

**TUGAS BESAR 2**  
**IF3170 - Inteligensi Buatan**

**Deteksi Bentuk Geometri Berdasarkan *Knowledge Based System***



Oleh :

Pandyaka Aptanagi	/ 13517003
I Putu Gede Wirasuta	/ 13517015
M. Rifky I. Bariansyah	/ 13517081
Gardahadi	/ 13517144

# BAB I

## Tahapan dalam Pembangunan Aplikasi

Aplikasi yang kami bangun terdiri atas *image processing application*, *rule-based system*, dan *graphical user interface*

### 1.1 Tahapan Pembangunan *Image Processing Application*

Aplikasi menerima informasi mengenai gambar melalui *Image Processing Application*. Aplikasi ini dibangun menggunakan library image processing OpenCV. Berikut tahapan pembangunannya:

1. Aplikasi menerima image menggunakan `cv2.imread()`
2. Aplikasi melakukan deteksi sisi menggunakan `cv2.Canny()` yang disimpan pada variabel `edge`
3. Dengan variabel `edge` ini kemudian akan dipanggil `cv2.HoughLinesP()` untuk mendapatkan semua garis probabilistik dari gambar
4. Dengan menggunakan metode tersebut akan dihasilkan garis probabilistik luar dan dalam untuk sebuah garis pada gambar. Maka dari itu aplikasi akan mem-filter kedua garis tersebut menjadi satu garis dengan fungsi bantuan `similarLine()` dengan parameter `threshold` klasifikasi garis yang dianggap sama
5. Aplikasi menggunakan fungsi bantuan `countGradient()` untuk menghitung gradien
6. Aplikasi menggunakan fungsi bantuan `adjacentLine()` untuk mengembalikan semua garis yang bertemu pada sebuah titik
7. Aplikasi kemudian mengembalikan garis, gradien, dan garis - garis yang bertemu berdasarkan masukan gambar.

Informasi ini kemudian akan menjadi masukan bagi *rule-based system*.

### 1.2 Tahapan Pembangunan *Rule-Based System*

Rule-based sistem pada aplikasi ini dibangun menggunakan wrapper **clipsy**. Untuk membangun rule-based system, diperlukan komponen *rules* dan fakta.

Fakta diimplementasikan dengan menggunakan `(assert <fact>+)` yang berfungsi untuk menambahkan fakta kedalam *facts-list*. *Rules* diimplementasikan dengan menggunakan `defrule`. `defrule` memiliki struktur seperti dibawah ini:

```
(defrule <rule-name> [<optional comment>]
  <patterns>*
  =>
  <actions>* )
```

`defrule` digunakan untuk menyimpan pola sifat suatu bangunan. Misalnya pada implementasi `check_triangle`, rule yang menentukan apakah masukan merupakan segitiga atau bukan,

```
(defrule check_triangle
  (adjacent 3)
  =>
  (assert (triangle))
);
```

Melakukan pengecekan apabila terdapat fakta (`adjacent 3`) yang artinya terdapat tiga sisi yang saling berhubungan. Bila ya, maka akan ditambahkan fakta dengan (`assert(triangle)`) ke facts-list.

## 1.2 Tahapan Pembangunan *Graphical User Interface*

Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman **python 3** dengan memanfaatkan library **tkinter** dan **PIL**.

1. Aplikasi terdiri dari beberapa **frame** yang disediakan oleh *tkinter*. Frame-frame tersebut berguna sebagai *container* dari komponen-komponen yang ada di *graphical user interface* ini.
2. Untuk *frame* pada bagian *source image*, *detection image*, *detection result*, *matched facts*, dan *hit rules* akan diisi dengan komponen **canvas**. Komponen ini dapat memunculkan gambar atau teks sesuai kebutuhan.
3. Untuk *frame* pada bagian *menu button*, terdiri dari beberapa komponen *button* atau tombol, yang ketika dipilih akan men-*trigger function* sesuai dengan fungsi dari tombol-tombol tersebut. Komponen ini dibuat dengan menggunakan **button**.
4. Untuk *frame* pada bagian *menu shapes*, terdiri dari komponen **treeview**. Komponen ini akan membentuk sebuah 'pohon' dari pilihan-pilihan yang ada. Jika salah satu menu di-klik, akan memunculkan bentuk *shape* pada frame *detection image*.
5. *GUI* akan menerima masukan *image* dari pengguna, kemudian mengirimkan *image* tersebut untuk dilakukan ke tahap selanjutnya (*image processing* dan *detection*)

