

**PEMANFAATAN KOMPATIBILITAS
DATABASE PASCAL KONVENSIONAL
DENGAN DATABASE DELPHI RELASIONAL
DALAM PERANCANGAN SISTEM INFORMASI PENDIDIKAN**

TUGAS AKHIR

Disusun untuk memenuhi salah satu syarat mencapai gelar sarjana S1
Teknik Informatika



Disusun Oleh :

Nama : FUAD RIDLA NUGRAHA A.M

No.Mhs : 95351008

Nirm : 950051013113120126

**TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2005

LEMBAR PENGESAHAN PEMBIMBING

PEMANFAATAN KOMPATIBILITAS DATABASE PASCAL KONVENSIONAL DENGAN DATABASE DELPHI RELASIONAL DALAM PERANCANGAN SISTEM INFORMASI PENDIDIKAN

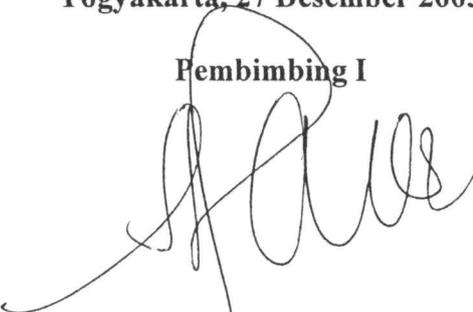
TUGAS AKHIR

Oleh :

Nama : FUAD RIDLA NUGRAHA A.M
NO. Mhs : 95 351 008
Nirm : 950051013113120126

Yogyakarta, 27 Desember 2005

Pembimbing I



SRI KUSUMA DEWI, S.Si, MT

**PEMANFAATAN KOMPATIBILITAS
DATA BASE PASCAL KONVENSIONAL
DENGAN DATA BASE DELPHI RELASIONAL
DALAM PERANCANGAN SISTEM INFORMASI PENDIDIKAN**

Oleh :

NAMA : FUAD RIDLA NUGRAHA A.M
NO. MHS : 95351008
NIRM : 950051013113120126



**Telah Dipertahankan di hadapan Panitia Penguji Skripsi
Jurusan Teknik Informatika
Universitas Islam Indonesia**

Tanggal :

Dosen penguji

1. SRI KUSUMA DEWI, S.Si,MT
2. YUDI PRAYUDI, S.Si,M.KOM
3. AMI FAUZIYAH, S.T,MT

Tanda Tangan

.....
.....
.....

Mengetahui,

**Dekan Fakultas Teknologi Industri
Universitas Islam Indonesia**



BACHRUN SUTRISNO, MSc

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Puji Syukur Alhamdulillah kami panjatkan kehadirat Allah SWT karena berkat Rahmat dan Hidayah-Nya sehingga kami dapat menyelesaikan Tugas Akhir yang berjudul Pemanfaatan Kompatibilitas Database Pascal Konvensional dengan DataBase Delphi Relasional Dalam Perancangan Sistem Informasi Pendidikan ini dengan Baik.

Penyusunan tugas akhir ini adalah salah satu syarat guna menyelesaikan jenjang kesarjanaan Strata 1 (S1) di Fakultas Teknologi Industrib Jurusan Teknik Informatika Universitas Islam Indonesia Yogyakarta.

Dalam Menyusum Tugas Akhir ini kami menyadari telah mendapatkan bantuan dan dorongan dari berbagai pihak. Untuk itu kami mengucapkan banyak terima kasih kepada :

1. Allah SWT, yang selalu memberikan memberikan Rahmat dan Rizkinya kepada seluruh Umat manusia.
2. Orang Tua kami yang selalu memberikan doa dan memotivasi untuk terus maju dan berjuang.
3. Ibu Sri Kusumadewi, Ssi, MT, Selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia Yogyakarta dan Selaku Dosen Pembimbing I.
4. Ibu Ami Fauzijah, ST, MT. Selaku Dosen Pembimbing II

5. Teman-teman angkatan 1995 Teknik Informatika Universitas Islam Indonesia Yogyakarta.
6. Kakakku Arief Burhannudin A.M S.H dan Lely Noor Maziyyah yang telah mensupport untuk menyelesaikan kuliah.
7. My Sweet Love (Winarti) kau anugerah yang ku miliki.
8. Picolo (you are my spirit),Dody (makasih udah nau anterin motor),Gok Ching (makasih atas tumpangan kost)
9. Teman-teman MOCY (Merzy Owner Club Yogyakarta) ,Mas Win, Paimo, Agung, Oo, Domo. Anak-anak Krangkungan , Selokan Mataram Yogyakarta (you Are the best Friends.thanks for your supports).
10. Pihak-pihak lain yang tidak dapat kami sebut, terima kasih atas bantuannya.

Kami Menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, Oleh karena itu masukan kritik dan saran yang membangun sangat kami butuhkan untuk kemajuan dan pengembangan Perangkat Lunak ini. Tidak lupa kami berharap Tugas Akhir ini bermanfaat bagi semua pihak untuk ke depannya kelak.

Wassalamu'alaikum Wr.Wb.

Penyusun

DAFTAR ISI

Halaman Judul.....	i
Lembar Pengesahan.....	ii
Kata Pengantar.....	iii
Daftar Isi.....	v
Daftar Tabel.....	ix
Daftar Gambar.....	x
Bab I Pendahuluan.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	6
1.6. Metodologi Penelitian.....	6
1.7. Sistematika Penulisan.....	9
Bab II Landasan Teori.....	11
2.1. Siklus Hidup Software (<i>Software Life Cycle</i>).....	11
2.1.1. Fase Analisis Sistem.....	12
2.1.1.1. Tahap Perencanaan (<i>Planning</i>).....	12
2.1.1.2. <i>Software Requirement Definition</i>	14
2.1.2. Fase Desain Sistem.....	15

2.1.2.1. Kriteria Modularisasi.....	16
2.1.2.2. Data Flow Diagram (DFD).....	17
2.1.2.3. <i>Flowchart</i>	18
2.1.3. Fase Implementasi.....	18
2.1.3.1. Bahasa Pemrograman Visual Delphi 7.0.....	19
2.1.3.1.1. Struktur Program Delphi.....	20
2.1.3.1.1.1. Form.....	22
2.1.3.1.1.2. Unit.....	23
2.1.3.1.1.3. Data Base.....	24
2.1.3.1.1.4. Operator.....	28
2.1.4. Fase Testing.....	31
2.1.4.1. Verifikasi dan Validasi.....	32
2.1.4.2. Penentuan <i>Acceptance Criteria</i> (Kriteria Serah Terima)	33
2.1.5. Fase <i>Maintenance</i>	33
2.1.5.1. Hal-hal yang mendukung proses <i>Maintenance</i>	34
Bab III Analisis Sistem.....	42
3.1. <i>System Definition</i>	43
3.1.1. <i>System User</i>	43
3.1.2. Analisis Jabatan.....	44
3.2. <i>System Process</i>	44
3.3. <i>System Problem And Solution</i>	50
3.4. <i>System Tools</i>	55

3.5. <i>System Plans</i>	55
Bab IV Desain Sistem.....	57
4.1. Tahap <i>External Design</i>	57
4.1.1. Tampilan Program Utama.....	58
4.1.2. Tampilan PSB (Penerimaan Siswa Baru).....	58
4.1.3. Tampilan Distribusi Kelas.....	61
4.1.4. Tampilan Pemantauan Proses Belajar Mengajar.....	61
4.1.5. Tampilan Raport.....	62
4.1.6. Tampilan Pemilihan Jurusan.....	62
4.1.7. Tampilan SPP.....	63
4.1.8. Tampilan BP.....	63
4.2. Tahap <i>Archithectural Design</i>	64
4.2.1. Desain Data Base	64
4.2.1.1. Fasilitas PSB.....	65
4.2.1.2. Fasilitas Distribusi Kelas.....	68
4.2.1.3. Fasilitas Pemantauan Proses Belajar Mengajar.....	68
4.2.1.4. Fasilitas Raport.....	69
4.2.1.5. Fasilitas Absensi.....	69
4.2.1.6. Fasilitas Penentuan Jurusan.....	70
4.2.1.7. Fasilitas SPP.....	70
4.2.1.8. Fasilitas Data Guru.....	71
4.2.1.9. Fasilitas Data Mata Pelajaran.....	71

4.2.2. Saling Ketergantungan (Interdependency).....	71
4.3. Desain Detail.....	72
Bab V Fase Implementasi.....	86
5.1. Program Utama.....	86
5.2. Fasilitas Penerimaan Siswa Baru.....	88
5.3. Fasilitas Distribusi Kelas.....	98
5.4. Absen.....	103
5.5. Fasilitas Ulangan Harian.....	105
5.6. SPP.....	110
5.7. Fasilitas BP.....	115
5.8. Fasilitas Raport.....	118
5.9. Fasilitas Pemilihan Jurusan.....	121
Bab VI Integrasi Program dan Fase <i>Testing</i>	134
6.1. Integrasi Program.....	134
6.2. <i>Testing</i>	136
6.2.1. <i>Alpha Test</i>	136
6.2.2. <i>Betha Test</i>	137
Bab VII Kesimpulan Dan Saran.....	139
A. Kesimpulan.....	139
B. Saran.....	140
Daftar Pustaka.....	141
Lampiran.....	142

Daftar Tabel

Tabel 2.1. Aktifitas Dan Daur Hidup Software.....	12
Tabel 2.2 Simbol DFD.....	34
Tabel 2.3.Simbol MEDIA.....	35
Tabel 2.4.Simbol PROSES.....	36
Tabel 2.5.Simbol RESCRIPTIVE.....	37
Tabel 2.6.Variant Tipe Data Integer.....	38
Tabel 2.7.Variant Tipe Data Real.....	38
Tabel 2.8.Variant Tipe Data Boolean.....	38
Tabel 2.9.Variant Tipe Data Char.....	39
Tabel 2.10.Variant Tipe Data String.....	39
Tabel 2.11.Operator Aritmetik Binari.....	39
Tabel 2.12.Operator Aritmetik Unari.....	40
Tabel 2.13.Operator Logika.....	40
Tabel 2.14.Operator Boolean.....	40
Tabel 2.15.Operator Relasional.....	41
Tabel 2.16.Urutan Operator.....	41
Tabel 4.1. Data Pribadi Calon Siswa.....	77
Tabel 4.2.Data Nilai Ujian Akhir Nasional Calon Siswa.....	77
Tabel 4.3. Data Nilai Ijazah Calon Siswa.....	78
Tabel 4.4. Data Kemampuan Akademis Calon Siswa.....	78
Tabel 4.5. Data Prestasi Non Akademik Calon Siswa.....	79
Tabel 4.6. Data Orang Tua Calon Siswa.....	79
Tabel 4.7. Data Pengumuman Hasil Seleksi PSB dan Daftar Ulang.....	79
Tabel 4.8. Data Siswa Aktif.....	80
Tabel 4.9. Data Nilai Ulangan Harian.....	80
Tabel 4.10.Data Nilai Raport.....	81
Tabel 4.11.Data Absensi.....	81
Tabel 4.12. Data Rekapitulasi Absen.....	82
Tabel 4.13. Data Minat Siswa Terhadap Jurusan.....	82
Tabel 4.14. Data Test IQ Siswa.....	82
Tabel 4.15. Data Pembayaran SPP.....	82
Tabel 4.16. Data Rekapitulasi SPP.....	83
Tabel 4.17.Data Pengajar.....	84
Tabel 4.18.Data Matapelajaran.....	84
Tabel 4.19.Interpedensi.....	85



