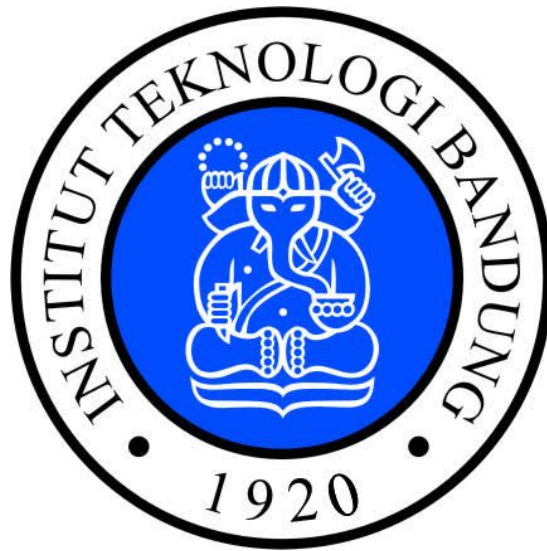


# **LAPORAN TUGAS BESAR 2 INTELIGENSI BUATAN SHAPE RECOGNITION**



Oleh:

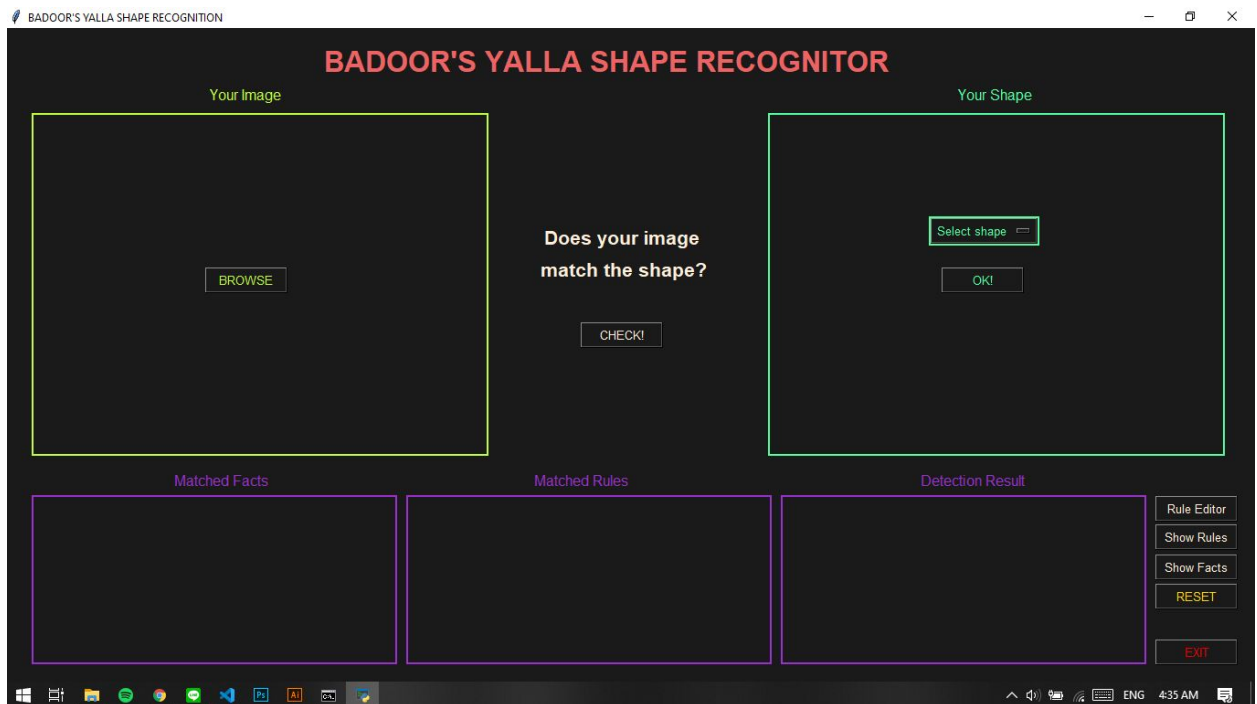
M. Raihan Asyraf D.	13517027
Ainun Fitryh Vianyryzki	13517057
Kintan Sekar Adinda	13517102
Abdurrahman Adni	13517117

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2019**

## A. Tahapan yang harus dilakukan saat membangun aplikasi.

1. Membuat *user interface* yang berisi elemen-elemen sebagai berikut.

- Image sumber
- Display image dari hasil test
- Keterangan tertulis (berhasil/gagal) atas hasil test
- Tampilan tertulis atas fakta yang terpilih (*match*) pada saat test
- Rules apa yang terpilih/digunakan saat proses deteksi.
- *Tree list of graphics* atas *Rules* yang digunakan
- *Select-button* untuk *upload original image*
- *Open-button* untuk *rule editor*
- *Button* yang menunjukkan keseluruhan *rules*
- *Button* yang menunjukkan fakta tentang *source-image*
- *User's main interface* dapat dilihat sebagai berikut:



Dalam membuat *user interface*, digunakan tkinter python.

2. Membuat *inference engine*.

*Inference engine* berguna untuk memproses fakta dan *rule* yang ada sehingga nantinya akan menghasilkan fakta-fakta baru yang akan dicocokkan dengan bentuk pilihan dari *user*. *Inference engine* dapat menggunakan 2 metode, yaitu DFS dan BFS dalam mengimplementasikan *Knowledge Based System*.

*Inference engine* dibuat dengan menggunakan python.

3. Membuat *rule of representation*, yaitu kumpulan rule yang nantinya akan dicocokkan dengan fakta-fakta yang ada untuk menghasilkan fakta baru yang pada akhirnya akan menghasilkan kesimpulan berupa jenis bentuk. Rule dibuat menggunakan python. Rule yang ada dibuat berdasarkan bentuk dasar yang harus dapat dideteksi oleh sistem sebagai berikut.

