**Modul 1**

1. A. Algoitma Bahasa Indonesia.

Input : a,b

State : q0, q1, q2

State Awal : q0

State Akhir : q1

* q0 :

masukan : a

kembali ke : q0

masukan : b

final : q1

* q1 :

masukan : b

ke : q2

* q2 :

masukan : b

final : q1

1. Flowchart

SELESAI

error

a

mulai

a, b

q0

b

q1

b

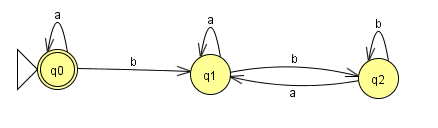
q2

b

1. Kesimpulan :

* Pesan akan diterima jika pesan tersebut sampai di state akhir ( q1 ), contohnya : a, b
* Pesan akan ditolak, jika pesan tersebut tidak sampai di state akhir ( q1 ), contohnya : a, b, b

1. Gambarkan DFA

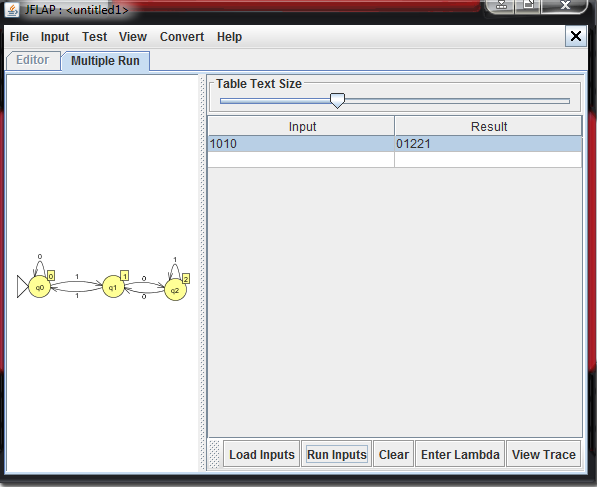


**Modul 2**

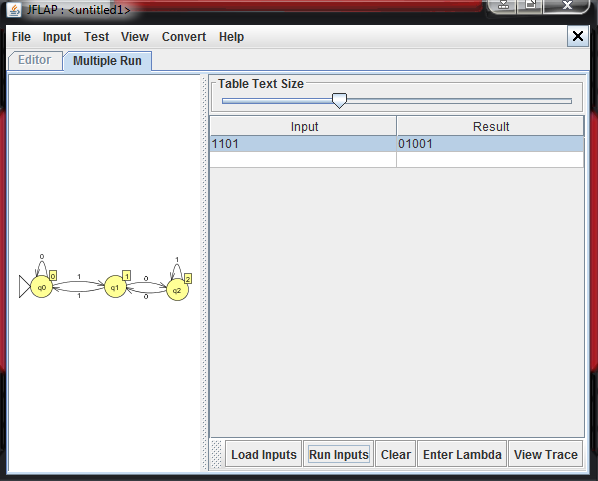
1. Kontruksikan pada mesin more untuk :
2. 10 mod 3
3. 13 mod 3
4. 17 mod 3

**Jawaban**

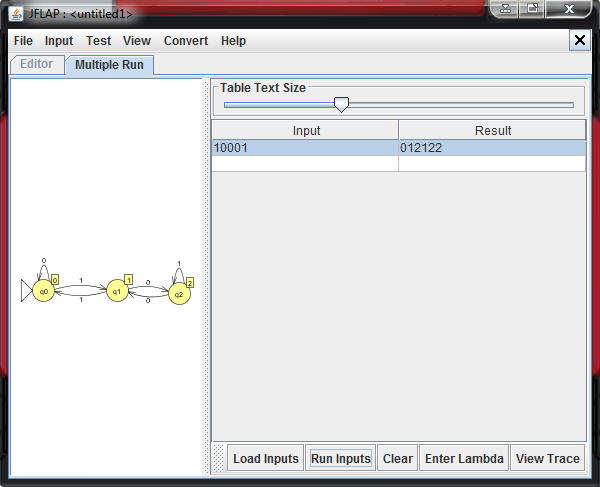
1. **10 mod 3**

****

1. **13 mod 3**

****

1. **17 mod 3**

****

1. Buatlah Algoritmanya pada masing mesin more tersebut:

**Jawaban:**

**Algoritma :**

**M =** Q,ε,δ,S,Δ,λ

**Q =** { q0,q1,q2}

€ = { 0,1 }

**S =** { q0 }

**Δ =** { 0, 1, 2 }

**Λ =** { q0 = 0},{ q1 = 1}, {q2 = 2}

|  |  |  |
| --- | --- | --- |
| δ | 0 | 1 |
| q0 | q0 | q1 |
| q1 | q2 | q0 |
| q2 | q1 | q2 |

