

## BAB 2 LATAR BELAKANG

### 2.1. Mobile Learning

Mobile learning secara singkat dapat didefinisikan sebagai pembelajaran yang memanfaatkan peluang menyelenggarakan Kegiatan Belajar Mengajar (KBM) melalui teknologi yang bersifat mobile. Pada konsep pembelajaran tersebut mobile learning membawa manfaat ketersediaan materi ajar yang dapat di akses setiap saat dan visualisasi materi yang menarik.

Ditinjau dari konsep dasarnya, mobile learning memiliki keterkaitan dengan apa yang dahulu dikenal dengan sebutan distance learning dan e-learning. Bedanya mobile learning fokus pada mobilitas pembelajar dan interaksinya dengan teknologi mobile. Dan hal ini yang diperkirakan para ahli akan mempengaruhi para pembelajar dan berbagai institusi pendidikan untuk melakukan evaluasi ulang atas apa yang dahulu disebut sebagai pengajaran dan pembelajaran.

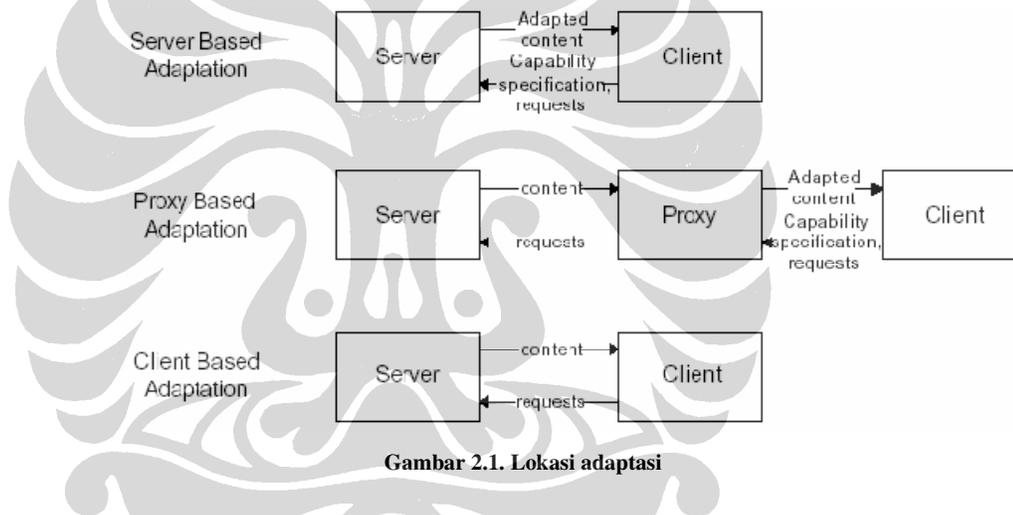
Beberapa pertimbangan di bawah ini perlu diperhatikan dalam pembangunan sebuah sistem mobile sehubungan dengan keterbatasan yang dimiliki peralatan mobile[9]:

1. Karakteristik mobile device yang digunakan. Pada [2], didefinisikan bahwa mobile device untuk learning adalah divais apapun yang berukuran kecil, dan cukup sederhana untuk dibawa sehari-hari dan akan digunakan untuk menunjang proses pembelajaran. Biasanya jika dibandingkan dengan PC desktop, divais ini jauh lebih kecil, misalnya PDA dan telepon selular dengan layar yang berukuran kecil. Selain layar yang kecil, pada divais yang hanya memiliki keyboard telepon tanpa kemampuan touch screen, kecepatan memasukkan informasi menjadi lebih terbatas. Selain itu, perangkat mobile pada umumnya biasanya memiliki keterbatasan memory, dukungan terhadap media dan memiliki prosesor yang lebih lambat.
2. Konektivitas, menjadi salah satu syarat utama untuk sistem mobile apapun, termasuk kebutuhan akan bandwidth yang cukup tinggi.

Walaupun tersedia berbagai teknologi untuk mengakses internet dari sebuah mobile device, pada umumnya bandwidth yang dimiliki tergolong lebih kecil dibandingkan dengan dengan koneksi wired. User sering terdiskoneksi, baik karena infrastruktur yang belum mendukung ataupun biaya yang masih termasuk tinggi dan user lebih memilih untuk terhubung dengan internet menggunakan koneksi yang lebih murah.

## 2.2. Arsitektur Umum Adaptasi Pada Mobile Learning

Berdasarkan lokasi dimana proses adaptasi terjadi, maka arsitektur adaptasi dapat dibedakan menjadi tiga yaitu server-based, proxy-based, dan client-based. Ketiga arsitektur adaptasi tersebut ditunjukkan pada Gambar 2 [10].



Gambar 2.1. Lokasi adaptasi

Pada adaptasi berbasis *server*, seluruh proses adaptasi dilakukan di *server* adaptasi (*adaptation server*). *Server* ini menjadi jembatan antara *server* konten dengan server *Web*. Sesuai dengan informasi tentang profil dari perangkat *client*, kemudian dilakukan proses adaptasi, dan hasilnya baru dikirimkan melalui jaringan Internet kepada *client*. Dengan melakukan adaptasi di *server* maka dapat dihindarkan pengiriman paket-paket yang tidak sesuai dengan kapabilitas perangkat *client*. Arsitektur ini memiliki kelemahan yaitu kesibukan yang tinggi pada sisi *server*, meskipun dengan perkembangan kemampuan *processor* pada komputer *server* hal ini menjadi lebih mudah diatasi.

