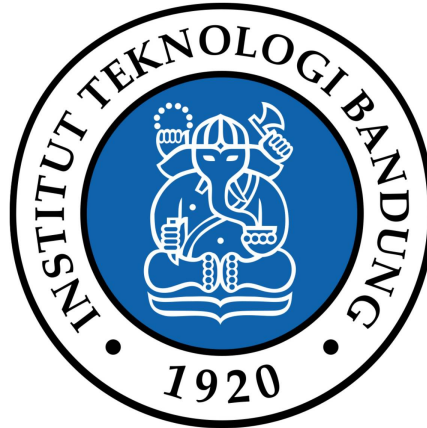


UJIAN AKHIR SEMESTER (UAS)

II3150 - Sistem Multimedia

Pengolahan Citra menggunakan Bahasa Pemrograman Python



Dipersiapkan oleh:

Vhydie Gilang Christianto The

18218016

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Program Studi Sistem dan Teknologi Informasi STEI – ITB	Mata Kuliah	Halaman
		II3150	24
			19/12/2020

Pemilihan Soal

Berdasarkan petunjuk pengerjaan soal yang diberikan, diperoleh informasi bahwa pengerjaan dilakukan dengan memilih 3 dari 5 soal yang diberikan. Berdasarkan pertimbangan yang akan dijelaskan berikutnya, berikut adalah 3 soal yang saya pilih:

1. [Soal Nomor 1] Ciptakan citra dari teks “NIM - Nama lengkap” dan berikan juga “tanggal dan waktu” penciptaan file tersebut. Kemudian sisipkan citra tersebut ke dalam foto Anda dan buatlah menjadi format GIF. Ini salah satu contoh hasilnya, punya Anda lokasi teksnya harus berbeda.
2. [Soal Nomor 2] Ciptakan citra dari teks “NIM - Nama lengkap” dan berikan juga “tanggal waktu” penciptaan file tersebut. Kemudian sisipkan citra tersebut ke dalam foto Anda dan tambahkan filter embossing pada gambar tersebut. Ini salah satu contoh hasilnya, punya Anda lokasi teksnya harus berbeda.
3. [Soal Nomor 3] Ambil gambar background pemandangan sekitar rumah Anda menggunakan smartphone atau apapun. Kembali menggunakan Python, import gambar pemandangan tersebut dan sisipkan teks “Nim - Nama lengkap” dan berikan juga “tanggal waktu” penciptaan file tersebut. Kemudian kombinasikan hasil gambar tersebut dengan foto Anda (Foreground) dengan menggunakan 3 variasi nilai alpha channel yang berbeda, kemudian tampilkan hasilnya.

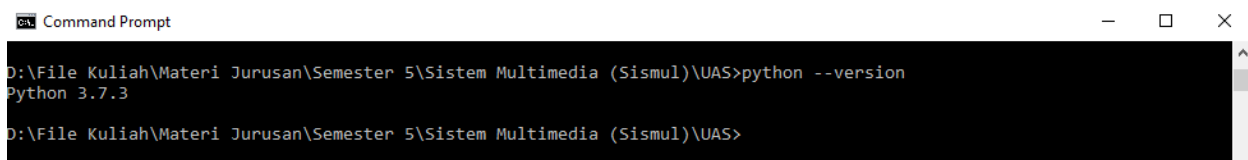
Ketiga soal tersebut saya pilih berdasarkan berbagai pertimbangan dari berbagai hal yaitu:

1. Ketiga soal tersebut memiliki *scope* yang sama yang berkaitan dengan pengolahan citra dari hasil pencuplikan satu atau beberapa *frame* sehingga proses pembelajaran hanya difokuskan ke satu hal. Hal ini akan membawakan pembahasan yang jauh lebih komprehensif dibanding mempelajari banyak hal namun tidak mendalam.
2. Ketiga soal ini akan memanipulasi citra sedemikian rupa dan tidak membutuhkan perangkat atau *resources* yang “berat” sehingga dapat dikerjakan dalam masa waktu yang diberikan, sambil mengerjakan beberapa ujian akhir lainnya.
3. Saya sendiri jauh lebih tertarik dengan citra baik berupa gambar maupun video, dibanding audio, karena saya sangat minim pengetahuan tentang audio dan proses belajar Audacity akan mengambil waktu yang cukup panjang.

Python dan *Library*

Python adalah salah satu bahasa pemrograman yang paling sering digunakan akhir-akhir ini. Biasanya, saya menggunakan Jupiter Notebook untuk melakukan pemrograman dan visualisasi citra menggunakan Python karena jauh lebih mudah dari segi penggunaannya. Namun, kali ini saya menggunakan Python melalui IDE Visual Studio karena kode yang dihasilkan akan jauh lebih sistematis dan terarah.

Terkait versi Python yang saya gunakan, saya mengecek dengan melakukan *command* **python --version** karena saya lupa telah mengunduh versi yang mana. Berikut adalah tampilannya.

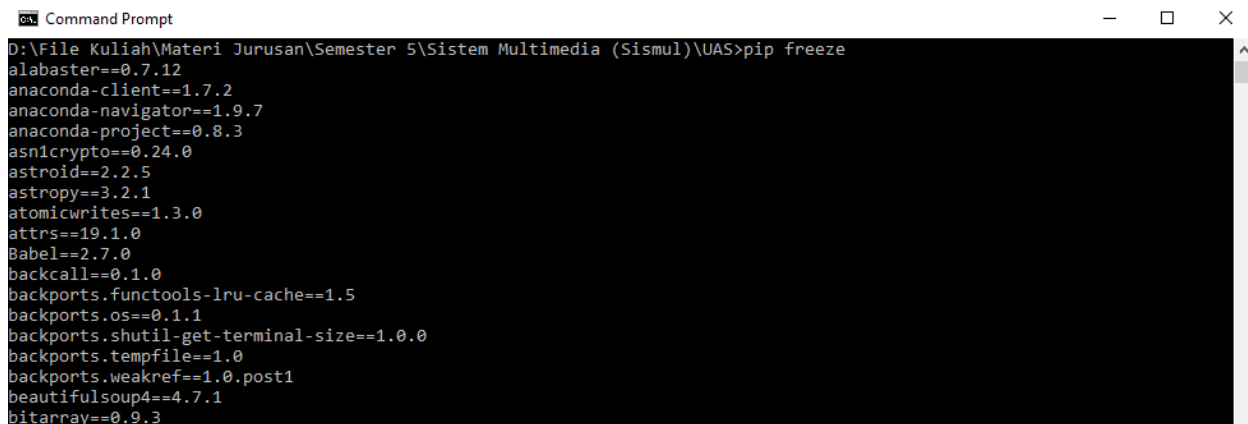


```
Command Prompt
D:\File Kuliah\Materi Jurusan\Semester 5\Sistem Multimedia (Sismul)\UAS>python --version
Python 3.7.3
D:\File Kuliah\Materi Jurusan\Semester 5\Sistem Multimedia (Sismul)\UAS>
```

