

KONVOLUSI

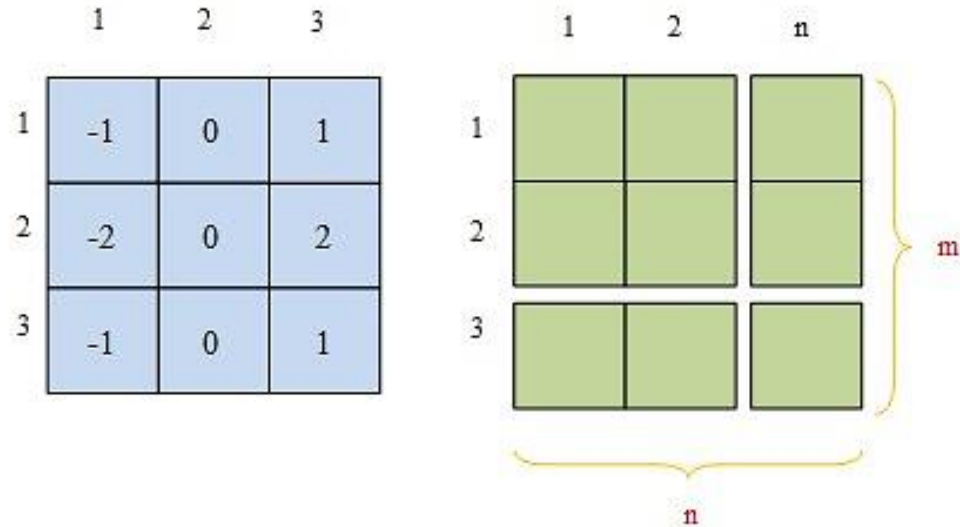
PERTEMUAN KE 6

PENGERTIAN KONVOLUSI

Konvolusi pada citra sering disebut konvolusi 2D, didefinisikan sebagai proses untuk memperoleh suatu piksel didasarkan pada **nilai piksel itu sendiri dan tetangganya**, dengan melibatkan suatu matrik yang disebut **kernel**, yang mempresentasikan pembobotan.

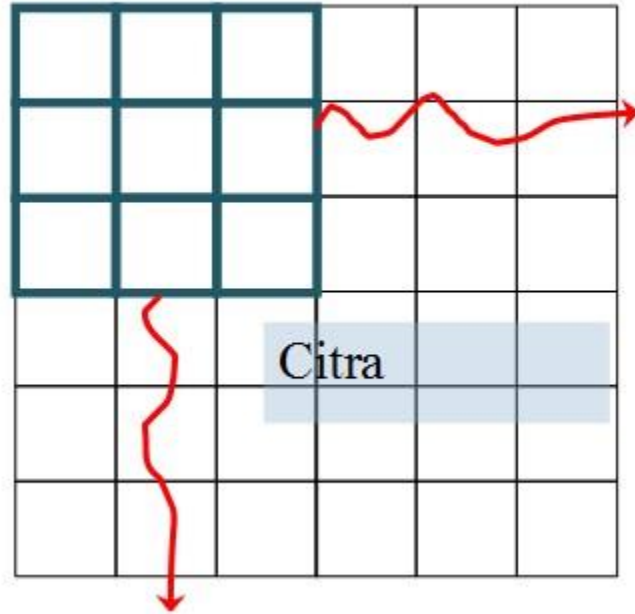


Wujud kernel umumnya bujur sangkar, tetapi dapat pula berbentuk persegi panjang. Seperti ditunjukkan pada Gambar 6.1



Gambar 6.1 Contoh kernel untuk konvolusi berukuran 3 x 3 dan m X n

Kernel digerakkan di
sepanjang baris dan kolom



Operasi ini dilakukan dengan menumpangkan suatu jendela (kernel) yang berisi angka-angka pengali, pada setiap piksel yang ditimpali. Kemudian nilai rata-rata diambil dari hasil kali tersebut.