Daftar Gambar

Gambar 2.1. Data Flow Diagram Sederhana.....	17
Gambar 3.1. Struktur Organisasi Sekolah.....	44
Gambar 4.1. Lay Out Program Utama.....	72
Gambar 4.2. Lay Out Tampilan Jurnal.....	73
Gambar 4.3. Lay Out Tampilan Distribusi Kelas.....	73
Gambar 4.4. Lay Out Tampilan Input Nilai Ulangan Harian.....	74
Gambar 4.5. Lay Out Tampilan Nilai Ulangan Harian Per MataPelajaran.....	74
Gambar 4.6. Lay Out Tampilan Nilai Ulangan Harian Per Kelas.....	75
Gambar 4.7. Lay Out Tampilan Nilai Raport.....	75
Gambar 4.8. Lay Out Tampilan Proses Penjurusan.....	76
Gambar 5.1. Tampilan Program Utama Sistem Informasi Pendidikan.	87
Gambar 5.2. Tampilan Input Data.PSB.....	89
Gambar 5.3. Tampilan Distribusi Kelas.....	98
Gambar 5.4. Tampilan Absensi Siswa.....	105
Gambar 5.5. Tampilan Awal Input Nilai Ulangan Harian	105
Gambar 5.6. Tampilan Proses Input Nilai Ulangan Harian	108
Gambar 5.7. Tampilan Proses Input SPP.....	113
Gambar 5.7. Tampilan Proses Input Data BP.....	116
Gambar 5.8. Tampilan Input Kasus BP.....	116
Gambar 5.9. Tampilan Input Nilai Ulangan Sumatif	118
Gambar 5.10. Tampilan Penentuan Rumus Nilai Raport.....	119
Gambar 5.11. Tampilan Input Minat Siswa.....	121
Gambar 5.12. Tampilan Penentuan Matapelajaran Penentu Proses Penjurusan.....	122
Gambar 5.13. Tampilan Proses Penjurusan.....	133
Gambar 4.18. Diagram Konteks.....	143
Gambar 4.12. DFD Level-1 Sistem Informasi Pendidikan.....	144
Gambar 4.1. Flowchart Pendaftar Siswa Baru.....	145
Gambar 4.2. Flowchart Distribusi Kelas.....	146
Gambar 4.4. Flowchart Pembuatan Raport.....	147
Gambar 4.6. Flowchart Proses BP/BK.....	148
Gambar 4.13. DFD Level-2 Input Data.....	149
Gambar 4.14. DFD Level-2 Penerimaan Siswa Baru(PSB).....	150
Gambar 4.15. DFD Level-2 Proses Distribusi Kelas.....	151
Gambar 4.16. DFD Level-2 Proses Pemantauan Belajar mengajar.....	152
Gambar 4.17. DFD Level-2 Proses Pembuatan Raport.....	153
Gambar 4.5. Flowchart Proses Pembayaran SPP,.....	154
Gambar 4.3. Flowchart Proses Belajar Mengajar.....	155

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi komputer selama satu dekade terakhir ini telah menciptakan suatu trend teknologi dalam pengelolaan sistem informasi. Trend ini berhasil menggeser Standarisasi Pengelolaan Sistem Informasi Manual Konvensional menjadi Standarisasi Teknologi Komputer yang berbasis pada Otomatisasi, keakuratan dan kecepatan yang tinggi.

Tetapi pada kenyataannya Standarisasi ini belum diimplementasikan secara utuh dalam pengelolaan sistem informasi yang berbasis komputer. Pemanfaatan komputerisasi dalam pengelolaan sistem informasi dewasa ini hanya sebatas sebagai "Mesin Ketik Plus", Kalkulator dan sebagai media penyimpanan dan *retrive* data manual. Tetapi belum sampai pada pemanfaatan aspek otomatisasi teknologi komputer dalam pengelolaan sistem informasi. Padahal *issue* komputerisasi harus mencakup aspek otomatisasi seperti yang dikatakan oleh seorang guru besar University of Tokyo yaitu Prof. Yoshiaki Shirai :

"Progress in data processing techniques using computers has meant automation of mental work that was previously performed by human beings. In particular, once the works sequence has been established, operations such as writing up a bank ledger on withdrawal or depositing of funds, calculation of the orbit of an artificial satellite, or searching for references, or financial report etc. can be performed more quickly and accurately by computer than by people". (Yoshiaki Shirai, "Artificial Intelligence, Concepts, Techniques and Applications", 1986)

Aspek otomatisasi ini hanya bisa diimplementasikan dalam pengelolaan sistem informasi jika terdapat *software* yang memang didesain khusus untuk pengelolaan sistem informasi tertentu. Sedangkan pada kenyataannya di pasaran sedikit sekali *software* yang didesain untuk pengelolaan sistem informasi tertentu itupun harus melalui penyesuaian dengan kondisi *user*.

Dunia Pendidikan salah satu institusi yang terkena imbas dari dampak teknologi komputerisasi, hampir semua sekolah maupun institusi pendidikan lainnya sudah menggunakan piranti komputer sebagai “alat bantu”, tetapi kenyataan yang ada sekarang ini tampak bahwa penggunaan komputer masih terbatas pada aplikasi-aplikasi yang sifatnya masih manual.

Sistem informasi pendidikan yang selama ini dipakai di dunia pendidikan masih menggunakan sistem informasi manual *konvensional*, sehingga banyak permasalahan-permasalahan klasik yang belum dapat diselesaikan secara *komprehensif*. Permasalahan-permasalahan itu diantaranya : administrasi keuangan, pemantauan hasil proses belajar mengajar, administrasi data pribadi siswa dan guru, dan permasalahan lain yang menuntut kecepatan proses yang tinggi seperti pembuatan raport, jadwal pelajaran dan pembagian kelas.

Untuk menyelesaikan permasalahan di atas mutlak diperlukan *software* yang didesain khusus untuk menangani permasalahan tersebut. Dan dalam pembuatan *software* ini diperlukan perencanaan yang matang dengan menggunakan kaidah-kaidah *software engineering*. Perancangan *software* sistem informasi pendidikan dengan menggunakan kaidah dasar *software engineering* memang terkesan lebih

rumit dibandingkan dengan perancangan sistem pemrograman *konvensional*, tetapi hasil yang diperoleh dari sistem perancangan dengan menggunakan kaidah *software engineering* memiliki reliabilitas, portabilitas yang tinggi, sehingga proses *maintenance* lebih mudah dilakukan.

Salah satu pertimbangan dalam kaidah *Software Engineering* yang sangat mendasar adalah pertimbangan bahasa pemrograman yang akan dipakai. Pertimbangan ini menjadi referensi utama dalam desain database, desain tampilan dan desain fasilitas-fasilitas dalam suatu sistem.

Pemrograman yang berorientasi objek dan berbasis GUI (*Grafic User Interface*) menjadi standard utama *product software* sekarang ini sedangkan untuk menghasilkan software dengan standart ini diperlukan bahasa pemrograman Visual yang berorientasi Object.

Delphi merupakan suatu bahasa Pemrograman yang sudah berorientasi object dengan sistem *Every Handler*. Pada dasarnya merupakan deviasi dari bahasa pemrograman Pascal klasik. Delphi sebagai *Inheritance* dari Pascal mempunyai sifat *Full Compatible* dengan Induknya, Begitu juga dengan sistem database yang dapat dikonversi Pascal yang menggunakan file bertipe (*Typed File*).

1.2. Rumusan Masalah

Menurut Tavri D Mahyuzir dalam “Analisis dan Perancangan Perangkat Lunak” bahwa suatu proyek pembuatan perangkat lunak (*software*) harus melalui 5 tahapan yang dikenal sebagai 5 daur hidup suatu *software* (*5-life cycles*) yaitu : analisis, desain, *coding*, *testing* dan *maintenance*. Tahapan *5-life cycles* merupakan

bagian mata rantai yang tidak bisa dipisahkan dan setiap tahapan mempunyai aktifitas dan target (*goal*) tersendiri tetapi saling berhubungan dengan yang lain.

Dalam penelitian tugas akhir ini, penulis untuk mencoba memformulasikan permasalahan proses desain dan implementasi sistem informasi pendidikan dengan mengacu pada 5 *software's life cycles* Dengan memanfaatkan Kompatibilitas Database Pascal Klasik Dengan Database Relasional yang disupport oleh Delphi.

1.3. Batasan Masalah

Dalam penelitian ini Penulis akan berusaha untuk berkonsentrasi pada permasalahan seputar aktifitas *5-software's life cycles* dengan *restriksi-restriksi* penelitian sebagai berikut :

1. Fase Analisis

Pada tahap ini penulis hanya akan meneliti bagaimana cara mengumpulkan data potensi *user* dan kebutuhan *user*, *requirement analysis*, menjadwalkan setiap aktifitas dalam proyek pengembangan *software* dan bagaimana cara membuat *form-form* pengumpulan data.

2. Desain

Pada tahap ini penulis lebih memfokuskan desain organisasi, struktur data dan *performance* sistem informasi pendidikan secara menyeluruh.

3. *Coding*

Pada tahap ini, penulis akan melakukan pembuatan *coding / Script* program dengan menggunakan bahasa pemrograman Delphi dan dengan

perancangan database yang mengacu pada database Pascal Klasik dan database Relasional Delphi.

