

**LAPORAN  
PENELITIAN MANDIRI**



**Judul :  
PEMBUATAN APLIKASI PENDETEKSI TINGGI MUKA AIR  
MENGUNAKAN OCR**

**Ketua Tim Penyusun :  
WINARNO SUGENG    0420106301**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL  
BANDUNG  
2023**

## LEMBAR PENGESAHAN

Judul Penelitian : PEMBUATAN APLIKASI PENDETEKSI TINGGI MUKA AIR MENGGUNAKAN OCR

Nama Rumpun Ilmu : Informatika

Biaya Penelitian yang dikeluarkan : Rp. 650.000,-

**Ketua Peneliti** :

a. Nama Lengkap : Winarno Sugeng, Dr., Ir., M.Kom.  
b. NIDN : 0420106301  
c. Jabatan Fungsional : Lektor Kepala / IV-A  
d. Program Studi : Informatika  
e. Nomor HP : 0812-2020-9336  
f. Alamat surel (e-mail) : [winarno@itenas.ac.id](mailto:winarno@itenas.ac.id)

### Anggota

a. Nama Lengkap : Mochamad Hafied Ibni Anwar.  
b. NIM : 152019129  
c. Program Studi : Informatika

Mengetahui,  
Ketua Prodi Informatika



(Lisa Kristiana, Dr.sc., MT)  
NIDN : 0425107503

Bandung, 17 July 2023  
Ketua tim penyusun



(Winarno Sugeng, Dr., Ir., M.Kom.)  
NIDN : 0420106301


Disahkan oleh :

Fakultas Teknologi Industri Itenas  
Dekan,



(Jono Suhartono, S.T., M.T., Ph.D.)  
NIDN : 0406017801

LP2M Itenas  
Ketua,



(Iwan Juwana, S.T., M.EM., Ph.D.)  
NIDN : 0403017701

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR .....</b>	<b>iv</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian.....	2
1.4 Ruang Lingkup Penelitian .....	2
<b>BAB II LANDASAN TEORI .....</b>	<b>4</b>
2.1 Pengolahan Citra digital .....	4
2.2 <i>Python</i> .....	4
2.3 <i>OpenCV</i> .....	4
2.4 Pre-Processing .....	5
2.5 Segmentasi.....	5
2.6 OCR ( <i>Optical Character Recognition</i> ) .....	5
2.7 CCTV .....	6
2.8 Peilschaal.....	6
2.9 Bot Telegram .....	7
2.10 Webhooks.....	8
2.11 Ngrok.....	9
<b>BAB III METODA PENELITIAN .....</b>	<b>10</b>
3.1. Analisis dan gambaran sistem .....	10
3.2. Analisis kebutuhan perangkat keras .....	11
3.3. Analisis kebutuhan perangkat lunak.....	11
3.4. Flowchart dari sistem .....	11
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN.....</b>	<b>16</b>
4.1 Implementasi .....	16
4.2 Implementasi sistem .....	16
4.2.1. <b>Input Image</b> .....	16
4.2.2. <b>Tampilan citra ketika dilakukan <i>cropping</i></b> .....	16

4.2.3.	Mengubah warna pada citra .....	17
4.2.4.	Pemberian nilai ambang ( <i>Threshold</i> ) pada citra.....	18
4.2.5.	Menampilkan angka yang berhasil dideteksi menggunakan OCR ....	19
4.2.6.	Pengiriman ke telegram.....	19
4.3.	Pengujian Sistem .....	20
4.3.1.	Pengujian pada <i>peilschaal</i> dari tinggi 0-90 cm .....	20
4.3.2.	Pengujian pada <i>peilschaal</i> dari tinggi 20-90 cm .....	21
4.3.3.	Pengujian pada <i>peilschaal</i> dari tinggi 30-90 cm .....	22
4.3.4.	Memberikan perintah “ <i>start</i> ” kepada bot telegram.....	22
4.3.5.	Memberikan perintah “ <i>Image</i> ” kepada bot telegram .....	23
4.3.6.	Memberikan perintah “ <i>Help</i> ” kepada bot telegram.....	23
4.3.7.	Memasukan perintah yang tidak tersedia pada bot telegram .....	24
<b>BAB VI PENUTUP .....</b>		<b>25</b>
6.1.	Kesimpulan.....	25
<b>DAFTAR PUSTAKA.....</b>		<b>26</b>

## DAFTAR GAMBAR

Gambar 1. Contoh program OpenCV .....	5
Gambar 2. Contoh output OpenCV .....	5
Gambar 3. Deteksi angka menggunakan OCR .....	6
Gambar 4. Hasil deteksi menggunakan OCR .....	6
Gambar 5. Kamera CCTV .....	6
Gambar 6. Peilschaal .....	7
Gambar 7. Bot Telegram .....	7
Gambar 8. Logo webhooks .....	8
Gambar 9. Alur webhooks .....	8
Gambar 10. Alur ngrok .....	9
Gambar 11. Diagram Blok .....	10
Gambar 12. Flowchart Sistem .....	12
Gambar 13. Sub process cropping image .....	13
Gambar 14. Sub process grayscale .....	14
Gambar 15. Sub process send to telegram .....	15
Gambar 16. Tampilan Awal Citra .....	16
Gambar 17. Tampilan Citra Awal (Pembanding) .....	17
Gambar 18. Cropping Image .....	17
Gambar 19. Hasil grayscale .....	18
Gambar 20. Hasil threshold .....	18
Gambar 21. Hasil menggunakan OCR .....	19
Gambar 22. Hasil pada terminal .....	19
Gambar 23. Pengiriman gambar ke telegram .....	20
Gambar 24. Pengubahan tinggi air secara manual .....	20
Gambar 25. Pengujian 0-90 cm (gambar) .....	21
Gambar 26. Pengujian 0-90 cm (terminal) .....	21
Gambar 27. Pengujian 20-90 cm (gambar) .....	21
Gambar 28. Pengujian 20-90 cm (terminal) .....	22
Gambar 29. Pengujian 30-90 cm (gambar) .....	22
Gambar 30. Pengujian 30-90 cm (terminal) .....	22
Gambar 31. Memberikan perintah "start" pada bot telegram .....	23
Gambar 32. Memberikan perintah "Image" pada bot telegram .....	23
Gambar 33. Memberikan perintah "help" pada bot telegram .....	24
Gambar 34. Memberikan perintah yang tidak tersedia pada bot telegram .....	24

# **BAB I**

## **PENDAHULUAN**

Pada BAB I ini diuraikan mengenai latar belakang, rumusan masalah, tujuan penelitian dan sistematika penulisan.

### **1.1 Latar Belakang**

Banjir merupakan bencana alam yang sering terjadi, yang mengakibatkan banyak kerugian di sekitar lingkungan tempat tinggal, khususnya yang berada di dataran rendah atau daerah yang dekat dengan aliran sungai (Samijayani, 2014). Tidak ada yang dapat memperkirakan kapan bencana alam akan terjadinya, dalam situasi ini banyak kerugian yang disebabkan oleh banjir antara lain kehilangan harta benda bahkan sampai kehilangan nyawa. Dari bencana yang dihadapi oleh masyarakat yang bermukim di dekat aliran sungai deras yang berlangsung lama, warga atau petugas sekitar harus cek kelokasi untuk melihat ketinggian air sungai (Pratiwi, 2020) (Safitri, 2022).

