

Pemecah Sudoku Interaktif Berbasis Python

Proposal Tugas Akhir

Kelas TA 1

Muhammad Zakaria Musa

NIM: 1103130047



Program Studi Sarjana Teknik Informatika

Fakultas Informatika

Universitas Telkom

Bandung

2017

Lembar Persetujuan

Pemecah Sudoku Interaktif Berbasis Python

A Python-Based Interactive Sudoku Solver.

Muhammad Zakaria Musa

NIM: 1103130047

Proposal ini diajukan sebagai usulan pembuatan tugas akhir pada
Program Studi Sarjana Teknik Informatika
Fakultas Informatika Universitas Telkom

Bandung, 28 Agustus 2017
Menyetujui

Calon Pembimbing 1

Calon Pembimbing 2

Yanti Rusmawati, Ph.D.
NIP: 15711785-1

Muhammad Arzaki, M.Kom.
NIP: 15871701-2

Abstrak

Abstrak

Kata Kunci: metode formal, SAT solver, Sudoku, pengujian kotak hitam . .
. .

Daftar Isi

Abstrak	i
Daftar Isi	ii
I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Rencana Kegiatan	2
1.6 Jadwal Kegiatan	3
II Kajian Pustaka	4
2.1 Metode Formal	4
2.2 Pengujian Kotak Hitam	4
2.3 Logika Proposisi	4
2.4 SAT Solver	5
2.5 Python	5
2.6 <i>Model Checker</i> NuSMV	5
III Metodologi	6
3.1 Analisa spesifikasi dan Kebutuhan sistem	6
3.2 Translasi spesifikasi sistem	6
3.3 Pembuatan aplikasi	6
3.4 Pengujian aplikasi	7
3.5 Analisis sistem	7
Daftar Pustaka	8
Lampiran	9

Bab I

Pendahuluan

1.1 Latar Belakang

Sudoku berasal dari kata *Sūji wa dokushin ni kagiru* yang berarti digitnya harus tunggal adalah suatu *puzzle* (teka-teki) yang direpresentasikan oleh sebuah matriks (*array* dua dimensi) berukuran $n^2 \times n^2$ yang dibangun dari n^2 dengan submatriks (atau blok) yang berukuran $n \times n$. Sudoku merupakan *puzzle* yang biasanya dimainkan oleh satu orang. Pada awal permainan, terdapat beberapa sel yang telah terisi yang disebut dengan pemberian (*givens*). Untuk menyelesaikan permainan ini, seorang pemain harus mengisi setiap sel yang belum terisi dengan angka di antara 1 sampai n^2 sedemikian sehingga setiap baris, setiap kolom, dan setiap blok (submatriks berukuran $n \times n$) memuat tepat satu bilangan di antara 1 sampai n^2 . Biasanya suatu sudoku didesain agar tepat memiliki satu kemungkinan solusi. Hal ini juga mengakibatkan sudoku dapat diselesaikan hanya dengan mengandalkan penalaran yang sederhana. Pengisian suatu sel dapat dilakukan dengan meninjau kemungkinan dari isi sebuah sel.

Banyak pemain yang terjebak pada teka-teki sudoku dan tidak dapat melanjutkan permainan. Hal ini terjadi karena pemain mengisi nilai yang salah atau sudoku tidak memiliki solusi. Oleh karena itu banyak pemain yang membutuhkan pemecah sudoku yang dapat memberikan keterpenuhan dari sebuah sudoku. Sehingga pemain dapat mengetahui sebuah sudoku memiliki solusi atau tidak.

Hingga saat ini, sudah banyak penelitian yang membahas penyelesaian sudoku secara matematis maupun komputasional. Salah satu metode yang cukup dikenal adalah penyelesaian sudoku dengan memanfaatkan masalah keterpenuhan formula proposisional (*propositional satisfiability problem*). Dengan pendekatan ini, syarat-syarat yang harus dipenuhi oleh suatu sudoku dimodelkan dengan satu atau lebih formula logika proposisi. Keterpenuhan (*satisfiability*) dari himpunan formula yang memodelkan syarat-syarat ini akan menjamin bahwa suatu sudoku memiliki suatu solusi.

Pada tugas akhir ini, penulis akan membuat pemecah sudoku interaktif berbasis python. Bahasa python dipilih karena implementasi SAT solver dalam

bahasa python masih masih sedikit dibandingkan pada bahasa C dan C++. Aplikasi akan menggunakan pengujian kotak hitam untuk menilai kualitas aplikasi.

1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah pada tugas akhir ini adalah bagaimana cara membuat SAT solver berbasis python untuk pemecah sudoku yang interaktif.

1.3 Batasan Masalah

Batasan pada tugas akhir ini terbatas pada:

1. Penulis hanya membangun aplikasi sudoku solver berukuran (4×4) , (9×9) , (16×16) .
2. Program mengeluarkan keterpenuhan dari sudoku.
3. Program mengeluarkan solusi dari sudoku jika pengguna menginginkan.
4. Aplikasi akan diuji dengan metode pengujian kotak hitam (*black box testing*).

1.4 Tujuan

Tujuan yang ingin dicapai pada tugas akhir ini adalah untuk membuat pemecah sudoku yang interaktif.

1.5 Rencana Kegiatan

Pada pengerjaan tugas akhir ini beberapa hal yang akan saya lakukan adalah sebagai berikut:

1. Analisa spesifikasi dan Kebutuhan sistem.
2. Translasi spesifikasi sistem.
3. Pembuatan aplikasi.
4. Pengujian aplikasi.
5. Analisis sistem.
6. Penulisan laporan.