4. *Testing*

Pada tahap ini penulis hanya memfokuskan pada permasalahan teknik *testing software* baik menggunakan kriteria programmer (*alpha test*) maupun menggunakan kriteria *user (betha test)*

1.4. Tujuan Penelitian

Tujuan penelitian “Perancangan Sistem Informasi Pendidikan” adalah :

1. Mempelajari proses perancangan *Software* Sistem Informasi Pendidikan dari mendefinisikan sistem informasi pendidikan sampai proses *Testing*
2. Meneliti sejauh mana keefektifan dan keefesienan teori-teori tentang desain dan implementasi *software* dengan menggunakan *5 cycles software engineering* pada dunia bisnis *software* khususnya dalam proyek perancangan sistem informasi pendidikan.
3. Mengetahui tingkat *deviasi* teori tentang desain *software* dengan dunia nyata dan menemukan suatu solusi baru untuk mengatasi *deviasi teoritis* yang disebabkan adanya adaptasi dengan kondisi real dunia bisnis *software*.
4. Mengetahui tip dan trik praktis dalam proses pembuatan suatu *software* sistem informasi.

5. Mengetahui kelemahan dan kelebihan desain database dengan menggunakan dua jenis database yaitu database Pascal Klasik dan database Relasional.

1.5. Manfaat Penelitian

Manfaat dari hasil penelitian tugas akhir ini, adalah :

1. Bagi penulis memperoleh pengetahuan bagaimana cara membuat sistem informasi pendidikan yang *reliable, portable*, efisien dan efektif yang mengacu pada teoritika *software engineering* dan adaptif dengan kondisi real dunia bisnis *software*.

Bagi Penulis dapat memperoleh Pengetahuan tentang kelemahan dan kelebihan pemrograman dengan menggunakan sistem klasik dan sistem database Relasional.

2. Bagi Dunia pendidikan, *software* yang dihasilkan dari penelitian ini bisa dipergunakan oleh institusi pendidikan sehingga proses pertukaran informasi menjadi lebih mudah dan cepat yang pada akhirnya dapat mempercepat proses *decision maker*.

1.6. Metodologi penelitian

Metode penelitian tugas akhir ini dilakukan melalui dua tahap, yaitu :

1. Metode pengumpulan data, data yang dikumpulkan dalam penelitian ini

Meliputi :

- a. Data primer

Data inti yang akan digunakan sebagai data pokok dalam proses analisis data. Data primer yang diperoleh secara langsung dari responden atau *user*. Teknik pengumpulan data ini dilakukan dengan metode sebagai berikut :

a) Teknik *Observasi*

yaitu melakukan pengamatan dan pencatatan secara langsung dengan meninjau langsung ke object data. Dalam penelitian tugas akhir ini teknik obeservasi digunakan oleh penulis dalam mengumpulkan survei perangkat keras yang dipakai *user*.

b) Teknik *Interview*

Yaitu metode penelitian yang melakukan wawancara langsung dengan *user*. Teknik ini digunakan oleh Penulis dalam pengumpulan data yang berhubungan dengan kemampuan SDM *user*, Struktur organisasi, alur data, dan fasilitas yang diinginkan oleh *user*.

c) Teknik *Questioner*

Yaitu metode penelitian dengan mengajukan daftar pertanyaan-pertanyaan kepada *user* sehingga dapat memperoleh informasi yang dibutuhkan. Teknik ini digunakan oleh penulis dalam melakukan proses *betha test*.

b. Data Sekunder

yaitu data penunjang / pelengkap yang akan digunakan dalam proses analisis data. Data sekunder ini dapat dikumpulkan dengan menggunakan metode sebagai berikut :

a) Studi literatur

yaitu mempelajari buku-buku referensi *software engineering* serta buku-buku lain yang dapat menunjang penelitian ini.

b) Telaah Dokumen

Yaitu mempelajari dokumen-dokumen pembuatan *software* sistem informasi yang berhubungan dengan bidang permasalahan.

2 Metode analisis data

Metode analisis data yang digunakan dalam penelitian tugas akhir ini adalah : metode analisis data kualitatif deskriptif yang dilakukan dengan cara :

- 1) Meringkas data, dimana data mentah diseleksi dan dikelompokkan dalam group-group tertentu.
- 2) Menjabarkan data dan menghubungkan koneksitas antar group data.
- 3) Menarik kesimpulan yang kemudian menarik prediksi atas kemungkinan selanjutnya, dalam hal ini apakah pembuatan *software* sistem informasi pendidikan sudah sesuai dengan keinginan *user* dan desain sistem *software*.

Dalam penelitian Tugas akhir ini, penulis menganalisis data dari dua bagian, yaitu :

- a) Analisis data yang digunakan melalui proses pengamatan langsung, wawancara dan *questioner*.
 - b) Analisis data dokumentasi, yaitu menganalisis setiap proses dalam *5-life cycles* dalam penyusunan *schedule*.
- 3 Tahap Implementasi

Pada tahap implementasi ini, penulis menggunakan Bahasa pemrograman Delphi 5.0 dengan sistem database ganda (database klasik dan database Relasional) dalam perancangan sistem informasi pendidikan.

I.7 Sistematika Penulisan

Bab I PENDAHULUAN

Pada bab ini diuraikan latar belakang penelitian tugas akhir, dasar teori, rumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

Bab II Landasan Teori

Pada bab ini diuraikan konsep dasar *software engineering* dan Konsep pemrograman dengan menggunakan bahasa pemrograman Visual Delphi 5.0

Bab III Analisis Sistem

Pada bab ini diuraikan tentang bagaimana mengimplementasikan teori analisis sistem pada konsep *software engineering* yang terbagi menjadi beberapa sub fase yaitu : *system definition, project plan, software requirements definition* dalam perancangan sistem informasi pendidikan.

Bab IV Pada bab ini diuraikan bagaimana mendesain sistem informasi pendidikan dengan konsep *software engineering* yang terdiri dari beberapa sub fase yaitu desain external, desain arsitektur dan detail desain serta pertimbangan-pertimbangan yang harus diperhatikan dalam mendesain sebuah *software*.

Bab V Pada bab ini diuraikan bagaimana mengimplementasikan hasil dari fase desain dalam fase *coding*.

Bab VI Pada bab ini diuraikan bagaimana mengimplementasikan teknik *testing* pada sebuah produk *software* terintegrasi seperti teknik *testing* *alpha test* dan *beta test*, teknik penentuan *acceptance criteria* internal maupun *customer*.

Bab VII Pada bab ini diuraikan kesimpulan dari penelitian tugas akhir ini dan saran-saran dalam pembuatan *software* maupun perbaikan sistem manajemen pendidikan.

BAB II

LANDASAN TEORI

2.1. Siklus Hidup *Software* (*Software Life Cycle*)

Perencanaan proses pengembangan *software* harus melalui berbagai pertimbangan. Pertimbangan pertama yang harus dilakukan oleh *software engineer* adalah mendefinisikan model *life cycle* dari *software* yang akan dirancang. *software life cycle* meliputi semua aktifitas yang dibutuhkan untuk mendefinisikan, mendesain, mengimplementasikan, menguji dan *maintenance* sebuah produk *software*.

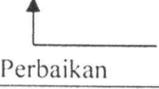
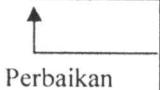
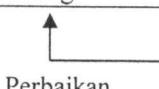
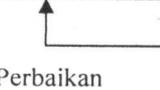
Pada umumnya penggunaan *Software 5 life cycles* hanya untuk *project-project Large Size Project* (*project* berskala besar). Ini disebabkan oleh :

1. Langkah perancangan program dengan menggunakan *5 life Cycles Software* terkesan lebih rumit.
2. Banyak Membutuhkan waktu dalam desain maupun implementasi.

Fairley menetapkan ada lima fase dalam *software life cycle* yang dijadikan landasan pengembangan dan *maintenance* sebuah *software*, yaitu :

1. Fase Analisa (*Analysis Phase*)
2. Fase Desain (*Design Phase*)
3. Fase Implementasi (*Implementation Phase*)
4. Fase Uji Coba (*System Test Phase*)
5. Fase *Maintenance* (*Maintenance Phase*)

Tabel 2.1. Aktifitas Dan Daur Hidup Software

Analysis Phase	Design Phase	Implementasi	System Test	Maintenance Phase
<ul style="list-style-type: none"> ➤ Perencanaan ➤ Identifikasi Sistem 				
	<ul style="list-style-type: none"> ➤ Arsitektur ➤ Detail Sistem 			
		<ul style="list-style-type: none"> ➤ Pemrograman ➤ Penelusuran ➤ Testing 		
			<ul style="list-style-type: none"> ➤ Integrasi Program ➤ Serah Terima (Acceptual) 	
				<ul style="list-style-type: none"> ➤ Pengembangan ➤ Adaptasi ➤ Penetapan

2.1.1 Fase Analisis Sistem

Fase ini merupakan fase pertama dalam 5 daur hidup *software*. Dalam fase ini ada dua aktifitas penting yang harus dikerjakan oleh *Software engineer* yaitu :

2.1.1.1. Tahap Perencanaan (Planning)

Pada tahap ini seorang *software engineer* harus memahami problem *customer* kemudian melakukan inventarisasi terhadap kemungkinan penyelesaian masalah, menentukan solusi problem *customer* yang tepat dengan kondisi sistem, menentukan kriteria penerimaan (*acceptance criteria*) dan merencanakan semua proses pengembangan. Hasil yang didapat dari tahap ini dapat diklasifikasikan menjadi :

a. System Definition

System Definition dapat diartikan sebagai proses mendeskripsikan system yang akan dikembangkan. Proses deskripsi ini dengan mengekspresikan sejumlah data sistem ke dalam bahasa formal dengan dilengkapi oleh tabel, grafik, *event graph*, persamaan matematis dan lain sebagainya. Leman memperinci data yang *representatif* untuk dijadikan deskriptor *system* di antaranya : *system users* (meliputi struktur organisasi dan uraian tugas), *System process*, *System Rules*, *System problems & solutions*, *system tools* dan *System plans* [Leman97] Deskripsi dalam tahap *system definition* ini hanya sebatas pemaparan system secara kuantitatif tidak mencakup deskripsi proses kontrol system.

Leman menilai bahwa Tahap *system definition* ini merupakan tahap yang kritis karena kesalahan yang terjadi pada tahap ini akan menyebabkan kesalahan pada tahap desain dan implementasi. Untuk menghindari kesalahan ini diperlukan pemahaman langsung dari *domain* dan lingkungan problem.