Gambar 1 Pengecekan Versi Python yang Digunakan

Dari gambar di atas, terlihat bahwa saya menggunakan versi **Python 3.7.3**. Ini bukan versi yang terbaru karena saya telah menginstallnya dari 2018 sementara versi terbaru adalah Python 3.9.1. Namun, fungsionalitas dalam Python versi ini tetap sama pada beberapa hal, apalagi dalam hal pengolahan citra, sehingga tidak ada urgensi untuk meningkatkan versi Python ini.

Dalam keberjalanannya, saya telah menggunakan beberapa *library* dan *extension*. Saya mengecek hal tersebut dengan menggunakan *command* **pip freeze** yang akan menampilkan semua *extension* yang telah terinstall dalam perangkat komputer saya. Berikut adalah tampilannya.

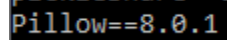


```
Command Prompt
D:\File Kuliah\Materi Jurusan\Semester 5\Sistem Multimedia (Sismul)\UAS>pip freeze
alabaster==0.7.12
anaconda-client==1.7.2
anaconda-navigator==1.9.7
anaconda-project==0.8.3
asn1crypto==0.24.0
astroid==2.2.5
astropy==3.2.1
atomicwrites==1.3.0
attrs==19.1.0
Babel==2.7.0
backcall==0.1.0
backports.functools-lru-cache==1.5
backports.os==0.1.1
backports.shutil-get-terminal-size==1.0.0
backports.tempfile==1.0
backports.weakref==1.0.post1
beautifulsoup4==4.7.1
bitarray==0.9.3
```

Gambar 2 Pengecekan Versi *Library* yang Digunakan

Namun, tidak semua *extension* dan *library* tersebut saya gunakan. Saya hanya menggunakan beberapa hal, terkhusus dalam pemrosesan citra ini yang rinciannya adalah sebagai berikut.

1. **Pillow [versi 8.0.1]**, untuk mengolah citra dan memanipulasi beberapa hal yang berkaitan dengan citra. Berikut buktinya.

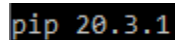


```
Pillow==8.0.1
```

Gambar 3 Pengecekan Versi Pillow yang Digunakan

Di dalamnya, digunakan beberapa *library* yang sangat menunjang dalam proses manipulasi citra yang terdiri dari:

- a. **Image**, untuk pengolahan citra secara umum,
 - b. **ImageDraw**, untuk mengatur penggambaran citra dari *scratch* yang bisa berguna untuk membuat citra atau teks dalam bentuk citra,
 - c. **ImageFont**, untuk mengatur ukuran citra yang berupa teks, dan
 - d. **ImageFilter**, untuk membantu *user* dalam menyediakan beberapa filter yang telah disediakan Python seperti Emboss.
2. **Pip [versi 20.3.1]**, ekstensi Python yang sangat berguna dalam banyak hal mulai dari mengunduh hingga menginstall beberapa *library*. Berikut buktinya.



```
pip 20.3.1
```

Gambar 4 Pengecekan Versi Pip yang Digunakan

3. Selain kedua *extension* tersebut, terdapat pula beberapa *library* bawaan Python yang digunakan dalam pemrosesan citra ini yang terdiri atas:
 - a. **datetime**, untuk mengambil waktu saat ini yang tercatat dalam perangkat untuk nantinya dijadikan citra sebagai waktu pembuatan citra tersebut, dan
 - b. **imageio [versi 2.5.0]**, untuk menyimpan citra dan membuatnya dalam bentuk .GIF dengan durasi dan berbagai atribut tertentu.

Program

Pada bagian ini, akan dilampirkan program Python yang digunakan untuk menyelesaikan masing-masing persoalan yang diberikan. Pembahasan kode akan diteruskan pada bagian selanjutnya. Program ini juga dapat diakses secara langsung via GitHub berikut.

https://github.com/Vhydie/UAS_SistemMultimedia

Berikut adalah cuplikan program terkait.

1. Soal Nomor 1, pembuatan GIF.

Berikut programnya yang dapat diakses via GitHub melalui *file* UAS_Nomor_1.py.

```
# Definition
from PIL import Image, ImageDraw, ImageFont
import datetime, imageio

# Initialization
# Import image dan mengatur size image agar tetap sama
imgAsli_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
img_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
imgAsli_2 = Image.open("Sistem Multimedia.png").resize((960, 960))
# Membuat image baru dengan latar belakang warna hitam
imgAsli_3 = Image.new('RGB', (960, 960), color = 'black')
img_3_1 = Image.new('RGB', (960, 960), color = 'black')
img_3_2 = Image.new('RGB', (960, 960), color = 'black')
img_3_3 = Image.new('RGB', (960, 960), color = 'black')
# Mengurus waktu pembuatan citra
date_string = datetime.datetime.now().strftime("%d/%m/%Y - %H:%M:%S")
# Pesan yang akan dimasukkan dalam citra
msg_1 = "18218016 - Vhydie Gilang Christianto The\n" + date_string
msg_2 = "Terima kasih\ntelah mengajar selama 1 semester ini"
msg_3 = "Sehat selalu\nPak Yusep, Kang Rizki, Teh Chessa!"

# Main Code
# Menggambar kembali image
draw1 = ImageDraw.Draw(img_1)
draw3_1 = ImageDraw.Draw(img_3_1)
draw3_2 = ImageDraw.Draw(img_3_2)
```

```

draw3_3 = ImageDraw.Draw(img_3_3)
# Mengambil ukuran text
w, h = draw1.textsize(msg_1)
# Menambahkan text dalam image
draw1.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw1.text((w/2,800), msg_1, fill="white", font = ImageFont.truetype("arial.ttf", 40))
draw3_1.text((w/2,240), msg_1, fill="white", font = ImageFont.truetype("arial.ttf", 40))
draw3_2.text((w/2,360), msg_2, fill="white", font = ImageFont.truetype("arial.ttf", 40))
draw3_3.text((w/2,480), msg_3, fill="white", font = ImageFont.truetype("arial.ttf", 40))
# Memasukkan image dalam list yang akan ditampilkan
images = [imgAsli_2, imgAsli_2, imgAsli_1, img_1, imgAsli_3, img_3_1, img_3_2, img_3_3]
# Mengeluarkan dan menghasilkan GIF
imageio.mimsave('18218016_Nomor1.GIF', images, duration = 0.7, loop = 0)

