

Peningkatan Kualitas Citra Scanning dengan Menggunakan Metode Gaussian Filter Berbasis Android

Ananda Hariati

Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: anandahariati1994@gmail.com

Abstrak-Dalam menambah pengetahuan umum kita mengenai negara-negara yang ada di dunia kita membutuhkan informasi. Perbaikan kualitas citra (image enhancement) merupakan salah satu proses awal dalam pengolahan citra (preprocessing). Perbaikan kualitas diperlukan karena seringkali citra yang dijadikan objek pembahasan mempunyai kualitas yang buruk, misalnya citra mengalami derau (noise) pada saat pengiriman melalui saluran transmisi, citra terlalu terang/gelap, citra kurang tajam, kabur, dan sebagainya. Citra digital sebagai salah satu bentuk data digital saat ini banyak dipakai untuk menyimpan photo, gambar ataupun hasil karya dalam format digital. Salah satu contoh dari hal yang bersifat negatif tersebut adalah dengan mencuri dan memanipulasi citra yang dimiliki suatu pihak yang akan dikirim ke pihak lainnya menjadi gambar yang tidak sama dengan gambar asli yang dikirim oleh pengirim dengan tujuan merugikan pihak penerima. Pengolahan citra memegang peranan yang sangat penting dalam kehidupan sehari-hari, tapi sering tidak disadari akan pentingnya informasi yang ditampilkan dalam suatu pengolahan citra tersebut. Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Agar citra yang mengalami gangguan mudah diinterpretasi, maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik.

Kata Kunci: Peningkatan Kualitas, Gaussian Filter, Android

Abstract-Image quality improvement (image enhancement) is one of the initial processes in image processing (preprocessing). Quality improvement is needed because often the image used as the object of discussion has poor quality, for example the image experiences noise when sending through the transmission line, the image is too bright / dark, the image is less sharp, blurred, and so on. Digital image as a form of digital data is currently widely used to store photos, images or works in digital format. One example of a negative thing is to steal and manipulate the image of a party that will be sent to another party into an image that is not the same as the original image sent by the sender with the aim of harming the recipient. Image processing plays a very important role in everyday life, but often it is not realized the importance of the information displayed in an image processing. Image processing is image processing, especially by using a computer, becoming a better quality image. In order for images that are subjected to interference to be easily interpreted, the image needs to be manipulated into another image with better quality.

Keywords: Quality Improvement, Gaussian Filter, Android

1. PENDAHULUAN

Pada dasarnya perbaikan citra (gambar) bertujuan meningkatkan kualitas citra tampilan, citra untuk pandangan manusia atau mengkonversi suatu citra agar memiliki format yang lebih baik sehingga citra tersebut menjadi lebih mudah diolah dengan mesin (komputer). Proses peningkatan kualitas citra merupakan suatu proses untuk mengubah sebuah citra menjadi citra baru sesuai dengan kebutuhan melalui berbagai cara. Salah satu cara untuk melakukan peningkatan kualitas citra ialah dengan melakukan pembesaran citra metode yang digunakan untuk melakukan pembesaran citra dengan menambah nilai piksel citra baru pada citra, sehingga dihasilkan citra yang memiliki kerapatan piksel yang lebih tinggi dari citra semula.

Filter gaussian sangat baik untuk menghilangkan *noise* yang bersifat sebaran normal, yang banyak di jumpai pada sebaran citra hasil proses digitasi menggunakan kamera karena merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri. Citra hasil *scanning* bisa ditingkatkan kualitasnya dengan menerapkan metode *gaussian filter*. Metode *gaussian filter* ini dapat digunakan untuk meningkatkan kualitas citra yang berhubungan dengan pencahayaan yaitu dengan mengatur tingkat kecerahan (*brightness*) maupun kontras dari sebuah citra sehingga diperoleh citra yang memiliki kualitas lebih baik.[1].