Untuk mengumpulkan data deskripsi system ini, seorang *software engineer* dapat menggunakan metode *interview*, observasi, *questioner* dan pembacaan dokumen perusahaan. Tahap ini menuntut keahlian komunikasi dari seorang *software engineer* dalam melakukan eksplorasi informasi *customer*, karena terkadang *customer* tidak

mengetahui bahwa permasalahannya dapat dipecahkan dengan sistem komputerisasi.

b. Project Plan

Pada tahap *Project Plan* ini *software engineer* melakukan pembuatan model 5 fase daur hidup *software* seperti struktur organisasi proyek, *draft* penjadwalan proyek, persiapan *rekrutment* karyawan baru, persiapan alat dan teknik yang akan digunakan dalam fase pengembangan. Semua yang dihasilkan dari tahapan *project plan* ini masih bersifat 'sementara' dan akan *direfine* pada fase pengembangan.

2.1.1.2 *Software Requirement Definition*

Software requirement definition adalah sub fase dari *planning* yang berisi deskripsi dari kebutuhan *software*. Deskripsi ini mengacu pada definisi yang telah dibuat pada sub fase *system definition* tetapi pada sub fase ini deskripsi sistem dibuat lebih detail dan lebih spesifik sehingga deskripsi karakteristik keunggulan produk *software* dapat dihasilkan.

Software requirement definition pada dasarnya hanya mengimplementasikan kata tanya "what" pada suatu produk *software*. Dan hasil dari *Software requirement definition* ini dijadikan dasar dari sistem desain yang akan mengimplementasikan "how" pada suatu produk *software*.

Fairley menyimpulkan seluruh aktifitas analisa sistem dalam suatu ukuran baku yang dapat dijadikan standarisasi suatu produk *software*. Ukuran standard fase analisis sistem itu adalah :

1. Tingkat portabilitas
2. Tingkat Reliabilitas / Keandalan (*Realibility*)
3. Tingkat Efisiensi (*Efficiency*)
4. Tingkat Human Engineering (*Interfacing With User*)
5. Tingkat Uji Coba (*Testability*)
6. Tingkat Pemahaman (*Understandability*)
7. Tingkat Modifikasi (*Modifiability*)

2.1.2. Fase Desain sistem

Setelah memahami sistem yang ada termasuk solusi dari setiap problem *user* dan kebutuhan pengguna (*user requirement*) maka tahap selanjutnya mendesain sistem baru agar semua permasalahan yang ada pada sistem lama dapat tercover dengan baik. Fase ini memberikan gambaran rancang bangun (*blue print*) yang lengkap sebagai penuntun (*guideline*) bagi *programmer* dalam mengembangkan aplikasi. Sesuai dengan sistem yang akan dikomputerisasikan maka yang harus didesain mencakup *hardware*, *software*, *database*, aplikasi dan gambaran rincian tugas. Fairley membagi fase desain menjadi tiga sub fase yaitu : *external* desain, *Architectural* desain dan detail desain. Desain arsitektur dan detail desain mengacu pada desain *internal software*.

1. *External Desain* mencakup penyusunan, perencanaan dan menentukan karakteristik produk *software external*. Karakteristik *external* itu di antaranya : tampilan program, format laporan, sistem penyimpanan data *external*, karakteristik fungsional, unjuk kerja yang diharapkan. *External* desain sebenarnya dimulai pada fase analisa sistem terutama pada sub fase *requirements definition*. Dengan kata lain *external* desain hanya merupakan proses penajaman fase *requirements definition* dengan mengubah kata tanya dari “*what*” menjadi “*why*”.
2. *Architectural Design* memfokuskan pada perbaikan deskripsi konseptual sistem, identifikasi fungsi-fungsi internal, proses dekomposisi modul ke fungsi yang lebih sederhana, menentukan struktur data, teknik penyimpanan, hubungan antar fungsi dan data.
3. *Detail Design* lebih menekankan pada detail setiap proses desain yang nantinya terekam dalam *blue print* dan akan digunakan lebih lanjut dalam fase *implementation* dan fase *Testing*.

2.1.2.1 Kriteria Modularisasi

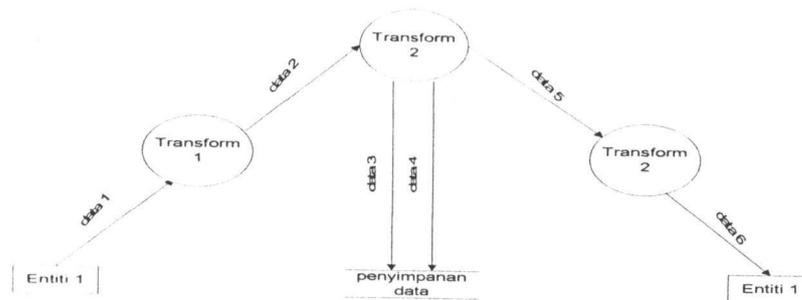
Modularisasi memungkinkan seorang *software engineer* untuk melakukan dekomposisi sebuah sistem menjadi beberapa sub sistem, dan sub sistem menjadi beberapa *procedure* dan *function*. Dengan menggunakan teknik modularisasi ini data abstraksi dan unit *independen* secara mudah dapat diimplementasikan.

Teknik modularisasi ini memiliki karakteristik di antaranya :

1. Modul hanya berisi instruksi-instruksi, proses logika dan struktur data
2. Modul dapat *dicompile* secara terpisah dan disimpan dalam sebuah *library*.
3. Modul dapat diintegrasikan dalam sebuah program.
4. Bagian modul dapat digunakan dengan sistem pemanggilan menggunakan nama dan parameter.
5. Modul dapat menggunakan modul lain.

2.1.2.2. Data Flow Diagram (DFD)

Data Flow Diagram adalah teknik grafis yang digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data hingga keluar (TDM91). Contoh sederhana dapat digambarkan sebagai berikut :



Gambar 2.1. Data Flow Diagram Sederhana

Dari gambar di atas dapat dilihat bahwa pembuatan *Data Flow Diagram* (DFD) dimulai dari sumber informasi data yang disebut data *entity*. Dari *Entiti* data bergerak untuk diproses oleh *Transform* 1 sehingga menghasilkan data 2

yang akan diproses oleh *Transform 2* dan menghasilkan data 3 yang disimpan pada tempat penyimpanan data yang berupa *file data*.

Selanjutnya proses yang dilakukan oleh *Transform 2* adalah membaca atau mengambil data 4 dari tempat penyimpanan data kemudian diolah sehingga menghasilkan data 5 yang akan diproses oleh *Transform 3* sehingga menghasilkan data 6 yang akan dimanfaatkan oleh tujuan informasi atau disebut juga *Entiti*.

Level 01 pada DFD disebut model dasar system yang menggambarkan keseluruhan elemen perangkat lunak dengan satu proses atau *Transform*, satu masukan data data dan satu keluaran. Simbol-simbol yang digunakan pada penggambaran DFFD dapat dilihat pada tabel 2.2 simbol DFD.

2.1.2.3. Flowchart

Flowchart merupakan gambar simbol-simbol logika yang berguna untuk mendeskripsikan algoritma program dengan mengontrol aliran satu kerja ke kerja lainnya. Simbol-simbol *flowchart* di antaranya simbol Media, simbol proses, simbol Rescriptive. Gambar simbol, nama Simbol dan fungsi dari simbol-simbol *flowchart* dapat dilihat pada tabel 2.3. tabel 2.4, dan tabel 2.5 simbol flowchart.

2.1.3 Fase Implementasi

Seluruh langkah dalam *Software engineering* yang telah dijabarkan, semuanya bertujuan untuk menerjemahkan keperluan sistem ke dalam bentuk yang

dimengerti oleh komputer. Pada fase ini dibahas langkah pengkodean yaitu proses *transformasi* hasil perancangan (algoritma) ke dalam bahasa pemrograman yang telah ditentukan.

2.1.3.1 Bahasa Pemrograman Visual Delphi 7.0

Delphi sebenarnya merupakan bahasa pemrograman hasil evolusi dari bahasa pemrograman Pascal yang didesain oleh Niklaus Wirth. Pengembangan ini didasari oleh suatu konsep yang dalam dunia *programming* dikenal dengan konsep OOP (*Object Oriented Programming*) yang mulai dikenal sejak diluncurkan Turbo Pascal versi 5.5. Pengembangan ini juga didasari pada konsep Komputer grafis yang dikenal konsep GUI (*Grafik User Interface*) sehingga Setiap program Aplikasi yang dihasilkan dari IDE (*Integrated Development Environment*) memiliki Kompatibilitas dan tampilan standard Windows.

Pengembangan-pengembangan yang disupport oleh bahasa pemrograman Delphi memang berbeda dengan versi terakhir bahasa Pemrograman Pascal klasik (Turbo Pascal 7). Pengembangan ini di antaranya meliputi :

1. Pengembangan tipe data

Tipe data yang disupport oleh Delphi lebih kompleks dibandingkan dengan tipe data Pascal klasik. Delphi mensupport tipe-tipe data yang

digunakan dalam database seperti *DATE*, *MEMO*, *BLOB*, *OLE* dan lain sebagainya.

2. Pengembangan Struktur

Delphi melakukan pengembangan-pengembangan dengan menghilangkan batas pada tipe *Array* sehingga penggunaan *Array* dalam Delphi lebih fleksibel dibandingkan dalam Pascal Konvensional.

3. Adanya Properti Terprogram

Dalam Delphi dikenal istilah Properti yang dapat diasumsikan sebagai variable tetap bagi sebuah objek. Dengan adanya pengembangan di atas Delphi lebih cerdas dibandingkan bahasa pemrograman Pascal Konvensional tetapi Delphi tidak melupakan pendahulunya, semuanya *Script* yang ditulis dalam bahasa Pascal *Full Compatibel* di lingkungan Delphi.

2.1.3.1.1. Struktur Program Delphi

Suatu Aplikasi atau *Project* yang didesain dengan bahasa pemrograman Delphi akan terdiri dari beberapa *file* yaitu file Project(.dpr), file unit(.pas), file Form(.Dfm), file Project Option, file Resource(.Res), file Backup(~dp,~df,~pa)

1. *File Project (.dpr)*

File Project (.dpr) ini dipakai untuk menyimpan informasi mengenai *form* dan *unit*. Program *inisialisasi* ada disini.

2. *File Unit (.Pas)*

File ini menyimpan program. Beberapa unit berhubungan dengan *Form*, beberapa lainnya berisi *procedure* dan *function*.

3. *File Form (.Dfm)*

File ini adalah *file biner* yang dibuat Delphi untuk menyimpan informasi mengenai *form*. Setiap *file form* mempunyai sebuah *file unit (.Pas)*.

4. *File Project Option*

File ini berisi setting option dari *project*.

