



**PERANCANGAN APLIKASI DETEKSI TEPI DENGAN
METODE SOBEL DAN PREWITT UNTUK PENGENALAN
POLA MENGGUNAKAN VISUAL STUDIO 2013**

Disusun dan Diajukan Untuk Memenuhi Persyaratan Ujian Akhir
Memperoleh Gelar Sarjana Komputer Pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH:

NAMA : YUKE OCTAVIA IRAWAN
NPM : 1514370753
PROGRAM STUDI : SISTEM KOMPUTER

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCABUDI
MEDAN
2019**

LEMBAR PENGESAHAN

**PERANCANGAN APLIKASI DETEKSI TEPI DENGAN
METODE SOBEL DAN PREWITT UNTUK PENGENALAN
POLA MENGGUNAKAN VISUAL STUDIO 2013**

Disusun Oleh :

Nama : Yuke Octavia Irawan
Npm : 1514370753
Program Studi : Sistem Komputer

**Skripsi telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 8 November 2019**

Dosen Pembimbing I


Eko Hariyanto, S.Kom., M.Kom

Dosen Pembimbing II


Hermansyah, S.Kom., M.Kom

Mengetahui,

Dekan Fakultas Sains dan Teknologi


Sri Shindi Indira, S.T., M.Sc

Ketua Program Studi


Eko Hariyanto, S.Kom., M.Kom

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini :

Nama : Yuke Octavia Irawan
Npm : 1514370753
Prodi : Sistem Komputer
Konsentrasi : Keamanan Jaringan Komputer
Judul Skripsi : Perancangan Aplikasi Deteksi Tepi Dengan Metode Sobel Dan Prewitt Untuk Pengenalan Pola Menggunakan Visual Studio 2013

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil Plagiat
2. Saya tidak akan menuntut perbaikan nilai indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau
3. Skripsi saya dapat dipublikasikan oleh pihak lembaga, dan saya tidak akan menuntut akibat publikasi tersebut

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya, terima kasih

Medan, 19 November 2019



Yuke Octavia Irawan
(1514370753)

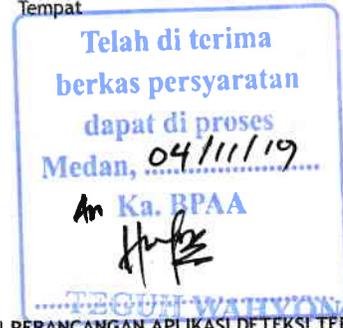
Telah Diperiksa oleh LPMU
dengan Plagiarisme... 47%
Medan, 04 NOV 2019

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau



Medan, 04 November 2019
Kepada Yth : Bapak/Ibu Dekan
Fakultas SAINS & TEKNOLOGI
UNPAB Medan
Di -
Tempat



Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama : YUKE OCTAVIA IRAWAN
Tempat/Tgl. Lahir : Binjai / 11 Oktober 1998
Nama Orang Tua : IRWAN SYAH
N. P. M : 1514370753
Fakultas : SAINS & TEKNOLOGI
Program Studi : Sistem Komputer
No. HP : 082272823998
Alamat : Jl. Gaharu Binjai Utara

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul PERANCANGAN APLIKASI DETEKSI TEPI DENGAN METODE SOBEL DAN PREWIT UNTUK PENGENALAN POLA MENGGUNAKAN VISUAL STUDIO 2013, Selanjutnya saya menyatakan :

- Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
- Tidak akan menuntun ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
- Telah tercap keterangan bebas pustaka
- Terlampir surat keterangan bebas laboratorium
- Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
- Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
- Tertampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
- Skripsi sudah dijilid lux 2 examplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 examplar untuk penguji (bentuk dan warna penjiilid diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
- Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
- Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
- Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
- Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	100.000
2. [170] Administrasi Wisuda	: Rp.	1.500.000
3. [202] Bebas Pustaka	: Rp.	100.000
4. [221] Bebas LAB	: Rp.	5.000
Total Biaya	: Rp.	1.705.000

uang kuliah
(10% dan 1 tahun)

Rp. 2.875.000
Rp. 4.580.000

4/11/2019 (Fani)

Ukuran Toga :

S



Hormat saya
YUKE OCTAVIA IRAWAN
1514370753

Catatan :

- 1. Surat permohonan ini sah dan bertaku bila ;
 - a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
 - b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2. Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.





UNIVERSITAS PEMBANGUNAN PANCA BUDI FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)

PERMOHONAN JUDUL TESIS / SKRIPSI / TUGAS AKHIR*

Saya yang bertanda tangan di bawah ini :

Nama Lengkap	: YUKE OCTAVIA IRAWAN
Tempat/Tgl. Lahir	: BINJAI / 11 Oktober 1998
Nomor Pokok Mahasiswa	: 1514370753
Program Studi	: Sistem Komputer
Konsentrasi	: Keamanan Jaringan Komputer
Jumlah Kredit yang telah dicapai	: 141 SKS, IPK 3.40
Nomor Hp	: 082272823998
Dengan ini mengajukan judul sesuai bidang ilmu sebagai berikut :	

No.	Judul
1.	PERANCANGAN APLIKASI DETEKSI TEPI DENGAN METODE SOBEL DAN PREWIT UNTUK PENGENALAN POLA MENGGUNAKAN VISUAL STUDIO 2013

Perhatian : Diisi Oleh Dosen Jika Ada Perubahan Judul

Stempel Yang Tidak Perlu



Medan, 1 Maret 2019
Pemohon,

(Yuke Octavia Irawan)

Tanggal : 1 Maret 2019
Disetujui oleh:
Dekan

(Sri Shindi Indana, S.Pd., M.Pd., M.Pi., M.Sc.)

Tanggal : 1 Maret 2019
Disetujui oleh:
Dosen Pembimbing I :

(Eko Hariyanto, S.Kom., M.Kom)

Tanggal : 1 Maret 2019
Disetujui oleh:
Ka. Prodi Sistem Komputer

(Eko Hariyanto, S.Kom., M.Kom)

Tanggal : 1 Maret 2019
Disetujui oleh:
Dosen Pembimbing II:

(Hermansyah, S.Kom., M.Kom)

Plagiarism Detector v. 1281 - Originality Report

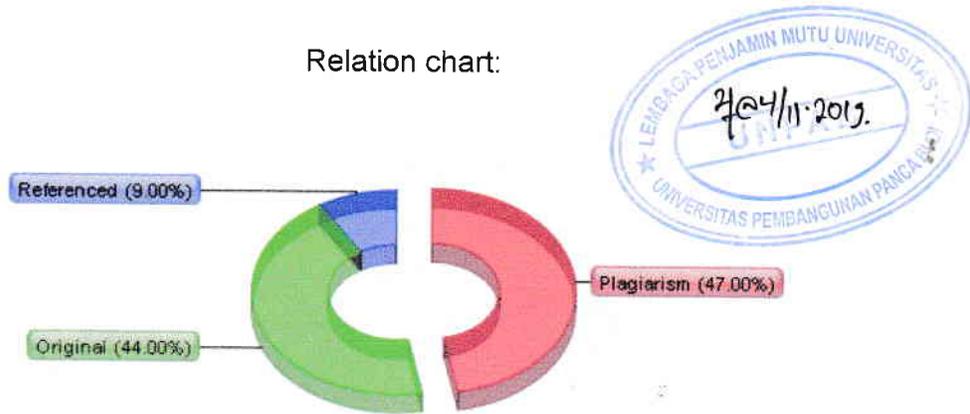
Analyzed document: 04/11/2019 17:18:00

"YUKE OCTAVIA IRAWAN_1514370753_SISTEM KOMPUTER.docx"

Check Type: Internet - via Google and Bing

Licensed to: Universitas Pembangunan Panca Budi_License2

Relation chart:



Distribution graph:

Comparison Preset: Rewrite. Detected language: Indonesian

Top sources of plagiarism:

- 10% wrds: 819 <https://disscuscomputer.wordpress.com/2011/06/14/aplikasi-program>
- 8% wrds: 646 <https://eksis67234.blogspot.com/2014/11/pti-9.html>
- 8% wrds: 742 <http://fasilkom.mercubuana.ac.id/wp-content/uploads/2017/10/Pengolahan-Citra-TI...>

Show other Sources:]

Processed resources details:

133 - Ok / 14 - Failed

Show other Sources:]

Important notes:

Wikipedia:	Google Books:	Ghostwriting services:	Anti-cheating:
Wikipedia Detected!	[not detected]	[not detected]	[not detected]



UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
 Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
 website : www.pancabudi.ac.id email: unpad@pancabudi.ac.id
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
 Fakultas : SAINS & TEKNOLOGI
 Dosen Pembimbing I : Eko Hariyanto, S.Kom., M.Kom
 Dosen Pembimbing II : Hermansyah, S.kom., M.Kom
 Nama Mahasiswa : YUKE OCTAVIA IRAWAN
 Jurusan/Program Studi : Sistem Komputer
 Nomor Pokok Mahasiswa : 1514370753
 Bidang Pendidikan : SI
 Judul Tugas Akhir/Skripsi : Perancangan Aplikasi Deteksi Tepi Dengan Metode Sobel
 Dan Prewit Untuk Pengenalan pola Menggunakan
 Visual studio 2013

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
12-19	* Aa sempura	<i>[Signature]</i>	BAB I
13-19	* perbaiki rumusan masalah	<i>[Signature]</i>	
13-19	* Aa BAB I	<i>[Signature]</i>	
14-19	* tambah lean menu materi pada program	<i>[Signature]</i>	
16-19	* perbaiki fungsi dan program	<i>[Signature]</i>	
17-19	* Aa program	<i>[Signature]</i>	
17-19	* Aa selesai hasil	<i>[Signature]</i>	
11-19	* Aa sudah meja hijau	<i>[Signature]</i>	
19-19	* Aa jilid skripsi	<i>[Signature]</i>	

Medan, 01 Maret 2019
 Diketahui/Disetujui oleh
 Dekan,



Sri Shindi Indira, S.T., M.Sc.



UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI
 Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
 Fakultas : SAINS & TEKNOLOGI
 Dosen Pembimbing I : Eko Hariyanto, S.Kom M.Kom
 Dosen Pembimbing II : Hermansyah, S.Kom M.Kom
 Nama Mahasiswa : YUKE OCTAVIA IRAWAN
 Jurusan/Program Studi : Sistem Komputer
 Nomor Pokok Mahasiswa : 1514370753
 Bidang Pendidikan : SI
 Judul Tugas Akhir/Skripsi : Perancangan Aplikasi Deteksi Tepi Dengan Metode Sobel Dan Proritt Untuk Pengenalan Pola Menggunakan Visual studio 2013

TANGGAL	PEMBAHASAN MATERI	PARAF	KETERANGAN
2-19	> Aca Jemplo	[Signature]	
4-19	< Revisi Bab II tambah teori & pembuatan box	[Signature]	
10-19	> Buat referensi kutipan > 2013	[Signature]	
14-19	< Revisi Posisi Penomoran bab	[Signature]	
16-19	> Revisi Bab II, bagian referensi juga demo program	[Signature]	
18-19	< Revisi Bab III	[Signature]	
19-19	< Demo program	[Signature]	
19-19	< Revisi Bab IV	[Signature]	
10-19	< Revisi semua	[Signature]	
10-19	< Aca semua	[Signature]	

12-19 - Aca sidang menggunakan figure
 -19 - Aca judul

[Signature]

Medan, 01 Maret 2019
 Diketahui/Disetujui oleh
 Dekan,

Sri Shindi Indira, S.T., M.Sc.



YAYASAN PROF. DR. H. KADIRUN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambing Telp. 061-8455571
Medan - 20122

KARTU BEBAS PRAKTIKUM

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama : YUKE OCTAVIA IRAWAN
N.P.M. : 1514370753
Tingkat/Semester : Akhir
Fakultas : SAINS & TEKNOLOGI
Jurusan/Prodi : Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.



ABSTRAK

YUKE OCTAVIA IRAWAN
“PERANCANGAN APLIKASI DETEKSI TEPI DENGAN METODE
SOBEL DAN PREWITT UNTUK PENGENALAN POLA
MENGGUNAKAN VISUAL STUDIO 2013”
2019

Salah satu teknik pengolahan citra adalah deteksi tepi (*edge detection*). Deteksi tepi sangat penting dalam pengolahan citra karena pendeteksian tepi mencirikan batas-batas objek dan karena tepi sangat berguna untuk proses segmentasi dan identifikasi objek dalam citra. Tujuan operasi pendeteksi tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Dalam penggunaannya, deteksi tepi menggunakan operator berbasis turunan pertama dan turunan kedua. Beberapa metode yang termasuk dalam operator berbasis turunan pertama di antaranya adalah metode *Robert*, *Sobel* dan *Prewitt*. Sedangkan metode yang termasuk ke dalam operator berbasis turunan kedua di antaranya adalah *Laplacian of Gaussian (LOG)* dan *Canny*. Sedangkan metode yang akan dibahas pada penelitian ini adalah metode *sobel* dan *prewitt*. Penelitian ini mencoba membuat sebuah program aplikasi yang dapat mendeteksi tepi kemudian membandingkan antara kedua metode tersebut yang berfokus pada analisis perbandingan mana yang lebih baik digunakan dalam pendeteksian tepi citra.

