

**SKRIPSI**  
**APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK PENDETEKSI JAWABAN**  
**PADA LEMBAR JAWABAN KOMPUTER**  
**MENGGUNAKAN *ALGORITMA SOBEL***

**Jenjang Strata I**



**Oleh:**

**GITA TRIYANA**

**G1A010046**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS BENGKULU**

**2014**

**APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK PENDETEKSI**

**JAWABAN PADA LEMBAR JAWABAN KOMPUTER**

**MENGGUNAKAN *ALGORITMA SOBEL***

**(Studi kasus SMP Negeri 2 Kota Bengkulu)**

**SKRIPSI**

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana  
Pada Fakultas Teknik Program Studi Teknik Informatika  
Universitas Bengkulu



Oleh

**GITA TRIYANA**  
**G1A010046**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS BENGKULU**

**2014**

## MOTTO DAN PERSEMBAHAN

Motto :

دَرَجَاتٍ تُؤْتِي الْعِلْمَ أَوْ وَالَّذِينَ آمَنُوا يَنْزِلُ اللَّهُ يَرَفَعُ

*Artinya : "Allah meninggikan orang-orang yang beriman diantara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat"*

*-Qs. AL Mujadalah ayat 11-*

*"Success is a state of mind. If you want success,  
start thinking of yourself as a success."*

Persembahan :

- Allah SWT.
- Nabi Muhammad SAW.
- Ayah dan Ibu tercinta, Muharam dan Elyantina sebagai pembimbing hidup yang luar biasa. Segala syukur dan terima kasih atas segala kasih sayang dan dukungan yang tidak mungkin dapat tergantikan.
- Kakak , Muthia Rozalina dan Dwi Handayani, serta Adik, Arief Havid Zasella
- Teman terdekat, Sandi Yusmantoro, S.T.
- Sahabat dan teman seperjuangan TI 2010.
- Almamater.

## KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillah penulis ucapkan atas segala rahmat dan ridho Allah SWT. sehingga penulis dapat menyelesaikan skripsi yang berjudul **Aplikasi Pengolahan Citra Digital untuk Pendeteksi Jawaban pada Lembar Jawaban Komputer menggunakan *Algoritma Sobel*** dengan lancar.

Skripsi ini merupakan tugas akhir untuk memenuhi salah satu syarat dalam menyelesaikan studi di Fakultas Teknik Universitas Bengkulu. Selain itu, skripsi ini juga merupakan suatu pembelajaran bagi penulis dalam banyak hal, baik ilmu pengetahuan, praktik ilmu yang didapat selama kuliah di Teknik Informatika dan bekal ilmu ke depannya.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, oleh karena itu penulis masih perlu banyak belajar lagi dan membutuhkan saran yang membangun dalam skripsi ini. Penulis juga menyadari bahwa tanpa bantuan, saran, dan bimbingan dari pihak-pihak yang telah membantu, penulis tidak akan mampu menyelesaikan tugas akhir ini. Untuk itu penulis mengucapkan terima kasih kepada yang terhormat:

1. Ayah (Muharam) dan Ibu (Elyantina) tercinta yang telah memberikan dukungan materi, motivasi, dan do'a dalam penyelesaian studi dan skripsi ini.

2. Bapak Khairul Amri, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Bengkulu yang telah memberikan kesempatan kepada penulis untuk menyelesaikan pendidikan di Fakultas Teknik Universitas Bengkulu.
3. Ibu Desi Andreswari, S.T., M.Cs., selaku Ketua Program Studi Teknik Informatika Universitas Bengkulu yang telah memberikan arahan dalam penyelesaian skripsi ini.
4. Bapak Drs. Asahar Johar, M.Si, selaku Pembimbing Utama yang penuh kesabaran dan bersedia meluangkan waktu untuk membimbing dan memotivasi penulis dalam menyelesaikan skripsi ini.
5. Ibu Desi Andreswari, S.T., M.Cs., selaku Pembimbing Pendamping yang penuh kesungguhan dan selalu sabar dalam membimbing serta bersedia untuk meluangkan waktu untuk membimbing penulis dalam menyelesaikan skripsi ini.
6. Bapak Rusdi Efendi, S.T., M.Kom., dan Ibu Ernawati, S.T, M.Cs., selaku penguji yang bersedia meluangkan waktu untuk menguji penulis.
7. Segenap Bapak dan Ibu dosen pengajar di Fakultas Teknik, khususnya Program Studi Teknik Informatika dan segenap staf administrasi di Fakultas Teknik Universitas Bengkulu.
8. Kepala Kurikulum SMP Negeri 2 Kota Bengkulu, Bapak Heri Winarno, S.Pd. beserta staf, yang telah banyak membantu dalam pemberian data dan informasi serta pengujian aplikasi.
9. Teman terdekat, Sandi Yusmanto, S.T yang telah banyak membantu penulis dalam memahami *source code* program.

10. Sahabat yang membantu memberikan semangat, Anindita Ayu, Ambar Pravitasari, Atri Deliya, Stephanie Anditasari, Dike Bayu Maghfira, Lisa Rolita, Ili Susanti serta teman seperjuangan skripsi Fathin Ulfa Karimah dan Ancemona Yuda.
11. Teman yang membantu disaat waktu yang mendesak, Rahmad Hidayat, Nasrullah, Roberto Sihombing dan Dery Kurniawan.
12. Semua pihak yang telah membantu proses penyelesaian skripsi yang tidak dapat disebutkan satu-persatu, khususnya TI 10 gellenjid.

Semoga seluruh bantuan, dukungan dan motivasi yang diberikan kepada penulis menjadi amal dan mendapat balasan dari Allah SWT. Semoga skripsi ini berguna bagi semua pihak di masa yang akan datang.

Bengkulu, Mei 2014

**Penulis**

**APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK PENDETEKSI  
JAWABAN PADA LEMBAR JAWABAN KOMPUTER  
MENGUNAKAN ALGORITMA SOBEL**

**(Studi kasus SMP Negeri 2 Kota Bengkulu)**

**Oleh**

**GITA TRIYANA**

**G1A010046**

**Email : triyanagita@yahoo.com**

**ABSTRAK**

Lembar Jawaban Komputer (LJK) merupakan lembar jawaban yang digunakan untuk melakukan ujian secara tertulis. Pemeriksaan LJK harus menggunakan alat *scanner* khusus, mengingat harga *scanner* sangat mahal maka penulis bermaksud untuk membangun aplikasi pengolahan citra digital untuk mendeteksi jawaban pada lembar jawaban komputer menggunakan Algoritma 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. Penulis mengubah data citra LJK menjadi bentuk deteksi tepi sobel untuk mempermudah mendeteksi jawaban pada LJK. Hasil deteksi jawaban pada LJK akan dilakukan pencocokkan dengan data kunci jawaban untuk memperoleh skor nilai. Selain itu laporan nilai siswa dapat ditampilkan secara otomatis menggunakan *Fastreport*. Aplikasi ini dibangun menggunakan bahasa pemrograman *Delphi*. Dan metode pengembangan sistem pada aplikasi ini menggunakan metode *Waterfall*. Sedangkan metode untuk perancangan sistem, penulis menggunakan *Unified Modeling Language* (UML). Hasil akhir dari penelitian ini adalah terciptanya suatu aplikasi pendeteksi jawaban pada LJK yang dapat digunakan melakukan pengoreksian LJK.

**Kata kunci: Pengolahan Citra Digital, Sobel, LJK, Deteksi LJK.**

**APPLICATION OF DIGITAL IMAGE PROCESSING FOR DETECTION  
ANSWER ON THE ANSWER COMPUTER SHEET  
USING ALGORITHM SOBEL  
(Case Study Junior High School 2 City of Bengkulu )**

**By**

**GITA TRIYANA**

**G1A008019**

**Email : triyanagita@yahoo.com**

**ABSTRACT**

*Answer Computer Sheet ( ACS ) an answer sheet that is used to the test in writing. Examination ACS must use a special scanner tool, considering the price scanners are very expensive , the authors intend to build applications for sensing digital image processing answers on the answer computer sheet using algorithm Sobel. Sobel's method is a development Robert's method using filter HPF ( High Pass Filter) is given a zero buffer . Excess of this Sobel's method is the ability to reduce noise before doing the calculation of edge detection. The author changed the image data ACS into a form Sobel edge detection to facilitate detection of the answers on ACS. Detection results on answer ACS will be data matching with answer key to obtain a score value. In addition the student score reports can be displayed automatically using FastReport. This application was built using the Delphi programming language. And methods of application system development using Waterfall method. While methods for system design , the author use the Unified Modeling Language (UML). The final result of this research is an application for correction ACS.*

***Keywords: Digital Image Processing, Sobel, ACS, Detection ACS.***



## DAFTAR ISI

|  |              |
|--|--------------|
| <b>HALAMAN JUDUL</b> .....             | <b>ii</b>    |
| <b>HALAMAN PERSETUJUAN</b> .....       | <b>iii</b>   |
| <b>HALAMAN PENGESAHAN</b> .....        | <b>iv</b>    |
| <b>MOTTO DAN PERSEMBAHAN</b> .....     | <b>v</b>     |
| <b>KATA PENGANTAR</b> .....            | <b>vi</b>    |
| <b>ABSTRAK</b> .....                   | <b>ix</b>    |
| <b>ABSTRACT</b> .....                  | <b>x</b>     |
| <b>DAFTAR ISI</b> .....                | <b>xi</b>    |
| <b>DAFTAR GAMBAR</b> .....             | <b>xiv</b>   |
| <b>DAFTAR TABEL</b> .....              | <b>xvii</b>  |
| <b>DAFTAR LAMPIRAN</b> .....           | <b>xviii</b> |
| <br>                                   |              |
| <b>BAB I PENDAHULUAN</b> .....         | <b>1</b>     |
| 1.1 Latar Belakang Penelitian .....    | 1            |
| 1.2 Rumusan Masalah .....              | 3            |
| 1.3 Batasan Masalah .....              | 4            |
| 1.4 Tujuan Penelitian .....            | 4            |
| 1.5 Manfaat Penelitian .....           | 5            |
| 1.6 Sistematika Penulisan .....        | 5            |
| <br>                                   |              |
| <b>BAB II LANDASAN TEORI</b> .....     | <b>7</b>     |
| 2.1 Lembar Jawaban Komputer .....      | 7            |
| 2.2 Pengolahan Citra Digital .....     | 8            |
| 2.3 Deteksi Tepi (Edge Detection)..... | 10           |
| 2.4 Algoritma Sobel .....              | 13           |
| 2.5 Model Pengembangan Waterfal .....  | 15           |
| 2.6 Unified Modelling Language .....   | 17           |
| 2.6.1 Struktur Diagram.....            | 19           |
| 2.6.2 Bihavior Diagram.....            | 23           |
| 2.7 Delphi 7 .....                     | 31           |
| 2.8 Basisdata.....                     | 32           |