Flowchart :

Start

0,1

q0

0,1

0

1

q1

0

0,1

0

q2

0 1

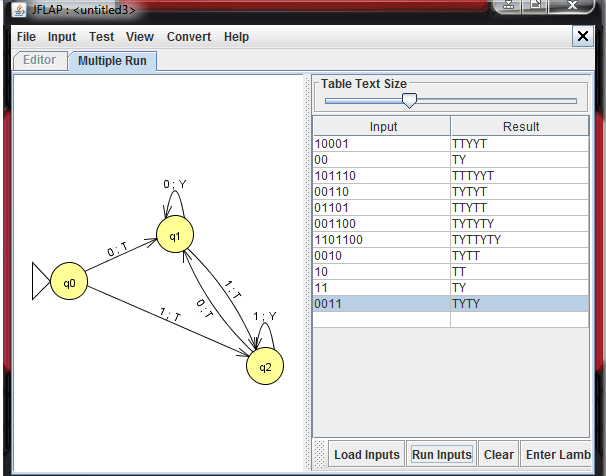
0,1

0,1,2

End

1. Tuliskan 10 inputan yang bisa diterima mesin mealy :

**Jawaban :**

****

2. Buatlah Algoritmanya pada mesin mealy tersebut :

**Jawaban :**

**M =** Q,ε,δ,S,Δ,λ

**Q =** { q0,q1,q2}

€ = { 0,1 }

**S =** { q0 }

**Δ =** { T,Y }

**Λ =** { q0 ,0 = T }

{ q0, 1 = T }

**Λ** = {q1,0 = Y }

{q1,1 = T }

**Λ =** {q2,0 = T }

{q2,1 = Y }

|  |  |  |
| --- | --- | --- |
| δ | 0 | 1 |
| q0 | q1 = T | q2 = T |
| q1 | q1 = Y | q2 = T |
| q2 | q1 = T | q2 = Y |

**Flowchart**

start

End

T,Y

0,1

q2

0,1

0,1

q1

q0

0,1

0

0

1

0 1

1

**Modul 3**

Flowchart

T

**START**

**Input A , B**

q0

q0 =n

Push = nZ

Pop = Z

A

T

**ACCEPT**

**REJECT**

**END**

T

q4

q2 = a

Push = nn

Pop = n

A

Y

q2

q1 = z

Push = 

Pop = n

A

q1

q1 = a

Push = nn

Pop = n

q3

Push = 

Pop = 

R

q1 = r

Kesimpulan dari tes di atas :

Tes ke-1 : “nazar”

Hasil : Accept atau diterima

Input ke-1 diterima atau sukses : karena FSA diatas menggunakan mekanisme tumpukan (stack), sehingga inputan diatas saling mengisi posisi ditumpukan dan dapat diterima hingga sampai dengan final state.

Tes ke-2 : “naz”

Hasil : Reject atau ditolak

di reject oleh FSA : karena ketika dilakukan tes FSA tidak dapat menemui final state, itu dikarenakan tumpukan yang di read oleh FSA hanya sampai ‘na’ dan tidak sampai pada state akhir

Kesimpulan keseluruhan : Jika perintah input yang dimasukkan dapat dibaca dengan baik oleh FSA maka inputan tersebut dapat diterima oleh FSA karena menemui final state, namun jika salah satu input ada yang lebih atau yang kurang, sehingga tidak dapat dibaca oleh FSA maka inputan tersebut ditolak atau reject oleh FSA karena tidak menemui final state.

