

**PERBAIKAN KUALITAS CITRA HASIL INVERSI DENGAN
MENGUNAKAN ALGORITMA UNSHARP MASKING FILTER (UMF)
PADA KOLEKSI MIKROFILM NEGATIF MUSEUM SONOBUDOYO
YOGYAKARTA**

Skripsi

untuk memenuhi sebagian persyaratan mencapai derajat Sarjana S-1



disusun oleh:

KHURAIM FATIK

07650036

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2013



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/3255/2013

Skripsi/Tugas Akhir dengan judul : Perbaikan Kualitas Citra Hasil Inversi Dengan Menggunakan Algoritma Unsharp Masking Filter (UMF) Pada Koleksi Mikrofilm Negatif Museum Sonobudoyo Yogyakarta

Yang dipersiapkan dan disusun oleh :
Nama : Khuraim Fatik
NIM : 07650036
Telah dimunaqasyahkan pada : Kamis, 10 Oktober 2013
Nilai Munaqasyah : A / B
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Aulia Faqih Rifa'i, M.Kom
NIP. 19860306 201101 1 009

Penguji I

Agus Mulyanto, M.Kom
NIP.19710823 199903 1 003

Penguji II

Ade Ratnasari, M.T
NIP. 19801217 200604 2 002

Yogyakarta, 25 Oktober 2013
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan



Prof. Drs. H. Akh. Minhaji, M.A, Ph.D
NIP. 19580918 198603 1 002

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Permohonan

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Khuraim Fatik

NIM : 07650036

Judul Skripsi :

Perbaikan Kualitas Citra Hasil Inversi Dengan Menggunakan

Algoritma Unsharp Masking Filter (UMF) Pada Keloeksi Mikrofilm

Negatif Museum Sonobudoyo Yogyakarta

sudah dapat diajukan kembali kepada Program Studi Tekni Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Teknik Informatika

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 6 Oktober 2013

Pembimbing



Aulia Faqih Rifa'i, M.Kom

NIP: 19860306 201101 1 009

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Khuraim Fatik
Nim : 07650036
Program Studi : Teknik Informatika
Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi dengan judul “Perbaikan Kualitas Citra Hasil Inversi Dengan Menggunakan Algoritma Unsharp Masking Filter (Umf) Pada Keloeksi Mikrofilm Negatif Museum Sonobudoyo Yogyakarta”, tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 6 Oktober 2013

METERAN
TEMPEL
7E900ABF796728190
6000 DJP
Menyatakan
uraim Fatik

NIM : 07650036

HALAMAN PERSEMBAHAN

Teriring ucapan syukur yang mampu ku ucapkan kepada-Mu ya Allah, semoga shalawat dan salam senantiasa tercurah kepada Baginda Nabi Muhammad SAW. Aku bersyukur kepadamu alhamdulillah, dengan bimbingan, karunia dan petunjuk-Mu, telah berhasil kuselesaikan tugas akhir ini. Ya Allah aku hanyalah manusia biasa, yang tanpa bimbinganmu aku bukanlah apa-apa. Aku hanyalah manusia biasa, yang tanpa orang-orang disekitarku aku juga bukanlah siapa-siapa. Untuk itu Ya Allah, ijinlanlah aku untuk menyampaikan rasa terimakasihku pada-Mu dan pada mereka. dan sudilah kiranya Engkau senantiasa menjaga dan menjaga mereka di jalan-Mu Ya Allah. dari relung hati yang terdalam kusampaikan rasa terimakasihku:

- Ibunda Suratdiyah tercinta yang sejak lahir sampai sekarang dan kelak akan senantiasa memberikan dukungan, do'a dan pengorbanannya untuk-ku. Terimakasih banyak karena selalu menyelipkan namaku disetiap do'a sepanjang sholat malam. Ya Allah hanya seuntai doa yang aku panjatkan, Ampunilah dosa-doa ibuku, sayangilah ibuku seperti ibuku menyayangiku dan Masukkanlah ibuku kedalam surga-Mu. Amiin.
- Terimakasih ayahanda Wakimo, tanpa engkau saya tak akan berjalan sejauh ini.
- Prof. Drs. H. Akh. Minhaji, M.A.,Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
- Bapak Agus Mulyanto, M. Kom, selaku Ketua Program Studi Teknik Informatika. Keberadaan beliau dalam menuntut ilmu sudah seperti Bapak kedua bagi penulis yang selalu sabar membimbing, mengarahkan, memberikan nasehat dan saran yang tak ternilai harganya. Semoga Allah selalu melindungi Pak Agus dan keluarga.
- Bapak Aulia Faqih Rifa'I, M.Kom selaku Dosen Pembimbing yang telah banyak sekali memberikan ilmu-ilmu dan diskusi yang sangat mengakselerasi bagi penambahan pengetahuan penulis, memberikan saran serta masukan kepada penulis dalam penyusunan skripsi. Semoga Allah senantiasa memberikan kemudahan dan petunjuk-Nya untuk bapak aulia dan keluarga. Tanpamu aku hanya butiran debu. Engkau adalah pelita dalam gelapku. Entah bagaimana nasibku jika engkau tidak menjadi dosen di UIN Sunan Kalijaga? Aku yakin, engkaulah takdir tuhan yang diturunkan ke bumi untuk menggiring "domba-domba" yang tersesat.

- *Para dosen Teknik Informatika, Pak Mustaqim, Pak Sumarsono, Pak Nurrochman, Pak Bambang, Pak Nasirudin, Bu Ade, Bu Uyun, Bu Ulfah, Pak Taufik, Pak Anshari dan pak Bambang Robiin.. terimakasih untuk semua ilmu yang telah dibagikan kepadaku... semoga Allah senantiasa memberikan kemudahan dan petunjuk-Nya untuk Bapak/Ibu dosen sekalian.*
- *Sahabat saya Sidik Somantrie, kamu adalah sahabat terbaik saya, Renaldi Forza, terimakasih atas semangatnya, Oo dan Ayub I Love U, Alpong, dan Idun terimakasih atas tempaan mentalnya.*
- *Sahabat-sahabat seperjuangan informatika 2007, M. Husna Mubarak, Setiya Budi, S.Kom, Rois Awang, Umpu, Ismail Sembiring, dan lain-lainnya yang tidak bisa aku sebutkan satu persatu. Banyak hal yang aku pelajari dari kalian. Semoga Allah selalu memberikan kemudahan untuk kalian sahabat ku.*
- *Semua pihak yang telah memberikan bantuan dan dukungan kepada penulis dalam penyusunan skripsi yang tidak bisa disebutkan satu per satu.*

HALAMAN MOTTO

~~~

*Sebuah Cerita Rakyat dari Afrika,*

*Setiap pagi seekor Rusa terjaga, dia sadar dia harus segera berlari, berlari lebih cepat dari Singa tercepat, atau dia akan mati diterkam singa.*

*Setiap pagi juga seekor Singa terjaga, dia sadar dia harus segera berlari, berlari lebih cepat dari Rusa terlambat, atau dia akan mati kelaparan.*

*Tidak peduli aku Singa atau Rusa, namun yang ku tahu, saat matahari terbit aku harus segera terjaga dan “berlari”, atau mati oleh musuh dan atau waktu itu sendiri.*

~~~



KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah Subhanahu wa ta'ala atas limpahan rahmat, hidayah, serta bimbingan-Nya. Shalawat serta salam semoga tercurah kepada Nabi Muhammad Shallallohu 'alaihi wa sallam. Akhirnya penulis dapat menyelesaikan penelitian Tugas Akhir yang berjudul *ANALISA PEAK SIGNAL TO NOISE RATION(PSNR) DALAM PERBAIKAN CITRA DIGITAL MIKROFILM DENGAN ALGORITMA UNSHARP MASK FILTER (UMF)*; Studi kasus di Museum Sonobudoyo Yogyakarta. Oleh karena itu, dengan segala kerendahan hati pada kesempatan ini penulis mengucapkan banyak terima kasih kepada:

1. Prof. Drs. H. Akh. Minhaji, M.A.,Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
2. Bapak Agus Mulyanto, S.Si, M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
3. Bapak Aulia Faqih Rifa'I, M. Kom. selaku dosen pembimbing yang selalu sabar membimbing, mengarahkan, memberikan nasehat dan saran selama penyusunan skripsi. Bapak dan Ibu Dosen serta jajarannya yang membagi ilmu pengetahuan dan waktunya selama ini.
4. Ibu dan Saudara saya tercinta.
5. Seluruh teman-teman keluarga besar Program Studi Teknik Informatika.

Penulis merasa masih banyak sekali kekurangan dan kelemahan dalam penelitian ini, oleh karena itu segala kritik dan saran senantiasa penulis harapkan dari para pembaca. Akhir kata, semoga penelitian ini dapat menjadi panduan serta referensi yang sangat berguna bagi pembaca dan dapat dimanfaatkan sebaik-baiknya.