|   |           |
|---|-----------|
| <b>BAB III METODOLOGI PENELITIAN .....</b>                                  | <b>34</b> |
| 3.1 Waktu dan Tempat Penelitian .....                                       | 34        |
| 3.2 Metode Pengambilan Sampel.....  | 34        |
| 3.2.1. Populasi.....  | 34        |
| 3.2.2. Sampel.....  | 34        |
| 3.3 Metode Pengumpulan Data .....   | 35        |
| 3.4 Metode Pengembangan Sistem .....  | 36        |
| 3.5 Metode Pengujian Sistem.....  | 40        |
| 3.6 Metode Uji Kelayakkan Sistem.....                                       | 41        |
| 3.7 Diagram Alir Penelitian .....   | 42        |
| 3.8 Jadwal Penelitian.....  | 43        |
| <br>  |           |
| <b>BAB IV ANALISA DAN PERANCANGAN SISTEM .....</b>                          | <b>44</b> |
| 4.1 Analisa Sistem .....  | 44        |
| 4.1.1 Identifikasi Masalah ( <i>Identify</i> ) .....                        | 44        |
| 4.1.2 Pemahaman Kerja Sistem yang ada ( <i>Understand</i> ) .....           | 44        |
| 4.1.3 Analisis Sistem ( <i>Analyze</i> ) .....                              | 44        |
| 4.1.4 Analisis Sebab Akibat .....   | 46        |
| 4.2 Perancangan Sistem .....  | 47        |
| 4.2.1 Perancangan Diagram Alir ( <i>Flowchart</i> ) Algoritma ElGamal ..... | 47        |
| 4.2.2 Perancangan Model UML .....   | 47        |
| 4.2.2.1 <i>Behavior Diagrams</i> .....                                      | 48        |
| 4.2.2.2 <i>Structure Diagram</i> .....                                      | 54        |
| 4.2.3 Perancangan <i>Database</i> .....                                     | 58        |
| 4.2.4 Perancangan <i>Table Relationship</i> .....                           | 61        |
| 4.2.5 Perancangan <i>User Interface</i> .....                               | 62        |
| <br>  |           |
| <b>BAB V HASIL DAN PEMBAHASAN.....</b>                                      | <b>69</b> |
| 5.1 Hasil Implementasi Aplikasi .....                                       | 69        |
| 1. Form Utama .....   | 69        |
| 2. Form Pengguna .....  | 72        |
| 3. Form Matapelajaran .....   | 73        |
| 4. Form Nomor Ujian .....   | 74        |
| 5. Form Kunci Jawaban .....   | 76        |
| 6. Form Koreksi .....   | 79        |
| 7. Form Report .....  | 94        |
| 5.2 Pengujian Blackbox Testing.....   | 102       |
| 1. Kasus Login .....  | 103       |
| 2. Kasus Tambah, Simpan dan Hapus Data .....                                | 104       |

|   |            |
|---|------------|
| 3. Kasus Penyimpanan Data Hasil Deteksi .....             | 105        |
| 4. Kasus Menampilkan Data Jawaban dan Kunci Jawaban ..... | 106        |
| 5.3 Pengujian Waktu .....                                 | 107        |
| 1. Manajemen Input Data .....                             | 107        |
| 2. Manajemen Koreksi LJK .....                            | 108        |
| 5.4 Pengujian Kelayakkan.....                             | 109        |
| 5.5 Perhitungan Manual .....                              | 110        |
| <b>BAB VI KESIMPULAN DAN SARAN .....</b>                  | <b>113</b> |
| 6.1 Kesimpulan .....                                      | 113        |
| 6.2 Saran .....   | 114        |
| <b>DAFTAR PUSTAKA .....</b>                               | <b>115</b> |
| <b>LAMPIRAN .....</b>                                     | <b>117</b> |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Model Tepi Satu Dimensi .....                  | 10 |
| Gambar 2.2 Proses Deteksi Tepi Citra .....                | 11 |
| Gambar 2.3 Proses Filter .....                            | 13 |
| Gambar 2.4 Mask Sobel.....                                | 14 |
| Gambar 2.5 Arah Gradien .....                             | 15 |
| Gambar 2.6 <i>Class Diagram</i> .....                     | 20 |
| Gambar 2.7 <i>Deployment Diagram</i> .....                | 21 |
| Gambar 2.8 <i>Component Diagram</i> .....                 | 22 |
| Gambar 2.9 <i>Composite Structure Diagram</i> .....       | 23 |
| Gambar 2.10 <i>Activity Diagram</i> .....                 | 25 |
| Gambar 2.11 <i>Sequence Diagram</i> .....                 | 26 |
| Gambar 2.12 <i>Communication Diagram</i> .....            | 28 |
| Gambar 2.13 <i>Behavior State Machine Diagram</i> .....   | 29 |
| Gambar 2.14 <i>Use Case Diagram</i> .....                 | 30 |
| Gambar 3.1 Model Waterfal .....                           | 39 |
| Gambar 3.2 Diagram Alir Penelitian .....                  | 42 |
| Gambar 4.1 <i>Flowchart Algoritma Sobel</i> .....         | 47 |
| Gambar 4.2 <i>Use Case Diagram</i> .....                  | 49 |
| Gambar 4.3 <i>Activity Diagram</i> .....                  | 50 |
| Gambar 4.4 <i>Seuence Diagram</i> Login User .....        | 51 |
| Gambar 4.5 <i>Sequence Diagram</i> Koreksi LJK .....      | 52 |
| Gambar 4.6 <i>Communication Diagram</i> Login Admin ..... | 53 |
| Gambar 4.7 <i>Communication Diagram</i> Koreksi LJK ..... | 53 |
| Gambar 4.8 <i>State Machine Diagram</i> .....             | 54 |
| Gambar 4.9 <i>Class Diagram</i> .....                     | 55 |
| Gambar 4.10 <i>Composite Diagram</i> .....                | 56 |
| Gambar 4.11 <i>Component Diagram</i> .....                | 57 |
| Gambar 4.12 <i>Deployment Diagram</i> .....               | 58 |
| Gambar 4.13 <i>Entity Relatioship Diagram</i> .....       | 58 |
| Gambar 4.14 <i>Table Relationship Diagram</i> .....       | 61 |
| Gambar 4.15 Rancangan Form Utama .....                    | 62 |
| Gambar 4.16 Rancangan Form Pengguna .....                 | 63 |
| Gambar 4.17 Rancangan Form Matapelajaran .....            | 64 |
| Gambar 4.18 Rancangan Form Nomor Ujian .....              | 64 |
| Gambar 4.19 Rancangan Form Kunci Jawaban .....            | 65 |

|  |    |
|--|----|
| Gambar 4.20 Rancangan Form Koreksi .....                             | 66 |
| Gambar 4.21 Rancangan Form Report .....                              | 68 |
| Gambar 5.1 Form Utama .....  | 69 |
| Gambar 5.2 Login Sistem .....  | 70 |
| Gambar 5.3 Autentifikasi Login .....                                 | 70 |
| Gambar 5.4 Tampilan Menu Form Utama .....                            | 71 |
| Gambar 5.5 Form Pengguna .....                                       | 72 |
| Gambar 5.6 Tampilan Jika Data Berhasil Ditambah atau Diedit .....    | 73 |
| Gambar 5.7 Tampilan Jika Data Berhasil Dihapus.....                  | 73 |
| Gambar 5.8 Form Matapelajaran .....                                  | 74 |
| Gambar 5.9 Form Nomor Ujian .....                                    | 74 |
| Gambar 5.10 Browse Photo Siswa .....                                 | 75 |
| Gambar 5.11 Tampilan Photo Siswa Pada Form Nomor Ujian .....         | 76 |
| Gambar 5.12 Form Kunci Jawaban .....                                 | 77 |
| Gambar 5.13 Tampilan Form Untuk Memulai Input Kunci .....            | 77 |
| Gambar 5.14 Input Kunci Jawaban Pada Table Grid .....                | 78 |
| Gambar 5.15 Form Koreksi .....                                       | 79 |
| Gambar 5.16 Scan LJK (Pengujian ke-1) .....                          | 80 |
| Gambar 5.17 Deteksi Tepi Sobel (Pengujian ke-1) .....                | 81 |
| Gambar 5.18 Deteksi LJK (Pengujian ke-1) .....                       | 83 |
| Gambar 5.19 LJK Dengan Nomor Ujian 010793682 (Pengujian ke-1) .....  | 83 |
| Gambar 5.20 Deteksi Jawaban (Pengujian ke-1) .....                   | 84 |
| Gambar 5.21 Menyimpan Data LJK ke Database .....                     | 84 |
| Gambar 5.22 Koreksi LJK .....  | 85 |
| Gambar 5.23 Scan LJK (Pengujian ke-2) .....                          | 87 |
| Gambar 5.24 Deteksi Tepi Sobel (Pengujian ke-2) .....                | 87 |
| Gambar 5.25 Deteksi LJK (Pengujian ke-2) .....                       | 88 |
| Gambar 5.26 LJK Dengan Nomor Ujian 013196941 (Pengujian ke-2) .....  | 88 |
| Gambar 5.27 Deteksi Jawaban (Pengujian ke-2) .....                   | 89 |
| Gambar 5.28 Scan LJK (Pengujian ke-3) .....                          | 90 |
| Gambar 5.29 Deteksi Tepi Sobel (Pengujian ke-3) .....                | 90 |
| Gambar 5.30 Deteksi LJK (Pengujian ke-3) .....                       | 91 |
| Gambar 5.31 LJK Dengan Nomor Ujian 0185451730 (Pengujian ke-3) ..... | 91 |
| Gambar 5.32 Deteksi Jawaban (Pengujian ke-3) .....                   | 92 |
| Gambar 5.33 Input Data untuk Tampil Data Kunci dan Jawaban .....     | 93 |
| Gambar 5.34 Skor Nilai Hasil Koreksi .....                           | 93 |
| Gambar 5.35 Jumlah Nilai dan Nilai Akhir .....                       | 93 |
| Gambar 5.36 Form Report Matapelajaran .....                          | 95 |
| Gambar 5.37 Report Matapelajaran .....                               | 95 |

|   |     |
|---|-----|
| Gambar 5.38 Form Report Nomor Ujian .....                           | 96  |
| Gambar 5.39 Cetak Semua Data Nomor Ujian .....                      | 97  |
| Gambar 5.40 Cetak Semua Kartu Ujian .....                           | 98  |
| Gambar 5.41 Form Report Nilai .....                                 | 99  |
| Gambar 5.42 Cetak Nilai Berdasarkan Matapelajaran .....             | 100 |
| Gambar 5.43 Cetak Nilai Siswa Seluruh Matapelajaran .....           | 101 |
| Gambar 5.44 Login Berhasil .....                                    | 103 |
| Gambar 5.45 Login Gagal .....                                       | 103 |
| Gambar 5.46 Konfirmasi Data Sudah Ada .....                         | 104 |
| Gambar 5.47 Data Berhasil Disimpan .....                            | 104 |
| Gambar 5.48 Data Berhasil Dihapus .....                             | 105 |
| Gambar 5.49 Menyimpan Hasil Deteksi LJK .....                       | 105 |
| Gambar 5.50 Data Sudah Pernah Tersimpan di Tabel Jawaban .....      | 106 |
| Gambar 5.51 Tampil Data Jawaban dan Kunci Jawaban .....             | 106 |
| Gambar 5.52 Tampil Data Tidak Ada pada Tebl Kunci dan Jawaban ..... | 107 |