Pada penelitian sebelumnya telah dibahas terkait dengan Alat deteksi banjir diantaranya adalah penelitian yang dibuat oleh (Bahar, 2015), merancang model sistem peringatan dini banjir di kecamatan Satui menggunakan sensor kapastif alumunium foil. Penelitian lain juga telah dilakukan oleh (Valentin, 2021) dengan memanfaatkan sensor ultrasonic. Data nilai kapasitansi dapat ditampilkan dalam serial monitor dan output pada sistem alat menggunakan LED, LCD, dan buzzer untuk menampilkan hasil status dari ketinggian air sungai. (Muzakky, 2018) juga membuat Alat sistem deteksi banjir menggunakan Water level Sensor sebagai pembaca data dan NodeMcu ESP2866 sebagai pemroses sistem kemudian mengirimkan data ke smartphone lewat aplikasi BLYNK, dengan memberikan notifikasi berupa informasi level ketinggian air aman, siaga atau awas. (Astuti, 2018) membuat perancangan alat deteksi bencana banjir dengan Sensor Kapastif dan SMS Gateway dengan output sistem melalui LCD, LED dan buzzer, Ketika air mencapai status aman lampu LED hijau hidup, pada status siaga LED kuning hidup dan mengirimkan berupa SMS peringatan, serta pada status bahaya kemudian LED merah hidup dan mengirimkan SMS kemudian buzzer berbunyi.

Pada penelitian yang dilakukan (Mareta, 2017) membuat Alat pedeteksi ketinggian air interaktif dengan memakai Raspberry Pi sebagai komputer mini untuk mengatur kerja pada sistem, Alat ini berguna untuk mengawasi ketinggian air waduk dengan Aplikasi Telegram. (Adelia, 2020) Juga membuat Sistem pendeteksi bencana banjir dengan memakai mikrokontroler arduino yang mengendalikan sensor ultrasonik. Sensor ultrasonic akan mengirimkan level ketinggian air ke tampilan LCD dan kemudian LCD akan menampilkan status tinggi air dan akan mengirimkan data ke web untuk memberikan informasi status ketinggian air kepada warga.

## **1.2 Rumusan Masalah**

Permasalahan yang ada di PT. HigerTech Karya Sinergi adalah tidak adanya sistem untuk mendeteksi tinggi muka air sehingga data yang diterima oleh perusahaan hanya berupa gambar penggaris pengukur tinggi air (*peilschaal*) yang dipasang di pinggiran sungai.

## **1.3 Tujuan Penelitian**

Berdasarkan rumusan masalah, maka diajukan penelitian untuk membuat sebuah sistem kamera yang selanjutnya diolah menggunakan OCR yang dapat menampilkan tinggi muka air dan hasilnya dikirimkan ke petugas jaga, dengan memanfaatkan fasilitas yang ada dalam hal ini penggaris pengukur tinggi air (*peilschaal*) yang dipasang di pinggiran sungai sebagai deteksi banjir.

## **1.4 Ruang Lingkup Penelitian**

Dalam penelitian yang dilakukan agar fokus terhadap target penelitian yang dilakukan dibuat batasa sebagai berikut :

1. Sistem tidak menggunakan sensor pengukur ketinggian air, dipilih dikarenakan tidak merubah dari sistem pembacaan ketinggian air yang ada.
2. Sistem yang dibangun belum melibatkan database, dikarenakan merupakan penelitian awal dari sistem deteksi ketinggian air yang dilakukan.

3. Penelitian tidak dilakukan di sungai melainkan di simulasi atau model sungai yang berada di PT HigerTech Karya Sinergi.
4. Pemrograman menggunakan bahasa *Python* dengan melibatkan *OpenCV* (*Open Source Computer Vision*) sebagai *library open source* yang dikembangkan oleh intel yang fokus untuk menyederhanakan program untuk citra digital secara *real-time*
5. Hasil pembacaan setelah diolah sistem dikirim menggunakan *Bot Telegram*.



## **BAB II**

### **LANDASAN TEORI**

Pada bab ini dijelaskan mengenai teori pendukung yang digunakan pada pembuatan sistem pendeteksi tinggi muka air.

#### **2.1 Pengolahan Citra digital**

Pengolahan citra digital adalah suatu proses atau cara untuk mengubah atau memperbaiki citra menjadi citra yang sempurna atau lebih baik. Sedangkan citra menurut kamus Webster berarti representasi, kemiripan atau imitasi dari suatu objek. Citra dapat berupa hasil fotografi, lukisan, atau gambaran serta coret-coret yang terjadi di kertas, kanvas, dan di layar monitor. Dapat dikatakan juga citra merupakan sebaran variasi gelap-terang, redup-cerah, dan/atau warna-warni di suatu bidang datar.

#### **2.2 Python**

*Python* adalah bahasa pemrograman berorientasi objek tingkat tinggi, diciptakan oleh Guido Van Rossum. *Python* didesain agar mudah dimengerti, mudah digunakan, dan disebut sebagai bahasa yang *user-friendly* dan mudah digunakan oleh pemula.