Yogyakarta, Oktober 2013

Penulis



DAFTAR ISI

SAMPUL.....	i
HALAMAN PENGESAHAN	ii
SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR.....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN.....	xvii
DAFTAR RINGKASAN	xviii
INTISARI.....	xix
ABSTRACT	xx
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian.....	7
BAB II.....	8
TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	8
2.1. Tinjauan Pustaka	8
2.2. Landasan Teori.....	9
2.2.1. Teknik Fotografi	9
2.2.2. Citra Digital	12
2.2.3. Representasi Citra Digital.....	13
2.2.4. Pengolahan Citra Digital	15
2.2.5. Diagram Alir (<i>Flowchart</i>)	26
BAB III.....	30
METODE PENELITIAN	30

3.1. Tempat dan Waktu Penelitian	30
3.2. Obyek Penelitian	30
3.3. Alat dan Bahan Penelitian.....	31
3.3.1. Perangkat Keras	31
3.3.2. Perangkat Lunak	32
3.4. Bahan Penelitian.....	32
3.5. Metode Penelitian.....	33
3.6.1. Perancangan <i>DIY Negative Scanner</i>	35
3.6.2. Konsep Pindai Citra Mikrofilm	36
3.6.3. Inversi Citra <i>Negative Digital (ND)</i> ke <i>Positive Digital (PD)</i>	37
3.6.4. Operasi Algoritma <i>Unsharp Mask Filter (UMF)</i> pada citra <i>Positive Digital (PD)</i>	37
3.6.5. Perancangan Sistem Konversi Citra	39
3.6.6. Analisa Kualitas Citra.....	39
BAB IV	41
HASIL DAN PEMBAHASAN	41
4.1. Perakitan <i>DIY Negative Scanner</i>	41
4.2. Pemindaian Mikrofilm.....	44
4.3. Inversi Citra Mikrofilm.....	45
4.4. Operasi Algoritma <i>Unsharp Masking Filter</i>	46
4.5. Implementasi Sistem Konversi.....	50
4.6. Pengujian.....	51
4.5.1. Analisa Subyektif Citra Hasil UMF.....	51
4.5.2. Perhitungan PSNR	53
4.5.3. Komparasi Analisa Subyektif dan Analisa PSNR	55
BAB V.....	57
KESIMPULAN DAN SARAN	57
5.1. Kesimpulan	57
5.2. Saran	58
DAFTAR PUSTAKA	59
DAFTAR PUSTAKA	59
LAMPIRAN 1	63
LAMPIRAN 2	64
LAMPIRAN 3	65

LAMPIRAN 4	72
LAMPIRAN 5	75
LAMPIRAN 6	80



DAFTAR GAMBAR

Gambar 2.1. Contoh penggunaan ISO.....	10
Gambar 2.2. Contoh <i>aperture</i> pada lensa	11
Gambar 2.3. Contoh DoF	11
Gambar 2.4. Implementasi <i>Shutter Speed</i>	12
Gambar 2.5. Contoh Citra Binner	13
Gambar 2.6. Citra Skala Keabuan	14
Gambar 2.7. Macam-macam aplikasi dari perbaikan kualitas citra (<i>Image enhancement</i>)	17
Gambar 2.8. Salah satu aplikasidarioperasispasial: <i>noisereduction</i>	18
Gambar 2.9. Contoh Inversi Citra	20
Gambar 2.10. Contoh penajaman citra dengan <i>unsharp masking</i>	21
Gambar 2.11. Citra asli, setelah di- <i>unsharp mask</i> , dan yang <i>oversharpen</i>	21
Gambar 2.12. Langkah-langkah proses <i>unsharp masking</i>	22
Gambar 2.13. Mencari citra <i>edge</i> pada proses <i>unsharp masking</i>	22
Gambar 2.14. Signal dari citra yang sudah mengalami proses <i>sharpening</i>	23
Gambar 2.15. Operatorpadaproseslengkap <i>unsharp masking</i>	24
Gambar 3.1. Tripod Takara, Lensa, dan extension macro (pixo)	32
Gambar 3.2. Mikrofilm Manuskrip Milik Museum Sonobudoyo.....	33
Gambar 3.3. Diagram Alir Metode Penelitian	34
Gambar 3.4. Perancangan <i>DIY Negative Scanner</i>	36
Gambar 3.5. Langkah Perbaikan Citra PD Mikrofilm dengan UMF.....	38
Gambar 3.6. Rancangan GUI Sistem Konversi Citra.....	39

Gambar 4.1 Tahap Proses Sistem	41
Gambar 4.2. <i>DIY Negative Scanner</i>	42
Gambar 4.3. Pengaturan Mode kamera Canon 60D	43
Gambar 4.4. Pengaturan Mode Lensa	43
Gambar 4.5. Pencahayaan dengan Lampu Philips 42 Watt	44
Gambar 4.6. Proses UMF pada citra mikrofilmIMG_2210.JPG	47
Gatasmbar 4.7. GUI Sistem Konversi Citra	49



DAFTAR TABEL

Tabel 2.1 Perbandingan Tinjauan Pustaka	9
Tabel 2.2 Simbol Flowchart	29
Tabel 4.1 Contoh Perbandingan citra pre dan pasca inversi.....	45
Tabel 4.2 Responden quisioner	51
Tabel 4.3 Hasil Quisioner (Citra yang dipilih reponden pada nilai k sekian)	51
Tabel 4.4 Prosentasi nilai k pada masing-masing responden	52
Tabel 4.5 Nilai PSNR pada 30 Citra operasi UMF52	53
Tabel 4.6 Sampel komparasi ketajaman citra PDUMF dengan nilai PSNR pada $k=0,5$ $k=2,5$ dan $k=4,5$	55



DAFTAR LIST

List 4.1 Kode Sumber Utama Algoritma Inversi	45
List 4.2 Kode Sumber Utama Algoritma UMF.....	48
List 4.2 Kode Sumber Algoritma PSNR.....	53



DAFTAR LAMPIRAN

Lampiran 1 Tabel Hasil Operasi UMS rentang $\sum_{0,1}^{0,9} k + 0,1$	60
Lampiran 2 Tabel Hasil Operasi UMS dengan rentang $\sum_5^{45} k + 5$	61
Lampiran 3 Listing Program invert-UMF-PSNR.....	62
Lampiran 4 Listing Program Algoritma UMF	69
Lampiran 5 Citra Hasil Inversi	72
Lampiran 6 Citra Hasil Operasi UMF	77



DAFTAR RINGKASAN

DIY : *Do it Yourself*

DSLR : *Digital Single Lens Reflex*

ISO : *International Organization Standard (ISO 5800:1987)*

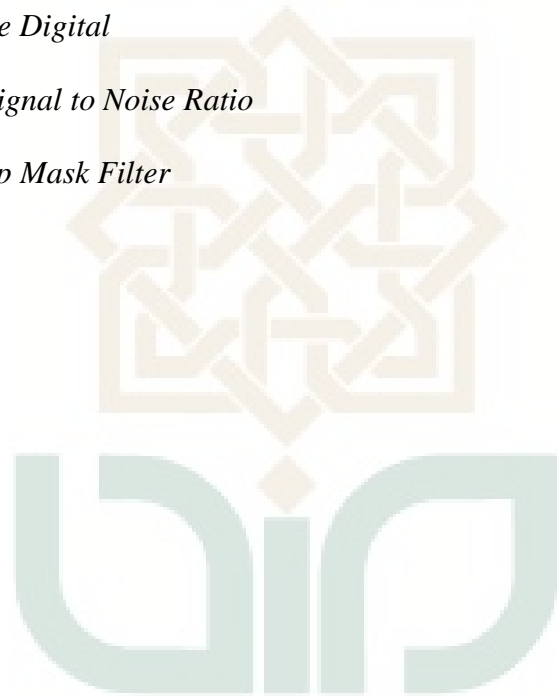
MSE : *Mean Squared Error*

ND : *Negative Digital*

PD : *Positive Digital*

PSNR : *Peak Signal to Noise Ratio*

UMF : *Unsharp Mask Filter*



**PERBAIKAN KUALITAS CITRA HASIL INVERSI DENGAN
MENGUNAKAN ALGORITMA UNSHARP MASK FILTER (UMF)
PADA ARSIP MIKROFILM NEGATIF MUSEUM SONOBUDOYO
YOGYAKARTA**

**Khuraim Fatik
NIM. 07650036**

INTISARI

Mikrofilm menjadi salah satu teknologi pengarsipan yang populer pada tahun 1980-an hingga tahun 1990-an. Konversi mikrofilm analog ke dalam format digital, *DIY Negative Scanner* atau dikenal dengan istilah pindai merupakan salah satu alternatif pengkonversian citra analog ke dalam format digital. Banyak kendala yang hadir dalam proses pengkonversian ini, seperti terciptanya derau (*Noise*), ketidaktajaman citra, dan berubahnya warna citra asli. Dalam penelitian ini penulis akan melakukan proses perbaikan citra disegmen penajaman citra mikrofilm.