5. *File Resource (.res)*

File biner ini berisi sebuah *icon* yang dipakai oleh *project*. *file* ini tidak dapat dibuat atau diubah oleh *user* karena Delphi secara terus menerus mengupdate atau membuat ulang *file* ini.

6. *File Backup (.~dp, ~df, ~pa)*

File-file backup untuk *project form* dan *unit*.

Kelompok berikutnya adalah *file-file* yang dibuat oleh *compiler* Delphi di antaranya file Executable(.exe).file unit project(.dcu) file dynamic link library(.dll), file Help(.hlp), file Image atau grafik(.wmf,.bmp,.ico).

1. *File executable (.exe)*

File executable dari aplikasi anda. *file* ini berdiri sendiri dan hanya memerlukan *file library, DLL, VBX* dan lain-lainnya.

2. *File unit object (.dcu)*

File unit (.dcu) file unit (.pas) versi terkompilasi dan akan dilink ke dalam *file executable* akhir.

3. *File dynamic link library (.dll)*

File ini dibuat jika anda merancang DLL sendiri.

4. *File Help (.hlp)*

File Help standard dari Windows yang dapat dipakai di aplikasi Delphi.

5. *File image atau grafik (.wmf,.bmp,.ico)*

File-file ini dipakai agar aplikasi lebih menarik.

2.1.3.1.1.1. Form

Form merupakan “windows” dalam sebuah aplikasi berbasis Windows. Sebuah aplikasi yang didesain dalam IDE

Delphi terdiri dari satu atau lebih *form* penyusun program. *Form-form* ini digunakan untuk menempatkan komponen-komponen Delphi, Kontrol dan lain sebagainya. Informasi yang berkenaan dengan *form* disimpan dalam dua *file* yaitu *file* .dfm dan .pas. *File* .dfm berisi informasi mengenai tampilan *form*, ukuran, lokasi dan lain sebagainya. Sedangkan program *form* dan program untuk mengontrol *form* disimpan dalam *file* .pas yang disebut juga sebagai *file* unit.

Setiap *form* dalam IDE Delphi memiliki property, dengan menggunakan property ini seorang programmer dapat merubah warna, ukuran, lokasi. Dan setiap *form* juga memiliki *event handler* yaitu kontrol berdasarkan *event* yang didesain oleh *programmer*.

2.1.3.1.1.2. Unit

Ada tiga jenis unit dalam Delphi yaitu :

1. Unit yang berhubungan dengan *form* (unit yang paling umum)
2. Unit yang digunakan membangun komponen.
3. Unit yang berisi *procedure* atau *function*.

Secara umum unit memiliki *syntax* sebagai berikut :

```
Unit Unit1;  
Interface
```

Uses Windows, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

Type

TForm1 = Class(*TForm*)

Procedure *FormCreate*(Sender : TObject);

Private

{Private Declarations}

Public

{Public Declarations}

End;

Var

Form1 : *TForm*;

Implementation

{*\$R *.DFM*}

2.1.3.1.1.3. Data Base

Dalam lingkungan Delphi, data base dapat didesain dengan tiga teknik, yaitu :

a. Dengan menggunakan teknik pascal konvensional,

Dengan menggunakan teknik ini data base yang dihasilkan biasanya menggunakan tipe-tipe data yang biasa digunakan dalam pascal konvensional. Tipe data yang digunakan dalam lingkungan pascal konvensional adalah tipe *integer*, tipe *real*, tipe *currency*, tipe *Boolean*, tipe *character*, tipe *string*, tipe *array*, tipe *record*, tipe *subrange*, tipe *set*..

1. Tipe *integer*

tipe data ini digunakan untuk menyatakan bilangan yang tidak mempunyai desimal. Tipe-tipe bilangan *integer* beserta jangkauan dan kebutuhan memorinya dapat dilihat pada tabel 2.6. Variant Tipe Data Integer.

2. Tipe *Real*

Tipe data *Real* dipakai untuk menyimpan bilangan yang mempunyai desimal. Tipe-tipe bilangan *real* dapat dilihat pada tabel 2.7. Variant Tipe Data Real. :

3. Tipe *Currency*

Tipe *Currency* mempunyai keuntungan yaitu tipe *currency* mempunyai ketelitian yang lebih tinggi dalam menangani bilangan yang besar. Tipe *Currency* dipakai pada *Currency Field* dan komponen-komponen lain serta kompatibel dengan tipe-tipe database yang menyatakan uang.

4. Tipe *Boolean*

Tipe *Boolean* dipakai untuk menyatakan nilai logika. Variabel tipe ini hanya dapat berisi *TRUE* atau *FALSE*. Tipe *Boolean* diperlukan untuk menyatakan data

yang isinya YA atau Tidak, BENAR atau SALAH, *ON* atau *OFF*.

Variabel tipe *Boolean* dapat menerima operator *AND*, *OR* dan *NOT*.

Tipe-tipe Boolean dapat dilihat pada tabel 2.8. Variant Tipe Data Boolean.

5. Tipe *Character*

Tipe *Character* dipakai untuk menyimpan satu huruf. Sebuah karakter berukuran satu byte. Jika Anda menghitung 2^8 , hasilnya adalah 256 dan berarti ada 256 karakter berbeda yang dapat mengisi sebuah variabel tipe karakter.

Tipe-tipe *Character* dapat dilihat pada tabel 2.9. Variant Tipe Data Char.

6. Tipe *String*

String adalah sederetan karakter yang membentuk kesatuan. Tipe *String* lebih banyak gunanya daripada tipe karakter. *String* dapat dipakai untuk menyimpan data Nama, Alamat, Kota dan lain-lain.

Tipe-tipe *String* dapat dilihat pada Tabel 2.10. Variant Tipe Data *String*.

7. Tipe *Array*

Tipe *Array* adalah sebuah variable tunggal yang dipakai untuk sekumpulan data. *Array* Dapat memanipulasi atau meng-copy *array* hanya dengan sebuah nama.

8. Tipe *Record*

Record dipakai untuk menyimpan sekumpulan data yang berhubungan .Elemen - elemen dalam *array* mempunyai tipe yang sama, tetapi elemen-elemen *record* dapat mempunyai tipe yang sama atau tipe yang berbeda.

9. Tipe *Subrange*

Tipe *Subrange* adalah range dari nilai-nilai dengan tipe tertentu. *Subrange* harus memiliki nilai terkecil dan nilai terbesar.

10. Tipe *Set*

Dengan *Set* Anda dapat mendefinisikan elemen-elemennya. Misalnya set yang berisi karakter-karakter yang menyatakan YA atau TIDAK, anda akan mendefinisikan set yang berisi empat karakter yaitu ['Y', 'y', 'T', 't']. Setelah itu anda dapat mendeteksi apakah sebuah data ada dalam set atau tidak.

2.1.3.1.1.4.Operator

1. Operator Aritmatik

a. Operasi Aritmatik Binari

Dapat dilihat pada Tabel 2.11. Operator Aritmatik Binari

b. Operasi Aritmatik Unari

Dapat dilihat pada Tabel 2.12. Operator Aritmatik Unari.

2. Operator Logika

Operator logika dibagi menjadi dua kelompok : operasi logika dan operasi *Boolean*. Operasi logika berhubungan dengan pergeseran atau pembagian pada level bit dan operasi membandingkan nilai pada level *TRUE* atau *FALSE*.

a. Operasi Logika

Dapat dilihat pada Tabel 2.13. Operator Logika.

b. Operator Boolean

Dapat dilihat pada Tabel 2.14. Operator Boolean.

c. Operator Relasional

Dipakai untuk membandingkan dua variable sama seperti di matematika. Dapat dilihat pada Tabel 2.15. Operator Relasional.

d. Urutan Operator.

Sama seperti di matematika, kita harus mengetahui urutan-urutan dalam mengevaluasi ekspresi. Anda harus mengetahui urutan operator.

Urutan operator Dapat dilihat pada Tabel 2.16. Urutan Operator .

Dengan menggunakan teknik desain data base berbasis pascal konvensional maka output yang dihasilkan akan berbentuk *file text* maupun *file* bertipe. Kedua *file* ini merupakan *file internal* Delphi (tidak bisa diakses oleh program aplikasi data base).

b. Dengan menggunakan *Format Table*, untuk mendesain dengan *format table* ada dua cara dalam Delphi :

1. Dengan menggunakan fasilitas *internal* Delphi dengan komponen visual *table*.
2. Dengan menggunakan fasilitas *external* yaitu dengan menggunakan database desktop.

Sedangkan database yang dihasilkan biasanya memiliki *format* Dbase atau Paradox.

c. Dengan menggunakan teknik *SQL (Structure Query Language)* dengan menggunakan bahasa *SQL* maka akses data

base dengan operasi tertentu seperti menggabungkan beberapa *table* menjadi lebih mudah.

2.1.4 Fase *Testing*

Fase *Testing* (uji coba) ini merupakan proses pencarian kesalahan dalam sebuah produk *software* dengan kata lain uji coba menentukan tingkat deviasi hasil *coding* terhadap hasil dari fase desain maupun dengan *user requirement*.

Ada 2 hal penting dalam fase uji coba ini yaitu :

1 Proses uji coba program integrasi

Semua *procedure*, fungsi dan modul yang telah selesai dan terbebas dari kesalahan *syntack* maupun semantik diintegrasikan menjadi satu menjadi program utama. Kemudian dilakukan uji coba terhadap program integrasi (program utama). Ada beberapa tahap penting dalam melakukan proses uji coba *software* integrasi yaitu Menentukan aliran informasi uji coba: dan Perancangan *Test Case*.

a. Menentukan aliran informasi uji coba

Dengan menentukan aliran informasi uji coba maka apabila terdapat suatu kesalahan maka langkah yang harus dilakukan oleh suatu *software engineer* menjadi lebih terarah. Sehingga proses *follow up* dari setiap penemuan kesalahan pada proses uji coba dapat diprediksikan dan dapat ditindak lanjuti dengan cepat.

b Perancangan *Test Case*

Dengan menentukan aliran informasi uji coba maka *software engineer* dapat merancang *test case* sehingga *software* dapat diuji coba pada kondisi-kondisi kritis. Dengan menggunakan perancangan *test case* ini diharapkan program yang dihasilkan dapat *running* pada situasi yang sangat kritis.