Kernel digeser sepanjang baris dan kolom dalam citra, sehingga diperoleh nilai baru pada citra keluaran, seperti pada Gambar 6.2.

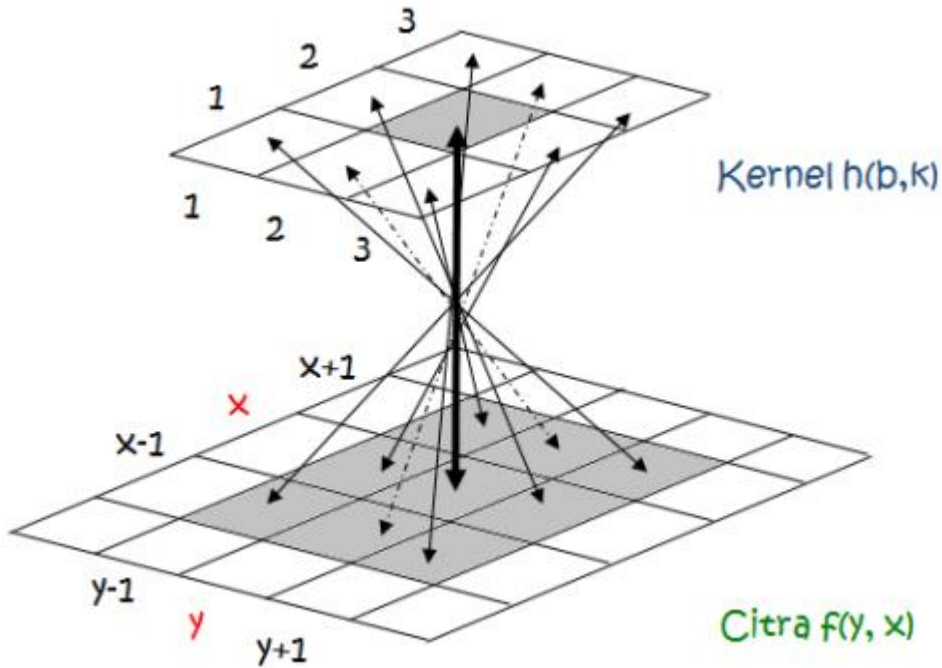
Gambar 6.2 Konvolusi dilakukan dengan melakukan proses di sepanjang kolom dan baris pada citra

Konvolusi dilakukan dengan rumus sebagai berikut :

$$g(y, x) = \sum_{p=-m_2}^{m_2} \sum_{q=-n_2}^{n_2} h(p + m_2 + 1, q + n_2 + 1) f(y - p, x - q) \dots\dots\dots (1)$$

Dalam hal ini:

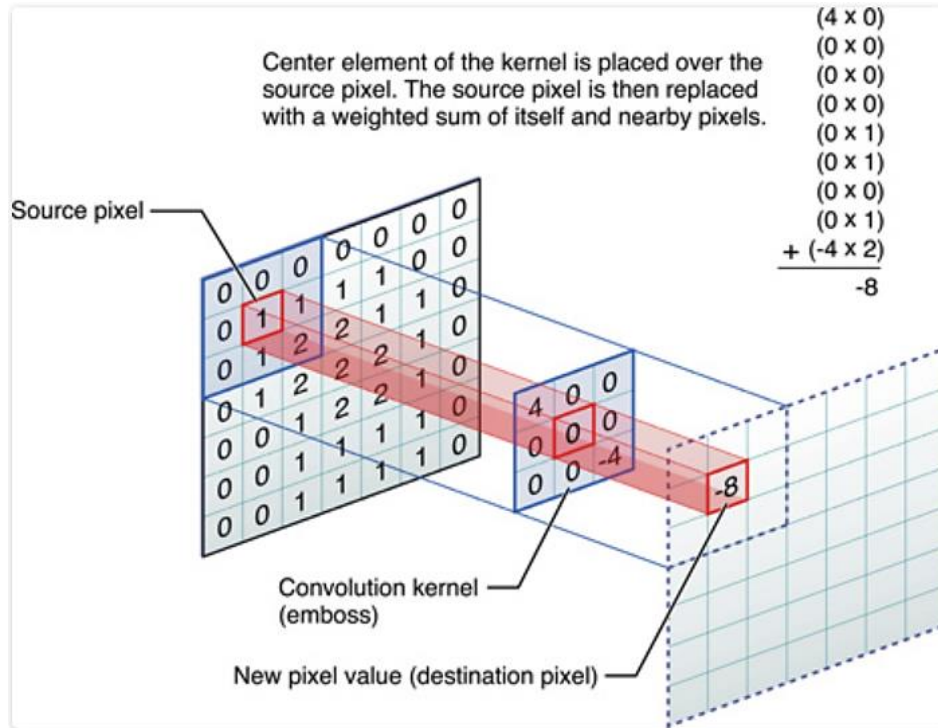
- m_2 adalah separoh dari tinggi kernel ($m_2 = \text{floor}(m/2)$)
- n_2 adalah separuh dari lebar kernel ($n_2 = \text{floor}(n/2)$)
- **floor** menyatakan pembulatan ke bawah, dan
- h menyatakan kernel dengan indeks dimulai dari 1



