

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi

Metodologi adalah tahapan – tahapan penelitian yang perlu diambil agar dapat mempermudah penyusunan dan penyelesaian Tugas Akhir ini, yaitu implementasi (CNF) SAT Solver dengan metode algoritma DPLL menggunakan AHK Script. Adapun metodologi yang digunakan sebagai berikut.

3.1.1 Studi Pustaka

Pada tahap pertama penelitian ini, akan dilakukan studi pustaka. Pada tahap studi pustaka ini dilakukan proses pencarian penelitian atau referensi yang berhubungan dengan SAT SOLVER dan ALGORITMA DPLL serta AHK SCRIPT. Studi pustaka ini dilakukan dengan cara mencari beberapa penelitian atau referensi yang berhubungan dengan topik tugas akhir ini dan membandingkan antar penelitian tersebut. Dengan tahapan ini diharapkan dapat menghasilkan rencana solusi yang akan dicoba untuk dapat diterapkan.

3.1.2 Pengumpulan Data

Pada tahap pengumpulan data ini dikumpulkannya data berupa input user yang berupa CNF-SAT Problem

3.1.3 Analisis

Tahap selanjutnya yaitu analisis. Pada tahap analisis dilakukan analisa dengan cara menganalisis data (input user) berdasar dengan teori maupun rumus rumus yang ada kemudian mengkonversikan menjadi bahasa yang dimengerti komputer kemudian menghitung hasil dari input. Pada tahapan analisis, terdapat 4 analisis kebutuhan yang akan diuraikan satu persatu di bawah ini :

a. Analisis kebutuhan

Dalam Tahapan Analisis kebutuhan dilakukan pengumpulan proses data dan informasi dimana selanjutnya akan digunakan untuk mendukung dan menunjang pembuatan sistem aplikasi yang akan dibuat sehingga memperoleh jawaban dari rumusan masalah yang ada. Pada penelitian ini metode yang digunakan adalah metode studi pustaka. Dimana,

pengumpulan data yang dilakukan pada penelitian ini adalah dengan melakukan studi pustaka yaitu pencarian informasi melalui media seperti buku, jurnal, ataupun internet. Studi pustaka dilakukan dengan mencari literature yang sesuai dengan penelitian yang membahas tentang *SAT solver* dan metode DPLL (*Davis-Putnam-Logemann-Loveland*) serta mencari informasi perangkat keras ataupun perangkat lunak yang digunakan agar sesuai dengan sistem dan sistem bisa berjalan dan berfungsi seperti yang diharapkan.

b. Analisis Kebutuhan Fungsi

Tahapan Analisis kebutuhan fungsi merupakan tahapan untuk menetapkan fungsi yang dapat dilakukan oleh sistem, sehingga rumusan masalah yang ada dapat terjawab. Dalam sistem ini memiliki fungsi-fungsi sebagai berikut :

User dapat menginput berupa logika proposisi dalam bentuk Conjunctive Normal Form (CNF) menggunakan format DIMACS, Sistem dapat menentukan nilai *false* atau *true* dari logika yang diberikan, Pengujian menggunakan berbagai kombinasi proposisi CNF-SAT.

c. Analisis Kebutuhan Masukan

Tahap analisis kebutuhan masukan yaitu melakukan tahapan untuk menentukan masukan yang dibutuhkan dalam pembuatan aplikasi dalam penyelesaian *SAT problem* ini.

Input yang diberikan harus sudah sesuai dengan kriteria bentuk standar CNF-SAT DIMACS. Adapun kriteria yang dimaksud adalah, terdiri dari 1 atau lebih klausa disjuntif yang dihubungkan dengan konektor logika *AND* (dan) apabila terdiri lebih dari 1 klausa, yang mana aturan penulisan simbol konektor diubah dengan "Space"/spasi untuk *OR*, "Next Line"/Baris yang berbeda untuk *AND*, simbol "-" (*subtractive*/kurang) untuk negasi, angka "0" (*zero/no*) untuk menandakan akhir klausa dan angka numeric selain "0" untuk tiap literal. Contoh konversi masukan logika proposisi yang dilakukan *SAT Solver* dapat dilihat pada Tabel 3.1 berikut ini.