## DAFTAR TABEL

|   |     |
|---|-----|
| Tabel 3.1 Jadwal dan Waktu Pelaksanaan Penelitian .....     | 43  |
| Tabel 4.1 Business Process .....                            | 44  |
| Tabel 4.2 Analisis Sebab Akibat.....                        | 46  |
| Tabel 4.3 Tabel Pengguna.....                               | 59  |
| Tabel 4.4 Tabel Tahun Pelajaran .....                       | 59  |
| Tabel 4.5 Tabel Matapelajaran.....                          | 59  |
| Tabel 4.6 Tabel Kunci Jawaban.....                          | 59  |
| Tabel 4.7 Tabel Jawaban Siswa .....                         | 60  |
| Tabel 4.8 Tabel Nomor Ujian .....                           | 60  |
| Tabel 5.1 Tabel Pengujian Waktu Manajemen Input Data.....   | 107 |
| Tabel 5.2 Tabel Pengujian Waktu Manajemen Koreksi LJK ..... | 108 |
| Tabel 5.3 Tabel Kuesioner .....                             | 109 |
| Tabel 5.4 Hasil Pengujian Kelayakkan.....                   | 110 |

## DAFTAR LAMPIRAN

|   |     |
|---|-----|
| Lampiran A Listing Program .....                  | A-1 |
| Lampiran B Lembar Jawaban Komputer .....          | B-1 |
| Lampiran C Surat Keterangan Penelitian.....       | C-1 |
| Lampiran D Kuesioner Uji Kelayakan Aplikasi ..... | D-1 |
| Lampiran E Data Kuesioner Uji Kelayakkan .....    | E-1 |
| Lampiran F Pengujian Waktu .....                  | F-1 |



# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pendidikan merupakan sebuah proses kegiatan yang disengaja atas input peserta didik untuk menimbulkan suatu hasil yang diinginkan sesuai tujuan yang ditetapkan. Sebagai sebuah proses yang disengaja maka pendidikan harus dievaluasi hasilnya untuk melihat apakah hasil yang dicapai telah sesuai dengan tujuan yang diinginkan dan apakah proses yang dilakukan efektif untuk mencapai hasil yang diinginkan. Evaluasi artinya penilaian terhadap tingkat keberhasilan peserta didik mencapai tujuan yang telah ditetapkan dalam sebuah program (Widoyoko ,2009).

Evaluasi pendidikan yang komprehensif harus dilakukan terhadap seluruh komponen dan sistem kerjanya. Menurut Widyoko (2009) “Evaluasi pendidikan adalah kegiatan atau proses penentuan nilai pendidikan, sehingga dapat diketahui mutu atau hasil-hasilnya” . Pendidikan melibatkan peserta didik, guru, metode, tujuan, kurikulum, media, sarana, kepala sekolah, pemerintah, masyarakat, pengguna lulusan, lingkungan fisik, manusia dan sebagainya. Oleh karenanya evaluasi pendidikan dilakukan atas komponen-komponen pendidikan tersebut. Evaluasi yang komprehensif menghasilkan informasi yang lengkap sebagai dasar perbaikan dalam pendidikan.

Sesuai dengan Undang-Undang Nomor 20 Tahun 2003 Pasal 57 (ayat 1), evaluasi dalam pendidikan sangatlah penting sebagai pengendalian mutu pendidikan secara nasional sebagai bentuk akuntabilitas penyelenggara pendidikan kepada pihak-pihak yang berkepentingan. Hasil yang diperoleh dari evaluasi dapat dijadikan umpan balik bagi guru dalam memperbaiki dan menyempurnakan program dan kegiatan pembelajaran. Undang-undang Nomor 20 Tahun 2003 Pasal 57 (ayat 2) menyatakan bahwa evaluasi dilakukan terhadap peserta didik, lembaga dan program pendidikan jalur formal dan nonformal untuk semua jenjang, satuan dan jenis pendidikan. Penilaian hasil belajar peserta didik harus dilakukan secara berkesinambungan untuk memantau proses, kemajuan, dan perbaikan hasil dalam bentuk *Try Out* Ujian Nasional (UN) yang biasanya dilakukan dengan melakukan pengisian pada Lembar Jawaban Komputer.

Disamping itu Software Pemeriksa Ujian diperlukan oleh sekolah mulai dari tingkat sekolah dasar hingga sekolah menengah. Kebutuhan tersebut tidak terlepas dari keinginan pihak sekolah dalam menyiapkan anak didiknya untuk menghadapi Ujian Nasional. Sudah bukan rahasia lagi bahwa Ujian Nasional telah menjadi momok tersendiri bagi peserta didik karena masih tingginya tingkat ketidaklulusan. Oleh karena itulah diperlukan latihan dalam mengisi lembar jawaban komputer. Akan tetapi mengingat mahalnya harga dari alat scan yang khusus digunakan untuk memeriksa LJK tersebut yang mencapai harga belasan hingga puluhan juta rupiah, tentu hal tersebut sangat memberatkan pihak sekolah untuk membelinya, terlebih lagi jika sekolah

tersebut tidak mendapatkan bantuan dari pemerintah. Salah satu cara untuk menghemat biaya pembelian alat scan tersebut adalah dengan menggunakan alat scan biasa yang telah terpasang pada printer. Kemudian hasil scan dengan menggunakan alat scan printer biasa tersebut diolah menggunakan algoritma sobel, sehingga nantinya akan didapatkan titik-titik yang memiliki piksel berwarna hitam sebagai jawaban pada LJK tersebut.

Algoritma sobel *edge detection* merupakan metode *edge detection* yang termasuk dalam gradient, *edge detector*. Piksel dari sebuah gambar yang dilakukan pendeteksian batas (*edge*) akan menjadi sebuah *edge* jika piksel tersebut melewati batas (*threshold*) tertentu. Sehingga apabila *threshold* telah ditetapkan maka nilai dari setiap piksel dapat kita tentukan apakah piksel tersebut merupakan batas (*edge*) atau bukan. Pendeteksian batas (*edge*) dari sebuah gambar dilakukan dengan melakukan operasi matrik sobel terhadap gambar yang akan dilakukan pendeteksian gambar. Matrik sobel merupakan matrik yang berukuran 3 x 3. (Murdianto, 2007)

Berdasarkan penjelasan diatas penulis berkeinginan untuk membangun sebuah Aplikasi pengolahan citra digital untuk pendeteksi jawaban pada lembar jawaban komputer menggunakan algoritma sobel.

## **1.2. Perumusan Masalah**

Berdasarkan latar belakang dapat dirumuskan permasalahan sebagai berikut:

Bagaimana membangun aplikasi pengolahan citra digital untuk pendeteksi jawaban pada lembar jawaban komputer menggunakan Algoritma sobel ?

### **1.3. Batasan Masalah**

Batasan masalah dalam penelitian ini sebagai berikut:

1. Pengolahan Citra Digital pada pendeteksian Lembar Jawaban Komputer menggunakan alat Scan biasa yang terpasang pada printer Canon MP-237.
2. LJK yang digunakan adalah LJK yang dicetak menggunakan aplikasi yang telah dibangun penulis.
3. LJK yang akan di-scan harus dalam keadaan baik, dan tidak rusak.
4. LJK yang akan di scan harus diletakkan secara rapi pada alat scan.
5. LJK yang digunakan merupakan LJK yang dibuat dari penulis.
6. Deteksi jawaban LJK yang terbaca pada salah satu pilihan jawaban yang diarsir, dan jawaban yang disilang tidak akan terdeteksi pada sistem.

### **1.4. Tujuan Penelitian**

Tujuan yang ingin dicapai dalam penelitian ini adalah merancang dan membangun “Aplikasi Pengolahan Citra Pada Pendeteksi Lembar Jawaban Komputer menggunakan Algoritma Sobel” yang memiliki fungsi utama sebagai berikut :

1. Menghasilkan aplikasi yang dapat mendeteksi jawaban siswa pada lembar LJK.

2. Meminimalisir pengeluaran biaya untuk membeli alat scan yang khusus untuk memeriksa jawaban pada lembar LJK.
3. Sekolah dapat secara berkala menyelenggarakan *Try Out* pada siswanya.
4. Sekolah dapat menampilkan nilai ujian *Try Out* secara otomatis menggunakan report.

### **1.5. Manfaat Penelitian**

Manfaat dari penelitian ini adalah:

- 1) Bagi penulis, dapat mengimplementasikan ilmu yang diperoleh selama di bangku perkuliahan.
- 2) Bagi pengguna perangkat lunak, dapat memberikan kemudahan dalam men-scan LJK tanpa harus membeli alat *Scanner khusus*. Aplikasi ini dapat digunakan untuk mendeteksi jawaban siswa pada LJK dan memeriksa kebenaran jawaban tersebut dan melatih siswa agar terbiasa mengisi LJK, sehingga tidak canggung lagi menghadapi ujian nasional.

### **1.6. Sistematika Penulisan**

Dalam penyusunan laporan penelitian, sistematika penulisan dibagi menjadi enam bab. Keenam bab tersebut antara lain pendahuluan, landasan teori, metode penelitian, analisis dan desain perangkat lunak, hasil pembahasan dan penutup.

## **BAB I PENDAHULUAN**

Bab ini berisi tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan, Manfaat dan Sistematika Penulisan Laporan.

## **BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan teori-teori secara garis besar yang berhubungan dengan penelitian, guna untuk memahami permasalahan yang berkaitan dengan aplikasi yang akan dibangun.

## **BAB III METODE PENELITIAN**

Pada bab ini berisi tentang metode-metode yang digunakan dalam penelitian, seperti jenis penelitian, teknik pengumpulan data, jenis dan sumber data, metode pengembangan sistem, metode pengujian dan jadwal penelitian.

## **BAB IV ANALISIS DAN DESAIN PERANGKAT LUNAK**

Bab ini menjelaskan setiap tahapan analisis dan perancangan sistem aplikasi yang akan dibangun dalam penelitian meliputi analisis sistem dan perancangan sistem.

## **BAB V HASIL DAN PEMBAHASAN**

Bab ini merupakan bab yang berisi hasil dan pembahasan yang menguraikan hasil perancangan sistem dan implementasinya.

## **BAB VI PENUTUP**

Bab ini merupakan bab penutup yang merupakan bab terakhir yang berisi kesimpulan dan saran dari pembuatan tugas akhir sampai ke pengembangan perangkat lunak kedepannya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Lembar Jawaban Komputer**

Di Indonesia, formulir isian berupa kertas yang akan diolah dengan komputer dan pemindai lebih dikenal sebagai Lembar Jawaban Komputer (LJK). LJK biasanya digunakan untuk ujian berbentuk pilihan ganda, kuesioner, formulir registrasi dan pendataan. Penggunaan LJK sebagai pengganti entri data secara manual dapat mempercepat pengolahan data. Kecepatan tersebut juga sangat ditentukan oleh kecepatan pemindai kecepatan perangkat lunak yang digunakan.

Penamaan LJK dapat menimbulkan pengertian yang kurang tepat atau ambiguitas. LJK bukanlah lembar yang berisi jawaban komputer, tetapi LJK adalah lembar jawaban yang akan diperiksa menggunakan alat bantu berupa komputer. Bila dibandingkan dengan sistem pengisian ujian/ registrasi/ pendataan on-line pada ratusan hingga jutaan peserta, penggunaan LJK masih lebih optimal karena dapat menghemat kebutuhan penyediaan komputer. Dengan sistem online harus tersedia 1 komputer untuk 1 responden atau peserta ujian, sedangkan untuk sistem offline, cukup 1 LJK untuk 1 responden atau peserta ujian.