Gambar 6.3 Contoh konvolusi pada citra

Konvolusi dilakukan antara citra dengan kernel, tujuannya ada berbagai macam, diantaranya untuk **blurring**, **mempertajam citra**, **embos**, dan **mendeteksi sisi pada citra**.

Contoh teknik konfolusi seperti ditunjukkan pada Gambar 6.3.



Gambar 6.4 Penjelasan konvolusi pada citra

Konvolusi dilakukan antara citra dengan kernel, tujuannya ada berbagai macam diantaranya untuk blurring, mempertajam citra, embos, dan mendeteksi sisi pada citra.

Ada banyak sekali kernel (umumnya matrik berukuran 3 X 3). Seperti ditunjukkan pada Gambar 6.4.

a	b	c
d	e	e
f	g	h

Original Image
Pixels



r	s	t
u	v	w
x	y	z

Filter

$$e_{processed} = v * e + z * a + y * b + x * c + w * d + u * e + t * f + s * g + r * h$$

Gambar 6.5 Perhitungan konvolusi secara manual

Berdasarkan Gambar 6.3, apabila citra yang berada pada jendela kernel berupa :

65	50	55
78	68	60
60	60	62

Dengan kernel
berupa :

-1	0	1
-2	0	2
-1	0	1

Maka nilai piksel hasil konvolusi berupa:

$$\begin{aligned}g(x, y) &= -1 \times 62 + 0 \times 60 + 1 \times 60 + (-2 \times 60) + 0 \times 68 + 2 \times 78 \\&\quad + (-1 \times 55) + 0 \times 50 + 1 \times 65 \\&= -62 + 0 + 60 - 120 + 0 + 152 - 55 + 0 + 65 \\&= \mathbf{40}\end{aligned}$$

Dengan demikian, nilai **68** akan diubah menjadi **40** pada citra keluaran

HASIL KERJA KONVOLUSI



(a) Citra gedung.png



(b) Hasil konvolusi citra gedung



(c) Citra altstadt.png



(d) Hasil konvolusi citra altstadt

Gambar 6.6 Contoh penerapan konvolusi

Contoh Listing Program untuk melakukan konvolusi Citra dengan mengabaikan bagian tepi:

Fungsi yang digunakan untuk konvolusi :

```
function [G] = konvolusi(F, H)
% KONVOLUSI Melakukan konvolusi kernel H dengan
citra F
% H harus mempunyai tinggi dan lebar ganjil
% Hasil: citra G
[tinggi_f, lebar_f] = size(F);
[tinggi_h, lebar_h] = size(H);
m2 = floor(tinggi_h/2);
n2 = floor(lebar_h/2);
F2=double(F);
for y=m2+1 : tinggi_f-m2
    for x=n2+1 : lebar_f-n2
        % Pelaksanaan konvolusi F(baris, kolom)
        jum = 0;
        for p=-m2 : m2
            for q=-n2 : n2
                jum = jum + H(p+m2+1,q+n2+1) * ...
                    F2(y-p, x-q);
            end
        end
        G(y-m2, x-n2) = jum;
    end
end
```