```

2. **Soal Nomor 2**, pembuatan citra dengan penambahan filter *embossing*.

Berikut programnya yang dapat diakses via GitHub melalui *file* UAS_Nomor_2.py.

```

# Definition
from PIL import Image, ImageDraw, ImageFont, ImageFilter
import datetime, imageio

# Initialization
# Import image dan mengatur size image agar tetap sama
img_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
# Mengurus waktu pembuatan citra
date_string = datetime.datetime.now().strftime("%d/%m/%Y - %H:%M:%S")
# Pesan yang akan dimasukkan dalam citra
msg_1 = "18218016 - Vhydie Gilang Christianto The\n" + date_string

# Main Code
# Menggambar kembali image
draw1 = ImageDraw.Draw(img_1)
# Mengambil ukuran text
w, h = draw1.textsize(msg_1)
# Menambahkan text dalam image
draw1.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw1.text((w/2,800), msg_1, fill="white", font = ImageFont.truetype("arial.ttf", 40))

```

```
# Menyimpan Image
img_1.save("18218016_Nomor2_Normal.png")
# Membuat Filter Embossing
img_emboss = img_1.filter(ImageFilter.EMBOSS)
# Menyimpan Image Hasil Embossing
img_emboss.save("18218016_Nomor2_Emboss.png")
```

3. **Soal Nomor 3**, penggabungan *foreground* dan *background* sesuai Alpha Channel.
Berikut programnya yang dapat diakses via GitHub melalui *file* UAS_Nomor_3.py.

```
# Definition
from PIL import Image, ImageDraw, ImageFont
import datetime, imageio

# Initialization
# Import image dan mengatur size image agar tetap sama
# Untuk background yakni pemandangan
background_1 = Image.open("Pemandangan.jpg").resize((960, 960))
background_2 = Image.open("Pemandangan.jpg").resize((960, 960))
background_3 = Image.open("Pemandangan.jpg").resize((960, 960))
# Untuk foreground yakni foto diri
foreground_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
foreground_2 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
foreground_3 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
# Mengurus waktu pembuatan citra
date_string = datetime.datetime.now().strftime("%d/%m/%Y - %H:%M:%S")
# Pesan yang akan dimasukkan dalam citra
msg = "18218016 - Vhydie Gilang Christianto The\n" + date_string

# Membuat alpha channel
# Untuk alpha 0.2 = 0.2 x 256 = 51.2, dibulatkan jadi 51
alpha_02 = Image.new("L", (960, 960), 51)
# Untuk alpha 0.5 = 0.5 x 256 = 128
alpha_05 = Image.new("L", (960, 960), 128)
# Untuk alpha 0.8 = 0.8 x 256 = 204.8, dibulatkan jadi 205
alpha_08 = Image.new("L", (960, 960), 205)
```

```

# Main Code

# Menggambar kembali image
draw_background_1 = ImageDraw.Draw(background_1)
draw_background_2 = ImageDraw.Draw(background_2)
draw_background_3 = ImageDraw.Draw(background_3)
# Mengambil ukuran text
w, h = draw_background_1.textsize(msg)
# Menambahkan text dalam image
draw_background_1.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw_background_2.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw_background_3.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw_background_1.text((w/2,800), msg, fill="red", font = ImageFont.truetype("arial.ttf", 40))
draw_background_2.text((w/2,800), msg, fill="red", font = ImageFont.truetype("arial.ttf", 40))
draw_background_3.text((w/2,800), msg, fill="red", font = ImageFont.truetype("arial.ttf", 40))
# Menyimpan Image Background sebelum ada alpha channel sama sekali
background_1.save("18218016_Nomor3_PemandanganNormal.png")

# Menambahkan alpha channel ke gambar RGB untuk masing-masing gambar
# Untuk kasus pertama, alpha foreground = 0.2, alpha background = 0.8
foreground_1.putalpha(alpha_02)
background_1.putalpha(alpha_08)
# Untuk kasus kedua, alpha foreground = 0.5, alpha background = 0.5
foreground_2.putalpha(alpha_05)
background_2.putalpha(alpha_05)
# Untuk kasus ketiga, alpha foreground = 0.8, alpha background = 0.2
foreground_3.putalpha(alpha_08)
background_3.putalpha(alpha_02)

# Menyatukan Image untuk masing-masing background dan foreground
# Gambar pertama
img_1 = Image.alpha_composite(background_1, foreground_1)
img_1.save("18218016_Nomor3_Kombinasi1.png")
# Gambar kedua
img_2 = Image.alpha_composite(background_2, foreground_2)
img_2.save("18218016_Nomor3_Kombinasi2.png")
# Gambar ketiga
img_3 = Image.alpha_composite(background_3, foreground_3)
img_3.save("18218016_Nomor3_Kombinasi3.png")

```


Pembahasan Program

1. Soal Nomor 1, pembuatan GIF.

Program diawali dengan pendefinisian dan beberapa inisialisasi. Terkhusus untuk soal nomor 1 yakni pembuatan GIF, berikut adalah pendefinisian yang dilakukan.

```
from PIL import Image, ImageDraw, ImageFont
import datetime, imageio
```

Definisi ini telah dibahas sebelumnya pada bagian Python dan *library*. Hanya digunakan 5 *library* yang terdiri dari 3 *library* hasil install dari *extension* lain (Image, ImageDraw, dan ImageFont) dan 2 *library* bawaan Python yang dapat langsung digunakan tanpa mengunduh hal tambahan (datetime dan imageio).

