

# RANCANG BANGUN PERANGKAT LUNAK ALAT BANTU AJAR KALKULUS PROPOSISI MENGGUNAKAN METODE PARSING

Bustami

*Teknik Informatikan, Fakultas Teknik, Universitas Malikussaleh  
Alamat : Jl. Cot Tgk. Nie Reuleut Kec. Muara Batu Kab. Aceh Utaras  
Email : busabiel@gmail.com*

## Abstrak

*Kalkulus proposisi pada hakikatnya adalah suatu metode dalam komputasi yang digunakan untuk menghitung nilai kebenaran dari suatu proposisi. Penelitian ini berkaitan dengan pengembangan perangkat lunak yang dapat digunakan untuk menyelesaikan persoalan kalkulus proposisi. Perangkat lunak yang dibangun memiliki kemampuan untuk mengeksekusi masukan berupa kalimat proposisi dan menghasilkan keluaran berupa proposisi-proposisi atomik serta variabel proposisi yang mengikutinya, ekspresi logika dan tabel kebenaran dari proposisi yang diinputkan.*

*Sistem dibangun dengan menggunakan metode parsing untuk menguraikan proposisi majemuk menjadi proposisi-proposisi atomik serta untuk menguraikan ekspresi logika kedalam bentuk sub-sub ekspresi logika.*

*Hasil perangkat lunak ini diharapkan dapat memudahkan dan membantu siswa meningkatkan pemahamannya mengenai materi kalkulus proposisi.*

**Kata kunci:** *proposisi, operasi-operasi pada proposisi, parsing.*

## 1. Pendahuluan

Seiring dengan perkembangan teknologi komputer dan teknologi informasi, sistem pembelajaran dan pengajaran yang mendayagunakan perangkat komputer sebagai alat bantu dalam proses belajar mengajar disekolah telah menjadi kebutuhan. Salah satu sistem yang dimaksud adalah dalam bentuk aplikasi.

Melatih berfikir dan bernalar secara logis dan kritis serta mengembangkan aktifitas kreatif dalam memecahkan masalah dan mengkomunikasikan ide atau gagasan adalah tujuan umum pembelajaran matematika di sekolah. Konsep logika matematika adalah salah satu kompetensi yang mencerminkannya.

Logika matematika adalah cabang ilmu di bidang matematika yang memperdalam masalah logika, atau lebih tepatnya memperjelas logika dengan kaidah-kaidah matematika. Didalam matematika yang dimaksud dengan logika adalah suatu sistem matematika yang didasarkan pada proposisi. Logika yang menangani atau memproses atau memanipulasi penarikan kesimpulan secara logis dari proposisi-proposisi disebut Logika

Proposisional (*Propositional Logic* atau *Propositional Calculus*)[2].

Berdasarkan pengamatan, biasanya pembelajaran ini dilakukan bagi siswa oleh guru dengan belajar dari sarana pembelajaran yang masih manual yaitu dengan belajar dari buku-buku atau modul-modul lainnya yang terkait, hal ini tentunya menipiskan minat belajar siswa. Masalah yang timbul kemudian adalah cara ini dianggap kurang efektif di zaman pesatnya kemajuan teknologi yang memungkinkan kita belajar dengan cara yang lebih sederhana, interaktif dan praktis serta meminimalisir kesalahan. Selain itu pembelajaran kalkulus proposisi masih menjadi hal yang cukup sulit dipahami, sebagian siswa mengalami kesulitan dalam menyelesaikan soal-soal yang diberikan, apalagi jika proposisi yang harus diselesaikan merupakan proposisi majemuk yang mengandung jumlah variabel dan operator yang banyak, tentunya proses pencarian nilai kebenaran dari proposisi tersebut akan semakin rumit, dan membutuhkan ketelitian yang tinggi [1].

Berdasarkan uraian di atas perlu dibangun sebuah perangkat lunak pembelajaran kalkulus proposisi yang bertujuan untuk menarik minat siswa dalam mempelajari materi kalkulus proposisi serta membantu siswa menyelesaikan dan meningkatkan pemahamannya mengenai kalkulus proposisi dalam melakukan kegiatan belajar secara mandiri dan praktis.