Proses adaptasi berbasis *server* dapat dilakukan dengan menyediakan versi yang khusus untuk setiap *client* (*variant selection*), adaptasi terhadap struktur global dokumen sehingga menghasilkan struktur yang baru, adaptasi terhadap media sumber misalnya *video/image transcoding* (*media adaptation*), adaptasi dengan menggunakan *meta-information*, serta adaptasi dengan melakukan dekomposisi sebuah dokumen menjadi beberapa *delivery unit* (DU) seperti pada WML.

Pada arsitektur kedua, adaptasi dilakukan oleh *server proxy* lokal dari jaringan komputer *client*. Adaptasi dapat dilakukan dengan cara pemilihan *server* (*selecting a server*), adaptasi konten (*content adaptation*), maupun adaptasi protokol. Untuk adaptasi terhadap konten, metodenya sama dengan adaptasi berbasis server, hanya proses adaptasi terjadi di *proxy server*. Demikian juga informasi tentang profil perangkat *client* juga tetap dibutuhkan. Dengan mendistribusikan proses adaptasi ke *proxy server* maka beban kerja *Web server* menjadi lebih ringan. Namun arsitektur ini tidak efisien dalam penggunaan *bandwidth*, karena hanya sebagian kecil saja dokumen Web yang sesungguhnya digunakan oleh perangkat *handheld* ataupun *mobile phone*. Selain itu pengiriman dokumen Web yang kompleks akan membebani waktu akses *client* ke Internet.

Arsitektur adaptasi ketiga menyerahkan proses adaptasi kepada *client*, jadi adaptasi dilakukan oleh masing-masing perangkat yang digunakan oleh *client*. Keuntungan arsitektur ini adalah pengaturan presentasi dapat dilakukan dengan lebih fleksibel oleh *client*, sehingga menghasilkan tampilan yang spesifik. Namun kenyamanan ini menuntut *resource* yang besar pada perangkat *client* (memori dan *processor*) serta koneksi Internet yang memiliki kecepatan tinggi.

## 2.3. Dasar Teori Citra

### 2.3.1. Pengertian citra

Yang dimaksud dengan citra pada bidang ilmu pengolahan citra bisa didefinisikan sebagai fungsi dua dimensi  $f(x,y)$  di mana  $x$  dan  $y$  adalah koordinat spasial. Amplitudo  $f$  pada setiap pasang  $(x,y)$  disebut intensitas (*gray level*) citra pada titik tersebut. Gambar analog dibagi menjadi  $N$  baris dan  $M$  kolom sehingga

menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut dengan piksel.

Jika  $x$  dan  $y$  berhingga (finite) dan diskrit (tidak kontinu) maka disebut citra digital. Citra digital terdiri dari sejumlah elemen berhingga yang masing-masing mempunyai lokasi dan nilai, dihasilkan dari gambar analog dua dimensi yang kontinu melalui proses sampling.

Proses sampling, disebut juga proses digitisasi, adalah proses untuk menentukan warna pada piksel tertentu pada citra dari sebuah gambar analog. Pada proses ini biasanya dicari warna rata-rata dari gambar analog yang kemudian dibulatkan.

### 2.3.2. Pengertian derau (noise)

Noise atau derau biasanya didapat dari proses digitisasi citra, yaitu proses menangkap objek di dunia nyata kedalam citra digital. Contoh digitisasi citra adalah memotret sesuatu dengan kamera digital.

Noise pada citra tidak hanya terjadi karena ketidaksempurnaan dalam proses *capture*, tetapi bisa juga disebabkan oleh kotoran-kotoran yang terjadi pada citra. Berdasarkan bentuk dan karakteristiknya, noise pada citra dibedakan menjadi beberapa macam yaitu Gaussian, Speckle, dan Salt & Pepper [.

Noise Gaussian merupakan model noise yang mengikuti distribusi normal standard dengan rata-rata nol dan standard deviasi . Efek dari gaussian noise ini, pada gambar muncul titik-titik berwarna yang jumlahnya sama dengan prosentase noise. Noise Speckle merupakan model noise yang memberikan warna hitam pada titik yang terkena noise. Sedangkan noise Salt & Pepper seperti halnya taburan garam, akan memberikan warna putih pada titik yang terkena noise



(a)

(b)

(c)

Gambar 2.2. Efek noise pada citra (a). Gaussian Noise (b). Speckle noise (c). Salt & Pepper[12]

### 2.3.3. Pengolahan citra

Untuk meningkatkan kualitas suatu citra, maka dilakukan pengolahan dengan berbagai metode yang bertujuan untuk mengurangi atau menghilangkan noise.

Operasi yang dilakukan untuk mentransformasikan suatu citra menjadi citra lain dapat dikategorikan berdasarkan tujuan transformasi maupun cakupan operasi yang dilakukan terhadap citra.