Melalui penelitian ini penulis melakukan proses perbaikan citra melalui beberapa langkah, yaitu: Inversi Citra *Negative Digital* (DP) ke mode *Positive Digital* (PD), kemudian citra PD dikenai operasi *Unsharp Mask Filter* (UMF), yang pada akhirnya akan dianalisa menggunakan dua metode analisis, yang pertama analisa subyektif dengan responden pihak profesional dibidang fotografi dan mikrofilm, yang kedua analisa perhitungan *Peak Signal to Noise Ratio* (PSNR) adalah nilai perbandingan antara citra asli dan citra hasil *filtering*, Dalam hal ini antara citra PD dengan citra hasil operasi UMF.

Hasil akhir dari penelitian ini adalah hubungan antara nilai PSNR dan kualitas citra dalam konteks hasil operasi citra dengan metode UMF, diperoleh kesimpulan bahwa nilai PSNR berbanding terbalik dengan ketajaman citra. Semakin kecil nilai PSNR maka makin tajam citra yang dihasilkan oleh operasi UMF.

Kata Kunci : perbaikan citra digital, *unsharp mask filter*, *peak signal to noise ratio*, inversi, citra.

**INVERTED IMAGE ENHANCEMENT USING UNSHARP MASK FILTER
(UMF) FOR NEGATIVE MIKROFILM ARCHIVE OF MUSEUM
SONOBUDOYO YOGYAKARTA**

**Khuraim Fatik
NIM. 07650036**

ABSTRACT

Mikrofilm archiving become one of the popular technology at 1980s. Analogue mikrofilm conversion into digital format, DIY Negative Scanner was an alternative of analogue images conversion into digital format. Many obstacles raised in the conversion process, such as the noise production, Blurring image , and changing the original color image . In this study the authors will make the process of image enhancement segmented image enhancement mikrofilm .

This research was intended to image enhancement process through several steps: Negative Digital Image Inversion (ND) into Positive Digital (PD) mode , Then, the Possitive Digital was convoluted using Unsharp Mask Filter (UMF) operation. The result will be analyzed using two analytical methods, first based on respondent subjective analysis and professionals who has high concern on photography and mikrofilm. The second method was calculations Peak Signal to Noise ratio (PSNR), the comparison between the original image and the filtered image, in this case between PD image with the image of the operating results of UMF.

The result of this research is the relationship between PSNR and image quality in the context of the operating results of the image with UMF method , it is concluded that the PSNR value is inversely proportional to the sharpness of the image . The smaller the value the sharper the image PSNR generated by operations UMF.

Keyword: *image enhancement, unsharp mask filter, peak signal to noise ratio, inverse,image.*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Naskah kuno adalah warisan budaya yang hingga saat ini masih bisa dirasakan keberadaannya. Naskah kuno atau manuskrip merupakan dokumen dari berbagai macam jenis yang ditulis dengan tangan tetapi lebih mengkhhususkan kepada bentuk yang asli sebelum dicetak (Purwono, 2004). Naskah yang dimaksud adalah naskah yang mengandung nilai-nilai yang menyentuh berbagai aspek kehidupan masyarakat sebagai gambaran kehidupan manusia pada masa silam serta kebudayaannya. Nilai-nilai ini merupakan informasi kepada kita tentang bagaimana mereka hidup, pekerjaan sehari-hari, apa yang dirasakan dan bagaimana sikap hidup mereka (Ikram, 1983). Undang-undang menyebutkan bahwa naskah Kuno atau manuskrip adalah dokumen dalam bentuk apapun yang ditulis dengan tangan atau diketik yang belum dicetak atau dijadikan buku tercetak yang berumur 50 tahun lebih (UU Cagar Budaya No. 5 Bab 1 Pasal 2, 1992).

Karya-karya peradaban masa lampau banyak mengandung nilai budaya, nilai moral, nilai keagamaan, nilai sosial yang dapat menggambarkan kondisi sosial masyarakat pada masa tertentu dan sekaligus sebagai perwujudan jati diri suatu bangsa. Berbicara tentang naskah kuno, berarti berbicara tentang informasi, karena naskah kuno memiliki nilai informasi yang sangat berharga baik ditinjau dari aspek sejarah naskah tersebut maupun kandungan informasi yang termuat di dalam naskah tersebut. Maka perlu dilakukan preservasi terhadap

fisik naskah sesuai tujuan preservasi yaitu agar informasi yang terkandung di dalam manuskrip tersebut terjaga dan dapat digunakan secara optimal. (Primadesi, 2010)

Preservasi naskah kuno perlu untuk dilaksanakan agar warisan nenek moyang tidak musnah dan bermanfaat bagi kehidupan masyarakat. Upaya ini dilakukan melalui penyimpanan di berbagai perpustakaan dan museum, baik skala daerah, nasional, maupun internasional. Upaya tersebut mencakup restorasi, konservasi, dan pembuatan salinan (*backup*) naskah dalam bentuk media lain (digital dan analog). Pada tahun 1980-an hingga akhir tahun 1990-an, upaya pembuatan salinan naskah dilakukan dengan media mikrofilm. Seiring dengan perkembangan teknologi, aktivitas alih media pun mengalami revolusi penting pada awal milenium kedua, yakni dengan digunakannya teknologi digital dalam pembuatan salinan naskah, baik melalui kamera digital maupun mesin pindai. Aktivitas alih media teknologi mikrofilm pun mulai ditinggalkan, karena dianggap tidak efisien lagi, baik dalam upaya pembuatan maupun penggunaannya oleh pembaca (Fathurahman, 2009).

Di Yogyakarta sendiri upaya preservasi telah dilakukan oleh Museum Sonobudoyo Yogyakarta. Ery Sutiyadi selaku Kasi Koleksi, Konservasi, dan Preparasi memaparkan bahwa museum telah melakukan perawatan manuskrip secara preventif, yaitu melakukan pemantauan secara berkala dengan memanfaatkan teknologi pengatur *humidity*, untuk menjaga kondisi naskah maka temperature yang diijinkan berkisar antara 26-30⁰C, dan kelembaban udara sebesar 30-60%. Pada tahun 1986 Museum Sonobudoyo bekerjasama dengan The

Ford Foundation dari Amerika melakukan pengarsipan kedalam bentuk mikrofilm, sejak saat itu telah terarsip sebanyak sekitar 1250 naskah baik berupa lontar maupun kertas (Sustiyadi, 2013).

Mikrofilm adalah proses fotografi dimana dokumen atau arsip direkam pada film dalam ukuran yang diperkecil untuk memudahkan penyimpanan, transportasi, dan penggunaan (Amsyah, 2005). Mikrofilm merupakan teknologi grafis analog, sehingga pada era digital saat ini masyarakat ataupun pihak-pihak yang memanfaatkan teknologi mikrofilm akan menemukan beberapa kendala baik dalam menjaga maupun membacanya. Untuk menjaga keawetan mikrofilm perlu diperhatikan temperatur dan kelembaban ruang penyimpanannya, sedangkan untuk membaca mikrofilm perlu digunakan mikrofilm reader, sedangkan reader yang ada di Museum Sonobudoyo sudah rusak (Sustiyadi, 2013). Kendala seperti itulah yang menginspirasi penulis untuk menciptakan sebuah inovasi digitalisasi mikrofilm, yang ternyata sejalan dengan program Divisi Koleksi, Konservasi, dan Preparasi Museum Sonobudoyo.

Pemindaian mikrofilm kedalam format digital menggunakan alat pindai khusus yang dikenal dengan nama *Micro-image Capture*, mesin pemindai ini tidak seperti mesin pindai untuk *scanner flatbed negative film 16mm (negative film 16mm)* merupakan film yang digunakan pada kamera SLR analog, sedangkan mikrofilm memiliki dimensi 35mm) yang diperjualbelikan secara umum. Kelebihan dari *Micro-image Capture* adalah mampu melakukan pemindaian foto hingga 2.000 citra/ menit dan mampu menghasilkan citra digital dengan resolusi sampai 2592x1944. Namun fasilitas tersebut harus dibayar dengan harga

\$3,500.00 sampai \$ 60,000.00, jika dirupiahkan mesin pindai paling terjangkau seharga 30 juta-an (<http://www.microfilmworld.com>, 2013). Mesin *Micro-image Capture* mampu melakukan pemindaian foto hingga 2.000 citra/ menit (<http://www.nextscan.com>, 2013). *DIY Negative Scanner* menjadi alternative baru yang lebih ekonomis untuk melakukan pemindaian mikrofilm kedalam format digital, yaitu dengan menggunakan kamera DSLR (*Digital Single-Lens Reflex*) yang dimodifikasi sedemikian rupa. Namun dalam prakteknya mesin pindai buatan ini memiliki banyak kekurangan seperti: ketajaman sudut (*edge*) citra yang rendah, tingkat derau citra yang tinggi, dan ketajaman warna cenderung memudar.