#### **2.3 OpenCV**

*OpenCV (Open Source Computer Vision)* adalah sebuah *library open source* yang dikembangkan oleh intel yang fokus untuk menyederhanakan program untuk citra digital secara *real-time*. *OpenCV* memiliki banyak fitur, antara lain pengenalan wajah, *tracking object*, dan berbagai jenis metode untuk AI (*Artificial Intellegence*). Berikut adalah salah satu contoh penggunaan dari library *OpenCV* dan hasil yang ditunjukkan pada Gambar 1 dan Gambar 2

```
import cv2

image = cv2.imread('Image/DB/Try2.jpg')

cv2.imshow('Image', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Gambar 1. Contoh program *OpenCV*



Gambar 2. Contoh output *OpenCV*

**2.4 Pre-Processing**

Proses *pre-processing* merupakan metode yang digunakan untuk memperbaiki kualitas citra agar menjadi lebih baik dari pada sebelumnya dengan cara memanipulasi nilai parameter pada citra.

**2.5 Segmentasi**

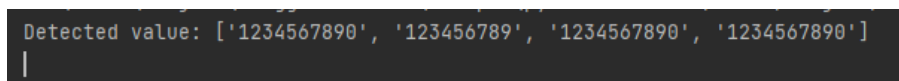
Segmentasi citra adalah metode yang digunakan untuk memisahkan citra menjadi beberapa bagian sesuai dengan kemiripan atribut. Terdapat metode paling sederhana pada segmentasi citra yaitu *thresholding*, fungsi dari *thresholding* yaitu mengubah warna menjadi putih dan hitam dengan mengatur nilai ambang secara manual agar objek dengan *background* dapat terpisah.

**2.6 OCR (*Optical Character Recognition*)**

OCR merupakan sebuah metode untuk mengkonversi teks yang dipindai atau dicetak sehingga komputer dapat mengenali angka dan huruf pada gambar. Berikut adalah salah satu contoh dari penggunaan OCR beserta hasilnya yang ditampilkan pada Gambar 3 dan Gambar 4.



Gambar 3. Deteksi angka menggunakan OCR



Gambar 4. Hasil deteksi menggunakan OCR

## 2.7 CCTV

CCTV (*Closed Circuit Television*) adalah kamera yang digunakan untuk memantau, mengawasi, atau merekam keadaan disekitar untuk keamanan. Kamera CCTV biasanya digunakan untuk daerah pertokoan, perumahan, lalu lintas, perkantoran, dan sekolah. Ditampilkan sebuah gambar dari CCTV pada Gambar 5.