**Modul 4**

**ALGORITMA**

START

q0

q0 = a

a = x

Head go right

q1

a = a

Head go right

a

x

aabbcc

q1 = a

y

q1 = y

y

q1 = b

y = y

Head go right

b = y

Head go right

q3

y = y

Head go left

y

q3 = y

z

q3 = z

z

q3 = b

z = z

Head go left

b = z

Head go left

a

q3 = a

a = a

Head go left

x

q3 = x

x = x

Head go right

q2

z = z

Head go right

z

q2 = z

b

q2 = b

z

q2 = c

b = b

Head go right

c = z

Head go left

q0 = y

y = y

Head go left

y

q0 = null

null = null

Head Stop

null

q4

x = x

Head go left

x

q4 = x

null = null

Head go right

null

q4 = null

END

q6

Accept

Reject

null = null

Head stop

q5 = a

null

x = x

Head go right

y = y

Head go right

q5 = b

x

q5 = z

y

q5 = y

z

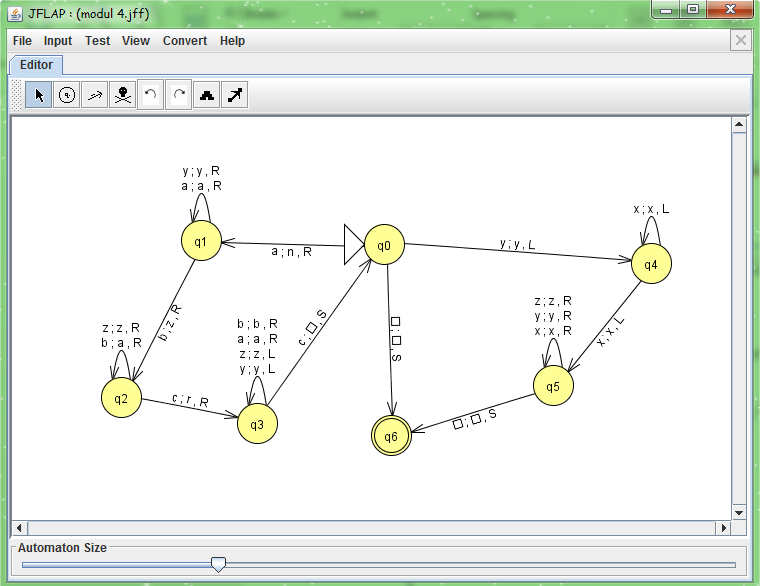
z = z

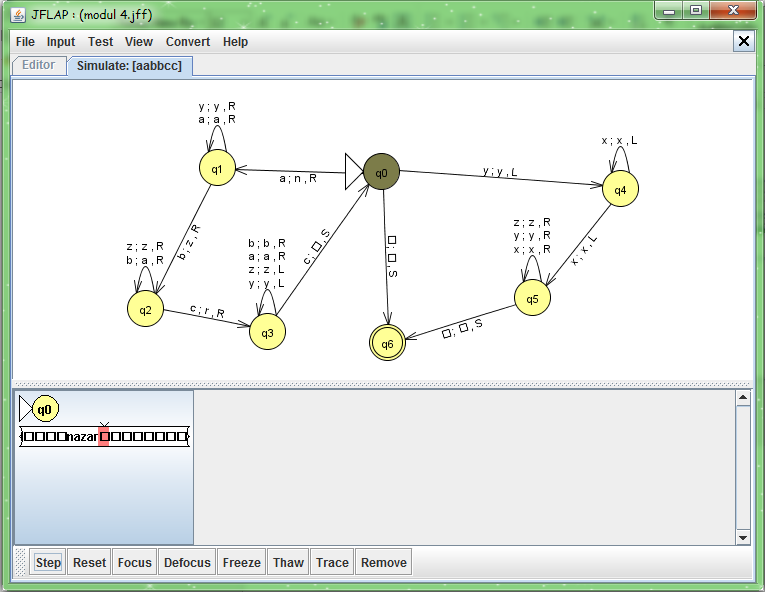
Head go right

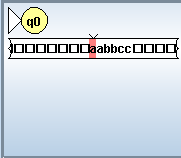
q5

1. Buat kembali seperti langkah Praktikum Dengan Hasil Akhir Menjadi Nama Masing-masing

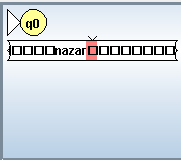
Gambar State :