Kata Kunci : *Sobel, Prewitt, Robert, Edge Detection*

DAFTAR ISI

	Halaman
KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL.....	vi
DAFTAR LAMPIRAN.....	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB II LANDASAN TEORI	5
2.1 Definisi Perancangan.....	5
2.2 Definisi Aplikasi	7
2.2.1 Sejarah dan Perkembangan Aplikasi	9
2.3 Deteksi Tepi	11
2.4 Definisi Citra	13
2.4.1 Pengolahan Citra	14
2.4.2 Citra Digital.....	15
2.5 Flowchart.....	17
2.6 UML (Unified Modeling Language).....	21
2.7 Microsoft Visual Studio	23
BAB III METODE PENELITIAN	33
3.1 Tahapan Penelitian	33
3.2 Metode Pengumpulan Data	34
3.3 Rancangan Penelitian	34
3.4 Perancangan Sistem.....	39
BAB IV HASIL DAN PEMBAHASAN.....	48
4.1 Kebutuhan Spesifikasi Minimum Hardware dan Software	48
4.2 Pengujian Aplikasi Program dan Pembahasan	48
BAB V PENUTUP.....	60
5.1 Kesimpulan.....	60

5.2	Saran.....	61
-----	------------	----

**DAFTAR PUSTAKA
BIOGRAFI PENULIS
LAMPIRAN-LAMPIRAN**

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Citra atau *image* merupakan salah satu komponen multimedia yang memiliki peranan penting yaitu sebagai suatu bentuk informasi visual. Meskipun begitu, dalam proses pengambilan citra, terkadang mengalami penurunan mutu, misalnya mengandung derau (*noise*), kurang tajam sebuah citra dan sebagainya. Citra seperti ini akan sangat sulit diinterpretasikan untuk pengolahan citra digital lebih lanjut, oleh sebab itu diperlukan sebuah program yang dapat melakukan pengolahan citra supaya citra yang mengalami penurunan mutu tersebut dapat diinterpretasikan.

Banyak cara yang dapat dilakukan untuk pengolahan sebuah citra di antaranya adalah penyaringan (*filtering*), penskalaan (*scalling*), *feature extraction*, *pattern recognition* dan juga segmentasi citra. Segmentasi citra memiliki beberapa manfaat, misalnya di bidang kedokteran sebagai identifikasi penyakit, di bidang pertanian dan di bidang telekomunikasi untuk pengenalan pola wajah.

Salah satu teknik pengolahan citra adalah deteksi tepi (*edge detection*). Deteksi tepi sangat penting dalam pengolahan citra karena pendeteksian tepi mencirikan batas-batas objek dan karena tepi sangat berguna untuk proses segmentasi dan identifikasi objek dalam citra. Tujuan operasi pendeteksi tepi

adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra.

Dalam penggunaannya, deteksi tepi menggunakan operator berbasis turunan pertama dan turunan kedua. Beberapa metode yang termasuk dalam operator berbasis turunan pertama di antaranya adalah metode *Robert*, *Sobel* dan *Prewitt*. Sedangkan metode yang termasuk ke dalam operator berbasis turunan kedua di antaranya adalah *Laplacian of Gaussian (LOG)* dan *Canny*. Sedangkan metode yang akan dibahas pada penelitian ini adalah metode *sobel* dan *prewitt*.

Penelitian ini mencoba membuat sebuah program aplikasi yang dapat mendeteksi tepi kemudian membandingkan antara kedua metode tersebut yang berfokus pada analisis perbandingan mana yang lebih baik digunakan dalam pendeteksian tepi citra.

Berdasarkan latar belakang tersebut penulis tertarik untuk membuat penelitian yang berjudul **“PERANCANGAN APLIKASI DETEKSI TEPI DENGAN METODE SOBEL DAN PREWITT UNTUK PENGENALAN POLA MENGGUNAKAN VISUAL STUDIO 2013”** agar dapat mengetahui hasil dari pendeteksian tepi tersebut.

1.2 Perumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada latar belakang masalah, maka dapat dirumuskan beberapa masalah yang lebih rinci yang kemungkinan akan muncul dalam penelitian ini, yaitu :

1. Bagaimana membuat sebuah program aplikasi yang dapat mengolah citra digital ?
2. Bagaimana proses pendeteksian tepi menggunakan metode *sobel* dan *prewitt* didalam sebuah aplikasi ?
3. Bagaimana perbandingan kinerja antara metode *sobel* dan *prewitt* dalam proses segmentasi batas tepi citra?

1.3 Batasan Masalah

Dalam penyusunan skripsi ini, penulis membatasi pembahasannya hanya pada beberapa poin seperti:

1. Citra yang digunakan berformat *Joint Photographic Expert's Group* (.jpeg) dengan gabungan warna RGB.
2. Penggunaan metode yang digunakan hanya metode *sobel* dan *prewitt*.
3. Perancangan dan pengimplementasian program aplikasi dibuat dengan menggunakan program *visual studio* versi 2013.
4. Penelitian ini tidak membahas proses pengambilan citra.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari perancangan dan pembuatan aplikasi pendeteksi tepi ini adalah sebagai berikut :

1. Menghasilkan sebuah program aplikasi yang dapat digunakan dalam pengolahan citra digital.

2. Program aplikasi dapat menerapkan metode *sobel* dan *prewitt* untuk pengolahan citra.
3. Merealisasikan program aplikasi pendeteksian pola menggunakan metode *sobel* dan *prewitt* yang dibuat menggunakan program visual studio 2013.
4. Dan untuk mengetahui perbandingan metode *sobel* dengan *prewitt* dalam proses segmentasi citra untuk mengenali sebuah pola.

1.5 Manfaat Penelitian

Adapun manfaat yang diharapkan dari perancangan dan pembuatan aplikasi pendeteksi tepi ini adalah sebagai berikut :

1. Agar dapat mempermudah dalam mendeteksi tepi objek pada citra yang ingin di teliti.
2. Meningkatkan kualitas visual atau menonjolkan beberapa aspek informasi yang terkandung di dalam citra digital.
3. Memberikan informasi mengenai pendeteksian batas tepi suatu citra sehingga dapat dijadikan acuan atau referensi bagi akademis dan mahasiswa.
4. Dan agar dapat mengetahui perbandingan metode terbaik antara metode *sobel* dan *prewitt* untuk segmentasi citra digital.

BAB II

LANDASAN TEORI

2.1 Definisi Perancangan

Langkah awal dalam membuat sebuah sistem adalah perancangan dari sistem tersebut. Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem.

Perancangan adalah proses pengembangan spesifikasi baru berdasarkan rekomendasi hasil analisis sistem. perancangan dapat diartikan juga sebagai suatu kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan *alternative* sistem yang baik. Perancangan mempunyai 2 maksud, yaitu untuk memenuhi kebutuhan kepada pemakai sistem dan untuk memberikan gambaran yang jelas kepada pemogram komputer dan ahli-ahli teknik lainnya yang terlibat. Perancangan sistem juga mempunyai bagian-bagiannya, yaitu sebagai berikut :

a. Perancangan Database

Merupakan sejumlah kumpulan-kumpulan sebuah data yang sudah tersimpan didalam media penyimpanan sekunder yang dipakai untuk menyimpan

data-data panjang yang di gunakan sebagai input-an sistem. Kemudian data akan diolah menjadi data output atau keluaran sistem.

b. Perancangan Proses

Merupakan penjelasan suatu proses bekerjanya sistem untuk melakukan suatu pengolahan data input menjadi data output menggunakan fungsi yang sudah direncanakan.

c. Perancangan *interface*

Merupakan bagian dari software yang bisa digunakan oleh enduser yang bisa di lihat pada layar monitor apabila sebuah program dijalankan.

Perancangan juga merupakan proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya (Soetam Rizky, 2011. Dalam Mahyuni, Sharipuddin dan Martono, 2014).

Berdasarkan definisi perancangan di atas, maka penulis dapat menyimpulkan bahwa perancangan merupakan tahap lanjut dari analisis sistem untuk membuat sebuah rancangan sistem baru yang lebih baik dari sebelumnya.

Beberapa tahap perancangan atau desain sistem mempunyai dua tujuan utama yaitu :

1. Memenuhi kebutuhan pemakaian sistem (*User*).
2. Memberikan gambaran yang jelas dan menghasilkan rancangan bangun yang lengkap kepada pemrograman komputer dan ahli-ahli teknik lainnya yang terlibat dalam pengembangan atau pembuatan sistem yang secara rinci.

2.2 Definisi Aplikasi

Aplikasi merupakan program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *MS-Word*, *Ms-Excel*. Aplikasi berasal dari kata *application*, aplikasi merupakan suatu subteks perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan satu tugas yang diinginkan pengguna. Penggunaan aplikasi dalam suatu komputer merupakan instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

Menurut Kamus Besar Bahasa Indonesia (KBBI), aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang

menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata dan pemutar media. Adapun beberapa jenis aplikasi, yang dikelompokkan menjadi beberapa macam, yaitu sebagai berikut :

- a. *System Software*, yaitu aplikasi yang dapat mengelola dan mengendalikan proses operasi internal didalam sistem komputer.
- b. *Real Time Software*, yaitu aplikasi yang memiliki fungsi untuk mengamati, menganalisa, serta mengendalikan suatu kejadian di dunia nyata saat sedang berlangsung.
- c. *Business Software*, yaitu aplikasi yang diciptakan dan dikembangkan untuk keperluan bisnis, misalnya untuk mengatur sistem keuangan.
- d. *Engineering and Scientific Software*, yaitu aplikasi yang diciptakan dan dikembangkan untuk membantu manusia menyelesaikan suatu masalah yang sifatnya non algoritmik.
- e. *Web Based Software*, yaitu aplikasi yang memiliki fungsi sebagai media penghubung antara pengguna dengan internet secara langsung.
- f. *Personal Computer Software*, yaitu aplikasi yang digunakan untuk perangkat pengguna resmi dan juga pribadi yang sudah banyak digunakan saat ini.

Jadi aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan.

2.2.1 Sejarah dan Perkembangan Aplikasi

Sebelum aplikasi pertama di dunia ditemukan, ada sejarah yang sangat panjang dalam proses ditemukannya aplikasi. Jauh sebelum aplikasi yang ada pada komputer dapat berjalan secanggih saat ini, semua perangkat lunak (*software*) selalu mengandalkan Aljabar Boolean. Aljabar ini menggunakan kode binary digit (bit) yang terdiri dari dua angka yaitu 1 (benar/*on*) dan 0 (salah/*off*). Rumitnya (terlalu panjang) penggunaan kode binary digit ini membuat orang-orang pada saat itu mulai membuat kelompok-kelompok bit yang terdiri dari *nibble* (4 bit), *byte* (8 bit), *word* (2 byte), dan *doubeIword* (32 bit).

Hadirnya kelompok-kelompok bit ini terbilang cukup membantu dalam berbagai kegiatan yang dilakukan di perangkat lunak komputer. Dengan berbagai kelompok bit yang ada pada saat itu, orang-orang mulai merakit kode-kode tersebut menjadi berbagai struktur instruksi seperti transfer, operasi logika, dan penyimpanan, hingga pada akhirnya terbentuklah kode-kode baru yang kita kenal dengan nama *assembler*. Kode-kode *assembler* ini lah yang nantinya menjadi cikal-bakal dibuatnya berbagai macam aplikasi yang saat ini dapat mempermudah berbagai kegiatan manusia.

Seiring dengan perkembangan dan kemajuan zaman serta berkembangnya peradaban manusia, aplikasi komputer secara terus-menerus mengalami peningkatan dalam perkembangannya. Berdasarkan perkembangan aplikasi komputer itu sendiri, aplikasi komputer dibagi dalam beberapa era antara lain :

a. Era Pioneer

Pada era ini, bentuk aplikasi komputer pada awalnya adalah sambungan-sambungan kabel ke antar bagian dalam komputer. Cara dalam mengakses komputer adalah menggunakan *punched card* yaitu kartu yang di lubangi. Penggunaan komputer saat itu masih dilakukan secara langsung, sebuah program digunakan untuk sebuah mesin tertentu dan untuk tujuan tertentu. Pada era ini, aplikasi komputer atau software komputer masih merupakan satu-kesatuan dengan hardware komputer.

b. Era Stabil

Pada era ini, aplikasi komputer sudah mengalami kemajuan yang cukup pesat. Baris-baris perintah aplikasi komputer yang dijalankan oleh komputer bukan lagi satu-satu, tapi sudah dapat melakukan banyak proses secara bersamaan atau *multi tasking*. Aplikasi komputer pada era ini juga mampu menyelesaikan banyak pengguna (*multi user*) dan secara cepat/langsung (*real time*). Pada era ini jugalah mulai di kenal sistem basis data yang mampu memisahkan antara program dan data.

c. Era Mikro

Pada era mikro ini, aplikasi komputer dapat dibedakan menjadi beberapa bagian yaitu aplikasi sistem (Windows, Linux, Machintos,dll), *software* aplikasi (Ms.Office, Open Office, dll) dan *Languange Software* / Bahasa Pemograman (Assembler, Visual Basic, Delphi, dll)

d. Era Modern

Era sekarang termasuk dalam era modern yang di dalamnya berkembang pesat aplikasi komputer yang juga bisa diaplikasikan ke dalam perangkat lain seperti Android, Symbian, dll. Kemampuan yang ada pada aplikasi juga semakin meningkat, selain menangani masalah teknis, aplikasi juga dapat mengenal suara dan gambar.

2.3 Deteksi Tepi (*Edge Detection*)

Deteksi tepi banyak dipakai untuk mengidentifikasi suatu objek dalam sebuah gambar. Tujuan dari deteksi tepi adalah untuk menandai bagian yang menjadi detail citra dan memperbaiki detail citra yang kabur karena adanya kerusakan atau efek akuisisi data. Dalam citra, sebagian besar informasi terletak pada batas antara dua daerah yang berbeda.

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari objek-objek gambar. Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga. Macam-macam metode untuk proses deteksi tepi ini, antara lain sebagai berikut (Achmad Basuki, 2005. Dalam Apriyana dan Delta Sri Maharani, 2013).

a. Metode Robert

Metode Robert adalah nama lain dari teknik differensial yang dikembangkan di atas, yaitu differensial pada arah horizontal dan differensial pada

arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Kernel *filter* yang digunakan dalam metode Robert ini adalah :

$$H = [-1 \ 1] \text{ dan } H = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

b. Metode Prewitt

Metode Prewitt merupakan pengembangan metode Robert dengan menggunakan *filter* HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF (*High Pass Filter*). Kernel *filter* yang digunakan dalam metode Prewitt ini adalah :

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan}$$

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

c. Operator Sobel

Metode Sobel merupakan pengembangan metode Robert dengan menggunakan *filter* HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Kelebihan dari metode Sobel ini adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi. Kernel *filter* yang digunakan dalam metode Sobel ini adalah :

$$H = \begin{bmatrix} -1 & -0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan}$$

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

2.4 Definisi Citra

Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Melalui sebuah citra kita dapat melihat informasi dari sebuah objek. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (*a picture is more than a thousand words*) (Putra, Ardiansyah, 2007. Dalam Asmardi Zalukhu, 2016).

