

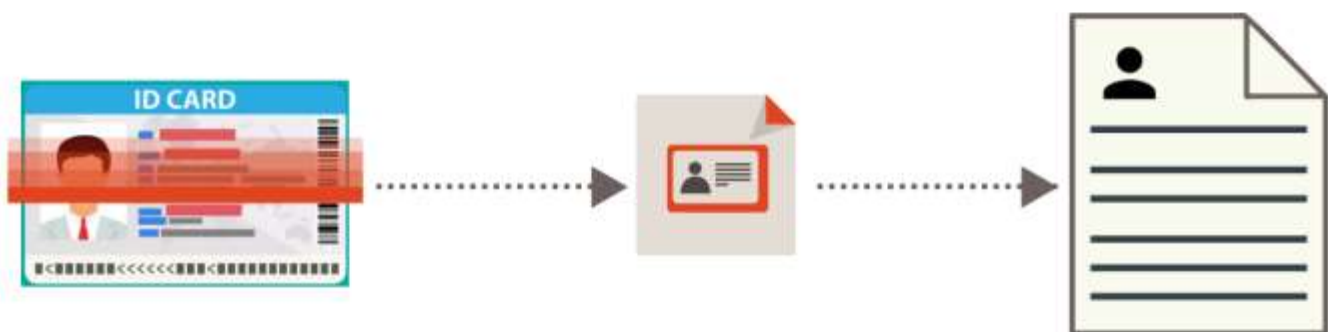
APLIKASI PERBAIKAN KUALITAS CITRA PADA EKSTRAKSI INFORMASI PADA KTP DAN SIM

ANGGOTA KELOMPOK A2

Dzulfiqar Zahran S. 152018004
Andika Budi Cahyadi 152018011
Alifian Alvarez F.S. 152018021

DOSEN PEMBIMBING

Irma Amelia D.,MT



PROJECT OVERVIEW

Proposal ini berfungsi untuk memenuhi salah satu tugas dari mata kuliah Pemograman Citra Digital yang bertemakan Smart City

KATA PENGANTAR

Dengan menyebut nama Allah subhanahu wa ta'ala yang Maha Pengasih lagi Maha Penyayang, kami panjatkan puji dan syukur ke hadirat-Nya, yang telah melimpahkan rahmat dan hidayah-Nya sehingga kami dapat mengajukan proposal aplikasi Pengolahan Citra Digital dengan judul “Aplikasi Perbaikan Kualitas Citra Pada Ekstraksi Informasi Pada KTP dan SIM”. Proposal ini kami susun untuk memenuhi tugas mata kuliah Pengolahan Citra Digital.

Proposal ini kami susun dengan bantuan dari berbagai pihak sehingga dapat memperlancar penyusunan proposal ini. Untuk itu, kami menyampaikan terima kasih kepada dosen mata kuliah Pengolahan Citra Digital, Irma Amelia Dewi M.T., dan semua pihak yang telah berkontribusi dalam penyusunan proposal ini.

Kami menyadari bahwa proposal ini masih memiliki kekurangan. Oleh karena itu, kami mengharapkan kritik dan saran demi penyempurnaan proposal ini. Kami juga berharap agar proposal ini dapat memberikan manfaat bagi kami secara khusus dan bagi masyarakat secara umum.

Bandung, 10 Maret 2020

Penyusun

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iii
Daftar Tabel	iv
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 TUJUAN	1
1.3 RUMUSAN MASALAH	1
1.4 RUANG LINGKUP	2
1.5 TINJAUAN PUSTAKA.....	2
BAB II LANDASAN TEORI.....	4
2.1 LANDASAN TEORI	4
BAB III METODE PENELITIAN	5
3.1 DESKRIPSI SISTEM	5
3.2 ALUR PROSES SISTEM	5
3.3 PARAMETER.....	5
3.4 OPERASI PRE-PROCESSING	5
Daftar Pustaka.....	21

DAFTAR GAMBAR

Gambar 3. 1 Fowchart Proses Grayscale.....	6
Gambar 3. 2 Kode Grayscale Image.....	7
Gambar 3. 3 Gambar Citra Awal.....	7
Gambar 3. 4 Gambar Citra Setelah melalui Proses Grayscale.....	8
Gambar 3. 5 Kode Memunculkan Citra ke GUI.....	10
Gambar 3. 6 List Wadah untuk menyimpan data Citra.....	11
Gambar 3. 7 Gambar Citra Awal (2)	12
Gambar 3. 8 Gambar ketika citra pada GUI.....	12
Gambar 3. 9 Flowchart Proses Pengaturan Kontrast	13
Gambar 3. 10 Kode Pengaturan Kontrast Citra	14
Gambar 3. 11 Gambar Citra Awal (3)	15
Gambar 3. 12 Gambar Citra setelah Proses Pengaturan Kontrast..	15
Gambar 3. 13 Flowchart Proses Thresholding.....	17
Gambar 3. 14 Kode Thresholding Citra	18
Gambar 3. 15 Gambar Citra Awal (4)	19
Gambar 3. 16 Gambar Citra setelah Proses Thresholding	19