Gambar Output Hasil :  


Gambar Input Dan Hasil Akhir :



1. Kesimpulan

Berdasarkan kepada praktikum yang dilakukan dengan input dan output yang dihasilkan mesin turing dapat disimpulkan bahwa mesin turing ini berfungsi pada suatu pita yang dapat diisi dengan symbol tunggal, lalu terdapat head, head merupakan pointer yang berwarna merah yang menunjukan posisi dari head, head dapat bergerak ke kiri ataupun ke kanan, head dapat membaca input pada pita dan mengubah/ melakukan penulisan kepada pita, sehingga state dan input dapat mendapatkan hasil (accept).

**Modul 5**

1. Buatlah kembali program-program pascal diatas kedalam bahasa C/C++.atau java

Contoh1 :

Source code :

***package tbo\_1;***

***import java.util.Scanner;***

***public class Tbo\_1 {***

***public static void main(String[] args) {***

***Scanner masukan = new Scanner (System.in);***

***String ekspresi;***

***char tampung;***

***int i;***

***System.out.print("Masukan Ekspresi : ");***

***ekspresi = masukan.next();***

***for (i=0; i<ekspresi.length();i++){***

***if ((ekspresi.charAt(i)>=65)&&(ekspresi.charAt(i)<=97)){***

***System.out.println("Huruf");***

***}***

***else {***

***System.out.println("Bukan Huruf");***

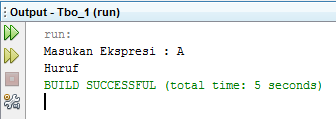
***}***

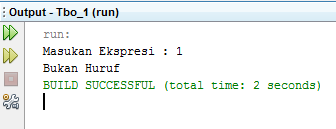
***}***

***}***

***}***

***ScreenShot :***





Contoh 2:

Source Code :

***package tbo2;***

***import java.util.Scanner;***

***public class TBO2 {***

***public static void main(String[] args) {***

***Scanner in=new Scanner(System.in);***

***int x,y;***

***System.out.print("Masukan Nilai X = ");x=in.nextInt();***

***System.out.print("Masukan Nilai Y = ");y=in.nextInt();***

***hitung (x,y);***

***}***

***public static void hitung(int a, int b) {***

***int c;***

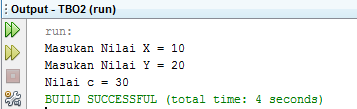
***c=a+b;***

***System.out.println("Nilai c = "+c);***

***}***

***}***

***ScreenShot :***



Contoh 3:

***Source Code:***

***package tbo3;***

***import java.util.Arrays;***

***import java.util.Scanner;***

***public class TBO3 {***

***public static void main(String[] args) {***

***Scanner in= new Scanner(System.in);***

***String Kalimat;***

***int posisi;***

***System.out.print("Masukan Sebuah Kalimat : ");Kalimat=in.nextLine();***

***String [] kata = Kalimat.split(" ");***

***System.out.println("--Hasilnya--");***

***for (int i=0;i<kata.length;i++){***

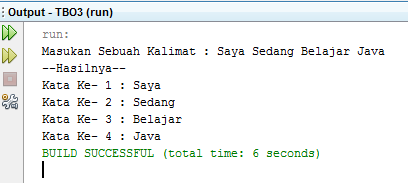
***posisi=i+1;***

***System.out.println("Kata Ke- " +posisi+ " : " + kata[i]);***

***}***

***}***

***ScreenShot :***



1. Buatlah kesimpulan terhadap proses kompilasi di atas.

* Pada contoh 1 merupakan program mentukan huruf atau bukan, jika kita menginputkan huruf output jika tidak maka akan muncul kata bukan huruf.
* Pada contoh 2 merupakan pengiriman parameter dengan nilai, seperti pada program kita menginputkan x,y tetapi pada proses program intnya adalah a,b.
* Pada contoh 3 kesimpulannya proses split kalimat ke dalam kata dengan spasi sebagai acuan untuk proses split pada program tersebut.