2. METODOLOGI PENELITIAN

2.1 Peningkatan Kualitas Citra

Peningkatan kualitas citra (*image enhancement*) merupakan salah satu proses awal dalam pengolahan citra (*image preprocessing*). Peningkatan kualitas diperlukan karena seringkali citra yang dijadikan objek pembahasan mempunyai kualitas yang buruk, misalnya citra mengalami derau (*noise*) pada saat pengiriman melalui saluran transmisi, citra terlalu terang/gelap, citra kurang tajam, kabur dan sebagainya. Melalui operasi pemrosesan awal inilah kualitas citra diperbaiki sehingga citra dapat digunakan untuk aplikasi lebih lanjut, misalnya untuk aplikasi pengenalan (*recognition*) objek di dalam citra [2].

Peningkatan kualitas citra adalah proses mendapatkan citra yang lebih mudah diinterpretasikan oleh mata manusia. Ciri-ciri tertentu yang terdapat di dalam citra lebih diperjelas kemunculannya. Secara matematis, *imageenhancement* dapat diartikan sebagai proses mengubah citra .

2.2 Scanning

Scanning adalah teknik membaca yang digunakan untuk mendapatkan suatu informasi atau fakta yang diinginkan dalam waktu yang sangat cepat. Teknik ini dilakukan dengan cara memindai kata – kata atau kata kunci yang sedang dicari. Oleh karena itu, membaca scanning tidak perlu membaca isi keseluruhan teks, tetapi hanya langsung mencari inti atau informasi yang akan kita cari.

Pakar mengenai scanning untuk memperoleh informasi tertentu telah mengembangkan keterampilan khusus yang berbeda dengan cara membaca yang lain. Selama menggerakkan mata secara tepat melihat keseluruhan materi bacaan, pembaca tidak menyerap makna tetapi hanya mengenali bahwa hal-hal yang dicarinya, tampaknya hal itu secara jelas seolah-olah tercetak tebal dan beberapa orang dapat mencapai kecepatan yang hampir tidak dapat dipercaya dalam melakukan scanning.[3]

2.3 Gaussian Filter

Gaussian filtering didapat dari operasi konvolusi. Operasi perkalian yang dilakukan ialah perkalian antara matriks *kernel* dengan matriks gambar asli. Matriks *kernel gauss* didapat dari fungsi komputasi dari distribusi *gaussian*, seperti pada persamaan di bawah ini:

$$G(i,j) = c.e^{-\frac{(i-u)^2+(j-v)^2}{2\sigma^2}} \quad (1)$$

Keterangan:

c dan σ = konstanta

G (i,j) = elemen matriks *kernel gauss* pada posisi (i,j)

(u,v) = indeks tengah dari matriks *kernel gauss*

Berikut ini contoh matrik *kernel Gauss* 3 X 3 dengan $\sigma= 1.0$

1	2	1
2	3	2
1	2	1

Perkalian antara bobot matriks gambar asli dengan bobot matrik *kernel gauss* dapat dirumuskan seperti pada di bawah ini:

$$Pixel B(i,j) = \frac{1}{K} \cdot \sum_{p=0}^{N-1} \left(\sum_{q=0}^{M-1} G(p,q) \cdot PixelA \left(i + p - \frac{(N-1)}{2}, j + q - \frac{(M-1)}{2} \right) \right) \quad (2)$$

Keterangan:

Pixel A = gambar A (Gambar Asli)

Pixel B(i,j) = bobot hasil perkalian pada posisi (i,j)

N = jumlah kolom matriks kernel

M = jumlah baris matriks kernel

K = penjumlahan semua bobot di G

G(p,q) = elemen matriks kernel gauss pada posisi (p,q)

