

BAB I

PENDAHULUAN

1.1 Latar Belakang

Satisfiability problem (*The Boolean Satisfiability Problem* atau SAT) merupakan suatu masalah yang menentukan kepuasan yang mana berkaitan dengan pengoptimalisasian kegiatan. *Satisfiability problem* (*SAT Problem*) adalah suatu *problem* atau permasalahan dimana sesuatu hal bisa digunakan untuk memberikan ketentuan apakah suatu formula tersebut adalah formula yang memiliki nilai *satisfiabel* atau *unsatisfiabel*. Sebuah formula dikatakan *satisfiabel* apabila terdapat kombinasi nilai - nilai kebenaran variabel yang terkandung didalamnya tersebut bernilai benar (*True/ (T)*) dengan kata lain bila mampu menggambarkan nilai benar-salah untuk suatu variabel sedemikian rupa sehingga menghasilkan pernyataan yang benar (*valid*), bila tidak *valid* maka disebut *unsatisfiable* (M. Huth dan M. Ryan, 2004). Ada berbagai metode untuk menyelesaikan masalah *Satisfiability* tersebut, beberapa diantaranya adalah dengan menggunakan algoritma Genetic dan DPLL. Untuk penelitian ini akan digunakan algoritma DPLL. Algoritma DPLL adalah algoritma yang menggunakan pencarian heuristik dilengkapi dengan sistem *backtracking* dalam pencarian heuristiknya sebagai cara untuk menyelesaikan SAT Problem yang diberikan. Sistem *backtracking* ini di fungsikan untuk menghindari pencarian pada bagian yang sudah pasti bukan merupakan hasil yang *Satisfiable*. Namun walaupun dengan sistem *backtracking* tersebut, masih akan cukup sulit dan memakan banyak waktu untuk menyelesaikan SAT Problem yang rumit dan panjang secara manual. Oleh karena itu dibuat lah sebuah aplikasi yang digunakan untuk menyelesaikan SAT Problem tersebut yang disebut dengan SAT Solver. Dalam penelitian ini dimaksudkan untuk membuat sebuah SAT Solver menggunakan bahasa pemrograman AHK (Auto Hotkey) script.

AHK Script merupakan bahasa pemrograman yang ditujukan untuk pembuatan makro, yaitu sebuah program dimana program tersebut akan menjalankan satu atau lebih perintah hanya dengan sedikit interaksi dari pengguna. Namun kini AHK script juga sudah mampu digunakan untuk pembuatan berbagai jenis program lainnya.

AHK Script memiliki beberapa fitur yang membuatnya unggul dari beberapa bahasa pemrograman yang lain. Salah satu keunggulannya adalah kemampuannya untuk mengintegrasikan dirinya dalam aplikasi lain yang sedang berjalan, sehingga dapat digunakan

untuk membantu kinerja aplikasi tersebut dan juga menggunakan aplikasi yang diintegrasikannya sebagai sarana *input output*. Bukan hanya itu saja, AHK script juga memiliki fitur *dynamic type variable*, dimana sebuah variabel tidak perlu di deklarasikan sebagai sebuah type data spesifik, tetapi variabel tersebut akan mengubah dirinya sendiri sesuai dengan isi dari variabel itu sendiri. Permisalan disini adalah apabila variabel “Var” belum di deklarasikan sebagai tipe data manapun kemudian di isi dengan angka saja, maka secara otomatis type datanya akan berubah menjadi int, double, dsbg. Namun apabila variable tersebut kemudian diisi dengan kata-kata, maka variable tersebut dengan spontan akan mengubah dirinya menjadi tipe data string. Fitur ini juga mampu di tindih dengan cara mendeklarasikan variabel tersebut sebagai type data tertentu sebelum nya, sehingga kemungkinan terjadinya kesalahan karena fitur ini juga lumayan kecil.

AHK Script di sini juga memiliki *library* yang cukup lengkap, bahkan beberapa perintah memiliki lebih dari satu bentuk untuk mempermudah pengguna dalam melakukan penulisan. AHK script juga mampu membaca dan menggunakan *library* dari aplikasi yang diintegrasikannya, walaupun prosesnya cukup rumit untuk dilakukan.

Itulah beberapa fitur-fitur *AHK script* yang lumayan menonjol. Tentu saja bukan hanya itu fitur *AHK script* yang dapat membuatnya dapat bersaing dengan bahasa pemrograman yang lain. Ukurannya yang berada di bawah 20Mb, kompilator yang tidak memakan banyak memory, memiliki fitur “*portable*” sehingga tidak perlu menginstal kembali untuk tiap kali pergantian komputer namun tetap terdapat fitur instal untuk mempermudah pengguna, dapat berjalan pada OS (*Operating System*) lama seperti *windows XP*, dan masih banyak lagi, membuat AHK script cukup populer di kalangan *programmer*, walaupun memang masih banyak yang belum mengetahui keberadaan bahasa pemrograman ini.

1.2 Rumusan Masalah

Menurut deskripsi latar belakang diatas, didapatkan bahwa penelitian ini dilaksanakan karena kesulitan mendapatkan hasil dalam waktu yang singkat yang disebabkan oleh beberapa faktor salah satunya adalah banyaknya perhitungan yang dilakukan apabila *variable* yang digunakan lebih dari dua. Maka dari itu, pada penelitian ini dibuat program Pemecah Sat (Sat Solver) yang menggunakan metode DPLL (*Davis-Putnam-Logemann-Loveland*) dengan AHK Script..