## Logika Proposisional

Logika proposisional merupakan studi dari hubungan logika matematika dengan pernyataan-pernyataan yang dinyatakan dalam hubungan kalimat. Proposisi adalah sebuah pernyataan yang bisa bernilai benar atau salah tetapi tidak sekaligus keduanya. Setiap pernyataan adalah kalimat, tetapi tidak semua kalimat adalah pernyataan. Kalimat yang dapat digolongkan sebagai pernyataan adalah kalimat yang menerangkan sesuatu (kalimat deklaratif) [4].

Setiap perangkat/operasi/operator pada logika memiliki nilai kebenarannya masing-masing sesuai jenis operasi logika yang digunakan. Operasi – operasi pada proposisi berdasarkan urutan prioritasnya adalah sebagai berikut:

1. Operasi Negasi (not)
2. Operasi Konjungsi (And)
3. Operasi Disjungsi (Or)

4. Operasi Implikasi (if...then)
5. Operasi Biimplikasi (if and only if)

### Metode Parsing

Parsing atau proses penurunan adalah proses penentuan struktur sebuah kalimat berdasarkan *grammar* (tata bahasa) dan *lexicon* (kosa kata tertentu). Proses *parsing* merupakan tahapan analisis sintaksis yang berguna untuk memeriksa urutan token. Salah satu proses yang dimaksud adalah proses pengenalan sebuah string masukan yang terdiri dari simbol-simbol apakah string tersebut sesuai dengan tata bahasa yang berlaku[3].

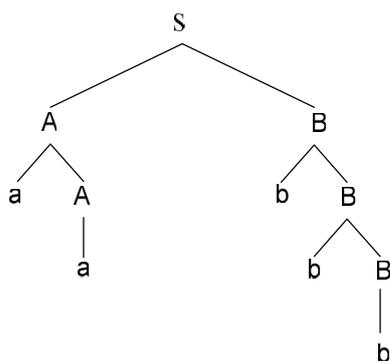
Pohon sintaks/pohon penurunan berguna untuk menggambarkan bagaimana memperoleh suatu string dengan cara menurunkan simbol-simbol variabel menjadi simbol-simbol terminal. Tata bahasa bebas konteks memiliki aturan produksi :

$S \rightarrow AB$  {S menurunkan Variabel A B}

$A \rightarrow aA \mid a$  {A menurunkan terminal a variabel A atau terminal a}

$B \rightarrow bB \mid b$  {B menurunkan terminal b variabel B atau terminal B}

Berikut adalah gambar pohon penurunan (parser) untuk memperoleh string 'aabb'



Gambar 1. Pohon Parser

Sebagaimana dikemukakan Dita Anindhika (2010), Metode Parsing dapat digolongkan sebagai berikut :

a. *Top Down Parsing*

*Top down parser* memulai pemeriksaan dari simbol awal kalimat dan mencoba untuk mencari bentuk simbol terminal berikutnya yang sesuai dengan jenis kata dari kalimat masukan.

b. *Bottom Up Parsing*

*Bottom up parser* mencari dari simbol terminal menuju kearah pembentukan simbol awal sebuah kalimat. Salah satu metodenya adalah metode *grammar presenden* sederhana.

Dalam penelitian ini penulis menggunakan metode *top down parsing*, dimana *top down parsing* bekerja dengan cara menguraikan kalimat proposisi majemuk mulai dari *constituent* (unsur-unsur pembentuk kalimat yang dapat berdiri sendiri) yang terbesar sampai

menjadi *constituent* terkecil. Hal ini dilakukan terus menerus sampai semua komponen yang dihasilkan ialah *constituent* terkecil dalam kalimat.