Tabel 3.1 Konversi logika proposisi menjadi format DIMACS CNF dan penyelesaiannya

Keterangan	Logika Proposisi	Konversi CNF
Setiap literal dirubah menjadi angka, konektor “and” diubah menjadi pergantian baris, konektor “or” dirubah menjadi spasi dan di tambahkan 0 pada setiap akhir klausa	$(a \vee b) \wedge (b \vee c)$	1 2 0 2 3 0
Mengubah nilai variable menjadi <i>True</i> atau <i>False</i> sesuai metode yang di gunakan	1 2 0 2 3 0	1 T 0 T 3 0
Penghapusan klausa yang memiliki nilai <i>True</i> di dalamnya	1 T 0 T 3 0	(Formula kosong)
Memberikan nilai <i>Satisfiable</i> apabila tidak terdapat sisa klausa dalam formula	(Formula Kosong)	<i>Satisfiable</i>

d. Analisis Kebutuhan Keluaran

Pada tahapan ini akan dibuat sebuah aplikasi yang menentukan masukan yang diberikan apakah *satisfiable* atau *unsatisfiable*. Dalam penelitian ini Kebutuhan keluarannya adalah sebagai berikut :

Kalimat yang mengatakan apakah masukan "*satisfiable*" atau "*unsatisfiable*", dan hasil langkah demi langkah dari setiap kalkulasi yang dilakukan oleh *SAT Solver*.

e. Analisis Kebutuhan Perangkat Lunak

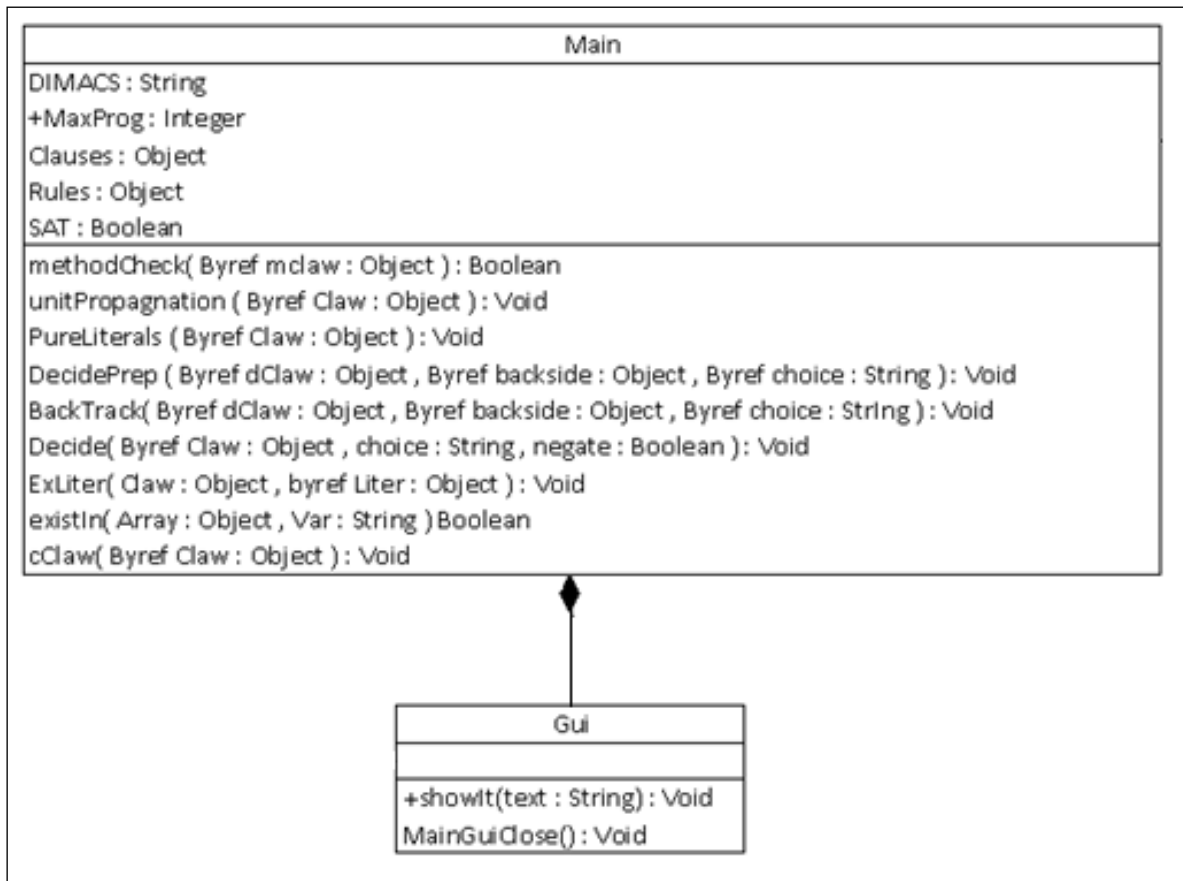
Dalam penelitian ini menggunakan perangkat lunak dalam menuliskan kode program. Adapun perangkat lunak yang digunakan dalam penelitian ini adalah *AHK Script*.