DAFTAR TABEL

Tabel 3. 1 Sample Citra RGB Awal.....	8
Tabel 3. 2 Tabel Citra Setelah Proses Grayscale.....	9
Tabel 3. 3 Sampel Citra Grayscale.....	15
Tabel 3. 4 Hasil dari Proses Pengaturan Kontras.....	16
Tabel 3. 5 Sampel Citra Kontrast.....	19
Tabel 3. 6 Hasil dari Proses Thresholding.....	20

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Smart City adalah konsep perencanaan kota dengan memanfaatkan perkembangan teknologi yang akan membuat hidup lebih mudah dan sehat dengan tingkat efisiensi dan efektifitas yang tinggi. Smart City juga kota yang secara antisipatif mampu mengelola sumber daya secara inovatif dan berdaya saing, dengan dukungan teknologi dalam rangka mewujudkan kota yang nyaman dan berkelanjutan

Semakin berkembangnya zaman, semakin banyaknya penduduk di negara Indonesia. Hal itulah menyebabkan pengolahan informasi setiap penduduk Indonesia menjadi lebih banyak dan repot, karena masih menggunakan pengiputan manual untuk setiap penduduk. Hal tersebut menjadi salah satu faktor penghambat kemajuan Smart City untuk indonesia

Karena masalah diatas, kami ingin membuat sebuah solusi untuk membantu menghilangkan masalah tersebut, yaitu dengan membuat sebuah aplikasi yang bernama “Ekstrasi Informasi Pada KTP dan SIM” yang berbasis Computer Vision. Dengan aplikasi ini pemerintahan dapat mendapatkan data penduduk Indonesia secara otomatis dengan memotret Kartu Tanda Penduduk atau Surat Izin Mengendarai penduduk Indonesia.

1.2 TUJUAN

Dari latar belakang tersebut, kami memiliki tujuan yang terdiri dari :

1. Untuk mengetahui cara mendapatkan Informasi dari Kartu Tanda Penduduk dan Surat Izin Mengemudi
2. Untuk mengetahui cara mengekstrak informasi dari Kartu Tanda Penduduk dan Surat Izin Mengemudi

1.3 RUMUSAN MASALAH

Rumusan Masalah dari Proyek yang kami kerjakan yaitu:

1. Bagaimana cara mendapatkan informasi yang terdapat pada Kartu Tanda Penduduk dan Surat Izin Mengemudi?
2. Bagaimana cara mengekstrak informasi dari Kartu Tanda Penduduk dan Surat Izin Mengemudi?

1.4 RUANG LINGKUP

Pada pembahasan ini terfokus pada :

1. Apa saja proses Pre-Processing pada citra yang akan diolah
2. Penjelasan langkah – langkah Pre-Processing yang akan di lakukan pada citra

1.5 TINJAUAN PUSTAKA

1. *Improved Local Binary Pattern for Real Scene Optical Character Recognition*

Oleh: Chu-Sing Yang,
Yung-Hsuan Yang
Department of Electrical Engineering
National Cheng Kung University

2. *Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm*

Oleh: Joseph Tarigan,
Nadia,
Ryanda Diedan,
Yaya Suryana
Universitas Bina Nusantara

3. *Mobile Financial Management Application using Google Cloud Vision API*

Oleh: Kurniawan Dwi Saputra,
Della Anggi Rahmaastri,
Karina Setiawan,
Dewi Suryani,
Yudy Purnama
Falkultas Teknik Komputer
Universitas Bina Nusantara

4. *An Examination of Character Recognition on ID card using Template Matching Approach*

Oleh: Michael Ryan,
Novita Hanafiah

Universitas Bina Nusantara

5. *Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application*

Oleh: Abdul Robby G,

Antonia Tandra,

Imelda Susanto,

Jeklin Harefa,

Andry Chowanda

Falkultas Teknik Komputer

Universitas Bina Nusantara

BAB II

LANDASAN TEORI

2.1 LANDASAN TEORI

- Citra (Bahasa Inggris: image) adalah kombinasi antara titik, garis, bidang, dan warna untuk menciptakan suatu imitasi dari suatu objek—biasanya objek fisik atau manusia. Citra bisa berwujud gambar (picture) dua dimensi, seperti lukisan, foto, dan berwujud tiga dimensi, seperti patung.
- Pengolahan citra digital merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra dengan bantuan komputer.
- Pengolahan citra digital dapat dikelompokkan dalam dua jenis kegiatan :
 - Memperbaiki kualitas suatu gambar, sehingga dapat lebih mudah diinterpretasi oleh mata manusia.
 - Mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis.
- Scanner adalah sebuah alat pemindai salah satu perangkat input pada komputer yang merupakan suatu alat yang berfungsi untuk menduplikat objek layaknya seperti mesin fotokopy ke dalam bentuk digital.
- Fungsi Scanner yaitu dapat menduplikasi *hard-copy* seperti kertas, makalah, maupun foto ke dalam bentuk digital.
- Tesseract adalah mesin pengenalan karakter optik untuk berbagai sistem operasi.