### 1.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan.

### 1.4 Analisis Kebutuhan Pengguna

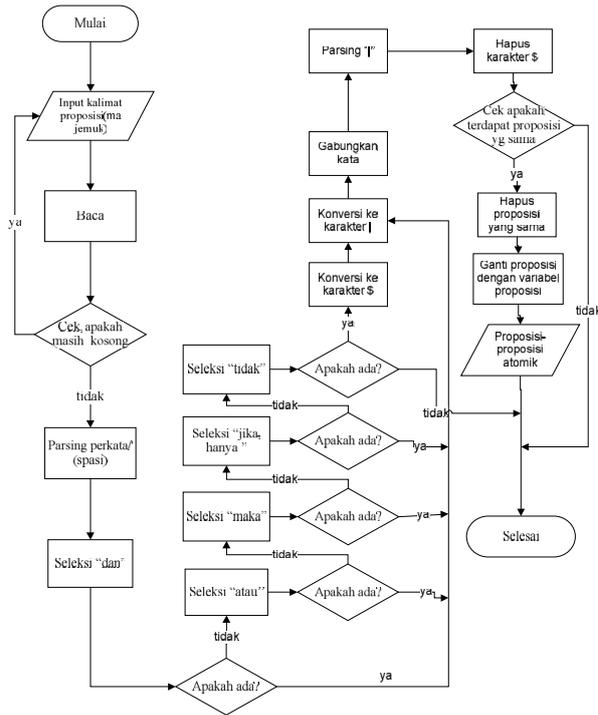
Pengguna aplikasi pembelajaran ini adalah siswa SMA/MA, namun tidak menutup kemungkinan aplikasi ini dapat digunakan oleh siapapun yang ingin menggunakannya sebagai sarana pembelajaran. Untuk itu perangkat lunak yang dirancang menggunakan antar muka berbasis GUI (*Graphic user Interface*) yang dirancang dengan sederhana, menarik dan interaktif agar *user* dapat mengoperasikannya dengan mudah (*user friendly*).

### 1.5 Analisis Kebutuhan Input

Masukan (*input*) dari *user* bagi perangkat lunak ini yaitu kalimat pernyataan/proposisi yang terdapat pada materi kalkulus proposisi dalam logika matematika, yang didalamnya mengandung variabel-variabel proposisi dan perangkat logika/operasi-operasi dalam proposisi. Selain itu sistem juga menerima masukan dalam bentuk ekspresi logika. *User* dapat memilih salah satu bentuk input sesuai dengan yang diinginkan.

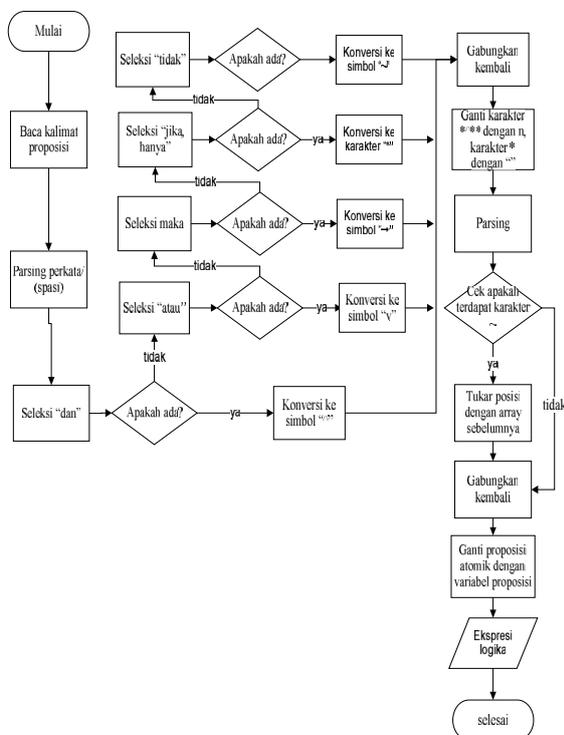
### 1.6 Perancangan Sistem

Perancangan sistem pada penelitian ini dibagi dalam dua tahap yaitu: perancangan *flowchart* (diagram alir) yang akan menunjukkan aliran-aliran data yang terdapat pada aplikasi. Selanjutnya yaitu perancangan form aplikasi (*user interface*).



