**Progres Report Tugas Besar Tahap I**

# Kelompok:

|  |  |  |
| --- | --- | --- |
| **No** | **NIM** | **Nama** |
| 1 | 1301184251 | Galih Wimba |
| 2 | 1301184291 | Muhammad Daffa Regenta S |
| 3 | 13011842251 | Muhammad Naufal Rizky |
| - | - | - |

# Spesifikasi Program

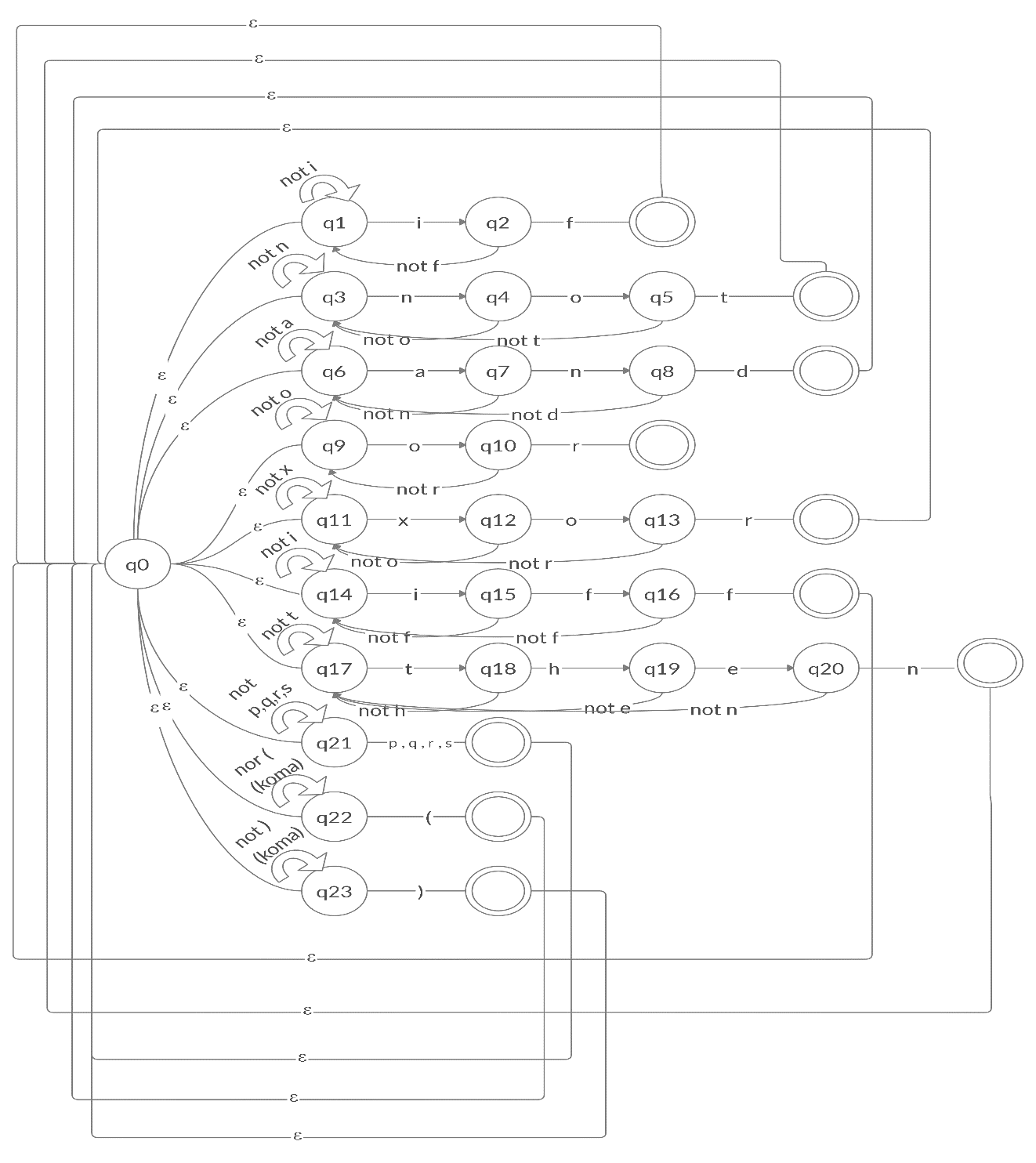
Program Parser Sederhana untuk Formula Logika Proposisi adalah program berbasis java yang menerapkan Finite Automata untuk mengenali dan mengubah logika proposisi string berdasarkan besaran lexic dan token lexicnya. Program ini akan mendefinisikan, menganalisa dan mengenali logika proposisi untuk kemudian menghasilkan Sequence Token Lexic berdasarkan inputan user.

