Tugas Besar Teori Bahasa dan Automata

Membuat Lexical Analyzer dan Parser Sederhana untuk Pengenalan Kode Pemrograman

Disusun Oleh :

Dhafa Nur Fadhilah (1301213263)

Fannisa Eimin Aurora (1301213150)

Muh. Ghazali (1301213378)

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2023

1. **Buatlah Context Free Grammar yang merepresentasikan aturan/sintaks kondisi IF dalam bahasa pemrograman GO**

Kode program:

if a==b {c=a+b} else {c=a}

Grammar:

<statement> ::= if <kondisi> <aksi_1> else <aksi_2>

<kondisi> ::= <variable> <operator> <variable>

<aksi_1> ::= {<variable> = <variable> + <variable>} |

{<variabel> = <variabel> - <variabel>} |

{<variabel> = <variabel> * <variabel>} |

{<variabel> = <variabel> / <variabel>}

<aksi_2> ::= {<variabel> = <variabel>}

<variabel> ::= a | b | c

<operator> ::= < | > | ==

Simbol non-terminal: <statement> (starting simbol), <kondisi>, <aksi_1>, <aksi_2>, <variabel>, <operator>

Simbol terminal: if, else, a, b, c, =, +, -, *, /, {, }, ==, <, >

Keterangan:

A memiliki isian dari <aksi_1>

B memiliki isian dari <aksi_2>

***Pada kasus ini, nilai awal a=8, b=2, c=‘ ’.*

	if	else	a	b	c	=	+	-	*	/	{	}	==	>	<	EOS
<statement>	if <kondisi> <aksi_1> else <aksi_2>	error	error	error	error	error	error	error	error	error	error	error	error	error	error	error
<kondisi>	error	error	<variable> <operator> <variable>	<variable> <operator> <variable>	error	error	error	error	error	error	error	error	error	error	error	error
<aksi_1>	error	error	A	A	A	A	A	A	A	A	A	A	error	error	error	error
<aksi_2>	error	error	B	B	B	B	error	error	error	error	B	B	error	error	error	error
<variabel>	error	error	a	b	c	error	error	error	error	error	error	error	error	error	error	error
<operator>	error	error	error	error	error	error	error	error	error	error	error	error	==	>	<	error

Kode program:

if a==b {c=a+b} else {d=a}

Grammar:

<statement> ::= if <kondisi> <aksi_1> else <aksi_2>

<kondisi> ::= a>a | a>b | a<a | a<b | a==a | a==b

<aksi_1> ::= {c=a+b} | {c=a-b} | {c=a*b} | {c=a/b}

<aksi_2> ::= {c=a} | {c=b}

Simbol non-terminal: <statement> (starting simbol), <kondisi>, <aksi_1>, <aksi_2>,

Simbol terminal: if, else, a>a, a>b, a<a, a<b, a==a, a==b, {c=a+b}, {c=a-b}, {c=a*b}, {c=a/b}, {c=a}, {c=b}

**Pada kasus ini, nilai awal a=8, b=2, c=0, d=0.*

Rancangan CFG:

S → if C A1 else A2

C → a>a | a>b | a<a | a<b | a==a | a==b

A1 → {c=a+b} | {c=a-b} | {c=a*b} | {c=a/b}

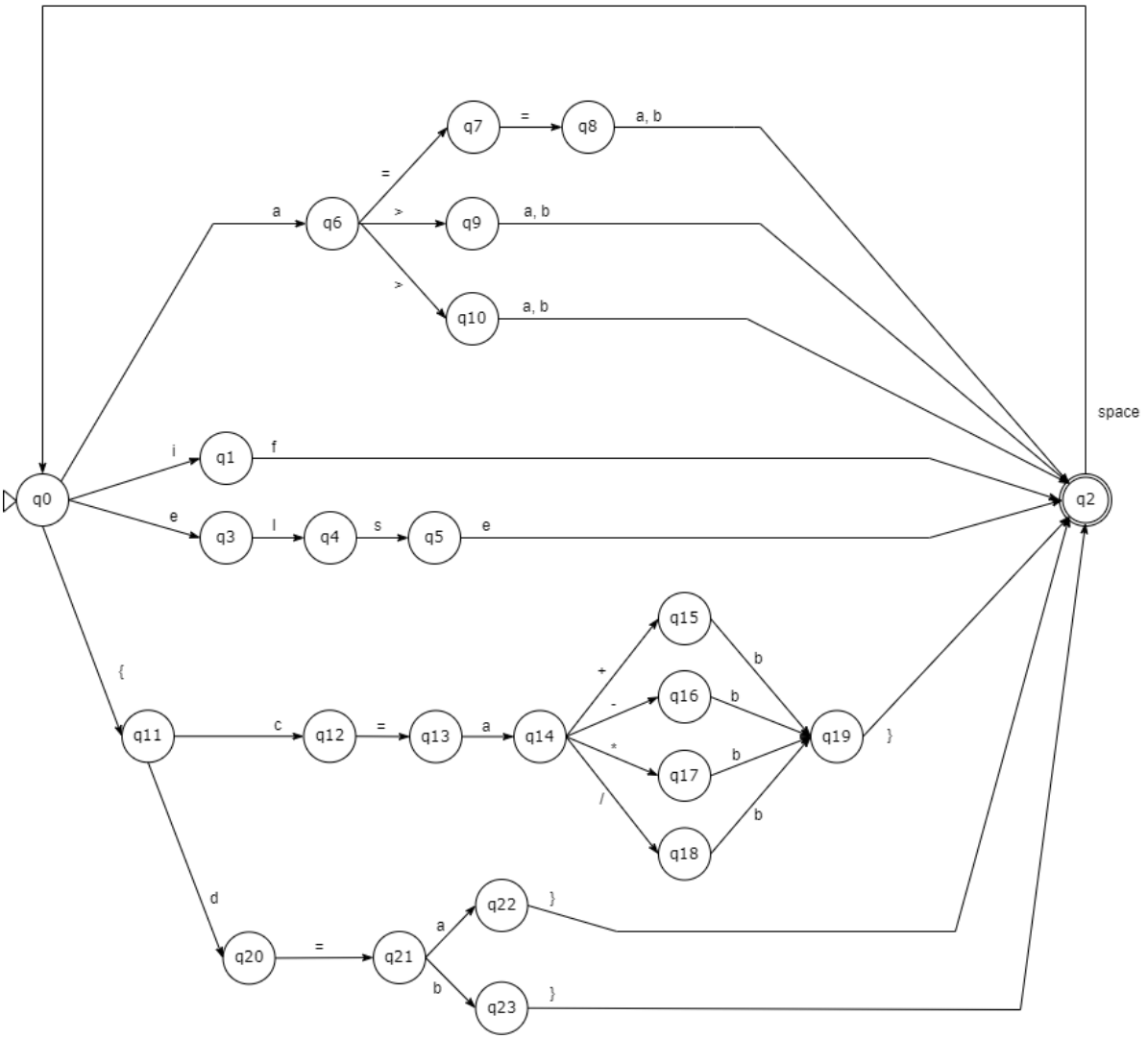
A2 → {c=a} | {c=b}

Starting simbol: S

Simbol non-terminal: S, C, A1, A2

Simbol terminal: if, else, a>a, a>b, a<a, a<b, a==a, a==b,{c=a+b}, {c=a-b}, {c=a*b}, {c=a/b}, {c=a}, {c=b}

2. **Buatlah rancangan Finite Automata untuk lexical analyzer kata/token/simbol terminal sesuai dengan yang didefinisikan pada Grammar.**