Gambar 5. Kamera CCTV

## 2.8 Peilschaal

*Peilschaal* atau pencatat duga air merupakan alat ukur ketinggian air sungai yang berbentuk penggaris dan akan menjadi pengukur debit air yang berpotensi

menyebabkan banjir dan biasanya *peilschaal* mempunyai tinggi 100 cm. berikut adalah gambar dari *peilschaal* pada Gambar 6.



**Gambar 6. Peilschaal**

## 2.9 Bot Telegram

Telegram menyediakan sebuah fitur API yang memungkinkan adanya fitur bot pada setiap chat yang digunakan oleh pengguna. Fitur bot digunakan untuk mempermudah pengguna untuk melakukan percakapan hanya dengan memasukan sebuah kata kunci. Berikut adalah logo dari bot telegram yang ditampilkan pada Gambar 7.



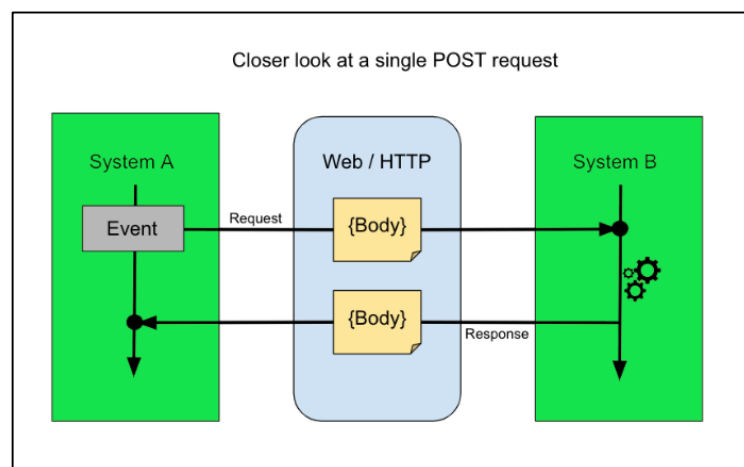
**Gambar 7. Bot Telegram**

## 2.10 Webhooks

*Webhooks* merupakan HTTP *callback*, *webhooks* akan bekerja ketika terjadi *request* atau *trigger* dari luar. Dalam hal ini HTTP *post* merupakan *request* yang diterima. Fungsi utama dari *webhooks* yaitu menerima data secara *real-time* dan mengirimkannya kembali ke pengguna. Berikut ditampilkan logo dari *webhooks* dan alur dari *post request* dari pengguna pada Gambar 8 dan Gambar 9.



Gambar 8. Logo webhooks

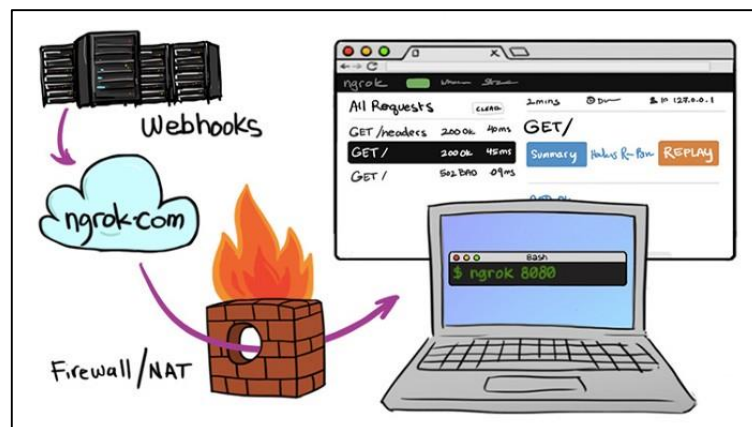


Gambar 9. Alur webhooks

*Webhooks* merupakan jembatan antara chatbot dan sistem pengolahan pesan, dengan menggunakan pesan teks pengguna dapat berinteraksi dengan sistem. Melalui *webhooks* Telegram dan API.AI akan saling berinteraksi sehingga ketika terjadi sebuah *request* dari pengguna maka API.AI akan mengolah request tersebut dan akan mengidentifikasi maksud dari *request* tersebut dan akan dikirimkan kembali ke *webhooks* untuk diolah lebih lanjut.

## 2.11 Ngrok

Ngrok merupakan sebuah *tunnelling* atau bisa disebut juga sebagai sebuah aplikasi yang menjembatani untuk menembus sistem *firewall* yang ada pada jaringan sehingga IP lokal dapat diubah menjadi IP publik dan akan diubah menjadi sebuah URL agar dapat diakses melalui internet. Berikut adalah tampilan dari alur ngrok yang dibuat secara sederhana ditampilkan pada Gambar 10.