Pemakaian fungsi konvolusi ditunjukkan berikut:

```
>> H = [-1 0 -1; 0 4 0; -1 0 -1];
>> F = imread('C:\Image\gedung.png');
>> K = konvolusi (F, H);
>> K
>> K2 = uint8 (K);
>> imshow (K2);
```

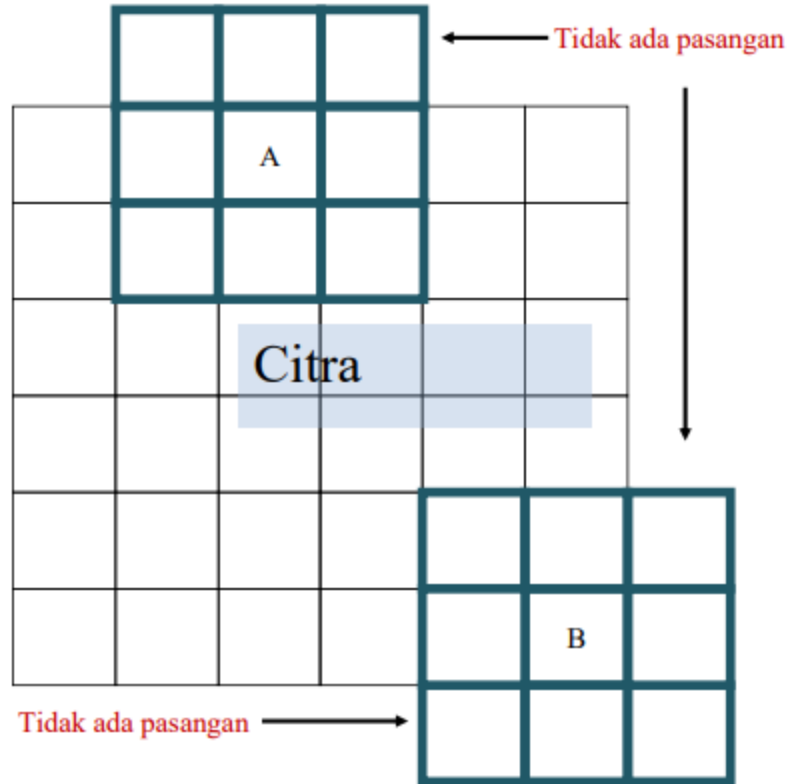


PROBLEM PADA KONVOLUSI

PROBLEM PADA KONVOLUSI

Piksel yang berada pada tepi citra, yaitu baris dan kolom yang terletak di tepi citra, tidak memiliki tetangga yang lengkap. Seperti ditunjukkan pada Gambar 6.6. Sehingga rumus konvolusi tidak berlaku pada piksel seperti itu

HASIL KERJA KONVOLUSI



Gambar 6.7 Problem pada konvolusi

Terdapat beberapa solusi, untuk mengatasi problem konvolusi. Pada piksel yang tidak memiliki tetangga lengkap:



Abaikan piksel pada bagian tepi

Oleh karena pada bagian tepi citra tetangga tidak lengkap sehingga piksel pada posisi tersebut tidak dikenai konvolusi. Sebagai konsekuensinya citra yang tidak mengalami konvolusi akan diisi nol, atau diisi nilai pada citra asal.



