

# **LAPORAN TUGAS PEKAN 8 ALGORITMA DAN PEMROGRAMAN**

“GUI Operator Logika”



Dosen Pengampu:

DR. Wahyudi, S.T., M.T.

Asisten Lab:

Muhammad Zaki Al Hafiz

Disusun Oleh:

Hamdi Sidqi Alifi

2511531009

Fakultas Teknologi Informasi

Departemen Informatika

Universitas Andalas

2025

## **KATA PENGANTAR**

Puji dan syukur senantiasa penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan ini guna memenuhi penugasan pasca-pratikum mata kuliah Algoritma Pemrograman pekan ke-5 yang dilaksanakan pada hari Rabu tanggal 19 November 2025.

Pada kesempatan ini, Penulis ingin mengucapkan terima kasih kepada Bapak DR. Wahyudi. S.T.M.T, dosen pengampu mata kuliah Algoritma Pemrograman, yang telah memberikan tugas ini. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung serta membantu penulis dalam penyelesaian laporan tugas Algoritma Pemrograman ini.

Penulis sadar bahwa laporan ini masih memiliki beberapa kekurangan dikarenakan terbatasnya pengalaman dan pengetahuan yang penulis miliki. Oleh karena itu, penulis harap adanya bentuk saran dan kritik yang membangun dari berbagai pihak.

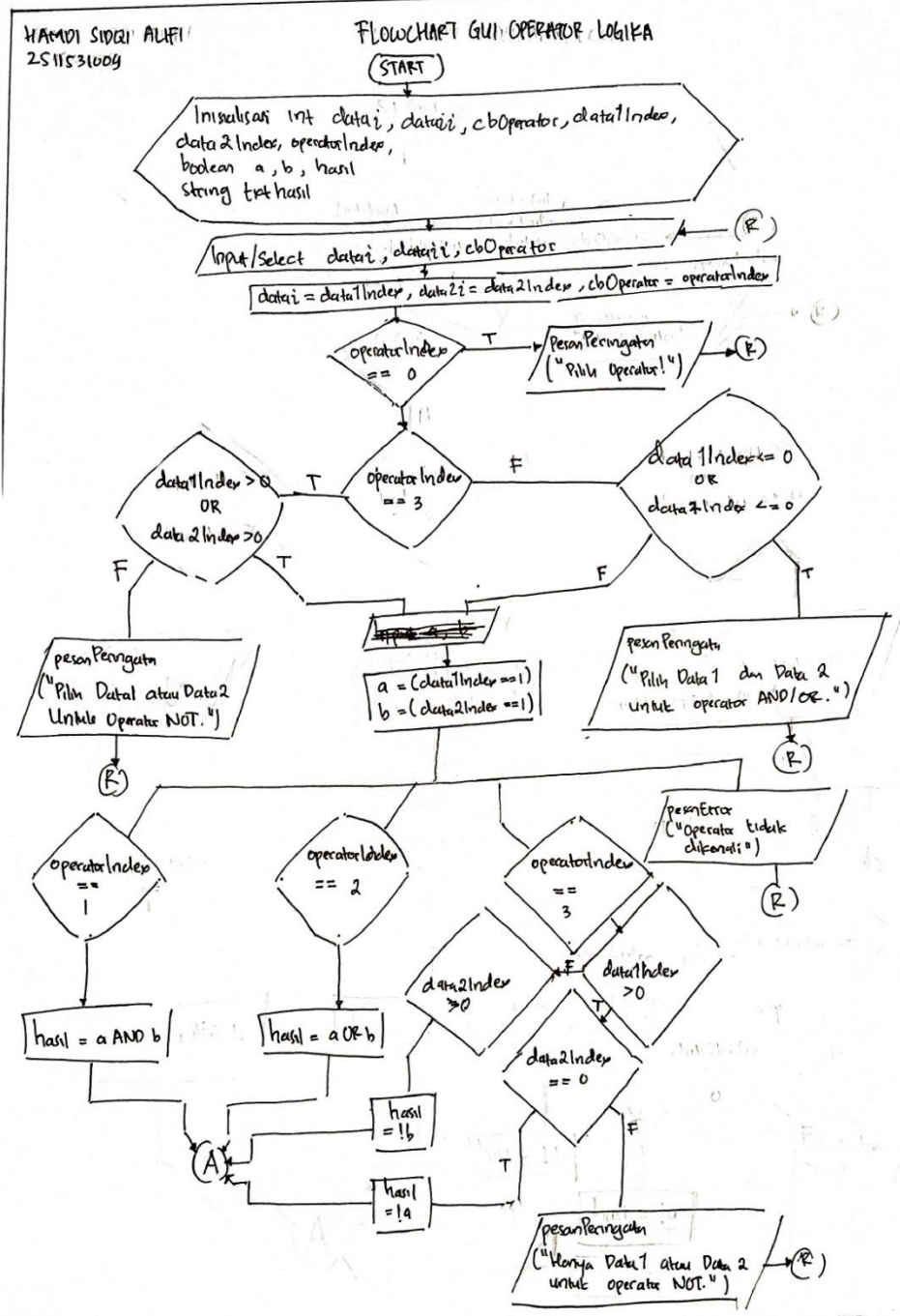
Akhir kata, penulis berharap laporan ini dapat bermanfaat bagi perkembangan dunia pendidikan.

