

Penyederhanaan Tata Bahasa Bebas Konteks dalam Bentuk Normal *Chomsky* Menggunakan PHP

¹Rico Andrian, ²Wamiliana dan ³Ismail Indra Pratama

¹Jurusan Ilmu Komputer FMIPA Unila

³Jurusan Ilmu Komputer FMIPA Unila

² Jurusan Matematika FMIPA Unila

Abstract

In this research we develop an application to simplify context-free grammar into a Chomsky normal form. This software was made to complete the process of simplification of the context-free grammar (CFG) so that user can identify the stages of a process of simplification, easy to understand the simplification at any stages as well as speed up the process. This software is created using the PHP programming language and use context-free grammar (CFG) as the input. The process order is carried out starting from removing epsilon, unit, and useless production so that the result is in Chomsky Normal Form. Black Box testing results with the method of partitioning software equivalence suggests that simplification of software context-free grammar into a Chomsky normal form using PHP was successful and suitable with the user needs.

Keywords: *Chomsky Normal Form (CNF), Context-Free Grammar (CFG), PHP, simplification of the Context-Free Grammar (CFG).*

1. Pendahuluan

Tata bahasa bebas konteks (*Context Free Grammar* atau CFG) merupakan salah satu bahasa formal yang dapat digunakan untuk mendefinisikan sintaks bahasa pemrograman. Suatu tata bahasa bebas konteks dapat berbentuk sangat melebar, sangat menyempit, atau terjadi rekursif kiri, yang semuanya sering dinamakan bentuk tidak normal. Suatu tata bahasa bebas konteks dapat dimodifikasi ruas kanan aturan produksinya sedemikian sehingga panjangnya hanya satu atau dua karakter. Tentu dapat dengan mudah dibayangkan bahwa pohon penurunan string yang terbentuk akan menjadi lebih sederhana, yaitu pohon biner dimana setiap simpul hanya memiliki cabang satu atau dua. Tata bahasa dengan batasan seperti ini disebut tata bahasa bebas konteks dalam bentuk normal *Chomsky* (*Chomsky Normal Form* atau CNF) [1].

Permasalahan yang sering terjadi ketika melakukan penyederhanaan tata bahasa adalah tidak adanya hubungan interaktif antara media yang digunakan seperti buku, majalah dan modul dengan *user* sebagai pengguna program, sehingga cenderung menimbulkan rasa malas dan jenuh bagi *user* jika hanya sekedar memahami teori dan konsep saja tanpa ada penerapan langsung menggunakan media visual seperti komputer. Proses penyederhanaan tata bahasa juga dapat menimbulkan kerumitan jika tidak menggunakan media visual, yaitu ketika melakukan proses penyederhanaan terdapat pohon penurunan yang ambigu yakni pohon penurunan yang berbeda yang menghasilkan aturan produksi yang tidak berarti dan tidak efisien [2].

Penelitian ini mengembangkan suatu perangkat lunak yang dapat membantu *user* memudahkan proses penyederhanaan, khususnya penyederhanaan tata bahasa bebas konteks (*Context Free Grammar*) dalam bentuk normal *Chomsky*. Penelitian ini juga dapat mengatasi rasa malas dan jenuh dari *user* dengan adanya hubungan interaktif antara *user* dengan komputer. *User* merasa lebih berminat dan tertarik untuk melakukan proses penyederhanaan tata bahasa bebas konteks daripada hanya sekedar memahami konsep dan teori.

1.1 Tata Bahasa Bebas Konteks

Tata bahasa bebas konteks (*Context Free Grammar*), biasa disingkat dengan CFG memiliki batasan sebagai berikut:

Tata-bahasa Tipe 2 (*Context -Free Grammar*): $G(\Sigma, N, S, P)$, adalah tata bahasa yang memiliki aturan produksi: $\alpha \rightarrow \beta$ dengan batasan:

α : hanya terdiri dari 1 simbol non terminal saja, atau $\alpha \in N$.

β : tidak dibatasi, atau $\beta \in \{(\Sigma \cup N)^*\}$.

Dimana N adalah himpunan berhingga dari nonterminal, Σ adalah himpunan berhingga dari terminal, S adalah terminal spesifik yang dinamakan simbol awal, dan P adalah himpunan dari produksi-produksi [3].