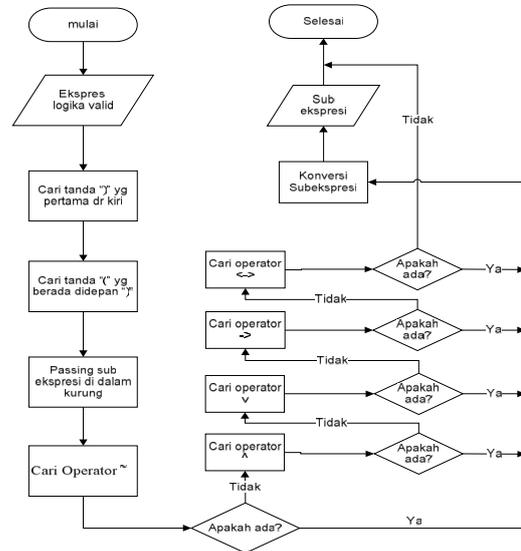
Gambar 2. Flowchart Proses Parsing Kalimat Proposisi

Pada flowchart diatas, dilakukan proses parsing kalimat proposisi, sehingga dapat dideteksi apakah kalimat tersebut merupakan kalimat proposisi atau bukan kalimat proposisi.



Gambar 3. Flowchart Proses Mengubah Kalimat Proposisi Menjadi Ekspresi Logika

Setelah dilakukan proses parsing kalimat proposisi, selanjutnya dilakukan proses mengubah kalimat proposisi menjadi ekspresi logik seperti terlihat pada gambar 3. Proses parsing ekspresi logika dapat dilihat pada gambar 4 dibawah.



Gambar 4. Flowchart Proses Parsing Ekspresi Logika

## 2. Pembahasan

Secara umum proses penyelesaian kalkulus proposisi pada perangkat lunak ini melalui beberapa tahap yaitu, Kalimat proposisi yang diinputkan akan diuraikan terlebih dahulu yaitu, dari proposisi majemuk menjadi proposisi-proposisi atomik. Setelah proposisi-proposisi atomik dihasilkan selanjutnya kalimat proposisi tersebut akan diubah ke dalam bentuk ekspresi logika. Ekspresi logika yang digunakan dalam perangkat lunak ini adalah ekspresi logika dalam bentuk ekspresi aritmatika, dimana operasi/operator diletakkan diantara operand-operand. Operator pada perangkat lunak ini merupakan operator logika yang terdiri dari " $\sim$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ,  $\leftrightarrow$ ". Sedangkan operand pada perangkat lunak ini adalah variabel-variabel proposisi yang mengikuti proposisi-proposisi atomik yang diwakili oleh huruf p, q, r dan s.

Pada saat proses pembacaan, struktur ekspresi yang tidak sesuai dengan ketentuan struktur proposisi akan menyebabkan terjadinya kesalahan. Kesalahan ini akan dilaporkan kepada pengguna, dan pengguna harus memperbaiki kesalahan tersebut, agar sistem dapat melanjutkan proses selanjutnya.

Sistem dibangun dengan menggunakan metode *Leftmost Derevation* untuk menurunkan (*parsing*) ekspresi logika yang dihasilkan. Berdasarkan metode *Leftmost Derevation*, ekspresi logika akan diperiksa dan diturunkan dari sebelah kiri ke sebelah kanan. Ekspresi logika yang dihasilkan harus dipecah menjadi sub-sub ekspresi logika yang paling sederhana (terdiri atas 1 perangkat/operator logika dan 2 operand/variabel

proposisi), supaya perangkat lunak dapat menyelesaikan ekspresi logika tersebut.

Pemecahan ekspresi logika ke bentuk sub-sub ekspresi logika harus mengikuti urutan atau prioritas operasi. Operasi dengan prioritas yang lebih tinggi harus dikerjakan terlebih dahulu. sub ekspresi logika akan disimpan dalam bentuk  $P[n] = \langle \text{operand} \rangle \langle \text{operator} \rangle \langle \text{operand} \rangle$ . Setelah subekspresi – subekspresi dihasilkan maka sistem akan membuat tabel kebenaran dari setiap subekspresi.