## **INSTRUKSI**

Dalam pekan ini, mahasiswa yang mengambil mata kuliah Algoritma dan Pemrograman diberikan sebuah tugas, yaitu mengembangkan sebuah program operator logika dengan menggunakan windowbuilder. Caranya sama seperti membuat Operator Aritmatika seperti yang telah penulis lakukan pada praktikum Algoritma dan Pemrograman pekan kedelapan, hanya saja tipe data yang digunakan pada input dan output adalah boolean, bukan integer seperti sebelumnya.

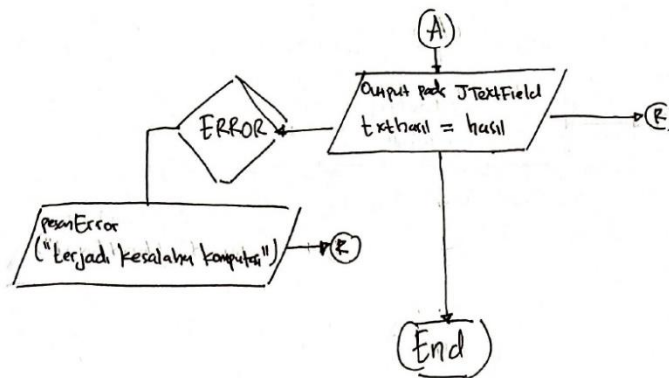
# PELAKSANAAN

## Flowchart



HAMDI SIDQI ALFI  
2511531009

## FLOWCHART GUI OPERATOR LOGIKA#2



### Pseudo-code

#### GUI Operator Logika

{ mengembangkan aplikasi yang dapat menampilkan hasil operasi logika dengan satu dan/atau dua input data }

#### Deklarasi

int datai, dataii, cbOperator, data1Index, data2Index, operatorIndex  
boolean a, b, hasil  
string txthasil

## Algoritma dan Logika

pesanPeringatan(pesan : String) → String

### Algoritma/Syntax

1. showMessageDialog(pesan → Warning Message)

pesanError(pesan : String) → String

### Algoritma/Syntax

1. showMessageDialog(pesan → Error Message)
1. select.data1 {“Pilih Data”, “TRUE”, “FALSE”}
2. select.data2 {“Pilih Data”, “TRUE”, “FALSE”}
3. select.cbOperator {“Pilih Operator”, “AND”, “OR”, }
4. data1Index = data1
5. data2Index = data2
6. operatorIndex = cbOperator
7. if (operatorIndex == 0)
8.     pesanPeringatan(“Pilih Operator!”)
9.     return
10. if (operatorIndex == 3)
11.     if (data1Index > 0 OR data2Index > 0)
12.         // try-catch
13.     else
14.         pesanPeringatan(“Pilih Data 1 atau Data 2 untuk operator NOT.”)
15.     return
16. else if (data1Index <= 0 OR data2Index <= 0)
17.     pesanPeringatan(“Pilih Data 1 dan Data 2 untuk operator AND/OR.”)
18.     Return
19. try
20.     a = (data1Index)
21.     b = (data2Index)
22.     switch (operatorIndex)
23.         case 1 : hasil = a AND b
24.         case 2 : hasil = a OR b
25.         case 3 :  
           if (data1Index > 0)  
               if (data2Index == 0)  
                   hasil = !a  
               else  
                   pesanPeringatan(“Hanya Data 1 atau Data 2 untuk operator NOT.”)
- else if (data2Index > 0)  
               hasil = !b
26.         default :  
           pesanError(“Operator tidak dikenali”)
- return
27.     txthasil = (String(hasil))
28. catch (exception ex)
29.     pesanError(“Terjadi kesalahan komputasi”)