3. **Buatlah rancangan parse-table LL (1) sesuai Grammar yang sudah didefinisikan.**

	if	else	a > a	a > b	a < a	a < b	a == a	a == b	{c=a+ b}	{c=a- b}	{c=a* b}	{c=a/ b}	{c=a}	{c=b}	EOS
<statement>	if <kondisi> <aksi_1> else <aksi_2>	error	error	error	error	error	error	error	error	error	error	error	error	error	error
<kondisi>	error	error	a > a	a > b	a < a	a < b	a == a	a == b	error	error	error	error	error	error	error
<aksi_1>	error	error	error	error	error	error	error	error	{c=a+ b}	{c=a- b}	{c=a* b}	{c=a/ b}	error	error	error
<aksi_2>	error	error	error	error	error	error	error	error	error	error	error	error	{c=a}	{c=b}	error

4. **Buatlah program lexical analyzer dan parser sederhana sesuai rancangan Finite Automata dan parse table yang sudah dibuat.**

```
# kondisi IF ELSE Golang
# Kelompok 12:
# > Dhafa Nur Fadhilah (1301213263)
# > Fannisa Eimin Aurora (1301213150)
# > Muh. Ghazali (1301213378)

import string

print("\n====== Lexical Analyzer dan Parser Sederhana Kondisi IF ElSE (GO) =====\n")
print("[Kelompok 12 IF-45-08]")
print("> Dhafa Nur Fadhilah   - 1301213263")
print("> Fannisa Eimin Aurora - 1301213150")
print("> Muh. Ghazali         - 1301213378\n")
print("\nLexical Analyzer\n")

print('> "Nilai variabel saat ini: a=10, b=5, c=0, d=0" \n')

# -- contoh input --
print("Tuliskan kondisi IF ELSE dengan syntax pada GO: \n",
"if a ? $ { c = a ^ b } else { d = @ }  \n",
"1.? - operator pembanding: == | > | < \n",
"2.$ - variabel: a | b \n",
"3.^ - operator aritmatika: + | - | * | /\n",
"4.@ - variabel: a | b")
print("Contoh: " "\033[1m" + "if a==a {c=a+b} else {d=a}" + "\033[0m")
print("--------------------------------")
sentence = input()
print("--------------------------------")
print()
input_string = sentence+'#'

# -- inisialisasi --
alphabet_list = list(string.ascii_letters) + list(string.digits) + ['=', '>', '<', '+', '-', '*', '/', '(', ')', '{', '}', ';', '#']
state_list = ['q0','q1','q2','q3','q4','q5','q6','q7','q8',
              'q9','q10','q11','q12','q13','q14','q15','q16',
              'q17','q18','q19','q20','q21','q22','q23']

transition_table = {}

for state in state_list:
    for alphabet in alphabet_list:
        transition_table[(state, alphabet)] = 'error'
    transition_table[(state, '#')] = 'error'
    transition_table[(state, ' ')] = 'error'

# Tabel transisi untuk accepted state
transition_table[('q2', '#')] = 'accept'

# Tabel transisi untuk: space
transition_table[('q2', ' ')] = 'q0'

# Tabel transisi untuk: if
transition_table[('q0', 'i')] = 'q1'
transition_table[('q1', 'f')] = 'q2'

# Tabel transisi untuk: else
transition_table[('q0', 'e')] = 'q3'
transition_table[('q3', 'l')] = 'q4'
transition_table[('q4', 's')] = 'q5'
transition_table[('q5', 'e')] = 'q2'

# Tabel transisi untuk: a==a & a==b
transition_table[('q0', 'a')] = 'q6'
transition_table[('q6', '=')] = 'q7'
transition_table[('q7', '=')] = 'q8'
transition_table[('q8', 'a')] = 'q2'
transition_table[('q8', 'b')] = 'q2'
```

```
# Tabel transisi untuk: a>b & a>b
transition_table[('q0', 'a')] = 'q6'
transition_table[('q6', '>')] = 'q9'
transition_table[('q9', 'a')] = 'q2'
transition_table[('q9', 'b')] = 'q2'

# Tabel transisi untuk: a<b
transition_table[('q0', 'a')] = 'q6'
transition_table[('q6', '<')] = 'q10'
transition_table[('q10', 'a')] = 'q2'
transition_table[('q10', 'b')] = 'q2'

# Tabel transisi untuk: {d=
transition_table[('q0', '{')] = 'q11'
transition_table[('q11', 'd')] = 'q20'
transition_table[('q20', '=')] = 'q21'

# Tabel transisi untuk: a}
transition_table[('q21', 'a')] = 'q22'
transition_table[('q22', '}')] = 'q2'

# Tabel transisi untuk: b}
transition_table[('q21', 'b')] = 'q23'
transition_table[('q23', '}')] = 'q2'

# Tabel transisi untuk: {c=a
transition_table[('q0', '{')] = 'q11'
transition_table[('q11', 'c')] = 'q12'
transition_table[('q12', '=')] = 'q13'
transition_table[('q13', 'a')] = 'q14'

# Tabel transisi untuk: +b
transition_table[('q14', '+')] = 'q15'
transition_table[('q15', 'b')] = 'q19'

# Tabel transisi untuk: -b
transition_table[('q14', '-')] = 'q16'
transition_table[('q16', 'b')] = 'q19'

# Tabel transisi untuk: *b
transition_table[('q14', '*')] = 'q17'
transition_table[('q17', 'b')] = 'q19'

# Tabel transisi untuk: /b
transition_table[('q14', '/')] = 'q18'
transition_table[('q18', 'b')] = 'q19'