Sebuah citra dapat didefinisikan sebagai sebuah fungsi dua dimensi, $f(x, y)$, dimana x dan y adalah koordinat bidang (spasial), dan amplitude dari tiap pasangan koordinat (x, y) dinamakan intensitas citra pada titik tersebut.

Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpan. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinu seperti gambar

pada monitor televisi, foto sinar X, hasil CT Scan dll. Sedangkan pada citra digital adalah citra yang dapat diolah oleh computer.

Secara harfiah, Citra (*image*) adalah pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya dan bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik misalnya mata manusia, kamera, pemindai (*scanner*) dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam. Citra sebagian keluaran dari suatu sistem perekaman data dapat bersifat :

- a. Optik berupa foto.
- b. Analog berupa sinyal video.
- c. Digital yang langsung disimpan pada suatu pita magnetik.

2.4.1 Pengolahan Citra

Pengolahan citra (*image procesing*) merupakan suatu sistem di mana proses dilakukan dengan memasukan berupa citra dan hasilnya juga berupa citra. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan perkembangannya dunia komputasi yang ditansi dengan semakin meningkatnya dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra, maka *image processing* tidak dapat dilepaskan dengan bidang computer vision. Umumnya operasi-operasi pada pengolahan citra diterapkan pada suatu citra bila :

1. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau menonjolkan beberapa aspek informasi yang terkandung di dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan atau diukur.
3. Sebagian citra perlu digabung dengan bagian citra yang lainnya.
4. Sebagian citra perlu digabung dengan bagian citra lain.

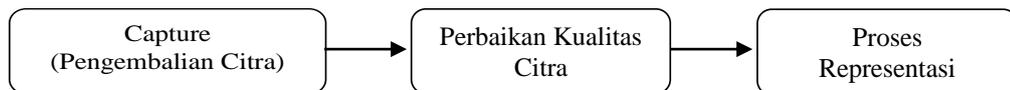
Pengolahan citra digital mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*featureimages*) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. Input dari pengolahan citra adalah citra, sedangkan outputnya adalah citra hasil pengolahan.

2.4.2 Citra Digital

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantitas. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain sampling pada citra menyatakan besar kecilnya ukuran pixel (titik) pada citra, dan kuantitatif menyatakan besar nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan bit yang digunakan oleh mesin dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra. Kualitas citra juga sangat dipengaruhi oleh tingkat keberadaan derau.

Setiap citra yang didapatkan secara optik, elektro-optik atau elektronik sangat dipengaruhi oleh alat penginderaan. Penurunan kualitas citra bisa terjadi karena derau sensor, kamera yang kurang fokus, dan lain-lain.

Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, maksudnya adalah citra dan keluarannya juga citra, namun citra keluarannya mempunyai kualitas lebih baik dari pada citra masukan. Proses pengolahan citra secara diagram, proses dimulai dari pengambilan citra, perbaikan citra, sampai dengan menyatakan representasi citra yang dicitrakan seperti pada gambar 2.1 berikut :



Gambar 2.1 Diagram Proses Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang lebih baik. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung didalam citra.

1. Elemen didalam citra perlu digabung dengan bagian citra yang lainnya.
2. Sebagian citra perlu digabung dengan bagian citra yang lainnya.

2.5 Flowchart

Flowchart merupakan bagian alir program (*Flowchart*) adalah bagian yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir” (Jogianto Hartono, 2001. Dalam Drs. Katen Lumbanbatu, M.Kom., Novriyeni, S.Kom., M.Kom. 2013).

Diagram alir (bahasa Inggris : *flowchart*) adalah sebuah jenis diagram yang mewakili algoritme, alir kerja atau proses, yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis, dan urutannya dihubungkan dengan panah. Diagram ini mewakili ilustrasi atau penggambaran penyelesaian masalah. Diagram alir digunakan untuk menganalisis, mendesain, mendokumentasi atau memajemen sebuah proses atau program di berbagai bidang.

Diagram alir digunakan untuk mendesain dan mendokumentasi proses atau program sederhana. Seperti jenis diagram lainnya, diagram ini membantu menggambarkan apa yang sedang terjadi dan dengan demikian membantu mengerti sebuah proses. Dan mungkin saja menentukan kekurangan fitur, atau bagian yang kurang jelas didalam sebuah proses. Inilah fungsi dari flowchart, sebagai berikut :

1. Memastikan program memiliki alurnya sendiri.
2. Melihat keseluruhan program.
3. Melihat proses dari sebuah program ketika dijalankan.
4. Pedoman dalam menyusun atau mengembangkan aplikasi.

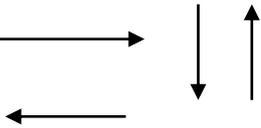
Terdapat beberapa bentuk diagram alir, dan setiap bentuk memiliki urutan dan peranan masing-masing. Dua bentuk persegi yang paling umum

digunakan dalam diagram alir, yaitu Langkah pemrosesan, yang sering disebut dengan *tindakan*, dan dinotasikan sebagai persegi panjang. Keputusan, biasanya dinotasikan sebagai belah ketupat. Adapun simbol-simbol flowchart sebagai berikut :

a. Flow Direction Symbols

Yaitu simbol yang dipakai untuk menghubungkan antara simbol yang satu dengan simbol lainnya atau disebut juga connecting line.

Tabel 2.1 Flow Direction Symbols

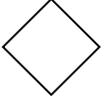
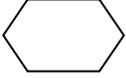
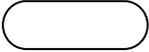
	Arus / Flow	Penghubung antara prosedur / proses
	Connector	Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang sama
	Off-line Connector	Simbol keluar / masuk prosedur atau proses dalam lembar / halaman yang lain

Sumber : Al-Bahra Bin Ladjamuddin (2013:266), dalam Jurnal Politeknik Negeri

b. Processing Symbols

Merupakan simbol yang menunjukkan jenis operasi pengolahan dalam suatu prosedur.

Tabel 2.2 Processing Symbols

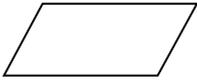
	Process	Simbol yang menunjukkan pengolahan yang dilakukan Komputer
	Decision	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban / aksi
	Predefined Process	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage
	Terminal	Simbol untuk permulaan atau akhir dari suatu program
	Manual Input	Simbol untuk pemasukan data secara manual on-line keyboard

Sumber : Al-Bahra Bin Ladjamuddin (2013:267), dalam Jurnal Politeknik Negeri Sriwijaya

c. Input Output Symbols

Simbol yang dipakai untuk menyatakan jenis peralatan yang digunakan sebagai media input atau output.

Tabel 2.3 Input Output Symbols

	Input-Output	Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
	Document	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output di cetak dikertas
	Disk and On-line Storage	Simbol untuk menyatakan input berasal dari disk atau output di simpan ke disk

Sumber : Al-Bahra Bin Ladjamuddin (2013:267), dalam Jurnal Politeknik Negeri

Sriwijaya

2.6 UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Windu dan Grace, 2013. Dalam Suendri, 2018).

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

UML menyediakan macam-macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

- a. *Use case*

Menggambarakan *external view* dari sistem yang akan kita buat modelnya. Model *use case* dapat dijabarkan dalam diagram *use case*, tetapi perlu diingat, diagram tidak indetik dengan model karena model lebih luas dari diagram. *Use case* harus mampu menggambarkan urutan aktor yang menghasilkan nilai terukur.

b. *Class Diagram*

Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek. Class memiliki tiga area pokok yaitu :

1. Nama, kelas harus mempunyai sebuah nama.
2. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
3. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

c. *Activity Diagram*

Diagram *activity* menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity* diagram juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan. “Diagram *activity* adalah aktifitas-aktifitas, objek, state, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

d. *Sequence Diagram*

Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

Selain menyediakan berbagai macam diagram untuk memodelkan aplikasi, UML juga mempunyai tujuan atau fungsi dalam penggunaannya, yaitu sebagai berikut :

1. Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai blueprint, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
6. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

2.7 Microsoft Visual Studio

Microsoft Visual Studio adalah sebuah *Integrated Development Environment* buatan *Microsoft Corporation*. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan diatas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Yang terdapat pada tampilan *Visual Studio* antara lain sebagai berikut :

a. Antar Muka *Visual Studio*

Interface atau antar muka *Visual Studio*, berisi menu, *toolbar*, *toolbox*, *form*, *project explorer* dan properti. Pembuatan program aplikasi menggunakan *Visual Studio* dilakukan dengan membuat tampilan aplikasi pada *form*, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di (*toolbox*), dan setiap komponen yang dipakai harus diatur propertinya lewat jendela (*property*). Menu pada dasarnya adalah operasional standar di dalam sistem operasi *windows*, seperti membuat form baru, membuat project baru, membuka project dan menyimpan *project*. Di samping itu terdapat fasilitas-fasilitas pemakaian *Visual Studio* pada menu. Untuk lebih jelasnya *Visual Studio* menyediakan bantuan yang sangat lengkap dan detail dalam MSDN (*Microsoft Developer Network*).

1. *Toolbox*

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu *project* aktif, artinya isi komponen dalam *toolbox* sangat tergantung pada jenis *project* yang dibangun. *Toolbox Visual Studio* dengan semua kontrol intrinsik. Jendela *Toolbox* merupakan jendela yang sangat penting. Dari jendela ini dapat mengambil komponen-komponen (*object*) yang akan ditanamkan pada form untuk membentuk *user interface*.

2. Variabel

Variabel adalah tempat dalam memori komputer yang diberi nama (sebagai pengenal) dan dialokasikan untuk menampung data. Sesuai data

yang ditampung maka variabel harus mempunyai tipe data yang sesuai dengan isinya.

3. Operator

Operator digunakan untuk menghubungkan variabel dengan variabel lain untuk melakukan berbagai manipulasi dan pengolahan data.

b. Konsep Dasar Pemrograman Dalam *Visual Studio*

Konsep dasar pemrograman *Visual Studio* adalah pembuatan form dengan mengikuti aturan pemrograman *Property*, Metode dan *Event*.

1. *Property*

Setiap komponen di dalam pemrograman *Visual Studio* dapat diatur propertinya sesuai dengan kebutuhan aplikasi.

2. Metode

Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode merupakan tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu program aplikasi.

3. *Event*

Setiap komponen dapat beraksi melalui *event*, seperti *event click* pada *commandbutton* yang tertulis dalam layar *script* `Command1_Click`.

Visual Studio pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Studio* (yang sering juga disebut dengan VB. Net selain

disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program berbasis *windows*. Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). *Visual Basic* mencakup sebuah kode editor yang didukung oleh fitur *intellisense* atau yang disebut dengan *code refactoring*. *Debugger* telah terintegrasi bekerja pada *level source level debugger* dan *level debugger mesin*. *Toll built in* mencakup *form desainer* untuk membangun sebuah aplikasi *GUI*, *web desainer*, *class desainer* dan *database schema desainer*.

Microsoft Visual Studio didukung bahasa pemrograman yang berbeda. Adapun bahasa pemrograman yang didukung oleh *Visual Basic Studio* adalah bahasa pemrograman *C++*, *Visual Basic*, *Visual C#*. *Visual Studio* juga dapat mendukung bahasa pemrograman lain seperti *M*, *python* dan *ruby* yang semuanya itu terdapat pada *pack extra* yang terpisah dari *visual studio* (Edy Winarno ST, M.Eng, Ali Zaki, SmitDev Community 2010, dalam buku “*Web Programming dengan Visual Basic 2010*”. Dalam Nancy Extise Putri, Supriandi Azpar, 2017).

Dibawah ini adalah beberapa versi dari *Microsoft Visual Studio* yaitu sebagai berikut :

a. *Microsoft Visual Studio 97*

Microsoft pertama kali merilis *Visual Studio* (nama kode Boston, untuk kota dengan nama yang sama, sehingga memulai nama kode VS yang terkait dengan tempat-tempat) pada tahun 1997, menggabungkan banyak alat pemrogramannya bersama-sama untuk pertama kalinya.

b. *Visual Studio .NET 2003*

Visual Studio .NET 2003 dikirimkan dalam lima edisi: akademik, standar, profesional, pengembang perusahaan, dan arsitek perusahaan. *Visual Studio .NET 2003 Enterprise Architect* edisi mencakup implementasi teknologi pemodelan *Microsoft Visio 2002*, termasuk alat untuk membuat representasi visual berbasis Bahasa Unified Modeling arsitektur aplikasi, dan pemodelan objek-peran (ORM) dan solusi pemodelan basis data logis. "Templat Perusahaan" juga diperkenalkan, untuk membantu tim pengembangan yang lebih besar melakukan standarisasi gaya pengkodean dan menegakkan kebijakan seputar penggunaan komponen dan pengaturan properti.

c. *Visual Studio 2005*

Visual Studio 2005, dengan nama sandi Whidbey (referensi ke Whidbey Island di wilayah Puget Sound), Dirilis secara online pada Oktober 2005 dan ke toko ritel beberapa minggu kemudian. Microsoft menghapus ".NET" moniker dari *Visual Studio 2005* (dan juga setiap produk lain dengan .NET dalam namanya), tetapi masih terutama menargetkan *.NET Framework*, yang ditingkatkan ke versi 2.0. Ini adalah versi terakhir yang tersedia untuk *Windows 2000* dan juga versi terakhir untuk dapat menargetkan *Windows 98*, *Windows Me* dan *Windows NET 4.0* untuk aplikasi C++. Nomor versi internal *Visual Studio 2005* adalah 8.0 sedangkan versi format file adalah 9.0. *Microsoft* merilis Paket Layanan 1 untuk *Visual Studio 2005* pada 14 Desember 2006. Pembaruan tambahan untuk Paket