### 2.1 Ilustrasi Penyelesaian Menggunakan Metode Parsing

Misalkan diketahui sebuah kalimat proposisi “jika anda mengambil mata kuliah logika matematika dan jika anda tidak memahami tautologi maka anda tidak lulus” maka proses penguraian kalimat proposisi majemuk tersebut akan melalui tahap-tahap berikut, Kalimat proposisi yang diinputkan terlebih dahulu diuraikan/dipisahkan dari perangkainya, sehingga dihasilkanlah proposisi-proposisi atomik sebagai berikut,

1. Anda mengambil mata kuliah logika matematika
2. Anda memahami tautologi
3. Anda lulus

Selanjutnya gantikan proposisi-proposisi tersebut dengan variabel proposisi.

[p] Anda mengambil mata kuliah logika matematika

[q] Anda memahami tautologi

[r] Anda lulus

Perangkai/operasi yang terkandung dalam kalimat proposisi yang diinputkan yaitu dan ( $\wedge$ ), negasi ( $\sim$ ), dan implikasi ( $\rightarrow$ ).

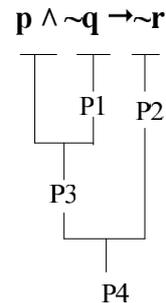
“jika anda mengambil mata kuliah logika matematika  $\wedge \sim$  anda memahami tautologi  $\rightarrow \sim$  anda lulus”

Selanjutnya ganti proposisi atomik dengan variabel-variabel proposisi. Sehingga hasil akhirnya berupa ekspresi logika  $p \wedge \sim q \rightarrow \sim r$ .

Proses selanjutnya yaitu pengecekan terhadap struktur ekspresi logika (operand dan operator). Berikut merupakan ketentuan dari struktur ekspresi logika.

1. Karakter pertama dari proposisi harus berupa karakter proposisi ‘p’, ‘q’, ‘r’, ‘s’, operator ‘ $\sim$ ’, atau delimiter ‘(’.
2. Karakter terakhir dari proposisi harus berupa karakter proposisi ‘p’, ‘q’, ‘r’, ‘s’, atau delimiter ‘)’.
3. Operand ‘p’, ‘q’, ‘r’, dan ‘s’ yang bukan merupakan karakter pertama dari proposisi harus didahului oleh operator ‘ $\sim$ ’, ‘ $\wedge$ ’, ‘ $\vee$ ’, ‘ $\rightarrow$ ’, ‘ $\leftrightarrow$ ’, atau delimiter ‘(’.
4. Operator ‘ $\sim$ ’ yang bukan merupakan karakter pertama dari proposisi harus didahului oleh operator ‘ $\sim$ ’, ‘ $\wedge$ ’, ‘ $\vee$ ’, ‘ $\rightarrow$ ’, ‘ $\leftrightarrow$ ’, atau delimiter ‘(’.
5. Operator ‘ $\wedge$ ’, ‘ $\vee$ ’, ‘ $\rightarrow$ ’, dan ‘ $\leftrightarrow$ ’ harus didahului oleh operand ‘p’, ‘q’, ‘r’, ‘s’, atau delimiter ‘)’.

Setelah proses pengecekan terhadap struktur dari ekspresi logika, proses akan dilanjutkan dengan membagi ekspresi logika ke dalam sub ekspresi logika.