1.2 Penyederhanaan Tata Bahasa Bebas Konteks

Tujuan dari penyederhanaan adalah melakukan pembatasan sehingga tidak menghasilkan pohon penurunan yang memiliki kerumitan yang tidak perlu atau aturan produksi yang tidak berarti. Cara Penyederhanaan:

1. Penghilangan produksi *useless* (tidak berguna).
2. Penghilangan produksi *unit*.
3. Penghilangan produksi *epsilon* (ϵ).

a. Penghilangan Produksi *Useless*

Produksi *useless* didefinisikan sebagai:

- Produksi yang memuat simbol *variabel* yang tidak memiliki penurunan yang menghasilkan terminal-terminal seluruhnya.
- Produksi yang tidak pernah dicapai dengan penurunan apapun dari simbol awal, sehingga produksi itu redundan (berlebih).

b. Penghilangan Produksi *Unit*

- Produksi dimana ruas kiri dan kanan aturan produksi hanya berupa satu simbol *variabel*, misalkan: $A \rightarrow B, C \rightarrow D$.
- Keberadaannya membuat tata bahasa memiliki kerumitan yang tak perlu.
- Penyederhanaan dilakukan dengan melakukan penggantian aturan produksi *unit*.

c. Penghilangan Produksi *Epsilon* (ϵ)

Produksi ϵ adalah produksi dalam bentuk $\alpha \rightarrow \epsilon$, atau bisa dianggap sebagai produksi kosong. Penghilangan produksi ϵ dilakukan dengan melakukan penggantian produksi yang memuat *variabel* yang bisa menuju produksi ϵ , atau biasa disebut *nullable* [1].

1.3 Bentuk Normal *Chomsky*

Bentuk normal *Chomsky* / *Chomsky Normal Form* (CNF) merupakan salah satu bentuk normal yang sangat berguna untuk tata bahasa bebas konteks (CFG). Bentuk normal *Chomsky* dapat dibuat dari sebuah tata bahasa bebas konteks yang telah mengalami penyederhanaan yaitu penghilangan produksi *useless*, *unit*, dan ϵ . Suatu tata bahasa bebas konteks dapat dibuat menjadi bentuk normal *Chomsky* dengan syarat tata bahasa bebas konteks tersebut:

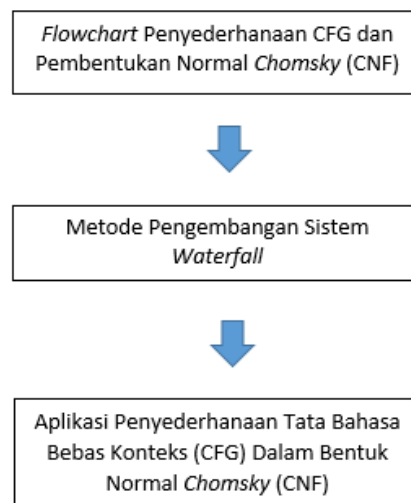
- Tidak memiliki produksi *useless*
- Tidak memiliki produksi *unit*
- Tidak memiliki produksi *epsilon* (ϵ)

Aturan produksi dalam bentuk normal *Chomsky* ruas kanannya tepat berupa sebuah terminal atau dua variabel. Berikut contoh dari [1] :

$$\begin{aligned} A &\rightarrow BC \\ A &\rightarrow b \\ B &\rightarrow a \\ C &\rightarrow BA \mid d \end{aligned}$$

2. Metode

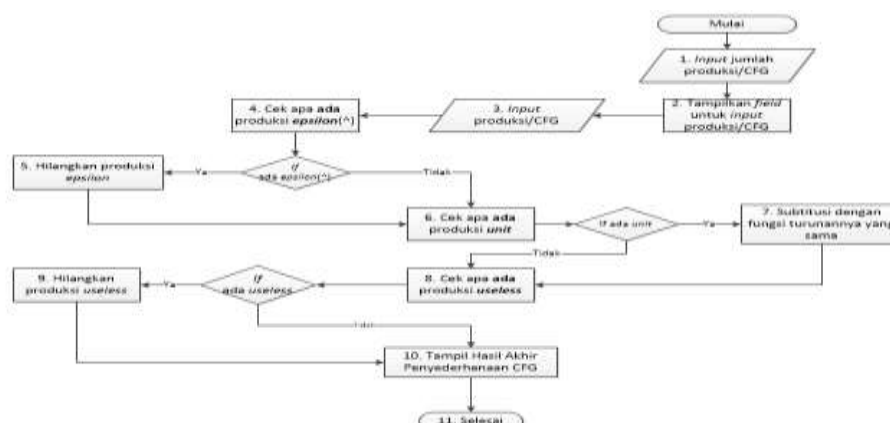
Metode penelitian yang dilakukan bertujuan untuk mengatur dan mempermudah pelaksanaan penelitian. Tahap-tahap penelitian dalam pengembangan aplikasi penyederhanaan tata bahasa bebas konteks (CFG) dalam bentuk normal *Chomsky* (CNF) terdapat pada Gambar 1.



