecil untuk setiap Map yang berhasil diselesaikan Sistem dengan nilai Rata-rata dari User Kategori-2.....	72

© UKDW

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dewasa ini telah mencapai tahap dimana hampir setiap perangkat elektronik telah dilengkapi dengan sistem otomatisasi dengan tujuan mempermudah manusia dalam melakukan pekerjaannya. Pada dasarnya sistem otomatisasi adalah sistem cerdas yang didalamnya telah ditanamkan kemampuan penalaran yang dapat menyelesaikan suatu permasalahan tertentu. Sistem cerdas sering digunakan untuk aplikasi permainan. Sistem kecerdasan pada *game* dibangun dengan tujuan untuk membantu *player* dalam memainkan *game* baik sebagai lawan main maupun sebagai panduan atau tuntunan dalam menyelesaikan misi dari suatu *game*.

Sebuah permainan yang sederhana namun membutuhkan kemampuan penalaran yang cukup rumit dapat ditemui pada permainan Sokoban. Sokoban adalah *game* yang berasal dari Jepang. Pada Sokoban ditampilkan sebuah tampilan gudang dimana gudang tersebut terdapat beberapa obyek menyerupai *box* dan sebuah subyek yang biasanya berbentuk manusia. *Box* tersebut berada pada posisi yang teracak dan subyek yang berupa manusia itu bertugas untuk mendorong *box* yang ada ke tempat tujuan dari masing-masing *box*. Permainan Sokoban memiliki beberapa *level* dengan tingkat kesulitan/kerumitan dan penyelesaian yang berbeda-beda. Selain itu dalam menyelesaikan permainan Sokoban ini dapat dihasilkan alur langkah-langkah yang tidak hanya satu walaupun tujuannya tetap sama.

Beberapa orang yang bergerak dalam bidang *software development* telah mencoba menggunakan beberapa metode pencarian dan optimalisasi untuk dapat membangun sebuah sistem cerdas yang dapat mencari solusi dengan langkah paling efisien dalam menyelesaikan tantangan yang tersedia dalam *game* Sokoban.

Namun ternyata para peneliti tidak dapat menegaskan algoritma yang manakah yang memang paling baik untuk setiap persoalan *game* Sokoban ini. Metode pencarian dan optimalisasi cukup banyak macamnya. Oleh karena itu penulis pun tertarik untuk mengimplementasikan metode *A** dan *Floyd Warshall* dengan *Pruning* untuk membangun sistem cerdas yang dapat menggantikan *player* untuk menyelesaikan permainan Sokoban ini. Dengan harapan agar metode yang dipilih oleh penulis dapat menjadi salah satu metode yang layak dipertimbangkan untuk membangun sistem cerdas dalam penyelesaian tantangan pada *game* Sokoban.

1.2 Rumusan Masalah

Berdasarkan uraian pada Sub Bab 1 maka akan dilakukan pembangunan Game Sokoban yang dilengkapi dengan sistem cerdas sebagai *solver*. Dimana *solver* akan dibangun menggunakan metode *A** sebagai agen pencari langkah-langkah terpendek dan *Floyd Warshall* dengan *Pruning* untuk mendapatkan keputusan jalur yang terpendek, sistem yang akan dibangun akan memiliki perumusan masalah adalah sebagai berikut:

- Apakah algoritma *A** dan *Floyd Warshall* dengan *Pruning* dapat diimplementasikan untuk menyelesaikan tantangan pada permainan Sokoban?

1.3 Batasan Masalah

Ruang lingkup permasalahan dalam penelitian ini dibatasi oleh hal-hal sebagai berikut :

1. Game Sokoban dirancang dengan navigasi dua dimensi.
2. Ukuran *map* maksimal yang tersedia adalah 10x10 *grid*.

3. Jumlah kotak maksimal yang akan diberikan dalam tantangan adalah 5 kotak.
4. *Map* yang ada pada Game Sokoban sudah tersedia secara manual.
5. Tingkatan kesulitan pada *game* dirancang sedemikian rupa bukan untuk menguji kemampuan *player*(manusia) namun untuk menguji kemampuan sistem cerdas.

1.4 Tujuan Penelitian

Membangun sebuah *game* Sokoban yang dapat dimainkan dan dengan dilengkapi sistem cerdas yang dapat menggantikan *player* untuk menyelesaikan *game* Sokoban ini sendiri.

Mengukur kemampuan metode A^* dan *Floyd Warshall* dengan *Pruning* dilihat dari hasil *cost* atau jumlah langkah yang efektif(seminimal mungkin) dalam menyelesaikan tantangan pada *game* Sokoban.

1.5 Metodologi Penelitian

Penelitian dilakukan oleh penulis secara bertahap dan sesuai dengan metodologi yang ada.

Metodologi yang digunakan dalam penelitian ini adalah :

1. Studi Pustaka

Studi Pustaka dilakukan dengan mempelajari teori-teori melalui buku, artikel, jurnal dan bahan lain yang mendukung yang berhubungan dengan Algoritma-algoritma *Heuristic Search* dan *Optimalization, rule play Sokoban*, dan metode lainnya yang dibutuhkan.

2. Perancangan sistem

Tahap ini berisi perancangan basis data dan perancangan antarmuka untuk sistem yang akan dibangun.

3. Pembangunan sistem

Tahap ini merupakan tahap pembuatan program sesuai dengan rancangan yang telah dibuat. Pembangunan sistem menggunakan bahasa HTML5 dan Javascript.