Ada berbagai macam metode dalam perbaikan kualitas citra seperti: manipulasi kontras, pengurangan derau, penajaman garis batas citra (*edge crispening and sharpening*), interpolasi dan pembesaran gambar. Metode penajaman citra (*sharpening*). Dalam konteks pemindaian mikrofilm, metode penajaman citra menjadi salah satu alternatif untuk memperbaiki kualitas citra. Penajaman citra mengubah nilai piksel secara sistematis sehingga menghasilkan efek kenampakan citra yang lebih ekspresif sesuai dengan kebutuhan pengguna. Meliputi semua operasi yang menghasilkan citra baru dengan kenampakan visual dan karakteristik spektral yang berbeda. Penajaman Kontras ini bertujuan untuk memperoleh kesan kontras yang lebih tinggi. Dengan mentransformasi seluruh nilai kecerahan maka hasilnya adalah berupa citra baru dengan nilai maksimum awal, dan nilai minimum baru lebih rendah dari nilai minimum awal dan jika dilihat secara visual hasilnya berupa citra baru yang variasi hitam putihnya lebih

menonjol sehingga tampak lebih tajam dan memudahkan proses interpretasi. (Sulistiyo, Yos, & Filipus, 2009).

Proses *unsharp masking* merupakan cara yang sangat efektif untuk meningkatkan ketajaman terutama untuk citra hasil *scanning* yang terkadang ketajamannya kurang. Namun proses ini dapat menghasilkan efek-efek yang mengganggu dan tidak diinginkan, efek yang dihasilkan akibat *oversharpen* disebut efek *halo*. *Unsharp masking filter* adalah sebuah metode yang meningkatkan kualitas ketajaman garis (*edge*) dan elemen gambar dengan frekuensi tinggi lainnya melalui suatu prosedur yang mengurangi (*subtract*) gambar asli dengan versi citra asli yang kurang tajam atau telah dihaluskan untuk mendapatkan hasil citra yang tajam (Fisher, Perkins, Walker, Wolfart, 1994).

1.2. Rumusan Masalah

Berdasar latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengkonversi mikrofilm kedalam format digital menggunakan kamera *Digital Single Lens Reflex* (DSLR).
2. Bagaimana melakukan operasi inversi citra hasil pindai dari citra negatif ke positif.
3. Apakah Algoritma *Unsharp Masking Filter* dapat digunakan untuk memperbaiki kualitas citra.

1.3. Batasan Masalah

1. Penelitian ini menggunakan Kamera Canon EOS 60D Kit dengan

lensa 18-55mm sebagai alat pindai mikrofilm alternatif.

2. Obyek pindai adalah mikrofilm Museum Sonobudoyo Yogyakarta yang merupakan dokumen tradisional naskah-naskah kuno. Mikrofilm dalam penelitian ini memiliki ukuran film 35mm. Untuk film dengan ukuran 16mm menjadi sebuah batasan, karena dalam proses pemindaian tentunya diperlukan lensa kamera dengan panjang fokal lebih besar.
3. Algoritma *Unsharp Masking Filter* digunakan sebagai operasi perbaikan citra dari berbagai jenis citra yang tersedia di Museum Sonobudoyo Yogyakarta.
4. Perbandingan nilai noise citra asli dengan citra hasil *Unsharp Mask* digunakan untuk menghitung *Peak Signal to Noise Ratio* (PSNR).
5. Kualitas citra digital mikrofilm ditentukan dengan analisa subyektif pihak profesional.

1.4. Tujuan Penelitian

1. Digitalisasi mikrofilm untuk melestarikan dokumen warisan budaya.
2. Membuat alat pindai mikrofilm alternatif menggunakan kamera DSLR (*Digital Single Lens Reflex*)
3. Memperbaiki kualitas citra digital mikrofilm dengan menggunakan Algoritma *Unsharp Masking Filter*.

1.5. Manfaat Penelitian

Digitalisasi naskah-naskah kuno di Museum Sonobudoyo Yogyakarta dengan memanfaatkan kamera DSLR sebagai alat pindai mikrofilm merupakan sebuah alternative ekonomis dalam rangka menjaga dan melestarikan warisan bangsa. Selain itu hasil konversi citra analog kedalam format citra digital dapat dibaca dengan mudah bagi pengurus maupun pengunjung Museum (tanpa harus menggunakan mesin pindai mikrofilm atau *Micro-image Capture*).



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasar pada penelitian yang telah dilakukan oleh penulis mengenai hubungan *Peak Signal to Noise Ratio* dengan Metode perbaikan citra *Unsharp Masking Filter* (UMF) pada citra hasil pindai yang telah diinversikan dalam format citra positif digital. Kesimpulan dari penelitian diatas adalah:

1. Pemindaian dengan teknik *DIY Negative scanner* dapat digunakan untuk melakukan pemindaian mikrofilm analog kedalam format digital.
2. Teknik pindai yang memanfaatkan kamera *Digital Single Lens reflex* (DSLR), lampu rumah, dan *flatbed* buatan dengan bahan sekitar dapat menjadi salah satu alternatif untuk melakukan pemindaian mikrofilm.
3. Metode *Unsharp Masking Filter* (UMF) dapat digunakan untuk memperbaiki kualitas citra. Hal ini dibuktikan dengan analisa subyektif citra
4. Kualitas citra yang dikenai operasi *Unsharp Masking Filter* (UMF) dipengaruhi oleh nilai konstanta (k) pada algoritma UMF. Pada konteks perbaikan citra digital mikrofilm, citra berkualitas memiliki nilai k dari 0,5 hingga 2,5.

5.2. Saran

Penelitian yang telah dilakukan tentunya tidak lepas dari kekurangan pada eksperimen. Oleh karena itu penulis menyampaikan saran diantaranya:

1. Perlu dilakukan penelitian lebih lanjut dalam upaya perbaikan citra digital mikrofilm, dengan menggunakan algoritma yang lain, terutama dalam segmentasi *Spatial Operation Image Enhancement* (Perbaikan citra digital dalam operasi spasial yaitu segmentasi penajaman dan pengurangan noise pada citra digital)
2. Perlu dikembangkan metode lain untuk mengkonversi mikrofilm dari analog ke digital. Sekaligus pengembangan metode lain untuk perbaikan citra digital mikrofilm.

DAFTAR PUSTAKA

- Ahmad, U. (2005). *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.
- Al-Dhifery, N. I. (Oktober 2005). Comparison between two forms of Unsharp Masking enhancement method. *Journal of the University of Kerbala Vol. 3 No. 12th*.
- Amsyah, D. Z. (2005). *Manajemen Kearsipan*. Jakarta: PT. Gramedia Pustaka Utama.S
- Azmi. (2005). *Strategi Pengaturan Arsip Statis Pada Lembaga Kearsipan Dalam Upaya Meningkatkan Akses Dan Mutu Layanan Arsip Statis Kepada Publik*.
- Balza, A., & Firdaus, K. (2005). *Teknik Pengolahan Citra Digital Menggunakan Delphi*. Yogyakarta: Andi Publishing.
- Basuki, A., Palandi, J. F., & Fatchurrochman. (2005). *Pengolahan Citra Digital Menggunakan Visual Basic*. Yogyakarta: Graha Ilmu.
- Bilinski, J. (2010). *DSLR A900 as Negative Scanner*. New York: Friedman Archives.
- CanonCorporation. (2008). *Canon 60D Manual Book*. Japan: Canon.
- Fajri, A. (2005). *Desain dan Implementasi Sistem Komputasi Terdistribusi untuk Kompresi Citra Medis Sinar X Menggunakan JPEG 2000*. Bandung: ITB.
- Faqih, A. (2013, September). Pengaturan Kamera 60D untuk Swa-Pindai Mikrofilm. (K. Fatik, Interviewer)
- Fathurahman, D. O. (2009, Desember 27). Wawancara Republika: Nasib Manuskrip Islam Nusantara Memprihatinkan. (A. Rido, Interviewer) Republika.