Gambar 5. Pembagian Ekspresi ke Dalam Sub Ekspresi

Berdasarkan metode *leftmost derivation*, maka dari ekspresi logika  $p \wedge \sim q \rightarrow \sim r$ , pembacaan dari sebelah kiri akan menemukan operasi konjungsi ‘ $\wedge$ ’ kemudian akan ditemukan operasi negasi ‘ $\sim$ ’ selanjutnya operasi implikasi ‘ $\rightarrow$ ’. Berdasarkan urutan prioritas dari operasi proposisi, operasi negasi memiliki kedudukan paling tinggi kemudian diikuti oleh operasi konjungsi, disjungsi, implikasi dan biimplikasi. Untuk itu sub ekspresi  $\sim q$  akan diturunkan terlebih dahulu dan hasil penurunan diberi label P1. Kemudian subekspresi  $\sim r$  dan hasil penurunan di beri label P2. Operand p dan hasil penurunan P1 akan dilakukan operasi konjungsi dan diberi label P3. Proses diakhiri dengan operasi implikasi antara hasil penurunan P3 dengan P2 dan diberi label P4.

Proses pembuatan tabel kebenaran dari ekspresi logika  $p \wedge \sim q \rightarrow \sim r$  adalah sebagai berikut,

Tabel 1. Tabel Kebenaran Sub Ekspresi  $p \wedge \sim q \rightarrow \sim r$

p	q	r	[p1] = $\sim q$	[p2] = $\sim r$	p[3]= $p \wedge \sim q$	p[4]= $p \wedge \sim q \rightarrow \sim r$
B	B	B	S	S	S	B
B	B	S	S	B	S	B
B	S	B	B	S	B	S
B	S	S	B	B	B	B
S	B	B	S	S	S	B
S	B	S	S	B	S	B
S	S	B	B	S	S	B
S	S	S	B	B	S	B

### 2.2 Implementasi Sistem

Perangkat lunak ini dibangun dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0* dengan didukung oleh beberapa komponen tambahan seperti *Video Soft Flexgrid 7.0* dan *Rich TextBox*.

#### 2.2.1 Form Menu Utama

Form utama berfungsi untuk menampilkan semua tombol menu yang terdapat didalam aplikasi. Pada form menu utama terdapat empat buah tombol yaitu, tombol Materi, tombol Input Kalimat Proposisi, tombol

Input Ekspresi Logika, dan tombol *Exit*. Pada *form* menu utama, *user* dapat memilih jenis input sesuai dengan yang diinginkan apakah berupa kalimat proposisi ataukah berupa ekspresi logika dengan menekan salah satu tombol yang tersedia. Adapun tampilannya dapat dilihat pada gambar berikut :

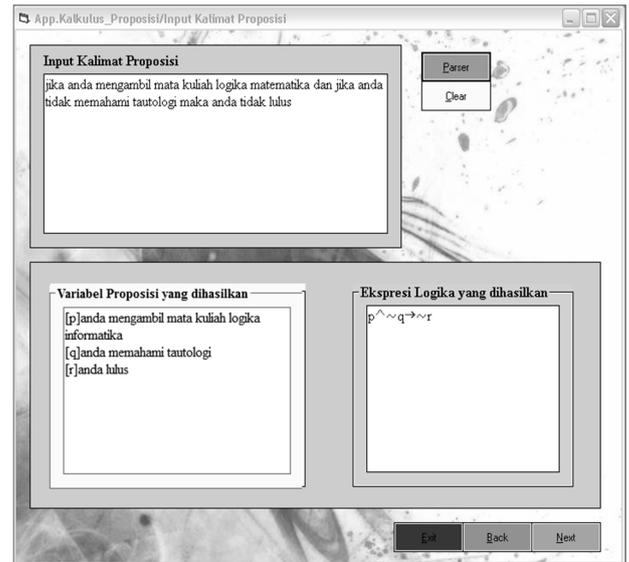


Gambar 6. Form Menu Utama

### 2.2.2 Form Input Kalimat Proposisi

*Form* ini ditampilkan jika *user* memilih menu jenis input kalimat proposisi pada *form* utama. Pada *form* ini terdapat lima buah tombol yaitu, Tombol *clear* berfungsi untuk mengosongkan isi *textbox*, tombol *back* berfungsi untuk kembali ke *form* awal, tombol *next* berfungsi untuk menuju *form* selanjutnya (*form* langkah) selain itu tombol *next* juga diberi perintah untuk mengecek kebenaran dari struktur ekspresi. Tombol *parser* berfungsi untuk memproses kalimat proposisi yang diinputkan.