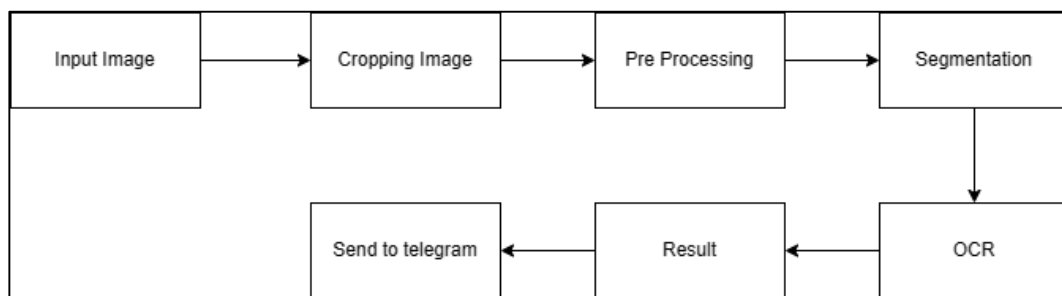
Gambar 10. Alur ngrok

## BAB III METODA PENELITIAN

Pada bab ini diuraikan analisis dan perancangan sistem pendeteksi tinggi muka air menggunakan *python*, *OpenCV*, dan OCR (*Optical Character Recognition*) dengan jelas.

### 3.1. Analisis dan gambaran sistem

Dalam pengembangan aplikasi pendeteksi tinggi muka air, sistem yang dibangun diharapkan mampu untuk mendeteksi angka pada penggaris *Peilschaal* yang digunakan. Adapun gambaran diagram blok dari sistem yang dibangun diperlihatkan pada Gambar 11 sebagai berikut :



**Gambar 11. Diagram Blok**

Berikut adalah penjelasan singkat dari setiap proses yang digunakan dalam pembuatan sistem :

1. *Input Image* : menampilkan citra yang akan digunakan untuk proses OCR (*Optical Character Recognition*).
2. *Cropping Image* : memotong bagian yang tidak perlu pada citra agar mempermudah proses pendeteksian angka.
3. *Pre-Processing* : melakukan perubahan warna pada citra menjadi abu-abu atau *grayscale*.
4. *Segmentation* : pemberian nilai ambang atau *thresholding* untuk memisahkan objek dengan latar belakang pada citra.
5. OCR : proses untuk mengidentifikasi karakter pada citra yang digunakan
6. *Result* : menampilkan hasil identifikasi menggunakan OCR yaitu berupa

angka pada citra yang digunakan.

7. *Send to telegram* : mengirimkan hasil akhir ke telegram.

### **3.2. Analisis kebutuhan perangkat keras**

Pada pembuatan aplikasi sistem pendeteksi tinggi muka air ini membutuhkan perangkat keras yang dapat membantu dan menunjang pembuatan sistem. Berikut adalah spesifikasi yang digunakan oleh penulis untuk membuat sistem :

1. *Laptop ACER Aspire 5*
2. *Processor* : *Intel Core i3 – 7020 U*
3. *VGA* : *Intel HD 620 & Nvidia MX230 2GB*
4. *RAM* : *12 GB*
5. *Internal Storage: 1 TB*
6. *CCTV*
7. *Peilschall*

### **3.3. Analisis kebutuhan perangkat lunak**

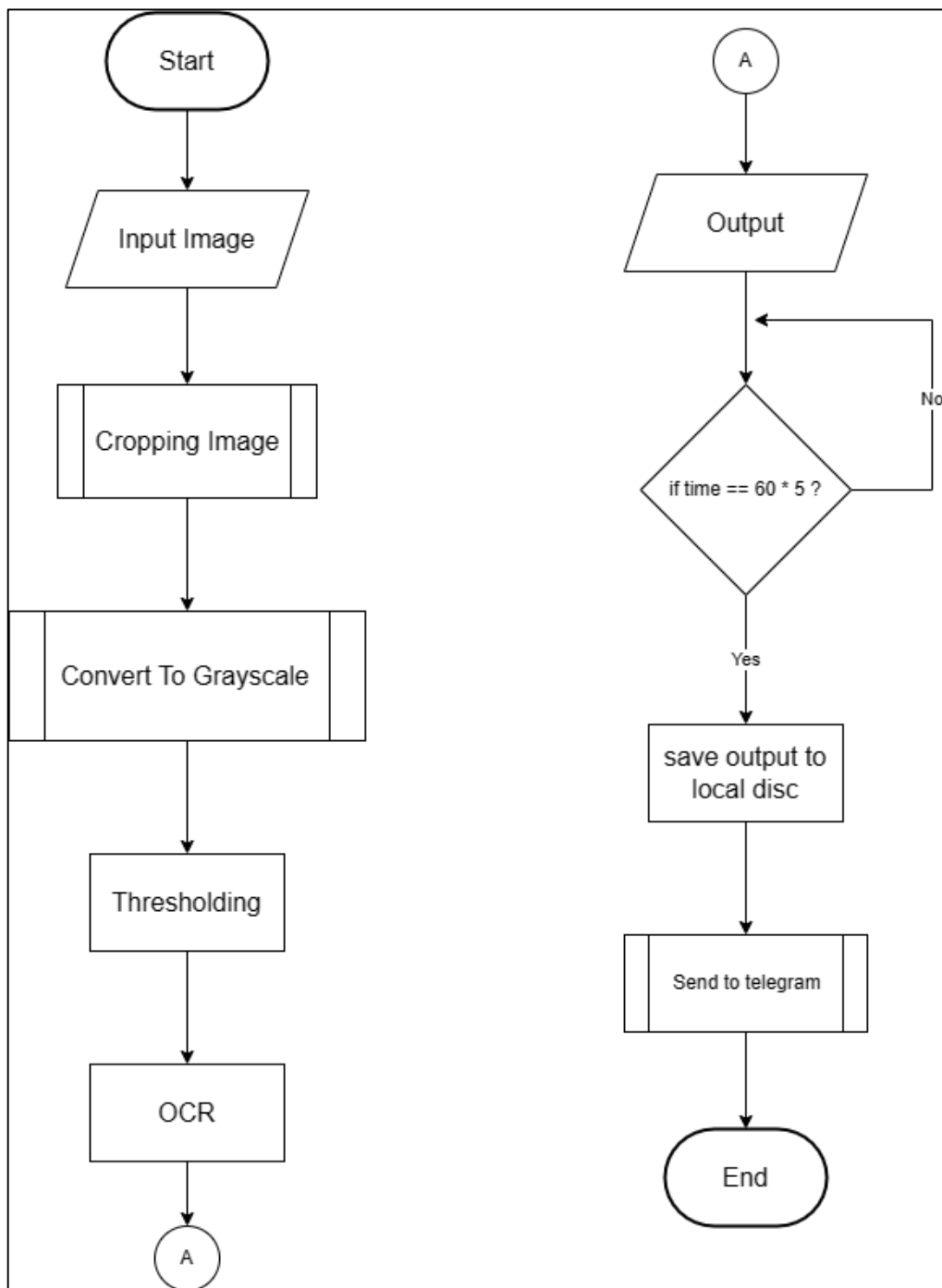