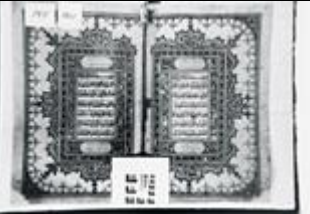
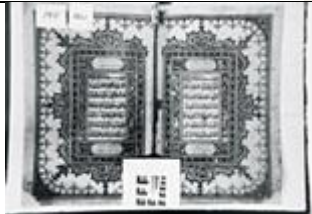
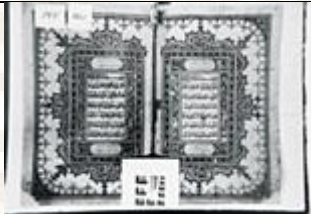
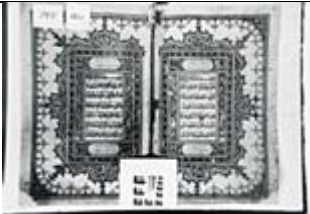
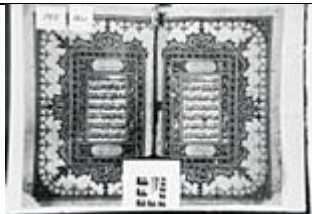
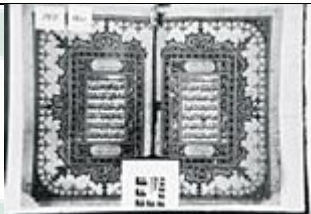
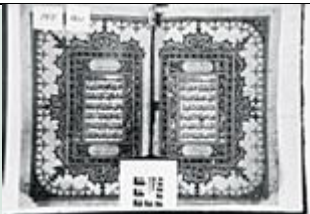
- Fitri, A. (2010). SKRIPSI. *Perbandingan Metode Low-Pass Filter Dan Median Filter Dalam Penghalusan Citra (Image Smoothing) Untuk Peningkatan Kualitas Citra (Image Enhancement)* .
- Gupta, G. (2011, November). International Journal of Soft Computing and Engineering (IJSCE); ISSN: 2231-2307, Volume-1, Issue-5,. *Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter.*
- Hargas, L., Hrianka, M., & Duga, A. (2007). *Noise Image Restoration by Spatial Filter.*
- <http://www.microfilmworld.com>. (2013).
<http://www.microfilmworld.com/filmandfichescanners.aspx>. Retrieved Oktober 3, 2013,
- <http://www.nextscan.com>. (2013). <http://www.nextscan.com/nextscan-products/?gclid=CODJvqis-7kCFRF64godVRgArw>. Retrieved Oktober 4, 2013, from <http://www.nextscan.com/nextscan-products/?gclid=CODJvqis-7kCFRF64godVRgArw>
- Ikram, A. (1983). *Beberapa Masalah Perkembangan Ilmu Filologi Dewasa Ini untuk Matakuliah Ilmu Filologi dan Penerapannya*. Jakarta : Universitas Indonesia.
- Katzenbeisser, S. (2000). *Information Hiding: Technique for Steganography and Watermarking*. London: Artech House.

- Lamabelawa, M. I. (2013). <http://jawawuan.web.ugm.ac.id/imagenegatif.pdf>. Retrieved September 29, 2013, from <http://www.ugm.ac.id/>: <http://jawawuan.web.ugm.ac.id/imagenegatif.pdf>
- Listiyani, E. (2009). Skripsi. *Implementasi Adaptive Median Filter Sebagai Reduksi Noise Pada Citra Digital* .
- Mahmood, N. H., Razif, M. R., & Gany, M. T. (2011, September). Comparison between Median, Unsharp and Wiener filter and its effect on ultrasound stomach tissue image segmentation for Pyloric Stenosis. *International Journal of Applied Science and Technology Vol. 1 No. 5* .
- Primadesi, Y. (2010). Jurnal Bahasa dan Seni Vol 11 No. 2 Tahun 2010 (120 - 127). *Peran Masyarakat Lokal dalam Usaha Pelestarian Naskah-Naskah Kuno Paseban* .
- Purwono. (2004). *Buku dan Perpustakaan: Catatan Memori Bangsa Pembangkit Nasionalisme*. Jakarta: Pustaka Sinar Harapan.
- Sudarsono. (2005). *Flowchart*. Retrieved Februari 6, 2013, from <http://sudarsono.staff.gunadarma.ac.id/Downloads/files/16512/Flowchart.pdf>
- Sukanto. (2004, Januari). SKRIPSI. *Perbandingan Ketelitian Antara Metode Gaussian Low Pass Filtering dan Metode Butterworth Low Pass Filtering dalam Noise Reduction Citra*.
- Sulistiyo, W., Yos, R. B., & Filipus, F. Y. (2009, November 14). Konferensi Nasional Sistem dan Informatika 2009. *Analisis Penerapan Metode Median Filter Untuk Mengurangi Noise Pada Citra Digital* , 189-190.

- Susilo, R. C. (2005). SKRIPSI. *Perancangan Program Aplikasi Noise Reduction dengan Metode Median Filter dan Weighted Median Filter.*
- Sutiyadi, E. (2013, September 6). Wawancara Penelitian: Optimalisasi Konversi Format Analog ke Digital. (K. Fatik, Interviewer, & K. Fatik, Editor)
Yogyakarta, Daerah Istimewa Yogyakarta, Indonesia.
- Undang-Undang Cagar Budaya No. 5 Bab 1 Pasal 2. (1992).
- Widiatmoko, D., & Bharata, J. W. (2006). *101 Tips dan Trik Fotografi.* Jakarta:
PT. Elex Media Komputindo.
- Zakaria, R. (2007). Perancangan Program Aplikasi Peningkatan Ketajaman Gambar Dengan Metode Gaussian Low-Pass Filter. *SKRIPSI BINUS* .

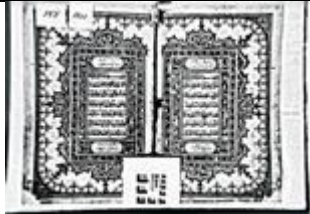
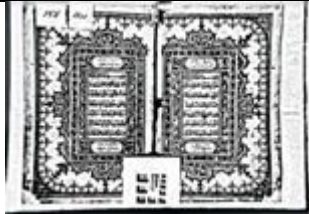
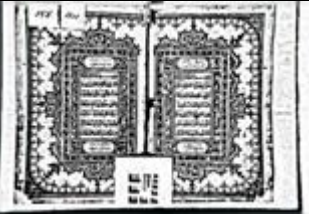
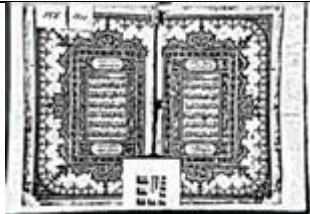
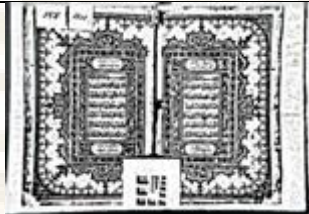
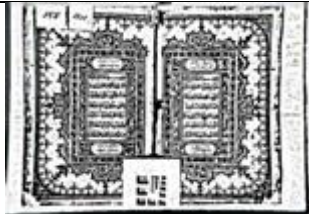
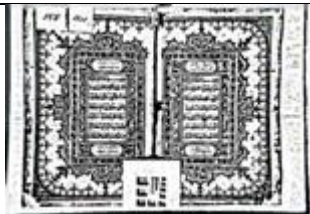
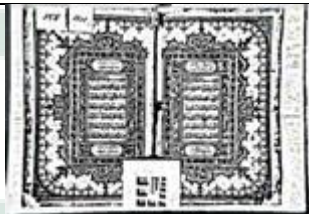
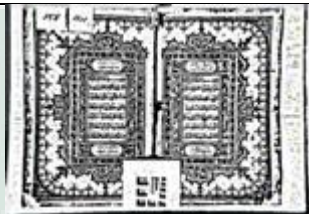
LAMPIRAN 1

Tabel Hasil Operasi UMS dengan rentang $\sum_1^9 k + 0,1$ pada citra IMG_2217.JPG

Citra PDUMF		
0,1	0.2	0.3
		
10.12446796520436	9.866186456141644	9.612674705821531
0.4	0.5	0.6
		
9.462171601932292	9.031815165986345	8.601634357129855
0.7	0.8	0.9
		
8.421458083954967	8.225023057477333	8.09030977547832

LAMPIRAN 2

Tabel Hasil Operasi UMS dengan rentang $\sum_1^9 k + 5$ pada citra IMG_2217.JPG

Citra PDUMF		
0,1	0.2	0.3
		
10.12446796520436	9.866186456141644	9.612674705821531
0.4	0.5	0.6
		
9.462171601932292	9.031815165986345	8.601634357129855
0.7	0.8	0.9
		
8.421458083954967	8.225023057477333	8.09030977547832

LAMPIRAN 3

Listing Program invert-UMF-PSNR dalam MainForm.Java

```
package psnrproj;

import java.awt.image.BufferedImage;
import java.awt.image.BufferedImageOp;
import java.awt.image.LookupOp;
import java.awt.image.ShortLookupTable;
import java.io.BufferedInputStream;
import java.io.BufferedWriter;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

/**
 * @author java
 */
public class MainForm extends javax.swing.JFrame {

    /**
     * Creates new form MainForm
     */
    private String alamatFile;
    private String alamatFile2;
    private String namaFile;
    private String namaFile2;
    private String ekstensi;
    private String path;
    public FileInputStream fis1, fis2;
    public InputStream is1, is2;
    public DataInputStream dis1, dis2;
    private static final short[] invertTable;
    private Thread t;

    static {
        invertTable = new short[256];
        for (int i = 0; i < 256; i++) {
            invertTable[i] = (short) (255 - i);
        }
    }

    public MainForm() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
}
```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    label_input = new javax.swing.JLabel();
    btnInput = new javax.swing.JButton();
    btn_proses = new javax.swing.JButton();
    status = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    label_input.setBackground(new java.awt.Color(51, 51, 255));
    label_input.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    label_input.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 255)));