Namun untuk pengoreksian lembar jawaban komputer pada system try-out UAN, pengoreksian masih dilakukan manual yaitu dengan mencocokkan

mistar pembulat secara manual yang berupa kunci jawaban. Oleh karena itu penulis merancang sebuah aplikasi pengolahan citra digital untuk mendeteksi jawaban pada Lembar Jawaban Komputer (Widoyoko, 2009).

## **2.2. Pengolahan Citra Digital**

Data atau informasi tidak hanya disajikan dalam bentuk teks, tetapi juga dapat berupa gambar, audio (bunyi, suara, musik), dan video. Keempat macam data atau informasi ini sering disebut multimedia. Era teknologi informasi saat ini tidak dapat dipisahkan dari multimedia. Situs web (*website*) di Internet dibuat semenarik mungkin dengan menyertakan visualisasi berupa gambar atau video yang dapat diputar. Beberapa waktu lalu istilah SMS (*Short Message Service*) begitu populer bagi pengguna telepon genggam (*handphone* atau HP). Tetapi, saat ini orang tidak hanya dapat mengirim pesan dalam bentuk teks, tetapi juga dapat mengirim pesan berupa gambar maupun video, yang dikenal dengan layanan MMS (*Multimedia Message Service*).

Citra (*image*) 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. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (*a picture is more than a thousand words*). Maksudnya tentu sebuah gambar dapat memberikan informasi yang lebih



banyak daripada informasi tersebut disajikan dalam bentuk kata-kata (tekstual) (Apriyana dkk, 2011).

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

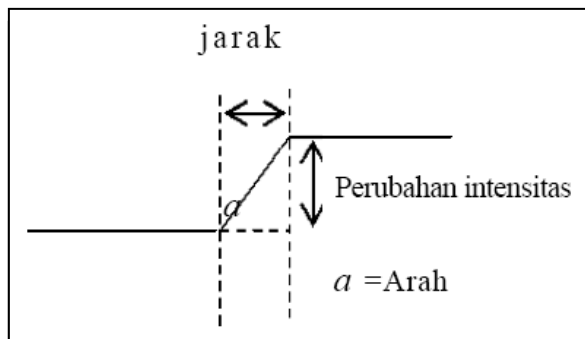
1. optik berupa foto,
2. analog berupa sinyal video seperti gambar pada monitor televisi,
3. digital yang dapat langsung disimpan pada suatu pita magnetik.

Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (*degradasi*), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasi baik oleh manusia maupun mesin, maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (Muthukrishnan dkk, 2011).

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra, kemudian elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur, sebagian citra perlu digabung dengan bagian citra yang lain (Parkesit, 2008).

### 2.3. Deteksi Tepi (Edge Detection)

Menurut Parkesit (2010) Tepi adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang memperlihatkan rincian pada gambar. Tepi dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda, tergantung pada perubahan intensitas. Deteksi tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra.

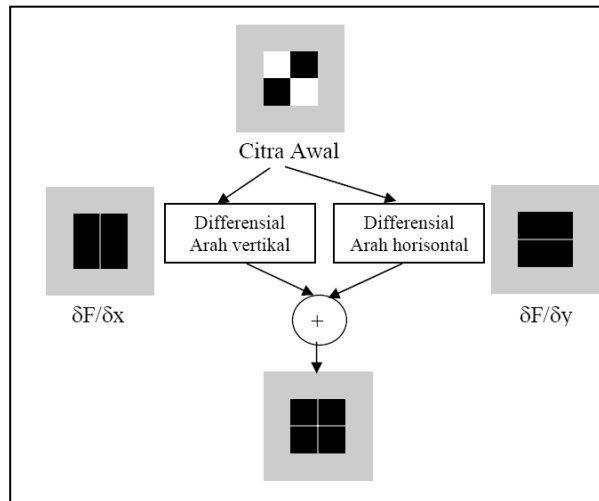


Gambar 2.1 Model Tepi Satu Dimensi  
(sumber : Parkesit, 2010)

Deteksi tepi pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah :

1. Untuk menandai bagian yang menjadi detail citra
2. Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra

Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Gambar 2.2 berikut ini menggambarkan bagaimana tepi suatu gambar diperoleh.



Gambar 2.2 Proses Deteksi Tepi Citra  
(sumber : Yunus, 2009)

Pada proses pencarian edge dilakukan dengan menggunakan teknik spatial filter (proses konvolusi). Berikut ini merupakan cara kerja spatial filter:

1. Lakukan penelusuran terhadap semua titik pada citra.
2. Rekam titik yang sedang diperiksa dan juga titik sekitarnya ke dalam dan juga titik sekitarnya ke dalam matrix neighbor

3. Isi matrix mask dengan angka (angka yang harus dimasukkan ditentukan oleh jenis operasi).
4. Kalikan matrix neighbor dengan matrix mask secara sekalar ( $Output[x,y] = Mask[x,y] * Neighbor[x,y]$ ).
5. Jumlahkan seluruh isi sel dari matrix output. Hasil penjumlahan ini adalah titik baru yang akan diletakkan pada koordinat titik yang sedang diproses.

Berdasarkan prinsip-prinsip filter pada citra maka tepi suatu gambar dapat diperoleh menggunakan High Pass Filter (HPF), yang mempunyai karakteristik:

$$\sum_y \cdot \sum_x G = 0$$

Contoh:

Diketahui fungsi citra  $f(x,y)$  sebagai berikut:

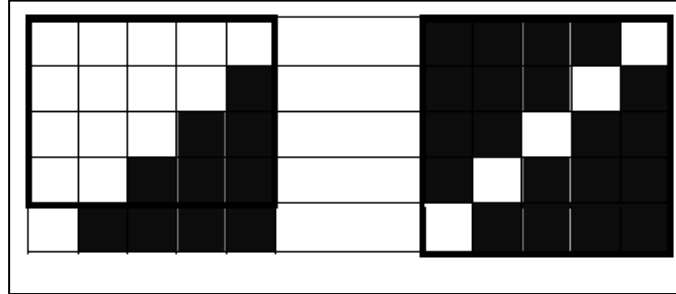
|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |

Dengan menggunakan filter :  $G(x,y) = (-1,1)$

Maka Hasil filter adalah :

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |

Bila digambarkan maka proses filter di atas mempunyai masukan dan keluaran sebagai berikut



Gambar 2.3 Proses Filter  
(sumber : Yunus, 2009)

Algoritma yang sering digunakan untuk mencari deteksi tepi, yaitu :

1. Robert
2. Prewit
3. Sobel
4. Frei-Chan

#### 2.4. Algoritma 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. Secara teknis operator Sobel merupakan operator diferensiasi diskrit, menghitung tafsiran gradien dari fungsi intensitas citra. Perhitungan dilakukan pada tiap point, memberikan kemungkinan arah pertambahan yang paling besar dari terang ke gelap dan rata-rata perubahan dalam arah tersebut. Secara otomatis gradien dari

fungsi intensitas citra ialah pada tiap image point sebuah vektor 2D dengan komponen yang diberikan dengan derivatif dalam arah vertikal dan horizontal (Fahmi, 2011).

Operator ini menggunakan dua 3x3 kernels yang dikonvolusi dengan citra asli untuk menghitung tafsiran dari derivatif pertama untuk perubahan horizontal dan satu untuk vertikal. Jika kita mendefinisikan A sebagai citra sumber dan  $G_x$  &  $G_y$  adalah dua citra yang tiap pada pointnya berisi tafsiran derivatif vertikal dan horizontal.

|                      |   |    |                      |    |    |
|----------------------|---|----|----------------------|----|----|
| -1                   | 0 | +1 | +1                   | +2 | +1 |
| -2                   | 0 | +2 | 0                    | 0  | 0  |
| -1                   | 0 | +1 | -1                   | -2 | -1 |
| <b>G<sub>x</sub></b> |   |    | <b>G<sub>y</sub></b> |    |    |

Gambar 2.4 Mask Sobel  
(sumber : Yunus, 2009)

Pada gambar diatas merupakan sepasang matrik sobel yang digunakan untuk mendeteksi batas dari sebuah gambar. Untuk melakukan deteksi batas gambar (edge), dilakukan dengan melakukan perkalian terhadap matriks tersebut secara terpisah. Matriks  $G_x$  berfungsi untuk memperkirakan gradient pada sumbu x, sedangkan matrik  $G_y$  berfungsi untuk memperkirakan gradient pada sumbu y.

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

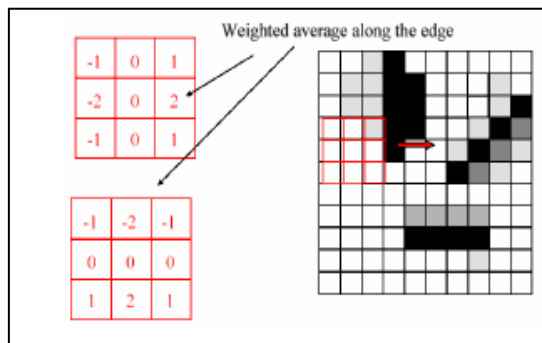
Operasi perkalian matriks sobel berbeda dengan operasi perkalian matriks pada umumnya. Pada dasarnya metode ini digunakan untuk memperkecil nilai piksel dari sebuah gambar yang bukan merupakan batas tepi, serta memperbesar nilai piksel dari bagian gambar yang mengandung batas tepi (Vairalkar dkk, 2012).

Pada tiap point dalam image, hasil tafsiran gradien dapat dikombinasikan untuk memberikan gradien besar menggunakan:

$$G = \sqrt{Gx^2 + Gy^2} \dots\dots\dots (2.1)$$

Dengan data ini kita dapat menghitung arah gradien:

$$\theta = \arctan \frac{Gy}{Gx} \dots\dots\dots (2.2)$$



Gambar 2.5 Arah Gradien  
(sumber : Fahmi, 2007)

## 2.5. Model Pengembangan *Waterfall*

Model ini mengusulkan sebuah pendekatan kepada perkembangan *software* yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan (Pressman, 2002)

Model sekuensial *waterfall* melingkupi aktivitas-aktivitas sebagai berikut :

1. Rekayasa dan permodelan sistem. Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen yang lain seperti *software*, manusia, dan *database*. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta disain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.
2. Analisis kebutuhan *software*. Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada *software*. Untuk memahami sifat program yang dibangun, analis harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan. Kebutuhan baik untuk sistem maupun *software* didokumentasikan dan dilihat lagi dengan pelanggan.
3. Desain. Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, yaitu struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.



4. Pengkodean. Desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.
5. Pengujian. Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika *internal software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.
6. Pemeliharaan. *Software* akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena *software* harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan *software* mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

## **2.6. Unified Modelling Language**

Model perangkat lunak dapat dianalogikan seperti pembuatan *blueprint* pada pembangunan gedung. Menurut Huda dan Nugroho (2010:137), UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk

visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Pada saat ini telah banyak versi UML yang digunakan dalam tahap analisis dan perancangan sistem. Versi yang terbaru saat ini adalah UML versi 2.5, namun versi yang telah diterima oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object oriented* dan *software component* adalah UML versi 2.0. (Dennis *et al.*, 2005:30). UML versi 2.0 tersebut terdiri dari empat belas teknik diagram, seperti yang dinyatakan oleh Dennis *et al.* (2005:30):

The Version 2.0 of the UML defines a set of fourteen diagramming techniques used to model a system. The diagrams are broken into two major groupings: one for modeling structure of a system and one for modeling behavior. The structure modeling diagrams include class, object, package, deployment, component, and composite structure diagrams. The behavior modeling diagrams include activity, sequence, communication, interaction overview, behavior state machine, protocol state machine, and use case diagrams.