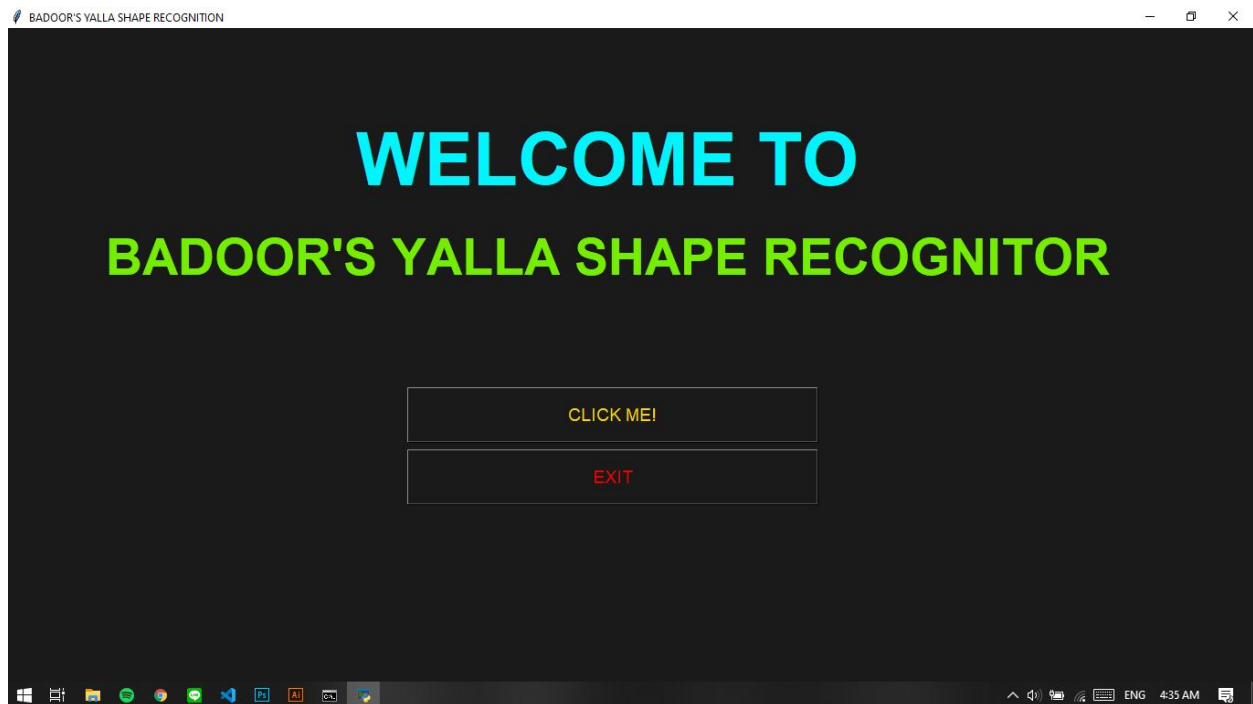
- Segitiga tidak beraturan:
    - a. Segitiga lancip
    - b. Segitiga tumpul
    - c. Segitiga siku-siku
    - d. Segitiga sama kaki
      - i. Segitiga sama kaki dan siku-siku
      - ii. Segitiga sama kaki dan tumpul
      - iii. Segitiga sama kaki dan lancip
    - e. Segitiga sama sisi
  - Segiempat tidak beraturan
    - a. Jajaran Genjang
      - i. Segiempat beraturan
      - ii. Segiempat berbentuk layang-layang
    - b. Trapezium
      - i. Trapezium sama kaki
      - ii. Trapezium rata kanan
      - iii. Trapezium rata kiri
  - Segi lima tidak beraturan
    - a. Segi lima sama sisi
  - Segi enam tidak beraturan
    - a. Segi enam sama sisi
4. Membuat *image processing* menggunakan *library* yang terdapat pada Open CV. *Image processing* berguna untuk memproses gambar yang diinput ke dalam sistem agar menghasilkan fakta-fakta yang nantinya akan dicocokkan oleh *inference engine*. Misalnya, jika dimasukkan gambar serupa segitiga, maka *image processing* akan menghasilkan fakta berupa sisi=3 dan sudut=3.

## B. Repository yang berisi dokumentasi lengkap

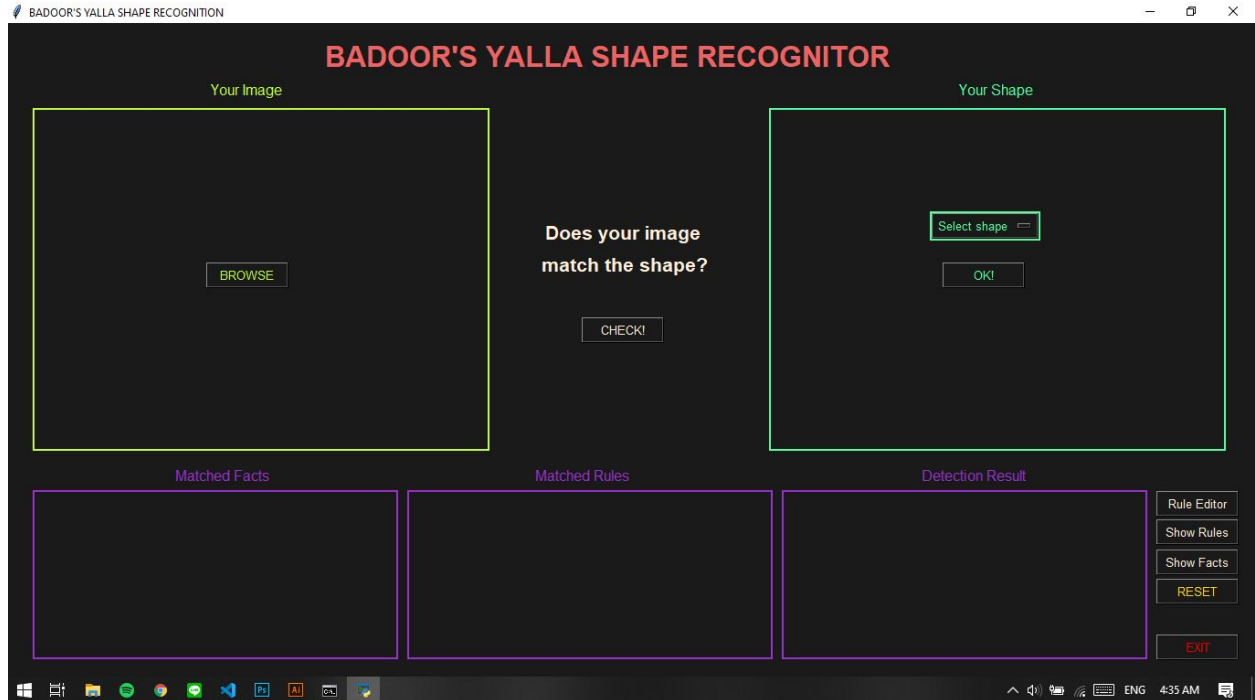
<https://github.com/AbdurrahmanAdni/Shape-Recognition>