Berdasarkan tujuan transformasi operasi pengolahan citra dikategorikan sebagai berikut [11]:

1. Peningkatan Kualitas Citra (Image Enhancement), bertujuan untuk meningkatkan fitur tertentu pada citra. Teknik ini digunakan untuk memperbaiki suatu citra yang noise-nya belum diketahui dan tidak dapat dimodelkan.
2. Pemulihan Citra (Image Restoration), bertujuan untuk mengembalikan kondisi citra pada kondisi yang diketahui sebelumnya akibat adanya pengganggu yang menyebabkan penurunan kualitas citra. Teknik ini digunakan pada operasi sebuah citra yang noise-nya telah diketahui jenisnya dan dapat dimodelkan.

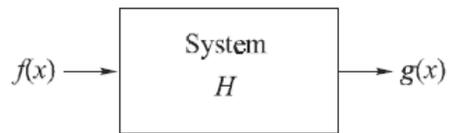
Alat bantu matematika yang sering dipakai dalam pengolahan citra ada bermacam-macam, antara lain statistik inheren, konvolusi, transformasi Fourier, dan representasi kontur. Berdasarkan [1], proses deblurring suatu image memerlukan perhitungan matematis dengan metode konvolusi

### 2.3.4. Sistem Linier

Pada pengolahan citra, sistem linier merupakan inti dari berbagai operasi filtering, termasuk pada operasi deblurring.

Gambar 9 adalah sistem yang terdiri dari sebuah unit yang mengubah fungsi masukan  $f(x)$  ke dalam keluaran (atau respon) dengan fungsi  $g(x)$ , di mana  $x$  adalah variable bebas seperti waktu, atau pada kasus image, posisi spasial.

Untuk kesederhanaan, asumsi bahwa  $x$  adalah variable kontinu, namun hasil yang didapatkan dapat diaplikasikan pula pada variable diskrit.



Gambar 2.3. Contoh model sistem linier

Sistem keluaran ditentukan oleh masukannya, property sistem, dan kondisi awal. Hal itu dapat dituliskan dengan:

$$G(x) = H[f(x)]$$

Di mana  $H$  adalah operator sistem, didefinisikan sebagai pemetaan atau penugasan dari suatu anggota kumpulan keluaran yang mungkin  $\{g(x)\}$  untuk tiap anggota dari kumpulan masukan yang mungkin  $\{f(x)\}$ . Dengan kata lain, operator sistem sepenuhnya membuat karakterisasi respon sistem untuk suatu kelompok masukan  $\{f(x)\}$ .

Sebuah operator  $H$  disebut operator linier untuk suatu kelas masukan  $\{f(x)\}$  jika:

$$\begin{aligned} H[\alpha_i f_i(x) + \alpha_j f_j(x)] &= a_i H[f_i(x)] + a_j H[f_j(x)] \\ &= a_i g_i(x) + a_j g_j(x) \end{aligned}$$

Untuk semua  $f_i(x)$  dan  $f_j(x)$  termasuk  $\{f(x)\}$ , di mana  $a$  adalah konstanta sembarang dan  $g_i(x) = H[f_i(x)]$  adalah keluaran dari masukan sembarang  $f_i(x) \in \{f(x)\}$ .

Sistem yang digambarkan dengan operator linear disebut dengan sistem linier. Sifat di mana suatu proses linier pada penjumlahan masukan adalah sama dengan menjalankan operasi tersebut secara individual dan kemudian menjumlahkan hasilnya, disebut dengan sifat *additivity*. Sifat di mana respon suatu sistem linier terhadap perkalian dengan suatu konstanta adalah sama dengan respon terhadap masukan asli dikalikan dengan konstanta tersebut dinamakan sifat *homogeneity*.

Operator  $H$  dikatakan *time invariant* (jika  $x$  merepresentasikan waktu), *spatial invariant* (jika  $x$  adalah variable spasial), atau *fixed parameter*, untuk suatu kelas masukan  $\{f(x)\}$  jika

$$G_i(x) = H[f_i(x)] \text{ sama dengan } g_i(x+x_0) = h [f_i(x+x_0)]$$

untuk semua  $f_i(x) \in \{f(x)\}$  dan semua  $x_0$

Suatu sistem yang digambarkan dengan operator fixed-parameter dikatakan fixed-parameter sistem. Pada dasarnya, semua berarti bahwa offsetting variable bebas dari masukan oleh  $x_0$  menyebabkan offset yang sama pada variable bebas dari keluaran. Karena itu, hubungan masukan-keluaran tetap sama.

Suatu operator dikatakan causal, dan karenanya sistem dideskripsikan oleh  $H$  adalah sistem kausal, jika tidak ada keluaran sebelum ada masukan. Dengan kata lain,

$$F(x) = 0 \text{ untuk } x < x_0 \text{ berarti bahwa } g(x) = H[f(x)] = 0 \text{ untuk } x < x_0$$

Terakhir, sistem linier  $H$  dikatakan stabil jika responnya terhadap batasan masukan terbatas pula. Yaitu, jika

$$|f(x)| < K \text{ berarti bahwa } |g(x)| < cK,$$

di mana  $K$  dan  $c$  adalah konstanta.

### 2.3.5. Filter

Filter digunakan dalam proses pengolahan citra untuk perbaikan kualitas citra (*image enhancement*), menghilangkan derau, deteksi tepi sehingga dapat dilakukan penajaman tepi, dan sebagainya.

Untuk memberikan filter pada citra, digunakan metode konvolusi [1].

