

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam kehidupan sehari-hari terdapat berbagai macam permasalahan yang terjadi. Beberapa permasalahan tersebut dapat diselesaikan secara matematis, salah satu metode yang dapat digunakan adalah SAT *Problem* (*Boolean Satisfiability Problem*). SAT *Problem* dapat digunakan untuk menentukan permasalahan kombinasi nilai kebenaran semua variabel proposisi dalam sebuah klausa/formula logika proposisi, sehingga membuat klausa/formula logika proposisi tersebut bernilai benar (Du, Gu, & Pardalos, 1997).

Karena tergolong dalam permasalahan NP-*Complete* (*Non Polynomial-Complete*), maka SAT *Problem* memiliki kompleksitas yang sangat besar. Sehingga apabila permasalahan SAT *Problem* diselesaikan dengan cara manual, tidak akan mudah untuk diselesaikan dan memakan waktu yang relatif lama. Untuk mempermudah proses penyelesaian masalah menggunakan metode SAT perlu adanya SAT *Solver*. SAT *Solver* adalah perangkat lunak atau *software* yang digunakan untuk menyelesaikan SAT *Problem* (Huth & Ryan, 2004).

Selama beberapa dekade terakhir penelitian tentang algoritma pembentuk SAT *Solver* yang efisien dan *scalable* untuk menyelesaikan SAT *Problem* berkembang dengan pesat. Perkembangan ini telah banyak berkontribusi pada kemajuan teknologi terutama dalam hal perkembangan kemampuan kita (manusia) untuk secara otomatis memecahkan masalah yang melibatkan puluhan ribu variabel dan jutaan klausa (*constraint*) (Ohrimenko, Stuckey, & Codish, 2007). Dalam dunia nyata contoh penggunaan SAT *Solver* adalah dalam masalah penjadwalan/perencanaan (E.Nugraheni, 2008), sistem manajemen bisnis (Jannach & Bundgaard, 2007), permasalahan tes industri (Ansótegui, Bonet, & Levy, 2013), perancangan AI (*Artificial Intelligence*) (Hamadi, Jabbour, & Saïs, 2016), permasalahan EDA (*Electronic Design Automation*) (Nam, Sakallah, & Rutenbar, 2002), dan sebagainya.

Dalam berbagai kasus dan permasalahan yang kompleks pada dunia nyata, SAT *Solver* terkadang akan mengalami kesulitan. Hal ini disebabkan karena pada awalnya SAT *Solver* dibuat dengan menggunakan algoritma *basic* (dasar) dan hanya untuk mengatasi permasalahan yang mudah. Oleh karena itu pengoptimalan algoritma pembentuk SAT *Solver* akan menjadi penting (*essential*). Peningkatan SAT *Solver* berarti kualitas solusi yang didapatkan akan lebih baik, proses komputasi yang memakan waktu lebih sedikit dan *resource* (biaya) yang diperlukan akan jadi jauh lebih rendah daripada SAT *Solver* sebelumnya.

Terdapat berbagai macam metode/algoritma yang dapat digunakan untuk membentuk SAT *Solver*. Namun pada umumnya yang digunakan adalah metode optimasi yang terbagi menjadi metode eksak (*exact method* atau *analitical method*) dan metode pendekatan (*approximate method*). Metode eksak adalah metode yang menghasilkan penyelesaian paling optimal. Hal tersebut dapat dibuktikan karena pembuktian metodenya dilakukan secara analitis menggunakan metode matematis. Sedangkan metode pendekatan tidak menjamin bahwa hasil penyelesaiannya adalah yang paling optimal. Tetapi, meskipun metode ini tidak menjamin hasil optimal, tetapi umumnya hasil penyelesaiannya cukup baik. Dari dua metode tersebut metode pendekatan lebih sering digunakan dalam pembentukan SAT *Solver*.

Pada metode pendekatan terdapat berbagai macam metode. Salah duanya adalah metode heuristik dan metaheuristik. Metode metaheuristik pada sebenarnya adalah metode pendekatan yang didasarkan pada metode heuristik. Sehingga sering kali metode heuristik diintegrasikan di dalam metode metaheuristik. Perbedaan mendasar keduanya adalah metode heuristik bersifat *problem dependent* sedangkan metode metaheuristik bersifat *problem independent*. *Problem dependent* artinya bergantung pada permasalahan, jadi metode heuristik itu hanya bisa dipakai untuk jenis permasalahan tertentu. Misalnya, metode *Nearest Neighborhood* (NN) termasuk metode heuristik. Metode hanya bisa dipakai pada permasalahan yang mengenal konsep neighborhood/tetangga. Sedangkan *problem independent* berarti tidak bergantung pada jenis permasalahan. Jadi penerapan metode metaheuristik tidak bergantung pada jenis permasalahan atau bisa dipakai untuk berbagai jenis permasalahan.