## C. User manual

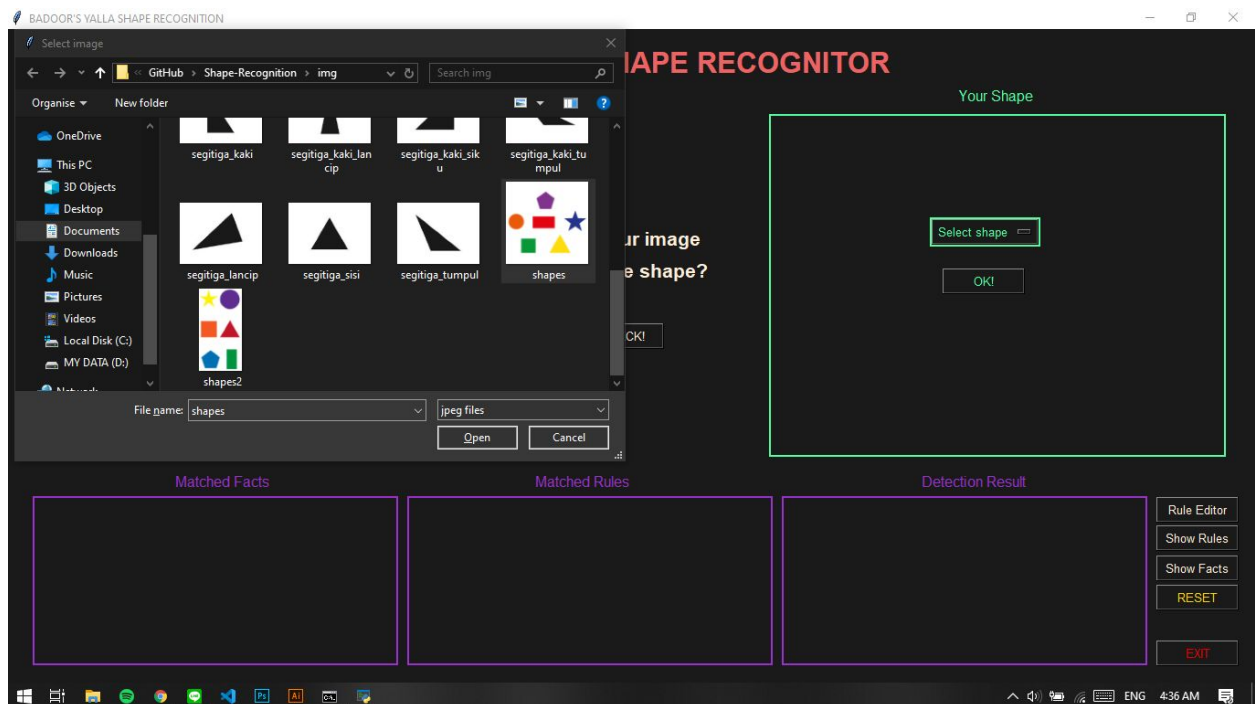
- Instalasi:
  1. Untuk memudahkan, gunakan sistem operasi Ubuntu.
  2. Install Python3
  3. Install opencv untuk digunakan sebagai image processing
- Menjalankan program:
  1. Buka terminal
  2. Pindah direktori ke dalam folder Shape-Recognition
  3. Jalankan GUI program dengan perintah "python GUI.py"



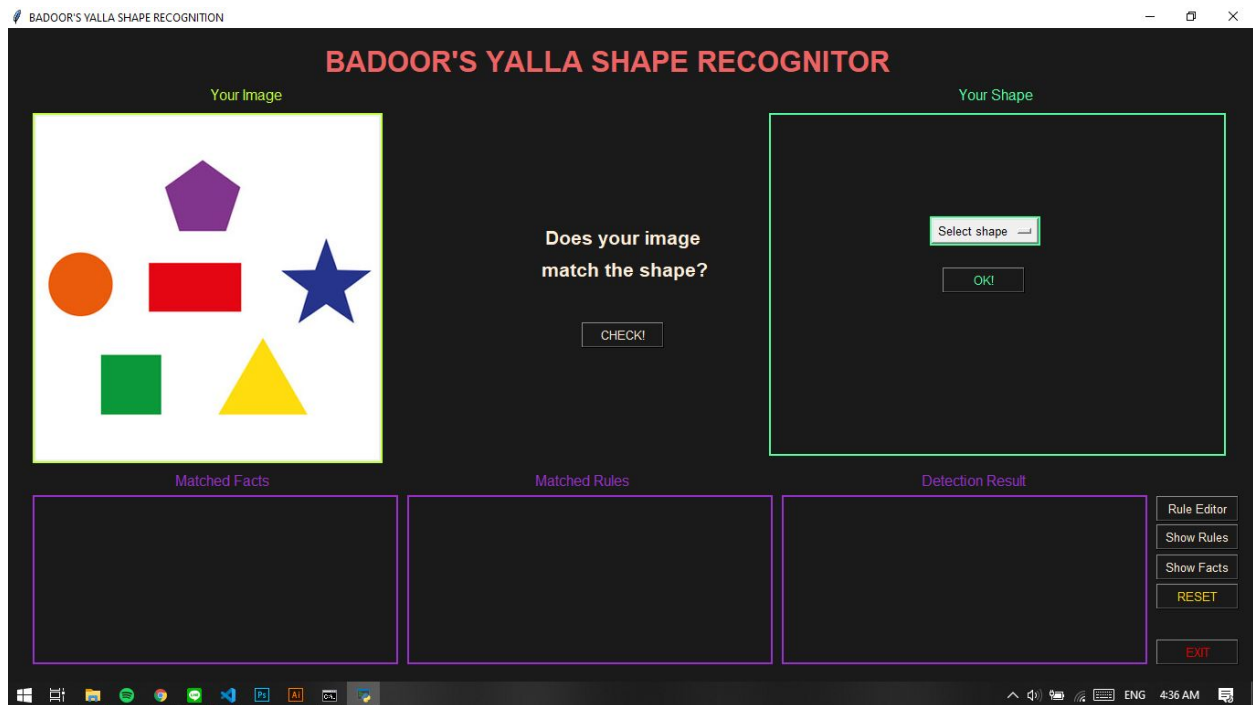
4. Pada *welcome page*, terdapat 2 *button*, yaitu CLICK ME! Dan EXIT. *Button* CLICK ME! digunakan untuk masuk ke *main page*. *Button* EXIT digunakan untuk keluar dari program.