Konvolusi 2 fungsi  $f(x)$  dan  $g(x)$  dapat dinyatakan dengan:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d\alpha$$

dengan  $\alpha$  adalah peubah bantu

Fungsi diskrit dari konvolusi  $f(x)$  dan  $g(x)$ :

$$f(x) * g(x) = \sum f(\alpha)g(x-\alpha)$$

### 2.3.6. Konvolusi sistem linier

Fungsi unit impulse, dilambangkan dengan  $\delta(x - \alpha)$ , didefinisikan dengan ekspresi

$$\int_{-\infty}^{\infty} f(\alpha) \delta(x - \alpha) d\alpha = f(x)$$

Output sistem diberikan oleh persamaan  $g(x) = H[f(x)]$ . Dalam termin fungsi impulse:

$$g(x) = H \left[ \int_{-\infty}^{\infty} f(\alpha) \delta(x - \alpha) d\alpha \right]$$

Dengan sifat additivity pada integral,

$$g(x) = \int_{-\infty}^{\infty} H[f(\alpha) \delta(x - \alpha)] d\alpha$$

Karena  $f(\alpha)$  independen terhadap  $x$ , dan menggunakan sifat homogeneity, maka

$$\begin{aligned} g(x) &= \int_{-\infty}^{\infty} f(\alpha) H[\delta(x - \alpha)] d\alpha \\ &= \int_{-\infty}^{\infty} f(\alpha) h(x, \alpha) d\alpha \end{aligned}$$

Termin

$$h(x, \alpha) = H[\delta(x - \alpha)]$$

Disebut dengan respon impuls dari  $H$ . Dengan kata lain,  $h(x, \alpha)$  adalah respon dari sistem linier terhadap unit impuls yang terletak pada koordinat  $x$  (asal impuls adalah nilai  $\alpha$  yang menghasilkan  $\delta(0)$ , dalam hal ini terjadi ketika  $\alpha = x$ ). Ekspresi

$$g(x) = \int_{-\infty}^{\infty} f(\alpha) h(x, \alpha) d\alpha$$

Disebut dengan superposition intergral of the first kind. Ekspresi ini merupakan hasil dasar yang menjadi inti dari teori sistem linier. Jika respon  $H$  terhadap suatu unit impulse (misalnya  $h(x, \alpha)$ ) diketahui, maka respon terhadap masukan apapun  $f$  dapat dihitung dengan menggunakan integral. Dengan kata lain, respon sistem linier dikarakterisasikan oleh respon impulse-nya.

Jika  $H$  adalah operator fixed-parameter, maka

$$H[\delta(x - \alpha)] = h(x - \alpha)$$

dan integral superposition menjadi:

$$g(x) = \int_{-\infty}^{\infty} f(\alpha) h(x-\alpha) d \alpha$$

Ekspresi di atas disebut dengan integral konvolusi. Respon suatu sistem linier, fixed-parameter sepenuhnya dikarakterisasikan dengan konvolusi masukan dengan respon impuls dari sistem. Karena variable  $\alpha$  pada persamaan sebelumnya di-integrasikan, maka konvolusi antara  $f$  dan  $g$  (yang sama-sama merupakan fungsi dari  $x$ ) sebagai

$$G(x) = f(x) * h(x)$$

Dengan kata lain,

$$f(x) * h(x) \triangleq \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) d \alpha$$

Transformasi Fourier ekspresi tersebut secara singkat adalah

$$F [f(x) * h(x)] = H(u) F(u)$$

Karena keluaran dari sistem linier, fixed-parameter adalah

$$G(x) = f(x) * h(x)$$

Jika kedua sisi ditransformasi Fourier, maka akan didapatkan hasil

$$G(u) = H(u) F(u)$$

Pentingnya hasil ini adalah, cara lain untuk melakukan konvolusi untuk mendapatkan keluaran sistem, adalah dengan menghitung transformasi Fourier dari respon impulse dan masukan, mengalikan hasilnya, dan melakukan inverse transformasi Fourier dari produk untuk mendapatkan  $g(x)$ , yaitu

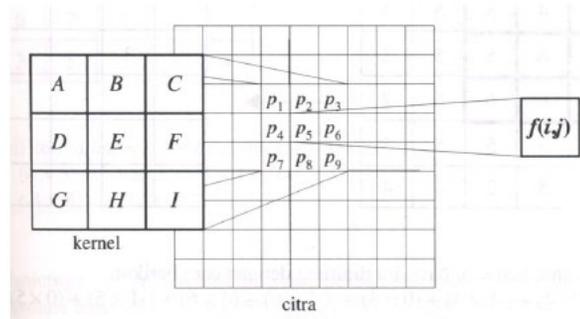
$$g(x) = \mathcal{I}^{-1}[G(u)]$$

$$g(x) = \mathcal{I}^{-1}[H(u)F(u)]$$

### 2.3.7. Konvolusi 2D

$$f(x, y) * g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b) g(x - a, y - b)$$

$G(x)$  merupakan convolution mask/filter/kernel atau template. Konvolusi dapat dinyatakan dalam bentuk matriks, dengan tiap elemen matriks filter merupakan koefisien konvolusi. Operasi konvolusi dapat digambarkan dengan menggeser kernel,  $G(x)$ , pixel per pixel, hasil disimpan dalam matriks baru. Operasi konvolusi dapat diilustrasikan sebagai berikut:



Gambar 2.4. Operasi Konvolusi pada citra

$$F(i,j) = Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9$$

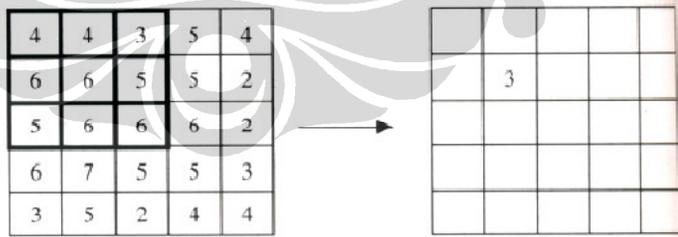
Contoh konvolusi:

Citra yang dapat digambarkan dengan fungsi  $f(x,y)$  berukuran  $5 \times 5$  dan sebuah kernel berukuran  $3 \times 3$ :

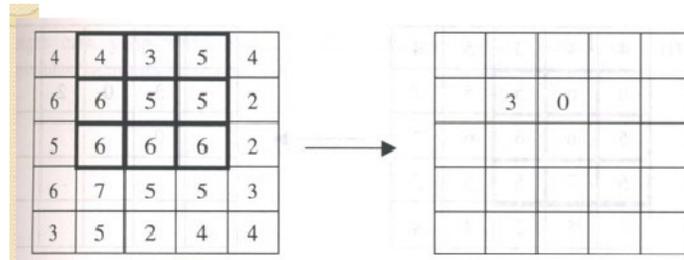
$$f(x,y) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix}$$

$$g(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

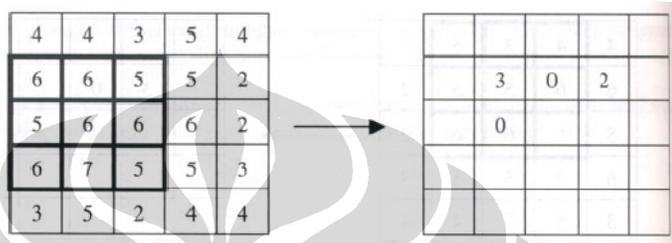
Tanda ■ melambangkan posisi (0,0) dari kernel



Gambar 2.5. Konvolusi langkah I



Gambar 2.6. Konvolusi langkah II



Gambar 2.7. Konvolusi langkah III

	4	0	8	
	0	2	6	
	6	0	2	

Gambar 2.8. Hasil akhir konvolusi

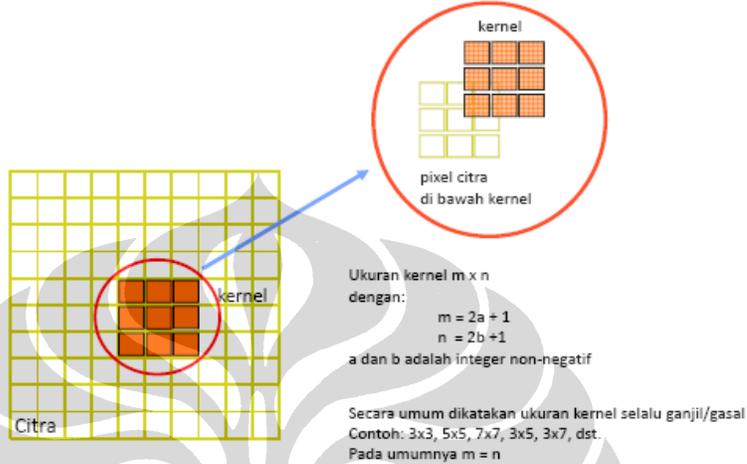
Bila hasil konvolusi negative, maka nilai dijadikan 0, hal ini dinamakan clipping, Bila hasil konvolusi lebih besar dari derajat keabuan maksimum, maka nilai diubah ke derajat keabuan maksimum, hal ini juga dinamakan clipping.

Penapisan (filtering) termasuk pengolahan lokal, yaitu dalam transformasinya melibatkan:

- nilai-nilai pixel tetangganya
- nilai-nilai suatu subcitra yang memiliki dimensi yang sama. Subcitra ini dikenal sebagai filter, mask, kernel, template, atau window. Nilai dalam subcitra tidak disebut sebagai nilai intensitas pixel, tetapi sebagai koefisien

Penapisan yang dibicarakan saat ini adalah penapisan spasial (spatial filtering). Ada banyak macam filter, antara lain adalah:

- 1. Low pass, biasanya digunakan untuk menghasilkan citra yang lebih lembut (smooth) sehingga terkesan kabur (blur)
- 2. Median, bermanfaat untuk mengatasi masalah derau
- 3. High pass, akan menghasilkan citra yang lebih tajam (sharp)



Gambar 2.9. Penapisan pada citra

2.3.8. Metode konvolusi pada pengolahan citra

Operasi matematis yang terlibat dalam proses blurring adalah konvolusi. Secara sederhana, konvolusi bekerja dengan cara: citra menjalankan suatu array pada citra yang asli. Array itu disebut dengan kernel konvolusi. Untuk tiap pixel pada citra, diambil nomor yang sesuai dari kernel dan pixel yang dilalui, dikalikan, dan ditambahkan semua hasil tadi untuk membuat pixel baru. Sebagai contoh, blur sederhana yang dicari dengan menghitung rata-rata tiap pixel dan 8 pixel sekitarnya. Kernel yang dibutuhkan adalah[4]:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Jumlah seluruh pixel sama dengan 1, yang berarti citra hasil akan sama terangnya dengan citra yang asli.