Gambar 1. Tahap-tahap penelitian.

2.1 Flowchart Penyederhanaan Tata Bahasa Bebas Konteks (CFG)

Tahap pertama dalam penelitian ini yaitu membuat *flowchart* penyederhanaan tata bahasa bebas konteks (CFG) berdasarkan algoritma penyederhanaan CFG. *Flowchart* penyederhanaan CFG terdapat pada Gambar 2.



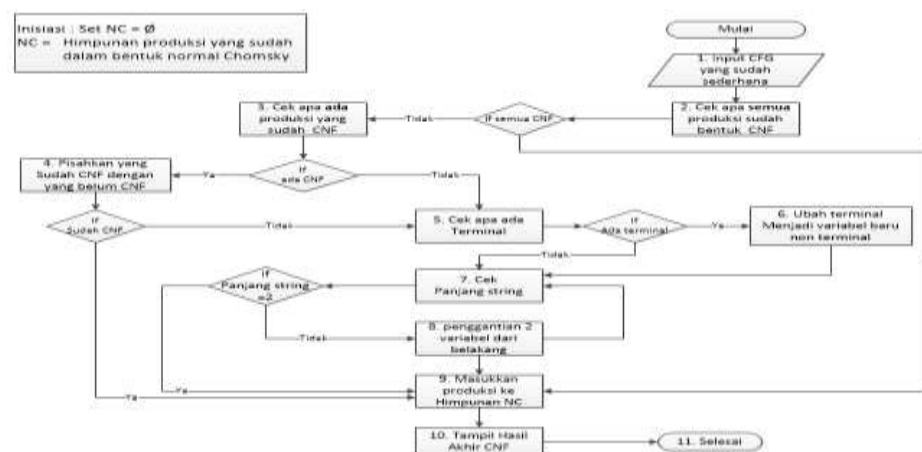
Gambar 2. Flowchart Penyederhanaan CFG.

Algoritma penyederhanaan tata bahasa bebas konteks (CFG) adalah sebagai berikut.

1. Masukkan jumlah produksi yang akan disederhanakan.
2. Tampilkan *field* untuk menginput produksi sesuai jumlah produksi yang dimasukkan pengguna.
3. Masukkan produksi/CFG yang ingin disederhanakan.
4. Cek produksi/CFG, apakah CFG mengandung produksi *epsilon*.
 - Jika ya, ke langkah 5.
 - Jika tidak, ke langkah 6.
5. Hilangkan produksi *epsilon* (" ^ ").
 Misal:
 $A \rightarrow AB, B \rightarrow b \mid ^$ menjadi $A \rightarrow AB, B \rightarrow b$.
 $B \rightarrow ^$ dihilangkan karena merupakan produksi *epsilon*.
6. Cek produksi/CFG, apakah CFG mengandung produksi *unit*.
 - Jika ya, ke langkah 7.
 - Jika tidak, ke langkah 8.
7. Substitusi dengan produksi turunannya yang sama.
 Misal:
 $A \rightarrow B, B \rightarrow c$ menjadi $A \rightarrow c, B \rightarrow c$.
 $A \rightarrow B$ disubstitusikan menjadi $A \rightarrow c$.
8. Cek produksi/CFG, apakah CFG mengandung produksi *useless*.
 - Jika ya, ke langkah 9.
 - Jika tidak, ke langkah 10.
9. Hilangkan produksi *useless*.
 Misal:
 $S \rightarrow ABC \mid AB, A \rightarrow a, B \rightarrow b$ menjadi $S \rightarrow AB, A \rightarrow a, B \rightarrow b$.
 $S \rightarrow ABC$ dihilangkan karena C merupakan produksi *useless*.
10. Tampilkan hasil produksi/CFG yang sudah disederhanakan.
11. Selesai.

2.2 Flowchart Pembentukan Normal Chomsky (CNF)

Tahap selanjutnya dalam penelitian ini yaitu membuat *flowchart* pembentukan normal Chomsky (CNF) berdasarkan algoritma pembentukan normal Chomsky. *Flowchart* pembentukan normal Chomsky terdapat pada Gambar 3.