Berdasarkan pernyataan Dennis *et al.* tersebut, UML versi 2.0 mendefinisikan empat belas diagram UML yang dapat digunakan untuk memodelkan suatu sistem. Diagram yang digunakan dibagi menjadi dua kelompok utama: satu untuk pemodelan struktur suatu sistem dan satu lagi untuk pemodelan sifat suatu sistem. *structure diagrams* terdiri dari *class diagram*, *object diagram*, *package diagram*, *deployment diagram*, *component diagram*, dan *composite structure diagram*. Sedangkan *behavior diagrams* terdiri dari *activity diagram*, *sequence diagram*, *communication diagram*, *interaction overview diagram*, *behavior state*

*machine diagram, protocol state machine diagram, dan use case diagram.*

### **2.6.1. Struktur Diagram**

Menurut Dennis *et al.* (2005:30) “Structure diagrams provide a way for representing the data and static relationships that are in an information system”. Dari pengertian tersebut *structure diagrams* menyediakan cara untuk mewakili data dan hubungan statis yang berada dalam sistem informasi. Berikut adalah diagram-diagram yang terdapat dalam *structure diagram*:

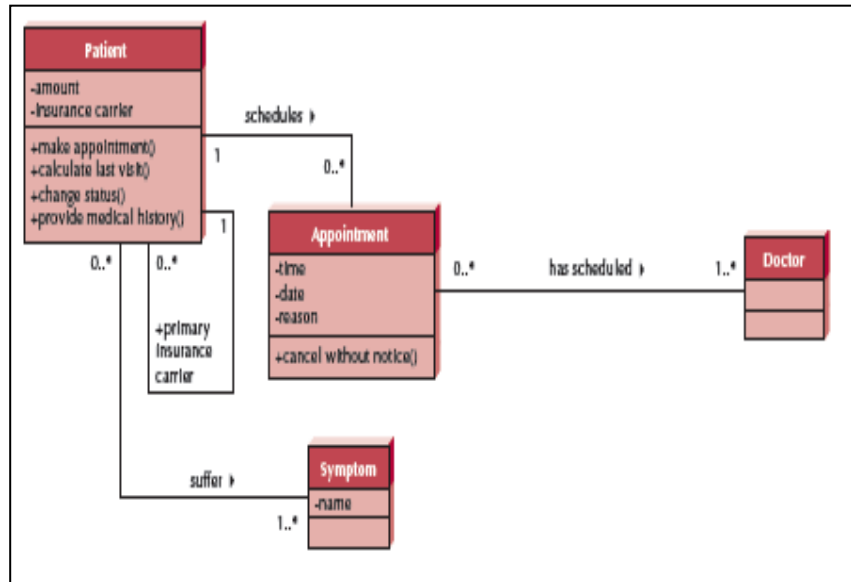
#### 1. *Class diagram*

Dennis *et al.* (2005:31) menyatakan bahwa:

The primary purpose of the class diagram is to create a vocabulary that is used by both the analyst and users. Class diagrams typically represent the things, ideas or concepts that are contained in the application. The class diagram would also portray the relationships among classes.

Berdasarkan pernyataan diatas dapat disimpulkan bahwa *class diagram* adalah kumpulan kelas-kelas objek serta penggambaran hubungannya, yang bertujuan untuk menciptakan sebuah kosa kata yang digunakan oleh analis dan pengguna.

Diagram kelas biasanya merupakan hal-hal, ide-ide atau konsep dalam aplikasi.



Gambar 2.6 Class Diagram  
(Sumber: Dennis et al, 2005: 217)

## 2. Deployment Diagram

Menurut Huda dan Nugroho (2010: 145), “*Deployment diagram* menggambarkan detail bagaimana komponen dibentuk dan didistribusikan dalam infrastruktur sistem”, hal tersebut hampir sama seperti pendapat Dennis et al (2005: 32) yang menyatakan bahwa:

Deployment diagrams are used to represent the relationships between the hardware components used in the physical infrastructure of an information system. They also can be used to represent the software components and how they are deployed over the physical architecture or infrastructure of an information system. In this case, a deployment diagram represents the environment for the execution of the software.

Berdasarkan pernyataan-pernyataan diatas dapat disimpulkan bahwa *deployment diagram* memodelkan bagian-bagian perangkat lunak suatu sistem ke perangkat keras yang akan mengeksekusinya

dan diagram ini juga digunakan untuk mewakili hubungan antara komponen-komponen hardware yang digunakan dalam infrastruktur fisik sistem informasi.



Gambar 2.7 *Deployment Diagram*  
(Sumber: Dennis *et al*, 2005: 413)

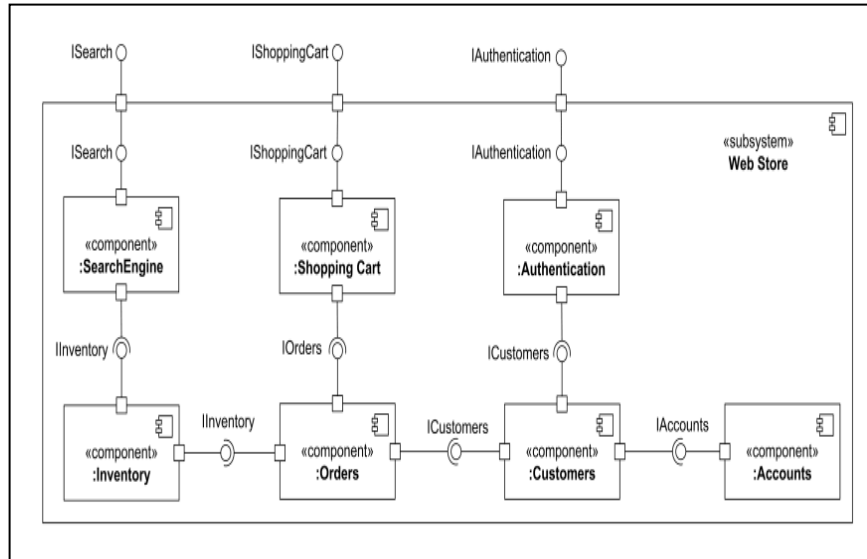
### 3. *Component Diagram*

Menurut Huda dan Nugroho (2010: 145), “*Component diagram* menggambarkan struktur dan hubungan antarkomponen perangkat lunak”. Menurut pendapat yang lain yang dinyatakan oleh Dennis *et al.* (2005:32):

Component diagrams allow the designer to model physical relationships among the physical modules of code. The diagram when combined with the deployment diagram can be used to portray the physical distribution of the software modules over a network.

Berdasarkan pernyataan-pernyataan diatas dapat disimpulkan bahwa *component diagram* memungkinkan desainer untuk memodelkan hubungan fisik antara modul fisik dari kode. Diagram ini bila dikombinasikan dengan *deployment diagram* dapat digunakan

untuk menggambarkan distribusi fisik dari modul perangkat lunak melalui jaringan.



Gambar 2.8 *Component Diagram*  
(Sumber: <http://www.uml-diagrams.org/>)

#### 4. *Composite Structure Diagram*

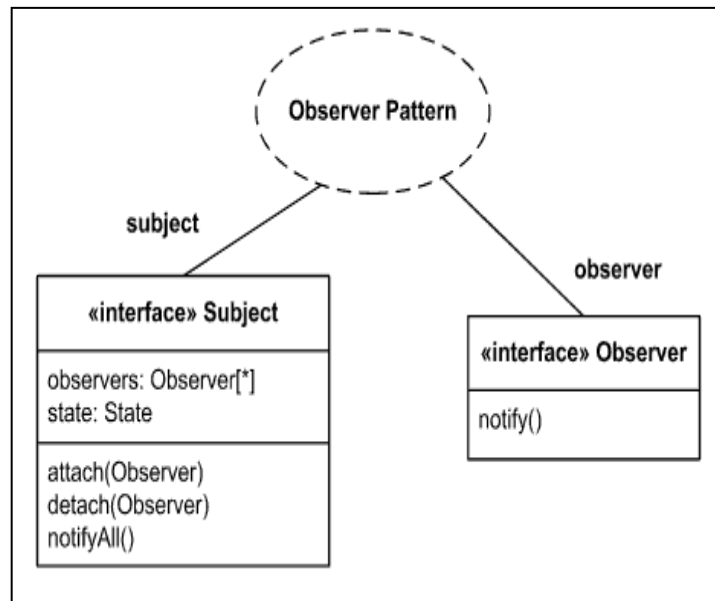
UML 2.0 menyediakan diagram baru ketika struktur internal dari kelas bersifat kompleks, diagram tersebut adalah *composite structure diagram*. Hal tersebut seperti yang dinyatakan Dennis *et al.* (2005:32),

“Composite structure diagrams are used to model the relationships among parts of a class”.

Di penjelasan selanjutnya Dennis *et al.* (2005:32) menjelaskan

“Composite structure diagrams also are useful when modeling the internal structure of a component for a component-based system”.

Berdasarkan pernyataan tersebut, *composite structure diagram* ini digunakan untuk menggambarkan hubungan antara bagian-bagian dari sebuah kelas. *Composite structure diagram* juga berguna ketika dilakukan pemodelan struktur internal komponen untuk sistem berbasis komponen.



Gambar 2.9 *Composite Structure Diagram*  
(Sumber: <http://www.uml-diagrams.org/>)

### 2.6.2. Behavior Diagram

Menurut Dennis *et al.* (2005:33) yang menyatakan bahwa:

Behavior modeling diagrams provide the analyst with a way to depict the dynamic relationships among the instances or objects that represent the business information system. They also allow the modeling of the dynamic behavior of individual objects throughout their lifetime. The behavior diagrams support the analyst in modeling the functional requirements of an evolving information system.

Berdasarkan pernyataan tersebut, dapat diketahui bahwa *behavior diagrams* menyediakan analisis cara menggambarkan hubungan dinamis antara instansi atau benda yang mewakili sistem informasi bisnis. *Behavior diagrams* juga memungkinkan pemodelan perilaku dinamis dari objek individu sepanjang masa aktifnya. *Behavior diagram* meliputi diagram-diagram berikut:

1. *Activity Diagram*

Menurut Dennis, *et al* (2005:33) yang menyatakan bahwa:

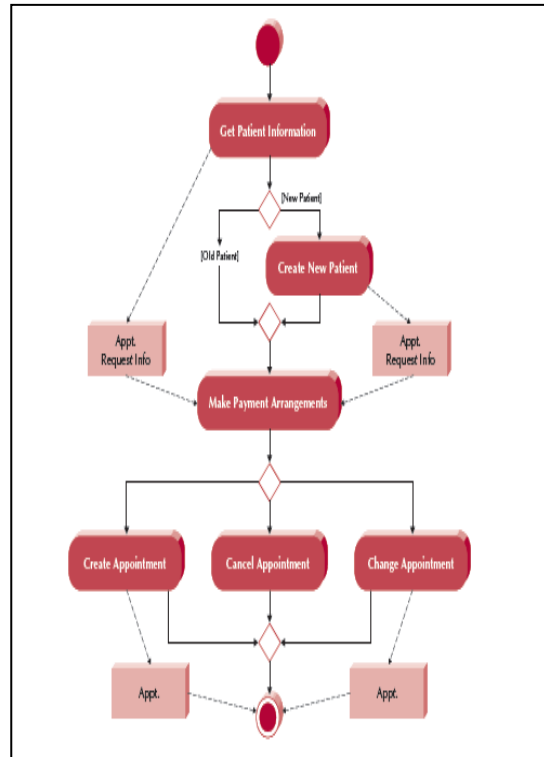
Activity diagrams provide the analyst with the ability to model processes in an information system. Activity diagrams can be used to model workflows, individual use cases, or the decision logic contained within an individual method. They also provide an approach to model parallel processes.