    btnInput.setText("BUKA");
    btnInput.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnInputActionPerformed(evt);
        }
    });

    btn_proses.setText("PROSES");
    btn_proses.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btn_prosesActionPerformed(evt);
        }
    });

    status.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(61, 61, 61)
                .addComponent(btnInput, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
java.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(btn_proses, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
java.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(status, javax.swing.GroupLayout.DEFAULT_SIZE, 547,
Short.MAX_VALUE)
                .addContainerGap())
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(61, 61, 61)
                    .addComponent(label_input, javax.swing.GroupLayout.PREFERRED_SIZE,
695, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(54, Short.MAX_VALUE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(539, 539, 539)
                    .addComponent(status, javax.swing.GroupLayout.PREFERRED_SIZE, 34,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnInput, javax.swing.GroupLayout.PREFERRED_SIZE,
34, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(btn_proses, javax.swing.GroupLayout.PREFERRED_SIZE,
34, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(19, 19, 19))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(label_input, javax.swing.GroupLayout.PREFERRED_SIZE,
488, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(93, Short.MAX_VALUE))
    );

pack();
} // </editor-fold>

private void btnInputActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
JFileChooser jfc = new JFileChooser("D:\\");
int hasil = jfc.showOpenDialog(this);
if (hasil == JFileChooser.APPROVE_OPTION) {
    alamatFile = jfc.getSelectedFile().getAbsolutePath();
    namaFile = jfc.getSelectedFile().getName();
    this.path = jfc.getCurrentDirectory().getAbsolutePath();
    boolean ekstensi = jfc.getSelectedFile().getName().endsWith(".JPG");
    boolean ekstensi2 = jfc.getSelectedFile().getName().endsWith(".jpg");
    //boolean ekstensi2=jfc.getSelectedFile().getName().endsWith(".JPG");
    if (ekstensi == true) {
        label_input.setIcon(new javax.swing.ImageIcon(alamatFile));

        //jTextField1.setText("png");
    } else if (ekstensi2 == true) {
        label_input.setIcon(new javax.swing.ImageIcon(alamatFile));
    } else {
        JOptionPane.showMessageDialog(null, "File Tidak dapat dibuka, \npilih
gambar dengan format jpg", "Peringatan", JOptionPane.ERROR_MESSAGE);
    }
    //JOptionPane.showMessageDialog(null,alamatFile+" Nama file : "+namaFile);
}
}

private void btn_prosesActionPerformed(java.awt.event.ActionEvent evt) {

    if(alamatFile!=null){
        if(alamatFile.isEmpty()==false){
            btn_proses.setEnabled(false);
            t = new Thread(new Worker(this));
            t.start();
        }
    }else{
        JOptionPane.showMessageDialog(this, "Pilih file gambar yang akan di proses");
    }
}

public void INVERT() {
    BufferedImage orig;
    BufferedImage inverted;
    try {

```

```

String namafile = alamatFile;
String ext = namafile.substring(namafile.lastIndexOf("."));
//System.out.println("Memroses file " + namafile);
orig = ImageIO.read(new BufferedInputStream(new FileInputStream(alamatFile)));
inverted = invertImage(orig);
if (ext.equalsIgnoreCase(".jpeg")) {
    this.ekstensi = "jpeg";
    ImageIO.write(inverted, "jpeg", new File(alamatFile + "inverted.jpeg"));
} else if (ext.equalsIgnoreCase(".jpg")) {
    this.ekstensi = "jpg";
    ImageIO.write(inverted, "jpg", new File(alamatFile + "inverted.jpg"));
} else if (ext.equalsIgnoreCase(".png")) {
    this.ekstensi = "png";
    ImageIO.write(inverted, "png", new File(alamatFile + "inverted.png"));
} else {
    System.out.println("Gambar tidak mendukung");
}
} catch (final IOException e1) {
    e1.printStackTrace();
} finally {
    orig = null;
    inverted = null;
}
}

private BufferedImage invertImage(final BufferedImage src) {
    final int w = src.getWidth();
    final int h = src.getHeight();
    final BufferedImage dst = new BufferedImage(w, h, BufferedImage.TYPE_INT_RGB);

    final BufferedImageOp invertOp = new LookupOp(new ShortLookupTable(0,
invertTable), null);
    return invertOp.filter(src, dst);
}

private double PSNR(String alamatFile, String alamatFile2) {
    try {
        // TODO add your handling code here:

        try {
            fis1 = new FileInputStream(alamatFile);
            is1 = new BufferedInputStream(fis1, 512);
            dis1 = new DataInputStream(is1);
            fis2 = new FileInputStream(alamatFile2);
            is2 = new BufferedInputStream(fis2, 512);
            dis2 = new DataInputStream(is2);
        } catch (FileNotFoundException ex) {
            System.err.println(ex);
        }

        int L = 512 * 512;
        double m = 0;
        int i;
        for (i = 0; i < L; i++) {
            try {
                int r1 = dis1.read();
                int r2 = dis2.read();
                int r3 = r1 - r2;
                m += (double) Math.pow(r3, 2);
            } catch (IOException ex) {
                System.err.println(ex);
            }
        }
    }
}

```

```

    }
    if (m == 0) {
        System.out.println("PSNR=*");
    } else {
        double p = 10 * Math.log((255 * 255) / (m / L)) / Math.log(10);
        //txt_psnr.setText(String.valueOf(p));
        return p;
    }
    dis1.close();
    dis2.close();
} catch (IOException ex) {
    System.err.println(ex);
}
return 0;
}
// Variables declaration - do not modify
private javax.swing.JButton btnInput;
private javax.swing.JButton btn_proses;
private javax.swing.JLabel label_input;
private javax.swing.JLabel status;
// End of variables declaration

public void USM() {

    for (int i = 1; i <= 9; i++) {
        String imageFilename = alamatFile + "inverted."+this.ekstensi;

        File originalImgFile = new File(imageFilename);
        File blurredImgFile = new File(imageFilename+"_blurred."+this.ekstensi);
        File usmImgFile = new File( this.path+"\\\\"+i+"_sharpened."+this.ekstensi );

        BufferedImage loadedImage = null;
        long startTime = System.currentTimeMillis();
        System.out.printf("At %tT.%<tL Loading image '%s' from disk...\n", startTime,
imageFilename);
        try {
            loadedImage = ImageIO.read(originalImgFile);
        } catch (Exception e) {
            System.out.println("Error opening image" + e.toString());
        }
        int imgWidth = loadedImage.getWidth();
        int imgHeight = loadedImage.getHeight();

        System.out.printf("Loaded image '%s' of size (%d x %d) pixels to an
BuffredImage object..\n",
            imageFilename, imgWidth, imgHeight);

        //the following line defines the small box for filter kernel
        int boxWidth = 20, boxHeight = 20;
        //the parameters, (left, top, seWidth, selHeight) defines a selection of
original image
        int left = 10, top = 10; //left and top should be more than half of box width
and height
        //selection width and height is chosen to center the selection
        //can also be used any given selection
        int right = (imgWidth - left), bottom = (imgHeight - top);

        //USM filter parameters
        float usmAmount = 0.1f*i;
        int usmThreshold = 3;

        USM u = new USM();
    }
}

```



```

int[][] origPixels = u.loadFromImgBuffer(loadedImage);
System.out.printf("Loaded BufferedImage RGB data to a 2D array...\n");
int[][] blurredPixels = new int[imgWidth][imgHeight];

System.out.printf("Now applying boxcar blur filter. Please wait...\n");
//applying the boxcar filter with given parameters and save the result in
//another 2D array called blurredImage
try {
    //both the loadedImage and blurredPixels will hold the result of the
boxcar function and
    u.boxCar(loadedImage, origPixels, blurredPixels, left, top,
            right, bottom, boxWidth, boxHeight);
    System.out.printf("Filtering finished, saving the blurred image.\n");
    //save the loadedImage imagebuffer to a new file
    try {
        ImageIO.write(loadedImage, "jpg", blurredImgFile);
        System.out.printf("Saved the blurred image '%s'\n",
blurredImgFile.getName());
    } catch (Exception e) {
        System.out.println("Error saving blurred image" + e.toString());
    }

    System.out.printf("Now applying unsharp mask filter. Please wait...\n");
    //sharpenes the image using USM and it back to loadedImage ImageBuffer
object
    u.unsharpMask(loadedImage, origPixels, blurredPixels, left, top,
            right, bottom, usmAmount, usmThreshold);

    //save the imagebuffer to a new file
    try {
        ImageIO.write(loadedImage, "jpg", usmImgFile);
        long elapsed = System.currentTimeMillis() - startTime;
        System.out.printf("Elapsed %5d mS, saved the sharpened image '%s'\n",
elapsed, usmImgFile.getName());
    } catch (Exception e) {
        System.out.println("Error saving sharpened image" + e.toString());
    }