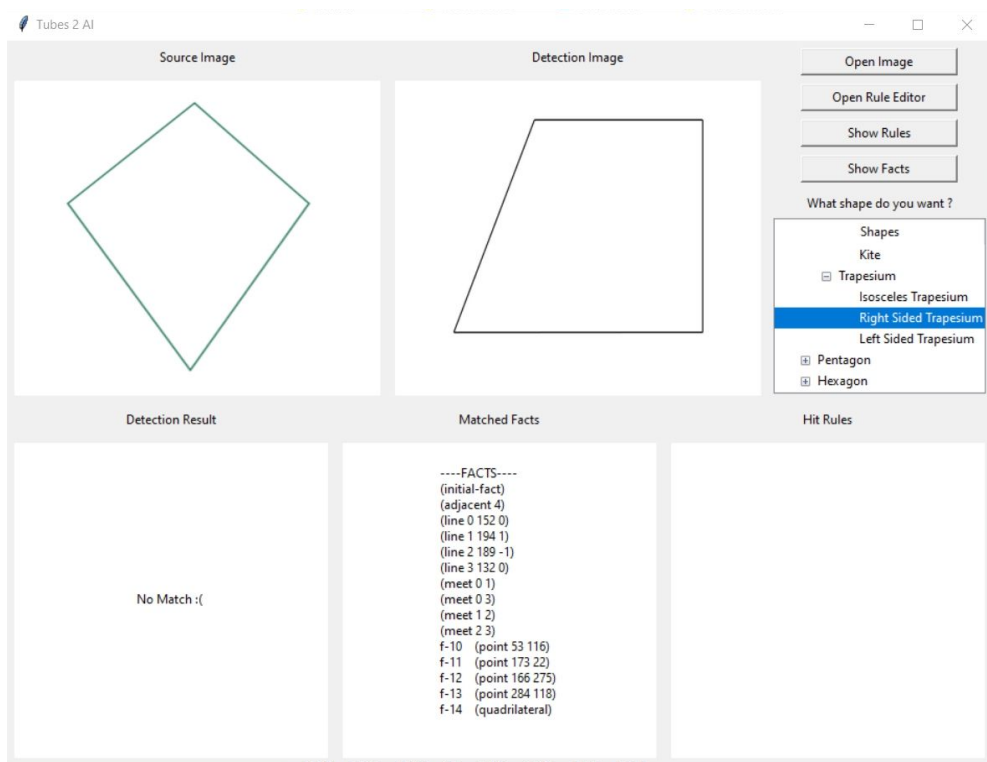
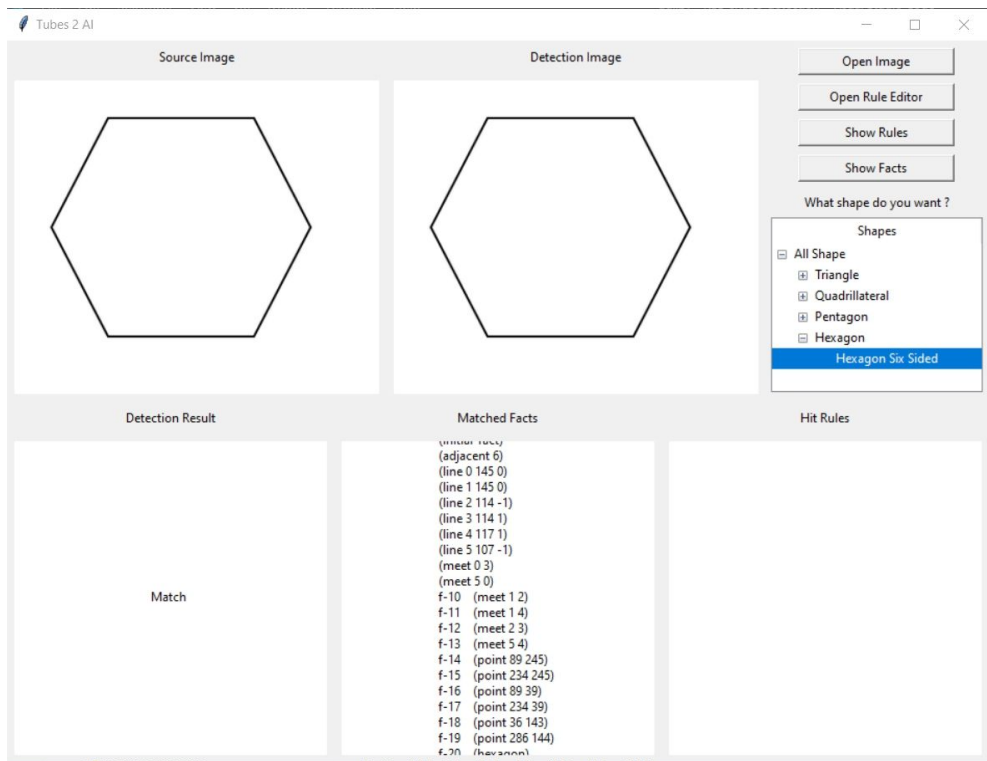
## BAB II

