LAPORAN TUGAS AKHIR LOGIKA MATEMATIKA

KALKULATOR GERBANG LOGIKA

DISUSUN OLEH :

- Restu Wibisono (2340506061)
- Faizal D Nugraha (2340506065)
- Ilham Kukuh F (2320506043)
- Arel Lioza Akhmad (2320506050)

JURUSAN TEKNOLOGI INORMASI

FAKULTAS TEKNIK

UNIVERSITAS TIDAR

2023

LAPORAN

TUGAS AKHIR LOGIKA MATEMATIKA

Diisi Mahasiswa Praktikan	
Nama Praktikan	Restu, Faizal, Ilham, Arellioza
NPM	2340506061, 2340506065, 2320506042, 2320506050
Kelompok	Kel. Tidak Cemara
Judul Praktikum	Kalkulator Gerbang Logika
Tanggal Praktikum	30 November 2023
Diisi Dosen Pengampu	
Tanggal Pengumpulan	
Catatan	

PENGESAHAN	NILAI
Disahkan oleh :	
Dosen Pengampu	
D. Jayus Noor Salim	

PROGRAM STUDI S1 TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TIDAR

2023

BAB I
PENDAHULUAN

1.1 Tujuan Praktikum

Beberapa tujuan yang ingin dicapai dalam praktikum ini:

1. Mempelajari gerbang logika atau memahami dasar-dasar logika matematika.
2. Untuk memahami fungsi dari tiap-tiap gerbang logika, seperti AND (konjungsi), OR (disjungsi), NOT (negasi), NAND, NOR, dan XOR.
3. Memahami kegunaan gerbang khusus seperti XOR (eksklusif OR), NAND (NOT AND), dan NOR (NOT OR)
4. Untuk melengkapi tugas akhir " Logika Matematika".

1.2 Dasar Teori

Gerbang logika merupakan komponen dasar elektronika yang beroperasi berdasarkan prinsip logika matematika. Gerbang logika memiliki dua nilai dasar, yaitu true (benar) dan false (salah). Nilai-nilai ini dapat ditampilkan dengan angka 1 dan 0, masing-masing.

Tabel kebenaran adalah tabel yang digunakan untuk menampilkan semua kemungkinan nilai input dan output dari sebuah fungsi logika atau gerbang logika dari berbagai nilai keluaran dari suatu gerbang logika dari setiap kombinasi nilai masukan.

Dalam gerbang logika ada beberapa operator yang biasa digunakan seperti:

1. OR Gate (Gerbang OR)

Gerbang OR memiliki dua atau lebih input dan hanya memiliki satu output.

- Output gerbang OR akan bernilai 1 jika salah satu atau semua masukan bernilai 1.
- Jika semua masukan bernilai 0, maka output akan bernilai 0.
- OR disimbolkan dengan (∨)

2. AND Gate (Gerbang AND)

Gerbang AND memiliki dua atau lebih input dan hanya memiliki satu output.

- Output gerbang AND akan bernilai 1 hanya jika semua masukan bernilai 1.
- Jika salah satu atau semua masukan bernilai 0, maka output akan bernilai 0.

- AND disimbolkan dengan (Λ)

3. NOT Gate (Gerbang NOT)

Gerbang NOT hanya memiliki satu masukan dan satu output.

- Output gerbang NOT akan bernilai 1 jika masukan bernilai 0.
- Jika masukan bernilai 1, maka output akan bernilai 0.
- NOT disimbolkan dengan (~)

4. NOR Gate (Gerbang NOR)

Gerbang NOR (negasi OR) adalah gabungan dari gerbang OR dan gerbang NOT.

- Output gerbang NOR akan bernilai 1 jika semua masukan bernilai 0.
- Jika salah satu atau semua masukan bernilai 1, maka output akan bernilai 0.
- NOR disimbolkan dengan OR gate dan NOT gate

5. NAND Gate (Gerbang NAND)

Gerbang NAND (negasi AND) adalah gabungan dari gerbang AND dan gerbang NOT.

- Output gerbang NAND akan bernilai 1 jika salah satu atau semua masukan bernilai 0.
- Jika semua masukan bernilai 1, maka output akan bernilai 0.
- NAND disimbolkan dengan AND gate dan NOT gate

6. XOR Gate (Gerbang XOR)

Gerbang XOR (eksklusif OR) memiliki dua masukan dan satu output.

- Output gerbang XOR akan bernilai 1 jika salah satu masukan bernilai 1, tetapi tidak keduanya.
- Jika kedua masukan bernilai sama, maka output akan bernilai 0.
- XOR disimbolkan dengan (+)

BAB II
METODE PRAKTIKUM

2.1 Alat

Komputer: Pastikan memiliki komputer yang dapat menjalankan Python. Komputer harus memiliki sistem operasi yang kompatibel dengan Python dan spesifikasi yang memadai untuk menjalankan aplikasi Python dengan lancar.

