

# **Tugas MK. Grafika Komputer Terapan TA 2023- 2024**

## **Tugas 2 Anti Aliasing**

**Ditulis oleh**

**Dani Iryanto – 152022038  
Afrijal Rusanjung -  
152022052**

**Kelompok 14 Kelas BB**

**Tanggal Penugasan : 23 Oktober 2023  
Tanggal Pengumpulan : 30 Oktober 2023**



**Prodi Informatika  
Fakultas Teknologi  
IndustriInstitut Teknologi  
Nasional Bandung  
2023**

## **1. Konsep Anti-Aliasing**

Anti-aliasing menyempurnakan adegan yang dirender dengan menghilangkan efek bergerigi yang terlihat di sepanjang tepi lurus dan dalam teks. Tepi bergerigi terjadi karena bentuk persegi setiap piksel di layar Anda. Gambar beresolusi lebih rendah memiliki peluang lebih tinggi untuk menampilkan tepi bergerigi karena jumlah pikselnya lebih rendah. Dalam mode Render, Anti-aliasing dicapai dengan menggambar ulang adegan yang dirender beberapa kali, setiap kali membuat gambar bergoyang sedikit, sehingga bagian tepi yang keras diperhalus dengan menampilkan piksel yang sedikit lebih terang berdekatan dengan piksel yang gelap. Mata tertipu karena melihat garis-garis yang lebih lurus dan lekukan yang lebih halus (bukannya tepi yang bergerigi). AGi32 melakukan ini secara otomatis ketika Anti-aliasing diaktifkan saat Anda bernavigasi. Gambar menjadi anti-alias saat navigasi berhenti.

## **2. Jenis-jenis Anti-Aliasing**

Terdapat beberapa jenis Anti Aliasing ( AA ) yang sering ditemui pada game dengan kualitas yang berbeda, yaitu :

### **1. MSAA ( Multisampling Anti-Aliasing)**

MSAA merupakan teknik Anti Aliasing yang paling sering digunakan. Teknik yang diterapkan hanya dilakukan pada bagian frame yang terlihat bergerigi saja, sehingga bisa saja gambar yang dihasilkan masih menyisakkan efek patah-patah di beberapa bagian.

### **2. TXAA ( Temporal Anti-Aliasing )**

TXAA bekerja dengan memproses frame yang telah ditampilkan dan menyempurnakan frame untuk ditampilkan kemudian. TXAA ini mampu membuat gambar yang dihasilkan terlihat kabur.

### **3. FXAA ( Fast Approximate Anti Aliasing )**

FXAA merupakan anti aliasing yang tidak terlalu mempengaruhi performa dan mempunyai keunggulan dalam hal kecepatan karena mampu mengurangi aliasing dalam waktu yang relatif cepat.

### 3. Contoh Anti-Aliasing FXAA

Untuk contoh yang di buat disini menggunakan Anti-Aliasing FXAA (Fast Approximate Anti Aliasing) menggunakan python dan module opencv-python contoh dari codingan beserta penjelasan pada setiap baris code sebagai berikut:

Source Code:

```
# Import library opencv dan numpy
import cv2
import numpy as np

# Bikin fungsi untuk kasih efek FXAA ke gambar
def fxaapisanpisan(gambar, strength=0.5):
    # Atur batas dan batas minimum buat nangkep tepinya
    batas_tepi = 0.125
    batas_tepi_min = 0.0312

    # Atur koefisien buat hitung kecerahan
    kecerahan = np.array([0.299, 0.587, 0.114])

    # Kloning gambar asli
    hasil_fxaa = gambar.copy()

    # Looping lewat setiap piksel gambar, kecuali tepinya
    for y in range(1, gambar.shape[0] - 1):
        for x in range(1, gambar.shape[1] - 1):
            # Hitung kecerahan buat setiap piksel sama tetangganya
            kecerahan_tl = np.dot(gambar[y - 1, x - 1], kecerahan)
            kecerahan_tr = np.dot(gambar[y - 1, x + 1], kecerahan)
            kecerahan_bl = np.dot(gambar[y + 1, x - 1], kecerahan)
            kecerahan_br = np.dot(gambar[y + 1, x + 1], kecerahan)

            # Hitung nilai kecerahan minimum dan maksimum
            kecerahan_min = min(kecerahan_tl, kecerahan_tr, kecerahan_bl, kecerahan_br)
            kecerahan_max = max(kecerahan_tl, kecerahan_tr, kecerahan_bl, kecerahan_br)

            # Hitung gradasi
            gradasi = (kecerahan_max - kecerahan_min) / kecerahan_max

            # Kasih efek FXAA kalo gradasi dan beda kecerahan memenuhi syarat
            if gradasi < batas_tepi and abs(kecerahan_tl - kecerahan_br) >= batas_tepi_min and
abs(kecerahan_tr - kecerahan_bl) >= batas_tepi_min:
                hasil_fxaa[y, x] = gambar[y, x] * strength + 0.5 * (gambar[y - 1, x] + gambar[y, x -
1] + gambar[y, x + 1] + gambar[y + 1, x]) * (1 - strength)

        return hasil_fxaa

# Baca gambar dari path yang ditentukan
img = cv2.imread('G:/apple.png') # !Catatan! Path gambar harus sama dengan path folder gambar
tersimpan

# Kasih efek FXAA ke gambar dengan memanggil fungsi yang sudah di buat di awal
hasil_1 = fxaapisanpisan(img)

# Tunjukkan gambar yang udah diberi efek FXAA dan gambar yang asli
cv2.imshow('Gambar dengan Efek FXAA', hasil_1)
cv2.imshow('Gambar Asli', img)

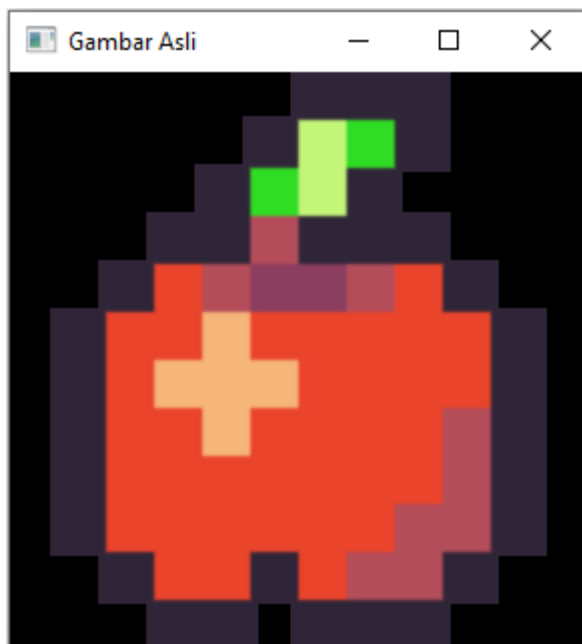
# Nunggu tombol untuk tutup jendela
```

```
cv2.waitKey(0) #7 jika menekan tombol 0 maka  
cv2.destroyAllWindows() #fungsi ini dijalankan yaitu untuk menghentikan semua proses program
```

FXAA ON:



FXAA OFF:



## 4. Kesimpulan

Secara singkat, anti-aliasing adalah teknik untuk membuat gambar yang terlihat lebih halus dan mengurangi efek garis berjumbai atau tidak halus dalam grafik digital. Ini dicapai dengan menggunakan berbagai metode matematis atau filter untuk menghaluskan tepi objek, menciptakan tampilan yang lebih alami dan berkualitas tinggi. Dengan mengurangi aliasing, gambar digital terlihat lebih jelas dan lebih enak dilihat oleh mata manusia. Teknik ini digunakan luas dalam dunia permainan video, grafik komputer, animasi, dan aplikasi visual lainnya.

## **DAFTAR PUSTAKA**

<https://www.artvisi.or.id/smoothing-dan-blurring-menggunakan-opencv-python/>  
<https://docs.opencv.org/>  
<https://numpy.org/doc/>