1.3 Batasan Masalah

- a. Program hanya dapat digunakan untuk menyelesaikan masalah SAT (SAT Problem) yang berbentuk Conjunctive yang Normal Form (CNF) dengan format DIMACS.
- b. Program hanya dapat melakukan perhitungan menggunakan metode DPLL.
Untuk mempermudah penggunaan dan pembuatan, interface program akan bersifat minimalis dan simple

1.4 Tujuan Penelitian

Tujuan dalam penelitian ini adalah membuat *SAT Solver* yang dapat menyelesaikan SAT Problem.

1.5 Manfaat Penelitian

Penelitian ini diharapkan bermanfaat untuk :

- a. Mengurangi waktu yang dibutuhkan untuk menyelesaikan perhitungan CNF (conjunctivenormal form)-SAT Problem.
- b. Mengurangi kemungkinan terjadinya kesalahan yang sering terjadi akibat perhitungan manual.

1.6 Metodologi Penelitian

Tugas Akhir ini disusun menggunakan identifikasi masalah, analisis kebutuhan, perancangan, implementasi, dan pengujian sebagai bentuk Metodologi Penelitiannya. Hal ini dilakukan untuk membantu penyelesaian tugas akhir penyelesaian menjadi lebih terkendali. Penjabaran metodologi yang digunakan antara lain:

a. Identifikasi Masalah

Identifikasi masalah dilakukan dengan cara melihat dan memperkirakan permasalahan yang timbul dan apa yang dibutuhkan yang mencakup cara kerja dan ruang lingkup *SAT Solver* yang akan dibuat dengan cara:

1. Melakukan analisis terhadap permasalahan *CNF-SAT* dan cara penyelesaiannya.
2. Melakukan analisis terhadap algoritma DPLL dan implementasinya untuk menyelesaikan permasalahan *CNF-SAT*.

3. Menganalisis penggunaan AHK Script untuk menyelesaikan permasalahan *CNF-SAT* menggunakan algoritma DPLL.
4. Membaca banyak literatur dari berbagai sumber seperti penelitian terdahulu, yang berhubungan dengan penyelesaian tugas akhir ini.

b. Analisis Kebutuhan

Analisis kebutuhan apa saja yang diperlukan untuk membuat program dengan AHK Script yang dapat menyelesaikan permasalahan *CNF-SAT* menggunakan algoritma DPLL dilakukan pada tahapan ini.

c. Perancangan

Dilakukan perancangan program pada tahapan ini dengan metode Algoritma DPLL menggunakan AHK Script agar dapat menerima masukan berupa proposisi logika *boolean* dalam bentuk CNF yang dapat menyelesaikan permasalahan *CNF-SAT*.

d. Implementasi

Implementasi merupakan tahap pengimplementasian dari perancangan yang telah dilakukan pada tahap sebelumnya. Tahapan ini dilakukan dengan mengimplementasikan algoritma DPLL dalam penyelesaian permasalahan *CNF-SAT* menggunakan AHK Script. Dalam penelitian ini juga akan dilakukan implementasi penerapan algoritma DPLL secara manual untuk menyelesaikan permasalahan *CNF-SAT*.

e. Pengujian

Tahapan terakhir dari penelitian ini adalah melakukan pengujian dari tahapan sebelumnya, karena program harus diuji terlebih dahulu apakah program dapat memberikan hasil yang sesuai dengan kebutuhan. Pengujian juga dilakukan dalam rangka mencari tahu apakah *SAT Solver* yang dibuat dapat berjalan dengan lancar. Selain itu tahap ini sekaligus berfungsi untuk melihat kelebihan dan kekurangan *SAT Solver* dalam penerapan penyelesaian permasalahan SAT Problem. Adapun pengujian dilakukan dengan cara penginputan data nyata yang digunakan untuk melihat kemampuan *SAT Solver*.

1.7 Sistematika Penulisan

Sistematika penulisan ini dibuat berdasarkan Buku Panduan Tugas Akhir Fakultas Teknologi Industri Universitas Islam Indonesia Tahun 2013 dan Template Tugas Akhir Jurusan Teknik Informatika Fakultas Teknologi Industri Tahun 2017, dimana sistematika penulisan ini dimaksudkan untuk memberikan gambaran umum mengenai penelitian yang dijalankan. Sistematika penulisan ini dibagi dalam lima bab yaitu:

BAB I PENDAHULUAN

Bab pendahuluan berisi tentang permasalahan umum, latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian serta sistematika penulisan penelitian penyelesaian permasalahan *CNF-SAT*.

BAB II TINJAUAN PUSTAKA

Pada bagian ini menjelaskan teori-teori yang digunakan dalam penelitian penyelesaian permasalahan *CNF-SAT*. Adapun teori yang disampaikan meliputi *Implementasi, Conjunctive Normal Form, Satisfiability, Sat Solver, Algoritma DPLL (Davis-Putnam-Logemann-Loveland) dan AHK Script*.

BAB III METODOLOGI

Metodologi berisi tentang analisis kebutuhan dan perancangan program untuk menyelesaikan permasalahan *CNF-SAT*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bagian ini menjelaskan mengenai hasil implementasi dan pengujian program penyelesaian permasalahan *CNF-SAT menggunakan AHK Script*, yaitu tahapan pengujian untuk mengetahui apakah *SAT Solver* yang dibuat dapat berjalan dengan lancar. Selain itu tahap ini juga berfungsi untuk melihat kelebihan dan kekurangan *SAT Solver* dalam penerapan penyelesaian permasalahan SAT Problem. Adapun pengujian dilakukan dengan cara penginputan data dummy yang digunakan untuk melihat kemampuan *SAT Solver*.

BAB V KESIMPULAN DAN SARAN

Padabab ini akan dijelaskan mengenai kesimpulan yang didapatkan setelah penelitian yang dilakukan selesai. Selain itu penelitian ini juga akan memberikan saran sebagai sarana penelitian selanjutnya.