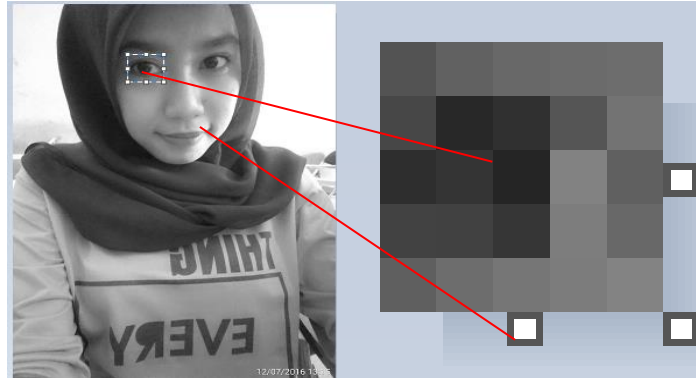
Gambar yang akan diproses dibagi menjadi 2 jenis piksel, yaitu piksel batas dan piksel dalam. Piksel batas yaitu piksel yang berada dipaling luar pada gambar, selain piksel tersebut disebut piksel dalam. Untuk piksel yang berada di dalam, perkalian dilakukan menggunakan rumus no 5, yaitu menjadikan piksel yang dicari nilai barunya sebagai piksel tengah dan bobotnya dikalikan dengan bobot pada piksel tengah matrik kernel, lalu dijumlahkan dengan hasil perkalian antara bobot piksel-piksel tetangga-tetangganya dengan bobot piksel matrik kernel. Untuk piksel yang berada disudut atau perbatasan, sebelum dilakukan perkalian, sebelumnya harus mencari bobot pada piksel-piksel luar (*dummy*). Bobot piksel-piksel ini dicari dengan menggunakan interpolasi yaitu dengan melihat dua piksel di dekatnya yang searah (horizontal atau vertikal). Apabila ada piksel yang memiliki bobot lebih kecil dari 0 maka bobot dijadikan 0. Apabila ada piksel yang memiliki bobot lebih besar dari 255 maka bobotnya dijadikan 255.[4]

2.4 Android

Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (mobiledevices) yang terdiri dari sistem operasi, middleware, dan aplikasi-aplikasi utama. Awalnya, android dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Sistem operasi android kemudian diluncurkan bersamaan dengan dibentuknya organisasi Open Handset Alliance tahun 2007. Selain google, beberapa nama-besar juga ikut serta dalam Open Handset Alliance, antara lain Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Thosiba, dan intel[5].

3. HASIL DAN PEMBAHASAN

Persiapan citra dapat dilihat di bawah ini dengan mempersiapkan sample citra awal



Gambar 2. Contoh Pixel

Tabel 1. Nilai Pixel Grayscale:

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126

Tabel 2. Nilai Kernel Awal

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

Aturan Gaussian:

$$G(x,y) = \frac{1}{2\sigma^2 \cdot \pi} \left(e^{-\frac{x^2+y^2}{2 \cdot \sigma^2}} \right)$$

Penjabaran:

a) $G(0,0) \longrightarrow \begin{matrix} x = 0 \\ y = 0 \end{matrix}$

$$G(0,0) = \frac{1}{2 \cdot (1)^2 \cdot 3,14} \left(e^{-\frac{0^2+0^2}{2 \cdot (1)^2}} \right)$$

$$G(0,0) = \frac{1}{2 \cdot (1) \cdot 3,14} \left(e^{-\frac{0}{2}} \right)$$

$$G(0,0) = 0,159$$

b) $G(0,1) \longrightarrow \begin{matrix} x = 0 \\ y = 1 \end{matrix}$

$$G(0,1) = \frac{1}{2 \cdot (1)^2 \cdot 3,14} \left(e^{-\frac{0^2+1^2}{2 \cdot (1)^2}} \right)$$