# Tabel transisi untuk: }
transition_table[('q19', '}')] = 'q2'

# -- lexical analysis --
idx_char = 0
state = 'q0'
current_token = ''
while state !='accept':
    current_char = input_string[idx_char]
    current_token += current_char
    state = transition_table[(state, current_char)]
    if state == 'q2':
        print(current_token, ", valid")
        current_token = ''
    if state == 'error':
        print(current_token, ", tidak valid")
        break;
    idx_char = idx_char + 1

# -- output --
lexical_correct = False
if state == 'accept':
    print('Syntax kondisi IF ELSE berikut: ', sentence, ', valid')
    lexical_correct = True
else:
    print('Ada kesalahan syntax atau pernyataan pada konsisi IF ELSE: ', sentence)
    print('Tidak lanjut ke tahap Parser, karena ada kesalahan')

# Tahap Parser
if lexical_correct:
    print()
    print("== Tahap Parser ==")
    print("--------------------------")
    print(sentence)
    tokens = sentence.split()
    tokens.append('EOS')
    print("--------------------------")
    print()
```

```
# Definisi Simbol
non_terminals = ['S', 'C', 'A1', 'A2']
terminals = ["if","else","a>a","a>b","a<a","a<b","a==a","a==b","{c=a+b}","{c=a-b}","{c=a*b}","{c=a/b}","{d=a}","{d=b}"]

# Definisi Tabel Parser
parse_table = {}

parse_table[("S", "if")] = ["if", "C", "A1", "else", "A2"]
parse_table[("S", "else")] = ["error"]
parse_table[("S", "a>a")] = ["error"]
parse_table[("S", "a>b")] = ["error"]
parse_table[("S", "a==a")] = ["error"]
parse_table[("S", "a==b")] = ["error"]
parse_table[("S", "a<a")] = ["error"]
parse_table[("S", "a<b")] = ["error"]
parse_table[("S", "{c=a+b}")] = ["error"]
parse_table[("S", "{c=a-b}")] = ["error"]
parse_table[("S", "{c=a*b}")] = ["error"]
parse_table[("S", "{c=a/b}")] = ["error"]
parse_table[("S", "{d=a}")] = ["error"]
parse_table[("S", "{d=b}")] = ["error"]
parse_table[("S", "EOS")] = ["error"]

parse_table[("C", "if")] = ["error"]
parse_table[("C", "else")] = ["error"]
parse_table[("C", "a>a")] = ["a>a"]
parse_table[("C", "a>b")] = ["a>b"]
parse_table[("C", "a==a")] = ["a==a"]
parse_table[("C", "a==b")] = ["a==b"]
parse_table[("C", "a<a")] = ["a<a"]
parse_table[("C", "a<b")] = ["a<b"]
parse_table[("C", "{c=a+b}")] = ["error"]
parse_table[("C", "{c=a-b}")] = ["error"]
parse_table[("C", "{c=a*b}")] = ["error"]
parse_table[("C", "{c=a/b}")] = ["error"]
parse_table[("C", "{d=a}")] = ["error"]
parse_table[("C", "{d=b}")] = ["error"]
parse_table[("C", "EOS")] = ["error"]

parse_table[("A1", "if")] = ["error"]
parse_table[("A1", "else")] = ["error"]
parse_table[("A1", "else")] = ["error"]
parse_table[("A1", "a>a")] = ["error"]
parse_table[("A1", "a>b")] = ["error"]
parse_table[("A1", "a==a")] = ["error"]
parse_table[("A1", "a==b")] = ["error"]
parse_table[("A1", "a<a")] = ["error"]
parse_table[("A1", "a<b")] = ["error"]
parse_table[("A1", "{c=a+b}")] = ["{c=a+b}"]
parse_table[("A1", "{c=a-b}")] = ["{c=a-b}"]
parse_table[("A1", "{c=a*b}")] = ["{c=a*b}"]
parse_table[("A1", "{c=a/b}")] = ["{c=a/b}"]
parse_table[("A1", "{d=a}")] = ["error"]
parse_table[("A1", "{d=b}")] = ["error"]
parse_table[("A1", "EOS")] = ["error"]

parse_table[("A2", "if")] = ["error"]
parse_table[("A2", "else")] = ["error"]
parse_table[("A2", "a>a")] = ["error"]
parse_table[("A2", "a>b")] = ["error"]
parse_table[("A2", "a==a")] = ["error"]
parse_table[("A2", "a==b")] = ["error"]
parse_table[("A2", "a<a")] = ["error"]
parse_table[("A2", "a<b")] = ["error"]
parse_table[("A2", "{c=a+b}")] = ["error"]
parse_table[("A2", "{c=a-b}")] = ["error"]
parse_table[("A2", "{c=a*b}")] = ["error"]
parse_table[("A2", "{c=a/b}")] = ["error"]
parse_table[("A2", "{d=a}")] = ["{d=a}"]
parse_table[("A2", "{d=b}")] = ["{d=b}"]
parse_table[("A2", "EOS")] = ["error"]
```

```
    # Inisialisasi Stack
    stack = []
    stack.append("#")
    stack.append("S")

    # Inisialisasi Input
    idx_token = 0
    symbol = tokens[idx_token]

    # Proses Parsing
    while (len(stack) > 0):
        top = stack[len(stack)-1]
        print("top = ",top)
        print("bilangan = ",symbol)
        if top in terminals:
            print("top stack adalah simbol terminal")
            if top == symbol:
                stack.pop()
                idx_token = idx_token + 1
                symbol = tokens[idx_token]
                if symbol == "EOS":