Gambar 3. Flowchart Pembentukan Normal Chomsky (CNF).

Algoritma Aplikasi Penyederhanaan Tata Bahasa Bebas Konteks (CFG) dalam Bentuk Normal Chomsky (CNF) adalah sebagai berikut.

Inisiasi: Set $NC = \emptyset$

(NC = Himpunan produksi yang sudah dalam bentuk normal *Chomsky*)

1. Masukkan CFG yang sudah disederhanakan.
2. Cek apakah **semua** produksi sudah dalam bentuk CNF.
 - a. Jika ya, ke langkah 9.
 - b. Jika tidak, lanjut ke langkah 3.
3. Cek apakah **ada** produksi yang sudah dalam bentuk CNF.
 - a. Jika ada, lanjut ke langkah 4.
 - b. Jika tidak ada, langsung ke langkah 5.
4. Pisahkan CFG yang sudah dalam bentuk CNF dengan CFG yang belum CNF.
 - a. Jika CFG sudah dalam bentuk CNF, ke langkah 9.
 - b. Jika CFG belum dalam bentuk CNF, lanjut ke langkah 5.
5. Cek apakah string produksi yang ruas kanan memuat simbol terminal.
 - a. Jika ya, ke langkah 6.
 - b. Jika tidak, langsung ke langkah 7.
6. Ubah simbol terminal menjadi variabel baru berupa variabel non terminal.
7. Cek panjang string produksi ruas kanan.
 - a. Jika panjang string = 2 ke langkah 9.
 - b. Jika panjang string >2, ke langkah 8.
8. Lakukan penggantian 2 variabel dari belakang secara berulang-ulang hingga tersisa 2 variabel non terminal.
9. Masukkan produksi ke himpunan NC.
10. Tampilkan hasil produksi yang sudah dalam bentuk CNF, serta variabel baru yang dihasilkan selama proses penyederhanaan.
11. Selesai.

2.3 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Waterfall / Linear Sequential Model*. Metode ini bersifat sistematis atau secara berurutan dalam membangun suatu sistem. Tahapan pengembangan *waterfall* dalam pembuatan aplikasi penyederhanaan tata bahasa bebas konteks ke bentuk normal *Chomsky* antara lain, *system engineering, requirement analysis, design, coding, testing, maintenance* [4].

2.3.1 System Engineering

Tahap pertama dilakukan pengumpulan data dan penetapan kebutuhan semua elemen sistem. Pengumpulan data dilakukan dengan mendefinisikan tentang sistem atau aplikasi yang akan dikembangkan serta mendefinisikan manfaat dan tujuan pengembangan sistem atau aplikasi tersebut. Penelitian ini akan mengangkat tema tentang penyederhanaan tata bahasa bebas konteks (CFG) dalam bentuk normal *Chomsky* (CNF).

2.3.2 Analisis/ Requirement Analysis

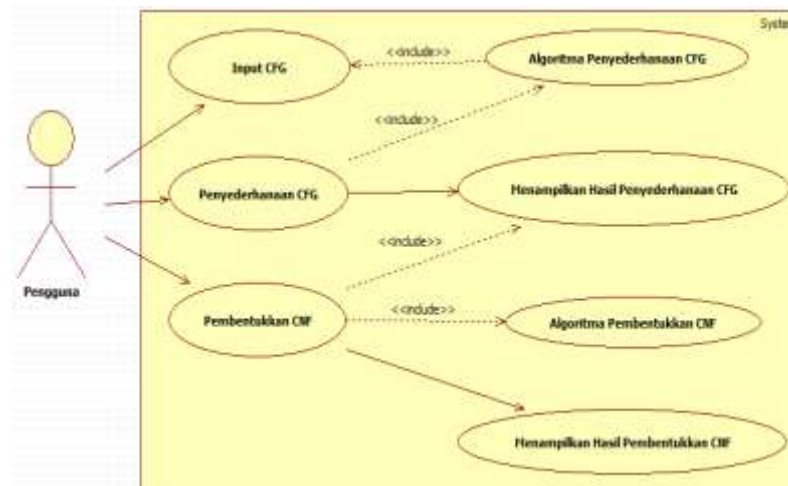
Tahap analisis/*requirement analysis* adalah melakukan analisis terhadap permasalahan-permasalahan yang dihadapi, analisis perangkat lunak dan perangkat keras yang dibutuhkan dalam pembuatan aplikasi, analisis fungsi-fungsi yang dibutuhkan serta kebutuhan-kebutuhan lain dalam pembuatan aplikasi ini. *Domain* informasi dari *software*, seperti fungsi-fungsi yang dibutuhkan, *user interface*, dan sebagainya, harus dimengerti untuk mengetahui sifat dari aplikasi yang akan dibuat. Tujuan analisis adalah agar menemukan kelemahan-kelemahan sistem atau aplikasi yang akan dibuat sehingga dapat diusulkan perbaikannya.