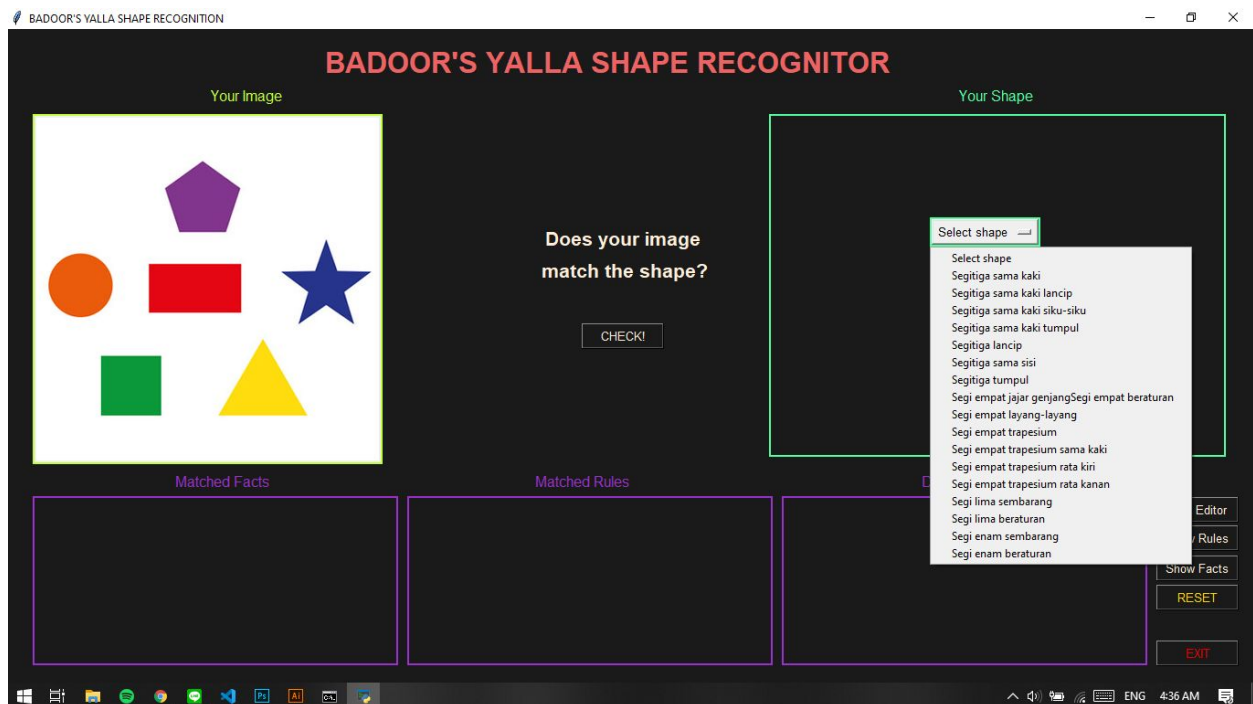
5. Pada *main page*, terdapat 2 *input field*, yaitu Your Image dan Your Shape. Selain itu, terdapat *button* CHECK untuk menjalankan *backend*. Di bagian bawah program, terdapat 3 *field* untuk menampilkan Matched Facts, Matched Rules, dan Detection Result. Di bagian kanan bawah terdapat 5 *button*, yaitu Rule Editor, Show Rules, Show Facts, RESET, dan EXIT.



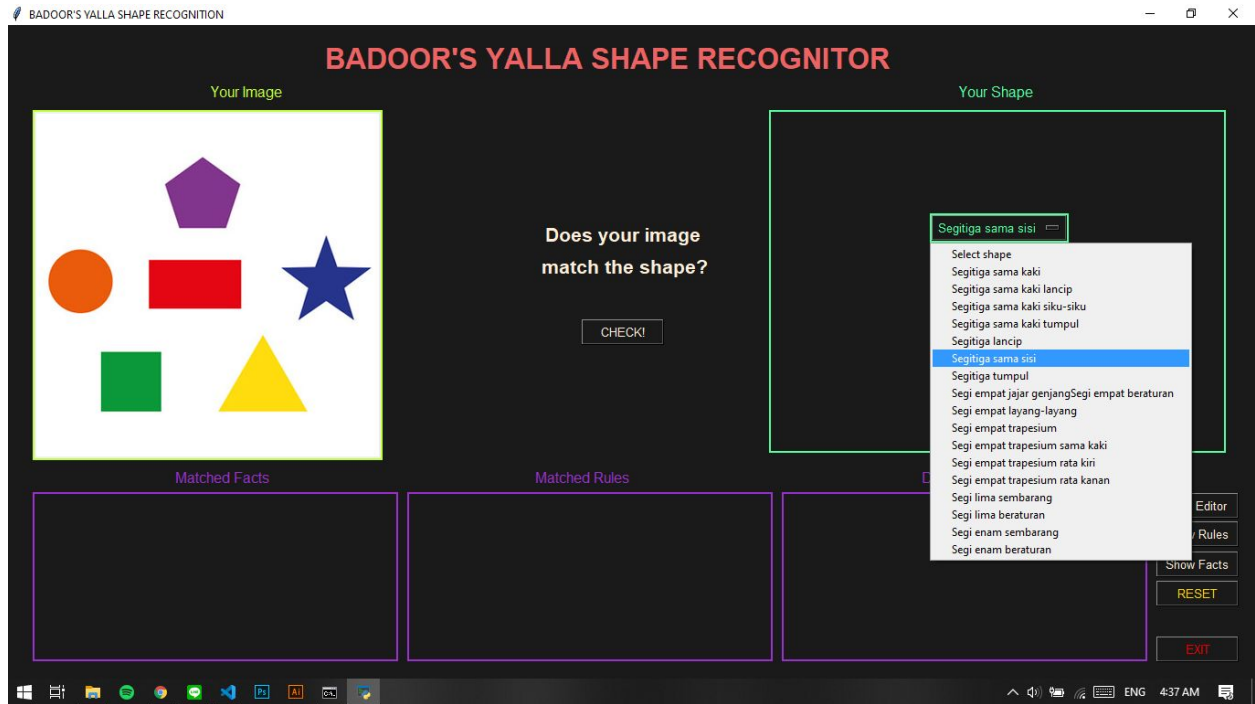
6. Jika kita mengklik *button* BROWSE pada *input field* Your Image, maka akan muncul *file manager* untuk *select* gambar yang ingin dikenali. Klik gambar yang ingin dipilih, kemudian klik *open*.