Pada penjelasan lainnya, menurut Huda dan Nugroho (2010: 142) menyatakan bahwa “*Activity diagram* menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram ini tidak menggambarkan proses internal sebuah sistem maupun interaksi antar-subsistem, tetapi lebih menggambarkan proses-proses secara umum”.

Berdasarkan pernyataan-pernyataan diatas dapat ditarik kesimpulan bahwa *activity diagram* menunjukkan aktivitas sistem dalam bentuk aksi-aksi dan seorang analis dapat menggunakan *activity diagram* untuk memodelkan proses dalam suatu sistem informasi. Diagram ini juga dapat digunakan untuk alur kerja model, *use case*



individual, atau logika keputusan yang terkandung dalam metode individual. Berikut ini gambar 2.12 pada diagram activity.



Gambar 2.10 Activity Diagram  
(Sumber: Dennis et al,2005: 167)

## 2. Interaction Diagram

Menurut Dennis et al. (2005:33)

“Interaction diagrams portray the interaction among the objects of an object-oriented information system”.

Berdasarkan pernyataan tersebut, dapat disimpulkan bahwa *interaction diagram* menggambarkan interaksi antara objek dari suatu sistem informasi berorientasi objek.

UML 2.0 menyediakan empat diagram interaksi yang berbeda: *sequence*, komunikasi, gambaran interaksi, dan diagram waktu.

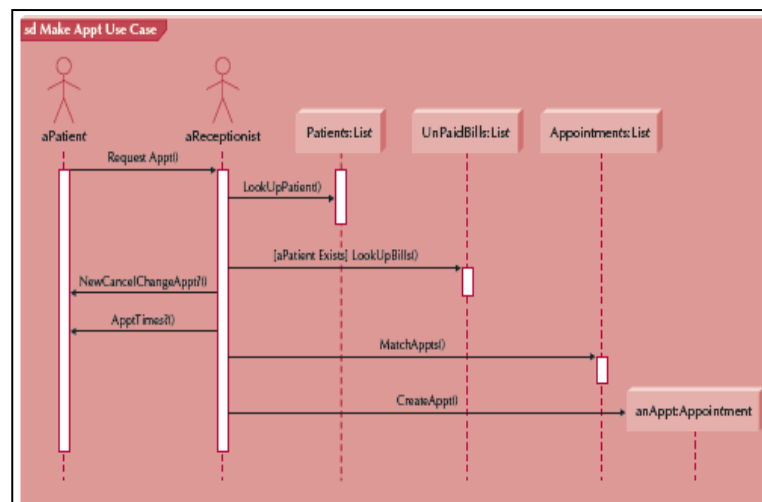
Berikut adalah penjelasan diagram-diagram yang terdapat dalam *interaction diagram*:

a. *Sequence Diagram*

Dennis *et al.* (2005:33) menyatakan:

Sequence diagrams allow an analyst to portray the dynamic interaction among objects in an information system. Sequence diagrams are by far the most common kind of interaction diagram used in object-oriented modeling. They emphasize the time-based ordering of the activity that takes place with a set of collaborating objects.

Berdasarkan pernyataan tersebut, *sequence diagram* digunakan untuk menggambarkan interaksi dinamis antara objek-objek dalam suatu sistem informasi. *Sequence diagram* merupakan jenis interaksi yang paling umum digunakan dalam pemodelan berorientasi objek. *Sequence diagram* lebih menekankan penyusunan berbasis waktu untuk kegiatan yang dilakukan dengan satu set dari objek yang berkolaborasi.



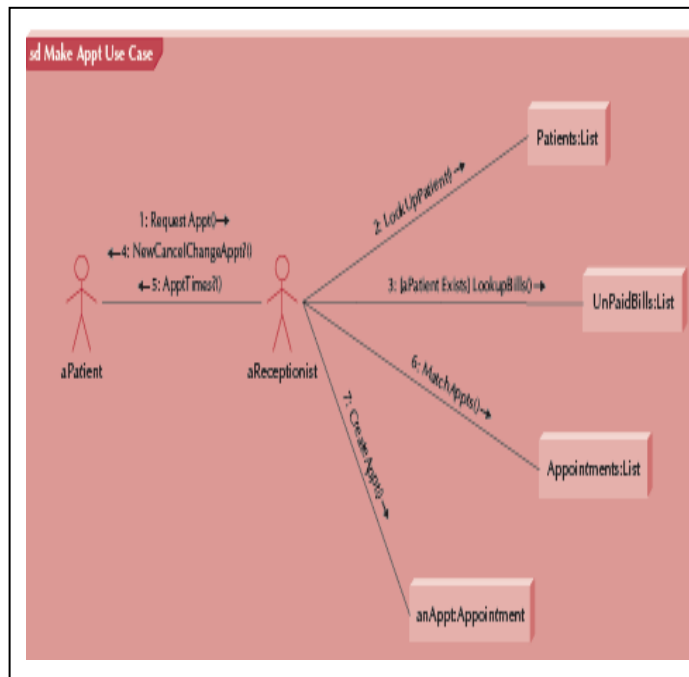
Gambar 2.11 *Sequence Diagram*  
(Sumber: Dennis *et al.*, 2005: 239)

b. *Communication Diagram*

Dennis *et al.* (2005:33) menyatakan:

Communication diagrams provide an alternative view of the dynamic interaction that takes place among the objects in an object-oriented information system. Where sequence diagrams emphasize the time-based ordering of an activity, communication diagrams focus on the set of messages that are passed within a set of collaborating objects. In other words, communication diagrams depict how objects collaborate to support some aspect of the required functionality of the system. The sequence or time-based ordering of the messages is shown through a sequence numbering scheme.

Berdasarkan pernyataan tersebut, dapat disimpulkan bahwa *communication diagram* memberikan pandangan alternatif dari interaksi dinamis yang terjadi di antara objek dalam suatu sistem informasi berorientasi objek. Jika *sequence diagram* lebih menekankan pada waktu dari suatu kegiatan, maka *Communication Diagram* lebih berfokus pada suatu set pesan yang disampaikan dalam sekumpulan objek yang berkolaborasi. Dengan kata lain, *Communication Diagram* menggambarkan bagaimana objek dapat berkolaborasi untuk mendukung beberapa aspek dari fungsi yang diperlukan suatu sistem. Urutan atau penyusunan berdasarkan waktu dari pesan ditunjukkan melalui skema urutan penomoran.



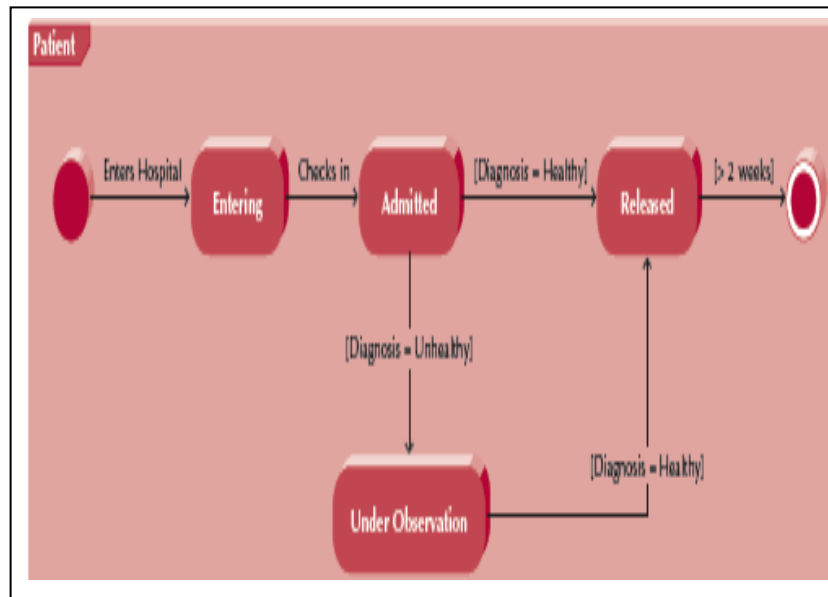
Gambar 2.12 *Communication Diagram*  
(Sumber: Dennis et al,2005: 239)

### 3. *State Machines Diagram*

Dennis *et al.* (2005:34) menyatakan bahwa:

In UML 2.0, there are two different state machines: behavior and protocol. Behavior state machines are used to describe the changes that an object may go through during its lifetime. Protocol state machines portray a specified sequence of events to which an object may respond.

Menurut Dennis *et al.* (2005:34) “Behavior state machines provide a method for modeling the different states, or sets of values, that instances of a class may go through during their lifetime”. Berdasarkan pernyataan tersebut, *behavior state machine diagram* menyediakan metode untuk menggambarkan perbedaan dari suatu keadaan yang terjadi.



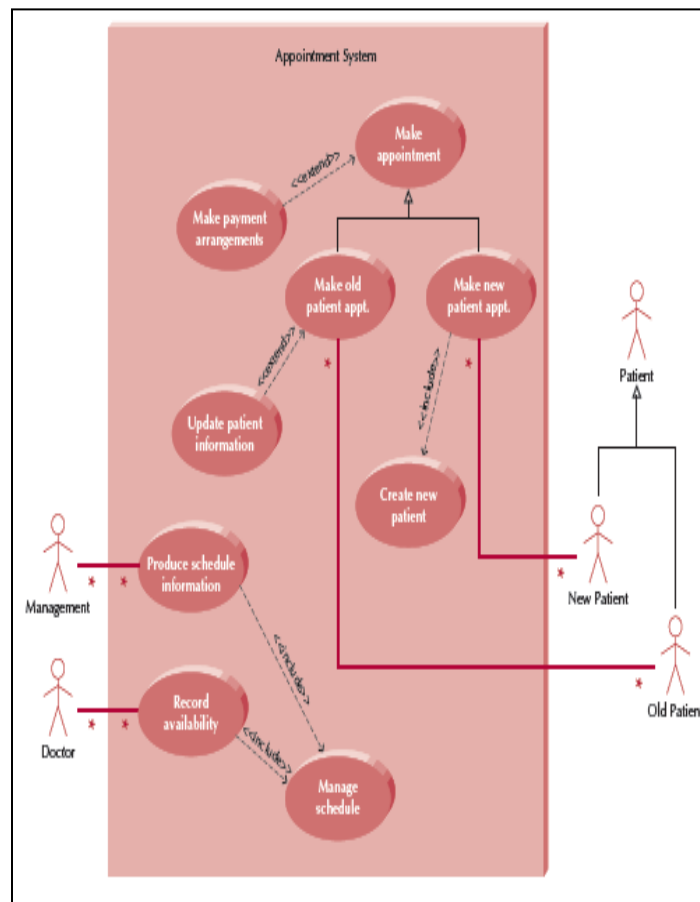
Gambar 2.13 Behavior State Machine Diagram  
(Sumber: Dennis et al, 2005: 251)

#### 4. Use Case Diagram

Menurut Huda dan Nugroho (2010:138) yang menyatakan bahwa “Use Case diagram menggambarkan fungsionalitas dari sebuah sistem yang mempresentasikan sebuah interaksi antara aktor dan sistem”. Hal tersebut hampir sama seperti Dennis *et al.* (2005:34) yang menyatakan:

Use case diagrams allow the analyst to model the interaction of an information system and its environment. The environment of an information system includes both the end user and any external system that interacts with the information system. The primary use of the use case diagram is to provide a means to document and understand the requirements of the evolving information system. Use cases and use case diagrams are some of the most important tools that are used in object-oriented systems analysis and design.