2.2 Bahan

Editor teks atau IDE: Pilih editor teks atau IDE (lingkungan pengembangan terintegrasi) yang sesuai dengan preferensi. Beberapa opsi populer termasuk PyCharm, Visual Studio Code, Atom, Sublime Text, atau IDLE (bawaan Python). Pastikan memiliki editor teks atau IDE yang mendukung Python dan diinstal di komputer.

2.3 Langkah Kerja

1. Pertama siapkan alat dan bahan yang akan digunakan atau dibutuhhkan dalam praktikum

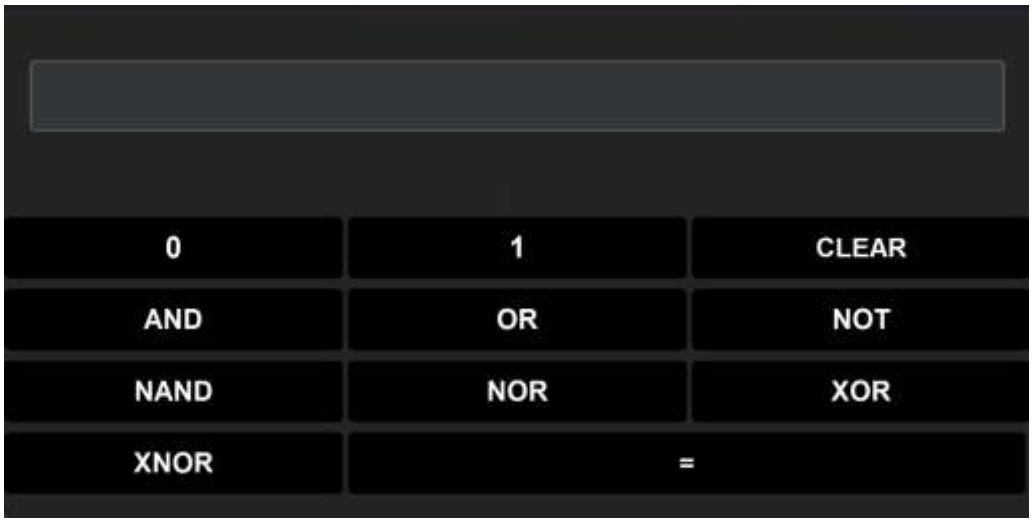
```
def show_page_two():
    page_one_frame.place_forget()
    my_label.place_forget()
    labeljudul.pack_forget()

    #2nd page
    label_two.place(x=0, y=0, relwidth=1, relheight=1)
    #entry
    entry.grid(row=0, column=1, columnspan=5,pady=50)
    #filler
    fillabel1.grid(row=4, column=0)
    fillabel2.grid(row=4, column=4)
    fillabel3.grid(row=0, column=0)
    #infobox
    infobox.grid(row=1, column=0, columnspan=5)
    #first row button
    button0.grid(row=2, column=1,pady=5)
    button1.grid(row=2, column=2,pady=5)
    buttonclear.grid(row=2, column=3, pady=5)
    #second row button
    buttonand.grid(row=3, column=1,pady=5)
    buttonor.grid(row=3, column=2,pady=5)
    buttonnot.grid(row=3, column=3,pady=5)
    #third row button
    buttonnand.grid(row=4, column=1, pady=5)
    buttonnor.grid(row=4, column=2, pady=5)
    buttonxor.grid(row=4, column=3, pady=5)
    #fourth row button
    buttonxnor.grid(row=5, column=1, pady=5)
    buttoncal.grid(row=5, column=3, pady=5)
```

2. Source code diatas adalah fungsi Python yang disebut show_page_two. Fungsi ini tampaknya digunakan untuk mengatur tata letak antarmuka pengguna dengan mengatur atau menempatkan beberapa elemen GUI (Graphical User Interface)
3. Fungsi ini menampilkan berbagai elemen GUI dan mengatur tata letaknya dengan menggunakan metode .grid() untuk menentukan posisi elemen-elemen tersebut dalam grid. Selain itu, beberapa elemen dari halaman pertama disembunyikan dengan menggunakan metode .place_forget() dan .pack_forget().