    } catch (Exception e) {
        System.out.printf("Image selection is probably too large:(%d, %d, %d,
%d)\n%s\n",
            left, top, right, bottom, e.toString());
    }
}

public void PSNRcomparation() {
    String hasil="";
    for(int i=1;i<=9;i++){
        double p = PSNR(alamatFile+"inverted."+this.ekstensi,
this.path+"\\ "+i+"_sharpened."+this.ekstensi);
        hasil+="PSNR pada nilai "+i+" =
"+String.valueOf(p)+"\n"+System.getProperty("line.separator");
    }

    try {
        //File txt = new File();
        System.out.println(path);
        FileWriter fw = new FileWriter(path+"\\HasilPSNR.txt");
        BufferedWriter out = new BufferedWriter(fw);
        out.write(hasil);

        out.close();
    }
}

```

```

        } catch (IOException ex) {
            Logger.getLogger(MainForm.class.getName()).log(Level.SEVERE, null, ex);
        }

    }

    public void setStatus(String st){
        status.setText(st);
    }

    public void setProsesEnable(){
        btn_proses.setEnabled(true);
    }

    public void setProsesFinish(){
        JOptionPane.showMessageDialog(this, "Proses selesai");
    }

    public class Worker implements Runnable{

        MainForm mn;

        public Worker(MainForm mn){
            this.mn=mn;
        }

        @Override
        public void run() {
            mn.setStatus("Tahap1: Harap Tunggu Proses Sedang Melakukan Inversi
Gambar...");
            mn.INVERT();
            mn.setStatus("Tahap2: Harap Tunggu Proses Sedang Melakukan Unsharp Mask...");
            mn.USM();
            mn.setStatus("Tahap3: Harap Tunggu Proses Sedang Melaskukan Komparasi
PSNR...");
            mn.PSNRcomparation();
            mn.setStatus("");
            mn.setProsesEnable();
            mn.setProsesFinish();
        }

    }
}

```

LAMPIRAN 4

Listing Program Algoritma UMF

```
package psnrproj;

import java.awt.image.BufferedImage;
import java.io.File;
import javax.imageio.ImageIO;

public class USM {

    //This is the boxcar filter function accepting a 2D array of RGB pixels,
    //A selection for the filter area and a filter kernel size.
    //it also fills an imagebuffer object with blurred image data, which can be saved to
    disk
    public void boxCar(BufferedImage loadedImage, int[][] origPixels,
        int[][] blurredPixels, int left, int top, int right, int bottom,
        int filtX, int filtY) throws ArrayIndexOutOfBoundsException {

        //a boundaries of a small 2d array containing adjacent box of pixels for a given
        //pixel is sent to blurPixels function. This function processes for R,G,B and
        //alpha bytes in each pixel in the box seperately and reconstucts the averaged
        pixel

        for (int j = top; j < bottom; j++) {
            for (int i = left; i < right; i++) {

                //blur pixels using averaging a box of pixels surrounding the given
                //pixel (i,j) and saving the result in both blurredPixels and
                //loadedImage ImageBuffer object
                blurredPixels[i][j] = blurPixels(origPixels, (i - filtX / 2), (j - filtY /
                2),
                    (i + filtX / 2), (j + filtY / 2));
                loadedImage.setRGB(i, j, blurredPixels[i][j]);
            }
        }

        //This function returns a RGB value taking the mean of
        //RGB values of each pixel in the filter kernel box
        public int blurPixels(int[][] origPixels, int left, int top, int right, int bottom) {
            //transperency is not considered
            int alpha = 0xff000000, red = 0, green = 0, blue = 0;
            int boxSize = (right - left) * (bottom - top);

            //the following nested for loops takes the sum of RGB components of each
            //pixels in the box.
            for (int q = top; q < bottom; q++) {
                for (int p = left; p < right; p++) {
                    int pixel = origPixels[p][q];
                    red += ((pixel >> 16) & 0xff);
                    green += ((pixel >> 8) & 0xff);
                    blue += ((pixel) & 0xff);
                }
            }
            //average is computed using integer arithmetic. If the box size is too large
            //this routine will fail. max box size = (INT_Max/256) = 8,388,608
            red /= boxSize;
            green /= boxSize;
            blue /= boxSize;
            //returns the reconstructed pixel back
            return (alpha | (red << 16) | (green << 8) | blue);
        }
    }
}
```

```

    }
    //average is computed using integer arithmetic. If the box size is too large
    //this routine will fail. max box size = (INT_Max/256) = 8,388,608
    red /= boxSize;
    green /= boxSize;
    blue /= boxSize;
    //returns the reconstructed pixel back
    return (alpha | (red << 16) | (green << 8) | blue);
}

//this function calculates the unsharpmask by subtracting originalpixels
//by blurred pixels. Then it sharpenes the original image by adding a
//weighted amount of the unsharpmask back to the original depending on
//the given threshold value
public void unsharpMask(BufferedImage usmImage,
    int[][] origPixels, int[][] blurredPixels, int left,
    int top, int right, int bottom, float amount, int threshold) {

    int orgRed = 0, orgGreen = 0, orgBlue = 0;
    int blurredRed = 0, blurredGreen = 0, blurredBlue = 0;
    int usmPixel = 0;
    int alpha = 0xFF000000; //transperency is not considered and always zero

    for (int j = top; j < bottom; j++) {
        for (int i = left; i < right; i++) {
            int origPixel = origPixels[i][j], blurredPixel = blurredPixels[i][j];

            //seperate RGB values of original and blurred pixels into seperate R,G and
            B values
            orgRed = ((origPixel >> 16) & 0xff);
            orgGreen = ((origPixel >> 8) & 0xff);
            orgBlue = (origPixel & 0xff);
            blurredRed = ((blurredPixel >> 16) & 0xff);
            blurredGreen = ((blurredPixel >> 8) & 0xff);
            blurredBlue = (blurredPixel & 0xff);

            //If the absolute val. of difference between original and blurred
            //values are greater than the given threshold add weighed difference
            //back to the original pixel. If the result is outside (0-255),
            //change it back to the corresponding margin 0 or 255
            if (Math.abs(orgRed - blurredRed) >= threshold) {
                orgRed = (int) (amount * (orgRed - blurredRed) + orgRed);
                orgRed = orgRed > 255 ? 255 : orgRed < 0 ? 0 : orgRed;
            }

            if (Math.abs(orgGreen - blurredGreen) >= threshold) {
                orgGreen = (int) (amount * (orgGreen - blurredGreen) + orgGreen);
                orgGreen = orgGreen > 255 ? 255 : orgGreen < 0 ? 0 : orgGreen;
            }

            if (Math.abs(orgBlue - blurredBlue) >= threshold) {
                orgBlue = (int) (amount * (orgBlue - blurredBlue) + orgBlue);
                orgBlue = orgBlue > 255 ? 255 : orgBlue < 0 ? 0 : orgBlue;
            }

            usmPixel = (alpha | (orgRed << 16) | (orgGreen << 8) | orgBlue);
            usmImage.setRGB(i, j, usmPixel);
        }
    }
}