2.3.3 Perancangan/ Design

Perancangan atau *design* merupakan tahap penerjemahan dari kebutuhan atau data yang telah dianalisis menjadi bentuk yang mudah dimengerti oleh pemakai (*user*). Desain dari sistem pada penelitian ini digambarkan dengan model UML berupa *use case diagram* dan *activity diagram*.

a. Use Case Diagram

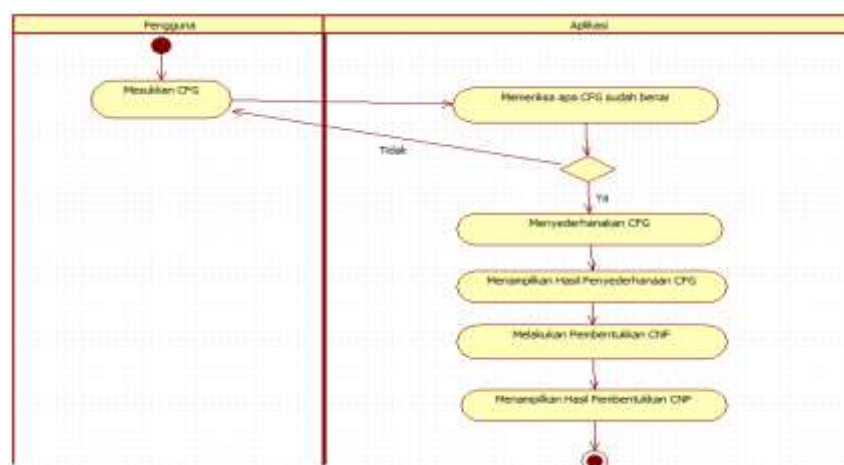
Use case diagram aplikasi penyederhanaan tata bahasa bebas konteks dalam bentuk normal *Chomsky* terdapat pada Gambar 4.



Gambar 4. *Use Case Diagram* Aplikasi Penyederhanaan Tata Bahasa Bebas Konteks Dalam Bentuk Normal *Chomsky* (CNF).

b. Activity Diagram

Activity diagram dari aplikasi penyederhanaan tata bahasa bebas konteks (CFG) dalam bentuk normal *Chomsky* (CNF) terdapat pada Gambar 5.



Gambar 5. *Activity Diagram* Aplikasi Penyederhanaan Tata Bahasa Bebas Konteks Dalam Bentuk Normal *Chomsky* (CNF).

2.3.4 Implementasi

Hasil implementasi dalam penelitian ini terdiri dari tampilan antarmuka aplikasi dan fungsi-fungsi aplikasi. Tampilan antarmuka aplikasi terdiri dari beberapa halaman antara lain, halaman utama (*Home*), halaman gambaran umum, halaman aturan *Chomsky* dan halaman bantuan. Tampilan halaman utama aplikasi penyederhanaan tata bahasa bebas konteks (CFG) dalam bentuk normal *Chomsky* adalah sebagai berikut.

a. Halaman Utama (*Home*)

Halaman utama (*Home*) adalah tampilan awal aplikasi yang muncul pertama kali ketika aplikasi dijalankan. Halaman ini merupakan halaman yang berfungsi untuk melakukan proses penyederhanaan CFG dan proses pembentukan normal *Chomsky*. Halaman ini terdiri dari empat *form* yaitu *form input* jumlah produksi, *form input* produksi, *form* hasil penyederhanaan CFG dan *form* hasil pembentukan normal *Chomsky* (CNF).

- **Form input Jumlah Produksi**

Form input jumlah produksi ini merupakan *form* yang berfungsi untuk memasukkan jumlah produksi atau CFG yang ingin disederhanakan.

Form input jumlah produksi aplikasi penyederhanaan tata bahasa bebas konteks dalam bentuk normal *Chomsky* dapat dilihat pada Gambar 6.

Gambar 6. *Form input* Jumlah Produksi.