BAB III

METODE PENELITIAN

3.1 DESKRIPSI SISTEM

Sistem ini melakukan pengambilan citra kartu dan mengekstrak informasi dari gambar yang diambil.

3.2 ALUR PROSES SISTEM

1. Citra akan di baca oleh sistem dan di proses menjadi citra keabuan.
2. Citra akan di atur contrast nya secara manual oleh user
3. Setelah itu, citra akan dikonversi dengan proses thresholding agar menjadi citra biner
4. Citra biner selanjutnya akan di tampilkan di dalam Label dan di simpan untuk proses ekstraksi dengan nama temp.png

3.3 PARAMETER

Dalam penelitian ini, objek yang akan diteliti berupa kartu, kartu tersebut adalah kartu SIM (Surat Izin Mengemudi) dan kartu KTP (Kartu Tanda Penduduk).

3.4 OPERASI PRE-PROCESSING

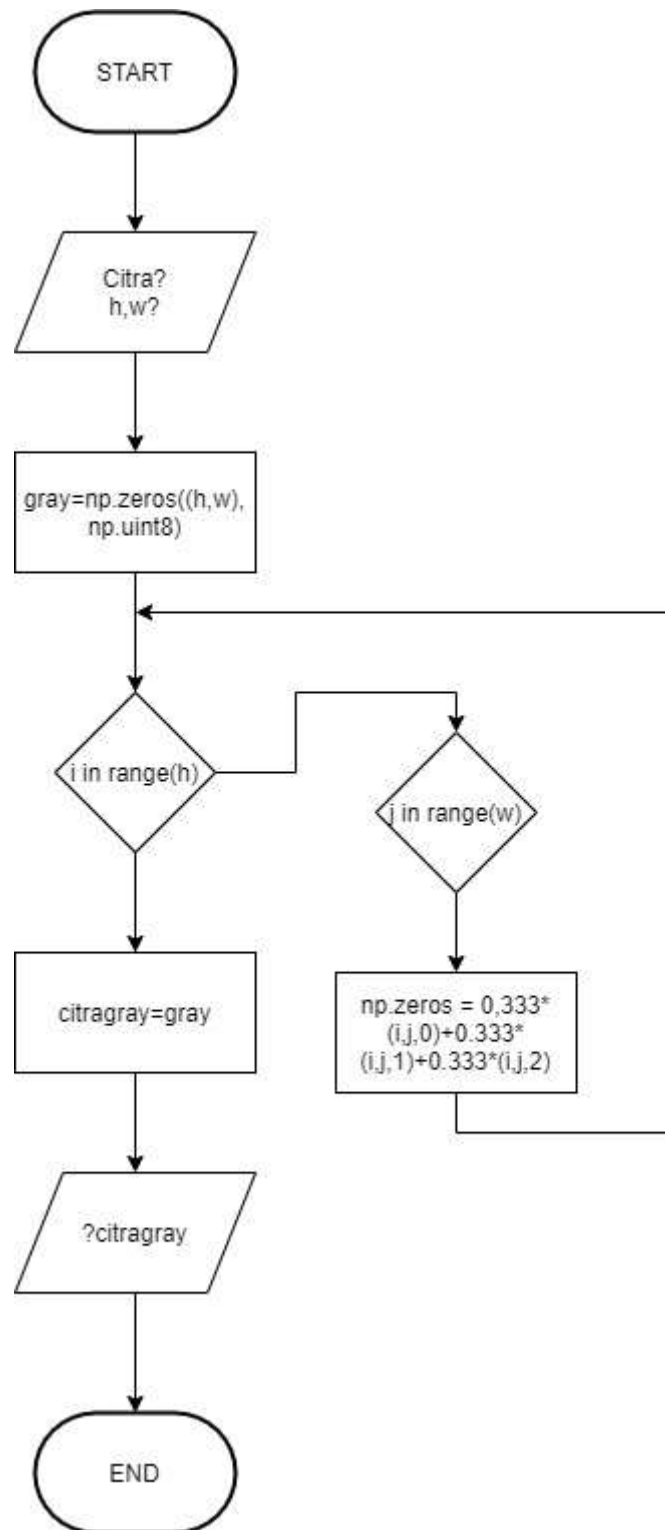
1. **Nama operasi** : Greyscale

Tujuan : Untuk mengubah citra RGB menjadi citra greyscale

Alur Proses : - Pertama, sistem akan mengambil ukuran citra

- Kemudian sistem akan membuat gambar yang serupa dengan ukuran gambar aslinya, tetapi semua nilai pixel nya nol.
- Sistem melakukan looping untuk mengubah nilai RGB menjadi nilai greyscale dengan cara $(0.333 * (\text{Nilai pixel R}) + 0.333 * (\text{Nilai pixel G}) + 0.333 * (\text{Nilai pixel B}))$ dan mengkliping nilai pixel
- Hasil dari looping tersebut akan dimasukkan ke gambar yang sudah dibuat sebelumnya.