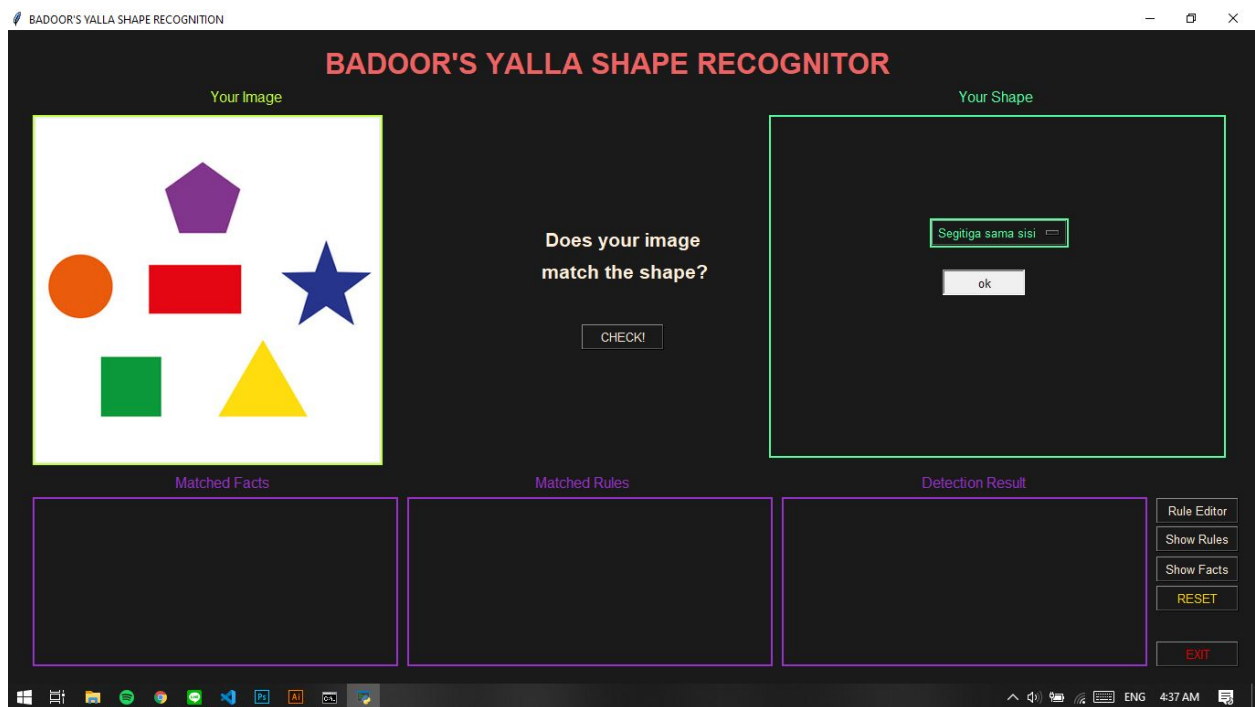
7. Gambar yang dipilih akan muncul pada *field* Your Image.



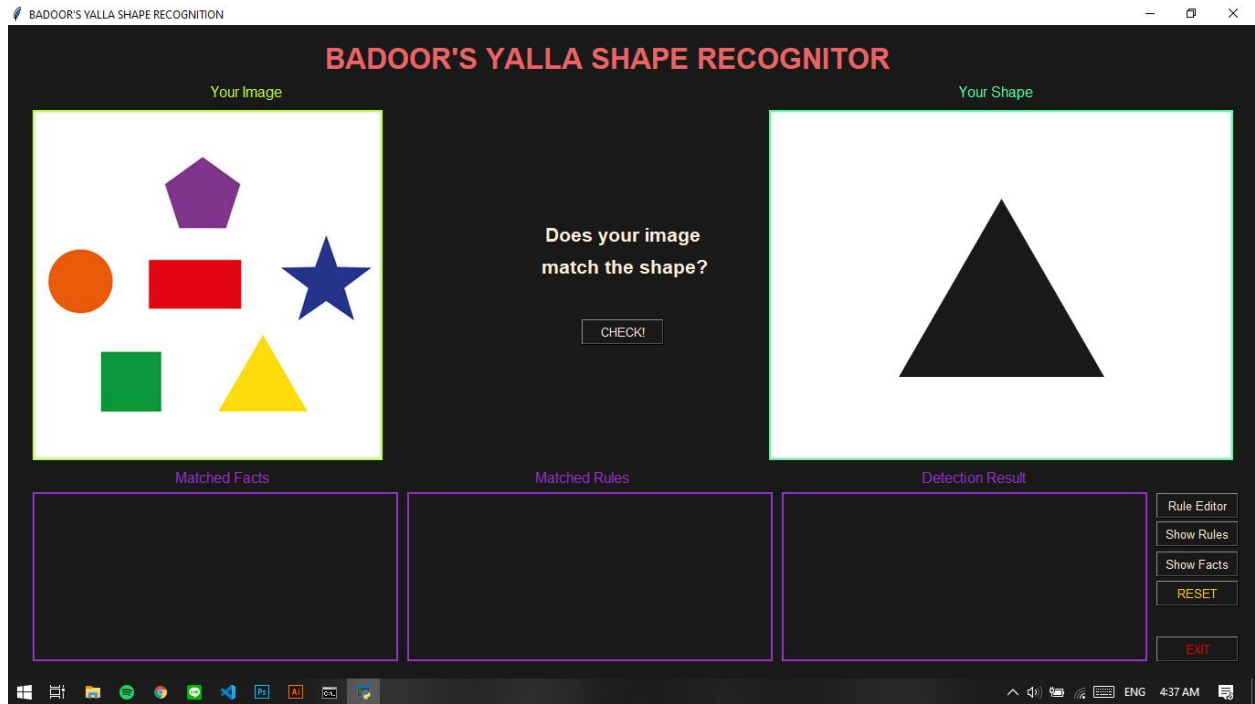
8. Jika kita mengklik button Select shape pada input field Your Shape, maka akan muncul berbagai macam bentuk dasar yang dapat dipilih untuk dicocokkan dengan image.