2.3.9. Deblurring

Blurring, atau degradasi, dari sebuah citra dapat dilihat seperti sebuah titik tajam semakin jauh dari titik itu semakin kotor, biasanya membentuk piringan.

Pada citra, ini berarti tiap pixel pada citra tersebar dan tercampur dengan pixel di sekitarnya[4]. Cara lain yang dapat mendeskripsikan blurring adalah tiap pixel pada citra yang menjadi target terdiri dari campuran pixel sekitarnya pada citra sumber.

Blurring dapat disebabkan oleh banyak factor, di antaranya adalah:

- Out of focus[2]. Hal yang sering terjadi adalah ketika kamera terfokus pada suatu titik yang tidak dimaksudkan fotografer untuk menjadi titik pusat dari image tersebut.
- Obyek bergerak ketika mengambil gambar[2]. Ketika keadaan gelap atau tidak banyak cahaya yang ada pada suatu ruangan, kamera akan mengompensasi dengan membuka shutter lebih lama dari normal. Ketika shutter terbuka, obyek telah bergerak. Hal ini terlihat dari hasil pemotretan. Jika beberapa bagian dari obyek jelas dan tajam sementara yang lainnya kabur (blurry) maka obyek telah bergerak ketika shutter terbuka.
- Kamera bergerak atau tidak stabil ketika mengambil gambar[2]. Jika foto diambil pada saat malam hari atau pada ruangan yang tidak terang pencahayaannya, maka shutter akan terbuka lebih lama. Ketika shutter terbuka lama, gerakan sekecil apapun, seperti melepas jari dari tombol shutter, dapat menyebabkan obyek yang diambil blurry. Citra yang mengalami blurry akibat hal ini terlihat dari blurry yang menyeluruh, tidak seperti blurry sebagian yang diakibatkan nomor 2.
- Depth of field terlalu dangkal[2]. Depth of field adalah istilah untuk efek kamera di mana obyek fokus dan latar belakang out of focus. Hal ini membuat obyek utama menjadi terlihat karena hanya obyek tersebut yang terlihat jelas. Jika depth of field terlalu dangkal, maka tidak semua bagian dari obyek utama akan terlihat jelas.

#### 2.3.10. Deblurring model

Citra yang terdegradasi atau blurred kira-kira dapat digambarkan dengan persamaan sebagai berikut:

$$\mathbf{I} = \mathbf{L} \otimes \mathbf{f} + \mathbf{n}$$

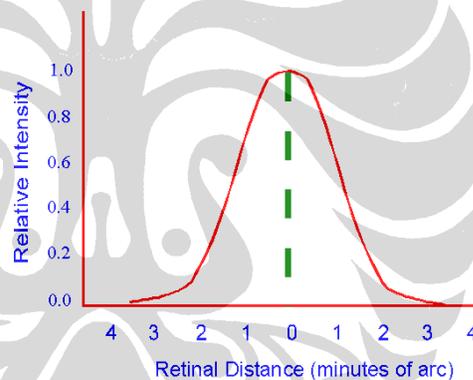
**I** merupakan image yang terdegradasi, yang direkam oleh kamera

**L** adalah image yang ingin dihasilkan

**f** merupakan operator distorsi, yang disebut juga dengan point spread function (PSF).

**n** adalah noise tambahan, yang didapatkan selama pengambilan citra, yang merusak citra.

Pada domain spasial, PSF menggambarkan derajat di mana sebuah sistem optis menyebarkan sebuah titik cahaya. PSF adalah invers dari transformasi Fourier terhadap fungsi transfer optis (Optical Transfer Function, OTF). Pada domain frekuensi, OTF menggambarkan respon dari sistem linear, dan *position-invariant* terhadap suatu sinyal. Operator distorsi, jika dikonvolusikan dengan citra, akan menghasilkan distorsi. Distorsi yang disebabkan oleh suatu fungsi PSF hanyalah salah satu dari tipe distorsi.

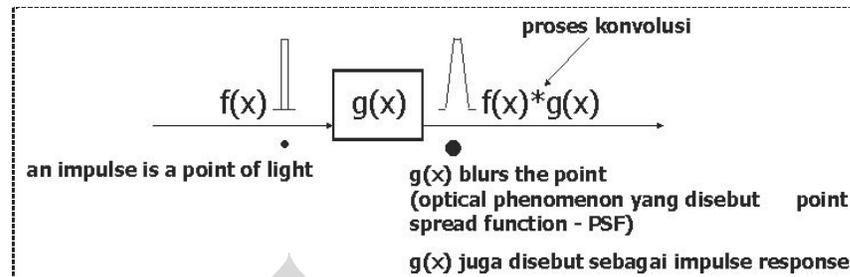


Gambar 2.10. Representasi grafis dari PSF [14]

Adanya impulse tersebut mengakibatkan perlunya memperbaiki citra agar sesuai dengan yang diinginkan. Transformasi Fourier yang mengubah representasi citra dari domain spasial ke domain frekwensi, digunakan untuk mengetahui sebaran spectrum suatu citra agar dapat mengetahui/memprediksi dampak impuls yang terjadi

Sebaliknya Inverse Fourier Transform (IFT) akan mengubah representasi citra dari domain frekwensi ke domain spasial ( $g(x, y) \rightarrow FT \rightarrow G(u, v) \rightarrow IFT \rightarrow g(x, y)$ ). FT ini akan memudahkan proses konvolusi dari bentuk integral menjadi bentuk perkalian biasa. Pada sistem yang linier, Fourier transform dari suatu impulse menghasilkan  $G(u,v)$  dan disebut sebagai system transfer function.