Program dilanjutkan ke bagian inisialisasi yang terpecah lagi atas beberapa bagian. Pertama-tama, hal yang harus dilakukan adalah memasukkan gambar diri dan gambar sistem multimedia yang telah dibuat sebelumnya. Hal ini akan dijadikan sebagai sebuah *frame* dalam GIF yang diurutkan sedemikian rupa. Berikut adalah gambar keduanya.



Gambar 5 Foto Diri dan Desain Sistem Multimedia yang Digunakan sebagai Input
Berikut adalah kode untuk memasukkan gambar yang sekaligus mengatur ulang ukuran gambar yang dimaksud.

```
imgAsli_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
img_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
imgAsli_2 = Image.open("Sistem Multimedia.png").resize((960, 960))
```

Kedua gambar tersebut dibuka dengan menggunakan *method* `open()`. Agar penggunaan gambar dan penampilan GIF memiliki ukuran *frame* yang sama, digunakan *method* `resize()` untuk mengubah ukuran gambar sehingga ketika digunakan memiliki *frame* yang sama.

Program dilanjutkan dengan membuat citra baru untuk menampilkan pesan berupa teks ucapan terima kasih kepada tenaga pendidik Sistem Multimedia. Berikut adalah kodenya.

```
imgAsli_3 = Image.new('RGB', (960, 960), color = 'black')
img_3_1 = Image.new('RGB', (960, 960), color = 'black')
img_3_2 = Image.new('RGB', (960, 960), color = 'black')
img_3_3 = Image.new('RGB', (960, 960), color = 'black')
```

Pada kode diatas, terlihat bahwa dibuat 4 citra yang memiliki ukuran *frame* yang sama dengan masukan gambar sebelumnya. Keempat citra ini belum terisi apapun sehingga hanya terlihat sebagai gambar hitam polos. Pada bagian ini, penciptaan citra hanya menggunakan RGB Channel dan tidak menambahkan Alpha Channel karena tidak ada permainan dari segi transparansi dalam GIF yang akan dibuat. Pembuatan ini menggunakan *method* `new()`.

Inisialisasi kemudian dilanjutkan dengan pengambilan tanggal dan waktu pembuatan citra dari perangkat yang sedang digunakan. Berikut adalah kodenya.

```
date_string = datetime.datetime.now().strftime("%d/%m/%Y - %H:%M:%S")
```

Berdasarkan kode di atas, digunakan *method* `datetime.now()` untuk mengambil tanggal dan waktu saat ini. Tanggal dan waktu tersebut lalu diubah menjadi *string* yang akan dimasukkan dalam teks yang akan ditampilkan dalam citra. Pengubahan ini dilakukan dengan menggunakan *method* `strftime()` sesuai format tertentu yang ingin ditampilkan pada citra. Dalam penggunaan kode di atas, %d menunjukkan *day* (tanggal), %m menunjukkan *month* (bulan), %Y menunjukkan *year* (tahun), %H menunjukkan *hour* (jam), %M menunjukkan *minute* (menit), sedangkan %S menunjukkan *second* (detik).

Akhir dari bagian inisialisasi adalah proses inisialisasi pesan yang akan ditampilkan pada masing-masing *frame*. Pada GIF kali ini, saya menampilkan 3 pesan berbeda dengan kode sebagai berikut.

```
msg_1 = "18218016 - Vhydie Gilang Christianto The\n" + date_string
msg_2 = "Terima kasih\ntelah mengajar selama 1 semester ini"
msg_3 = "Sehat selalu\nPak Yusep, Kang Rizki, Teh Chessa!"
```

Pesan pertama yang akan ditampilkan dalam citra, akan menggunakan *string* yang telah di-*convert* dari hasil pengambilan tanggal dan waktu yang telah dibahas sebelumnya. Selain itu, pesan kedua dan pesan ketiga hanyalah *string* biasa yang ingin ditampilkan. Penggunaan `\n` digunakan sebagai “*enter*” untuk membuat baris baru dalam *string* tersebut.

Setelah proses inisialisasi selesai, proses dilanjutkan dengan penciptaan sebuah variabel `draw`. Berikut adalah programnya.

```
draw1 = ImageDraw.Draw(img_1)
draw3_1 = ImageDraw.Draw(img_3_1)
draw3_2 = ImageDraw.Draw(img_3_2)
draw3_3 = ImageDraw.Draw(img_3_3)
```

Variabel `draw` ini diciptakan agar gambar yang dimaksud dapat digambar kembali atau bahkan ditambahkan komponen lainnya seperti teks. Pada program kali ini, hanya dibuat 4 variabel `draw` karena hanya 4 gambar saja yang akan dimanipulasi dengan menambahkan teks di dalamnya.

Perlu diketahui bahwa variabel `draw3_1`, `draw3_2`, dan `draw3_3` hanya akan digunakan untuk menambahkan teks dalam sebuah citra hitam polos yakni `img_3_1`, `img_3_2`, dan `img_3_3`. Namun, variabel `draw1` akan digunakan untuk menambahkan teks (`msg_1`) ke dalam citra gambar diri (`img_1`) sehingga diperlukan kode untuk mengambil ukuran teks tersebut yang nantinya akan digunakan untuk menempatkan teks tersebut dalam citra. Berikut adalah kode pengambilan ukuran teks tersebut.

```
w, h = draw1.textsize(msg_1)
```

Terlihat bawah digunakan *method* `textsize()` untuk mengambil ukuran pesan ketika telah dibuat oleh Python, lalu dimasukkan dalam variabel `w` (*width*) dan `h` (*height*).