4. Pengujian sistem

Pada tahap ini dilakukan pengujian terhadap program. Dengan memberikan pembandingan sebagai tolok ukur. Hal-hal yang dilakukan sebagai bentuk dari pengujian antara lain:

- Sistem akan diuji untuk menyelesaikan tantangan Sokoban dalam bentuk map yang telah tersedia pada aplikasi yang nantinya dibuat, penyelesaian yang diberikan oleh program akan dibandingkan dengan hasil penyelesaian dari beberapa subyek (manusia) yang telah ditunjuk dan bersedia sebagai bahan penelitian.
- Memberikan initial state yang berbeda dengan mengubah-ubah posisi dari pendorong dan kotak dari kondisi awal. Apakah sistem dapat melakukan proses penyelesaian atau tidak, dan apabila sistem tidak dapat menemukan solusi maka berhasilkah sistem tetap memberikan laporannya.

5. Analisis hasil percobaan dan evaluasi

Tahap ini digunakan untuk menganalisa dan mengevaluasi hasil dengan rancangan semula. Hasil dari pengujian sistem akan dibandingkan dengan harapan dan hipotesa yang diuraikan pada bagian Bab 3 dari karya tulis ini. Sehingga kekurangan dan kelebihan dari program ini dapat dijabarkan dengan jelas dan dapat ditarik kesimpulan sebagai hasil penelitian ini.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil implementasi dan analisis algoritma A* dan Floyd Warshall dengan Prunning dalam Permainan Sokoban, dapat diambil beberapa kesimpulan berikut.

- a. Algoritma dianggap sukses diterapkan dan berhasil membuat sistem yang dapat menyelesaikan permainan Sokoban mencapai 83,33% dari total jumlah *map* yang tersedia pada aplikasi. Adapun beberapa karakteristik dari jenis *map* yang tidak mampu diselesaikan oleh sistem yaitu:
 1. Sistem tidak dapat melakukan *solving* jika kondisi terdapat dua atau lebih *box* yang saling membutuhkan pergeseran pada *box* lain untuk membuka jalan. Seperti yang dijumpai pada kasus *map* 10.
 2. Sistem tidak dapat melakukan *solving* jika kondisi terdapat titik tujuan yang posisinya menutupi titik tujuan yang lainnya. Seperti yang dijumpai pada kasus *map* 11.

Dua karakteristik tersebut memiliki pokok masalah yang sama, yaitu dikarenakan proses sistem yang tidak diperbolehkan mendorong *box* yang bukan gilirannya untuk diselesaikan. Dengan pertimbangan bahwa mendorong *box* lain tidak menjamin akan memberikan kondisi yang lebih baik.

- b. Sistem mampu memberikan solusi dengan nilai *cost* yang lebih efektif apabila dibandingkan dengan *user* awam rata-rata, karena dapat menemukan nilai *cost* lebih pendek hingga 80% dari total jumlah *map* yang berhasil diselesaikan. Hasil nilai *cost* dari sistem tidak terlihat lebih efektif apabila dibandingkan dengan hasil dari rata-rata *user* yang telah terbiasa memainkan permainan Sokoban, karena hanya dapat menemukan nilai *cost* lebih pendek sebanyak 50% dari total jumlah *map* yang dapat diselesaikan.

5.2 Saran

Karena sistem yang telah dibuat masih jauh dari sempurna, maka berikut adalah saran untuk pengembangan sistem:

- a. Sebaiknya sistem yang dibuat didasarkan dengan prioritas agar semua *box* dapat didorong sampai tujuan tanpa membatasi pergerakan mendorong *box* lain. Dalam proses pencarian solusi menggunakan sistem iterasi agar kemungkinan solusi yang didapatkan bisa lebih banyak. Dengan maksud agar kedepannya sistem mampu menyelesaikan kasus yang tidak bisa diselesaikan oleh sistem saat ini.
- b. Tampilan sangat sederhana, akan lebih baik jika pada tampilan diberi tambahan animasi dan transisi. Dan bisa juga ditambahkan sound untuk menambah daya tarik user.

©UKYDIN

DAFTAR PUSTAKA

- Addison & Weasley.(2003). *Introduction to the Design & Analysis of Algorithms*.
- Baharsyah, M. P., Wicaksono, S.U., & Pamuji, T. (2009). *Eksplorasi Algoritma Brute Force, Greedy, dan Dynamic Programming untuk Persoalan Integer Knapsack*. Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung
- Culberson, J.(1999). *Sokoban is pspace-complete*, Informatics 4, fun with Algorithms, pp. 65–76.
- Dor, D., and Zwick, U.(1999) Sokoban and other motion planning problems., Computational Geometry, vol. 13, no. 4, pp. 215–228.
- Hoff, T.F. (2001). *Graph Theory: Finding all-pairs Shortest Paths and Counting Walks*. California: Academic Press - An imprint of Elsevier Science.
- Horowitz, Ellis, & Sartaj Sahni.(1978). *Fundamental of Computer Algorithm*. Pitman Publishing Limited,.
- Jarusek & Pelanek.(2010). *Human Problem Solving: Sokoban Case Study*. Masaryk University Brno: Faculty of Informatic.
- Junghanns & Schaeffer.(2001.) *Sokoban: Enhancing General Single-Agent Search Methods Using Domain Knowledge*, Artificial Intelligence, vol. 129, no. 1-2, pp. 219–251.
- Munir, Rinaldi.(2005) Diktat Kuliah IF2251 : Strategi Algoritmik.
- Schaul, T.(2005). *Master thesis: Evolving a compact,concept-based Sokoban solver*.