Sebaliknya  $g(x,y)$  sebagai inverse transform dari system transfer function disebut sebagai impulse response. Pada sistem optik,  $g(x,y)$  merupakan inverse transform dari optical transfer function  $G(u,v)$ , dan disebut point spread function.



Gambar 2.11. Point Spread Function [13]

Berdasarkan model di atas, tugas dasar proses deblurring adalah untuk mendekonvolusikan blurred image dengan PSF yang secara tepat menggambarkan distorsi yang terjadi. Kualitas image yang ter-deblurred utamanya ditentukan oleh PSF yang diketahui.

### 2.3.11. Seam Carving

Keberagaman karakteristik layar pada mobile device memerlukan gambar yang merupakan salah satu elemen kunci pada media digital, untuk dapat fleksibel terhadap berbagai macam layout secara otomatis. Kasus-kasus di mana ukuran atau aspek ratio dari sebuah gambar harus dapat berubah dan cocok dengan berbagai macam cara seperti telepon selular atau PDA[6].

Saat ini terdapat dua metode resizing yang banyak digunakan, yaitu scaling dan cropping. Scaling berarti gambar tersebut diubah ukurannya secara seragam. Jika misalnya sebuah citra akan dikurangi ukurannya dari 100 piksel menjadi 50 piksel, maka tiap dua piksel yang bersebelahan dihitung rata-ratanya. Teknik ini dapat diaplikasikan termasuk ketika lebar citra yang baru bukan merupakan pembagian lebar citra yang lama dan untuk melakukan resizing lebar dan tinggi suatu citra.

Scaling bersifat efisien dan tidak banyak informasi yang hilang dari citra asal. Walau begitu, metode scaling memiliki dua kelemahan, yaitu [5]:

- Jika dimensi citra yang akan dibuat tidak proporsional terhadap dimensi asli, maka citra target akan terdistorsi

- Metode tersebut tidak memperhitungkan isi citra.

Metode kedua, cropping, terdiri dari memilih suatu sub citra pada proporsi yang diinginkan dari citra asal, dengan kata lain, menghilangkan tepi citra. Dengan cara ini muncul pertanyaan, bagaimana pemilihan kotak mana yang akan disimpan atau bagaimana cara seleksi tepi mana yang akan dihilangkan.

Dimisalkan tiap piksel pada citra diberikan bobot yang merepresentasikan derajat pentingnya piksel tersebut. Pemilihan kotak cropping berarti memilih kotak yang berisi piksel-piksel dengan jumlah bobot tertinggi. Pada tulisan ini, bobot piksel disebut energy dari piksel. Karena itu cropping optimal akan memilih kotak mana yang memiliki energy tertinggi.

Seam carving menggunakan fungsi energi yang mendefinisikan pentingnya tiap piksel. Seam adalah sebuah jalur terkoneksi dari piksel-piksel berenergi rendah pada seluruh bagian gambar dari atas sampai bawah, atau dari kiri ke kanan. Dengan keberhasilan untuk membuang atau memasukkan seam baru, maka gambar akan dapat dikurangi ataupun diperbesar pada kedua arah.

Untuk memperkecil gambar, pemilihan seam memastikan bahwa dengan mempertahankan struktur image, maka piksel-piksel berenergi rendah dapat lebih banyak dikurangi dan piksel dengan energy yang lebih tinggi dipertahankan. Untuk memperbesar gambar, urutan penyisipan seam memastikan keseimbangan antara isi citra asli dengan piksel-piksel yang disisipkan.

$$e_1(\mathbf{I}) = \left| \frac{\partial}{\partial x} \mathbf{I} \right| + \left| \frac{\partial}{\partial y} \mathbf{I} \right|$$

Yang dimaksud dengan vertical seam adalah jalur 8 piksel yang terhubung pada citra, dihitung dari atas sampai bawah, terdiri dari 1 dan hanya 1 piksel pada tiap baris citra. Untuk tiap piksel, diperhatikan 8 piksel yang berdekatan dengannya.

Algoritma seam carving secara sederhana dapat dijelaskan sebagai berikut[7]:

- menghitung energy tiap pixel pada citra asal.
- membangun pemetaan dari energy kumulatif minimum (cumulative energy map)

- melakukan penelusuran pada data yang didapatkan di langkah 2 untuk mendapatkan jalur (seam) dengan energy terendah
- menghapus piksel-piksel yang masuk pada jalur seam

### *Perhitungan energi pada citra*

Terdapat banyak cara untuk menghitung energi tiap piksel. Salah satunya adalah dengan implementasi pengukuran gradient menggunakan Sobel operator.