- **Form input Produksi**

Form input produksi merupakan kelanjutan dari *form input* jumlah produksi, setelah pengguna memasukkan jumlah produksi di *form* sebelumnya, maka aplikasi akan secara otomatis menampilkan *field input* produksi sesuai jumlah yang telah diisikan pengguna di *form input* jumlah produksi. *Form input* produksi ini berfungsi untuk memasukkan produksi berupa tata bahasa bebas konteks (CFG) yang ingin disederhanakan. *Form input* produksi aplikasi penyederhanaan tata bahasa bebas konteks dalam bentuk normal *Chomsky* dapat dilihat pada Gambar 7.

Gambar 7. Form input Produksi/ CFG

- **Form Hasil Penyederhanaan CFG**

Form hasil penyederhanaan CFG merupakan *form* yang berfungsi untuk menampilkan hasil dari penyederhanaan CFG setelah pengguna mengklik tombol “Sederhanakan” di *form input* produksi. *Form* hasil penyederhanaan CFG aplikasi penyederhanaan tata bahasa bebas konteks dalam bentuk normal *Chomsky* dapat dilihat pada Gambar 8.

Gambar 8. Form Hasil Penyederhanaan CFG.

- **Form Hasil Pembentukan Normal Chomsky**

Form hasil pembentukan normal Chomsky berfungsi untuk menampilkan hasil pembentukan normal Chomsky setelah pengguna mengklik tombol “convert ke CNF” di form hasil penyederhanaan CFG. Form hasil pembentukan normal Chomsky aplikasi penyederhanaan tata bahasa bebas konteks dalam bentuk normal Chomsky menggunakan PHP dapat dilihat pada Gambar 9.

Gambar 9. Form Hasil Pembentukan Normal Chomsky (CNF).

Penggantian Produksi yang Belum CNF (jika Ruas Kanan Terdapat Simbol Terminal):

$S \rightarrow ABaC$ menjadi $S \rightarrow ABDC$
 $S \rightarrow AuC$ menjadi $S \rightarrow ADC$
 $S \rightarrow Aba$ menjadi $S \rightarrow ABD$
 $S \rightarrow Au$ menjadi $S \rightarrow AD$

Dimana :
 $D = a$

Penggantian Produksi yang Belum CNF (jika Ruas Kanan > 2)

$S \rightarrow ABDC \rightarrow S \rightarrow ABF$
 $S \rightarrow ADC \rightarrow S \rightarrow AF$
 $S \rightarrow ABD \rightarrow S \rightarrow AF$
 $S \rightarrow ABF \rightarrow S \rightarrow AK$

Dimana :

$F = DC$
 $F = BD$
 $K = BF$

Hasil Produksi yang Sudah CNF (jika Ruas Kanan > 2)

$S \rightarrow a$
 $S \rightarrow AB$
 $S \rightarrow AC$
 $S \rightarrow BC$

Hasil Hasil Normal Chomsky (CNF)

$S \rightarrow AB \mid AC \mid BC \mid a$
 $A \rightarrow BC \mid a$
 $B \rightarrow a$
 $C \rightarrow a$
 $D \rightarrow a$
 $E \rightarrow BC$
 $F \rightarrow AC$
 $G \rightarrow BC$

Gambar 9. Form Hasil Pembentukan Normal Chomsky (CNF) (lanjutan).

2.3.5 Pengujian/ Testing

Pengujian aplikasi ini menggunakan metode pengujian *black box testing*, dimana pengujian *black box testing* berfokus pada *output* yang dihasilkan dari *input*. Metode pengujian *black box* yang digunakan pada penelitian ini adalah metode *equivalen partitioning* (EP). Pengujian ini dilakukan dengan membagi *domain* masukan (*input*) ke dalam kelas-kelas sehingga *test case* pada aplikasi dapat diperoleh. Hasil pengujian fungsional dengan menggunakan metode pengujian *equivalen partitioning* pada aplikasi penyederhanaan tata bahasa bebas kontek (CFG) dalam bentuk normal *Chomsky* (CNF) dapat dilihat pada Tabel 1.

Tabel 1. Pengujian *Equivalen Partitioning*.