Layanan 1 yang menawarkan kompatibilitas *Windows Vista* dibuat tersedia pada 3 Juni 2007.

d. *Visual Studio* 2008

Visual Studio 2008, dan *Visual Studio Team System* 2008 nama kode Orcas (referensi ke Pulau Orcas, juga sebuah pulau di wilayah Puget Sound, seperti Whidbey untuk rilis 2005 sebelumnya), dirilis ke pelanggan MSDN pada 19 November 2007 bersama *.NET Framework* 3.5. Kode sumber untuk *Visual Studio* 2008 IDE tersedia di bawah lisensi sumber bersama untuk beberapa mitra dan ISV Microsoft. Microsoft merilis Paket Layanan 1 untuk *Visual Studio* 2008 pada 11 Agustus 2008. Nomor versi internal *Visual Studio* 2008 adalah versi 9.0 sedangkan versi format file adalah 10.0. *Visual Studio* 2008 adalah versi terakhir untuk mendukung penargetan *Windows* 2000 untuk aplikasi C++.

e. *Visual Studio* 2010

Pada 12 April 2010, Microsoft merilis *Visual Studio* 2010, dengan nama kode Dev10, dan *.NET Framework* 4. IDE *Visual Studio* 2010 dirancang ulang yang, menurut Microsoft, membersihkan organisasi UI dan "mengurangi kekacauan dan kerumitan." IDE baru lebih baik mendukung beberapa jendela dokumen dan jendela alat apung, sambil menawarkan multi-monitor yang lebih baik mendukung. Shell IDE telah ditulis ulang menggunakan *Windows Presentation Foundation* (WPF), sedangkan bagian dalam telah dirancang ulang menggunakan *Managed Extensibility Framework* (MEF) yang menawarkan lebih banyak poin ekstensibilitas daripada versi sebelumnya dari IDE yang

memungkinkan add-in untuk memodifikasi perilaku pengguna. IDE. Multi-paradigma ML-varian F# yang baru merupakan bagian dari *Visual Studio 2010*.

f. *Visual Studio Ultimate 2010*

Visual Studio Ultimate 2010 menggantikan *Visual Studio 2008 Team Suite*. Ini termasuk alat pemodelan baru, seperti *Explorer Arsitektur*, yang secara grafis menampilkan proyek dan kelas dan hubungan di antara mereka. Ini mendukung diagram aktivitas UML, diagram komponen, diagram kelas (logis), diagram urutan, dan diagram use case. *Visual Studio Ultimate 2010* juga mencakup Analisis Dampak Uji yang memberikan petunjuk tentang kasus uji mana yang dipengaruhi oleh modifikasi kode sumber, tanpa benar-benar menjalankan kasus uji. Ini mempercepat pengujian dengan menghindari menjalankan kasus pengujian yang tidak perlu.

g. *Visual Studio 2012*

Pembangunan terakhir *Visual Studio 2012* diumumkan pada 1 Agustus 2012 dan acara peluncuran resmi diadakan pada 12 September 2012. Tidak seperti versi sebelumnya, *Visual Studio 2012* tidak dapat merekam dan memutar makro dan editor makro telah dihapus. Fitur-fitur baru termasuk dukungan untuk WinRT dan C++ / CX (Extensi Komponen) dan C++ AMP (pemrograman GPGPU) Pewarnaan Semantik. Tampilan IDE dari *Visual Studio 2012* dengan versi sebelumnya, yaitu versi *Visual Studio 2010* cukup berbeda berbeda apalagi jika dilihat dari tampilan *skin* luarnya. Tampilan *Visual Studio 2012* mengaplikasikan gaya metro, dimana tampilan bergaya metro merupakan tampilan dasar dari sistem

operasi Microsoft terbaru yaitu Windows 8. Pada Visual Studio 2012 terdapat beberapa seksi workspace yang harus diketahui sebelum berkerja dengan *Visual Studio 2012*.

h. *Visual Studio 2013*

Pratinjau untuk *Visual Studio 2013* diumumkan pada konferensi Build 2013 dan tersedia pada 26 Juni 2013. *Visual Studio 2013 RC (Release Candidate)* dibuat tersedia untuk pengembang di MSDN pada 9 September 2013. Rilis final *Visual Studio 2013* tersedia untuk diunduh pada 17 Oktober 2013 bersama dengan .NET 4.5.1. *Visual Studio 2013* secara resmi diluncurkan pada 13 November 2013 di acara peluncuran virtual dengan penekanan utama S. Somasegar dan di-host di events.visualstudio.com. "*Visual Studio 2013 Pembaruan 1*" (*Visual Studio 2013.1*) dirilis pada 20 Januari 2014. *Visual Studio 2013.1* adalah pembaruan yang ditargetkan yang membahas beberapa bidang utama dari umpan balik pelanggan. "*Visual Studio 2013 Pembaruan 2*" (*Visual Studio 2013.2*) dirilis pada 12 Mei 2014. *Visual Studio 2013 Pembaruan 3* dirilis pada 4 Agustus 2014. Dengan pembaruan ini, *Visual Studio* menyediakan opsi untuk menonaktifkan menu all-caps, yang diperkenalkan di VS2012. "*Visual Studio 2013 Pembaruan 4*" (*Visual Studio 2013.4*) dirilis pada 12 November 2014. "*Visual Studio 2013 Pembaruan 5*" (*Visual Studio 2013.5*) dirilis pada 20 Juli 2015.

Visual Studio 2013 memperkenalkan pengalaman IDE terhubung, memungkinkan untuk memulai *Visual Studio 2013* lebih cepat. Ketika memulai *Visual Studio 2013* untuk pertama kalinya, akan diminta untuk *sign in*

menggunakan akun *Microsoft*. Setelah masuk, salinan *Visual Studio 2013* akan didaftarkan untuk digunakan tanpa perlu tanpa perlu langkah-langkah tambahan. *Visual Studio 2013* juga akan secara otomatis melakukan sinkronisasi pengaturan di perangkat memanfaatkan pengalaman IDE terhubung. Pengaturan *Synchronized* meliputi kategori seperti cara pintas *keyboard*, penampilan *Visual Studio* (tema, font, dll) dan banyak lagi, semua sambil menjaga dalam kendali penuh melalui pilihan *Synchronized* pengaturan.

Selain terbagi menjadi berbagai versi *Microsoft Visual Studio* juga dibagi menjadi berbagai edisi, yaitu sebagai berikut :

a. Komunitas

Ini adalah versi gratis yang diumumkan pada 2014. Semua edisi lainnya dibayar. Ini berisi fitur yang mirip dengan edisi Profesional. Menggunakan edisi ini, setiap pengembang dapat mengembangkan aplikasi gratis atau berbayar mereka sendiri seperti .Net aplikasi, aplikasi Web dan banyak lagi. Dalam organisasi perusahaan, edisi ini memiliki beberapa keterbatasan. Misalnya, jika organisasi Anda memiliki lebih dari 250 PC dan memiliki pendapatan tahunan lebih dari \$ 1 Juta (Dolar AS) maka Anda tidak diizinkan menggunakan edisi ini. Dalam organisasi non-perusahaan, hingga lima pengguna dapat menggunakan edisi ini. Tujuan utamanya adalah untuk menyediakan dukungan Ekosistem (Akses ke ribuan ekstensi) dan Bahasa (Anda dapat memberi kode dalam C #, VB, F #, C ++, HTML ++, JavaScript, Python, dll).

b. Profesional

Ini adalah edisi komersial Visual Studio. Muncul dalam Visual Studio 2010 dan versi yang lebih baru. Ini memberikan dukungan untuk mengedit XML dan XSLT dan termasuk alat seperti Server Explorer dan integrasi dengan Microsoft SQL Server. Microsoft menyediakan uji coba gratis untuk edisi ini dan setelah periode uji coba, pengguna harus membayar untuk terus menggunakannya. Tujuan utamanya adalah untuk menyediakan Fleksibilitas (alat pengembang profesional untuk membangun semua jenis aplikasi), Produktivitas (Fitur hebat seperti CodeLens meningkatkan produktivitas tim Anda), Kolaborasi (alat perencanaan proyek Agile, grafik, dll.) Dan manfaat Pelanggan seperti perangkat lunak Microsoft, plus Azure, Pluralsight, dll.

c. Perusahaan

Ini adalah solusi ujung ke ujung yang terintegrasi untuk tim dengan ukuran berapa pun dengan kualitas dan kebutuhan skala yang menuntut. Microsoft menyediakan uji coba gratis 90 hari untuk edisi ini dan setelah periode uji coba, pengguna harus membayar untuk terus menggunakannya. Manfaat utama dari edisi ini adalah sangat dapat diskalakan dan memberikan perangkat lunak berkualitas tinggi.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

3.2 Metode Pengumpulan Data

Untuk pengumpulan data, penelitian ini menggunakan metode Penelitian Kepustakaan (*Library research*) yaitu penelitian yang dilaksanakan dengan menggunakan literatur (kepuustakaan), baik berupa buku, catatan, maupun laporan hasil penelitian dari penelitian yang terdahulu. Mengumpulkan data melalui berbagai referensi yang relevan tanpa berhubungan secara langsung dengan tempat atau objek penelitian. Dan dengan menggunakan referensi-referensi yang relevan dengan penelitian.

3.3 Rancangan Penelitian

Adapun rancangan penelitian meliputi :

a. Penerapan Metode

Metode citra yang digunakan pada penelitian ini adalah metode *sobel* dan *prewitt*.

1. Metode *Prewitt*

Metode *prewitt* merupakan pengembangan metode *Robert* dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF (*High Pass Filter*). Kernel filter yang digunakan dalam metode *Prewitt* ini adalah :

Prewitt Horizontal			Prewitt Vertikal		
-1	0	1	1	1	1
-1	0	1	0	0	0
-1	0	1	-1	-1	-1

Gambar 3.2 Kernel Operator prewitt

2. Metode *Sobel*

Metode *sobel* merupakan pengembangan metode *robert* dengan menggunakan filter HPF (*High Pass Filter*) yang diberi satu angka nol penyangga. Kelebihan dari metode Sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi. Kernel filter yang digunakan dalam metode *sobel* ini adalah :

Sobel Horizontal			Sobel Vertikal		
-1	0	1	1	2	1
-2	0	2	0	0	0
-1	0	1	-1	-2	-1

Gambar 3.3 Kernel Operator Sobel

Operator *sobel* terdiri dari matrix 3x3 masing – masing adalah G_x dan G_y , matriks *mask* tersebut dirancang untuk memberikan respon secara maksimal terhadap tepi objek baik horizontal maupun vertikal. *Mask* dapat diaplikasikan secara terpisah terhadap *input* citra. Operator Sobel menggunakan kernel operator gradien 3 x 3, dengan koefisien yang telah ditentukan G_x dan G_y dapat dinyatakan sebagai berikut :

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

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

Kernel diatas dirancang untuk menyelesaikan permasalahan deteksi tepi baik secara vertikal maupun horizontal. Penggunaan kernel – kernel ni dapat digunakan bersamaan ataupun secara terpisah. Untuk mendapatkan nilai maksimum dari operator sobel, proses selanjutnya adalah dengan menghitung kekuatan tepi citra terhadap warna kecerahannya dengan cara mencari nilai *Magnitude* yang dapat dihitung dengan persamaan berikut :

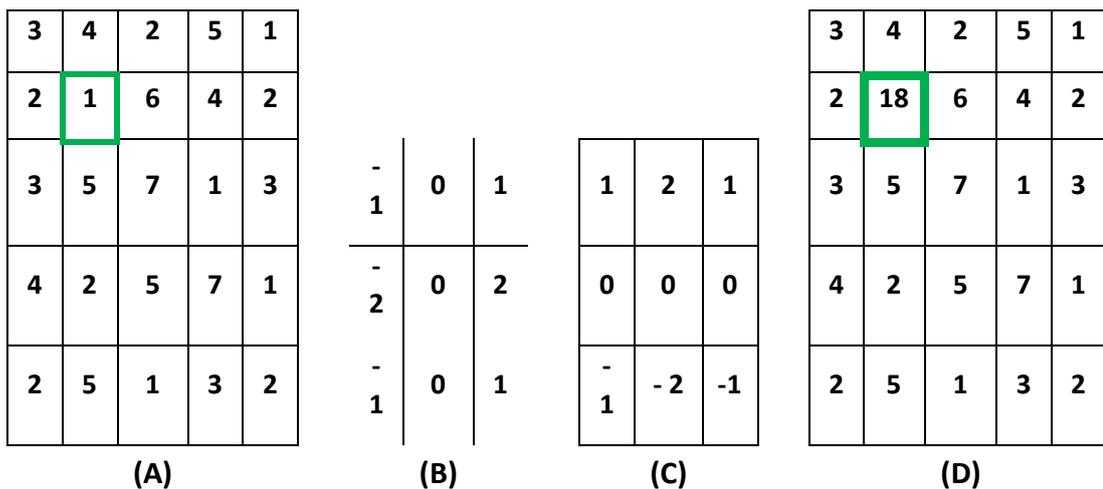
$$M = \sqrt{G_x^2 + G_y^2}$$

Karena menghitung akar adalah persoalan rumit dan menghasilkan nilai real, maka dalam mencari kekuatan tepi (*magnitude*) dapat disederhanakan perhitungannya.