Flowchart:



Gambar 3. 1 Fowchart Proses Grayscale

Kode:

```
def grayImage(self):
    h, w = self.image.shape[:2]
    gray = np.zeros((h, w), np.uint8)
    for i in range(h):
        for j in range(w):
            gray[i, j] = np.clip(0.333 * self.image[i, j, 0] + 0.333 * self.image[i, j, 1] + 0.333 * self.image[i, j, 2], 0, 255)
    self.image = gray
    self.image2 = self.image
    self.display(1)
```

Gambar 3. 2 Kode Grayscale Image

Penjelasan Kode:

1. h,w merupakan variabel untuk membuat gambar yang baru, isi dari variabel h,w adalah ukuran dari citra gambar.
2. gray merupakan variabel yang isinya citra dengan ukuran yang sesuai dengan h,w, akan tetapi nilai nya nol
3. for i in range (h) merupakan loop untuk nilai y, for j in range(w) merupakan loop untuk nilai x, variabel grey tadi nilai pixel nya akan digantikan dengan nilai yang baru, nilai yang baru tersebut didapatkan dari $(0.333 \cdot R) + (0.333 \cdot G) + (0.333 \cdot B)$. dilakukan clipping juga dimana batas bawah nya = 0 dan batas atasnya 255
4. self.image = gray, gambar sebelumnya digantikan dengan gambar yang berada di variabel grey
5. self.image 2 = self.image untuk menngkopikan gambar
6. self.display(1) untuk menampilkan citra hasil pada self.display(1)

Perbandingan Citra:



Gambar 3. 3 Gambar Citra Awal



Gambar 3. 4 Gambar Citra Setelah melalui Proses Grayscale

Analisis:

Setelah dilakukan operasi greyscale, nilai warna R, G dan B pada gambar, berubah warna citra berubah menjadi keabuan dengan 1 nilai warna yaitu Gray.

Perhitungan Studi Kasus:

Pixel Citra Awal (RGB):

R: 129	R: 101	R: 69
G: 130	G: 100	G: 67
B: 135	B: 105	B: 70
R: 148	R: 127	R: 92
G: 149	G: 126	G: 90
B: 154	B: 131	B: 93
R: 156	R: 144	R: 124
G: 157	G: 143	G: 122
B: 162	B: 148	B: 125

Tabel 3. 1 Sample Citra RGB Awal

Lakukan Proses Grayscale dengan rumus:

$$f(x, y) = (0,333 * R) + (0,333 * G) + (0,333 * B)$$

$$f(0, 0) = (0,333 * 129) + (0,333 * 130) + (0,333 * 135) = 131$$

$$f(0, 1) = (0,333 * 101) + (0,333 * 100) + (0,333 * 105) = 101$$

$$f(0, 2) = (0,333 * 69) + (0,333 * 67) + (0,333 * 70) = 68$$

$$f(1, 0) = (0,333 * 148) + (0,333 * 149) + (0,333 * 154) = 150$$

Dipetakan kembali menjadi ke dalam array menjadi seperti berikut:

131	101	68
150	127	91
158	144	123

Tabel 3. 2 Tabel Citra Setelah Proses Grayscale

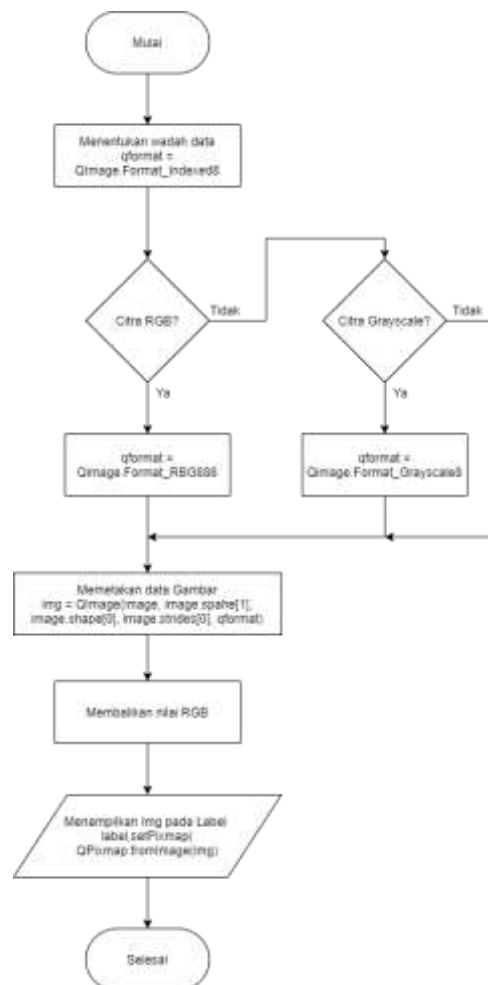
2. Nama operasi : Display image

Tujuan : Untuk menampilkan gambar

Alur Proses : - Menentukan bentuk penyimpanan warna dan data image

- Membuat sebuah image objek dari QImage menggunakan image yang sudah di load atau di proses
- Membalikan nilai RGB jika mungkin
- Menampilkan gambar pada label dengan cara menentukan QPixmap label