BAB III

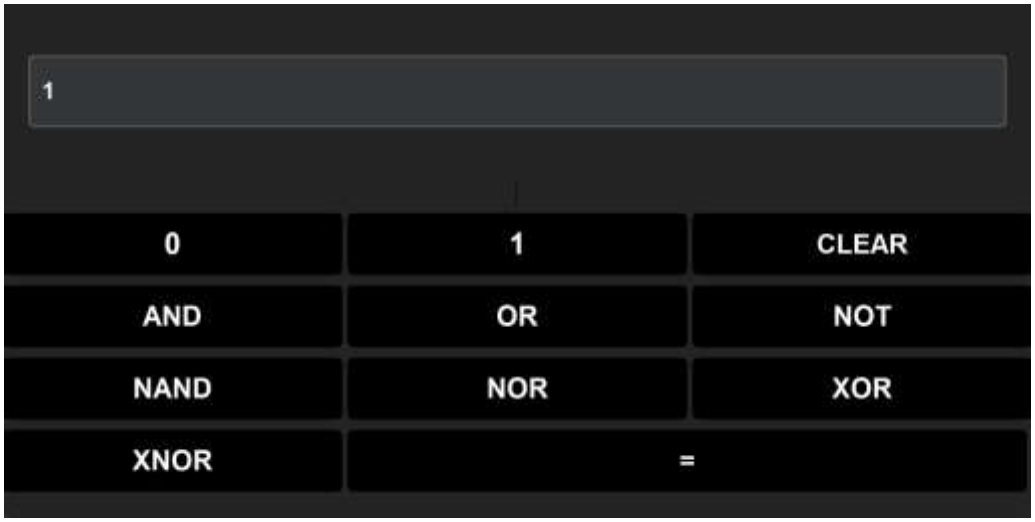
HASIL

1. Halaman utama dari program tersebut adalah sebagai berikut:.



User bisa memberikan input yang diinginkan, dan melakukan berbagai kombinasi gate seperti berikut:

Dengan memilih input apakah ingin 1 (True) atau 0 (False), kemudian user memilih gate logic yang ingin digunakan atau dicari. Output akan muncul pada box diatas seperti gambar berikut:



2. Logika Pemrograman yang Digunakan

Pada setiap Bahasa pemrograman memiliki operator logika dengan symbol yang berbeda-beda. Berikut adalah gerbang logika pada bahasa pemrograman JavaScript.

a. Logika AND

```
12          if operator == 'AND':
13              lvar.append((lvar[-2]) and (lvar[-1]))
```

1. if operator == 'AND':: Baris ini adalah pernyataan kondisional (if statement) yang memeriksa apakah nilai dari variabel operator sama dengan string 'AND'. Jika benar, maka blok pernyataan di dalamnya akan dieksekusi.
2. lvar.append((lvar[-2]) and (lvar[-1])): Ini adalah baris di dalam blok pernyataan kondisional. Mari kita bahas lebih rinci:
   a) lvar[-2] mengakses elemen kedua dari belakang dalam list lvar.
   b) lvar[-1] mengakses elemen terakhir dari list lvar.
   c) (lvar[-2]) and (lvar[-1]) adalah operasi logika AND antara dua nilai tersebut.
   d) Hasil dari operasi AND tersebut kemudian ditambahkan ke list lvar menggunakan metode append.

b. Logika OR

```
14          if operator == 'OR':
15              lvar.append((lvar[-2]) or (lvar[-1]))
```

1. if operator == 'OR':: Baris ini adalah pernyataan kondisional (if statement) yang memeriksa apakah nilai dari variabel operator sama dengan string 'OR'. Jika benar, maka blok pernyataan di dalamnya akan dieksekusi.
2. lvar.append((lvar[-2]) and (lvar[-1])): Ini adalah baris di dalam blok pernyataan kondisional. Mari kita bahas lebih rinci:
   a) lvar[-2] mengakses elemen kedua dari belakang dalam list lvar.
   b) lvar[-1] mengakses elemen terakhir dari list lvar.
   c) (lvar[-2]) and (lvar[-1]) adalah operasi logika OR antara dua nilai tersebut.
   d) Hasil dari operasi OR tersebut kemudian ditambahkan ke list lvar menggunakan metode append.

c. Logika NOT

```
    if lockop == 'unlock':
        operator = op
        lvar[-1] = int(not lvar[-1])
        envar = '(NOT[' + entry.get() +']) '
        entry.delete(0, 'end')
        entry.insert(0, envar)
        checkop = 'confsp'
        infobox.configure(text='')
    entry.configure(state='disabled')
```

1. if lockop == 'unlock':: Baris ini adalah pernyataan kondisional yang memeriksa apakah nilai dari variabel lockop sama dengan string 'unlock'. Jika benar, maka blok pernyataan di dalamnya akan dieksekusi.
2. operator = op: Variabel operator diatur menjadi nilai dari variabel op.
3. lvar[-1] = int(not lvar[-1]): Ini adalah baris yang memanipulasi nilai terakhir dalam list lvar. Nilai tersebut diubah menjadi kebalikan (negasi) dari nilai semula dengan menggunakan not, dan kemudian diubah menjadi tipe data integer menggunakan int.
4. envar = '(NOT[' + entry.get() +']) ': Variabel envar diatur dengan string baru yang berisi teks '(NOT[' + nilai input dari widget entry + '])'. Ini tampaknya digunakan untuk memodifikasi tampilan pada widget entry.
5. entry.delete(0, 'end'): Pernyataan ini menghapus semua karakter dari indeks 0 hingga akhir dari widget entry.
6. entry.insert(0, envar): Menyisipkan nilai dari variabel envar ke dalam widget entry pada posisi 0.
7. checkop = 'confsp': Variabel checkop diatur menjadi string 'confsp'.