$$G(0,1) = \frac{1}{2 \cdot (1) \cdot 3,14} \left(e^{-\frac{1}{2}} \right)$$

$$G(0,1) = 0,262$$

c) $G(0,2) \longrightarrow \begin{matrix} x = 0 \\ y = 2 \end{matrix}$

$$G(0,2) = \frac{1}{2.(1)^2.3,14} \left(\frac{e^{-0^2+2^2}}{2.(1)^2} \right)$$

$$G(0,2) = \frac{1}{2.(1).3,14} \left(\frac{e^{-4}}{2} \right)$$

$$G(0,2) = 1,175$$

d) $G(1,0) \longrightarrow \begin{matrix} x = 1 \\ y = 0 \end{matrix}$

$$G(1,0) = \frac{1}{2.(1)^2.3,14} \cdot e^{-\left(\frac{1^2+0^2}{2.(1)^2}\right)}$$

$$G(1,0) = \frac{1}{2.(1).3,14} \left(\frac{e^{-\frac{1}{2}}}{2} \right)$$

e) $G(1,1) \longrightarrow \begin{matrix} x = 1 \\ y = 1 \end{matrix}$

$$G(1,1) = \frac{1}{2.(1)^2.3,14} \left(\frac{e^{-\frac{1^2+1^2}{2.(1)^2}}}{2.(1)^2} \right)$$

$$G(1,1) = \frac{1}{2.(1).3,14} \left(\frac{e^{-\frac{2}{2}}}{2} \right)$$

$$G(0,2) = 0,346$$

f) $G(1,2) \longrightarrow \begin{matrix} x = 1 \\ y = 2 \end{matrix}$

$$G(1,2) = \frac{1}{2.(1)^2.3,14} \left(\frac{e^{-\frac{1^2+2^2}{2.(1)^2}}}{2.(1)^2} \right)$$

$$G(1,2) = \frac{1}{2.(1).3,14} \left(\frac{e^{-\frac{5}{2}}}{2} \right)$$

$$G(0,2) = 1,937$$

g) $G(2,0) \longrightarrow \begin{matrix} x = 2 \\ y = 0 \end{matrix}$

$$G(2,0) = \frac{1}{2.(1)^2.3,14} \left(\frac{e^{-\frac{2^2+0^2}{2.(1)^2}}}{2.(1)^2} \right)$$

$$G(2,0) = \frac{1}{2.(1).3,14} \left(\frac{e^{-\frac{4}{2}}}{2} \right)$$

$$G(2,0) = 1,175$$

h) $G(2,1) \longrightarrow \begin{matrix} x = 2 \\ y = 1 \end{matrix}$

$$G(2,1) = \frac{1}{2.(1)^2.3,14} \left(\frac{e^{-\frac{2^2+1^2}{2.(1)^2}}}{2.(1)^2} \right)$$

$$G(2,1) = \frac{1}{2.(1).3,14} \left(\frac{e^{-\frac{5}{2}}}{2} \right)$$

$$G(2,1) = 1,937$$

i) $G(2,2) \xrightarrow{\substack{x=2 \\ y=2}}$

$$G(2,2) = \frac{1}{2 \cdot (1)^2 \cdot 3,14} \left[e^{-\frac{2^2+2^2}{2 \cdot (1)^2}} \right]$$

$$G(2,2) = \frac{1}{2 \cdot (1) \cdot 3,142} \left[e^{-8} \right]$$

$$G(0,2) = 8,681$$

Tabel 3. Hasil Perhitungan Kernel

0,159	0,262	1,175
0,262	0,346	1,937
1,175	1,937	8,681

Hasil perhitungan kernel dijumlahkan seperti berikut :

$$0,159 + 0,262 + 1,175 + 0,262 + 0,346 + 1,937 + 1,175 + 1,937 + 8,681 = 15,934$$

Dari hasil penjumlahan diatas dibagi dengan hasil perhitungan kernel pada tabel 3.2 diatas, sehingga menghasilkan nilai kernel seperti tabel 3.3 dibawah ini :

Tabel 4. Hasil Nilai *Kernel* Akhir

0,010	0,016	0,074
0,016	0,022	0,122
0,074	0,122	0,545

Maka penyelesaiannya adalah sebagai berikut :

Proses 1:

Nilai pixel grayscale berwarna merah pada tabel dibawah ini dikalikan dengan nilai kernel akhir seperti berikut :