3.1.4 Perancangan

Setelah menyelesaikan tahapan analisis dilanjutkan dengan perancangan. Pada tahap perancangan ini, berdasarkan pada tahapan sebelumnya. Perancangan ini merupakan suatu cara untuk memperjelas bentuk rancangan dari sebuah penelitian. Tahap ini merupakan tahap yang penting sebelum dilanjutkan pada tahap berikutnya dalam sebuah penelitian. Tahapan ini merupakan tahapan untuk mendapatkan suatu metode yang nantinya akan digunakan dalam penelitian, sehingga sebuah penelitian akan mampu mengevaluasi dan mengidentifikasi kendala-kendala dalam pengembangan sistem aplikasi. Perancangan juga akan dapat memberikan penjelasan tentang gambaran sebuah sistem yang dibutuhkan untuk memenuhi kebutuhan yang telah dijabarkan pada tahapan sebelumnya. Perancangan sistem dapat pula menjelaskan bagian – bagian yang terdapat pada sistem secara terperinci agar dapat memenuhi kebutuhan sistem dan pengguna menjadi lebih mudah untuk menggunakan system yang digunakan. penguraikan tentang beberapa rancangan yaitu :

Diagram Kelas

Diagram kelas adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam suatu sistem atau perangkat lunak yang sedang dikembangkan. Diagram kelas meberikan sebuah gambaran statis tentang sistem/perangkat lunak beserta relasi-relasi yang terdapat didalam suatu sistem/perangkat lunak tersebut. Di dalam diagram kelas juga terdapat penjelasan dari masing-masing objek berupa property, metode dan relasi. Diagram kelas dalam aplikasi ini ditunjukkan oleh gambar 3.1 berikut ini.



Gambar 3.1 Diagram Kelas

Pada gambar 3.1 terdapat dua kelas, yaitu Gui dan Main. Adapun fungsi dari kelas-kelas tersebut akan dijelaskan sebagai berikut :

a. Kelas Main

Kelas Main berfungsi untuk memanggil fungsi dari kelas-kelas yang lain serta melakukan penghitungan langsung hasil akhir dari permasalahan yang diberikan. Adapun atribut/method dalam kelas main yang digunakan sebagai berikut:

1. String DIMACS

Variable yang digunakan untuk menyimpan isi file yang berisi formula DIMACS

2. Integer MaxProg

Variable yang digunakan untuk membuat Progress Bar

3. Object Clauses

Object / array untuk menyimpan semua klausa yang ada dalam String yang diberikan

4. Boolean SAT

Boolean sebagai penanda apakah ditemukan hasil yang “satisfiable” atau “unsatisfiable” pada proposisi yang diberikan

5. Object Rules

Berfungsi untuk menyimpan jenis aturan yang tertulis dalam String yang diberikan dan hanya berfungsi sebagai alat pembantu pengecekan kesalahan

6. FunctionmethodCheck(Byref Array)

Fungsi ini adalah fungsi yang berfungsi untuk menentukan metode yang cocok untuk menyelesaikan permasalahan yang diberikan

7. VoidunitPropagation(Byref Array)

Fungsi ini berfungsi untuk menghapus klausa yang memiliki nilai True berdasarkan aturan Unit Propagation

8. VoidPureLiterals(Byref Array)

Fungsi ini berfungsi untuk menghapus klausa yang memiliki nilai True berdasarkan aturan Pure Literals

9. VoidDecidePrep(Byref Array1, Byref Array2, Byref Array3)

Fungsi ini berfungsi untuk mempersiapkan berbagai nilai awal sebelum melakukan decide dan juga membuat Backup untuk penggunaan fungsi BackTrack()