Proses dilanjutkan dengan menambahkan teks ke dalam citra yang spesifik untuk nantinya ditampilkan dalam GIF. Berikut kodenya.

```
draw1.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw1.text((w/2,800), msg_1, fill="white", font = ImageFont.truetype("arial.ttf", 40))
draw3_1.text((w/2,240), msg_1, fill="white", font = ImageFont.truetype("arial.ttf", 40))
draw3_2.text((w/2,360), msg_2, fill="white", font = ImageFont.truetype("arial.ttf", 40))
draw3_3.text((w/2,480), msg_3, fill="white", font = ImageFont.truetype("arial.ttf", 40))
```

Terlihat bahwa program di atas diawali dengan membuat sebuah *rectangle* sebagai dasar dari pesan yang akan dimasukkan dalam gambar diri sehingga tetap dapat terlihat dengan jelas. Kotak ini dijadikan sebagai *background* dengan ukuran yang spesifik dan warna dasar hitam menggunakan *method* `rectangle()`. Selain itu, penggambaran teks dalam citra juga dilakukan serupa dengan menspesifikkan teks yang akan digambar, menggunakan variabel `msg_n` yang telah didefinisikan sebelumnya, menggunakan *method* `text()`. Teks juga diatur agar berwarna hitam, ukuran 40, dan menggunakan *font* Arial dengan menggunakan variabel `font`. Terlihat juga bahwa dalam penggambaran teks dan kotak sebagai *background*, digunakan variabel `w` (*width*) yang diperoleh dari hasil ekstraksi sebelumnya.

Ketika semua proses manipulasi citra selesai dilakukan, akan dibuat sebuah array yang berisi urutan citra yang akan ditampilkan dalam GIF. Berikut adalah kodenya beserta urutannya.

```
images = [imgAsli_2, imgAsli_2, imgAsli_1, img_1, imgAsli_3, img_3_1, img_3_2, img_3_3]
```

Berdasarkan kode di atas, urutan citra yang akan keluar dari GIF adalah citra sistem multimedia sebanyak 2 *frame*, citra gambar diri hasil ekstraksi sebanyak 1 *frame*, citra gambar diri yang telah ditambahkan dengan teks sebanyak 1 *frame*, citra kosong latar belakang hitam sebanyak 1 *frame*, lalu diakhiri dengan citra kosong latar belakang hitam dengan 3 pesan berbeda, masing-masing pesan ditampilkan dalam 1 *frame*.

Setelah pemrosesan manipulasi citra selesai dilakukan, citra tersebut akan dibuatkan dalam format .GIF dengan kode sebagai berikut.

```
imageio.mimsave('18218016_Nomor1.GIF', images, duration = 0.7, loop = 0)
```

Pada kode di atas, digunakan *method* `mimsave()` untuk menggabungkan citra yang terdapat dalam variabel `images` lalu menyimpannya dengan nama tertentu. Terlihat juga bahwa setiap citra dalam array `images` diatur sedemikian rupa sehingga tidak memiliki *loop* dan durasi pergantian citra berlangsung selama 0.7 milisekon. Hal ini dipilih setelah beberapa kali percobaan dan melihat hasil yang dihasilkan dimana teks harus tetap dapat dibaca oleh penonton. Gambar tersebut akan disimpan dalam direktori program Python yang dibuat karena tidak mendetailkan direktori tujuan penyimpanan GIF tersebut.

2. **Soal Nomor 2**, pembuatan citra dengan penambahan filter *embossing*.

Sama seperti nomor sebelumnya, program diawali dengan pendefinisian dan beberapa inisialisasi. Untuk soal nomor 2 ini, pendefinisian program dilakukan dengan memasukkan beberapa *library* dengan kode sebagai berikut.

```
from PIL import Image, ImageDraw, ImageFont, ImageFilter
import datetime, imageio
```

Pendefinisian ini serupa dengan soal nomor 1, namun ditambahkan *library* ImageFilter hasil unduhan dari *extension* Pillow. Oleh karena itu, terdapat 6 *library* yang terdiri dari 4 *library* hasil install dari *extension* lain (Image, ImageDraw, ImageFont, ImageFilter) dan 2 *library* bawaan Python yang dapat langsung digunakan tanpa mengunduh hal tambahan (datetime dan imageio).

Setelah mendefinisikan program untuk soal nomor 2, proses dilanjutkan dengan membuat inisialisasi program yang serupa dengan nomor 1, namun jauh lebih sederhana karena hanya membuat 1 citra saja. Sama seperti nomor 1, proses inisialisasi dimulai dengan memasukkan citra gambar diri yang telah terlampir pada **Gambar 5**, dengan kode sebagai berikut.

```
img_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
```

Kemudian, inisialisasi dilanjutkan dengan pengambilan tanggal dan waktu pembuatan citra dari perangkat yang sedang digunakan dengan kode sebagai berikut.

```
date_string = datetime.datetime.now().strftime("%d/%m/%Y - %H:%M:%S")
```

Proses inisialisasi ini diakhiri dengan pembuatan pesan dengan kode sebagai berikut.

```
msg_1 = "18218016 - Vhydie Gilang Christianto The\n" + date_string
```

Ketiga bagian kode inisialisasi ini serupa dengan bagian inisialisasi pada nomor 1, baik dari segi penjelasan maupun dari segi karakteristik variabel, namun jauh lebih sederhana dan tidak banyak.

Proses manipulasi citra memiliki proses yang sama dengan nomor 1 yakni dengan menetapkan variabel draw untuk menambahkan teks pada citra, dengan kode berikut.

```
draw1 = ImageDraw.Draw(img_1)
```

Proses lalu dilanjutkan dengan pengambilan ukuran teks sebagai berikut.

```
w, h = draw1.textsize(msg_1)
```

Selanjutnya, proses dilanjutkan dengan penambahan teks dan kotak yang berfungsi sebagai *background*, yang lokasi dan karakteristiknya sama dengan soal nomor 1 sebelumnya.

```
draw1.rectangle(((w/2)-10, 800, 870, 885), fill='black')  
draw1.text((w/2,800), msg_1, fill="white", font = ImageFont.truetype("aria  
1.ttf", 40))
```

Ketiga proses awal manipulasi citra di atas serupa dengan soal nomor 1, namun jauh lebih sederhana, sehingga tidak dijelaskan lebih lanjut.

Proses dilanjutkan dengan menyimpan citra awal sebelum ditambahkan filter, untuk melihat perbandingan sebelum dan setelah ditambahkan filter *embossing*, dengan kode sebagai berikut.

```
img_1.save("18218016_Nomor2_Normal.png")
```

Proses penyimpanan ini akan dilakukan di direktori program Python, namun jauh lebih sederhana dibanding nomor sebelumnya karena hanya menyimpan satu citra menggunakan *method* `save()`.