Terdapat beberapa kebutuhan perangkat lunak yang diperlukan untuk pembuatan sistem pendeteksi tinggi muka air diantaranya sebagai berikut :

1. *Windows 10 64-bit*
2. *Pycharm*

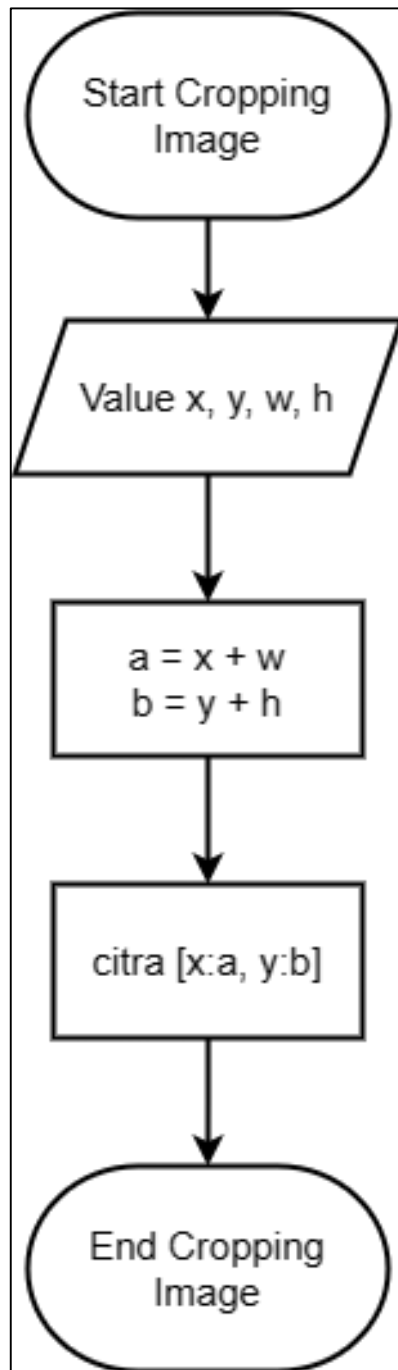
### **3.4. Flowchart dari sistem**

*Flowchart* adalah diagram yang menampilkan proses dan keputusan dalam suatu program. Setiap proses digambarkan dengan bentuk diagram dan dihubungkan dengan arah panah. Berikut adalah *flowchart* dari sistem yang telah dibuat dan ditampilkan pada Gambar 12. Selanjutnya ditampilkan *sub process cropping image, grayscale*, dan *send to telegram* pada Gambar 13, Gambar 14, dan Gambar 15.

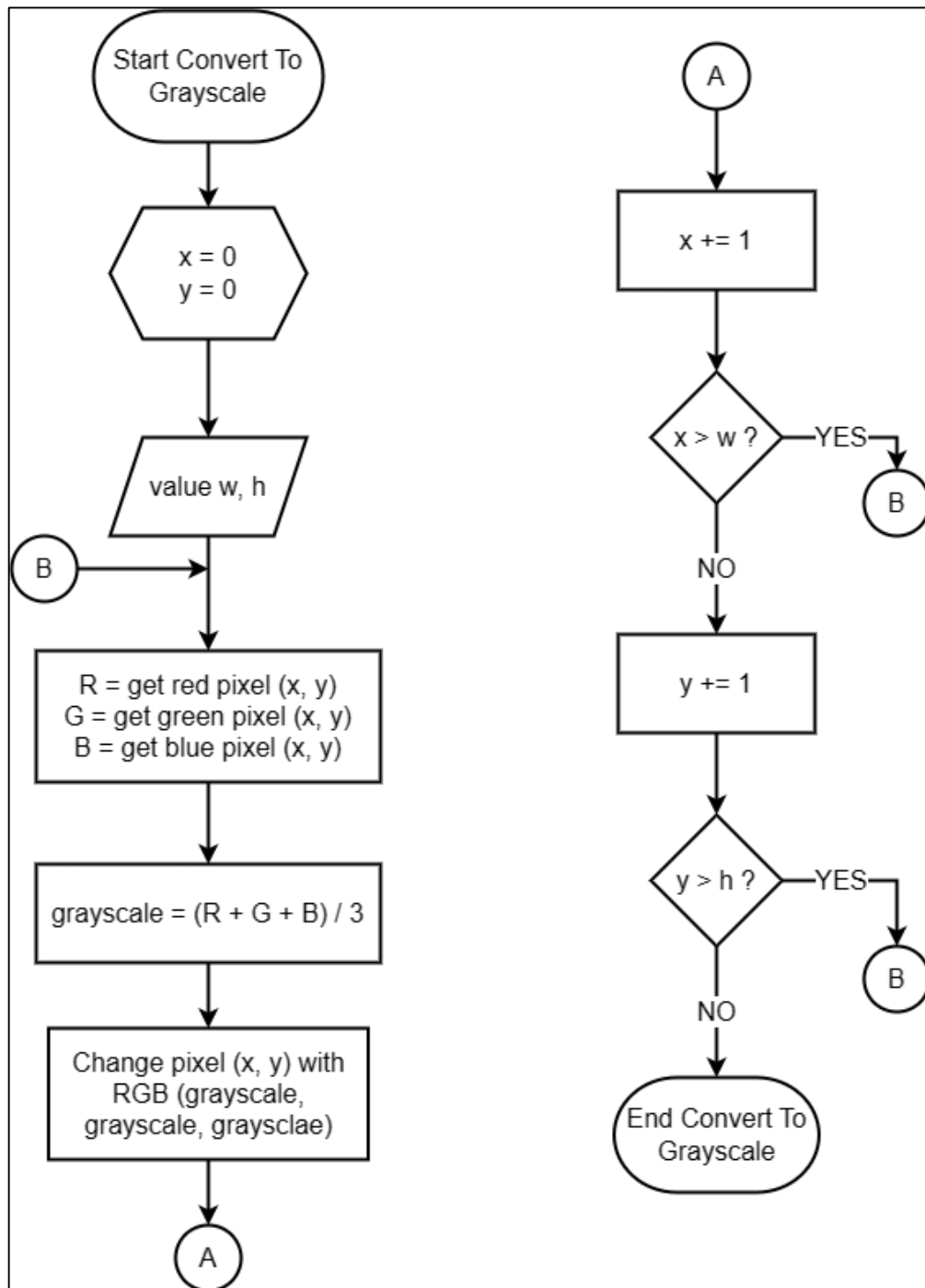




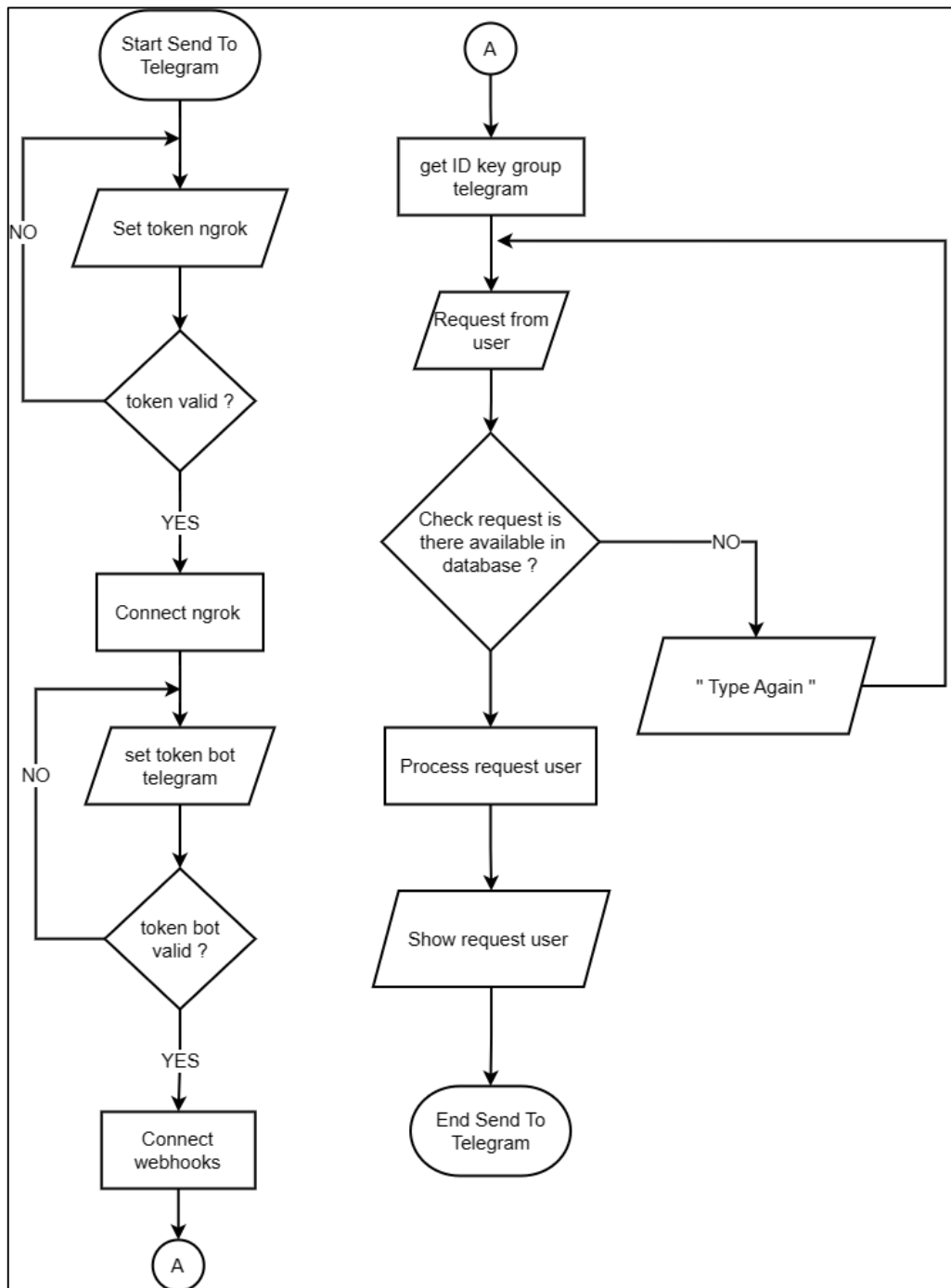
Gambar 12. Flowchart Sistem



Gambar 13. Sub process cropping image



Gambar 14. Sub process grayscale



Gambar 15. Sub process send to telegram

## BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab ini diperlihatkan hasil implementasi dan pengujian dari sistem pendeteksi tinggi muka air yang telah dibuat menggunakan bahasa pemrograman *python*.

### 4.1 Implementasi

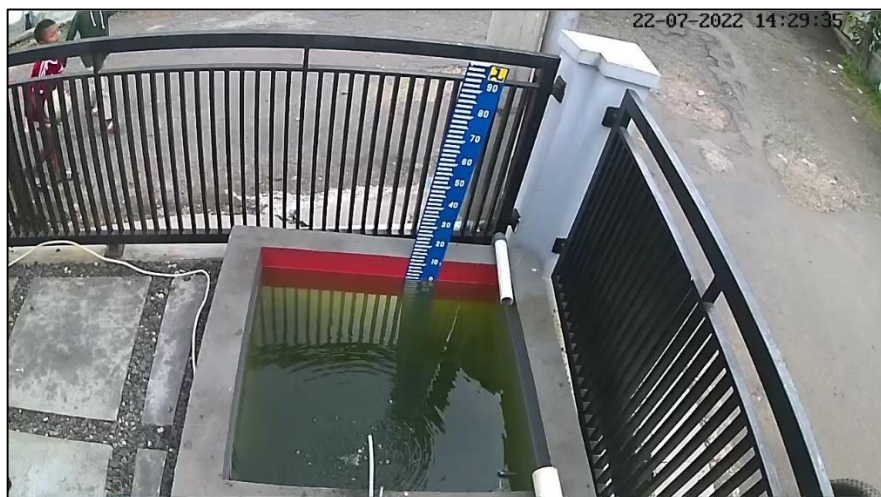
Implementasi adalah sebuah penerapan yang dilakukan berdasarkan dengan perencanaan dan analisis yang telah dibuat untuk mencapai tujuan tertentu.

### 4.2 Implementasi sistem

Berikut adalah implementasi dari sistem pendeteksi tinggi muka air yang telah dibangun menggunakan bahasa pemrograman *Python*.

#### 4.2.1. Input Image

Pada tahap pertama, citra yang diambil menggunakan kamera cctv yang terpasang dilapangan. Gambar diperlihatkan pada Gambar 16.



**Gambar 16. Tampilan Awal Citra**

#### 4.2.2. Tampilan citra ketika dilakukan *cropping*

Pada tahap kedua, citra dilakukan *cropping* untuk menghilangkan bagian yang tidak diperlukan dan agar hanya terlihat objek dari peilschaal. Berikut adalah perbandingan citra asli dan citra yang telah diberikan *cropping* pada Gambar 17 dan Gambar 18.



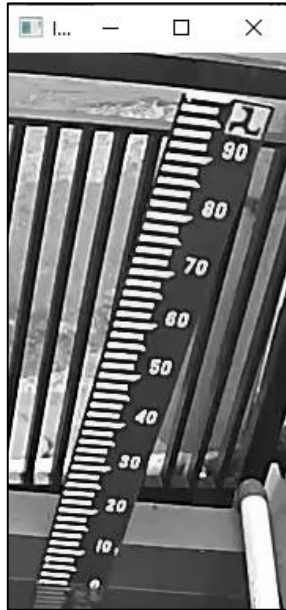
Gambar 17. Tampilan Citra Awal (Pembanding)