Besarnya *magnitude* gradient dapat dihitung lebih cepat lagi dengan menggunakan persamaan berikut :

$$M = |G_x| + |G_y|$$

Pada formula diatas diperlihatkan deteksi tepi dengan operator Sobel. Operasi konvulsi bekerja dengan menggeser kernel piksel per piksel, yang hasilnya kemudian disimpan dalam matriks baru. Konvulsi pertama dilakukan terhadap piksel yang bernilai 1 (di titik pusat *mask*)



Gambar 3.4 (A) citra asli, (B) G_x , (C) G_y , (D) Hasil Konvulsi

Nilai 18 pada citra hasil konvulsi diperoleh dengan perhitungan berikut :

$$G_x = (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11$$

$$G_y = (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7$$

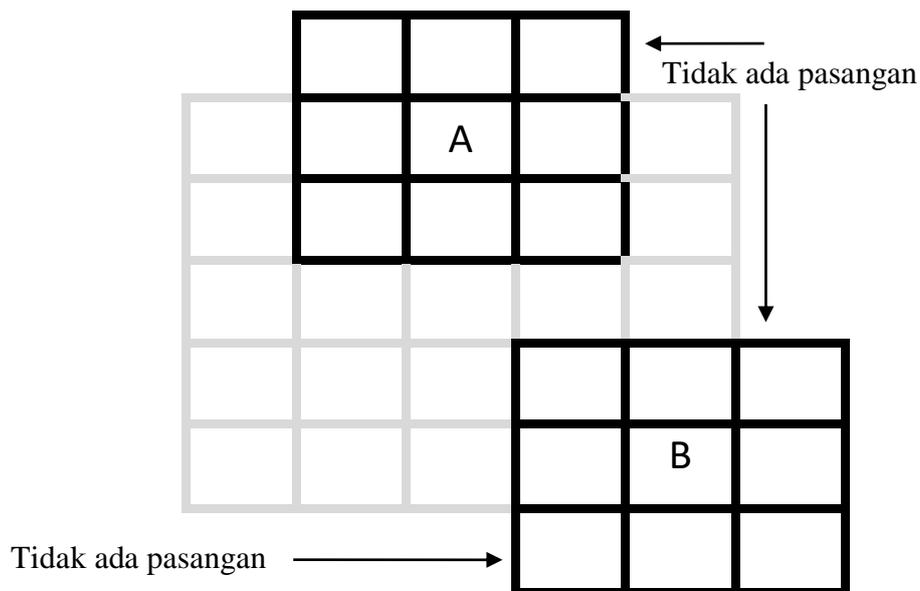
$$M = \sqrt{G_x^2 + G_y^2} = \sqrt{(11)^2 + (-7)^2} \cong |G_x| + |G_y| = |11| + |-7| = 18$$

Dengan demikian, nilai 1 diubah menjadi nilai 18 pada citra keluaran.

Dalam konvulsi terdapat dua kemungkinan yang jika ditemukan, diselesaikan dengan cara berikut, yaitu :

- a. Untuk hasil konvulsi menghasilkan nilai negatif maka nilai tersebut dijadikan 0.
- b. Jika hasil konvulsi menghasilkan nilai piksel lebih besar daripada nilai keabuan maksimum, maka nilai tersebut dijadikan nilai keabuan maksimum.

Pada matriks sobel dengan kernel 3 x3, terlihat bahwa tidak semua pixel dikenai konvulsi yaitu baris dan kolom yang terletak di tepi citra (*border*). Hal ini disebabkan karena piksel yang berada pada tepi citra tidak memiliki tetangga yang lengkap sehingga rumus konvulsi tidak berlaku pada piksel seperti itu. Gambar 3.5 menjelaskan contoh tentang hal ini. Sebagai contoh, konvulsi tidak mungkin dilakukan pada posisi A dan B.



Gambar 3.5 Masalah Pada Konvulsi

Masalah konvulsi pada piksel yang tidak mempunyai tetangga selalu terjadi pada piksel – piksel pinggir kanan, kiri, atas dan bawah. Solusi untuk masalah ini adalah :

- a. Abaikan piksel pada bagian tepi

Oleh karena pada bagian tepi citra tetangga tidak lengkap, sehingga piksel pada posisi tersebut tidak dikenail konvulsi. Sebagai konsekuensinya, citra yang tidak mengalami konvulsi akan diisi nol atau diisi sesuai pada citra asal. Alternative lain, bagian yang tidak diproses tidak diikutkan dalam citra hasil. Akibatnya ukuran hasil mengecil.

- b. Buat baris dan kolom tambahan pada bagian tepi

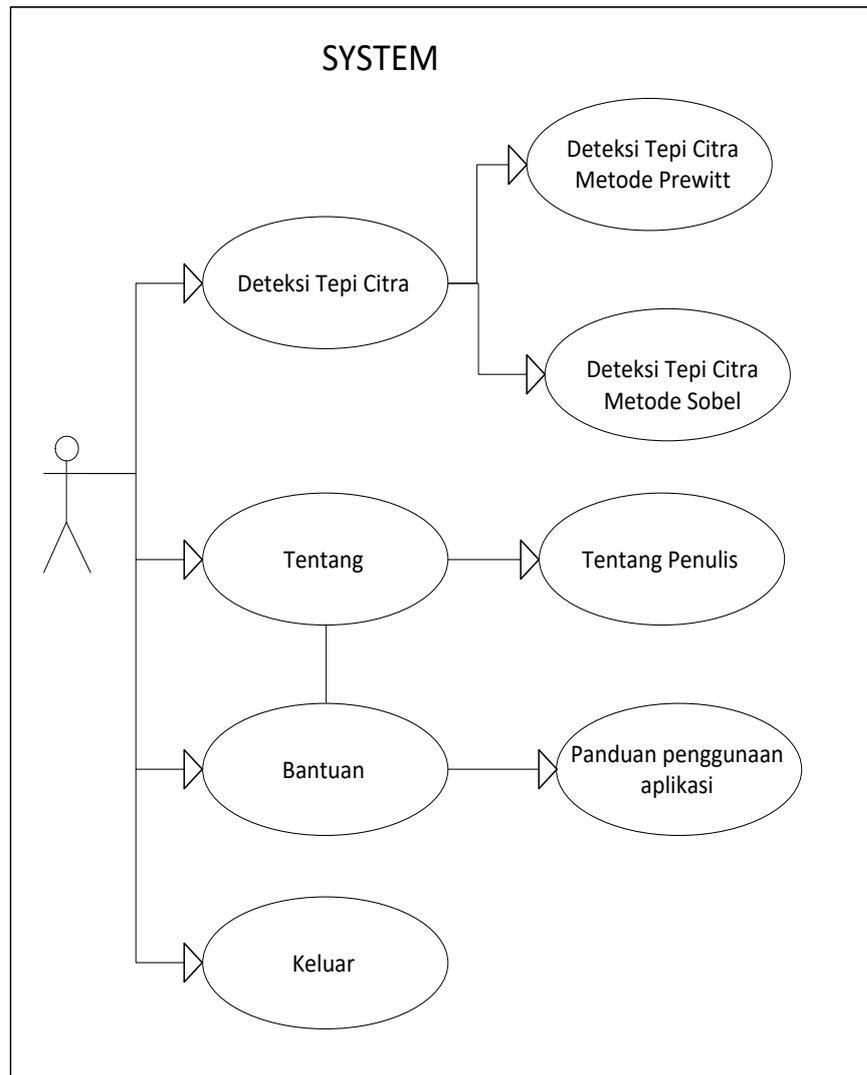
Baris dan kolom ditambahkan pada bagian tepi sehingga proses konvulsi dapat dilaksanakan. Dalam hal ini , baris dan kolom baru diisi dengan nilai nol.

3.4 Perancangan Sistem

Adapun perancangan system dari aplikasi deteksi tepi dengan metode sobel dan metode prewitt untuk pengenalan pola menggunakan Visual Studio 2013 ini yang digambarkan pada perancangan UML diantaranya sebagai berikut:

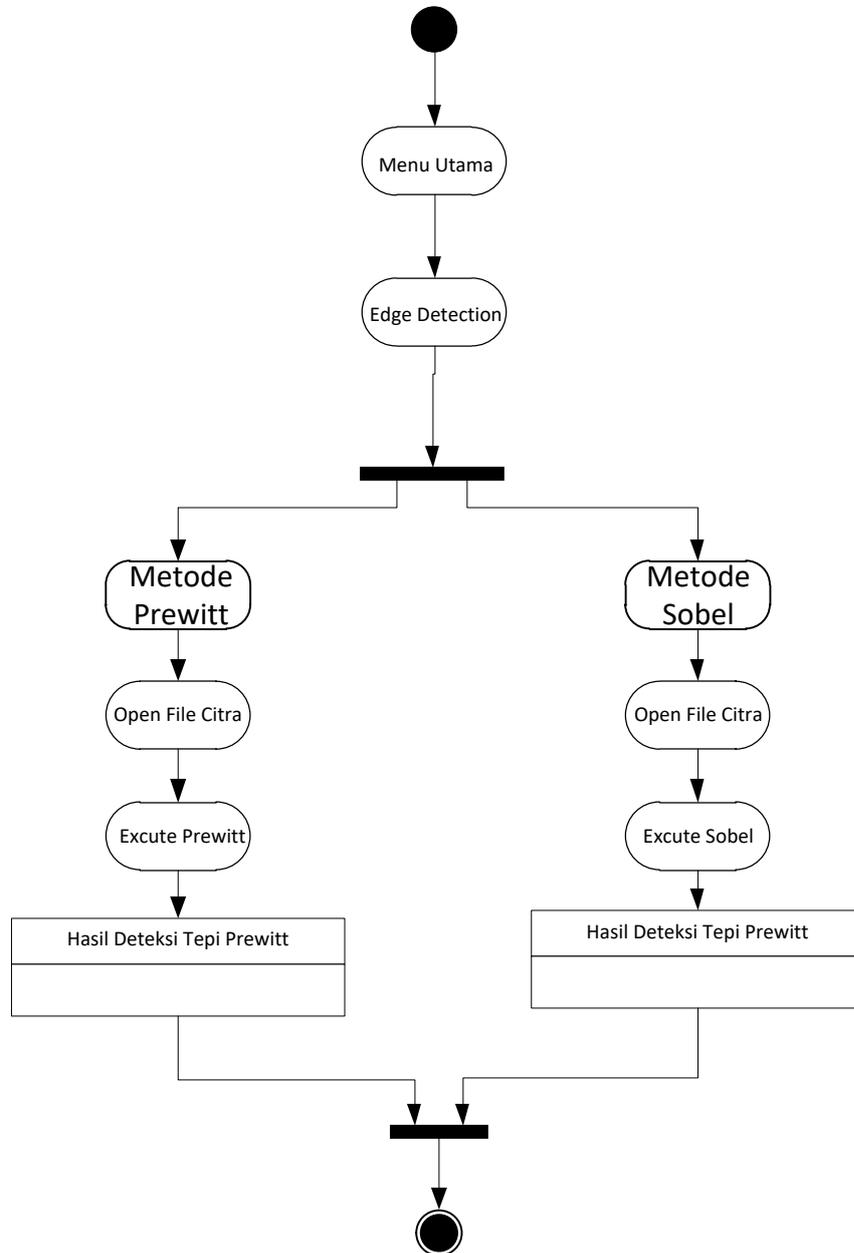
1. *Use Case Diagram*

Kegiatan interaksi Antara actor terhadap system ditunjukkan pada *use case diagram*, aktor yang terlibat dalam kegiatan tersebut adalah *user*. *Use case diagram* perangkat lunak yang dibangun terlihat pada gambar berikut :



Gambar 3.6 Use Case Diagram Perbandingan metode Prewitt dan Sobel pada perancangan aplikasi deteksi tepi citra

2. Activity Diagram



Gambar 3.7 Activity Diagram perbandingan metode *prewitt* dan *Sobel* pada perancangan aplikasi deteksi tepi citra

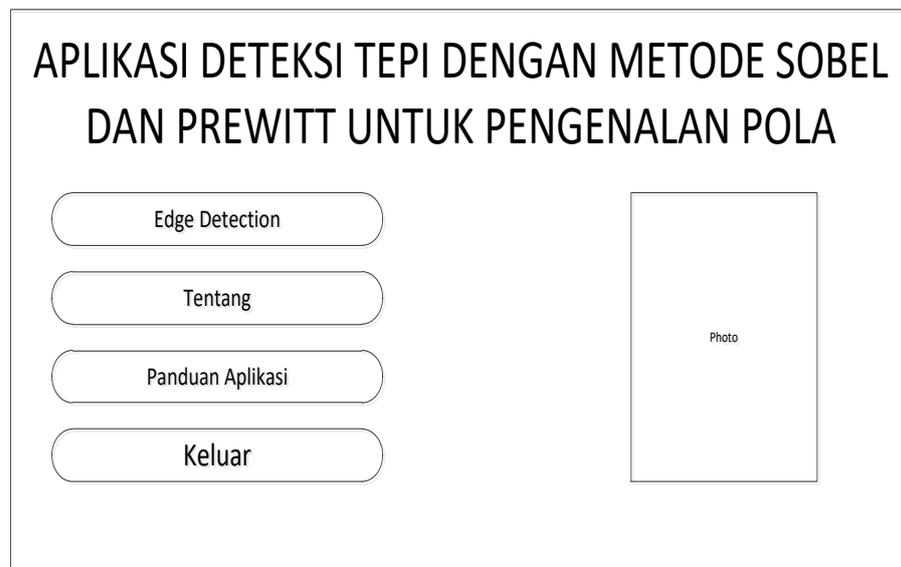
Pada bagian ini akan dimulai dengan membuka aplikasi untuk ke menu utama, pilih *edge detection* selanjutnya akan muncul muncul *form* pendeteksi tepi yaitu dengan metode *sobel* dan *prewitt* dan akan muncul hasil dari proses deteksi.