                    print("isi stack", stack)
                    stack.pop()
            else:
                print("Error")
                break;
        elif top in non_terminals:
            print("top stack adalah simbol non-terminal")
            if parse_table[(top, symbol)][0] != "error":
                stack.pop()
                symbols_to_be_pushed  = parse_table[(top, symbol)]
                for i in range(len(symbols_to_be_pushed)-1,-1,-1):
                    stack.append(symbols_to_be_pushed[i])
            else:
                if top == 'S':
                    print()
                    print("================ HASIL ==================")
                    print("Error ", symbol, " ini bukan C")
                    print(sentence,", bukan penulisan kondisi IF ELSE dalam GO")
                    print("=========================================")
                    break;
                else:
                    print()
                    print("================= HASIL ==================")
                    print("Error", symbol, " ini bukan ", top)
                    print(sentence,", bukan penulisan kondisi IF ELSE dalam GO")
                    print("=========================================")
                    break;
        else:
            print("error")
            break;
        print("Elemen stack: ", stack)
        print()

        # Conclusion
        #print()
        if symbol == "EOS" and len(stack) == 0:
            print("============================== HASIL ==============================")
            print("Input:", sentence, ", merupakan penulisan kondisi IF ELSE dalam GO yang benar.")
            print("===================================================================")
            print()

# Perhitungan hasil input
# masuk kondisi IF
if sentence=="if a==a {c=a+b} else {d=a}" or sentence=="if a>b {c=a+b} else {d=a}" or sentence=="if a==a {c=a+b} else {d=b}" or
sentence=="if a>b {c=a+b} else {d=b}":
  print(">> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=10, d=0")
elif sentence=="if a==a {c=a-b} else {d=a}" or sentence=="if a>b {c=a-b} else {d=a}" or sentence=="if a==a {c=a-b} else {d=b}" or
sentence=="if a>b {c=a-b} else {d=b}":
  print(">> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=6, d=0")
elif sentence=="if a==a {c=a*b} else {d=a}" or sentence=="if a>b {c=a*b} else {d=a}" or sentence=="if a==a {c=a*b} else {d=b}" or
sentence=="if a>b {c=a*b} else {d=b}":
  print(">> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=16, d=0")
elif sentence=="if a==a {c=a/b} else {d=a}" or sentence=="if a>b {c=a/b} else {d=a}" or sentence=="if a==a {c=a/b} else {d=b}" or
sentence=="if a>b {c=a/b} else {d=b}":
  print(">> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=16, d=0")
# masuk kondisi ELSE
if sentence=="if a==b {c=a+b} else {d=a}" or sentence=="if a>a {c=a+b} else {d=a}" or sentence=="if a<a {c=a+b} else {d=a}" or
sentence=="if a<b {c=a+b} else {d=a}" or sentence=="if a==b {c=a-b} else {d=a}" or sentence=="if a>a {c=a-b} else {d=a}" or
sentence=="if a<a {c=a-b} else {d=a}" or sentence=="if a<a {c=a-b} else {d=a}" or sentence=="if a==b {c=a*b} else {d=a}" or
sentence=="if a>a {c=a*b} else {d=a}" or sentence=="if a<a {c=a*b} else {d=a}" or sentence=="if a<a {c=a*b} else {d=a}" or sentence=="if
a==b {c=a/b} else {d=a}" or sentence=="if a>a {c=a/b} else {d=a}" or sentence=="if a<a {c=a/b} else {d=a}" or sentence=="if a<a {c=a/b}
else {d=a}":
  print(">> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=0, d=8")
if sentence=="if a==b {c=a+b} else {d=b}" or sentence=="if a>a {c=a+b} else {d=b}" or sentence=="if a<a {c=a+b} else {d=b}" or
sentence=="if a<b {c=a+b} else {d=b}" or sentence=="if a==b {c=a-b} else {d=b}" or sentence=="if a>a {c=a-b} else {d=b}" or
sentence=="if a<a {c=a-b} else {d=b}" or sentence=="if a<a {c=a-b} else {d=b}" or sentence=="if a==b {c=a*b} else {d=b}" or
sentence=="if a>a {c=a*b} else {d=b}" or sentence=="if a<a {c=a*b} else {d=b}" or sentence=="if a<a {c=a*b} else {d=b}" or sentence=="if
a==b {c=a/b} else {d=b}" or sentence=="if a>a {c=a/b} else {d=b}" or sentence=="if a<a {c=a/b} else {d=b}" or sentence=="if a<a {c=a/b}
else {d=b}":
  print(">> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=0, d=2")
```