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Realisasi yang Diharapkan	Hasil
1.	Fungsi pengecekan masukan (<i>input</i>).	Pengujian cek <i>input</i> jumlah produksi.	Pengguna memasukkan jumlah produksi berupa angka 1-55.	Tampil form tabel <i>input</i> produksi sesuai dengan jumlah yang di masukkan.	Berhasil
			Pengguna tidak memasukkan jumlah produksi.	Tampil pemberitahuan “silahkan masukkan jumlah produksi”.	Berhasil
			Pengguna memasukkan jumlah produksi angka “0”.	Tampil pemberitahuan “silahkan masukkan jumlah produksi selain angka 0”.	Berhasil
			Pengguna memasukkan jumlah produksi selain angka.	Tampil pemberitahuan “masukkan jumlah produksi dalam bentuk angka”.	Berhasil
			Pengguna memasukkan jumlah produksi lebih dari 55.	Tampil pemberitahuan “jumlah produksi maksimal 55”.	Berhasil
		Pengujian cek <i>input</i> produksi/ CFG.	Pengguna memasukkan ruas kiri CFG berupa huruf besar A-Z atau simbol karakter yang terdaftar sebagai Non terminal.	CFG <i>valid</i> , CFG dapat disederhanakan.	Berhasil
			Pengguna memasukkan ruas kiri CFG panjangnya lebih besar dari 1.	Tampil pemberitahuan “ <i>error</i> : panjang	Berhasil

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Realisasi yang Diharapkan	Hasil
				karakter ruas kiri harus 1".	
			Pengguna tidak mengisi salah satu <i>field input</i> produksi CFG.	Tampil pemberitahuan "please fill out this field".	Berhasil
			Pengguna memasukkan ruas kiri CFG selain huruf besar A-Z atau simbol karakter yang terdaftar sebagai simbol non terminal.	Tampil pemberitahuan "error: ruas kiri harus berupa huruf (A-Z) atau simbol karakter yang terdaftar".	Berhasil
2.	Fungsi Penyederhanaan CFG.	Pengujian penghilangan produksi <i>epsilon</i> " ^ ".	Pengguna memasukkan produksi/CFG yang mengandung produksi <i>epsilon</i> " ^ ".	Aplikasi dapat melakukan penghilangan produksi yang mengandung <i>epsilon</i> " ^ ".	Berhasil
		Pengujian penghilangan produksi <i>unit</i> .	Pengguna memasukkan produksi/CFG yang mengandung produksi <i>unit</i> .	Aplikasi dapat melakukan penghilangan produksi <i>unit</i> .	Berhasil
		Pengujian penghilangan produksi <i>useless</i> .	Pengguna memasukkan produksi/CFG yang mengandung produksi <i>useless</i> .	Aplikasi dapat melakukan penghilangan produksi <i>useless</i> .	Berhasil
		Pengujian menampilkan hasil akhir penyederhanaan CFG.	Pengguna mengklik tombol "sederhanakan".	Aplikasi dapat menampilkan hasil akhir penyederhanaan CFG.	Berhasil
3.	Fungsi Pembentukan Normal Chomsky.	Pengujian pemisahan CFG yang sudah CNF dengan CFG yang belum CNF.	Pengguna memasukkan produksi CFG yang sudah CNF dan CFG yang belum CNF.	Aplikasi dapat memisahkan CFG yang sudah CNF dengan CFG yang belum CNF.	Berhasil
		Pengujian penggantian simbol	Pengguna mengklik tombol "convert ke CNF".	Aplikasi dapat melakukan penggantian simbol	Berhasil

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Realisasi yang Diharapkan	Hasil
		terminal pada CFG yang belum CNF.		terminal dengan simbol Non terminal.	
		Pengujian penggantian produksi/CFG yang panjang ruas kanannya > 2 .	Pengguna mengklik tombol " <i>convert</i> ke CNF".	Aplikasi dapat melakukan penggantian produksi/CFG yang panjang ruas kanannya > 2 .	Berhasil
		Pengujian menampilkan hasil akhir pembentukan normal <i>Chomsky</i> .	Pengguna mengklik tombol " <i>convert</i> ke CNF".	Aplikasi dapat menampilkan hasil akhir pembentukan normal <i>Chomsky</i> .	Berhasil
4.	Fungsi menampilkan gambaran umum aplikasi.	Pengujian pada menu gambaran umum.	Pengguna mengklik menu gambaran umum.	Aplikasi dapat menampilkan informasi tentang gambaran umum aplikasi.	Berhasil
5.	Fungsi menampilkan aturan pembentukan normal <i>Chomsky</i> .	Pengujian pada menu aturan <i>Chomsky</i> .	Pengguna mengklik menu aturan <i>Chomsky</i> .	Aplikasi dapat menampilkan informasi langkah-langkah pembentukan normal <i>Chomsky</i> secara umum.	Berhasil
6.	Fungsi menampilkan informasi bantuan pada aplikasi.	Pengujian pada menu bantuan. cara pengoperasian aplikasi.	Pengguna mengklik cara pengoperasian aplikasi di menu bantuan.	Aplikasi dapat menampilkan informasi tentang langkah-langkah pengoperasian aplikasi.	Berhasil
		Pengujian pada menu bantuan informasi algoritma penyederhanaan CFG.	Pengguna mengklik algoritma penyederhanaan tata bahasa bebas konteks (CFG) di menu bantuan.	Aplikasi dapat menampilkan informasi algoritma penyederhanaan tata bahasa bebas konteks (CFG).	Berhasil
		Pengujian pada menu bantuan informasi algoritma pembentukan normal <i>Chomsky</i> (CNF).	Pengguna mengklik algoritma pembentukan normal <i>Chomsky</i> (CNF) di menu bantuan.	Aplikasi dapat menampilkan informasi algoritma pembentukan normal <i>Chomsky</i> (CNF).	Berhasil