program ini hanya dapat mendeteksi besaran dan token lexic tetapi tidak menentukan apakah logika proposisi tersebut valid apa tidak. Program ini dikembangan menggunakan development tools netbeans java.

* Program hanya dapat mengenali jenis lexic sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| **String Lexic** | **Jenis** | **Token** | **Keterangan** |
| Proposisi:  p, q, r, s | Operand | 1 | Hanya 1 simbol (di antara p, q, r, s) yang dikenal sebagai 1 proposisi |
| not | Operator | 2 | Contoh penulisan yang diterima:  o not proposisi  o not(proposisi)  o not(formula) |
| and | Operator | 3 | Contoh penulisan yang diterima:  o proposisi and proposisi  o (proposisi)and(proposisi)  o (formula)and(formula) |
| or | Operator | 4 | Contoh penulisan yang diterima:  o proposisi or proposisi  o (proposisi)or(proposisi)  o (formula)or(formula) |
| xor | Operator | 5 | Contoh penulisan yang diterima:  o proposisi xor proposisi  o (proposisi)xor(proposisi)  o (formula)xor(formula) |
| if | Operator | 6 | Aturan penulisan yang diterima:  o if proposisi then proposisi  o if(proposisi)then(proposisi)  o if(formula)then(formula) |
| then | Operator | 7 |
| iff | Operator | 8 | Aturan penulisan yang diterima:  o proposisi iff proposisi  o (proposisi)iff(proposisi)  o (formula)iff(formula) |
| ( | Grouping | 9 | Jumlah kurung buka dan tutup harus seimbang  Dituliskan di tempat-tempat yang tepat |
| ) | Grouping | 10 |

# Rancangan Finite Automata



DEFINISI FORMAL

M= ( Q, ∑, δ, q0)

Q = {q0,q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11,q12,q13,q14,q15,q16,q17,q18,q19,q20,q21,q22,q23}

∑ = {pqrs,not,and,xor,or,if,then,iff,(,)}

S = q0

δ = δ digambarkan sebagai table transisi

* Tabel Transisi

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| δ | P,q,r,s | N,o,t | A,n,d | O,r | X,o,r | I,f | T,h,e,n | I,f,f | ( | ) |
| 0 | {1} | {} | {} | {} | {} | {6} | {} | {} | {9} | {10} |
| 1 | 1 | {} | {3} | {4} | {5} | {} | {7} | {} | {} | {10} |
| 2 | 1 | {2} | {} | {} | {} | {} | {} | {} | {} | {} |
| 3 | 1 | {} | {} | {} | {} | {} | {} | {} | {} | {} |
| 4 | 1 | {} | {} | {4} | {} | {} | {} | {} | {} | {} |
| 5 | 1 | {} | {} | {} | {5} | {} | {} | {} | {9} | {} |
| 6 | 1 | {} | {} | {} | {} | {6} | {} | {} | {} | {} |
| 7 | 1 | {} | {} | {} | {} | {} | {7} | {} | {9} | {} |
| 8 | {} | {} | {} | {} | {} | {} | {} | {} | {} | {} |
| 9 | 1 | {2} | {3} | {} | {} | {} | {} | {} | {} | {} |
| 10 | {} | {} | {} | {} | {} | {} | {} | {} | {} | {10} |