5. **Lakukan pengujian pada program lexical analyzer dan parser yang sudah dibuat.**

- Kasus bila syntax kalimat benar dan masuk kondisi IF

```
====== Lexical Analyzer dan Parser Sederhana Kondisi IF ElSE (GO) =====

[Kelompok 12 IF-45-08]
> Dhafa Nur Fadhilah   - 1301213263
> Fannisa Eimin Aurora - 1301213150
> Muh. Ghazali         - 1301213378


Lexical Analyzer

> "Nilai variabel saat ini: a=10, b=5, c=0, d=0"

Tuliskan kondisi IF ELSE dengan syntax pada GO:
 if a ? $ { c = a ^ b } else { d = @ }
 1.? - operator pembanding: == | > | <
 2.$ - variabel: a | b
 3.^ - operator aritmatika: + | - | * | /
 4.@ - variabel: a | b
Contoh: if a==a {c=a+b} else {d=a}
--------------------------------
if a==a {c=a+b} else {d=a}
--------------------------------

if , valid
 a==a , valid
 {c=a+b} , valid
 else , valid
 {d=a} , valid
Syntax kondisi IF ELSE berikut:  if a==a {c=a+b} else {d=a} , valid
```

```
== Tahap Parser ==
-------------------------
if a==a {c=a+b} else {d=a}
-------------------------

top =  S
bilangan =  if
top stack adalah simbol non-terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'C', 'if']

top =  if
bilangan =  if
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'C']

top =  C
bilangan =  a==a
top stack adalah simbol non-terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'a==a']

top =  a==a
bilangan =  a==a
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2', 'else', 'A1']

top =  A1
bilangan =  {c=a+b}
top stack adalah simbol non-terminal
Elemen stack:  ['#', 'A2', 'else', '{c=a+b}']

top =  {c=a+b}
bilangan =  {c=a+b}
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2', 'else']

top =  else
bilangan =  else
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2']

top =  A2
bilangan =  {d=a}
top stack adalah simbol non-terminal
Elemen stack:  ['#', '{d=a}']

top =  {d=a}
bilangan =  {d=a}
top stack adalah simbol terminal
isi stack ['#']
Elemen stack:  []

============================== HASIL ==============================
Input: if a==a {c=a+b} else {d=a} , merupakan penulisan kondisi IF ELSE dalam GO yang benar.
===================================================================

>> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=10, d=0
```