10. VoidBackTrack(Byref Array1, Byref Array2, Byref Array3)

Fungsi ini berfungsi untuk melakukan Backtrack apabila terjadi kondisi dimana *SAT Solver* harus melakukan Backtrack untuk menyelesaikan permasalahan yang diberikan

11. VoidDicide(Byref Array, String, Boolean)

Fungsi ini berfungsi untuk menghapus klausa yang memiliki nilai True berdasarkan aturan Decide

12. VoidExLiter(Array1, Byref Array2)

Fungsi ini berfungsi untuk mengekstrak literal yang terdapat pada permasalahan

13. FunctionexistIn(Array, String)

Fungsi ini berfungsi untuk menentukan apakah terdapat nilai yang setara dengan nilai String yang diberikan di dalam Array yang diberikan

14. Void cClaw (Byref Array)

Memberikan keluaran berupa seluruh isi dari Array yang diberikan

b. Kelas GUI

Kelas GUI berfungsi untuk menampilkan interface dan menerima masukan dari *user*. Adapun atribut /method dalam kelas GUI ditunjukkan pada tabel 3.3.

1. VoidshowIt(String)

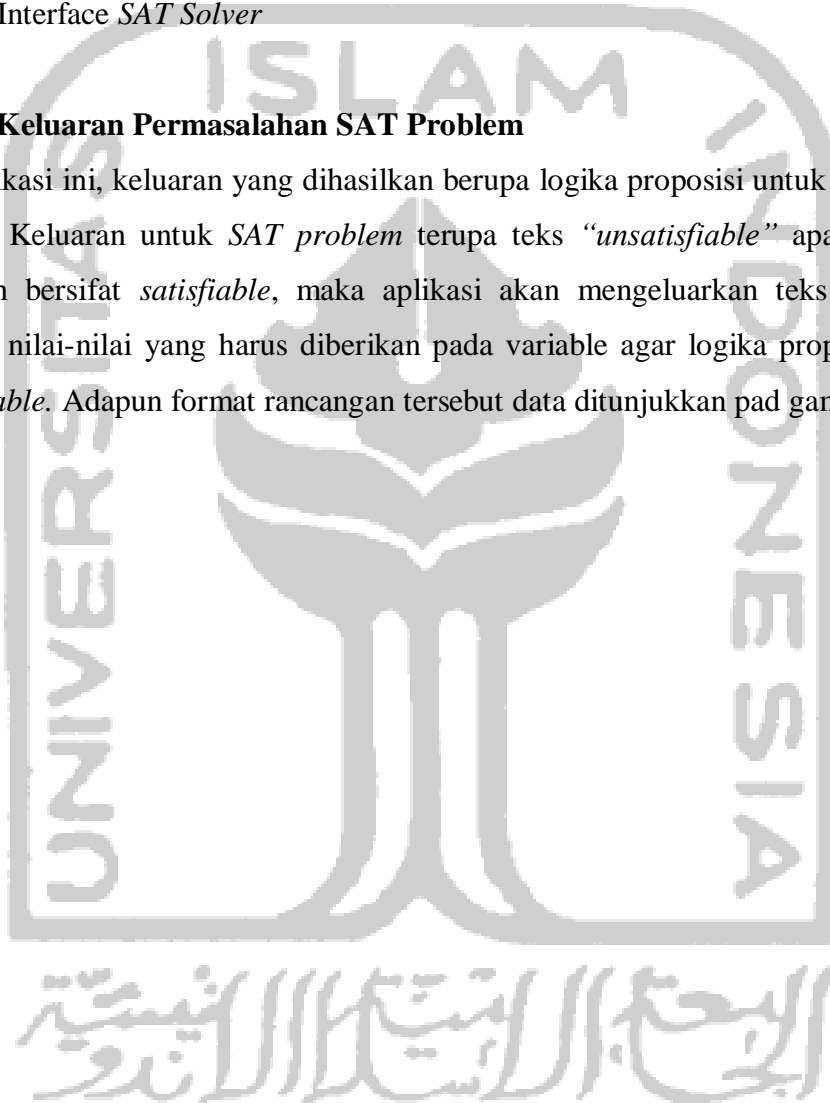
Berfungsi untuk menyimpan memberikan keluaran berupa String yang diberikan.