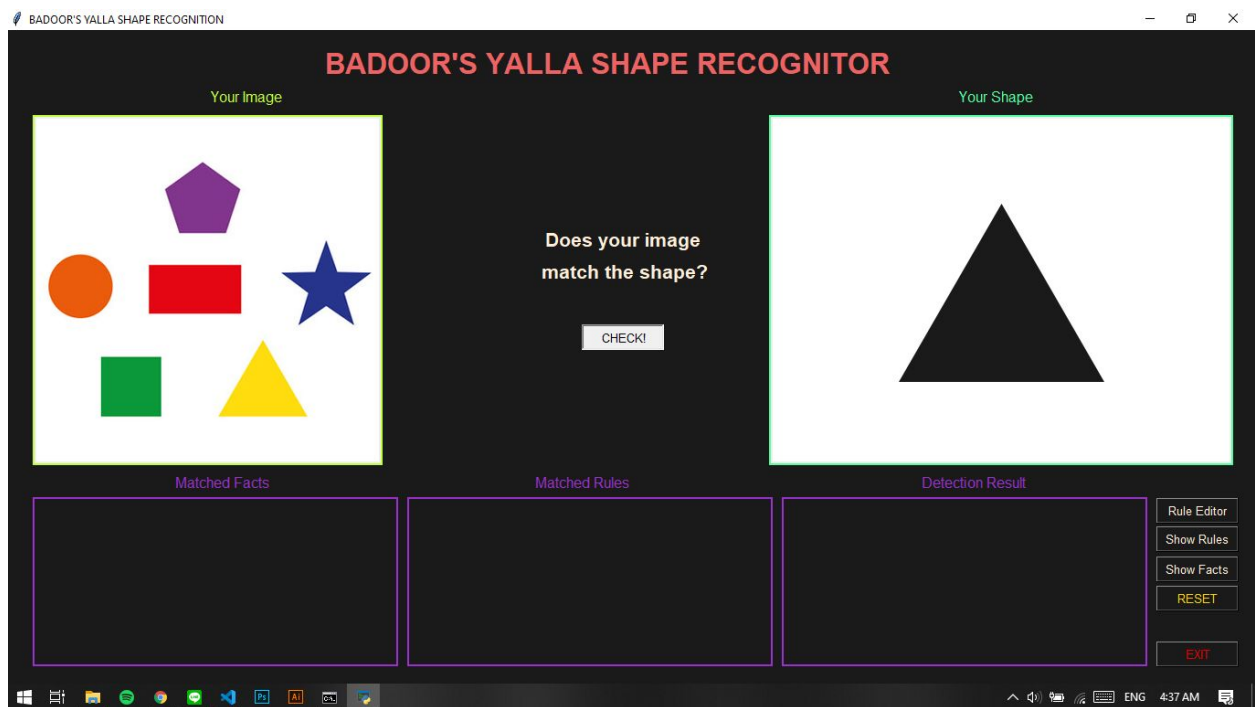
9. Klik salah satu bentuk dasar yang terdapat pada pilihan.