Berdasarkan pernyataan-pernyataan tersebut, *use case diagram* memungkinkan analisis untuk memodelkan interaksi antara sistem informasi dan lingkungannya. Lingkungan sistem informasi mencakup pengguna akhir dan setiap sistem eksternal yang berinteraksi dengan sistem informasi. Penggunaan utama dari *use case diagram* adalah untuk menyediakan sarana dalam mendokumentasikan dan memahami persyaratan sistem informasi yang sedang berkembang. *Use case* dan *diagram use case* adalah beberapa alat yang paling penting untuk digunakan dalam analisis dan desain sistem berorientasi objek.



Gambar 2.14 *Use Case Diagram*  
(Sumber: Dennis et al, 2005: 251)

## 2.7. Delphi 7

Delphi merupakan sebuah program berbasis windows yang memiliki bahasa mirip dengan pascal. Jadi bila anda telah menguasai pascal barangkali tidak akan kesulitan dalam menggunakan program ini. Perbedaan mendasar antara delphi dengan pascal adalah bahwa delphi memiliki oop (object orientation program) atau program yang berorientasi ke objek. Dengan demikian delphi lebih mudah digunakan dibandingkan dengan pascal, karena anda hanya tinggal mengatur properti atau sifat dari objek dan menambahkan sedikit pengkodean.

Delphi merupakan Lingkungan Pemrograman Terintegrasi (Integrate Development Environment / IDE). Delphi bukan bahasa pemrograman, tetapi perangkat lunak yang menyediakan seperangkat alat (tools) untuk membantu pemrogram dalam menulis program komputer. Lalu, bahasa apa yang digunakan oleh Delphi? Delphi menggunakan Object Pascal sebagai bahasa pemrogramannya. Object Pascal merupakan bahasa Pascal yang diberi tambahan kemampuan untuk menerapkan konsep-konsep OOP (Object Oriented Programming). Seluruh sintak Object Pascal menggunakan aturan yang ada di dalam Pascal, termasuk perintah-perintah dasar seperti *control structures*, *variables*, *array*, dan sebagainya.

Peralatan yang disediakan oleh Delphi memberikan kemudahan bagi pemrogram untuk membuat program secara visual (*visual programming*), *visual programming* adalah metoda dimana sebagian atau keseluruhan program dibuat dengan cara ‘menggambar’-kan tampilan / hasil akhir dan kemudian meminta

beberapa perangkat untuk membuat kode-kode program berdasarkan gambaran hasil akhir tersebut.

Karena program yang dibuat di dalam Delphi berjalan di dalam sistem operasi Windows maka kegiatan program dilakukan berdasarkan metoda *event-driven* programms. Event-driven programming adalah metoda mengeksekusi kode program berdasarkan pesan (*messages / events*) yang diberikan oleh pemakai ataupun oleh sistem operasi atau program lainnya.

Sebagai contoh : apabila pemakai menekan tombol kiri mouse dan kemudian melepaskannya dengan cepat (kita mengenal itu sebagai 'klik') maka tindakan tersebut akan membuat aplikasi menerima pesan 'MOUSE DOWN' yang disertai dengan informasi tombol mana yang ditekan dan lokasi kursor saat 'klik' dilakukan, tetapi apabila pemakai menekan tombol kiri mouse dan kemudian menggeser mouse tanpa melepaskan tombol kiri maka aplikasi akan menerima pesan 'MOUSE MOVE' (Hartono: 2008)

## **2.8. Basis Data**

Menurut Khadir (2008) Basis Data merupakan file yang berisikan tabel-tabel yang saling berinteraksi sehingga dapat diproses dan digunakan dengan cepat dan mudah. Tabel merupakan kumpulan data yang tersusun menurut aturan tertentu dan berhubungan dengan topik tertentu. Tabel diorganisasikan dalam dua bagian, bagian menurun atau kolom disebut dengan *field* dan bagian mendatar atau baris disebut dengan *record*. Berkas basis data



dibuktikan dengan adanya media penyimpanan fisik dengan adanya media penyimpanan fisik berupa *magnetic-tipe*, piringan (disket, CD).

Adapun Basis Data yang digunakan yaitu SQL SERVER yang merupakan sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Structured Query Language (SQL) adalah bahasa standar yang digunakan untuk memanipulasi dan memperoleh data dari sebuah database relasional. SQL mula-mula dibuat pada tahun 1970 dengan istilah SEQUEL, standarisasi yang pertama kali dibuat pada tahun 1986 oleh American National Standards Institute (ANSI), melalui publikasi Database Language SQL (ANSI X3. 136 – 1986) dan International Standards Organization (ISO) yang disebut sebagai SQL-86.

Microsoft SQL server 2008 adalah Sistem manajemen basis data yang memakai perintah Transact-SQL untuk mengirim perintah dari Komputer client ke Komputer server. Microsoft SQL Server 2008 dapat dijalankan pada windows 7. Sistem akan dipergunakan oleh beberapa komputer (client-server) maka database yang dibangun adalah merupakan database yang berfungsi untuk menunjang hal tersebut untuk itu dalam penulisan ini dipakai database Microsoft SQL Server 2008, selain itu database ini juga compatible dengan bahasa pemrograman yang digunakan. Secara teoritis, program SQL yang sudah terinstall pada komputer dapat menampung *32.767 database* dan terdapat lebih dari *2 billion objek*. (Kadhir, 2008).

## **BAB III**

### **METODE PENELITIAN**

#### **3.1. Tempat dan Waktu Penelitian**

Penelitian ini bertempat SMP Negeri 2 Kota Bengkulu. Penelitian ini dilakukan dengan mempelajari dokumen dan laporan yang berhubungan dengan sistem pemeriksaan LJK di SMPN2 di Kota Bengkulu. Sedangkan waktu penelitian dimulai sejak November 2013 hingga Februari 2014.

#### **3.2. Metode Pengambilan**

##### **3.2.1. Populasi**

Menurut Sugiyono (2011: 80), “Populasi adalah wilayah generalisasi yang terdiri atas obyek atau subyek yang mempunyai kuantitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulan”. Berdasarkan pengertian populasi tersebut, maka yang akan dijadikan populasi dalam penelitian ini adalah data isian Lembar Jawaban Komputer.

##### **3.2.2. Sample**

Dalam suatu penelitian diperlukan suatu sampel jika populasinya terlalu besar. Menurut Sugiyono (2011: 81), “Sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut”. Sampel

yang diambil harus mewakili karakteristik dari populasi yang sedang diteliti.

Pada penelitian ini digunakan teknik *Sampling Purposive*. Menurut Sugiyono (2011: 85) ”*Sampling Purposive* adalah teknik penentuan sampel dengan pertimbangan tertentu”. Dari uraian di atas, maka yang menjadi sampel dari penelitian ini adalah data isian LJK yang dimiliki SMP Negeri 2 Kota Bengkulu.

Sampel tersebut diambil berdasarkan pertimbangan bahwa LJK hasil *Try Out* 2013 merupakan data isian LJK terbaru yang dimiliki SMP Negeri 2 Kota Bengkulu, sehingga hal tersebut dapat memberikan gambaran terbaru pula bagi penulis untuk melakukan penelitiannya.

### **3.3. Metode Pengumpulan Data**

Data yang digunakan dalam penelitian ini merupakan data sekunder, yaitu data yang diperoleh dengan cara tidak langsung dari subjek atau objek yang diteliti, tetapi melalui bentuk literatur dan sampel yang nanti akan digunakan dalam pembuatan dan pengujian aplikasi.

Adapun teknik pengumpulan data yang digunakan dalam penelitian ini antara lain:

#### **1. Studi Pustaka**

Studi kepustakaan dilakukan dengan mengumpulkan data dan informasi yang digunakan sebagai acuan dalam pembuatan aplikasi pendeteksi

kemiripan dokumen teks. Data dan informasi dapat berupa buku-buku ilmiah, laporan penelitian, skripsi, jurnal dan sumber-sumber tertulis lainnya yang berhubungan dengan pemahaman algoritma Sobel, desain *Unified Modelling Language*, bahasa pemrograman *Delphi* dan basis data *SQL Server 2008*.

## 2. Observasi

Sugiyono (2006) dalam (Hasibuan, 2007:137) menyatakan bahwa observasi adalah dasar semua ilmu pengetahuan. Ilmuwan hanya dapat bekerja berdasarkan data, yaitu fakta mengenai dunia kenyataan yang diperoleh melalui observasi. Observasi yang dilakukan dalam penelitian ini adalah dengan mengumpulkan dokumen teks digital yang digunakan sebagai sampel dalam uji coba aplikasi. Sampel dapat berupa kumpulan Lembar Jawaban Komputer siswa yang akan dicocokkan dengan kunci jawabannya.

### **3.4. Metode Pengembangan Sistem**

Pengembangan Aplikasi Pengolahan Citra Pada Pendeteksi Lembar Jawaban Komputer dalam Tugas Akhir ini menggunakan model *waterfall*.

Adapun langkah-langkah yang dilakukan dalam pengembangan sistem ini secara garis besar adalah sebagai berikut:

#### 1. *Initial Investigation* (penelitian awal)

Penelitian awal dilakukan dengan mempelajari berbagai referensi sehingga didapat rumusan masalah, batasan, dan tujuan yang terdapat pada

Bab I serta pemilihan metode yang akan digunakan untuk membangun dan mengembangkan sistem.

## 2. *Requirements Definition* (definisi kebutuhan)

Pada tahap ini dilakukan pendefinisian kebutuhan yang diperlukan dalam proses perancangan sistem. Pendefinisian kebutuhan meliputi data yang diperlukan, kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*). Berikut perangkat keras dan perangkat lunak yang digunakan penulis dalam penelitian ini:

### a. Perangkat keras (*hardware*):

- 1) Satu unit laptop Compaq 510
- 2) Scanner Printer Canon MP230.