3. Desain *User Interface*

Dalam hal ini penulis akan membahas perancangan sistem yang akan dibangun secara terperinci, perancangan yang dibuat memuat dari perancangan *system* dan desain *interface*

a. Antarmuka Form Utama

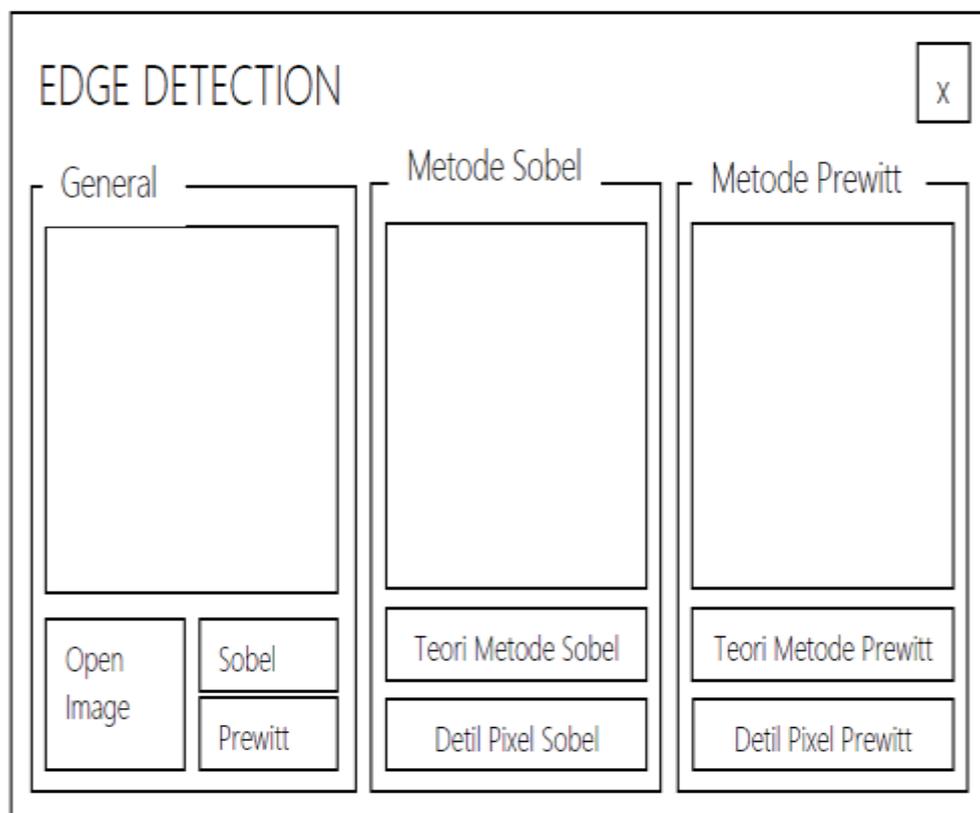
Rancangan form ini dibuat sebagai halaman utama dimana di form ini terdapat empat tombol yang akan membuka *form* lain seperti *form Edge Detection*, *form* tentang, *form* bantuan dan tombol keluar. Form ini nantinya akan memiliki fungsi yang berbeda-beda.



Gambar 3.8 Tampilan Menu Utama

b. Antar Muka Form Edge Detection

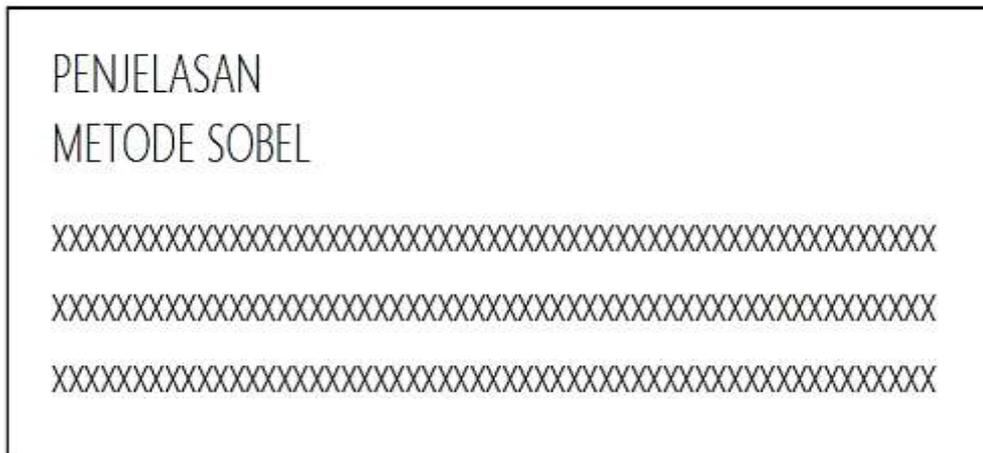
Rancangan form ini dibuat untuk melakukan proses deteksi tepi pada citra menggunakan metode *Prewitt* dan *Sobel*. Nantinya akan dimasukkan gambar pada form dan akan diproses untuk menghasilkan gambar deteksi tepi. Berikut *form edge detection*.



Gambar 3.9 Tampilan Menu *edge detection*

c. Antar Muka Form Metode Sobel

Rancangan Form ini dibuat untuk menampilkan penjelasan teori pada proses deteksi tepi dengan menggunakan metode *Sobel*. Berikut gambar form teori Metode Sobel.



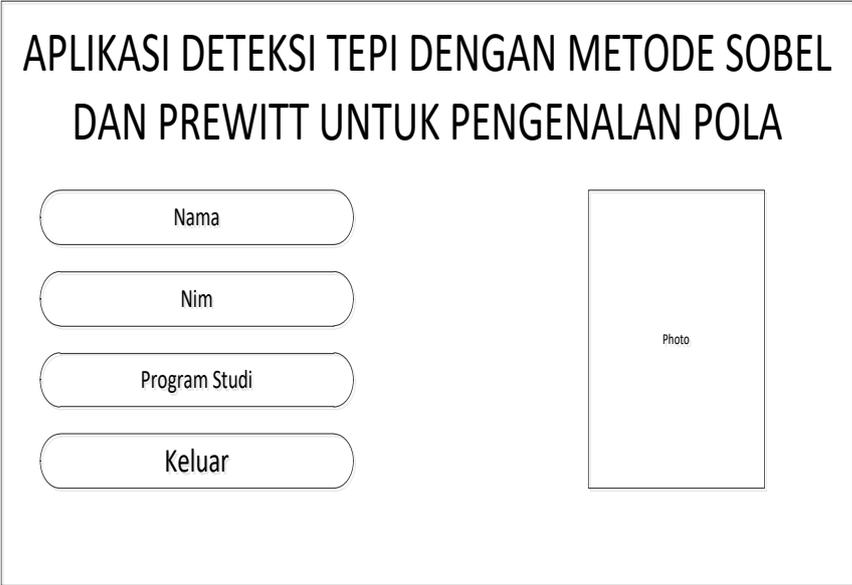
Gambar 3.10 Tampilan form teori metode sobel

d. Antar muka *Form* Teori Metode *Prewitt*

Rancangan *form* ini dibuat untuk menampilkan penjelasan teori pada proses deteksi tepi dengan menggunakan metode *prewitt*. *Form* ini juga akan lebih melengkapi dalam rancangan aplikasi ini. Berikut gambar *form* teori metode *prewitt*.

f. Antar muka Form Tentang

Rancangan form tentang ini dibuat untuk memberikan informasi dari pembuat program ini, yang terdiri dari nama, npm dan juga jurusan. Dan dapat dilihat pada gambar berikut :

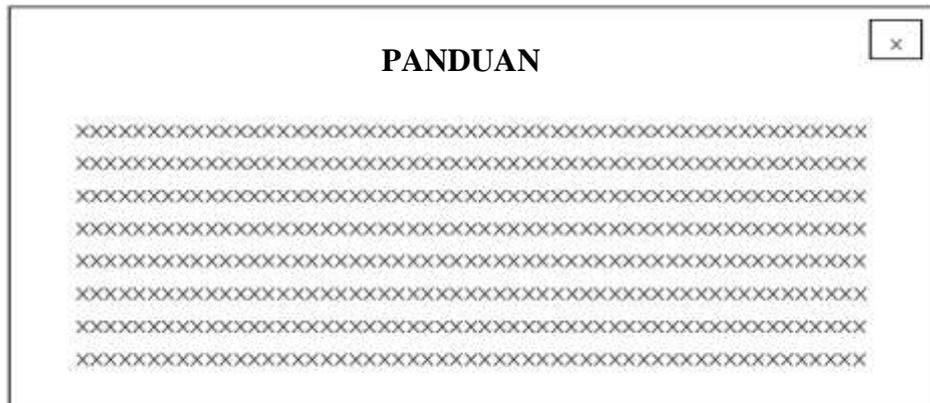


The image shows a screenshot of a web form titled "APLIKASI DETEKSI TEPI DENGAN METODE SOBEL DAN PREWITT UNTUK PENGENALAN POLA". The form contains four input fields on the left: "Nama", "Nim", "Program Studi", and "Keluar". On the right side, there is a rectangular area labeled "Photo" for uploading a profile picture.

Gambar 3.13 Tampilan Form Tentang

g. Antarmuka Form Panduan

Rancangan form Panduan ini berisikan panduan penggunaan aplikasi yang sudah dirancang dan ini akan memudahkan pengguna dalam pengoperasian aplikasi deteksi tepi dapat dilihat pada gambar berikut :



Gambar 3.14 Tampilan form panduan

BAB IV

HASIL DAN PEMBAHASAN

Sistem yang dirancang, menggunakan antar muka pengolahan data dari pengujian yang telah dirancang pada bab sebelumnya, sehingga implementasi system diimplementasi menjadi aplikasi yang dapat digunakan.

4.1 Kebutuhan Spesifikasi Minimum Hardware Dan Software

Dalam proses ini akan menggunakan metode *sobel dan prewitt* untuk pengenalan pola pada citra digital yang di butuhkan adalah sebagai berikut:

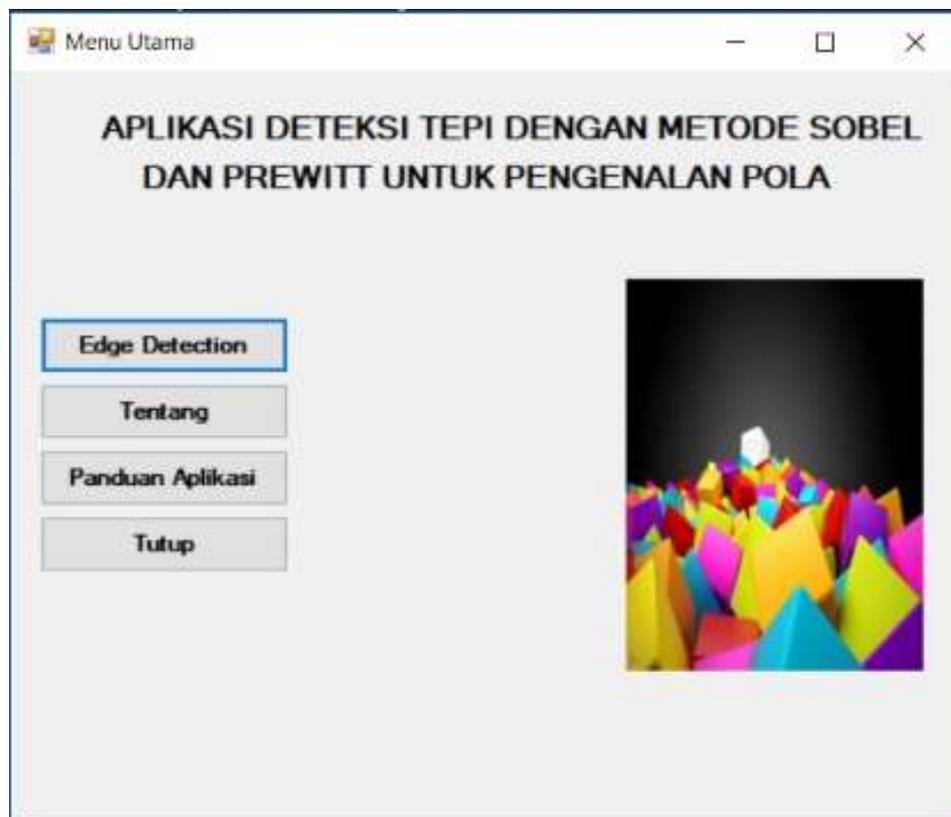
1. Perangkat Keras (*Hardware*)
 - a. Prosesor AMD APU A9-9420 Dual Core up to 3,6 GHz
 - b. Memori DDR4 4GB
 - c. Layar 14 inch
 - d. *Mouse*
 - e. *Keyboard*
2. Perangkat Lunak (*Software*)
 - a. Sistem Operasi Windows 10 Pro 62 bit
 - b. Ms VISUAL STUDIO 2013

4.2 Pengujian Aplikasi Program Dan Pembahasan

Adapun hasil dari tampilan program proses perbandingan antara metode *sobel dan prewitt* dapat dilihat pada gambar di berikut ini :

a. Menu Utama

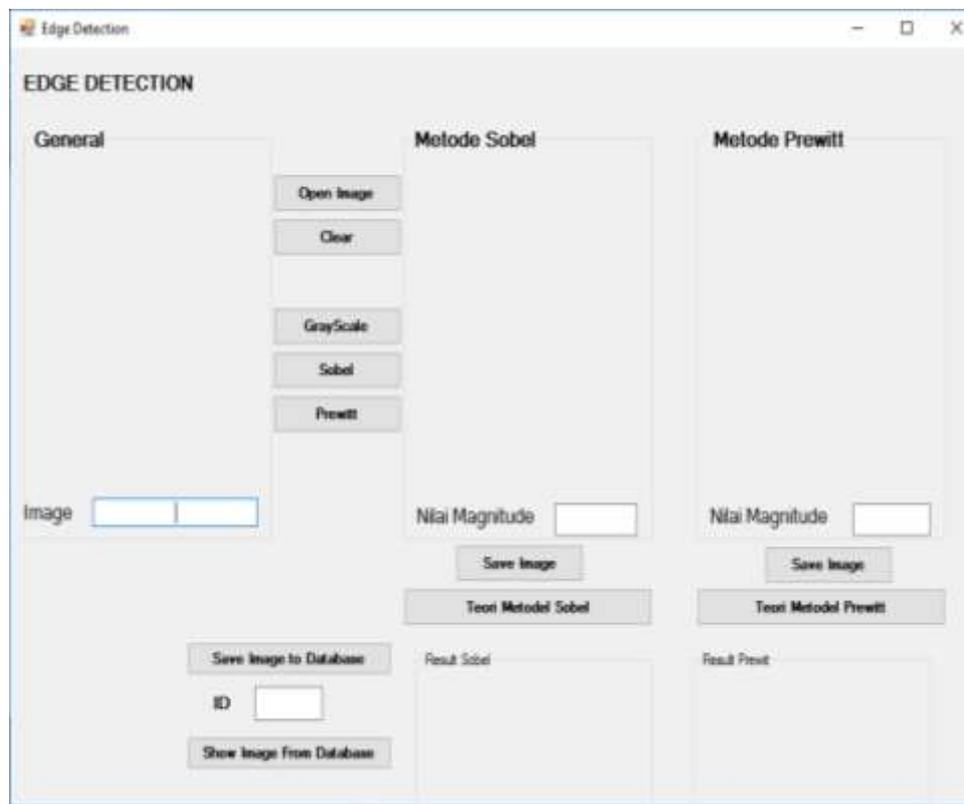
Halaman ini berfungsi sebagai menu utama dari aplikasi, terdapat beberapa menu berbentuk *button* yang berfungsi untuk ke menu fungsi berikutnya. Terdapat 3 menu utama, yaitu *Edge Detection*, tentang dan menu panduan aplikasi. Berikut ini adalah tampilan dari menu utama :