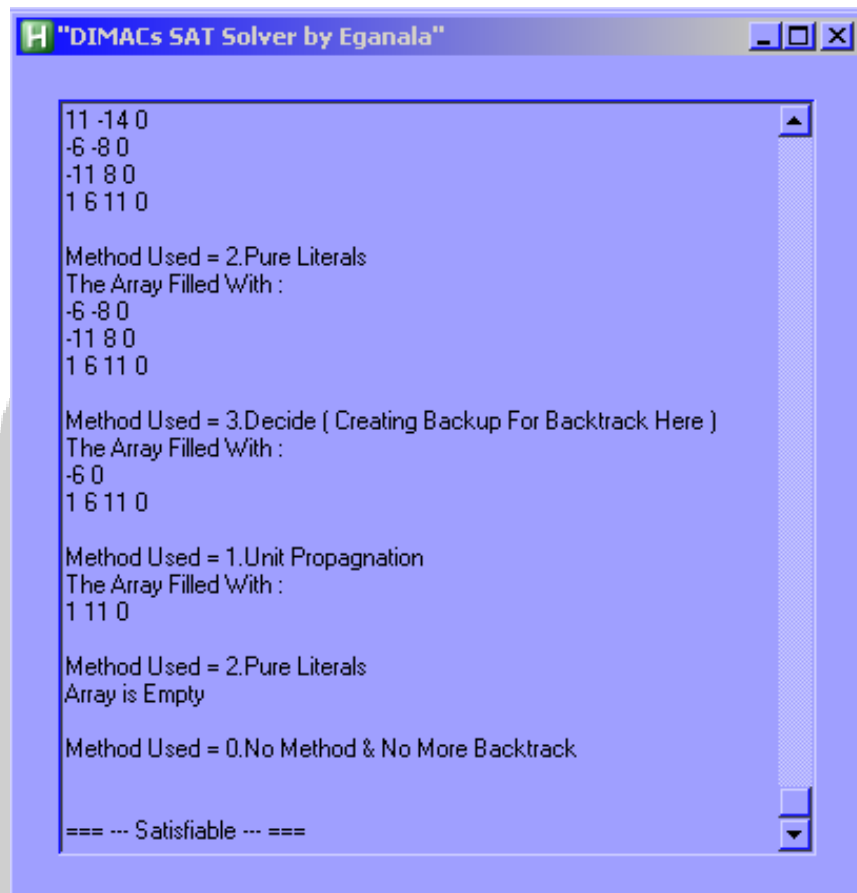
2. VoidMainGuiClose()

Fungsi ini adalah fungsi yang secara otomatis menutup aplikasi apabila pengguna menutup User Interface *SAT Solver*