8. infobox.configure(text=''): Mengatur teks dari suatu widget (mungkin widget bernama infobox) menjadi string kosong.
9. entry.configure(state='disabled'): Mengatur status widget entry menjadi 'disabled', sehingga tidak dapat diubah oleh pengguna.

d. Logika NOR

```
18          if operator == 'NOR':
19              lvar.append(int(not(lvar[-2] or lvar[-1])))
```

1. if operator == 'NOR':: Baris ini adalah pernyataan kondisional (if statement) yang memeriksa apakah nilai dari variabel operator sama dengan string 'NOR'. Jika benar, maka blok pernyataan di dalamnya akan dieksekusi.
2. lvar.append((lvar[-2]) and (lvar[-1])): Ini adalah baris di dalam blok pernyataan kondisional. Mari kita bahas lebih rinci:
   a) lvar[-2] mengakses elemen kedua dari belakang dalam list lvar.
   b) lvar[-1] mengakses elemen terakhir dari list lvar.
   c) (lvar[-2]) and (lvar[-1]) adalah operasi logika NOR antara dua nilai tersebut.
   d) Hasil dari operasi NOR tersebut kemudian ditambahkan ke list lvar menggunakan metode append.

e. Logika NAND

```
16          if operator == 'NAND':
17              lvar.append(int(not(lvar[-2] and lvar[-1])))
```

1. if operator == 'NAND':: Baris ini adalah pernyataan kondisional (if statement) yang memeriksa apakah nilai dari variabel operator sama dengan string 'NOR'. Jika benar, maka blok pernyataan di dalamnya akan dieksekusi.
2. lvar.append((lvar[-2]) and (lvar[-1])): Ini adalah baris di dalam blok pernyataan kondisional. Mari kita bahas lebih rinci:
   a) lvar[-2] mengakses elemen kedua dari belakang dalam list lvar.
   b) lvar[-1] mengakses elemen terakhir dari list lvar.
   c) (lvar[-2]) and (lvar[-1]) adalah operasi logika NAND antara dua nilai tersebut.
   d) Hasil dari operasi NAND tersebut kemudian ditambahkan ke list lvar menggunakan metode append.

f. Logika XOR

```
20          if operator == 'XOR':
21              lvar.append(lvar[-2] ^ lvar[-1])
```

1. if operator == 'XNOR':: Baris ini adalah pernyataan kondisional yang memeriksa apakah nilai dari variabel operator sama dengan string 'XNOR'. Jika benar, maka blok pernyataan di dalamnya akan dieksekusi.
2. lvar.append(int(not(lvar[-2] ^ lvar[-1]))): Ini adalah baris di dalam blok pernyataan kondisional. Mari kita bahas lebih rinci:
   a) lvar[-2] mengakses elemen kedua dari belakang dalam list lvar.
   b) lvar[-1] mengakses elemen terakhir dari list lvar.
   c) lvar[-2] ^ lvar[-1] adalah operasi bitwise XOR antara dua nilai tersebut.
   d) not(lvar[-2] ^ lvar[-1]) adalah negasi dari hasil operasi XOR.
   e) int(not(lvar[-2] ^ lvar[-1])) mengonversi hasil negasi ke dalam tipe data integer.
   f) Hasil dari operasi XNOR tersebut kemudian ditambahkan ke list lvar menggunakan metode append.

BAB IV
KESIMPULAN

Gerbang logika ini memberikan pemahaman tentang cara kerja gerbang logika dan fungsinya dalam sistem digital. Melalui pembelajaran ini, dapat dipelajari tabel kebenaran dari berbagai gerbang logika (termasuk AND, OR, NOT, NAND, NOR, dan XOR), sehingga mempelajari cara kerja gerbang-gerbang tersebut, dan menghasilkan output yang sesuai.

Selain itu juga memberikan pemahaman tentang penggunaan gerbang khusus seperti XOR, NAND, dan NOR. Dengan menerapkan gerbang ini pada kombinasi nilai masukan yang berbeda, dapat diamati perilakunya dan memahami bagaimana gerbang tersebut dapat digunakan dalam sistem elektronik yang lebih kompleks.

DAFTAR PUSTAKA

Hendrik, N., 2022. Gerbang Logika: Pengertian, Jenis, Fungsi, Simbol