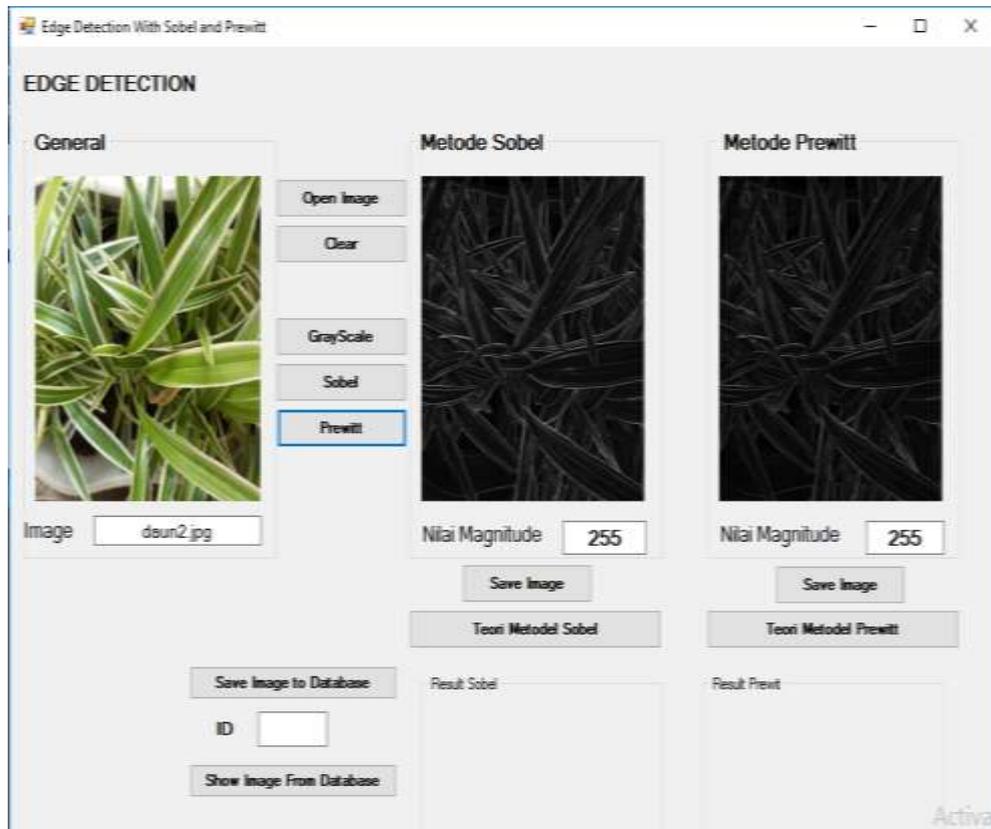
Gambar 4.1 Tampilan Menu Utama

b. Menu *Edge Detection*

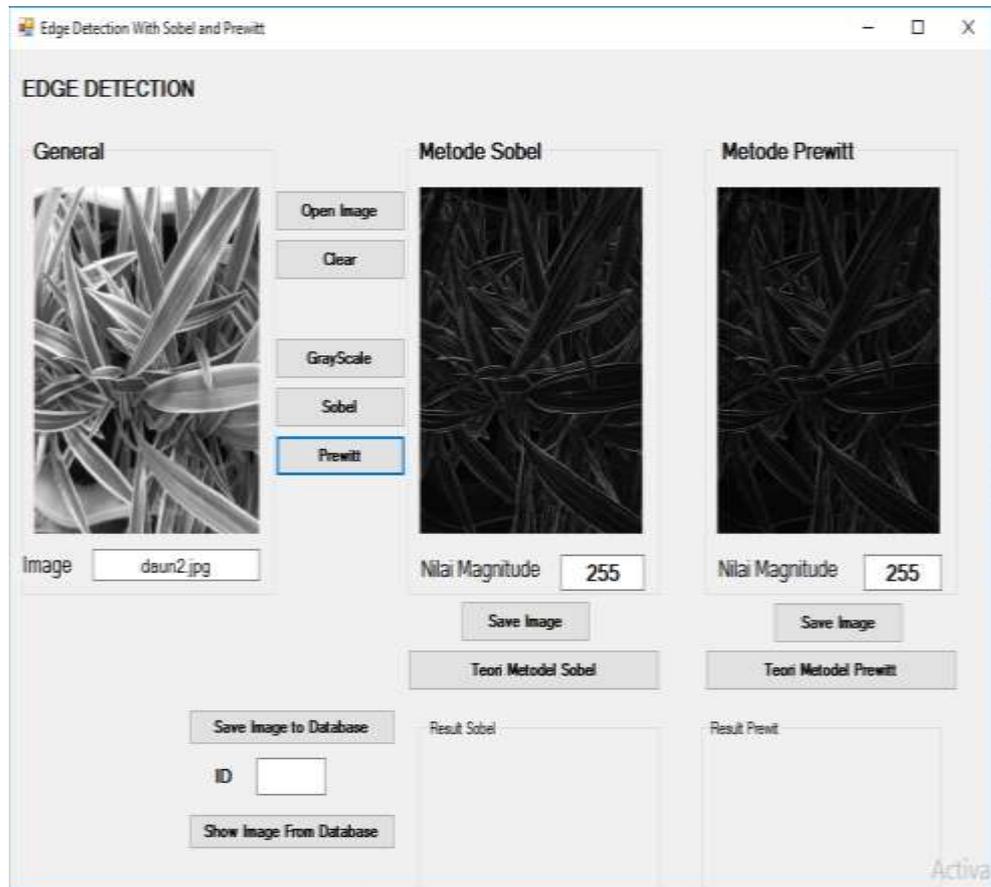
Halaman ini berfungsi untuk menguji citra yang telah di *capture* dengan menggunakan *webcam* yang telah disediakan. Pengujian citra dalam aplikasi ini adalah dengan menggunakan metode *sobel* dan *prewitt detection*. Berikut ini adalah tampilan halaman *edge detection* dan beberapa hasil tampilan dari *edge detection* dengan berbagai gambar yang berbeda :



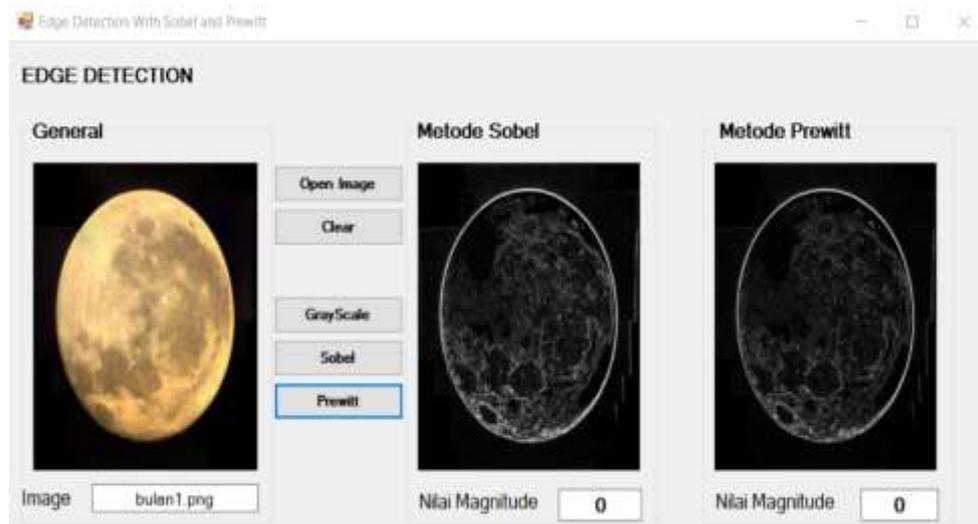
Gambar 4.2 Tampilan Halaman Edge Detection



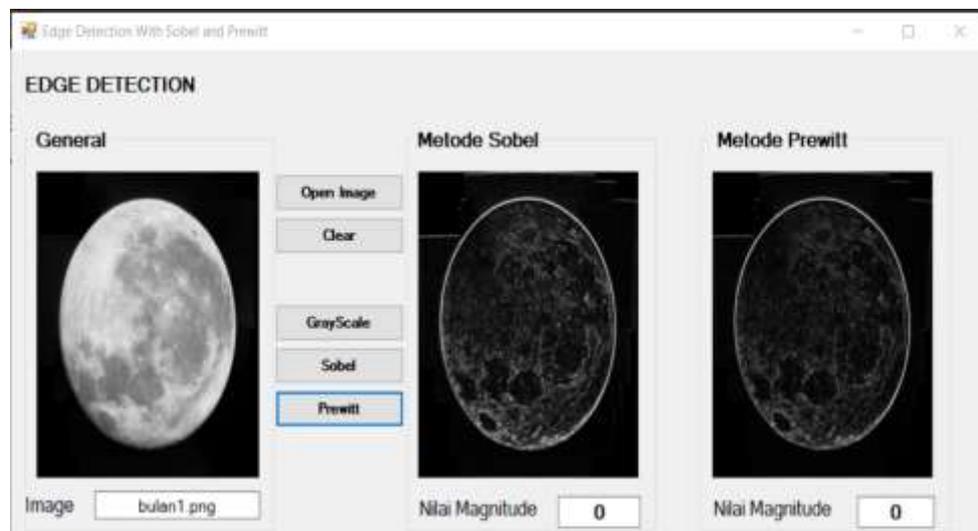
Gambar 4.3 Tampilan Halaman Edge Detection Dengan Contoh Citra tanpa proses Grayscale



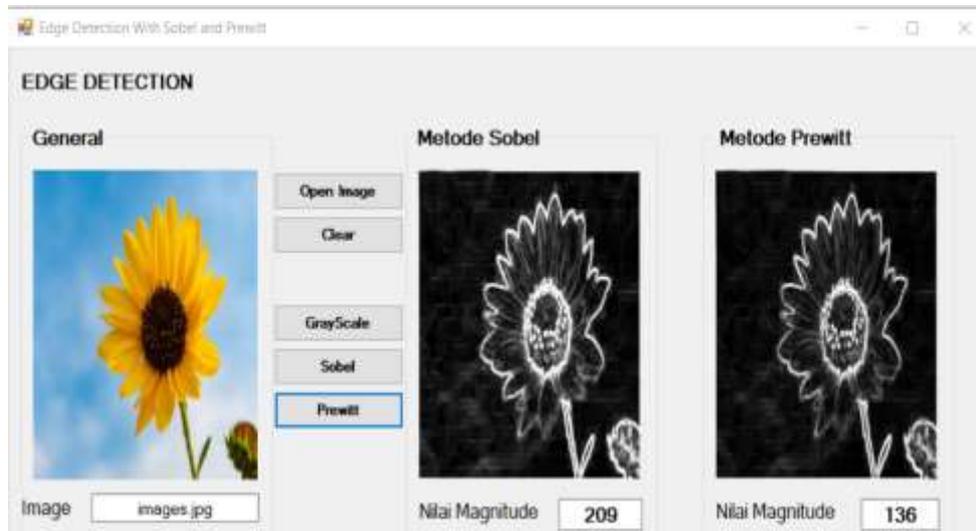
Gambar 4.4 Tampilan Halaman Edge Detection Dengan Contoh Citra dengan proses Grayscale



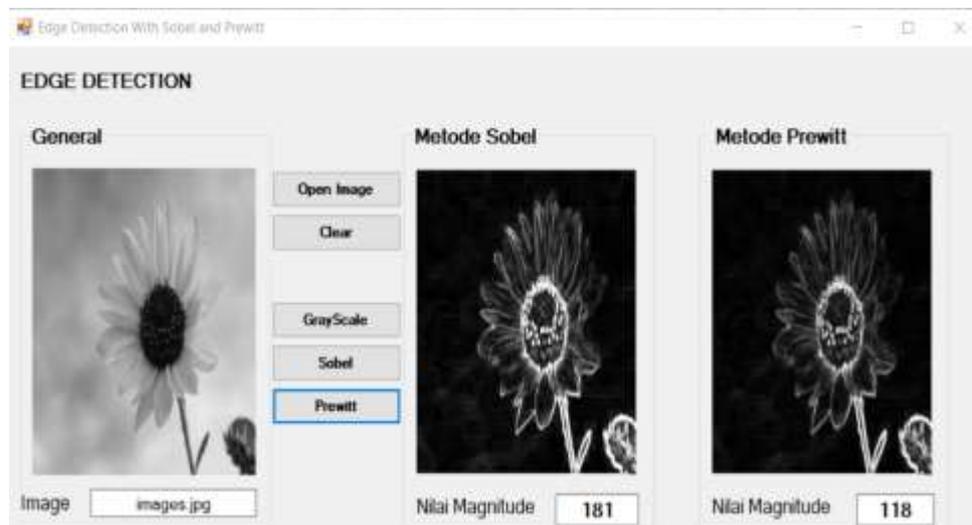
Gambar 4.5 Tampilan Halaman Edge Detection Dengan Contoh II Citra Tanpa Proses Grayscale



Gambar 4.6 Tampilan Halaman Edge Detection Dengan Contoh II Citra dengan proses Grayscale