- Kasus bila syntax kalimat benar dan masuk kondisi ELSE

```
====== Lexical Analyzer dan Parser Sederhana Kondisi IF ElSE (GO) =====

[Kelompok 12 IF-45-08]
> Dhafa Nur Fadhilah   - 1301213263
> Fannisa Eimin Aurora - 1301213150
> Muh. Ghazali         - 1301213378


Lexical Analyzer

> "Nilai variabel saat ini: a=10, b=5, c=0, d=0"

Tuliskan kondisi IF ELSE dengan syntax pada GO:
 if a ? $ { c = a ^ b } else { d = @ }
 1.? - operator pembanding: == | > | <
 2.$ - variabel: a | b
 3.^ - operator aritmatika: + | - | * | /
 4.@ - variabel: a | b
Contoh: if a==a {c=a+b} else {d=a}
------------------------------
if a<b {c=a+b} else {d=a}
------------------------------

if , valid
 a<b , valid
 {c=a+b} , valid
 else , valid
 {d=a} , valid
Syntax kondisi IF ELSE berikut:  if a<b {c=a+b} else {d=a} , valid

== Tahap Parser ==
--------------------------
if a<b {c=a+b} else {d=a}
--------------------------

top =  S
bilangan =  if
top stack adalah simbol non-terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'C', 'if']

top =  if
bilangan =  if
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'C']

top =  C
bilangan =  a<b
top stack adalah simbol non-terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'a<b']

top =  a<b
bilangan =  a<b
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2', 'else', 'A1']
```

```
top =  A1
bilangan =  {c=a+b}
top stack adalah simbol non-terminal
Elemen stack:  ['#', 'A2', 'else', '{c=a+b}']

top =  {c=a+b}
bilangan =  {c=a+b}
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2', 'else']

top =  else
bilangan =  else
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2']

top =  A2
bilangan =  {d=a}
top stack adalah simbol non-terminal
Elemen stack:  ['#', '{d=a}']

top =  {d=a}
bilangan =  {d=a}
top stack adalah simbol terminal
isi stack ['#']
Elemen stack:  []

============================== HASIL ==============================
Input: if a<b {c=a+b} else {d=a} , merupakan penulisan kondisi IF ELSE dalam GO yang benar.
===================================================================

>> Nilai variabel setelah kondisi IF ELSE: a=8, b=2, c=0, d=8
```