**Modul 6**

1. Buatlah kembali program-program pascal diatas kedalam bahasa C/C++.atau java?

package tbo\_4;

import static com.sun.javafx.fxml.expression.Expression.not;

import java.util.Scanner;

public class TBO\_4 {

public static void main(String[] args) {

Scanner masukan = new Scanner(System.in);

String variable;

System.out.print("Masukan Sebuah Variable :");variable=masukan.nextLine();

variable.startsWith(variable);

int i = 0;

if((variable.charAt(i)>=65)&&(variable.charAt(i)<=97)||(variable.charAt(i)>=97)&&(variable.charAt(i)<=122)){

System.out.println("Variable Benar");

}

else {

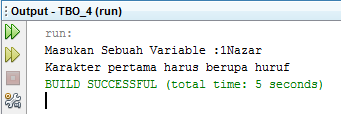
System.out.println("Karakter pertama harus berupa huruf");

}

}

}

Screenshot :



Contoh 2 :

package tbo\_5;

import java.util.Scanner;

public class Tbo\_5 {

public static void main(String[] args) {

Scanner in = new Scanner (System.in);

String variable;

System.out.print("Masukan variable : ");variable=in.nextLine();

for (int i=1;i<variable.length();i++){

if ((variable.charAt(i)>=65)&&(variable.charAt(i)<=97)||(variable.charAt(i)>=97)&&(variable.charAt(i)<=122)){

System.out.println("Variable benar");

}

else {

System.out.println("Ada kesalahan dengan variabel Anda");

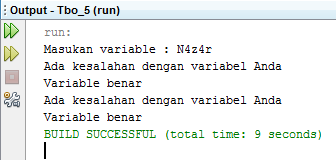
}

}

}

}

Screenshot :



1. Coba dan amati program pascal dibawah ini, lalu berikan kesimpulan.

**Modul 7**

**1. Buat program dan Lexical Analysis (scanner) untuk :**

**a) Statement : Fahrenheit := 32 + celcius \* 1.8**

**b) Statement : Phi := 3,14 + r \* r**

Source Code :

#include<conio.h>

#include<string.h>

#include<ctype.h>

#include<stdio.h>

class Lex\_Ana

{

public :

int i,j,k,ti,oprfl,delfl,keyfl,Litfl,cLitfl,ptfl,tfl,mxIdeni,Liti,mxIdenj,mxOpri,mxOprj,mxDeli,Erri;

int uqIdi,uqOpi,uqDei,defOpi,defKeyi,cLiti;

char mStr[50],mxIden[20][20],mxOpr[20][20],mxDel[20],tmpStr[50];

char defOprStr[20][20],defOpr[15],defDel[15],defKey[20][20];

char uqOpr[20][20],uqIden[20][20],uqIdfl[20],uqOpfl[20],uqDefl[20],uqDel[20];

char Opr[20][20],invalTok[20][20],Con[20][20],Iden[20][20],

Key[20][20],inTok[20],Lit[20][20],Err[20][20],cLit[20][20];

int Opri,invToki,Coni,Ideni,Keyi,Deli,inToki;

Lex\_Ana(char \*str)

{

strcpy(mStr,str);

//Defined Keywords

defKeyi=13;

strcpy(defKey[0],"int");

strcpy(defKey[1],"float");

strcpy(defKey[2],"double");

strcpy(defKey[3],"for");

strcpy(defKey[4],"if");

strcpy(defKey[5],"else");

strcpy(defKey[6],"while");

strcpy(defKey[7],"do");

strcpy(defKey[8],"return");

strcpy(defKey[9],"char");

strcpy(defKey[10],"break");

strcpy(defKey[11],"goto");

strcpy(defKey[12],"void");

//Defined Operators

defOpi=18;

strcpy(defOprStr[0],"+");

strcpy(defOprStr[1],"-");

strcpy(defOprStr[2],"\*");

strcpy(defOprStr[3],"/");

strcpy(defOprStr[4],"%");

strcpy(defOprStr[5],"=");

strcpy(defOprStr[6],"<");

strcpy(defOprStr[7],">");

strcpy(defOprStr[8],"<=");

strcpy(defOprStr[9],">=");

strcpy(defOprStr[10],"==");

strcpy(defOprStr[11],"=+");

strcpy(defOprStr[12],"=-");

strcpy(defOprStr[13],"=\*");

strcpy(defOprStr[14],"=/");

strcpy(defOprStr[15],"=%");

strcpy(defOprStr[16],"++");

strcpy(defOprStr[17],"--");

strcpy(defOpr,"+-/\*%=<>");

strcpy(defDel,";,:{}[]()");