Buat baris tambahan pada bagian tepi

Baris dan kolom di tambahkan pada bagian tepi , sehingga proses konvolusi dapat dilaksanakan. Dalam hal ini baris dan kolom baru diisi dengan nilai 0

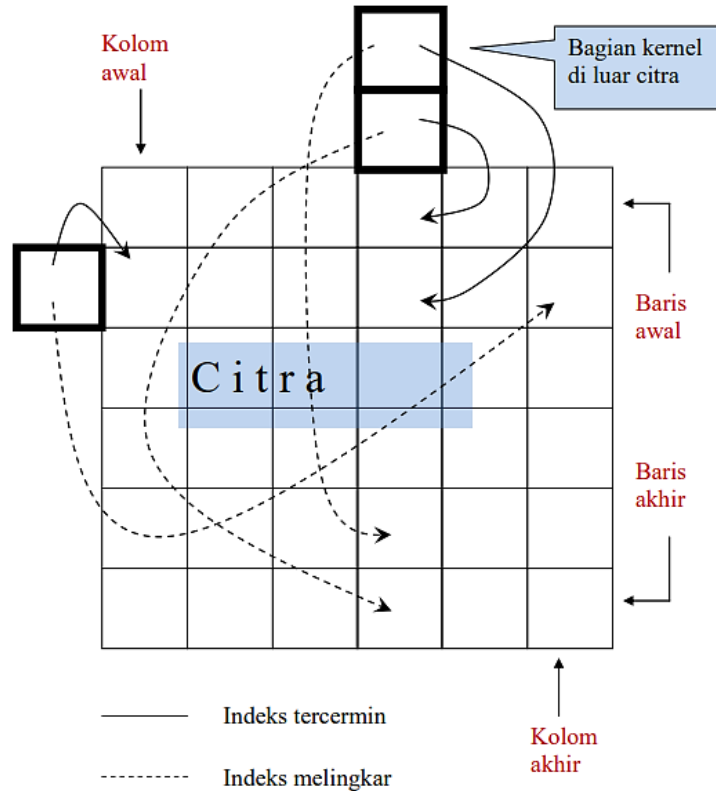


Ambil bagian yang tidak punya pasangan, dengan bagian lain dari citra

Ada dua pendekatan cara yang dapat digunakan :

- Mengisi dengan citra pada bagian tepi
- Melakukan penggulungan secara periodis

Pengambilan data untuk posisi diluar area citra



Gambar 6.8 Mengambil data untuk posisi kernel diluar area citra

Solusi untuk mengatasi problem konvolusi



(a) Citra asli



(b) Bagian tepi diberi nilai nol



(c) Pengisian dari baris atau kolom terpinggir



(d) Pengisian dengan pencerminan



(e) Pengulangan dari tepi yang berseberangan

Gambar 6.9 Mengatasi problem konvolusi

Contoh Listing Program unuk melakukan konvolusi Citra dengan menggunakan indek cermin:

```
function [G] = konvolusi2(F, H)
% KONVOLUSI2 Melakukan konvolusi kernel H
dengan citra F
% (Versi Algoritma 4.3)
% H harus mempunyai tinggi dan lebar ganjil
% Hasil: citra G
[tinggi_f, lebar_f] = size(F);
[tinggi_h, lebar_h] = size(H);
m2 = floor(tinggi_h/2);
n2 = floor(lebar_h/2);
F2=double(F);
for y=1 : tinggi_f
    for x=1 : lebar_f
        % Pelaksanaan konvolusi F(baris, kolom)
        jum = 0;
        for p=-m2 : m2
            for q=-n2 : n2
                % Penanganan x
                x2 = x-q;
                if x2 < 1
                    x2 = -x2 + 1;
                else
                    if x2 > lebar_f
                        x2 = 2 * lebar_f - x2 + 1;
                    end
                end
            end
        end
    end
end
```

```
% Penanganan y
y2 = y-p;
if y2 < 1
    y2 = -y2 + 1;
else
    if y2 > tinggi_f
        y2 = 2 * tinggi_f - y2 + 1;
    end
end

jum = jum + H(p+m2+1,q+m2+1) * ...
    F2(y2, x2);

end
end

G(y, x) = jum;

end
```

Thanks!

Any questions?