Flowchart:



Gambar 3. 5 Flowchart Display Image

Kode:

```
def display(self, windows):
    qformat = QImage.Format_Indexed8
    if len(self.image.shape) == 3:
        if (self.image.shape[2]) == 4:
            qformat = QImage.Format_RGBA8888 # Format_Grayscale8
        else:
            qformat = QImage.Format_RGB888 # Format_Grayscale16
    img = QImage(self.image, self.image.shape[1], self.image.shape[0], self.image.strides[0], qformat)
    img = img.rgbSwapped()
    if windows == 1:
        self.imageOut.setPixmap(QPixmap.fromImage(img))
        cv.imwrite('temp.png', self.image)
```

Gambar 3. 6 Kode Memunculkan Citra ke GUI

Penjelasan Kode:

1. Membuat sebuah wadah untuk menyimpan data warna pada gambar dan memberi wadah tersebut nama qformat
2. Jika gambar tersebut memiliki warna dan warna tersebut mempunyai 3 Warna maka wadah tersebut memiliki format RGB888, jika warna yang dimiliki citra hanya 1, maka wadah tersebut memiliki format Grayscale8 untuk 8bit grayscale atau Grayscale16 untuk 16bit grayscale
3. Buat sebuah image dari komponen komponen `self.image` menggunakan parameter – parameter berikut `QImage(image, ukuran_lebar_citra, ukuran_panjang_citra, pembulatan nilai ke nilai terkecil, wadah_data_qformat)`
4. Balikan nilai RGB pada citra jika ada dengan kode `img.rgbSwapped()`
5. Simpan gambar pada label yang telah di sediakan dengan perintah `label.setPixmap(QPixmap.fromImage(image))`
6. Lalu simpan gambar tersebut sebagai image sementara untuk di baca oleh OCR.

QImage::Format_Indexed8	3	The image is stored using 8-bit indexes into a colormap.
QImage::Format_RGB888	13	The image is stored using a 24-bit RGB format (8-8-8).
QImage::Format_RGB444	14	The image is stored using a 16-bit RGB format (4-4-4). The unused bits are always zero.
QImage::Format_ARGB4444_Premultiplied	15	The image is stored using a premultiplied 16-bit ARGB format (4-4-4-4).
QImage::Format_RGBX8888	16	The image is stored using a 32-bit byte-ordered RGB(x) format (8-8-8-8). This is the same as the Format_RGBA8888 except alpha must always be 255. (added in Qt 5.2)
QImage::Format_RGBA8888	17	The image is stored using a 32-bit byte-ordered RGBA format (8-8-8-8). Unlike ARGB32 this is a byte-ordered format, which means the 32bit encoding differs between big endian and little endian architectures, being respectively (0xRRGGBBAA) and (0xAABBGGR). The order of the colors is the same on any architecture if read as bytes 0xRR,0xGG,0xBB,0xAA. (added in Qt 5.2)
QImage::Format_Grayscale8	24	The image is stored using an 8-bit grayscale format. (added in Qt 5.5)
QImage::Format_Grayscale16	28	The image is stored using an 16-bit grayscale format. (added in Qt 5.13)

Gambar 3. 7 List Wadah untuk menyimpan data Citra

Perbandingan Citra



Gambar 3. 8 Gambar Citra Awal (2)



Gambar 3. 9 Gambar ketika citra pada GUI

Analisis:

Image yang diinputkan, akan ditampilkan di dalam GUI dan terlihat lebih mengecil untuk menyesuaikan ukuran dengan label