}

int IsMxIden(char ch)

{

return((IsIden(ch)||IsFloat(ch))?1:0);

}

int IsIden(char ch)

{

return((isalpha(ch)||ch=='\_')?1:0);

}

int IsFloat(char ch)

{

return((isdigit(ch)||ch=='.')?1:0);

}

int IsOpr(char ch)

{

oprfl=0;

for(ti=0;ti<strlen(defOpr)&&!oprfl;ti++)

{

if(ch==defOpr[ti])

{

oprfl=1;

}

}

return oprfl;

}

int IsDel(char ch)

{

delfl=0;

for(ti=0;ti<strlen(defDel)&&!delfl;ti++)

{

if(ch==defDel[ti])

{

delfl=1;

}

}

return delfl;

}

int IsKey(char \*str)

{

keyfl=0;

for(ti=0;ti<defKeyi&&!keyfl;ti++)

{

if(!strcmp(defKey[ti],str))

{

keyfl=1;

}

}

return keyfl;

}

void Find\_Lex()

{

i=0;

cLiti=0;

Liti=0;

Erri=0;

mxIdeni=0;

mxIdenj=0;

mxOpri=0;

mxOprj=0;

mxDeli=0;

inToki=0;

while(mStr[i]!='\0')

{

if(IsMxIden(mStr[i]))

{

while(IsMxIden(mStr[i]))

{

mxIden[mxIdeni][mxIdenj++]=mStr[i++];

if(!IsMxIden(mStr[i]))

{

mxIden[mxIdeni++][mxIdenj]='\0';

mxIdenj=0;

}

}

}

else if(mStr[i]=='"')

{

Litfl=1;

k=0;

tmpStr[k++]=mStr[i];

for(j=i+1;j<strlen(mStr)&&Litfl;j++)

{

tmpStr[k++]=mStr[j];

if(mStr[j]=='"')

{

tmpStr[k]='\0';

strcpy(Lit[Liti++],tmpStr);

Litfl=0;

i=j+1;

}

}

if(Litfl)

{

strcpy(Err[Erri++],"Invalid Use of \"");

i++;

}

}

else if(mStr[i]=='\'')

{

cLitfl=0;

k=0;

tmpStr[k++]=mStr[i];

for(j=i+1;j<strlen(mStr)&&!cLitfl;j++)

{

tmpStr[k++]=mStr[j];

if(mStr[j]=='\'')

{

tmpStr[k]='\0';

if(strlen(tmpStr)!=3)

{

strcpy(Err[Erri++],"Invalid Use of \'");

cLitfl=1;

i++;

}

else

{

strcpy(cLit[cLiti++],tmpStr);

cLitfl=1;

i=j+1;

}

}

}

if(!cLitfl)

{

strcpy(Err[Erri++],"Invalid Use of \'");

cLitfl=1;

i++;

}

}

else if(IsOpr(mStr[i]))

{

while(IsOpr(mStr[i]))

{

mxOpr[mxOpri][mxOprj++]=mStr[i++];

if(!IsOpr(mStr[i]))

{

mxOpr[mxOpri++][mxOprj]='\0';

mxOprj=0;

}

}

}

else if(IsDel(mStr[i]))

{

while(IsDel(mStr[i]))

{

mxDel[mxDeli++]=mStr[i++];

}

mxDel[mxDeli]='\0';

}

else if(mStr[i]!=' ')

{

inTok[inToki++]=mStr[i++];

inTok[inToki]='\0';

}

else

{

i++;

}

}

}

void Rem\_Dup()

{

uqIdi=0;

uqDei=0;

uqOpi=0;

for(i=0;i<20;i++)

{

uqOpfl[i]=0;

uqIdfl[i]=0;

uqDefl[i]=0;