2 Teknik *Testing* (uji coba)

Dalam melaksanakan proses *Testing* (uji coba) diperlukan teknik-teknik uji coba. Ada beberapa teknik standard dalam melakukan proses uji coba di antaranya :

a. Uji coba *White Box*

Uji coba sistem yang mengacu pada mekanisme *Testing* dibawah ini :

- 1) Melakukan uji coba seluruh *independent path* di dalam modul sekurang-kurangnya 1 kali.
- 2) Menguji seluruh keputusan logika.
- 3) Menguji setiap *loop* dengan batasannya.
- 4) Menguji setiap struktur data *internal* dengan tingkat validitasnya.



b. Uji Coba *Basis Path*

Uji coba *basis path* ini merupakan pengembangan dari uji coba *white box*, hanya saja pada uji coba *basis path* lebih menekankan pada pengujian setiap keputusan logik dari perancangan *procedural* untuk memprediksikan model uji coba yang disebut dengan *basis path*. Kemudian dengan menggunakan *basis path* semua perintah diuji minimal satu kali

2.1.4.1. *Verifikasi Dan Validasi*

Apabila terdapat kesalahan pada proses uji coba maka teknik yang dilakukan untuk memperbaiki suatu produk *software* adalah dengan melakukan *verifikasi* dan *validasi*.

Menurut Hoover & Perry Verifikasi adalah proses untuk menentukan apakah operasi logika (program komputer) berhubungan dengan logika *flowchart* atau dengan istilah yang sederhana dapat dipertanyakan “*apakah terdapat error dalam program ?*”

Sedangkan *Validasi* adalah proses untuk menentukan suatu konsep dan abstraksi sistem merupakan representasi akurat dari sistem *real*. Atau dengan istilah yang sederhana dapat dipertanyakan “deskripsi, abstraksi ataupun konsep sistem menggambarkan sistem yang sebenarnya”.

2.1.4.2 Penentuan *Acceptance Criteria* (kriteria serah terima)

Uji coba yang didasarkan pada *penentuan* kriteria sebenarnya dapat dibedakan menjadi dua, yaitu uji coba Alpha test dan uji coba Beta Test:

1. *Alpha test*

Uji coba yang didasarkan pada kriteria internal team *software engineer*. Kriteria ini biasanya ditentukan berdasarkan pada hasil hasil desain dan *test case*.

2. *Betha Test*

Uji coba yang dilakukan dengan dasar kriteria yang ditetapkan oleh *customer*. Biasanya *software engineer* menggunakan *form-form testing* untuk menentukan kekurangan dari sistem yang dikembangkan dalam perspektif *customer*.

2.1.5. Fase *Maintenance*

Maintenance adalah istilah yang digunakan untuk menjabarkan setiap aktifitas dari *software engineer* yang terjadi pasca instalasi perangkat lunak. Ada beberapa aktifitas dalam fase *maintenance*, di antaranya :

- 1 Perbaikan, Penambahan atau peningkatan produk perangkat lunak, aktifitas ini meliputi :
 - 1) Penambahan fungsi-fungsi baru.
 - 2) Perbaikan tampilan atau modus interaktif.
 - 3) Memperbarui dokumen *eksternal*.

- 4) Memperbarui dokumen *internal*.
 - 5) Memperbarui karakteristik *performance* sistem.
2. Adaptasi terhadap lingkungan baru, aktifitas ini meliputi :
- 1) Pemindahan perangkat lunak ke mesin yang berlainan.
 - 2) Modifikasi untuk dapat mempergunakan *protokol* atau *device* yang berlainan.
3. Pembetulan, aktifitas ini meliputi :
- Pembenaran kesalahan yang timbul setelah produk perangkat lunak dipakai oleh *user* (kesalahan ini terjadi akibat sistem *test* yang kurang baik).

2.1.5.1 Hal-hal yang mendukung proses *Maintenance*

Ada beberapa hal yang sangat mendukung proses *maintenance*, yaitu :

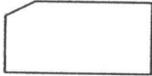
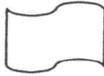
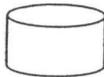
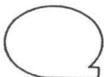
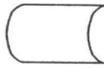
- 1 Perangkat lunak didesain dengan sistem modularitas
- 2 Setiap modul mempunyai kejelasan instruksional dan target.
- 3 Kejelasan dokumentasi *internal* dan dokumentasi pendukung lainnya.

Tabel 2.2. Simbol DFD

Simbol	Arti
	<i>Entiti / Source / Link</i>
	<i>Proses / Transform</i>
	Aliran Data
	Penyimpanan Data

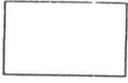
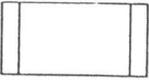
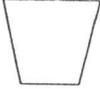
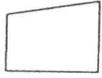
a. Simbol Media

Tabel 2.3. Simbol MEDIA

Simbol	Nama Simbol	Fungsi
	<i>Punched Card</i>	Digunakan untuk menggambarkan beberapa data yang diinput melalui <i>card</i> .
	<i>Document</i>	Digunakan untuk menggambarkan data yang bias dibaca seperti data cetakan masukan atau keluaran.
	<i>Online Display</i>	Digunakan untuk menggambarkan data atau informasi yang ditampilkan pada layar monitor.
	<i>Paper Tape</i>	Digunakan untuk menggambarkan data atau informasi yang tersimpan di <i>paper tape</i> .
	<i>Magnetik Disk</i>	Digunakan untuk menggambarkan data atau informasi yang tersimpan di <i>Magnetik Disk</i> .
	<i>Magnetic Tape</i>	Digunakan untuk menggambarkan data atau informasi yang tersimpan di <i>Magnetic Tape</i> .
	<i>Online Storage</i>	Digunakan untuk menggambarkan data atau informasi yang tersimpan secara <i>online</i> .

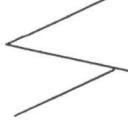
b. Simbol Proses

Tabel 2.4. Simbol PROSES

Simbol	Nama Simbol	Fungsi
	<i>Computer Processing</i>	Menggambarkan sebuah pemrosesan fungsi dalam <i>Flowchart</i> .
	<i>Predefine Processing</i>	Menggambarkan nama Proses seperti <i>subroutine</i> atau modul dalam <i>flowchart</i> .
	<i>Input / Output</i>	Menggambarkan beberapa operasi <i>input</i> atau <i>output</i> .
	<i>Decision</i>	Menggambarkan beberapa nilai dalam suatu proses di mana keputusan harus diputuskan untuk menentukan aksi selanjutnya.
	<i>Sorting</i>	Menggambarkan beberapa operasi yang meliputi penyortiran atau permintaan data.
	<i>Manual Operation</i>	Menggambarkan operasi yang dilakukan tidak <i>online</i> yang tidak membutuhkan alat mekanik
	<i>Manual Input</i>	Digunakan untuk menggambarkan beberapa operasi pemasukan yang tidak mekanik.
	<i>Auxilliary Operation</i>	Digunakan untuk menggambarkan Beberapa proses yang dilakukan oleh Sistem Computer.

c. Simbol Rescriptive

Tabel 2.5. Simbol RESCRIPTIVE

Simbol	Nama Simbol	Fungsi
	<i>Directional Flow</i>	Digunakan untuk menggambarkan Arah urutan proses atau <i>event</i> lainnya.
	<i>Comunication Link</i>	Digunakan untuk menggambarkan beberapa <i>transmisi</i> data oleh komunikasi data dengan metode komunikasi.
	<i>Terrminal</i>	Digunakan untuk menggambarkan permulaan dari akhir dari sebuah set proses yang berhubungan dengan computer.
	<i>Connector</i>	Digunakan untuk menggambarkan hubungan <i>entry</i> data atau keluar ujung dari <i>flowchart</i> .
	<i>Offpage Connector</i>	Digunakan untuk menggambarkan hubungan dari <i>flowchart</i> bersambung ke beberapa halaman.

Tabel 2.6. Variant Tipe Data Integer.

Tipe	Jangkauan	Memori (Byte)
<i>Byte</i>	0 - 255	1
<i>Word</i>	0 - 65535	2
<i>ShortInt</i>	-128 - 127	1
<i>SmallInt</i>	-32768 - 32767	2
<i>Integer</i>	-2147483648 - 2147483647	4
<i>Cardinal</i>	0 - 2147483647	4
<i>LongInt</i>	-2147483647 - 2147483647	4

Tabel 2.7 Variant Tipe Data Real.

Tipe	Jangkauan	Memori
<i>Real</i>	$\pm 2.9 \cdot 10^{-39}$ sampai $\pm 1.7 \cdot 10^{38}$	6
<i>Single</i>	$\pm 1.5 \cdot 10^{-45}$ sampai $\pm 3.4 \cdot 10^{38}$	4
<i>Double</i>	$\pm 5.0 \cdot 10^{-324}$ Sampai $\pm 1.7 \cdot 10^{308}$	8
<i>Extended</i>	$\pm 3.4 \cdot 10^{-4932}$ Sampai $\pm 1.1 \cdot 10^{4392}$	10
<i>Comp</i>	-2^{63} Sampai $2^{63}-1$	8

Tabel 2.8. Variant Tipe Data Boolean.

Tipe	Memori
<i>Boolean</i>	1
<i>ByteBool</i>	1
<i>Bool</i>	2
<i>WordBool</i>	2
<i>LongBool</i>	4

Tabel 2.9. Variant Tipe Data Char.

Tipe	Ukuran (byte)	Isinya
<i>ANSIChar</i>	1	1 Karakter ANSI
<i>WideChar</i>	2	1 Karakter Unicode
<i>Char</i>	1	Sama dengan WideChar

Tabel 2.10. Variant Tipe Data Char.