### Dokumentasi dan *Repository* Aplikasi

#### 2.1 Dokumentasi Aplikasi

Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman **python 3** dengan memanfaatkan beberapa *library* utama yaitu **tkinter** untuk pengembangan *graphical user interface (GUI)*, **PIL** untuk merekayasa gambar, **opencv** untuk mencari *facts* dan *rule* dari sebuah gambar, dan **clipsy**.

*File* utama dari aplikasi ini adalah *file* **main.py** yang akan menampilkan *graphical user interface* dari aplikasi ini. Pada *GUI* terdapat beberapa pilihan bentuk geometri yang ingin dideteksi pada sebuah gambar. Hasil dari deteksi bentuk akan ditampilkan hasilnya pada kotak *detection result*, *matched facts*, dan *hit rules*.



## 2.2 *Repository* Aplikasi

Source code aplikasi ini tersimpan pada *repository* **GitHub** yang ada pada:

[\*https://github.com/pandyakaa/KBS-shape-detection\*](https://github.com/pandyakaa/KBS-shape-detection)

## BAB III

### *User Manual*

1. Lakukan *clone* terhadap *repository* aplikasi

```
git clone https://github.com/pandyakaa/KBS-shape-detection
```

2. *Install requirements* yang dibutuhkan untuk menjalankan aplikasi

```
pip install requirements.txt
```

3. Jalankan *file main.py* untuk menggunakan aplikasi

```
python main.py
```

4. Akan muncul tampilan *graphical user interface* dari aplikasi
  - a. Untuk memasukkan gambar yang ingin dideteksi, pilih tombol **Open Image**
  - b. Untuk memasukkan *shape* sebagai pendeteksi, pilih melalui kotak menu **Shapes**
  - c. Untuk memunculkan *Rule Editor*, pilih tombol **Open Rule Editor**
  - d. Untuk memunculkan daftar *rules*, pilih tombol **Show Rules**
  - e. Untuk memunculkan daftar *facts*, pilih tombol **Show Facts**
5. Setelah memilih gambar yang ingin dideteksi dan *shape* pendeteksi, maka aplikasi akan secara otomatis memunculkan hasil dari deteksi

