

**BUKU PEDOMAN PENGGUNAAN SISTEM *FISH MOVEMENT*  
*TRACKING***

## DAFTAR ISI

1. PENDAHULUAN.....	3
1.1. Tujuan Pembuatan Dokumen.....	3
1.2. Deskripsi Umum Sistem .....	3
1.3. <i>Source Code</i> .....	3
2. STRUKTUR FILE .....	3
2.1. Gambaran Umum.....	3
2.2. <i>/datasets</i> .....	3
2.3. <i>/groundtruth</i> .....	4
2.4. <i>/screenshots</i> .....	4
2.5. <i>contour_tracing.py</i> .....	4
2.6. <i>detector.py</i> .....	4
2.7. <i>evaluator.py</i> .....	4
2.8. <i>kalman_filter.py</i> .....	4
2.9. <i>tracker.py</i> .....	4
2.10. <i>main.py</i> .....	5
3. PETUNJUK PENGGUNAAN .....	5
3.1. <i>Datasets</i> .....	5
3.2. <i>Command Args main.py</i> .....	5
3.3. Contoh Penggunaan <i>Command main.py</i> .....	5
3.4. <i>Command Args evaluator.py</i> .....	6
3.5. Contoh Penggunaan <i>Command evaluator.py</i> .....	6

# 1. PENDAHULUAN

## 1.1. Tujuan Pembuatan Dokumen

Dokumen ini dibuat agar pengguna dapat mengoperasikan dan menjalankan sistem *fish movement tracking* dengan maksimal.

## 1.2. Deskripsi Umum Sistem

Sistem *fish movement tracking* adalah sistem berbasis *command-line* yang dibuat dengan bahasa pemrograman python. Sistem ini bertujuan untuk menghitung rata-rata jumlah ikan dalam sebuah video. Untuk menjalankan sistem, pengguna perlu meng-install python versi 3 (tiga) terlebih dahulu, serta menyiapkan dataset berupa video yang ingin dilacak objeknya.

## 1.3. Source Code

Seluruh *source code* sistem *fish movement tracking* dapat dilihat melalui link <https://github.com/cah-kangkung/object-tracking>.

# 2. STRUKTUR FILE

## 2.1. Gambaran Umum

- */datasets*
- */groundtruth*
- */screenshots*
- *contour\_tracing.py*
- *detector.py*
- *evaluator.py*
- *generate\_result.sh.example*
- *kalman\_filter.py*
- *main.py*
- *tracker.py*

## 2.2. */datasets*

Folder ini berfungsi untuk menyimpan datasets yang akan digunakan dan dibaca oleh sistem. Nama folder dapat berupa apapun.

### 2.3. */groundtruth*

Folder ini berisikan *groundtruth image* yang digunakan untuk mengukur performa GMM dan Operasi Morfologi.

### 2.4. */screenshots*

Folder */screenshots* akan di-generate otomatis oleh sistem. Folder ini berisikan *snapshot frame image* yang merupakan keluaran dari masing-masing metode. Parameter *-ss* digunakan untuk mendefinisikan *frame* mana saja yang akan disimpan.

### 2.5. *contour\_tracing.py*

File *contour\_tracing.py* adalah file *class* yang berisikan fungsi-fungsi python untuk melakukan operasi *contour tracing / border following*. Fungsi *findContourCustom(self, img)* adalah fungsi utama yang akan dipanggil oleh fungsi lain yang membutuhkannya.

### 2.6. *detector.py*

File *detector.py* bertugas untuk melakukan operasi objek deteksi secara keseluruhan. Class ini menghasilkan koordinat titik tengah, *contours*, dan *foreground mask* dari objek yang berhasil dideteksi. Fungsi *detect(self, frame)* adalah fungsi utama *class* ini.

### 2.7. *evaluator.py*

Bertugas untuk mengevaluasi performa GMM dan Operasi morfologi menggunakan *metric accuracy, precision, recall, dan f1 score*.

### 2.8. *kalman\_filter.py*

Berisikan fungsi fungsi untuk memprediksi *track* masing-masing object yang berhasil dideteksi.

### 2.9. *tracker.py*

File *tracker.py* adalah file yang bertugas untuk melakukan operasi *tracking* secara keseluruhan. Class Tracker menyimpan track masing-masing objek yang nantinya akan di-update setiap frame oleh Kalman Filter. Terdapat juga fungsi untuk menghitung *RMSE* track yang diprediksi oleh KF dan juga fungsi untuk menghitung rata-rata jumlah objek.

### 2.10. *main.py*

File yang menyatukan seluruh class dan fungsi yang dibutuhkan.

## 3. PETUNJUK PENGGUNAAN

### 3.1. *Datasets*

Letakan data pada folder */datasets*. Sistem ini menggunakan video sebagai data masuknya. Format yang sudah diuji dan berhasil berjalan antara lain adalah .mp4, .avi, dan .flv.

### 3.2. *Command Args main.py*

*Command Args* atau parameter yang digunakan oleh *main.py* antara lain adalah:

- -p = Lokasi sumber video
- -n = Nama video
- -s = Ukuran skala *downsampling*
- -kes = Ukuran *structuring element* operasi erosi
- -ked = Ukuran *structuring element* operasi dilasi
- -ss = *Comma-seperated value*, berisikan frame yang akan disimpan.

### 3.3. Contoh Penggunaan *Command main.py*

```
python main.py -p datasets/filtered_input_video/9866-compressed.mp4 -n 9866 -s 8 -kes 5 -kds 11 -ss 509,789,849
```

Selain menuliskan *command* langsung diterminal, kita dapat menggunakan file *generate\_result.sh.example* untuk menjalankan *multiple command* secara bersamaan. Salin file *generate\_result.sh.example* dan ubah namanya menjadi *generate\_result.sh*. Kemudian masukkan *command* yang diinginkan dipisahkan dengan *line* baru, dan jalankan file bash tersebut dengan *command*

### 3.4. Command Args evaluator.py

*Command Args* atau parameter yang digunakan oleh `main.py` antara lain adalah:

- `-name` = Nama file
- `-gtp` = Lokasi *Groundtruth*
- `-pp` = Lokasi *Predicted*

Lokasi *Predicted* adalah lokasi file yang ingin dibandingkan dengan *groundtruth* untuk menghasilkan skor *metric*, *accuracy*, *recall*, *precision*, dan *f1 score*.

### 3.5. Contoh Penggunaan *Command evaluator.py*

```
python evaluator.py -name 9908 -gtp groundtruth/9908.jpg -pp  
screentshots/9908.jpg
```