Perhitungan Studi Kasus: -

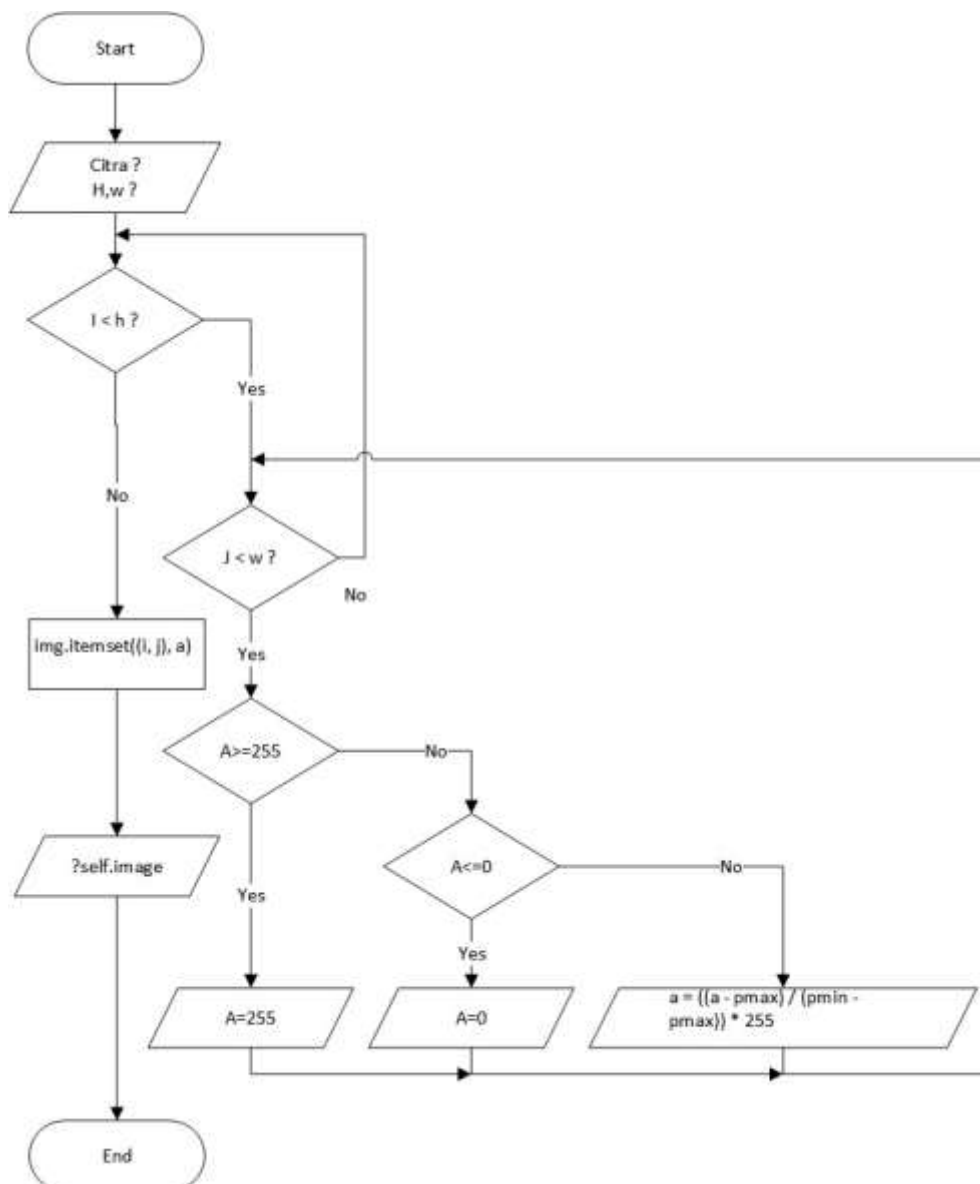
3. Nama Operasi : Pengontrasan

Tujuan : Untuk mengubah nilai kontras pada gambar

Alur Operasi : - Pertama sistem akan mengambil ukuran citra

- Kemudian sistem akan membuat gambar yang serupa dengan ukuran gambar aslinya, tetapi semua nilai pixel nya nol.
- Sistem akan melakukan looping untuk mengubah nilai kontras pada gambar dengan cara (nilai pixel gambar * nilai kontras)
- Hasil dari looping tersebut akan dimasukkan ke gambar yang sudah dibuat sebelumnya.

Flowchart:



Gambar 3. 10 Flowchart Proses Pengaturan Kontras

Kode :

```
def kontrasImage(self):
    contrast = self.kontrastSlider.value() / 1000
    self.label_2.setText(str(contrast))
    h, w = self.image2.shape[:2]
    biner_ZeroTwo = np.zeros((h, w), np.uint8)
    for i in range(h):
        for j in range(w):
            a = self.image2.item(i, j)
            b = math.ceil(a * contrast)
            if b > 255:
                b = 255
            elif b < 0:
                b = 0
            else:
                b = b
            biner_ZeroTwo.itemset((i, j), b)
    self.image = biner_ZeroTwo
    self.image3 = biner_ZeroTwo
    self.display(1)
```

Gambar 3. 11 Kode Pengaturan Kontrast Citra

Penjelasan Kode :

1. Nilai variabel contrast berasal dari nilai slider dibagi 1000
2. Self.label_2 untuk menampilkan nilai slider kontras
3. biner_ZeroTwo untuk membuat gambar berdasarkan ukuran citra asli, tetapi nilai pixel nya nol
4. for I in range(h) untuk looping nilai y, for j in range(w) untuk loop nilai x
5. a sebagai variabel yang berisi nilai pixel dari gambar
6. b sebagai nilai pixel yang baru, dimana nilai nya merupakan $a * \text{contrast}$
7. Jika b lebih besar dari 255 maka $b = 255$
8. Jika b kurang dari 0 maka $b = 0$
9. Jika b ada diantara 0 dan 255 maka $b = b$
10. biner_ZeroTwo.itemset untuk mengganti nilai yang ada pada citra dengan nilai b
11. self.image untuk variabel citra yang baru
12. self.image3 untuk mengkopi gambar
13. self.display untuk menampilkan citra hasil