Proses lalu dilanjutkan dengan menambahkan filter *embossing* pada citra yang sebelumnya telah ditambahkan teks. Berikut adalah kode penambahan filter tersebut.

```
img_emboss = img_1.filter(ImageFilter.EMBOSS)
```

Program di atas menggunakan *method* `filter()` dengan mengubah citra yang telah dibuat sebelumnya menggunakan variabel yang telah disediakan Pillow yakni `EMBOSS`. Agar tidak mengubah gambar aslinya, manipulasi citra menggunakan filter ini dimasukkan ke dalam sebuah variabel baru.

Proses diakhiri dengan menyimpan citra hasil penambahan filter tersebut ke direktori yang sama dengan penyimpanan citra asli dengan kode sebagai berikut.

```
img_emboss.save("18218016_Nomor2_Emboss.png")
```

Dengan demikian, citra hasil penambahan filter *embossing* telah dapat diakses dan nomor 2 telah selesai.

3. **Soal Nomor 3**, penggabungan *foreground* dan *background* sesuai Alpha Channel.

Sama seperti soal sebelum-sebelumnya, program diawali dengan proses definisi sebagai berikut.

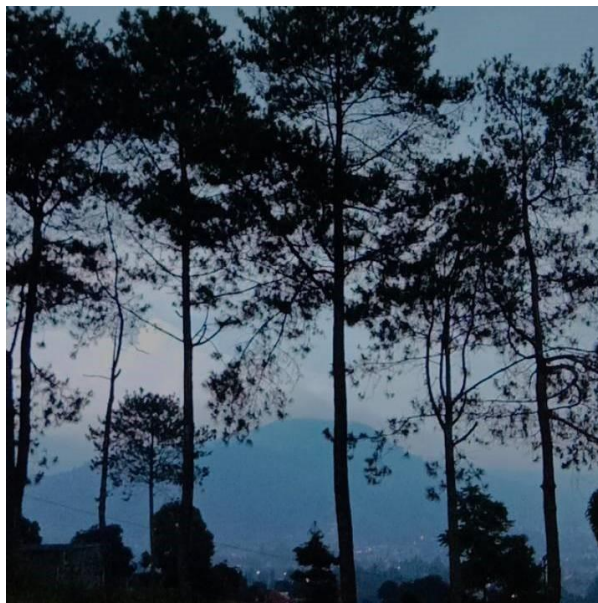
```
from PIL import Image, ImageDraw, ImageFont
import datetime, imageio
```

Definisi ini sama seperti biasanya sehingga tidak perlu dijelaskan lebih lanjut.

Proses dilanjutkan dengan inisialisasi program yang dimulai dengan memasukkan citra ke dalam program dengan kode berikut.

```
background_1 = Image.open("Pemandangan.jpg").resize((960, 960))
background_2 = Image.open("Pemandangan.jpg").resize((960, 960))
background_3 = Image.open("Pemandangan.jpg").resize((960, 960))
foreground_1 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
foreground_2 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
foreground_3 = Image.open("Vhydie Gilang Christianto The.jpg").resize((960, 960))
```

Terdapat 2 citra yang dimasukkan dalam pengerjaan soal ini yakni citra gambar diri sebagai *foreground* yang terlampir pada **Gambar 5**, dan citra pemandangan sebagai *background* yang terlampir berikut ini.



Gambar 6 Gambar Pemandangan yang Digunakan

Proses inisialisasi lalu dilanjutkan dengan pengambilan tanggal dan waktu pembuatan citra, serta teks yang ditambahkan dalam citra tersebut dengan kode sebagai berikut.

```
date_string = datetime.datetime.now().strftime("%d/%m/%Y - %H:%M:%S")
msg = "18218016 - Vhydie Gilang Christianto The\n" + date_string
```

Kode ini sama seperti kode sebelum-sebelumnya.

Proses inisialisasi diakhir dengan pembuatan Alpha Channel yang akan ditambahkan dalam masing-masing citra yang telah dimasukkan dengan kode berikut.

```
alpha_02 = Image.new("L", (960, 960), 51)
alpha_05 = Image.new("L", (960, 960), 128)
alpha_08 = Image.new("L", (960, 960), 205)
```

Perlu diketahui bahwa kedua citra yang dimasukkan (gambar diri dan pemandangan) hanya memiliki RGB Channel dan tidak memiliki Alpha Channel sehingga hal ini perlu diinisialisasikan. Dibuat tiga buah Alpha Channel yakni saat 0.2 ($0.2 \times 256 = 51.2$, dibulatkan jadi 51), saat 0.5 ($0.5 \times 256 = 128$), dan saat 0.8 ($0.8 \times 256 = 204.8$, dibulatkan jadi 205).

Setelah selesai dengan inisialisasi, program dilanjutkan dengan membuat variabel draw dan mengambil *width* dan *height* dari pesan yang akan ditampilkan, sama seperti nomor-nomor sebelumnya dengan kode berikut.

```
draw_background_1 = ImageDraw.Draw(background_1)
draw_background_2 = ImageDraw.Draw(background_2)
draw_background_3 = ImageDraw.Draw(background_3)
w, h = draw_background_1.textsize(msg)
```

Proses ini kemudian dilanjutkan dengan membuat kotak yang akan menjadi *background* teks serta teks itu sendiri yang akan ditambahkan dalam citra yang berperan sebagai *background* dan *foreground* dengan kode berikut.

```
draw_background_1.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw_background_2.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw_background_3.rectangle(((w/2)-10, 800, 870, 885), fill='black')
draw_background_1.text((w/2,800), msg, fill="red", font = ImageFont.truetype("arial.ttf", 40))
draw_background_2.text((w/2,800), msg, fill="red", font = ImageFont.truetype("arial.ttf", 40))
draw_background_3.text((w/2,800), msg, fill="red", font = ImageFont.truetype("arial.ttf", 40))
```


Pada kode tersebut, terlihat bahwa akan dibuat 6 hal yang terdiri dari 3 kotak *background* dan 3 teks pesan yang akan dimasukkan dalam citra yang berperan sebagai *background*.