Perancangan Keluaran Permasalahan SAT Problem

Pada aplikasi ini, keluaran yang dihasilkan berupa logika proposisi untuk permasalahan *SAT Problem*. Keluaran untuk *SAT problem* berupa teks “*unsatisfiable*” apabila masukan yang diberikan bersifat *satisfiable*, maka aplikasi akan mengeluarkan teks “*satisfiable*” diikuti dengan nilai-nilai yang harus diberikan pada variable agar logika proposisi tersebut bernilai *satisfiable*. Adapun format rancangan tersebut data ditunjukkan pada gambar berikut.





```

H "DIMACs SAT Solver by Eganala"
11 -14 0
-6 -8 0
-11 8 0
1 6 11 0

Method Used = 2.Pure Literals
The Array Filled With :
-6 -8 0
-11 8 0
1 6 11 0

Method Used = 3.Decide ( Creating Backup For Backtrack Here )
The Array Filled With :
-6 0
1 6 11 0

Method Used = 1.Unit Propagnation
The Array Filled With :
1 11 0

Method Used = 2.Pure Literals
Array is Empty

Method Used = 0.No Method & No More Backtrack

=== --- Satisfiable --- ===

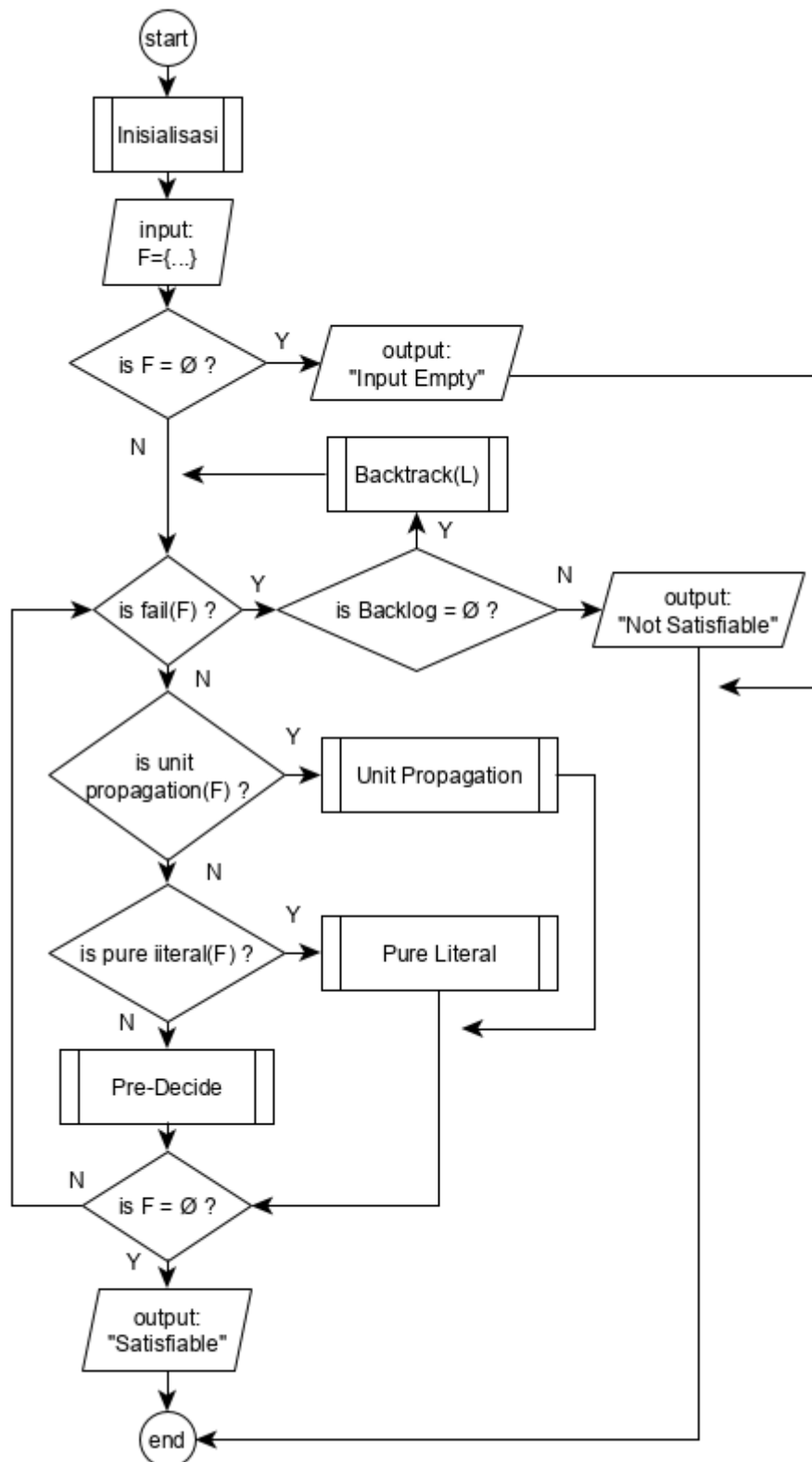
```

Gambar 3.2 Keluaran SAT Problem

Seperti yang terlihat pada Gambar 3.2, apabila formula yang diberikan bernilai *satisfiable*, maka akan menampilkan nilai dari variabel-variabel yang diberikan agar dapat menghasilkan formula yang *satisfied*.