Perbandingan citra :



Gambar 3. 12 Gambar Citra Awal (3)



Gambar 3. 13 Gambar Citra setelah Proses Pengaturan Kontras

Analisis:

Setelah dilakukan operasi pengontrasan, gambar menjadi berubah tingkat kecerahannya, begitu pun dengan tingkat kontrasnya (tergantung dengan konstanta contrast yang di gunakan), begitu pun nilai histogram nya akan berubah

Perhitungan Studi Kasus:

Hasil Citra Grayscale:

131	101	68
150	127	91
158	144	123

Tabel 3. 3 Sampel Citra Grayscale

Lakukan proses pengaturan contrast, dengan rumus:

$$f(x,y)' = f(x,y) * c \text{ dengan konstanta } c = 2,28$$

$$f(0,0)' = 131 * 2,28 = 298$$

$$f(0,1)' = 101 * 2,28 = 231$$

$$f(0,2)' = 68 * 2,28 = 156$$

$$f(1,0)' = 150 * 2,28 = 342$$

Lalu clip semua nilai pada citra agar tidak melebihi angka 255 atau kurang dari 0

Setelah itu, petakan hasilnya kedalam sebuah array, hingga seperti berikut:

255	231	156
255	255	208
255	255	255

Tabel 3. 4 Hasil dari Proses Pengaturan Kontras

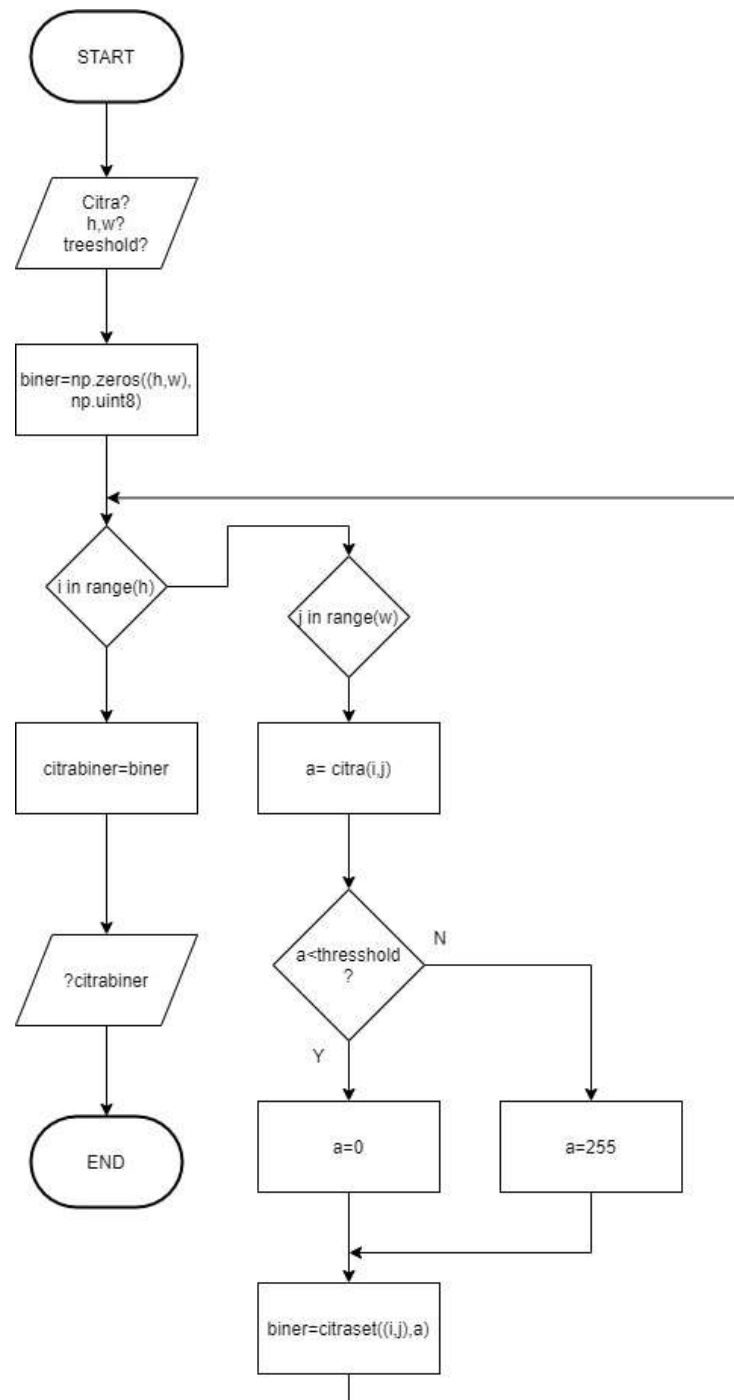
4. Nama Oprasi: Thresholding

Tujuan : Untuk mengubah citra menjadi citra biner

Alur Operasi : - Pertama sistem mengambil ukuran citra gambar asli

- Ambil nilai sembarang untuk batas threshold, seperti contohnya threshold = 84.
- Buat sebuah array kosong dengan ukuran yang sama dengan citra awal
- Cek setiap nilai pixel pada citra, apakah nilai pixel citra tersebut lebih dari threshold?
- Jika ya, maka warna citra tersebut menjadi putih (255)
- Jika tidak, maka warna citra tersebut menjadi hitam (0)
- Simpan semua nilai citra tersebut ke array kosong yang telah di buat sebelumnya
- Buat image baru menggunakan array yang telah di isi oleh nilai biner tadi.

Flowchart :



Gambar 3. 14 Flowchart Proses Thresholding

Kode :

```

def binerImage(self):
    tresshold = self.binerSlider.value()
    self.label.setText(str(tresshold))
    h, w = self.image2.shape[:2]
    biner_Zero = np.zeros((h, w), np.uint8)
    for i in range(h):
        for j in range(w):
            a = self.image3.item(i, j)
            if a < tresshold:
                a = 0
            else:
                a = 255
            biner_Zero.itemset((i,j), a)
    self.image = biner_Zero
    self.display(1)

```

Gambar 3. 15 Kode Thresholding Citra

Penjelasan Kode:

1. Membuat sebuah wadah untuk menyimpan data warna pada gambar dan memberi wadah tersebut nama qformat
2. Jika gambar tersebut memiliki warna dan warna tersebut mempunyai 3 Warna maka wadah tersebut memiliki format RGB888, jika warna yang dimiliki citra hanya 1, maka wadah tersebut memiliki format Grayscale8 untuk 8bit grayscale atau Grayscale16 untuk 16bit grayscale