}

for(i=1;i<mxIdeni+1;i++)

{

if(uqIdfl[i-1]==0)

{

strcpy(uqIden[uqIdi++],mxIden[i-1]);

for(j=i;j<mxIdeni;j++)

{

if(!strcmp(mxIden[j],mxIden[i-1]))

{

uqIdfl[j]=1;

}

}

}

}

for(i=1;i<mxOpri+1;i++)

{

if(uqOpfl[i-1]==0)

{

strcpy(uqOpr[uqOpi++],mxOpr[i-1]);

for(j=i;j<mxOpri;j++)

{

if(!strcmp(mxOpr[j],mxOpr[i-1]))

{

uqOpfl[j]=1;

}

}

}

}

for(i=1;i<mxDeli+1;i++)

{

if(uqDefl[i-1]==0)

{

uqDel[uqDei++]=mxDel[i-1];

uqDel[uqDei]='\0';

for(j=i;j<mxDeli;j++)

{

if(mxDel[j]==mxDel[i-1])

{

uqDefl[j]=1;

}

}

}

}

}

void Sep\_Val()

{

Opri=0;

invToki=0;

Ideni=0;

Coni=0;

Keyi=0;

for(i=0;i<uqIdi;i++)

{

if(IsIden(uqIden[i][0]))

{

if(IsKey(uqIden[i]))

{

strcpy(Key[Keyi++],uqIden[i]);

}

else

{

ptfl=1;

for(j=0;j<strlen(uqIden[i])&&ptfl;j++)

{

if(uqIden[i][j]=='.')

{

strcpy(invalTok[invToki++],uqIden[i]);

ptfl=0;

}

}

if(ptfl)

{

strcpy(Iden[Ideni++],uqIden[i]);

}

}

}

else

{

tfl=1;

ptfl=0;

for(j=0;j<strlen(uqIden[i])&&tfl;j++)

{

if(IsIden(uqIden[i][j]))

{

strcpy(invalTok[invToki++],uqIden[i]);

tfl=0;

}

else if(uqIden[i][j]=='.')

{

if(ptfl)

{

strcpy(invalTok[invToki++],uqIden[i]);

tfl=0;

}

if(!ptfl)

{

ptfl=1;

}

}

}

if(tfl)

{

strcpy(Con[Coni++],uqIden[i]);

}

}

}

for(i=0;i<uqOpi;i++)

{

tfl=1;

for(j=0;j<defOpi&&tfl;j++)

{

if(!strcmp(uqOpr[i],defOprStr[j]))

{

strcpy(Opr[Opri++],uqOpr[i]);

tfl=0;

}

}

if(tfl)

{

strcpy(invalTok[invToki++],uqOpr[i]);

}

}

}

void Display()

{

printf("Identifiers : ");

printf("\n");

for(i=0;i<Ideni;i++)

{

printf("> ");

puts(Iden[i]);

}

printf("Angka: ",Coni);

printf("\n");

for(i=0;i<Coni;i++)

{

printf("> ");

puts(Con[i]);

}

printf("Literals: ",Liti+cLiti);

printf("\n");

for(i=0;i<Liti;i++)

{

printf("> ");

puts(Lit[i]);

}

for(i=0;i<cLiti;i++)

{

printf("> ");

puts(cLit[i]);

}

printf("Keywords: ",Keyi);

printf("\n");

for(i=0;i<Keyi;i++)

{

printf("> ");

puts(Key[i]);

}

printf("Operators: ",Opri);

printf("\n");

for(i=0;i<Opri;i++)

{

printf("> ");

puts(Opr[i]);

}

printf("Tanda Baca: ",uqDei);

printf("\n");

if(uqDei)

{

printf("> ");

puts(uqDel);

}

printf("Token Error: ",Erri);

printf("\n");

for(i=0;i<Erri;i++)

{

puts(Err[i]);

}

}

};

void main()

{

clrscr();

char mstr[50];

printf("\nMasukan String : ");

gets(mstr);

Lex\_Ana oblex(mstr);

oblex.Find\_Lex();

oblex.Rem\_Dup();

oblex.Sep\_Val();

oblex.Display();

getch();

}

Screenshot :

a. 

b. 