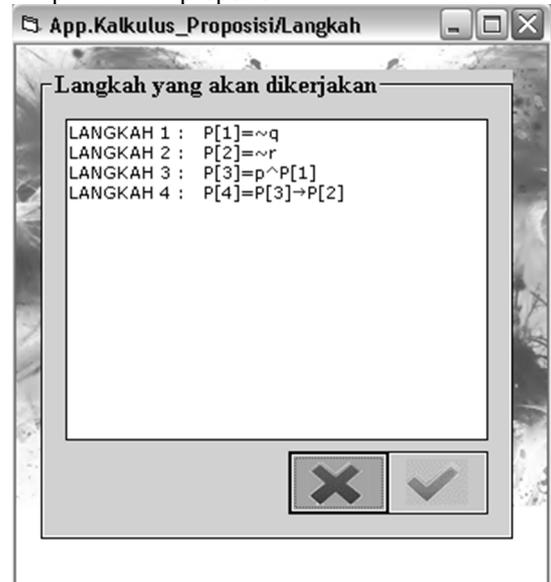
Pada *form* ini *user* diminta untuk memasukkan kalimat proposisi pada *textbox* yang tersedia, kalimat proposisi inilah yang nantinya diproses sehingga output yang dihasilkan sistem berupa proposisi atomik serta variabel proposisi yang mengikutinya dan ekspresi logika. Berikut merupakan tampilan *form* input kalimat proposisi yang sebelumnya telah diinputkan kalimat proposisi sebagai contoh.



Gambar 7. Form Input Kalimat Proposisi

### 2.2.3 Form Langkah

*Form* langkah menampilkan langkah yang akan dikerjakan sistem setelah ekspresi logika diuraikan menjadi subekspresi – subekspresi mulai dari langkah1 P[1] sampai dengan langkah P[n]. Berikut akan ditampilkan langkah pengerjaan dari ekspresi logika yang dihasilkan dari kalimat proposisi yang diinputkan pada *form* input kalimat proposisi.



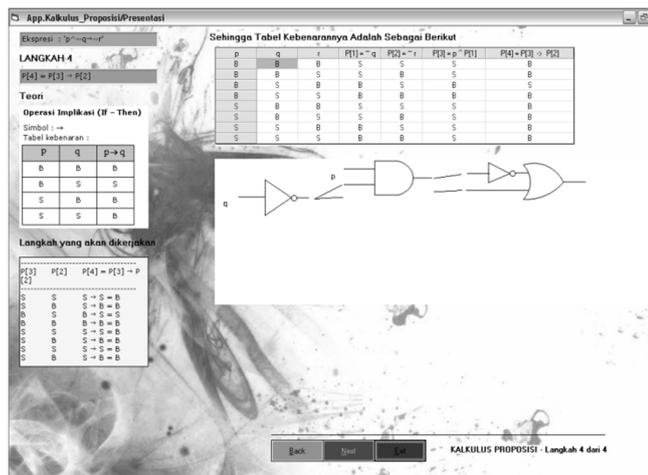
Gambar 8. Form Langkah

Dari tampilan di atas dapat dilihat bahwa terdapat empat buah langkah penyelesaian, yaitu [P1], [P2], [P3] dan [P4]. Terlihat bahwa operasi negasi akan terlebih dahulu dikerjakan baru diikuti oleh operasi konjungsi kemudian biimplikasi. Hal ini sesuai dengan urutan prioritas dari masing-masing operasi proposisi.

### 2.2.4 Form Presentasi

*Form* presentasi akan menampilkan hasil penyelesaian dari setiap langkah yang dihasilkan yaitu

tabel kebenaran dari ekspresi logika yang dihasilkan. selain itu juga dijelaskan mengenai teori dari masing-masing operasi.



Gambar 9. Form Presentasi

Dari hasil pengujian terhadap keseluruhan form yang ada terlihat bahwa sistem telah mampu menyelesaikan kalimat proposisi yang diinputkan dengan baik dan benar.