Gambar 18. Cropping Image

#### 4.2.3. Mengubah warna pada citra

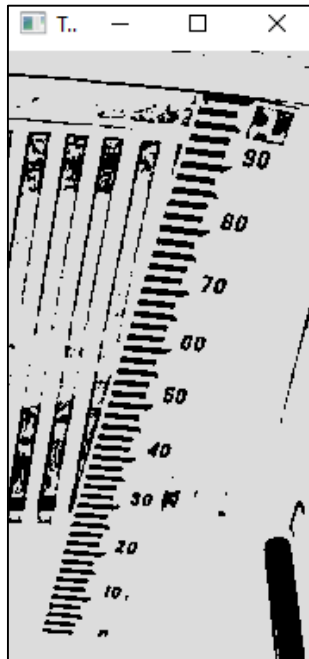
Pada tahap ketiga, warna pada citra diubah menjadi *grayscale* atau skala keabuan, penggunaan *grayscale* berguna untuk memisahkan objek dan *background* pada citra. Berikut adalah citra yang telah diberikan efek *grayscale* ditampilkan pada Gambar 19.



Gambar 19. Hasil grayscale

#### 4.2.4. Pemberian nilai ambang (*Threshold*) pada citra

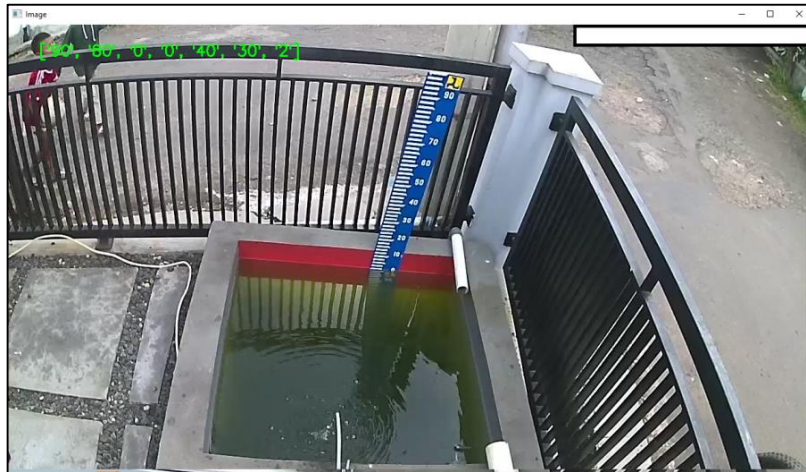
Pada tahap keempat, citra diberikan nilai ambang (*Threshold*) untuk memisahkan objek dengan *background*, kemudian diberikan efek *invert* pada warna citra agar angka pada *peilschaal* dapat dideteksi terlihat seperti pada Gambar 20.



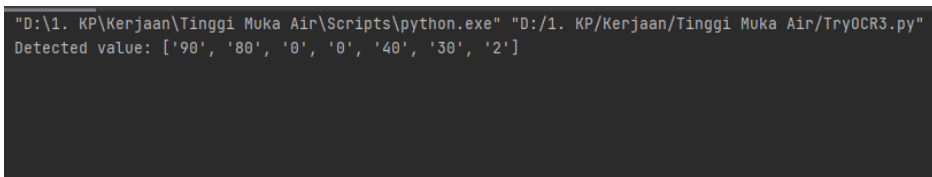
Gambar 20. Hasil threshold

#### 4.2.5. Menampilkan angka yang berhasil dideteksi menggunakan OCR

Tahap kelima, angka yang berhasil dideteksi menggunakan OCR (*Optical Character Recognition*) ditampilkan pada citra seperti pada Gambar 21 dan Gambar 22.



Gambar 21. Hasil menggunakan OCR

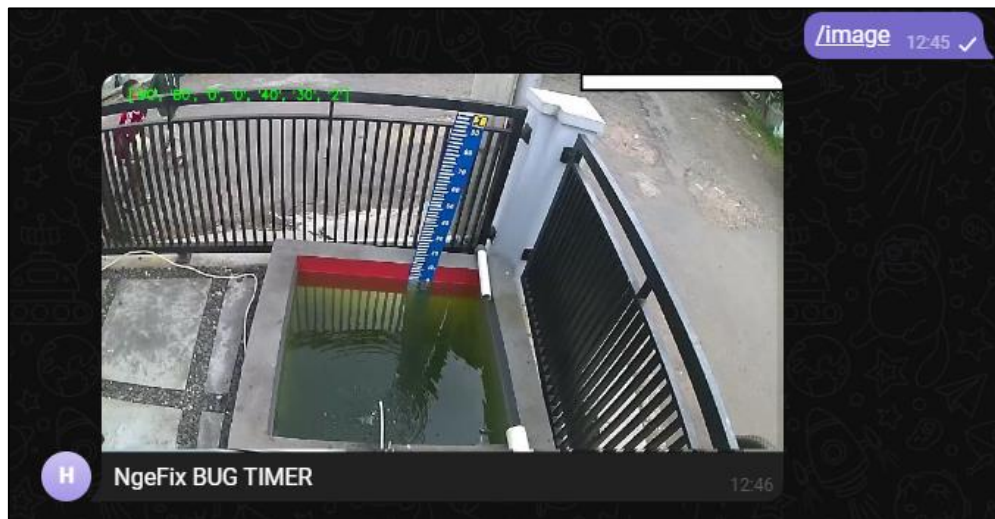


Gambar 22. Hasil pada terminal

#### 4.2.6. Pengiriman ke telegram

Tahap terakhir yaitu pengiriman yang dikirim menuju telegram dengan selang waktu 5 menit untuk setiap pengiriman gambar dari lapangan yang ditampilkan pada Gambar 23.





Gambar 23. Pengiriman gambar ke telegram

### 4.3. Pengujian Sistem

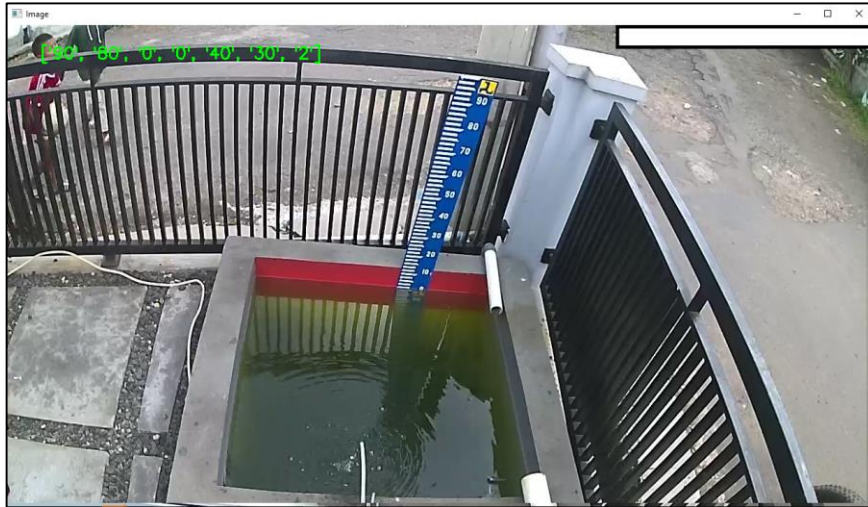
Berikut adalah pengujian sistem yang dilakukan dengan melakukan perubahan tinggi air secara manual pada *input* sistem sebagai contoh: memberikan kotak warna untuk menutupi angka 0-20 pada *peilschaal* seperti pada Gambar 24.



Gambar 24. Perubahan tinggi air secara manual

#### 4.3.1. Pengujian pada *peilschaal* dari tinggi 0-90 cm

Untuk pengujian pertama *output* yang dikeluarkan hanya dapat mendeteksi 5 angka pada *peilschaal* yaitu 90, 80, 0, 40, 30, dan 2. Seperti yang terlihat pada Gambar 25 dan Gambar 26.



Gambar 25. Pengujian 0-90 cm (gambar)