1.6 Jadwal Kegiatan

Jadwal pengerjaan tugas akhir sesuai dengan alur yang telah dibuat.

Tabel 1.1: Jadwal Kegiatan.

No	Jenis Kegiatan	Bulan						
		November 2017	Desember 2017	Januari 2018	Februari 2018	Maret 2018	April 2018	Mei 2018
1	Analisa spesifikasi dan kebutuhan sistem							
2	Translasi spesifikasi sistem							
3	Pembuatan aplikasi							
4	Pengujian aplikasi							
5	Analisis sistem							
6	Penulisan Laporan							

Bab II

Kajian Pustaka

Pada bab ini penulis akan menjelaskan teori yang digunakan selama pengerjaan tugas akhir.

2.1 Metode Formal

Metode formal adalah teknik yang digunakan untuk memodelkan suatu sistem yang kompleks [1]. Metode formal dapat digunakan untuk mengembangkan sebuah aplikasi baik itu perangkat lunak atau perangkat keras. Pada masa spesifikasi, metode formal dapat digunakan untuk memberikan gambaran dari sistem yang akan dikembangkan sedetail yang diinginkan. Metode formal dapat digunakan untuk memandu kegiatan pengembangan selanjutnya dan dapat digunakan untuk memverifikasi bahwa pra-syarat untuk sistem yang dikembangkan telah lengkap dan dapat dilanjutkan ke tingkat selanjutnya.

2.2 Pengujian Kotak Hitam

2.3 Logika Proposisi

Logika proposisi adalah bahasa formal untuk memodelkan situasi yang kita sehingga kita dapat memberi alasan tentang hal itu secara formal. Logika proposisi terdiri dari sebuah nilai kebenaran dari proposisinya.

Operator yang digunakan pada logika proposisi adalah sebagai berikut

- Konjungsi(dan) pada operator ini kedua proposisi yang dihubungkan harus bernilai benar agar formula bernilai benar. Operator ini dilambangkan dengan \wedge .
- Disjungsi(atau)
- Negasi(tidak)
- Implikasi(jika-maka)
- Biimplikasi(jika dan hanya jika)

2.4 SAT Solver

2.5 Python

2.6 *Model Checker* NuSMV

NuSMV adalah *model checker* yang

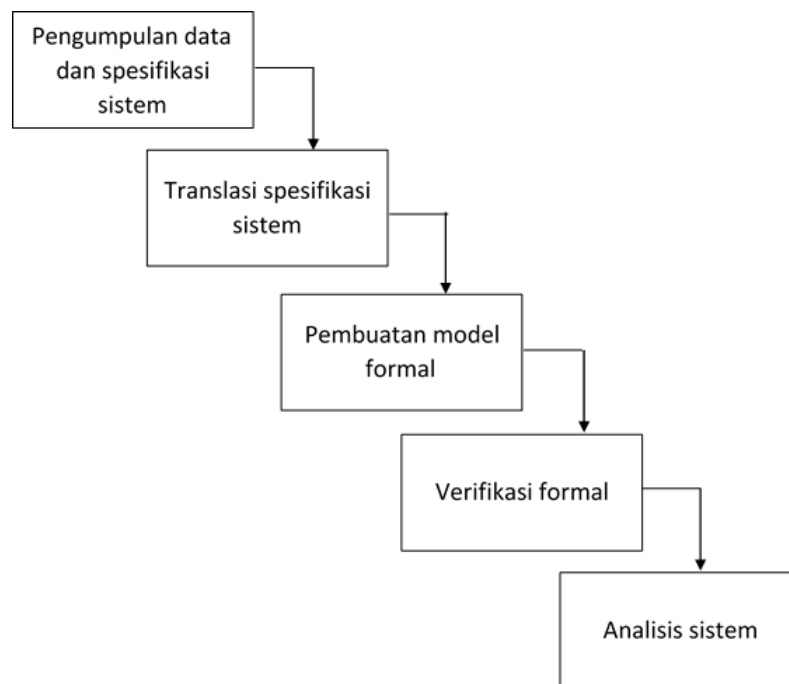
Skrip NuSMV 1 Contoh Penulisan pada NuSMV

```
1: MODULE main
2: VAR
3:   --anggota himpunan state terdiri dari s0, s1, s2, dan s3
4:   state : s0,s1,s2,s3;
5: ASSIGN
6:   --state dimulai dari s0
7:   init(state) := s0;
8:   --transisi antar state
9:   next(state) :=
10:  case
11:    state = s0 : s1;
12:    state = s1 : s0,s2;
13:    state = s2 : s0,s3;
14:    state = s3 : s2;
15:  esac;
16: LTLSPEC  G (! (p & q) -> F r);
17: LTLSPEC  G ( (q & r) -> F p);
```

Bab III

Metodologi

Pada bab ini menjelaskan mengenai metodologi yang digunakan selama pengerjaan tugas akhir.



Gambar 3.1: Contoh metodologi.

3.1 Analisa spesifikasi dan Kebutuhan sistem

Data didapatkan dari

3.2 Translasi spesifikasi sistem

Data yang telah didapatkan akan diterjemahkan dalam

3.3 Pembuatan aplikasi

Data yang telah didapatkan akan diubah dalam model formal

3.4 Pengujian aplikasi

Model formal yang telah dibuatkan akan diverifikasi menggunakan

3.5 Analisis sistem

Hasil verifikasi akan dianalisis

Daftar Pustaka

- [1] M. Huth and M. Ryan, *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge University Press, 2004.

Lampiran

Pada bab ini akan berisi lampiran dari pengerjaan tugas akhir, contohnya skrip NuSMV.