Tipe	Panjang	Isi
<i>ShortString</i>	255	ANSIChar
<i>ANSIString</i>	Sampai 3 GB	ANSIChar
<i>String</i>	255 atau samapai 3 GB	ANSIChar
<i>WideString</i>	Sampai 1,5 GB	WideChar

a. Operasi Aritmatik Binari

Tabel 2.11. Operator Aritmetik Binari

Operator	Operasi	Tipe Data	Tipe Hasil
+	Penambahan	<i>Integer</i>	<i>Integer</i>
		<i>Real</i>	<i>Real</i>
-	Pengurangan	<i>Integer</i>	<i>Integer</i>
		<i>Real</i>	<i>Real</i>
*	Perkalian	<i>Integer</i>	<i>Integer</i>
		<i>Real</i>	<i>Real</i>
/	Pembagian	<i>Integer</i>	<i>Integer</i>
		<i>Real</i>	<i>Real</i>
Div	Pembulatan Bulat	<i>Integer</i>	<i>Integer</i>
		<i>Real</i>	<i>Real</i>
Mod	Sisa Pembagian	<i>Integer</i>	<i>Integer</i>

b. Operasi Aritmatik Unari

Tabel 2.12. Operator Aritmetik Unari

Operator	Operasi	Tipe Data	Tipe Hasil
+	Identifikasi tanda	<i>Integer</i>	<i>Integer</i>
		<i>Real</i>	<i>Real</i>
-	Negasi tanda	<i>Integer</i>	<i>Integer</i>
		<i>Real</i>	<i>Real</i>

a. Operasi Logika

Tabel 2.13. Operator Logika

Operator	Operasi	Tipe Data	Tipe Hasil
<i>Not</i>	Negasi Level bit	<i>Integer</i>	<i>Boolean</i>
<i>And</i>	And Level bit	<i>Integer</i>	<i>Boolean</i>
<i>Or</i>	Or Level bit	<i>Integer</i>	<i>Boolean</i>
<i>Xor</i>	Xor Level bit	<i>Integer</i>	<i>Boolean</i>
<i>Shl</i>	Geser ke kiri Level bit	<i>Integer</i>	<i>Boolean</i>
<i>Shr</i>	Geser ke kanan Level bit	<i>Integer</i>	<i>Boolean</i>

b. Operator Boolean

Tabel 2.14. Operator Boolean

Operator	Operasi	Tipe Data	Tipe Hasil
<i>Not</i>	<i>Negasi</i>	<i>Boolean</i>	<i>Boolean</i>
<i>And</i>	<i>And logika</i>	<i>Boolean</i>	<i>Boolean</i>
<i>Or</i>	<i>Or logika</i>	<i>Boolean</i>	<i>Boolean</i>
<i>Xor</i>	<i>Xor logika</i>	<i>Boolean</i>	<i>Boolean</i>

c. Operator Relasional

Tabel 2.15. Operator Relasional

Operator	Operasi
=	Sama dengan
◇	Tidak Sama dengan
<	Lebih Kecil
>	Lebih Besar
<=	Lebih kecil sama dengan
>=	Lebih besar sama dengan
<i>In</i>	Keanggotaan

d.Urutan Operator

Tabel 2.14. Urutan Operator

Operator	Urutan	Kategori
@, <i>not</i>	Pertama	Operator unari
<i>*</i> , <i>/</i> , <i>div</i> , <i>mod</i> , <i>and</i> , <i>shl</i> , <i>shr</i> , <i>as</i>	kedua	Operator perkalian
<i>+</i> , <i>-</i> , <i>or</i> , <i>xor</i>	Ketiga	Operator Penambahan
=, ◇, >, <, <=, >=, <i>in</i> , <i>is</i>	Keempat	Operator Relasional

BAB III

ANALISIS KEBUTUHAN SISTEM

Pada Fase analisis desain sistem ini, Penulis menggunakan teknik *observasi*, wawancara dan studi dokumentasi untuk mengeksploitasi dan membuat deskripsi sistem informasi pendidikan. Langkah ini dilakukan untuk menggabungkan sub fase *defining system* dengan *user requirement*. Karena proyek pengembangan *software* sistem informasi pendidikan ini merupakan proyek” Inisiatif Pengembang “bukan proyek inisiatif *user* sehingga batasan masalah yang akan dilokalisir menjadi sulit ditemukan sebelum survey dilakukan.

Survey yang akan dilakukan ini bertujuan untuk mendefinisikan batasan masalah (fasilitas yang akan dikembangkan), deskripsi sistem, deskripsi lingkungan sistem dan permasalahan-permasalahan yang terdapat di seputar rencana fasilitas sistem. Hasil survey ini merupakan deskripsi *Real system*, *problem sytem solving*, lingkungan system, aturan-aturan system.

Berdasarkan hasil survey di atas dapat disimpulkan bahwa ada beberapa permasalahan dalam dunia pendidikan yang belum diselesaikan secara komperhensif dengan komputerisasi yang masih menggunakan standard manual. Permasalahan-permasalahan tersebut di atas di antaranya :

1. Penerimaan Siswa Baru.
2. Distribusi Kelas.
3. Pemantauan Proses Belajar Mengajar.

4. Pemilihan Jurusan Untuk SMU.
5. Pembuatan Raport.
6. SPP.
7. BP.

Permasalahan di atas akan dijadikan batasan permasalahan dalam pengembangan sistem informasi pendidikan.

3.1. System Definition.

System definition adalah salah satu sub fase dalam fase analisis sistem. Sub Fase ini berisi aktifitas deskripsi sistem dalam bentuk bahasa formal. Ada beberapa item standard yang harus dideskripsikan oleh seorang *Software Engineer* di antaranya:

3.1.1. System User

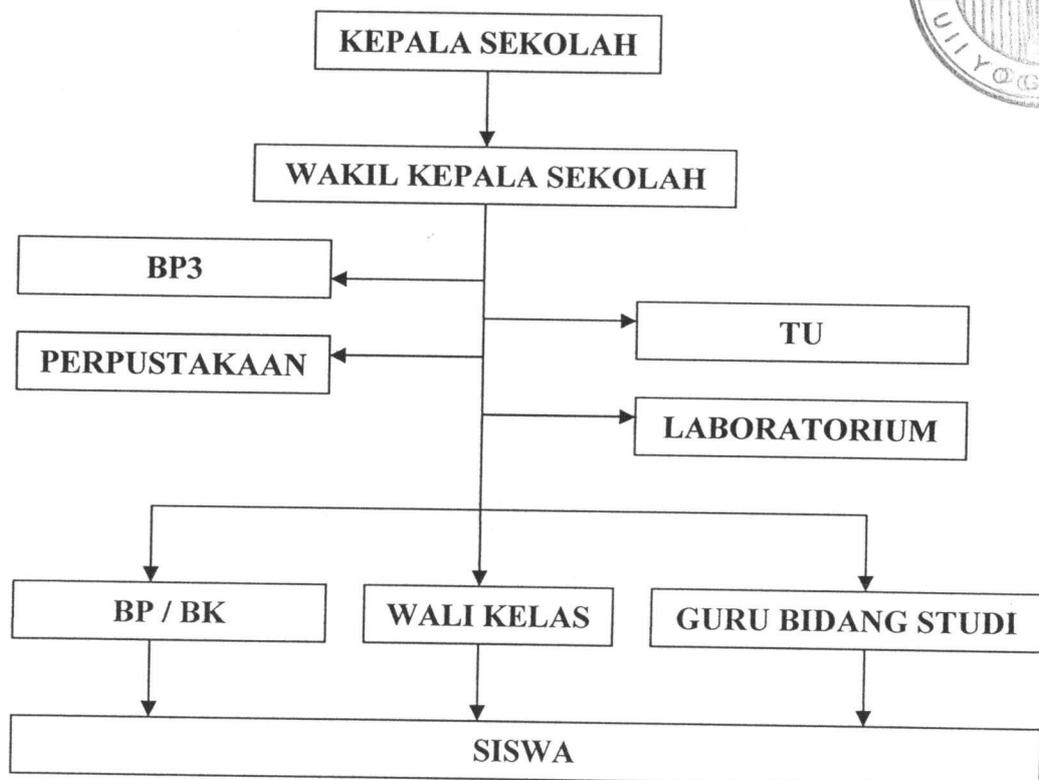
System user adalah semua personel yang menjalankan atau terlibat dalam sebuah sistem. Dalam pengembangan yang terdiri dari :

1. Kepala Sekolah.
2. Wakil Kepala Sekolah.
3. BP3.
4. BP / BK.
5. Guru / Wali Kelas.
6. TU.
7. Siswa.

3.1.2. Analisis Jabatan

Tujuan dari analisis jabatan adalah untuk mempelajari jabatan-jabatan dari masing-masing *system user* yang berkaitan dengan sistem yang akan dikembangkan. Jabatan-jabatan *system user* dapat dideskripsikan dalam sebuah diagram gambar sebagai berikut :

STRUKTUR ORGANISASI SEKOLAH



Gambar 3.1. Struktur Organisasi Sekolah

3.2. System Process

System Process adalah rangkaian aktifitas yang harus dikerjakan dengan aturan – aturan tertentu. Dalam Pengembangan *software* sistem informasi pendidikan.

System proses ini berdasarkan batasan masalah (rencana fasilitas yang akan dikembangkan), yaitu :

1. Proses penerimaan siswa baru

Penerimaan siswa baru dilakukan secara berkala setiap tahun sekali, yaitu pada tahun ajaran baru. Tahapan dalam Proses penerimaan siswa baru dapat dideskripsikan sebagai berikut :

- a. Menentukan Waktu Pendaftaran dan Syarat – syarat pendaftaran siswa baru.
- b. Melakukan proses pendaftaran siswa baru.
- c. Melakukan proses seleksi berdasarkan kriteria tertentu (dengan menggunakan nilai UAN, Test dan besar sumbangan).
- d. Melakukan pengumuman hasil seleksi.
- e. Membuat laporan tentang kegiatan penerimaan siswa baru.

2. Pembagian Kelas (Distribusi Kelas)

Distribusi kelas merupakan salah satu permasalahan dalam penyelenggaraan proses pendidikan. Proses ini dibedakan menjadi dua yaitu : distribusi kelas untuk siswa baru dan distribusi kelas untuk siswa lama. Untuk proses distribusi kelas baru, tahap proses yang harus dilalui adalah memberikan nomor induk, menentukan kapasitas masing - masing kelas dan kemudian membaginya berdasarkan aturan yang telah ditetapkan dalam *system rules*. Sedangkan untuk siswa lama proses yang

harus dilakukan adalah menetapkan kapasitas kelas dan membaginya berdasarkan aturan yang ditetapkan dalam *system rules*.

3. Pemantauan hasil proses belajar mengajar

Pemantauan terhadap hasil proses belajar mengajar merupakan proses yang cukup rumit, karena tolak ukur yang bisa ditetapkan untuk menentukan indikasi tingkat keberhasilan belajar mengajar hanya dengan menggunakan evaluasi, sehingga tahap proses belajar mengajar ini diawali dengan pencatatan hasil evaluasi harian, evaluasi catur wulan (cawu), evaluasi semester atau evaluasi akhir tahun (EBTA). Kemudian membuat ukuran-ukuran statistik dalam menentukan indikasi tingkat keberhasilan proses belajar mengajar seperti daya serap, nilai rata – rata, nilai tertinggi, nilai terendah dan lain sebagainya. Kemudian menyusun laporan dalam format tertentu tentang hasil yang dicapai dalam proses belajar mengajar.

4. Pemilihan Jurusan.

Pemilihan jurusan merupakan fasilitas khusus untuk sekolah menengah umum (SMU). Proses ini biasanya diawali dengan pendataan minat setiap siswa terhadap jurusan tertentu, kemudian menetapkan jumlah daya tampung masing-masing jurusan dengan mempertimbangkan potensi sekolah. Setelah itu baru dilakukan pemilihan jurusan dengan pertimbangan daya tampung masing-masing jurusan, minat dan bakat siswa.

