

Grayscale:

- **Mengambil Data Gambar:**
 - `ctx.getImageData(0, 0, width, height)` mengambil data piksel dari gambar yang saat ini digambar di kanvas, dengan ukuran lebar dan tinggi yang ditentukan.
 - **Mengakses Data Piksel:**
 - Array data berisi data piksel dalam bentuk `Uint8ClampedArray`. Setiap 4 nilai berturut-turut mewakili satu piksel: [R, G, B, A] (Merah, Hijau, Biru, Alpha).
 - **Meloop Setiap Piksel:**
 - Loop ini mengiterasi setiap piksel dalam gambar dengan menaikkan indeks `i` sebanyak 4, memastikan setiap piksel (R, G, B, A) diproses.
 - **Menghitung Nilai Grayscale:**
 - Untuk setiap piksel, hitung nilai grayscale dengan rumus:
$$\text{gray} = r * 0.3 + g * 0.59 + b * 0.11$$
, yang merupakan rata-rata berbobot dari nilai merah, hijau, dan biru.
 - **Menetapkan Nilai Grayscale:**
 - Nilai yang dihitung (gray) kemudian diterapkan pada komponen R, G, dan B untuk piksel tersebut, sehingga piksel tersebut berubah menjadi grayscale.
 - **Mengembalikan Data Gambar:**
 - `ctx.putImageData(imageData, 0, 0)` menggambar kembali data gambar yang telah dimodifikasi (grayscale) ke kanvas.
-

Blur:

- **Mengambil Data Gambar:**
 - `ctx.getImageData(0, 0, width, height)` mengambil data piksel dari gambar yang saat ini digambar di kanvas.
- **Mengakses Data Piksel:**
 - Array data berisi data piksel dalam bentuk `Uint8ClampedArray` yang memuat nilai-nilai warna RGB dan Alpha setiap piksel.
- **Menyiapkan Array untuk Data Blur:**
 - Array `blurredData` disalin dari data asli agar kita dapat memodifikasi nilai-nilai RGB tanpa mempengaruhi pengolahan piksel yang lain.
- **Menghitung Blur untuk Setiap Piksel:**

- Loop ini mengiterasi setiap piksel pada gambar.
- Untuk setiap piksel, kita menghitung nilai rata-rata RGB dari piksel-piksel di sekitarnya dengan jangkauan radius tertentu (dalam hal ini 5 piksel).
- **Mengumpulkan Piksel Sekitar:**
 - Untuk setiap piksel, loop kedua mengiterasi piksel-piksel di sekitar (dengan jarak dx dan dy), memastikan piksel yang berada di luar batas gambar diabaikan.
 - Nilai RGB dari piksel-piksel sekitarnya dijumlahkan dan dihitung rata-ratanya.
- **Menetapkan Data Blur:**
 - Setelah menghitung rata-rata RGB untuk setiap piksel, nilai tersebut diterapkan ke array blurredData.
- **Mengembalikan Gambar Blur:**
 - Data yang telah diblur kemudian diterapkan ke kanvas dengan `ctx.putImageData(blurredImageData, 0, 0)`.