## Java language

```
26 private void pesanPeringatan(String pesan) {
27     JOptionPane.showMessageDialog(this,pesan,"Peringatan",JOptionPane.WARNING_MESSAGE);
28 }
29 private void pesanError(String pesan) {
30     JOptionPane.showMessageDialog(this,pesan,"Kesalahan",JOptionPane.ERROR_MESSAGE);
31 }
32
```

Lines ini berfungsi untuk mengeluarkan dialog box agar pengguna tahu ketika dia melakukan kesalahan input

```
104 JButton Hitung = new JButton("Hitung");
105 Hitung.addActionListener(new ActionListener() {
106     boolean hasil;
107     public void actionPerformed(ActionEvent e) {
108         boolean hasil = false;
109
110         int data1Index = data1.getSelectedIndex();
111         int data2Index = data2.getSelectedIndex();
112         int operatorIndex = cbOperator.getSelectedIndex();
113
114         if (operatorIndex == 0) {
115             pesanPeringatan("Pilih Operator!");
116             return;
117         }
118
119         if (operatorIndex == 3) {
120             if (data1Index > 0 || data2Index > 0) {
121                 // lanjut try-catch
122             } else {
123                 pesanPeringatan("Pilih Data 1 atau Data 2 untuk operator NOT.");
124                 return;
125             }
126         }
127         else if (data1Index <= 0 || data2Index <= 0) {
128             pesanPeringatan("Pilih Data 1 dan Data 2 untuk operator AND/OR.");
129             return;
130         }
131     }
132 }
```

Tujuan dari pemindahan data integer combobox ke variabel baru ialah agar tidak menimbulkan *code redundant*

```
130     }
131 }
132
133 try {
134     boolean a = (data1Index == 1);
135     boolean b = (data2Index == 1);
136
137     switch (operatorIndex) {
138         case 1: // AND
139             hasil = a && b;
140             break;
141
142         case 2: // OR
143             hasil = a || b;
144             break;
145
146         case 3: // NOT
147             if (data1Index > 0) {
148                 if (data2Index == 0) {
149                     hasil = !a;
150                 } else {
151                     pesanPeringatan("Hanya Data 1 atau Data 2 untuk operator NOT.");
152                 }
153             }
154             else if (data2Index > 0) {
155                 hasil = !b;
156             }
157             break;
158     }
159 }
```



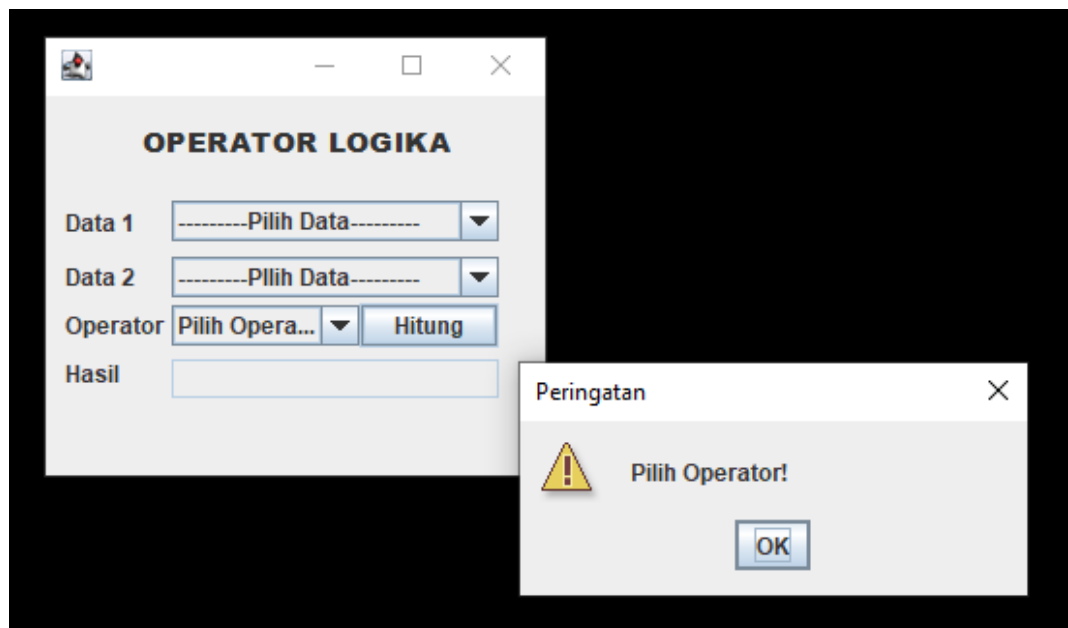
Disini penulis lebih merasa cocok untuk mengembangkan program apabila kita menggunakan switch-case daripada multi-if