# Rancangan Context Free Grammar dan Pushdown Automata

# Context Free Grammar (CFG):

S -> A

A -> 1

A -> 2A

A -> 6A7A

A -> 9A10

A -> ACA

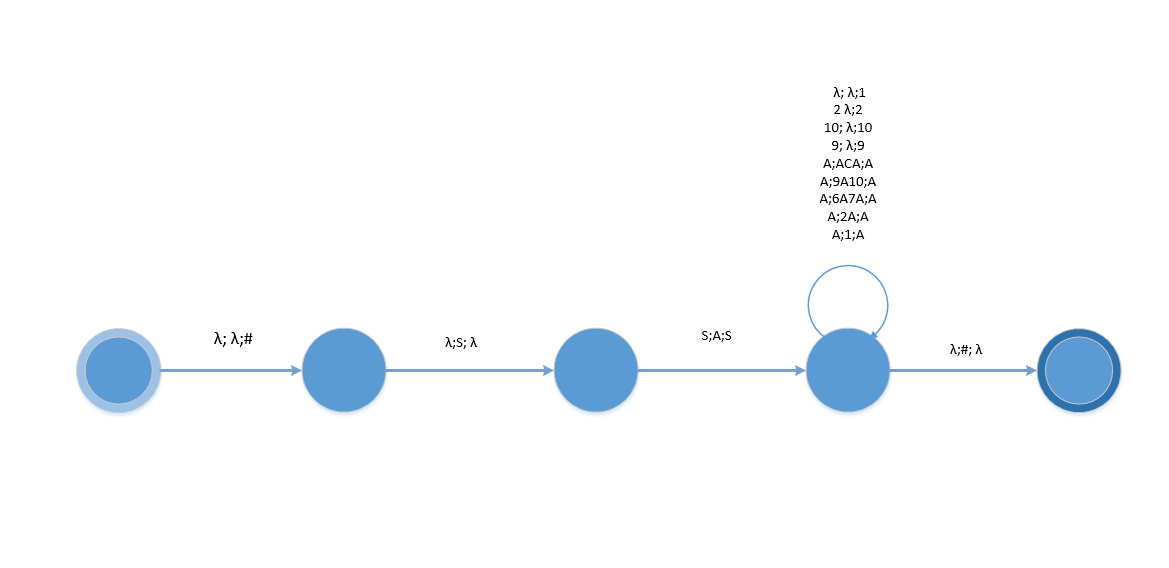
C -> 3

C -> 4

C -> 8

C -> 5

# Pushdown Automata



# Cara Kerja Program

ketika program dijalankan, terdapat sebuah field field inputan pada tampilan awal program yang dapat diisikan dengan Logika proposisi. Inputan tersebut oleh program dibaca sebagai string. Setelah field diisikan dengan inputan string dan menekan tombol analyze, maka program akan mengkonversi string tersebut menjadi sebuah ArrayList dan memasukkanya kedalam sebuah variable dengan tipe data yang sama

# Cara Kerja Program

Masukkan Inputan : p and q or r

Operand TRUE / Operand VALID

[1, rusak]

Kesimpulan : Tidak sesuai masih ada bug pada input selanjutnya

|  |  |  |  |
| --- | --- | --- | --- |
| Input | | Output | Keterangan |
| p and q or r | Operand TRUE / Operand VALID  [1, error] | | Belum sesuai masih ada bug pada input selanjutnya |
| if p then (not q s) | [6, error f] | | Belum sesuai masih ada bug pada input selanjutnya |
| p xor (q and not(p and q)) | Operand TRUE / Operand VALID  [1, error] | | Belum sesuai masih ada bug pada input selanjutnya |
| (p and q ifg(r or s) | [error] | | Error , tidak sesuai |
| not iff p | [error, error] | | Error , tidak sesuai |
| not not not q xor not p | [error, error] | | Error , tidak sesuai |
| if p then r | [6, error f] | | Belum sesuai masih ada bug pada input selanjutnya |
| if (if r then s) and p then q | [6, error f] | | Belum sesuai masih ada bug pada input selanjutnya |
| p and q iff q | [1, error] | | Belum sesuai masih ada bug pada input selanjutnya |
| if (if p then r) then r | [6, error f] | | Belum sesuai masih ada bug pada input selanjutnya |
| (p and q) and q | [error] | | Error , tidak Sesuai |
| (((p and q) or r) and s) | [error] | | Error , tidak Sesuai |

Screen Shoot Program