3. Buat sebuah image dari komponen komponen `self.image` menggunakan parameter – parameter berikut QImage(image, ukuran_lebar_citra, ukuran_panjang_citra, pembulatan nilai ke nilai terkecil, wadah_data_qformat)
4. Balikan nilai RGB pada citra jika ada dengan kode `img.rgbSwapped()`
5. Simpan gambar pada label yang telah di sediakan dengan perintah `label.setPixmap(QPixmap.fromImage(image))`
6. Lalu simpan gambar tersebut sebagai image sementara untuk di baca oleh OCR.

Perbandingan Citra:



Gambar 3. 16 Gambar Citra Awal (4)



Gambar 3. 17 Gambar Citra setelah Proses Thresholding

Analisis:

Karena biner hanya memiliki 2 nilai, yakni 0 dan 1, maka hasil citra hanya memiliki 2 warna, yaitu hitam dan putih, hal itu dikarenakan pixel yang melebihi nilai threshold akan berwarna putih, sedangkan warna yang kurang dari threshold akan berwarna hitam. Karena citra hanya memiliki 2 warna, kontras yang di miliki citra sangat tinggi.

Perhitungan Studi Kasus:

Hasil Citra Kontras:

255	231	156
255	255	208
255	255	255

Tabel 3. 5 Sampel Citra Kontras

Lakukan proses Thresholding pada sampel berikut dengan threshold = 210.
Rumus nya adalah

$$f(x, y)' = \begin{cases} 255, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases}$$

$$f(0, 0) = 255 \geq T = 255$$

$$f(0, 1) = 231 \geq T = 255$$

$$f(0, 2) = 156 < T = 0$$

$$f(1, 0) = 255 \geq T = 255$$

Petakan semua hasil tersebut pada array hingga seperti berikut hasilnya:

255	255	0
255	255	0
255	255	255

Tabel 3. 6 Hasil dari Proses Thresholding

DAFTAR PUSTAKA

- G, A. R., Tandra, A., Susanto, I., Harefa, J., & Chowanda, A. (2019). Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application. *Procedia Computer Science* (pp. 499 - 505). Bali: Elsevier B. V.
- Ryan, M., & Hanafiah, N. (2015). An Examination of Character Recognition on ID card using Template Matching Approach. *Procedia Computer Science* (pp. 520 - 529). Jakarta: Elsevier B. V.
- Saputra, K. D., Rahmaastri, D. A., Setiawan, K., Surnayi, D., & Purnama, Y. (2019). Mobile Financial Management Application using Google Cloud Vision API. *Procedia Computer Science* (pp. 596 - 604). Yogyakarta: Elsevier B. V.
- Tarigan, J., Nadia, Diedan, R., & Suryana, Y. (2017). Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm . *Procedia Computer Science* (pp. 365 - 372). Bali: Elsevier B. V.
- Yang, C.-S., & Yanga, Y.-H. (2017). Improved Local Binary Pattern for Real Scene Optical Character Recognition. *Pattern Recognition Letters*, 1 - 9.