### b. Perangkat lunak (*software*):

- 1) Sistem operasi Windows 7.
- 2) Delphi 7 sebagai perangkat lunak pembuatan *listing* program dan *design interface*.
- 3) SQL server 2008 sebagai *database server*.
- 4) Microsoft Office Visio 2003

## 3. *System Design* (perancangan sistem)

Perancangan sistem meliputi perancangan *database*, pemodelan perangkat lunak menggunakan UML (*Unified Modelling Language*) dan perancangan antarmuka perangkat lunak.

## 4. *Coding and Testing* (pengkodean dan pengujian)

- a. Pengkodean perangkat lunak menggunakan bahasa pemrograman Pascal dengan editor Delphi dan pengkodean database menggunakan SQL Server 2008.
- b. Pengujian menggunakan *Black Box testing*, yang meliputi :
  1. Validasi, dilakukan dengan pengujian normal dan pengujian tidak normal, serta melakukan pengujian terhadap kebenaran hasil deteksi Jawaban dari LJK.
  2. Desain tes, dilakukan dengan cara pengujian individual untuk melihat apakah desain yang dihasilkan berfungsi sesuai dengan perancangan.
  3. *Interface*, dilakukan dengan mengoperasikan perangkat lunak dan dilihat apakah perangkat lunak yang dibangun sesuai dengan alur sistem.
  4. Database, dilakukan dengan cara mengakses database, apakah berhasil diakses atau tidak.
  5. Analisa kinerja sistem, analisa dilakukan dengan memasukkan data siswa pada form nomor ujian. Setelah itu LJK siswa di scan terlebih dahulu untuk memperoleh citra LJK, citra LJK selanjutnya di ubah kedalam bentuk deteksi tepi sobel dan dilakukan pendeteksian jawaban pada citra LJK. Hasil deteksi LJK tersebut akan disimpan kedalam database untuk dilakukan pengoreksian dengan syarat kode pelajaran dan paket soal pada data LJK harus sama dengan data pada kunci jawaban. Hasil

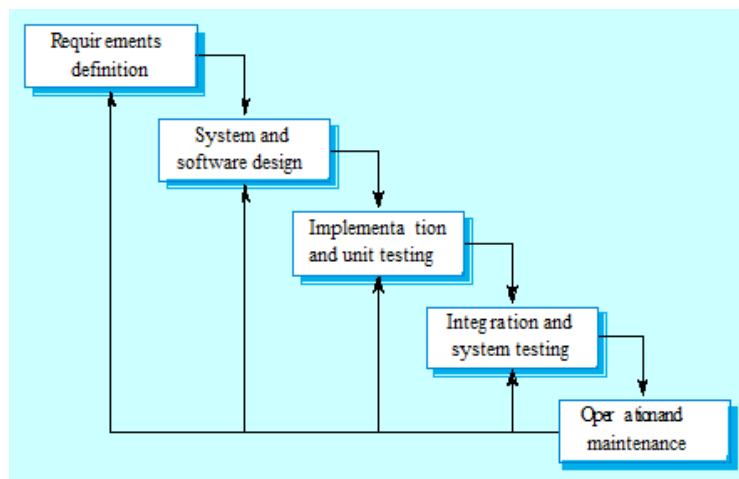
koreksi LJK tersebut akan ditampilkan skor nilai dan nilai akhir pada laporan nilai siswa.

#### 5. *Implementation* (implementasi)

Pada tahap ini yang perlu dilakukan adalah mempersiapkan perangkat keras yang telah dipilih penulis dan perangkat lunak yang telah dibangun penulis, sehingga sistem siap untuk dioperasikan.

#### 6. *Operation and Support* (pengoperasian dan dukungan)

- a. Pengoperasian dilakukan oleh guru SMPN 2 Kota Bengkulu dan kepala kurikulum yang bertugas mengurus pengoreksian LJK.
- b. *Support* atau dukungan agar aplikasi bisa digunakan dalam waktu panjang adalah dengan melakukan pemeliharaan dan pengembangan. penulis hanya sampai pada rekomendasi pemeliharaan dan pengembangan. Seperti rekomendasi untuk selalu melakukan *back-up* data setiap bulannya dan rekomendasi meningkatkan spesifikasi perangkat keras yang digunakan.



Gambar 3.1 Model Waterfall  
(sumber : Pressman, 2002)

### 3.5. Metode Pengujian Sistem

Menurut Pudjadi (2008: 7), “*black box testing* dilakukan tanpa pengetahuan detail tentang struktur internal dari sistem atau komponen yang dites”. *Black box testing* fokus pada kebutuhan fungsional perangkat lunak, hal tersebut berdasarkan spesifikasi kebutuhan sistem. *Black box testing* berusaha menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Menurut Ayuliana (2009: 1), uji coba *Black box testing* didesain untuk dapat menjawab pertanyaan-pertanyaan berikut:

- a. Bagaimana validitas fungsionalnya diuji?
- b. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik?
- c. Apakah sistem secara khusus sensitif terhadap nilai input tertentu?
- d. Bagaimana batasan-batasan kelas data diisolasi?
- e. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?
- f. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

*Black box testing* digunakan penulis untuk mendemonstrasikan fungsi perangkat lunak yang dioperasikan. Untuk menemukan kesalahan dan



menjawab semua pertanyaan-pertanyaan dari kriteria *black box testing*, maka spesifikasi yang akan diuji oleh penulis dalam penelitian ini adalah validasi, desain tes, *interface*, database dan kinerja sistem. Hal yang akan diuji pada percobaan *black-box* antara lain :

1. Pengujian terhadap antarmuka aplikasi
2. Pengujian subfungsi dari aplikasi yang meliputi pengujian algoritma Sobel
3. Pengujian aplikasi utuh dimana aplikasi utuh ini adalah gabungan subfungsi-subfungsi dan antar muka aplikasi.

### **3.6. Metode Uji Kelayakan Sistem**

Uji kelayakan dilakukan untuk mendapatkan penilaian langsung terhadap sistem yang dihasilkan. Target dari pengujian kelayakan sistem ini adalah responden (calon pemakai sistem). Adapun teknik dari uji kelayakan ini adalah :

#### **1. Kuesioner**

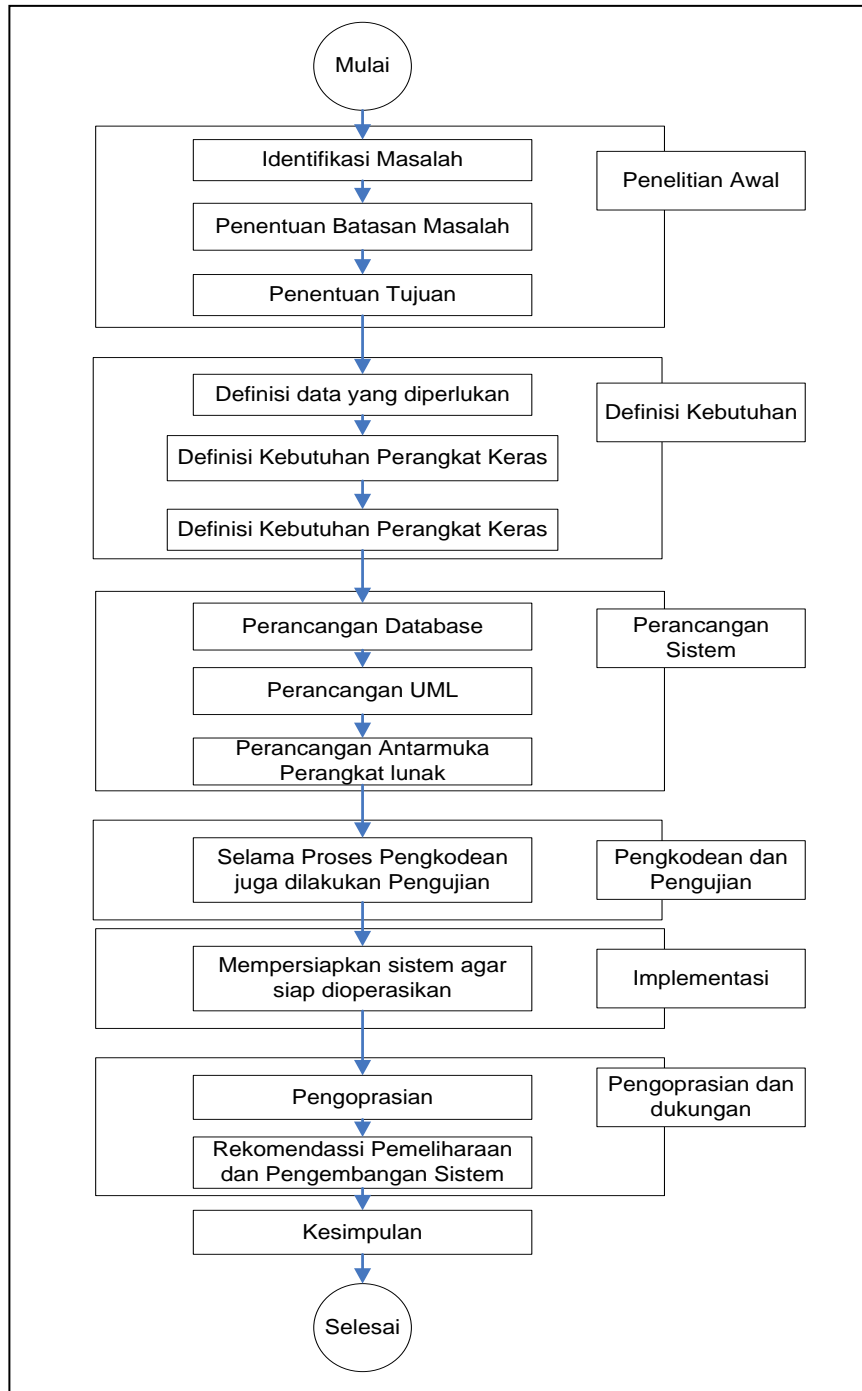
Kuesioner merupakan daftar pertanyaan yang diajukan pada seorang responden untuk mencari jawaban dari permasalahan yang diteliti. ciri.

#### **2. Tabulasi data**

Proses perhitungan data anget menggunakan skala *likert*. Skala *likert* adalah perhitungan skor pada tiap-tiap interval dari pernyataan yang diberikan ke responden.

### 3.7. Diagram Alir Penelitian

Tahapan-tahapan dalam model *Waterfall* ditunjukkan dalam diagram alir penelitian sebagai berikut:



Gambar 3.2 Diagram Alir Penelitian

### 3.8. Jadwal Penelitian

Berikut adalah jadwal dan waktu pelaksanaan penelitian dapat dilihat pada tabel 3.1 dibawah ini.

Tabel 3.1 Jadwal dan waktu pelaksanaan penelitian

| No | Kegiatan                   | Bulan |     |     |     |     |     |
|----|----------------------------|-------|-----|-----|-----|-----|-----|
|    |                            | Nov   | Des | Jan | Feb | Mar | Apr |
| 1  | Penelitian awal            | ■     |     |     |     |     |     |
| 2  | Defenisi kebutuhan         |       | ■   |     |     |     |     |
| 3  | Perancangan Sistem         |       | ■   | ■   |     |     |     |
| 4  | Pengkodean dan pengujian   |       | ■   | ■   | ■   |     |     |
| 5  | Implementasi               |       |     |     |     | ■   |     |
| 6  | Pengoperasian dan dukungan |       |     |     |     |     | ■   |
| 7  | Kesimpulan                 |       |     |     |     |     | ■   |