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
0,010	0,016	0,074		
0,016	0,022	0,122		
0,074	0,122	0,545		

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :

$$(83 \cdot 0,010) + (96 \cdot 0,016) + (106 \cdot 0,074) + (73 \cdot 0,016) + (35 \cdot 0,022) + (45 \cdot 0,122) + (46 \cdot 0,074) + (59 \cdot 0,122) + (36 \cdot 0,545) = 47$$

Proses 2:

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
0,010	0,016	0,074		
0,016	0,022	0,122		
0,074	0,122	0,545		

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :

$$(96 \cdot 0,010) + (106 \cdot 0,016) + (110 \cdot 0,074) + (35 \cdot 0,016) + (45 \cdot 0,022) + (87 \cdot 0,122) + (59 \cdot 0,074) + (36 \cdot 0,122) + (128 \cdot 0,545) = 101$$

Proses 3:

83	96	106	110	106
----	----	-----	-----	-----

73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
0,010		0,016	0,074	
0,016		0,022	0,122	
0,074		0,122	0,545	

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :
 $(106 \times 0,010) + (110 \times 0,016) + (106 \times 0,074) + (45 \times 0,016) + (87 \times 0,022) + (115 \times 0,122) + (36 \times 0,074) + (128 \times 0,122) + (97 \times 0,545) = 98$

Proses 4:

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
0,010		0,016	0,074	
0,016		0,022	0,122	
0,074		0,122	0,545	

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :
 $(73 \times 0,010) + (35 \times 0,016) + (45 \times 0,074) + (46 \times 0,016) + (59 \times 0,022) + (36 \times 0,122) + (61 \times 0,074) + (70 \times 0,122) + (51 \times 0,545) = 52$

Proses 5:

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
0,010		0,016	0,074	
0,016		0,022	0,122	
0,74		0,122	0,545	

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :
 $(35 \times 0,010) + (45 \times 0,016) + (87 \times 0,074) + (59 \times 0,016) + (36 \times 0,022) + (128 \times 0,122) + (70 \times 0,074) + (51 \times 0,122) + (127 \times 0,545) = 105$

Proses 6 :

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
0,010		0,016	0,074	
0,016		0,022	0,122	
0,074		0,122	0,545	

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :
 $(45 \times 0,010) + (87 \times 0,016) + (115 \times 0,074) + (36 \times 0,016) + (128 \times 0,022) + (97 \times 0,122) + (51 \times 0,074) + (127 \times 0,122) + (105 \times 0,545) = 102$

Proses 7:

83	96	106	110	106
73	35	45	87	115

46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
<hr/>				
	0,010	0,016	0,074	
	0,016	0,022	0,122	
	0,074	0,122	0,545	

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :
 $(46*0,010) + (59*0,016) + (36*0,074) + (61*0,016) + (70*0,022) + (51*0,122) + (96*0,074) + (108*0,122) + (119*0,545) = 98$

Proses 8:

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
<hr/>				
	0,010	0,016	0,074	
	0,016	0,022	0,122	
	0,074	0,122	0,545	

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :
 $(59*0,010) + (36*0,016) + (128*0,074) + (70*0,016) + (51*0,022) + (127*0,122) + 108*0,074) + (119*0,122) + (126*0,545) = 120$

Proses 9 :

83	96	106	110	106
73	35	45	87	115
46	59	36	128	97
61	70	51	127	105
96	108	119	126	126
<hr/>				
	0,010	0,016	0,074	
	0,016	0,022	0,122	
	0,074	0,122	0,545	

Nilai pixel grayscale berwarna merah dikalikan dengan nilai kernel akhir dihitung dengan cara berikut :
 $(36*0,010) + (128*0,016) + (97*0,074) + (51*0,016) + (127*0,022) + (105*0,122) + (119*0,074) + (126*0,122) + (126*0,545) = 119$