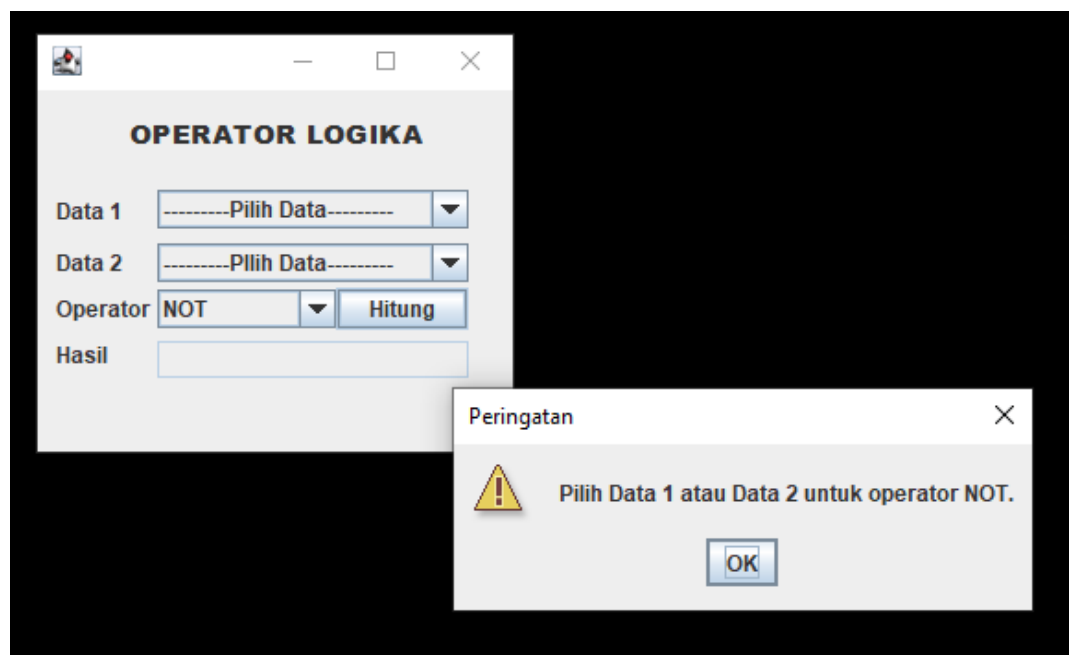
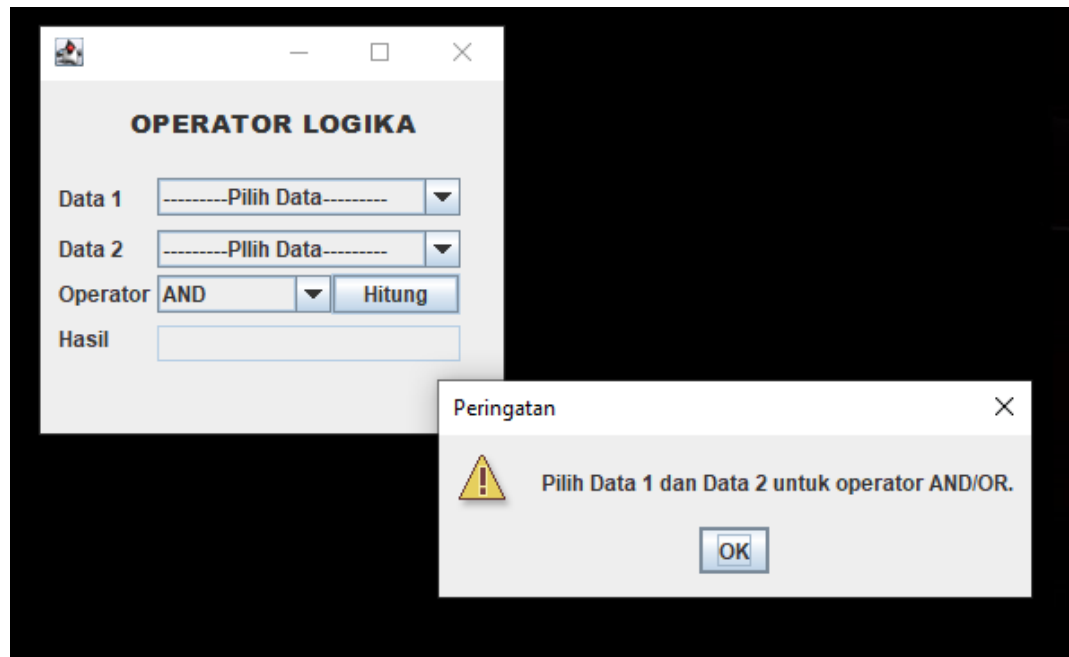
```
157         break;
158
159         default:
160             pesanError("Operator tidak dikenali.");
161             return;
162     }
163
164     txthasil.setText(String.valueOf(hasil));
165
166     } catch (Exception ex) {
167         pesanError("Terjadi kesalahan komputasi.");
168     }
169 }
```