Sebelum proses kombinasi dilakukan, dilakukan penyimpanan terhadap *background* yang telah diolah sehingga dapat dilihat wujud asli sebelum dikombinasikan, dengan kode berikut.

```
background_1.save("18218016_Nomor3_PemandanganNormal.png")
```

Proses lalu dilanjutkan dengan manipulasi dan penambahan Alpha Channel pada masing-masing citra yang sebelumnya belum ditambahkan dengan kode berikut.

```
foreground_1.putalpha(alpha_02)
background_1.putalpha(alpha_08)
foreground_2.putalpha(alpha_05)
background_2.putalpha(alpha_05)
foreground_3.putalpha(alpha_08)
background_3.putalpha(alpha_02)
```

Pada program di atas, perlu diperhatikan bahwa penambahan Alpha Channel disesuaikan dengan soal yang diberikan. Untuk kombinasi pertama, *foreground* memiliki Alpha channel 0.2 sedangkan *background* 0.8. Untuk kombinasi kedua, *foreground* memiliki Alpha channel 0.5 sedangkan *background* 0.5 juga. Untuk kombinasi ketiga, *foreground* memiliki Alpha channel 0.8 sedangkan *background* 0.2. Penambahan Alpha Channel ini menggunakan *method* `putalpha()`.

Proses diakhiri dengan penggabungan *foreground* dan *background* dengan Alpha Channel yang telah ditetapkan, lalu disimpan dalam sebuah *file* yang dilakukan dengan kode berikut.

```
img_1 = Image.alpha_composite(background_1, foreground_1)
img_1.save("18218016_Nomor3_Kombinasi1.png")
img_2 = Image.alpha_composite(background_2, foreground_2)
img_2.save("18218016_Nomor3_Kombinasi2.png")
img_3 = Image.alpha_composite(background_3, foreground_3)
img_3.save("18218016_Nomor3_Kombinasi3.png")
```

Digunakan *method* `alpha_composite()` untuk menggabungkan kedua *file* tersebut.

Hasil

1. Soal Nomor 1, pembuatan GIF.

Berdasarkan program yang telah dijelaskan sebelumnya, berikut adalah hasil GIF yang saya buat.



Video 1 GIF Hasil Pembuatan Citra

GIF yang dibuat tersebut dapat diakses pada *link* di bawah ini agar dapat ditampilkan maksimal.

https://drive.google.com/file/d/1PTMIekt63mIKPaF_qXK0Pse2Jduv_Be6/view?usp=sharing

Pembuatan GIF ini cukup mudah dimengerti karena pada dasarnya GIF adalah video. Seperti yang telah dipelajari selama 1 semester ini, video adalah gabungan dari beberapa *frame* dalam suatu durasi tertentu sehingga dapat diputar dan terlihat menyatu, yang konsepnya sama seperti GIF. GIF tersebut juga telah memenuhi spesifikasi tugas yang mewajibkan elemen nama, NIM, tanggal, dan waktu pembuatan GIF tersebut.

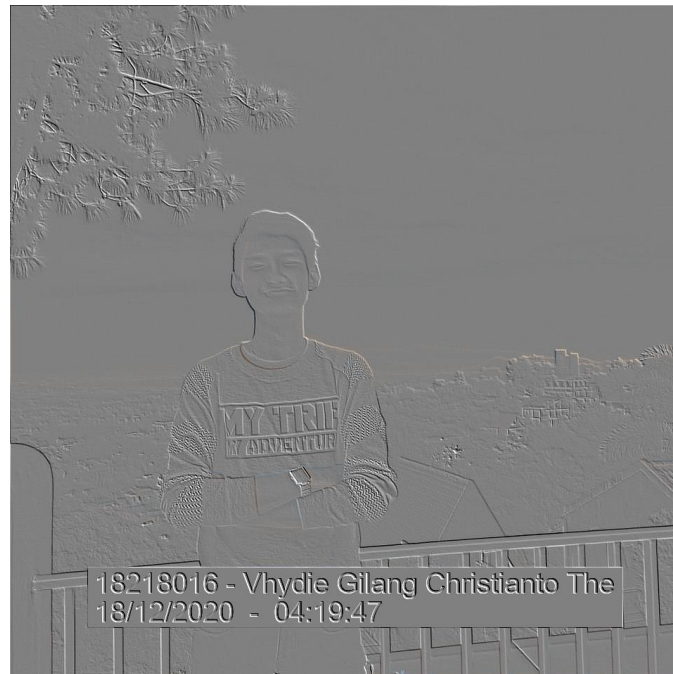
2. **Soal Nomor 2**, pembuatan citra dengan penambahan filter *embossing*.

Pada nomor 2, terdapat beberapa hasil yang diperoleh dari program tersebut yakni gambar pengolahan citra sebelum ditambahkan filter *embossing* sebagai berikut.



Gambar 6 Hasil Pengolah Citra pada Gambar Diri

Kemudian, gambar citra setelah ditambahkan filter *embossing* sebagai berikut.



Gambar 7 Hasil Pengolah Citra pada Gambar Diri dengan Filter *Embossing*

Kedua gambar di atas telah dikompres untuk keperluan dokumen karena batasan *file* sebesar 5 MB. Untuk mengakses citra yang lebih HD dapat mengakses tautan berikut.

Pengolah citra gambar diri:

<https://drive.google.com/file/d/1UDoS5-RUCx66rNMDoFjjYp9zC6hyV7Xw/view?usp=sharing>

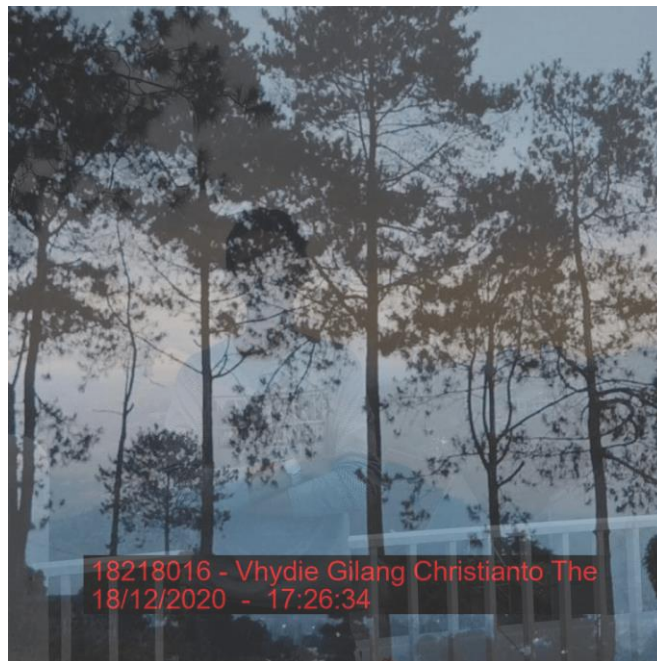
Pengolah citra gambar diri dengan filter *embossing*:

<https://drive.google.com/file/d/1JUiRn6jEpDgBYsdchLrwi7eWmzFne5NI/view?usp=sharing>

Berdasarkan hasil dari kedua gambar tersebut, dapat dilihat dengan jelas perbedaan sebelum dan setelah penambahan filter *embossing*. Salah satu hal yang perlu diperhatikan adalah *sharpness* dari hasil gambar *embossing* tersebut. Hal ini akan dipengaruhi oleh objek yang ada pada citra sebelumnya serta tulisan yang harus memiliki *background*. Bila tulisan/teks disertakan tanpa *background*, akan diperoleh hasil yang kurang memuaskan karena tulisan tidak terlihat dengan jelas.

3. **Soal Nomor 3**, penggabungan *foreground* dan *background* sesuai Alpha Channel.

Pada nomor 3, terdapat beberapa hasil yang diperoleh dari program tersebut yakni gambar pengolahan citra hasil kombinasi citra *foreground* dengan Alpha channel 0.2 dan citra *background* dengan Alpha Channel 0.8, sebagai berikut.



Gambar 8 Hasil Kombinasi Citra Pertama

Lalu, berikut adalah hasil pengolahan citra hasil kombinasi citra *foreground* dengan Alpha channel 0.5 dan citra *background* dengan Alpha Channel 0.5.



Gambar 9 Hasil Kombinasi Citra Kedua

Serta, berikut adalah hasil pengolahan citra hasil kombinasi citra *foreground* dengan Alpha channel 0.8 dan citra *background* dengan Alpha Channel 0.2.



Gambar 10 Hasil Kombinasi Citra Ketiga

Ketiga gambar di atas telah dikompres untuk keperluan dokumen karena batasan *file* sebesar 5 MB. Untuk mengakses citra yang lebih HD dapat mengakses tautan berikut.

Citra kombinasi pertama:

https://drive.google.com/file/d/1Wq42upleffcmI5wYmOjPwty_fKZQ_S0b/view?usp=sharing

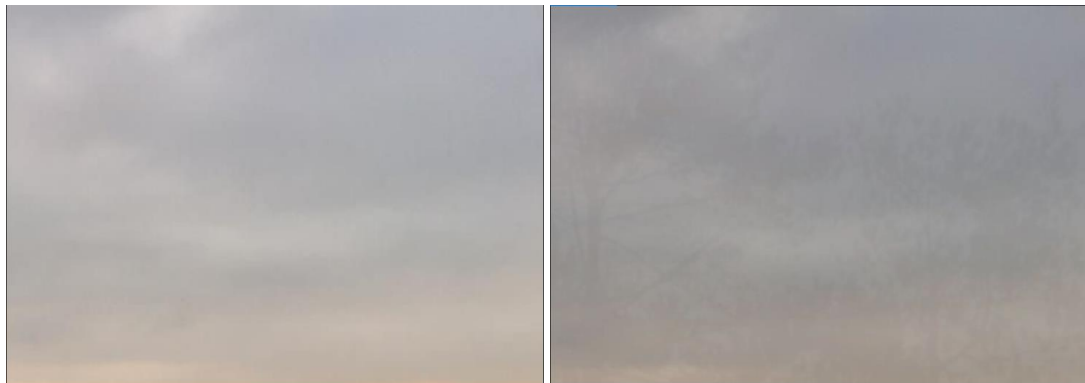
Citra kombinasi kedua:

https://drive.google.com/file/d/15_8v5YG4MxDs43uAzeS2lpkKuQNBuctP/view?usp=sharing

Citra kombinasi ketiga:

https://drive.google.com/file/d/1Flz6WDDy5gn_psxquYl-vk12XIEKc0Kp/view?usp=sharing

Pada ketiga gambar tersebut, terlihat perbedaan yang cukup signifikan hanya dengan mengubah Alpha Channel yang akan menentukan transparansi gambar ketika digabungkan. Bahkan, pada **Gambar 10**, gambar pemandangan dengan Alpha Channel 0.2 sudah hampir tidak terlihat lagi. Namun, jika diperbesar dengan cukup baik, terlihat perbedaan sebagai berikut.



Gambar 11 Perbedaan Gambar Asli dan Hasil Kombinasi Ketiga

Ketika gambar tersebut diperbesar 30%, nampak dengan cukup jelas perbedaan antara gambar asli yang terlihat jelas dan tanpa banyangan sama sekali, serta gambar hasil kombinasi ketiga yang memiliki beberapa banyangan pohon dari gambar pemandangan. Teks yang ditampilkan pada gambar pun kelihatan ketika gambar hasil kombinasi tersebut diperbesar dengan lebih dalam lagi.

KESIMPULAN

Berdasarkan tiga pengolahan citra yang dilakukan menggunakan Bahasa Pemrograman Python, dapat disimpulkan beberapa hal:

1. GIF yang pada dasarnya merupakan video adalah hasil kumpulan citra dalam *frame* yang ditampilkan secara sekuensial dan *iterative* untuk selang waktu tertentu.
2. Filter *embossing* adalah pengolahan citra secara digital yang mengalikan citra asli dengan suatu filter tertentu untuk mendapatkan hasil yang diinginkan. Ketajaman (*sharpness*) dari hasil yang diperoleh dipengaruhi oleh objek yang ada pada citra aslinya serta teknik peletakan teksnya.
3. Kombinasi 2 *file* citra yang berbeda sangat dipengaruhi oleh Alpha Channel yang berada dalam citra tersebut. Alpha Channel ini akan menentukan transparansi suatu citra dibanding citra lainnya. Dengan kata lain, semakin rendah Alpha Channel, semakin transparan pula suatu citra, apalagi ketika dikombinasikan.

DAFTAR PUSTAKA

<https://pillow.readthedocs.io/en/stable/installation.html>