3. Pembahasan

Hasil pengujian fungsional yang dilakukan dengan *black box testing* menggunakan metode *equivalent partitioning* pada Tabel 1 didapatkan bahwa aplikasi penyederhanaan tata bahasa bebas konteks (CFG) dalam bentuk normal *Chomsky* telah berhasil dan berjalan dengan baik. Pengujian yang dilakukan tidak menemukan kendala apapun dalam aplikasi, berdasarkan kasus uji yang diberikan, terlihat bahwa realisasi yang diharapkan sesuai dengan hasil yang tampil, sehingga dapat dikatakan aplikasi telah berhasil.

Fungsi-fungsi pada aplikasi berjalan dengan baik, berdasarkan Tabel 1 pengujian *equivalent partitioning* didapat hasil sebagai berikut.

1. Aplikasi dapat mengecek masukan (*input*) jumlah produksi yang diterima.
2. Aplikasi dapat mengecek masukan (*input*) produksi/CFG yang diterima.
3. Aplikasi dapat menampilkan seluruh proses yang terjadi dari penyederhanaan tata bahasa bebas konteks (CFG) yang belum disederhanakan sampai menampilkan hasil CFG yang sudah disederhanakan.
4. Aplikasi dapat menampilkan seluruh proses pembentukan normal *Chomsky* dari CFG yang sudah disederhanakan sampai menampilkan hasil akhir dalam bentuk normal *Chomsky*.
5. Aplikasi dapat menampilkan informasi tentang gambaran umum aplikasi.
6. Aplikasi dapat menampilkan informasi langkah-langkah pembentukan normal *Chomsky* secara umum.
7. Aplikasi dapat menampilkan informasi bantuan dari aplikasi.

4. Kesimpulan

Kesimpulan dari penelitian ini adalah :

- Aplikasi penyederhanaan tata bahasa bebas konteks ke dalam bentuk normal *Chomsky* telah berhasil dibangun untuk membantu pengguna dalam melakukan penyederhanaan tata bahasa bebas konteks (CFG) dan mengubah CFG yang sudah disederhanakan ke dalam bentuk normal *Chomsky* (CNF).
- Penyederhanaan tata bahasa bebas konteks (CFG) memiliki tiga proses yang harus dilakukan secara berurutan yaitu, penghilangan produksi *epsilon*, penghilangan produksi *unit*, dan produksi *useless*.
- Setiap produksi *unit* selesai di substitusi akan dihasilkan produksi *useless*.
- Bentuk normal *Chomsky* hanya dapat dilakukan apabila tata bahasa bebas konteks (CFG) sudah disederhanakan.
- Selama proses pembentukan normal *Chomsky* dimungkinkan akan menghasilkan aturan produksi dan simbol variabel baru.

5. Referensi

- [1] Rustamaji, H. C. 2004. *Materi Kuliah Teori Bahasa Dan Auto mata*. Modul Kuliah. Universitas Pembangunan Nasional "Veteran". Yogyakarta.
- [2] Merliana, M. 2005. *Perangkat Lunak Untuk Pembelajaran Penyederhanaan Tata Bahasa bebas Konteks dan Pengubahan ke dalam Bentuk Normal Greibach Dengan Metode Substitusi*. Skripsi, Unikom. Bandung.
- [3] Hamzah, A. 2009. *Teori Bahasa Dan Automata*. Yogyakarta: Aprind Press.
- [4] Jogiyoanto. 2010. *Analisis & Disain*. Yogyakarta: Penerbit Andi.