- Kasus bila syntax kalimat benar tapi urutan salah

```
====== Lexical Analyzer dan Parser Sederhana Kondisi IF ELSE (GO) =====

[Kelompok 12 IF-45-08]
> Dhafa Nur Fadhilah   - 1301213263
> Fannisa Eimin Aurora - 1301213150
> Muh. Ghazali         - 1301213378


Lexical Analyzer

> "Nilai variabel saat ini: a=10, b=5, c=0, d=0"

Tuliskan kondisi IF ELSE dengan syntax pada GO:
 if a ? $ { c = a ^ b } else { d = @ }
 1.? - operator pembanding: == | > | <
 2.$ - variabel: a | b
 3.^ - operator aritmatika: + | - | * | /
 4.@ - variabel: a | b
Contoh: if a==a {c=a+b} else {d=a}
--------------------------------
if else {d=a} a==b {c=a-b}
--------------------------------

if , valid
 else , valid
 {d=a} , valid
 a==b , valid
 {c=a-b} , valid
Syntax kondisi IF ELSE berikut:  if else {d=a} a==b {c=a-b} , valid

== Tahap Parser ==
--------------------------
if else {d=a} a==b {c=a-b}
--------------------------

top =  S
bilangan =  if
top stack adalah simbol non-terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'C', 'if']

top =  if
bilangan =  if
top stack adalah simbol terminal
Elemen stack:  ['#', 'A2', 'else', 'A1', 'C']

top =  C
bilangan =  else
top stack adalah simbol non-terminal

================= HASIL ====================
Error else  ini bukan  C
if else {d=a} a==b {c=a-b} , bukan penulisan kondisi IF ELSE dalam GO
============================================
```

- Kasus bila syntax kalimat salah

```
====== Lexical Analyzer dan Parser Sederhana Kondisi IF ElSE (GO) =====

[Kelompok 12 IF-45-08]
> Dhafa Nur Fadhilah   - 1301213263
> Fannisa Eimin Aurora - 1301213150
> Muh. Ghazali         - 1301213378


Lexical Analyzer

> "Nilai variabel saat ini: a=10, b=5, c=0, d=0"

Tuliskan kondisi IF ELSE dengan syntax pada GO:
 if a ? $ { c = a ^ b } else { d = @ }
 1.? - operator pembanding: == | > | <
 2.$ - variabel: a | b
 3.^ - operator aritmatika: + | - | * | /
 4.@ - variabel: a | b
Contoh: if a==a {c=a+b} else {d=a}
--------------------------------
if (a>b) else {d=a}
--------------------------------

if , valid
 ( , tidak valid
Ada kesalahan syntax atau pernyataan pada konsisi IF ELSE:  if (a>b) else {d=a}
Tidak lanjut ke tahap Parser, karena ada kesalahan
```