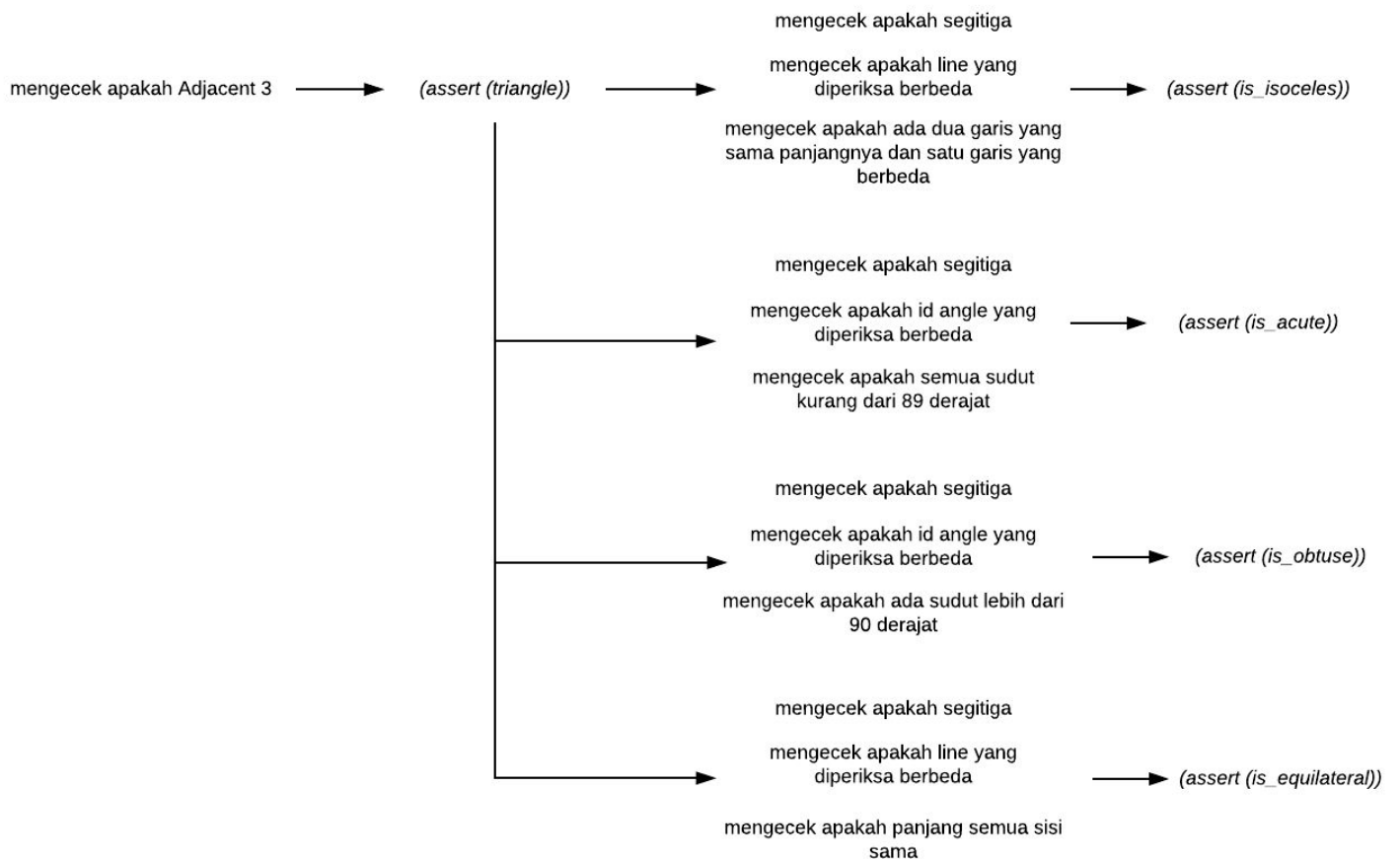
## BAB IV

### Proses *Updating* dan *Inferencing* pada Fakta yang Terlibat

Inference Engine mengontrol jalannya eksekusi secara keseluruhan. Inference Engine mencocokkan semua fakta dengan rules yang ada menggunakan cara kerja **recognize-act cycle**, yaitu