Belakangan ini tren perkembangan algoritma pembentuk SAT *Solver* tersebut mengarah ke *Soft Computing*. Salah satu cabang dari *Soft Computing* adalah Algoritma Genetika (*Genetic Algorithm*). Algoritma genetika adalah algoritma metaheuristik yang terinspirasi teori Darwin mengenai proses seleksi alam dan termasuk ke dalam kelas yang lebih besar yaitu Algoritma Evolusioner (*Evolutionary Algorithm*). Algoritma Genetika biasanya digunakan untuk menghasilkan solusi berkualitas tinggi untuk masalah optimasi dan pencarian dengan mengandalkan operator yang terinspirasi alam seperti *mutation*, *crossover*, dan *selection* (Mitchell, 1996). Karena dapat menghasilkan solusi berkualitas tinggi dengan *resource* relatif rendah Algoritma Genetika menjadi kandidat kuat algoritma pembentuk SAT *Solver*.

Sebenarnya sudah banyak SAT *Solver* yang dikembangkan menggunakan algoritma yang efektif dan efisien seperti: GRASP (Marques-Silva & Sakallah, 1996), Satz (Li & Anbulagan, 1997), WalkSat (Marques-Silva & Sakallah, 1996), zChaff (Moskewicz, 2001) dan mChaff (Vizel, Weissenbacher, & Malik, 2015), bahkan sebenarnya sudah ada penelitian mengenai SAT *Solver* menggunakan Algoritma Genetika (Marchiori & Rossi, 1999). Namun semua SAT *Solver* tersebut masih membutuhkan *resource* yang besar untuk dijalankan serta membutuhkan waktu yang lama dalam proses eksekusinya. Hal tersebut dikarenakan meskipun menggunakan metode heuristik dan metaheuristik yang merupakan metode pendekatan solusi yang ditawarkan berupa masih berupa eksak.

Solusi dalam penyelesaian SAT *Problem* adalah pemberian nilai kebenaran kepada variabel proposisi sehingga tepat menyebabkan setiap klausa penyusun formula proposisi bernilai benar. Formula proposisi hanya dapat dinyatakan "*satisfiable*" jika dan hanya jika seluruh klausa penyusunnya bernilai benar atau berupa eksak. Dalam penelitian ini solusi yang ditawarkan adalah pemberian variabel proposisi sehingga mengoptimalkan jumlah klausa yang benar atau berupa pendekatan. Hal ini akan sangat menguntungkan karena pendekatan merupakan masalah optimasi dan Algoritma Genetika adalah algoritma optimasi artinya *resource* yang diperlukan hanya akan bergantung seberapa lama proses optimasi berlangsung.

Oleh karena itu, penelitian ini akan mencoba mengembangkan *SAT Solver* (*software/perangkat lunak*) dengan menggunakan pemrograman JAVA dan konsep pendekatan solusi menggunakan Algoritma Genetika. Penerapan konsep pendekatan solusi diharapkan dapat meningkatkan kinerja *SAT Solver* dengan Algoritma genetika.

## 1.2 Rumusan Masalah

Dari latar belakang dan identifikasi masalah di atas, maka dapat dihasilkan rumusan masalah sebagai berikut:

Bagaimana mengimplementasikan Algoritma Genetika dengan solusi pendekatan untuk menyelesaikan *SAT Problem* menggunakan bahasa pemrograman JAVA?

## 1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Formula yang diselesaikan berbentuk CNF (*Conjunctive Normal Form*).
2. Program yang dibangun berbentuk konsol.
3. Masukan dan keluaran program berbentuk teks.

## 1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

1. Membangun *SAT Solver* (*software/perangkat lunak* yang dapat menyelesaikan *SAT Problem*).
2. Mengimplementasikan solusi pendekatan menggunakan algoritma genetika untuk menyelesaikan *SAT Problem* pada *SAT Solver*.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mempermudah dan mempercepat penyelesaian *SAT Problem*.
2. Mengatasi kompleksitas penyelesaian *SAT Problem* dengan Algoritma Genetika.
3. Sebagai bahan rujukan untuk pengembangan penelitian selanjutnya mengenai *SAT Problem*.

## 1.6 Sistematika Penulisan

Sistematika penulisan digunakan sebagai pedoman/gambaran umum mengenai penelitian yang dijalankan. Sistematika penulisan terbagi dalam lima bab sebagai berikut:

### BAB I PENDAHULUAN

Pendahuluan membahas permasalahan umum tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian serta sistematika penulisan penelitian penyelesaian permasalahan SAT *Problem*.

### BAB II LANDASAN TEORI

Pada bagian ini dijelaskan teori-teori yang digunakan dalam penelitian ini untuk penyelesaian permasalahan SAT *Problem*. Adapun teori yang disampaikan meliputi SAT *Problem*, NP-Complete, CNF, Algoritma Genetika, dan JAVA.

### BAB III METODOLOGI

Metodologi menjabarkan tentang analisis kebutuhan dan perancangan *software*/perangkat lunak untuk menyelesaikan SAT *Problem* dalam penelitian ini.

### BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bagian ini menjelaskan mengenai langkah-langkah dan hasil implementasi serta pengujian SAT *Solver* yang telah dikembangkan.

### BAB V KESIMPULAN DAN SARAN

Pada bagian ini akan dijabarkan kesimpulan yang didapatkan pada pengujian dan penelitian ini secara keseluruhan. Selain itu juga diberikan saran untuk penelitian selanjutnya.