Tabel 4. Citra Hasil Peningkatan Dengan Gaussian Filter

83	96	106	110	106
73	47	101	98	115
46	52	105	102	97
61	98	120	119	105
96	108	119	126	126

4. KESIMPULAN

Dari hasil penulisan dan analisa dari bab-bab sebelumnya, maka dapat diambil kesimpulan, dimana kesimpulan-kesimpulan tersebut kiranya dapat berguna bagi para pembaca, sehingga penulisan skripsi ini dapat lebih bermanfaat. Adapun kesimpulan-kesimpulan tersebut adalah sebagai berikut :

1. Dalam melakukan proses peningkatan kualitas citra *scanning* dilakukan dengan teknik perbaikan kualitas citra (*image enhancement*) yang merupakan salah satu proses awal dalam pengolahan citra (*image preprocessing*). Perbaikan kualitas citra dapat dilakukan dengan menerapkan fungsi transformasi, operasi matematis, pemfilteran atau dengan menerapkan metode salah satunya adalah metode *gaussian filter*.
2. Menerapkan metode *gaussian filter* dapat diterapkan dalam aplikasi peningkatan kualitas citra *scanning* yaitu dengan cara mengkonversikan nilai RGB tiap *pixel* citra ke bentuk citra *grayscale* sehingga diperoleh nilai

grayscale citra baru. Semakin tinggi nilai *grayscale* citranya maka citra hasil perbaikan kualitasnya akan memiliki nilai kontras yang semakin tinggi dan semakin baik.

REFERENCES

- [1] Andre Wedianto, Herlina Latipa Sari, and Yanolanda Suzantri H, "ANALISA PERBANDINGAN METODE FILTER GAUSSIAN, MEAN DAN MEDIAN TERHADAP REDUKSI NOISE," Jurnal Media Infotama, vol. 12, no. 1, pp. 21-30, Februari 2016.
- [2] Darma Putra, Pengolahan Citra Digital, 1st ed., Westriningsih, Ed. Yogyakarta, Indonesia: Andi, 2010.
- [3] Sianipar, "Pemrograman MATLAB Dalam Contoh dan Penerapan". Informatika, 2013.
- [4] Charles Jhony Mantho Sianturi, "Analisis Segmentasi Citra USG Hati Menggunakan Metode Fuzzy C-Mean," Citec Journal, vol. 2, no. 3, pp. 256-264, Mei-Juli 2015.
- [5] Sutoyo, T, dkk. 2009, "Teori Pengolahan Citra Digital", Penerbit Andi, Yogyakarta hal 9 - 27.
- [6] "pengertian-membaca-memindai-scanning-tujuan-cara-kerja-langkah-membaca," <http://www.spengetahuan.com>, 2017. [Online]. Available: <http://www.spengetahuan.com>.
- [7] Rosa A S and M Shalahuddin, Rekayasa Perangkat Lunak, Informatika Bandung ed., Informatika Bandung, Ed. Bandung, Indonesia: Informatika Bandung, 2013.
- [8] Z. P. Juhara, "Panduan Lengkap Pemrograman Android," Yogyakarta: Andi, 2016.
- [9] M. H. D. J. C. Masruri, "Membangun SMS Gateway dengan Gammu & Kalkun," jakarta: PT. Elex Media Komputindo, 2015.
- [10] E. Sutanta, Pengantar Teknologi Informasi, Yogyakarta: Graha Ilmu. 2005
- [11] Mohtashim, (2014, jun.12). UML Tutorial [online]. Available: <https://www.tutorialspoint.com/uml/index.htm>
- [12] N. S. H, Pemrograman Aplikasi Mobile Smartphone Dan Tablet Pc Berbasis Android, Revisi Ked. Bandung: Informatika Bandung, 2015
- [13] Android, "Versi Android," Citra satelit: ©2014 Astrium, DigitalGlobe, 2014. [Online]. Available: <https://www.android.com/>. [Accessed: 24-Jul- 2018].