### 2.2.5 Form Input Ekspresi Logika

Form input ekspresi logika ditampilkan jika user memilih jenis input proposisi berupa ekspresi logika. Pada form ini terdapat 15 buah tombol button yaitu, empat buah tombol operand/variabel yaitu “p, q, r dan s”. Lima buah tombol operator/operasi “~, ^, v, →, ↔”, dua buah delimiter “(, )” dan satu buah tombol back space “←”, selain itu juga terdapat tombol Back untuk kembali ke form awal, tombol Exit untuk keluar dari aplikasi dan tombol Next untuk menuju form selanjutnya.



Gambar 10. Form Input Ekspresi Logika

### 2.3 Pengujian

Setelah memastikan seluruh form yang terdapat pada perangkat lunak ini sudah berjalan dengan baik sesuai rancangan. Selanjutnya dilakukan pengujian terhadap output yang dihasilkan sistem, hal ini bertujuan

untuk mengetahui dan memastikan bahwa output yang dihasilkan sistem sesuai dengan input yang diberikan. Pengujian ini dilakukan dengan cara menginputkan beberapa kalimat proposisi, dimana ekspresi logika yang dihasilkan harus sesuai dengan dengan kalimat proposisi yang diinputkan, demikian pula dengan tabel kebenaran yang dihasilkan sistem.

Dari hasil pengujian yang telah dilakukan dapat diketahui bahwa sistem telah mampu menguraikan kalimat proposisi majemuk menjadi proposisi-proposisi atomik serta variabel yang mengikutinya, selain itu sistem juga mampu mengenali operasi-operasi proposisi yang terkandung didalam kalimat proposisi yang diinputkan, mampu mengubah kalimat proposisi tersebut kedalam bentuk simbol logika/ekspresi logika, dan mampu menghasilkan tabel kebenaran dari ekspresi logika yang dihasilkan.

### 3. Kesimpulan

Setelah membangun perangkat lunak alat bantu ajar kalkulus proposisi ini dan setelah melakukan pengujian, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Perangkat lunak yang dibangun mampu menguraikan kalimat proposisi majemuk menjadi proposisi-proposisi atomik serta mampu mengenali operasi-operasi proposisi yang terkandung didalamnya.
2. Perangkat lunak pembelajaran ini memiliki kemampuan untuk menyelesaikan operasi kalkulus proposisi dengan maksimal mengandung empat buah operand.
3. Diharapkan dapat dikembangkan aplikasi alat abntu ajar kalkulus berbasis android dengan menggunakan ekspersi logika yang lebih banyak.

### Daftar Pustaka

- [1] Anindhika, Dita. 2010. *Penggunaan Pohon dalam Mempresentasikan Struktur Kalimat Bahasa Alami*. Makalah IF2091 Struktur Diskrit - Sem.1 Tahun 2010/2011.
- [2] Munir, Rinaldi. *Matematika diskrit Edisi Ketiga*. Bandung : Informatika. 2007.
- [3] Suciadi, James. 2001. *Studi Analisis Metode – Metode Parsing dan Interpretasi Semantik pada Natural Language Processing*. Jurnal Informatika Vol. 2, No.1, Mei2001.
- [4] Soesianto, F dan Djoni Dwijono. *Logika Matematika Untuk Ilmu Komputer*. Yogyakarta : Andi Yogyakarta. 2006.

### Biodata Penulis

**Bustami**, memperoleh gelar Sarjana Sain (S.Si) pada Jurusan Matematika Fakultas MIPA Universitas Syiah Kuala Banda Aceh, lulus tahun 2001. Memperoleh gelar Magister Sain (M.Si) bidang Statistika Komputasi pada Sekolah Pasca Sarjana Institut Pertanian Bogor (IPB) tahun 2004 dan memperoleh gelar Magister Komputer (M.Kom) pada Program Pasca Sarjana STMIK Eresha Jakarta tahun 2013.