Hasil dari logika akan diinput ke dalam Component Class JTextField txthasil

Output :

Berikut adalah semua kombinasi input yang mungkin keluar jika program dijalankan





 — □ ×


### OPERATOR LOGIKA

Data 1

Data 2

Operator

Hasil

 — □ ×


### OPERATOR LOGIKA

Data 1

Data 2

Operator

Hasil

 — □ ×


### OPERATOR LOGIKA

Data 1

Data 2

Operator

Hasil

 — □ ×


**OPERATOR LOGIKA**

Data 1

Data 2

Operator

Hasil

 — □ ×


**OPERATOR LOGIKA**

Data 1

Data 2

Operator

Hasil

 — □ ×

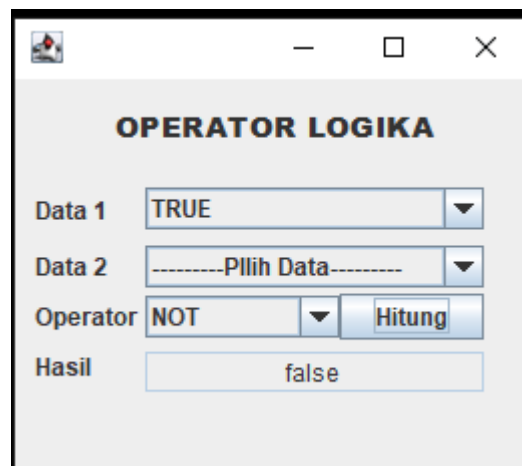
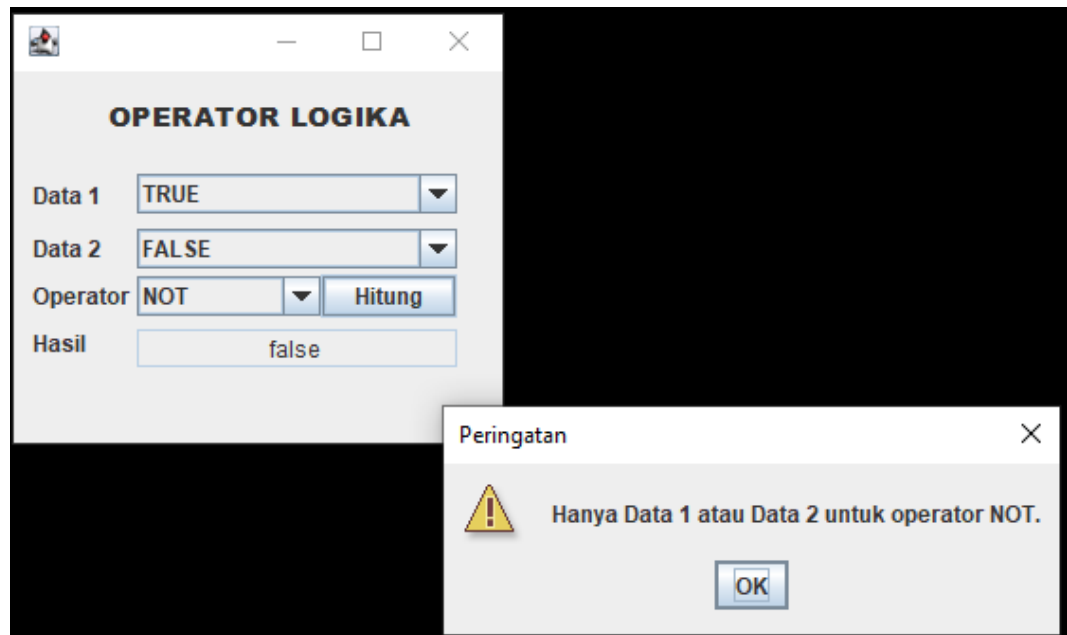
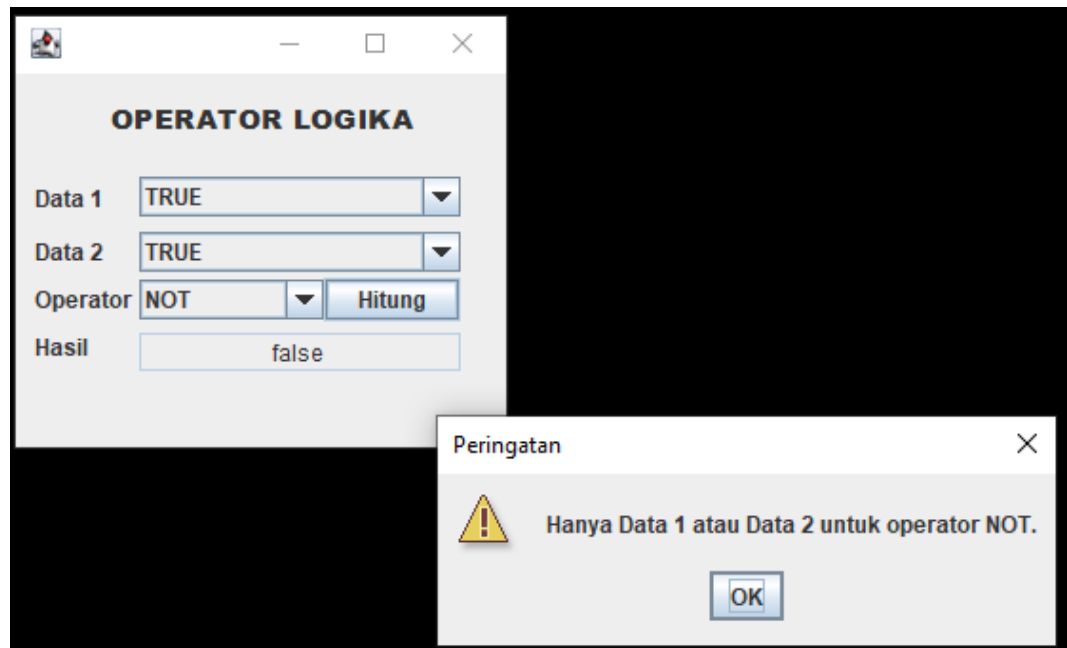
**OPERATOR LOGIKA**

Data 1

Data 2

Operator

Hasil



**OPERATOR LOGIKA**

Data 1 FALSE

Data 2 -----Pilih Data-----

Operator NOT

Hitung

Hasil true

**OPERATOR LOGIKA**

Data 1 -----Pilih Data-----

Data 2 TRUE

Operator NOT

Hitung

Hasil false

**OPERATOR LOGIKA**

Data 1 -----Pilih Data-----

Data 2 FALSE

Operator NOT

Hitung

Hasil true