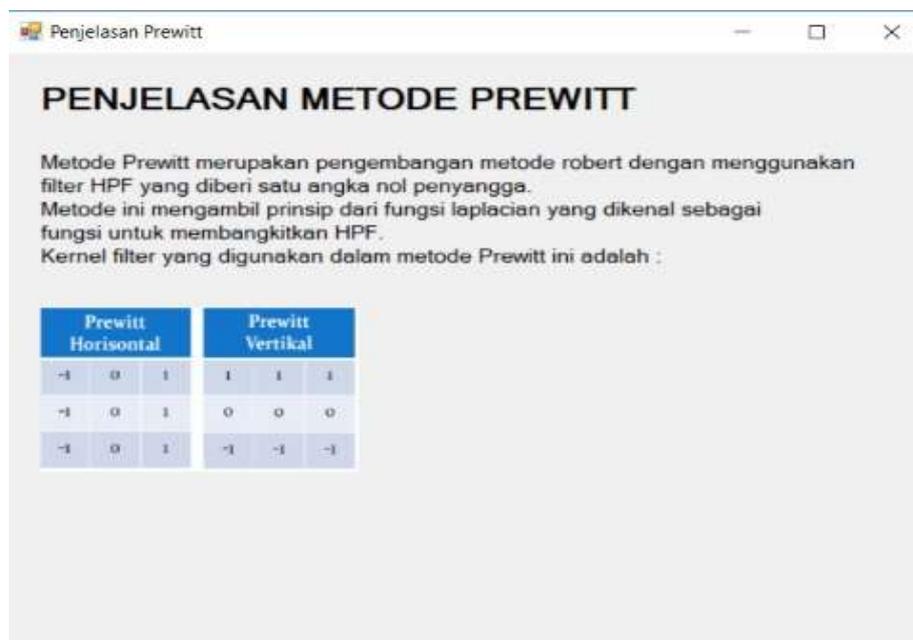
Gambar 4.7 Tampilan Halaman Edge Detection Dengan Contoh III Citra tanpa proses Grayscale



Gambar 4.8 Tampilan Halaman Edge Detection Dengan Contoh III Citra dengan proses Grayscale



Gambar 4.9 Tampilan Halaman Penjelasan Metode Sobel



Gambar 4.10 Tampilan Halaman Penjelasan Metode Sobel

Pada *form edge detection* terdapat berbagai macam *tools* yang memiliki fungsinya masing-masing, diantaranya sebagai berikut :

1. *Open image*

Tombol ini berfungsi untuk membuka gambar yang telah tersimpan di dalam computer, sehingga pengguna hanya perlu untuk langsung membuka gambar yang sudah ada.

2. *Clear*

Tombol ini berguna untuk menghapus gambar dan hasil citra pada box.

3. *Capture*

Tombol ini berfungsi untuk mengambil gambar ketika mengambil citra menggunakan kamera yang telah disiapkan, agar gambar dapat tampil pada *display* maka perlu di lakukan *capture*.

4. *Grayscale*

Tombol ini berfungsi untuk memproses citra dengan Metode *Grayscale* atau abu abu.

5. *Sobel*

Tombol ini berfungsi untuk menguji citra dengan metode sobel.

6. *Prewitt*

Tombol ini berfungsi untuk menguji citra dengan metode prewitt.

7. *Teori metode sobel*

Tombol ini berfungsi untuk menampilkan keterangan teori tentang metode sobel.

8. *Teori metode prewitt*

Tombol ini berfungsi untuk menampilkan keterangan teori tentang metode *prewitt*.

9. *Save image to database*

Tombol ini berfungsi untuk menyimpan gambar yang telah terdeteksi tepinya oleh metode *sobel* dan *prewitt*.

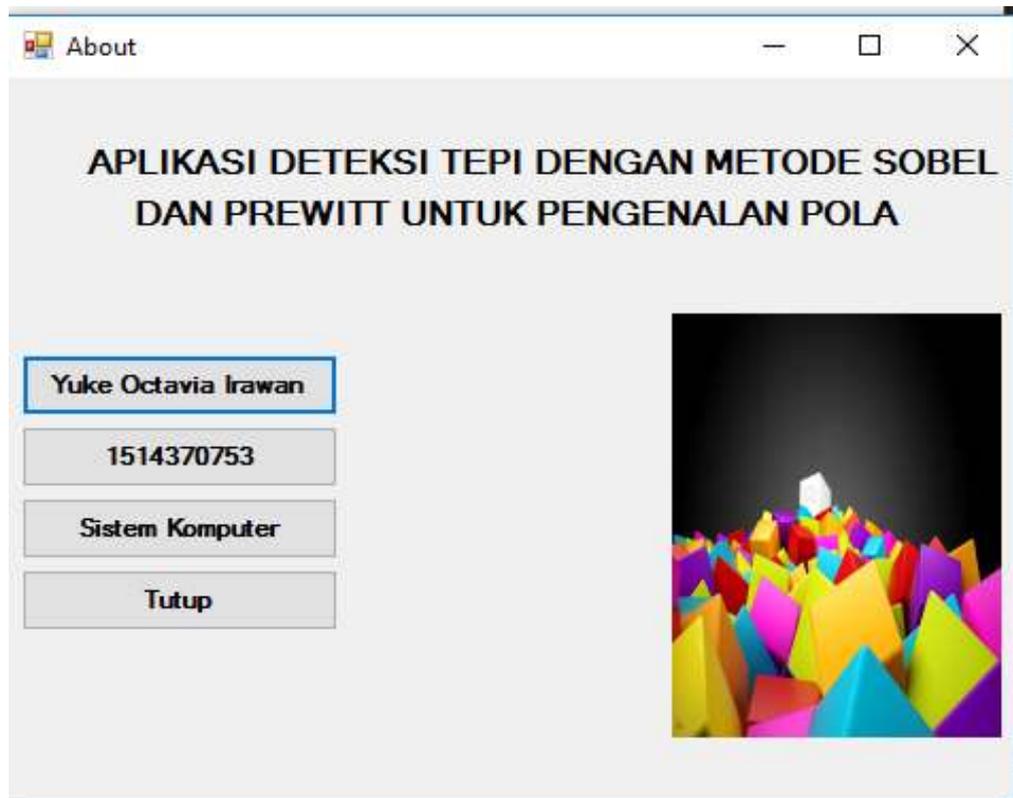
10. *Show image from database*

Tombol ini berfungsi untuk menampilkan gambar yang sebelum sudah disimpan pada database.

Seperti yang dijelaskan di atas bahwa semua *tools* memiliki fungsinya masing-masing pada suatu *form*.

c. Menu Tentang

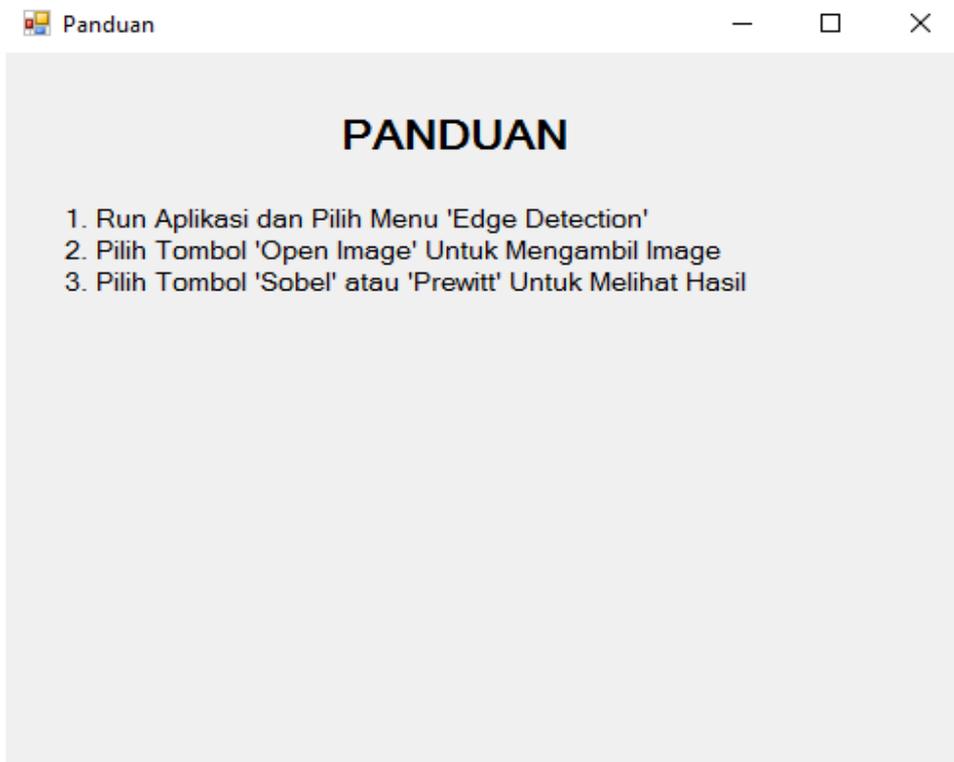
Halaman ini berfungsi sebagai informasi dari penulis, yaitu terdiri dari keterangan tentang nama, npm dan informasi jurusan. Berikut ini adalah tampilan dari menu tentang :



Gambar 4.11 Tampilan Halaman Menu Tentang Penulis

d. Menu Panduan Aplikasi

Halaman ini berfungsi sebagai informasi untuk menggunakan aplikasi yang benar. Berikut ini adalah tampilan dari menu tentang :



Gambar 4.12 Tampilan Halaman Panduan Aplikasi

BAB V

PENUTUP

5.1 Kesimpulan

Dengan adanya Aplikasi Deteksi Tepi Dengan Metode *Sobel* Dan *Prewitt* Untuk Pengenalan Pola Menggunakan Visual Studio 2013 ini dapat disimpulkan penulis sebagai berikut :

- a. Mengimplementasikan hasil penelitian perbandingan teori metode *sobel* dan *prewitt* menjadi sebuah aplikasi analisis perbandingan antara kedua metode tersebut.
- b. Metode yang digunakan dalam aplikasi ini adalah metode *prewitt* dan metode *sobel* yang mana kedua metode ini adalah metode citra deteksi tepi (*edge detection*).
- c. Perbandingan antara metode *sobel* dan *prewitt* dapat dilakukan dengan memindai citra yang telah ada dan di konversi menjadi hasil deteksi tepi citra.
- d. Dari hasil pengujian terhadap aplikasi deteksi tepi untuk pengenalan pola antara metode *sobel* dan *prewitt* dapat disimpulkan dalam beberapa percobaan yang dilakukan bahwa metode *sobel* dapat melakukan pengenalan pola lebih baik daripada metode *prewitt*. Pada hasil metode *sobel* terlihat tepi dan batas-batas pada citra lebih jelas terlihat.

5.2 Saran

Berdasarkan kesimpulan-kesimpulan yang telah dijelaskan diatas, maka penulis dapat memberikan beberapa saran yang dapat membantu mengatasi kekurangan yang ada diantaranya :

- a. Kepada pembaca, penulis menyarankan agar system aplikasi ini dapat di kembangkan menjadi perbandingan lebih banyak metode citra, serta dapat di sempurnakan kekurangan yang terdapat pada aplikasi ini.
- b. Perbandingan metode yang dapat diimplimentasi di aplikasi ini bias lebih dari 2 metode jika dikembangkan di versi berikutnya.

DAFTAR PUSTAKA

- Akbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. *Jurnal Teknik dan Informatika*, 5(1), 1-5.
- Aplikasi. (2015). Kamus Besar Bahasa Indonesia. Diakses 11 Oktober 2019, dari <http://kbbi.web.id/aplikasi>
- Apriyana, Delta Sri Maharani, Shinta Puspasari, Renni Angreni. Perbandingan Metode Sobel, Metode Prewitt dan Metode Robert untuk Deteksi Tepi Objek pada Aplikasi Pengenalan Bentuk Berbasis Citra Digital, 2-3. Diakses 11 Oktober 2019
- Asmardi Zalukhu. (2016). Implementasi Metode Canny dan Sobel untuk Mendeteksi Tepi Citra, 3 (6), 25. Diakses 11 Oktober 2019
- Azmi, Fadhillah, And Winda Erika. "Analisis Keamanan Data Pada Block Cipher Algoritma Kriptografi Rsa." *Cess (Journal Of Computer Engineering, System And Science)* 2.1: 27-29.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).
- Katen Lumbanbatu, Novriyeni. (2013). Perancangan Sistem Informasi Penyebaran Penduduk menggunakan Php My Sql pada Kecamatan Binjai Selatan, 7

- Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan*, 11(1), 1-6.
- Mahyuni, Sharipuddin, Martono. (2014). Perancangan Sistem Pengolahan Data pada SMA Negeri 6 Kabupaten Tebo, 8 (1), 182. Diakses 11 Oktober 2019
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Marlina, L., Putera, A., Siahaan, U., Kurniawan, H., & Sulistianingsih, I. (2017). Data Compression Using Elias Delta Code. *Int. J. Recent Trends Eng. Res*, 3(8), 210-217.
- Muttaqin, Muhammad. "Analisa Pemanfaatan Sistem Informasi E-Office Pada Universitas Pembangunan Panca Budi Medan Dengan Menggunakan Metode Utaut." *Jurnal Teknik Dan Informatika* 5.1 (2018): 40-43.
- Muttaqin, Muhammad. "Portal Academic Portal Innovation Based On Website In The Era Of Digital 4.0 Technology Now."
- Nency Extise Putri, Supriandi Azpar. (2017). Sistem Informasi Pengolahan Data Pendidikan Anak Usia Dini (PAUD) Terpadu Amalia Syukra Padang, 2
- Perwitasari, I. D. (2018). Teknik Marker Based Tracking Augmented Reality untuk Visualisasi Anatomi Organ Tubuh Manusia Berbasis Android. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 8-18.
- Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research*, 3(8), 196-201.
- Ramadhani, S., Suherman, S., Melvasari, M., & Herdianto, H. (2018). Perancangan Teks Berjalan Online Sebagai Media Informasi Nelayan. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).
- Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) pada Perancangan Sistem Informasi Remunerasi Dosen dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan), 3 (1), 2-3. Dakses 11 Oktober 2019