//function to load ARGB values of each pixel in to a 2D array

```

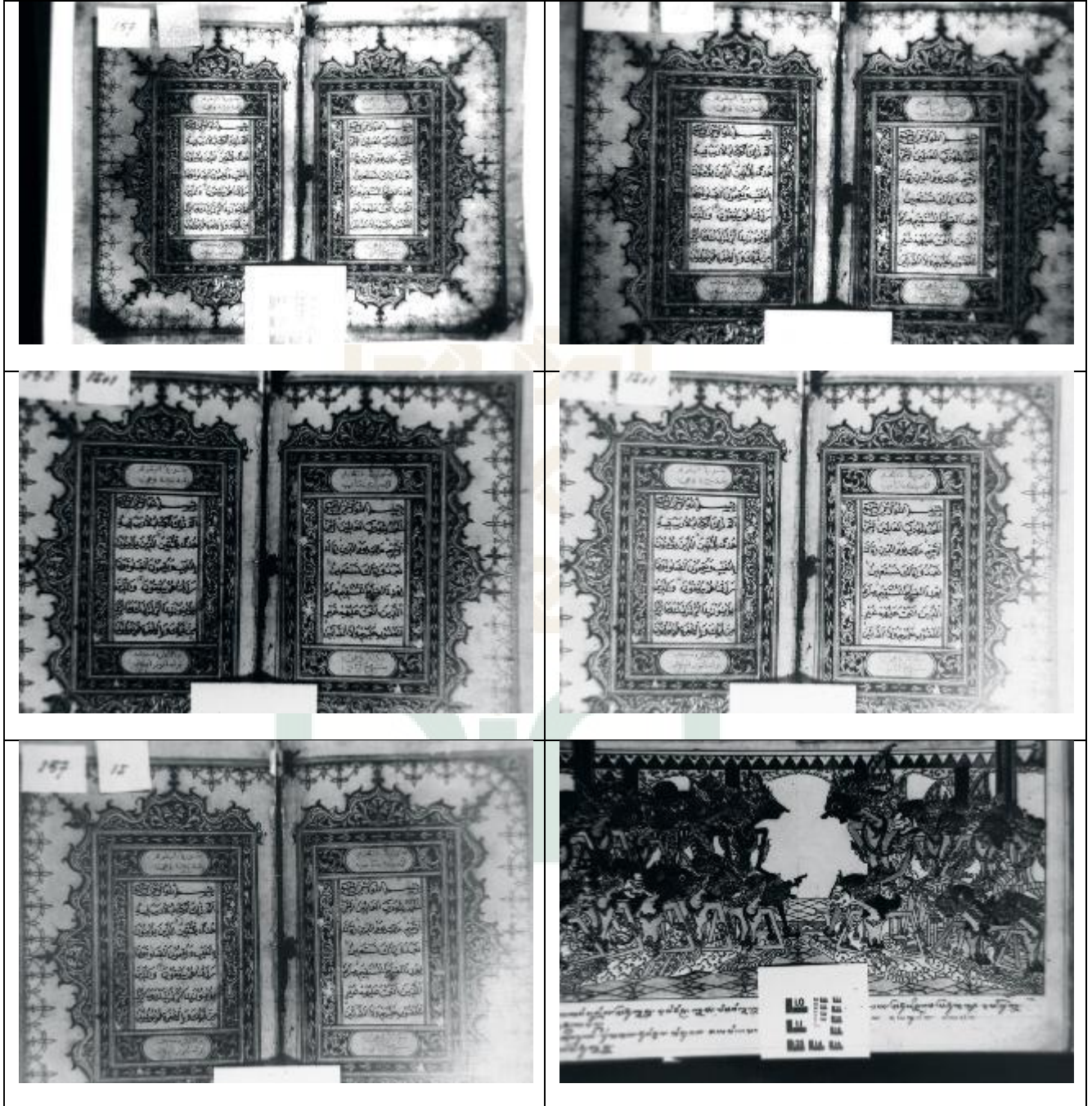
```
        usmPixel = (alpha | (orgRed << 16) | (orgGreen << 8) | orgBlue);
        usmImage.setRGB(i, j, usmPixel);
    }
}

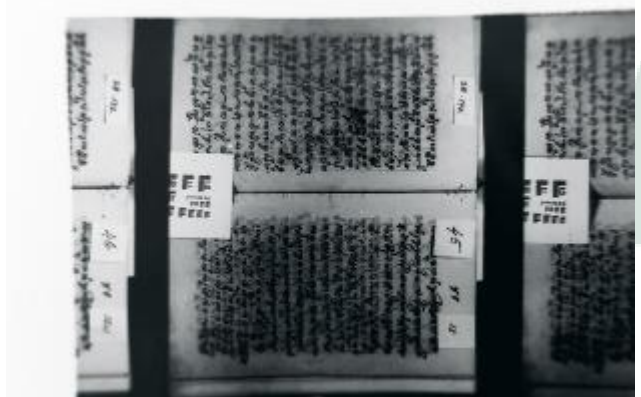
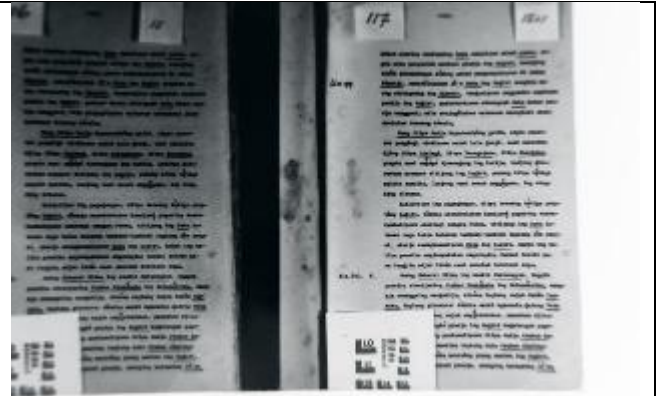
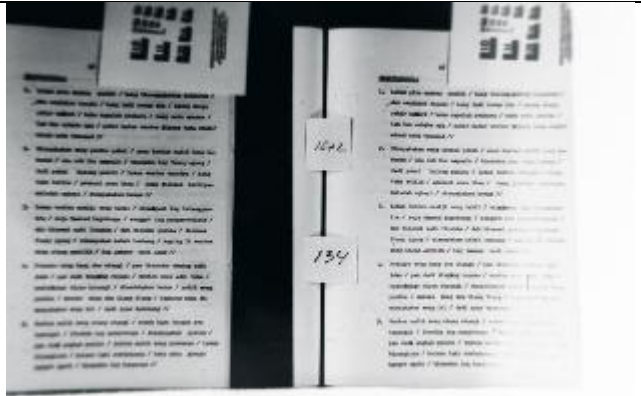
//function to load ARGB values of each pixel in to a 2D array
public int[][] loadFromImgBuffer(BufferedImage image) {
    int width = image.getWidth(), height = image.getHeight();
    int[][] pixels = new int[width][height];
    for (int j = 0; j < height; j++) {
        for (int i = 0; i < width; i++) {
            pixels[i][j] = image.getRGB(i, j);
        }
    }
    return pixels;
}
}
```



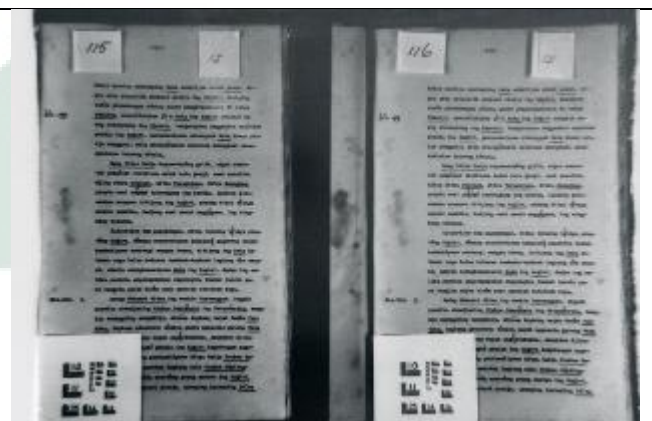
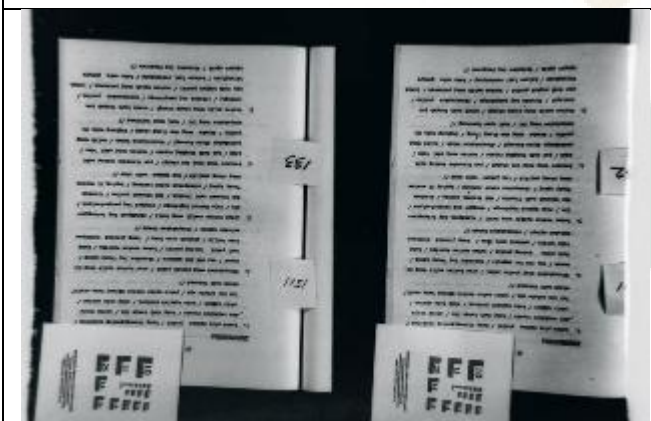
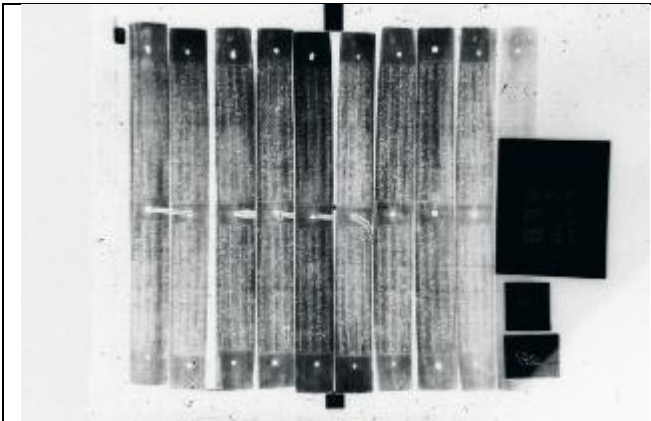
LAMPIRAN 5

Tabel Citra Hasil Inversi




















LAMPIRAN 6

Tabel hasil operasi UMF pada citra IMG_2201.JPG

 <i>k: 0,5 PSNR: 10.467</i>	 <i>k: 1 PSNR: 9.485</i>	 <i>k: 1,5 PSNR: 9.053</i>
 <i>k: 2 PSNR: 8.719</i>	 <i>k: 2,5 PSNR: 8.445</i>	 <i>k: 3 PSNR: 8.239</i>
 <i>k: 3,5 PSNR: 8.085</i>	 <i>k: 4 PSNR: 7.978</i>	 <i>k: 4,5 PSNR: 7.877</i>