Sobel operator sering juga disebut sebagai deteksi tepi. Secara matematis, operator Sobel menggunakan dua buah kernel ukuran 3x3 yang dikonvolusikan terhadap citra asli untuk menghitung perkiraan hasil citra – satu untuk perubahan horizontal, dan satu untuk vertical. Jika A adalah citra sumber,  $G_x$  dan  $G_y$  adalah dua citra yang pada tiap titik mengandung perkiraan hasil secara horizontal dan vertical, komputasi dapat dilakukan sebagai berikut[8]:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

di mana \* menandakan operasi konvolusi 2 dimensi

Koordinat x di sini didefinisikan sebagai peningkatan pada arah kanan, dan koordinat y didefinisikan sebagai peningkatan pada arah turun. Pada tiap titik citra asal, perkiraan hasil gradien dapat dikombinasikan untuk memberikan gradient magnitude, menggunakan[8]:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

Menggunakan informasi ini, arah gradient juga dapat dihitung[8]:

$$\Theta = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Di mana, sebagai contoh,  $\Theta$  adalah 0 untuk tepi vertical yang lebih gelap di sisi kiri.

Walaupun operator ini menghasilkan perkiraan yang kurang akurat terhadap gradient citra, namun tetap memberikan kualitas yang cukup untuk

kepraktisan penerapan pada banyak aplikasi. Sobel operator dapat dioperasikan dengan mudah karena hanya 8 titik citra di sekitar sebuah titik yang diperlukan untuk menghitung hasil yang bersangkutan dan hanya aritmetik integer yang diperlukan untuk menghitung perkiraan vector gradient. Lebih lanjut lagi, dua filter diskrit di atas dapat dipisahkan menjadi[8]:

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

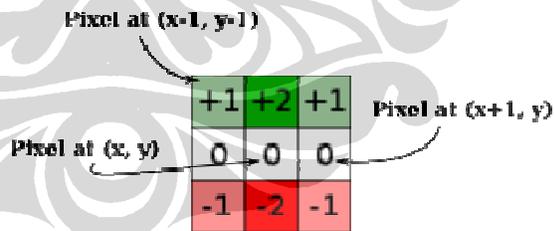
$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Dan dua turunan  $G_x$  dan  $G_y$  dapat dihitung dengan[8]

$$G_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} +1 & 0 & -1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * A$$

Gambar di bawah ini mengilustrasikan perhitungan gradient menggunakan operator Sobel:



Gambar 2.12. Menghitung energy pada citra[7]

Untuk menghitung y-gradien dari sebuah piksel, tambahkan nilai-nilai dari piksel yang bersebelahan, diskala mengikuti matriks sobel, dengan hasil  $G_y$ .

Evaluasi gradient untuk tiap kolom warna (R,G,B) secara bebas, misalnya  $G_yR$ ,  $G_yG$ , dan  $G_yB$  dan ambil rata-rata untuk  $G_y$ . Karena itu  $G_y$  dapat dinyatakan dengan

$$G_y = (G_yR + G_yG + G_yB)/3$$

Horizontal gradient dihitung dengan cara yang sama dan kedua gradient magnitude (x,y) disimpan pada posisi piksel pada energy map. Piksel-piksel yang berada pada tepian selalu diset memiliki energy maksimum.

#### *Cumulative Energy Map*

Langkah berikutnya adalah menyiapkan pemetaan dari energy-energi kumulatif tadi untuk dilakukan backtracing pada langkah berikutnya. Pemetaan energy dapat dihasilkan dengan rumus berikut ini:

$$M(x, y) = \text{energy}(x, y) + \min( M(x-1, y-1), M(x, y-1), M(x+1, y-1) )$$

Untuk menghitung energy kumulatif minimum untuk piksel (x,y), tambahkan energy piksel tersebut pada energy kumulatif minimum dari salah satu piksel sebelumnya.

#### *Backtracing*

Cumulatif Energy map (CE/M) yang telah didapatkan digunakan untuk mencari vertical seam dengan energy terendah. Backtracing adalah metode umum untuk melakukan proses ini. Dimulai dengan pencarian piksel dengan CE terendah pada baris terbawah. Setelah menambahkan piksel kedalam seam, lihat 3 piksel yang bersesuaian di baris atasnya dan pilih salah satu dengan CE terendah. Proses itu dapat digambarkan sebagai berikut:

7	2	6
9	8	10
15	23	11

Gambar 2.13. Backtracing[7]

Algoritma dimulai dari baris terbawah dan mendapati bahwa 11 adalah CE minimal. Tambahkan piksel tersebut dalam seam dan lihat dua piksel bersebelahan di atasnya. 8 adalah CE terendah, maka piksel inilah yang akan ditambahkan pada seam. Pada langkah selanjutnya, piksel nilai 2 adalah piksel dengan CE terkecil dan karenanya ditambahkan pada seam. Langkah ini terus dilakukan sampai iterasi tiba di baris teratas piksel.

Untuk mendapatkan vertical seam, dilakukan hal yang sama dengan langkah di atas, namun dengan arah dari kiri ke kanan.

Hasil seam carving sangat tergantung pada fungsi energy yang dipilih. Metode gradient ini dipilih karena mudah diterapkan namun tetap menghasilkan kualitas citra yang cukup baik.

Untuk memberikan bobot yang sesuai pada tiap piksel, digunakan gradient magnitude sebagai fungsi energy. Gradien magnitude secara sederhana berarti memperhitungkan gradient pada tiap piksel dan mengambil magnitudenya.