10. Klik button ok.

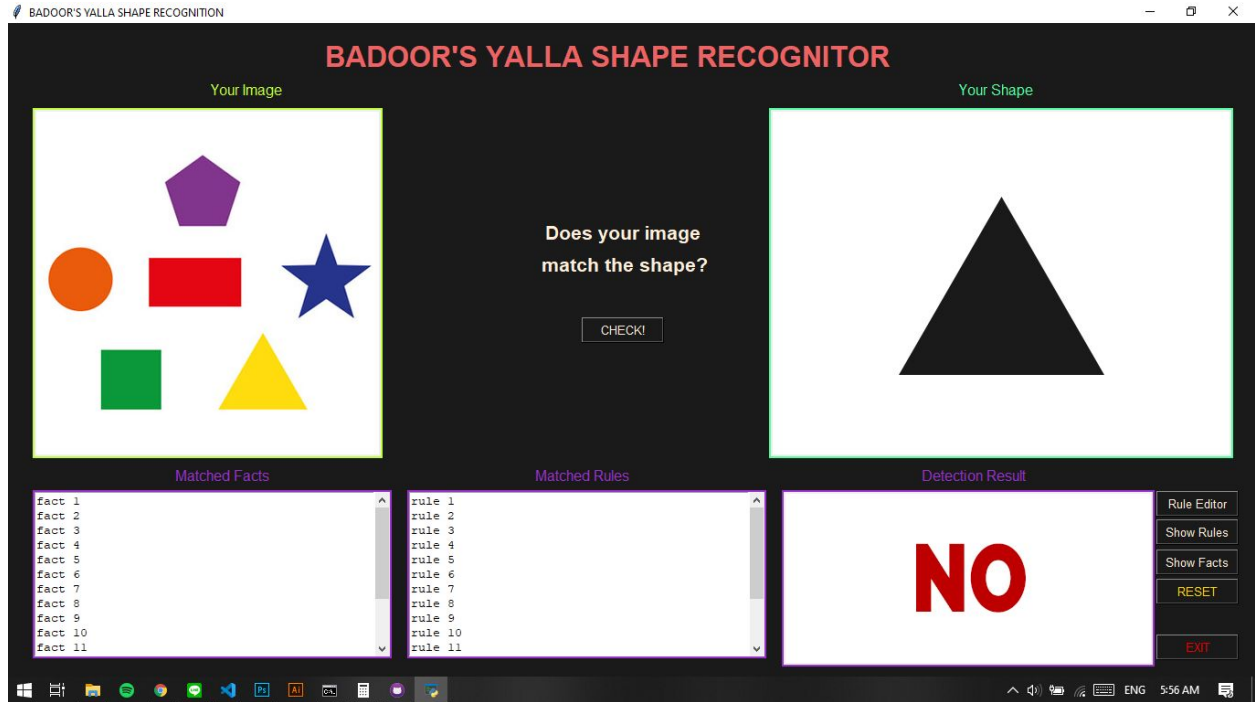


11. Shape yang dipilih akan muncul pada field Your Shape.

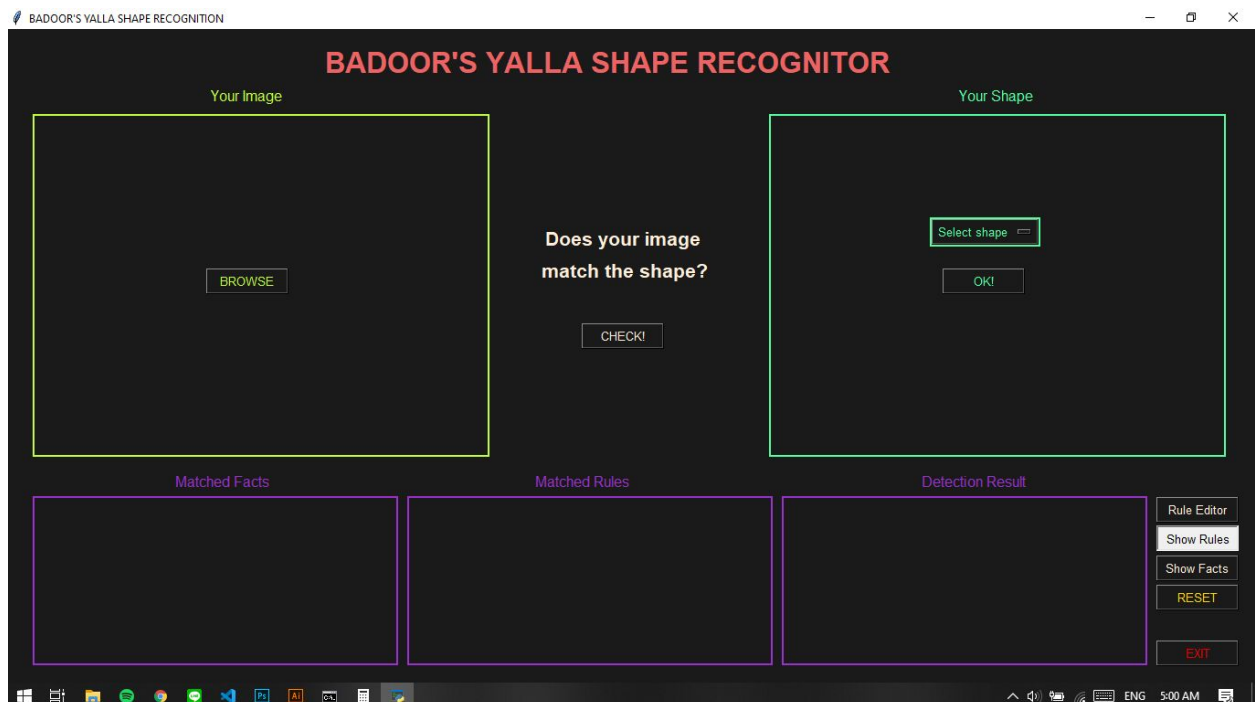


12. Klik button CHECK! di tengah program.

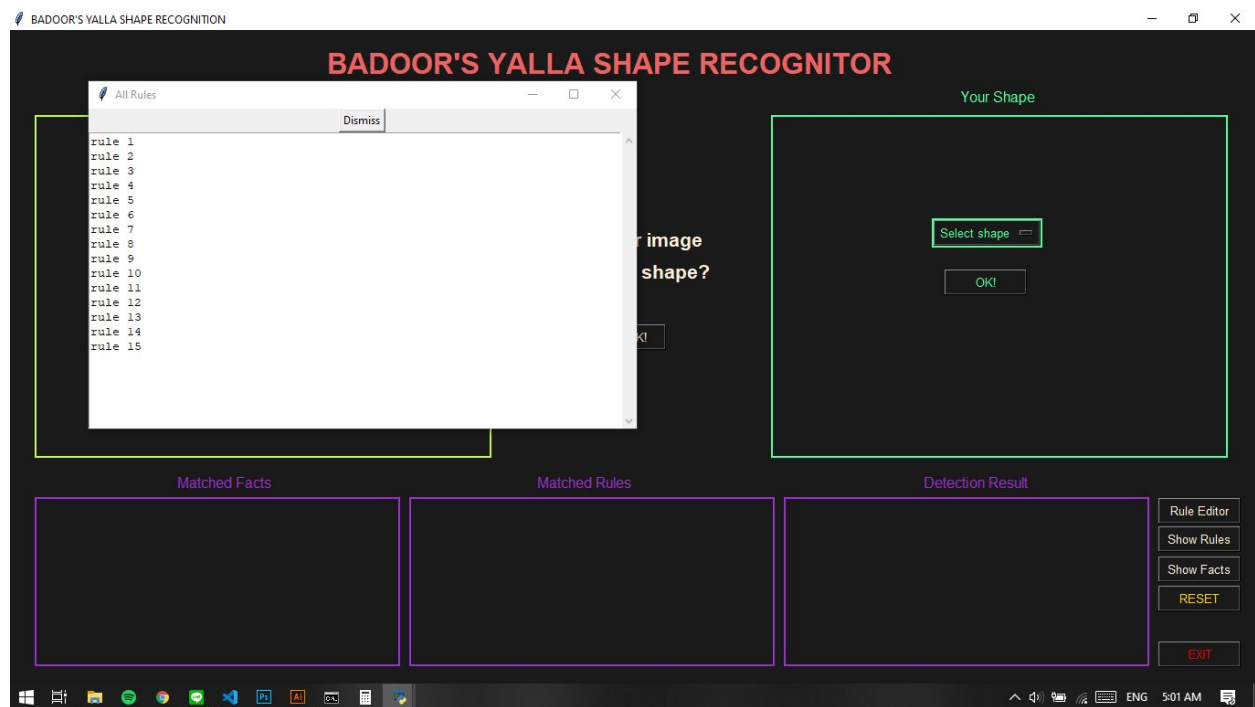




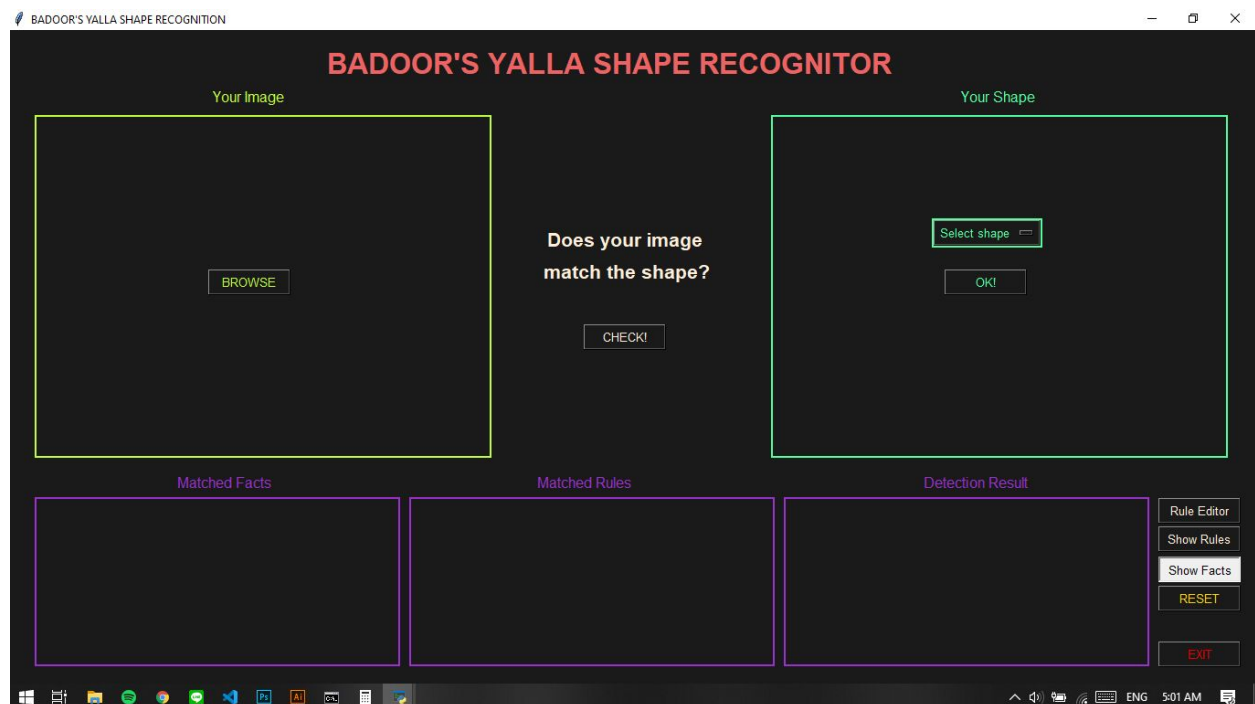
13. *Backend* akan berjalan dan hasilnya ditampilkan pada *field* Matched Facts, Matched Rules, Detection Result. Matched Facts berisi fakta-fakta yang cocok dengan *rules* yang ada. Matched Rules berisi kumpulan *rules* yang cocok dengan fakta yang dihasilkan oleh *image*. Detection Result berisi YES atau NO, YES jika *shape* ditemukan dalam *image* dan NO jika *shape* tidak ditemukan.



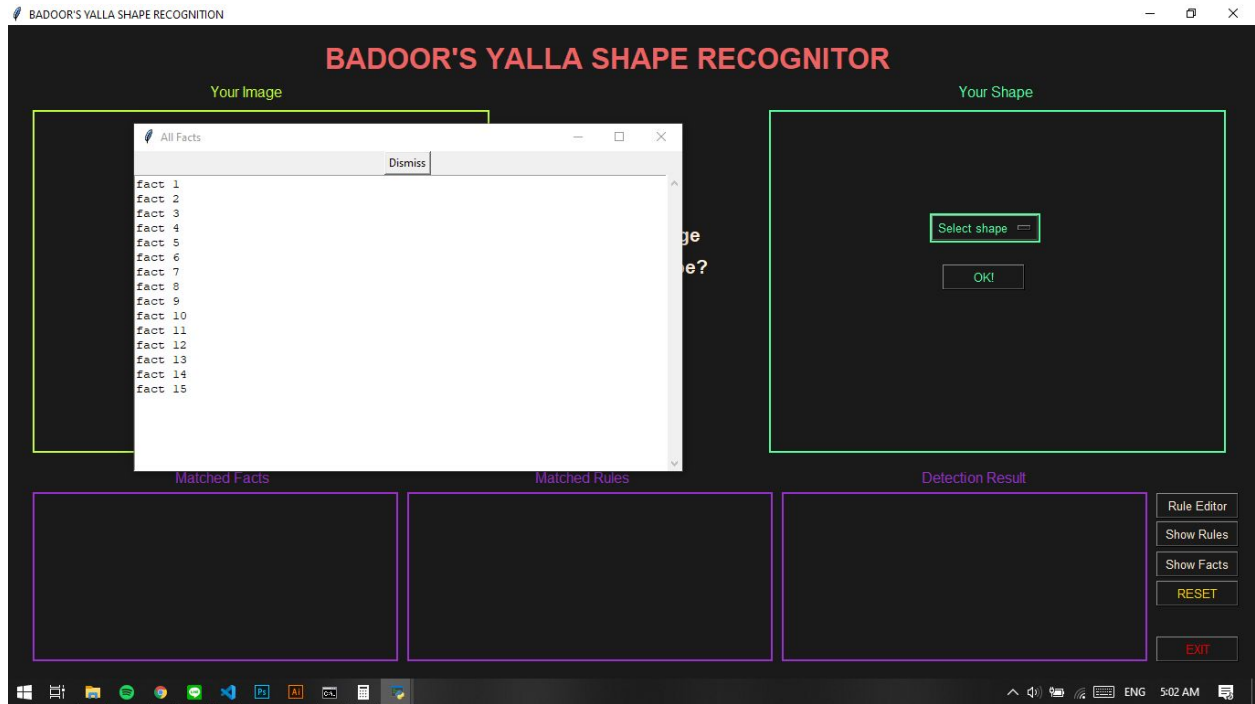
14. Terdapat beberapa fitur lain, salah satunya Show Rules.



15. Jika kita klik *button* Show Rules, akan muncul *pop-up window* yang menampilkan kumpulan *rules* yang dimiliki oleh sistem.



16. Fitur lainnya adalah Show Facts.



17. Jika kita klik *button* Show Facts, akan muncul *pop-up window* yang menampilkan kumpulan fakta yang dimiliki oleh *image* yang telah diinput oleh *user*.

**D. Proses updating dan inferencing atas fakta yang terlibat**

Dalam menjalankan inferencing, sistem dapat menggunakan metode BFS dan DFS. Jika rule pertama yang terdapat dalam queue menghasilkan fakta baru, maka kumpulan fakta yang ada akan di-update dengan menambahkan fakta baru tersebut. Kemudian, rule tersebut akan dibuang dari queue. Jika fakta baru tersebut ketika ditelusuri lagi menghasilkan rule baru, maka tambahkan rule baru tersebut ke dalam queue. Dalam metode DFS, rule dimasukkan ke awal queue, sementara dalam metode BFS, rule baru ditambahkan ke akhir queue.