```
"D:\1. KP\Kerjaan\Tinggi Muka Air\Scripts\python.exe" "D:/1. KP/Kerjaan/Tinggi Muka Air/TryOCR3.py"  
Detected value: ['90', '80', '0', '0', '40', '30', '2']
```

Gambar 26. Pengujian 0-90 cm (terminal)

#### 4.3.2. Pengujian pada *peilschaal* dari tinggi 20-90 cm

Pada pengujian kedua *output* yang ditampilkan hanya 5 angka yaitu 90, 40, 70, 40, dan 0. Seperti pada Gambar 27 dan Gambar 28



Gambar 27. Pengujian 20-90 cm (gambar)

```
"D:\1. KP\Kerjaan\Tinggi Muka Air\Scripts\python.exe" "D:\1. KP\Kerjaan\Tinggi Muka Air\TryOCR3.py"
Detected value: ['90', '40', '70', '0', '40', '0']
```

Gambar 28. Pengujian 20-90 cm (terminal)

#### 4.3.3. Pengujian pada *peilschaal* dari tinggi 30-90 cm

Pada pengujian ketiga sistem hanya dapat mendeteksi angka 90, 40, 70, 0, 20. Seperti pada Gambar 29 dan Gambar 30.



Gambar 29. Pengujian 30-90 cm (gambar)

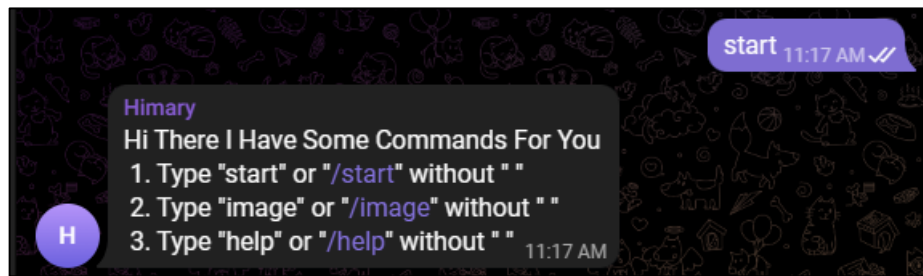
```
"D:\1. KP\Kerjaan\Tinggi Muka Air\Scripts\python.exe" "D:\1. KP\Kerjaan\Tinggi Muka Air\TryOCR3.py"
Detected value: ['90', '40', '70', '0', '40', '20']
```

Gambar 30. Pengujian 30-90 cm (terminal)

#### 4.3.4. Memberikan perintah “start” kepada bot telegram

Selanjutnya gambar yang telah dideteksi oleh OCR akan dikirimkan menuju telegram. Perintah “start” berfungsi untuk memulai percakapan awal dengan bot dan bot akan memberikan pilihan kepada user untuk perintah selanjutnya, seperti pada Gambar 31.

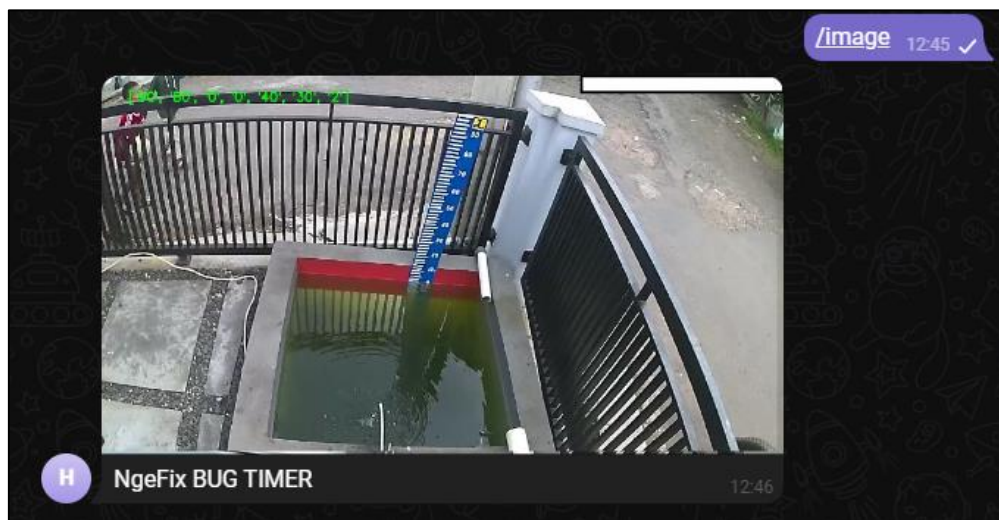




Gambar 31. Memberikan perintah "start" pada bot telegram

#### 4.3.5. Memberikan perintah “*Image*” kepada bot telegram

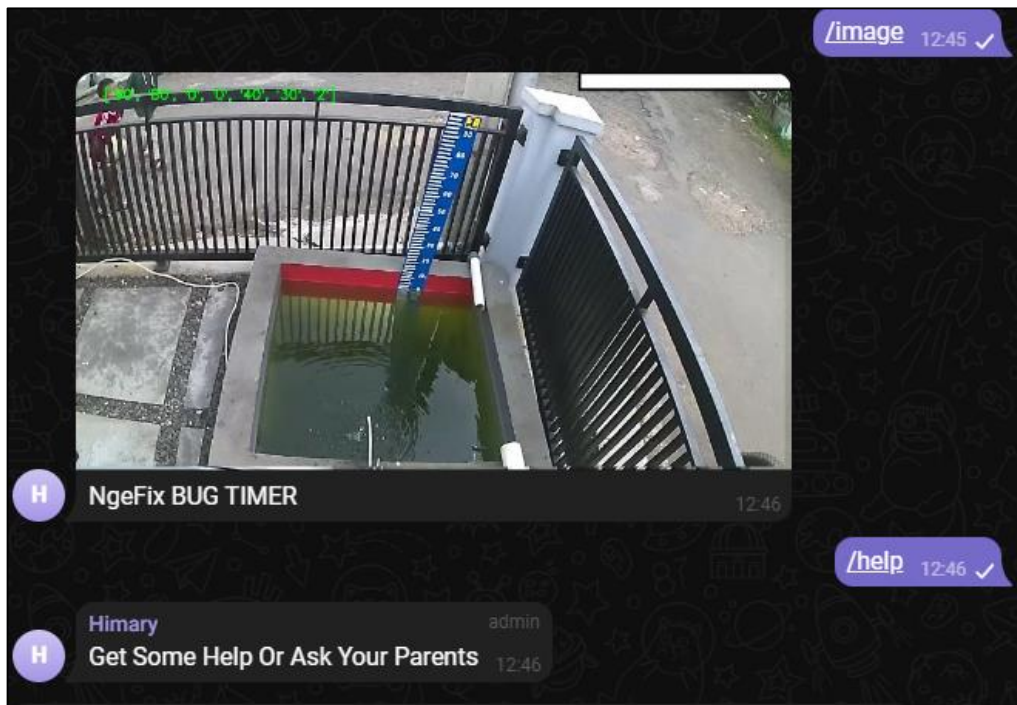
Perintah selanjut nya berfungsi untuk menampilkan gambar yang dari lapangan ke user, seperti yang terlihat pada Gambar 32.



Gambar 32. Memberikan perintah "Image" pada bot telegram

#### 4.3.6. Memberikan perintah “*Help*” kepada bot telegram

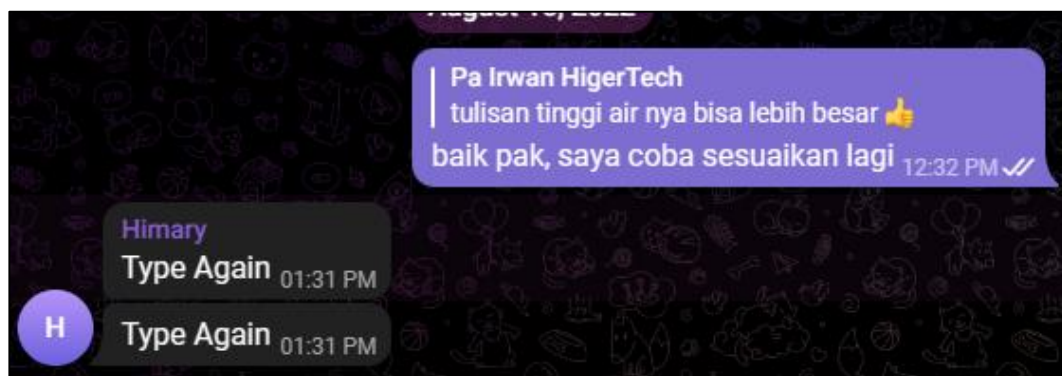
Perintah terakhir yang tersedia pada bot telegram yaitu “*help*” yang berfungsi ketika user tidak mengetahui mengenai bot tersebut, atau terdapat kendala pada saat memproses perintah lainnya. Diperlihatkan pada Gambar 33.



Gambar 33. Memberikan perintah "help" pada bot telegram

#### 4.3.7. Memasukan perintah yang tidak tersedia pada bot telegram

Ketika user memasukan perintah yang tidak tersedia pada bot telegram maka bot akan memerintahkan user untuk memasukan perintah yang benar, seperti pada Gambar 34.



Gambar 34. Memberikan perintah yang tidak tersedia pada bot telegram

## **BAB VI PENUTUP**

Pada bab ini berisikan tentang kesimpulan dan saran yang didapatkan dari perancangan, analisis, dan implementasi sistem pendeteksi tinggi muka air

### **6.1. Kesimpulan**

Berdasarkan hasil penelitian yang telah dilaksanakan perlu dilakukan pengujian lebih lanjut dikarenakan masih terdapat kendala dalam mendeteksi angka pada peilschaal. Hal tersebut dapat terjadi dikarenakan kualitas gambar yang buruk, angka yang sulit dideteksi dikarenakan warna dari *background*, nilai warna atau nilai ambang yang kurang tepat pada saat melakukan proses *thresholding*, dan ketinggian serta sudut kemiringan dari kamera dapat menjadi salah satu faktor tidak terdeteksinya angka sehingga hasil dari OCR tidak akurat dan maksimal.

## DAFTAR PUSTAKA

- Adelia, M. (2020). *Sistem Pendeteksi Bencana Banjir Menggunakan Arduino Berbasis Web*. Medan: Universitas Pembangunan Panca Budi.
- Astuti, W. a. (2018). Perancangan Deteksi Banjir Menggunakan Sensor Kapastif Mikrokontroler ATmega328p dan SMS Gateway. *Jurnal Informatika*, 255-261.
- Bahar, B. a. (2015). Model Sistem Peringatan Dini Banjir Di Kecamatan Satu Menggunakan Sensor Kapasitif Aluminium Foil. *Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, 545-552.
- Mareta, R. R. (2017). Pendeteksi Ketinggian Air Interaktif Dengan Aplikasi Telegram Berbasis Raspberry Pi. *JST (Jurnal Sains dan Teknologi)*, 279-289.
- Muzakky, A. N. (2018). Perancangan Sistem Deteksi Banjir Berbasis IoT. *In Conference on Innovation and Application of Science and Technology (CIASTECH)* (pp. 660-667). Malang: Universitas Widyagama.
- Pratiwi, D. S. (2020). Peningkatan Pengetahuan Masyarakat Terhadap Drainase Berporus yang Difungsikan Sebagai Tempat Peresapan Air Hujan. *Journal of Social Sciences and Technology for Community Service (JSSTCS)*.
- Safitri, D. P. (2022). Analisis Pola Aliran Banjir Pada Sungai Cimadur, Provinsi Banten Dengan Menggunakan Hec-Ras. *JICE (Journal of Infrastructural in Civil Engineering)*, 19-30.
- Samijayani, O. I. (2014). Implementasi Sistem Sensor Sederhana untuk Peringatan Banjir Melalui SMS. *Jurnal Al-Azhar Indonesia Seri Sains dan Teknologi*, 22-27.
- Valentin, R. (2021). Implementasi Sensor Ultrasonik Berbasis Mikrokontroler Untuk Sistem Peringatan Dini Banjir. *Jurnal Ilmiah Mahasiswa Kendali dan Listrik*, 32-41.