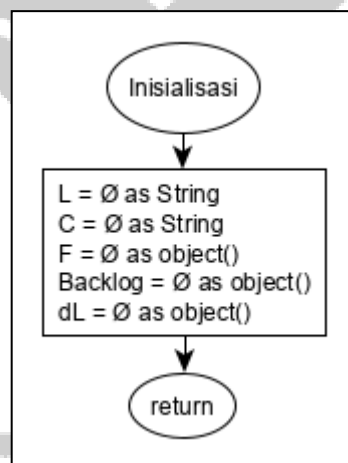
Flowchart SAT Solver

Flowchart adalah gambaran dalam bentuk diagram alir dari algoritma dalam suatu aplikasi yang menyatakan arah alur aplikasi dalam menyelesaikan suatu masalah. Dalam penelitian ini, *flowchart* dibuat untuk menjelaskan implementasi algoritma DPLL (*Davis-Putnam-Logemann-Loveland*) agar lebih mudah mengubahnya kedalam kode program. *Flowchart* DPLL dapat dijelaskan pada Gambar 3.3.

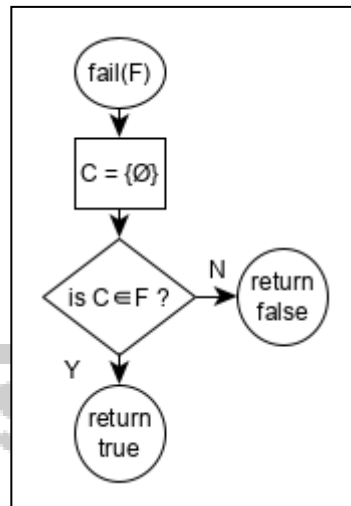


Gambar 3.3 Main Flowchart

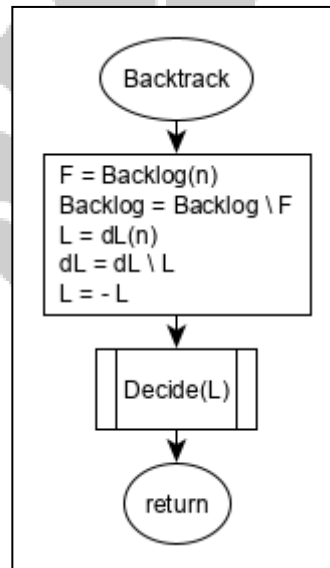
Pada gambar *Main Flowchat* di atas dapat dijelaskan bahwa, setelah *SAT Solver* dijalankan, *SAT Solver* akan melakukan inisialisasi terlebih dahulu, kemudian mengecek apakah input yang di berikan kosong atau tidak baru kemudian *SAT Solver* akan memilih metode yang sesuai untuk menyelesaikan permasalahan yang diberikan pada iterasi tersebut dan menyelesaikannya menggunakan metode yang telah dipilih. Kemudian *SAT Solver* akan menentukan apakah hasil penyelesaian pada itersai tersebut sudah berhasil mencapai nilai *satisfiable* atau tidak dengan cara melihat lagi apakah input yang diberikan sudah berupa variable kosong atau tidak, dengan syarat input tersebut sudah berhasil melewati pengecekan dan pengaplikasian metode pada iterasi pertama. Apabila sudah, maka permasalahan tersebut termasuk permasalahan yang *Satisfiable*. Tetapi apabila bila ditemukannya fail pada iterasi manapun, walaupun iterasi yang pertama sekali pun, dan tidak ditemukannya *Save Point* (*Backlog*) yang telah dibuat, maka permasalahan merupakan permasalahan yang *Not Satisfiable*. Untuk penjabaran fungsi pada *Main Flowchart* dapat dilihat pada Gambar 3.4 ~ Gambar 3.12