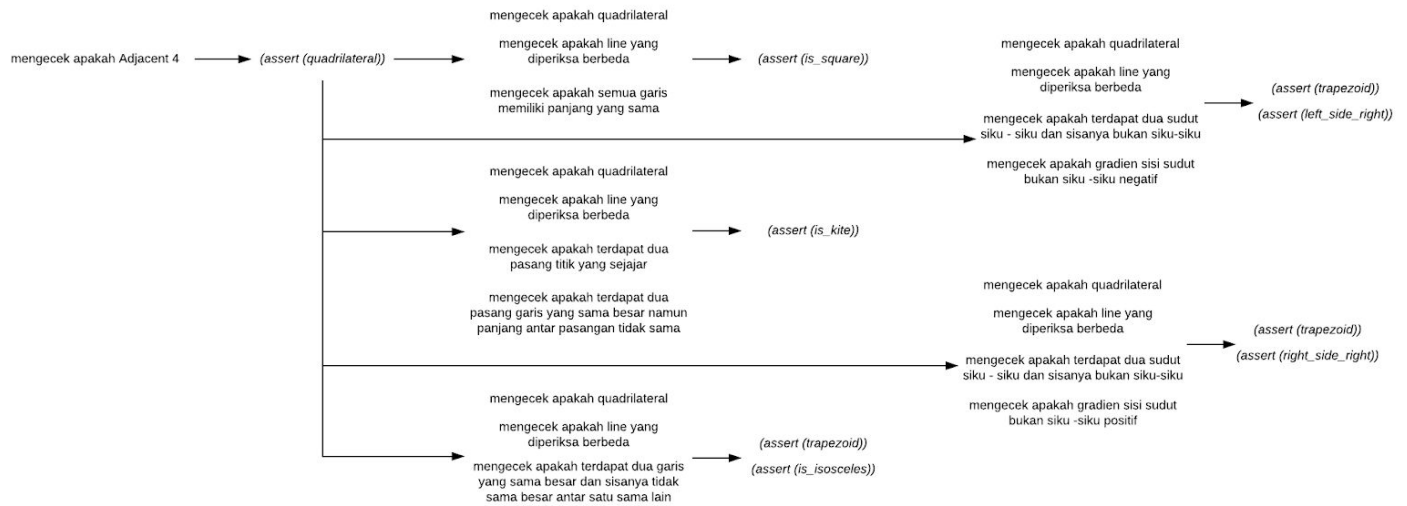
1. Cocokkan facts dengan rules yang ada
2. Pilih instansiasi rules yang akan dijalankan
3. Eksekusi aksi yang terkait dengan rule terpilih

#### a. Updating dan inferencing pada kelompok segi tiga tak beraturan

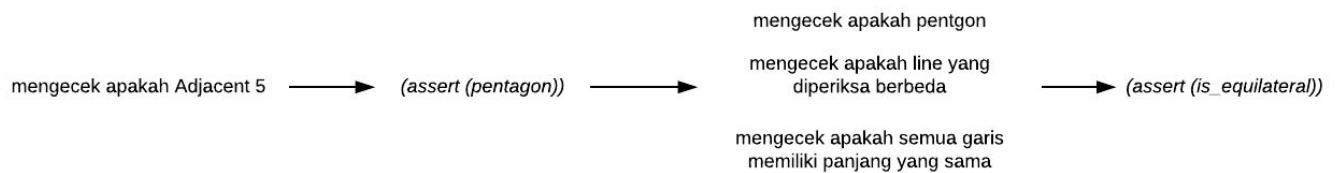




## b. Updating dan inferencing pada kelompok segi empat tak beraturan



## c. Updating dan inferencing pada kelompok segi lima tak beraturan



## d. Updating dan inferencing pada kelompok segi enam tak beraturan