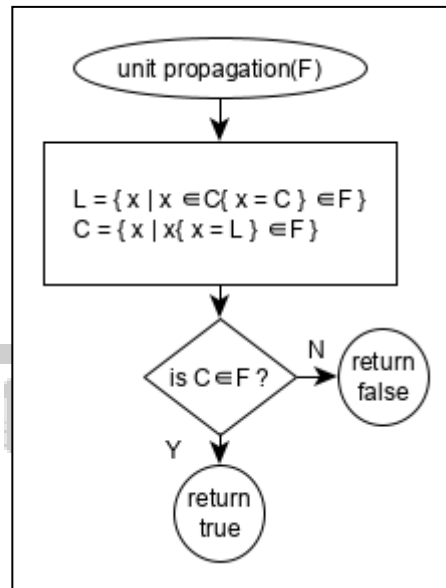
Gambar 3.4 *Flowchart* Inisialisasi



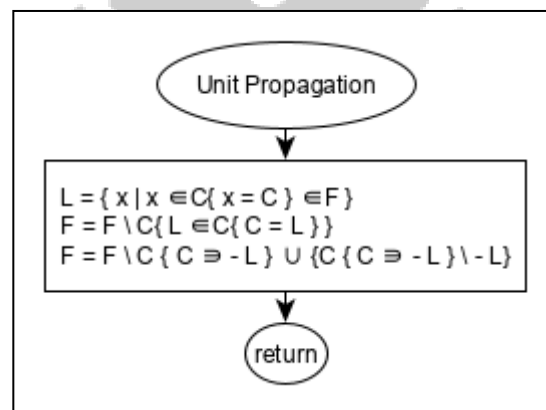
Gambar 3.5 Flowchart 'is' fail(F)



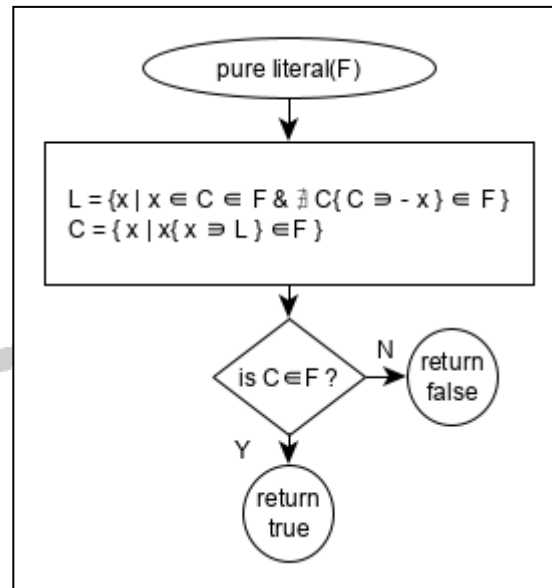
Gambar 3.6 Flowchart Backtrack



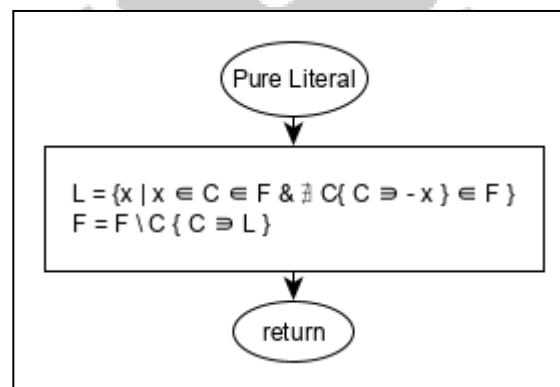
Gambar 3.7 Flowchart 'is' unit propagation(F)



Gambar 3.8 Flowchart Unit Propagation

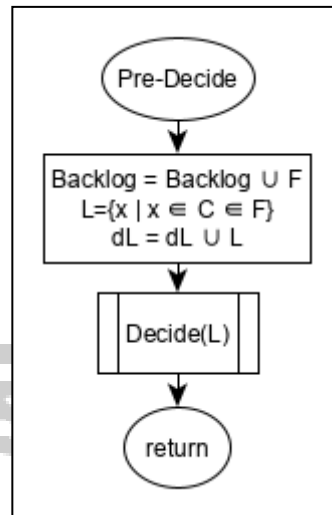
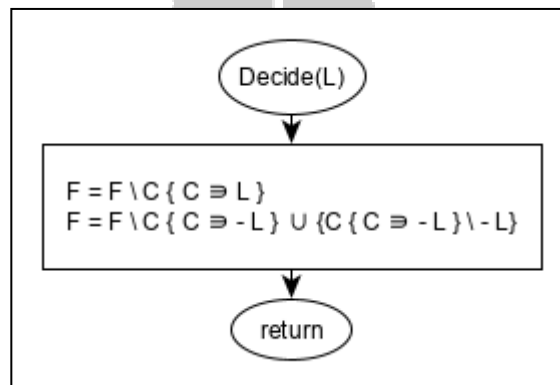


Gambar 3.9 Flowchart 'is' pure literal(F)



Gambar 3.10 Flowchart Pure Literal

الجامعة الإسلامية
 الإسلامية
 الأندلسية

Gambar 3.11 *Flowchart Pre-Decide*Gambar 3.12 *Flowchart Decide(L)*