5. Pembuatan Raport

Pembuatan raport merupakan tugas rutin yang harus dikerjakan oleh setiap wali kelas di akhir tahapan pendidikan. Proses ini diawali dengan memasukan daftar nilai hasil ulangan, tugas, dan nilai sub sumatif dengan menggunakan rumus matematika tertentu. Hasil perhitungan nilai raport ini ditampilkan ke dalam format laporan tertentu. Dalam fasilitas ini tidak ditampilkan format raport tertentu karena mengingat ada perbedaan kurikulum yang dipakai untuk kelas-kelas tertentu (kelas satu menggunakan kurikulum KBK sedangkan kelas dua dan kelas tiga menggunakan kurikulum lama)

6. SPP

Pembayaran spp Merupakan tugas rutin yang harus dikerjakan oleh TU / Bendahara administrasi sekolah. Proses ini diawali dengan pembayaran SPP oleh siswa. Hasil dari sistem ini adalah pemantauan pembayaran SPP otomatis.

7. BP

Sistem pemantauan siswa atau bimbingan dan konseling yang dilakukan oleh guru BP, merupakan tugas rutin Insidental. Proses ini diawali sewaktu guru BP menentukan adanya siswa yang perlu diberikan bimbingan dengan sistem ini daya Track Record siswa dapat di Retrieve secara cepat.

Dalam desain sistem informasi pendidikan ini *rules-rules* yang akan dicapai dalam pengembangan sistem, antara lain :

1. Fasilitas Penerimaan Siswa Baru (PSB)

Dalam Fasilitas Penerimaan Siswa Baru (PSB) digunakan *rules* sebagai berikut :

- a. Nomor Pendaftaran harus *Unique* untuk setiap calon siswa dan diusahakan nomor pendaftaran memuat informasi tentang jumlah formulir yang telah keluar.
- b. Sistem Jurnal berdasarkan nilai UAN (hanya berlaku untuk SMU).
- c. Seleksi penerimaan siswa baru didasarkan atas beberapa kriteria yaitu nilai UAN, Test dan Besar Sumbangan.
- d. Siswa yang berhak diterima merupakan siswa yang berada pada range daya tampung dalam data penerimaan siswa baru yang diurutkan dengan kriteria di atas

2. Fasilitas Pembagian Kelas

Pada Fasilitas pembagian kelas (Distribusi Kelas) diberlakukan *rules* sebagai berikut :

- a. Untuk Siswa baru sebelum diberlakukan pengurutan data berdasarkan nama siswa kemudian diberi nomor induk siswa (*Unique* untuk setiap siswa).
- b. Distribusi kelas ini menggunakan dua metode yaitu homogen (siswa yang memiliki kecerdasan yang hampir sama dikumpulkan dalam

kelas yang sama) dan heterogen (siswa dikumpulkan secara merata).

- c. Untuk siswa baru pengelompokan ini mengacu pada nilai UAN, Test atau ijasah. Sedangkan untuk siswa lama pengelompokannya dengan menggunakan nilai raport.

3. Fasilitas Pemantauan Proses Hasil Belajar

Pemantauan proses hasil belajar merupakan suatu deskripsi dari rangkuman proses hasil belajar yang diwakili daftar nilai setiap mata pelajaran. *Rules* yang berlaku pada fasilitas ini hanya seperti rata-rata dan daya serap (perhitungan Prosentase).

4. Fasilitas Pembuatan Raport.

Rules yang digunakan dalam pengembangan fasilitas pembuatan raport otomatis di antaranya :

- a. Membuat daftar nilai untuk setiap siswa dengan menggunakan rumus tertentu yang melibatkan 3 Variabel yaitu : ulangan harian, nilai tugas harian dan nilai Subsumatif.
- b. Nilai di bawah rata-rata dapat dikontrol dengan pertimbangan-pertimbangan akademis tertentu dan harus dikoordinasikan dengan guru bidang studi terkait.

5. Fasilitas SPP

Dalam Pencatatan SPP aturan-aturan yang diberikan sebagai berikut :

- a. Melakukan pencatatan terhadap siswa yang membayar SPP.

- b. Menampilkan kembali daftar siswa yang sudah membayar SPP dan siswa yang belum membayar SPP dalam waktu tertentu

6. Fasilitas BP / BK

Aturan-aturan yang diberikan dalam fasilitas ini sebagai berikut :

- a. Melakukan Pencatatan siswa yang konsultasi dengan BP / BK.
- b. Menampilkan kembali track record siswa selama konsultasi dengan BP / BK.

3.3. System Problem And Solution

Permasalahan Sistem Manual dalam pengelolaan informasi pendidikan di sekolah-sekolah pada umumnya dapat dikelompokkan sebagai berikut :

1. Penerimaan Siswa Baru

Permasalah yang timbul dalam penerimaan siswa baru antara lain :

a. Pemberian nomor urut pendaftaran

Dengan menggunakan sistem manual, panitia PSB harus memberikan nomor pada setiap formulir pendaftaran. Pekerjaan ini selain membosankan terkadang sering juga terjadi kesalahan.

Solusi dari permasalahan ini adalah mengimplementasikan sistem penomoran otomatis pada *software* Sistem Informasi Pendidikan.

b. Pembuatan Jurnal

Pembuatan jurnal merupakan permasalahan rumit membutuhkan waktu yang relatif lama. Pembuatan jurnal merupakan fasilitas yang sangat

penting terutama pada menit-menit terakhir menjelang penutupan PSB.

Untuk itu dituntut kecepatan dalam pembuatan jurnal PSB.

Solusi permasalahan ini ada dua dengan menggunakan sistem manual maka solusinya dengan manajemen arsip yaitu dengan mengelompokkan siswa yang memiliki nilai UAN sama, sehingga dapat mempermudah proses perhitungannya. Yang kedua adalah mengimplementasikan sistem jurnal otomatis pada *software* Sistem Informasi Pendidikan. Dengan menggunakan Sistem otomatis ini maka panitia PSB tidak perlu melakukan perhitungan, penulisan ataupun pengetikan jurnal, semuanya sudah digenerate secara otomatis.

- c. Pengurutan Data Pendaftar dengan kriteria-kriteria tertentu di antaranya : Nilai UAN, nilai test dan sumbangan.

Dengan menggunakan sistem manual pengurutan data dengan menggunakan kriteria tertentu merupakan hal yang sangat sulit apalagi jumlah data yang harus diurut sangat besar.

Solusi dengan menggunakan sistem manajemen arsip yang baik, tetapi solusi ini sangat sulit diimplementasikan jika kriteria yang digunakan selain nilai UAN. Apalagi jika menggunakan kriteria ganda (Nilai UAN, Test dan Nilai Sumbangan). Dengan menggunakan program aplikasi Lotus atau Excell, menggunakan program aplikasi ini proses penggurutannya lebih mudah dibandingkan dengan sitem manual, tetapi proses ini harus didahului dengan proses entry data,

dan kelemahannya dari solusi ini data yang telah *dientry* akan mubazir karena tidak dapat digunakan dalam proses-proses selanjutnya.

Solusi lainnya dengan menggunakan *software* khusus yang didesain untuk pengembangan sistem informasi pendidikan maka semua aplikasi di atas dapat di*handle* dengan baik.

2. Distribusi Kelas

Permasalahan di seputar pembagian kelas dengan menggunakan sistem manual diantaranya :

a. Pemberian nomor induk siswa baru

merupakan permasalahan yang relatif sulit diimplementasikan dengan menggunakan sistem manual.

Solusi dari permasalahan ini dengan menggunakan sistem komputerisasi yang didesain meng*handle* permasalahan pemberian nomor induk bagi siswa baru.

b. Distribusi Kelas

Dengan menggunakan sistem manual pembagian kelas relatif sulit diimplementasikan dengan sistem manual, apalagi jika dengan metode heterogen atau dengan menggunakan metode homogen.

Dengan menggunakan sistem otomatisasi, proses distribusi kelas dengan menggunakan metode heterogen maupun homogen menjadi

sangat mudah tanpa harus mengumpulkan variabel penentu seperti nilai test, nilai UAN ataupun nilai Raport.

3. Pemantauan Proses Belajar Mengajar.

Dengan menggunakan sistem manual, pemantauan proses belajar mengajar sangat sulit diberlakukan, karena sampai saat ini tolak ukur yang diberlakukan dalam menentukan tingkat keberhasilan proses belajar mengajar adalah tingkat nilai rata-rat siswa dan daya serap dalam setiap bidang studi. Dan sampai saat ini Laporan hasil evaluasi harian sering tidak “ dikerjakan “ secara profesional sehingga pemantauan proses belajar mengajar menjadi terlambat.

Solusi dari permasalahan ini dengan mengimplementasikan sistem otomatisasi yang berbasis komputer dengan penanganan masalah ini. Dan juga harus dilakukan pembenahan struktur organisasi dengan menambahkan satu staff baru yang bertugas sebagai personel *input* data hasil evaluasi harian setiap mata pelajaran.

4. Pembuatan Raport

Pembuatan raport merupakan permasalahan yang sulit bagi setiap guru bidang studi tertentu. Permasalahan ini menyangkut pembuatan nilai raport yang digenerate dari nilai ulangan harian, nilai tugas dan nilai sumatif. Kemudian nilai ini akan ditentukan berapa persen kontribusinya ke dalam raport.

Solusi dari permasalahan ini dengan memberlakukan sistem otomatisasi berbasis komputer. Dengan menggunakan sistem otomatisasi ini maka segala perhitungan dalam proses pembuatan raport dapat digenerate secara otomatis. Disamping itu sekolah dapat melakukan eksperimen dengan memasukan nilai koefisien masing-masing variabel untuk menentukan nilai rata-rata tertinggi.

5. Pemilihan Jurusan

Pemilihan jurusan selama ini diimplementasikan dengan menggunakan sistem manual. Proses menggunakan manual ini sangat sulit diimplementasikan karena harus mensortir potensi, bakat dan minat siswa kemudian membandingkan dengan jumlah daya tampung yang tersedia untuk masing-masing jurusan.

Solusi dari permasalahan ini hanya dengan menggunakan sistem otomatisasi berbasis komputer. Dengan menggunakan sistem ini maka pencarian data yang digunakan dalam pemilihan jurusan seperti potensi akademik (diwakili oleh raport), minat dan bakat (Tes IQ) dapat dilakukan dengan mudah tanpa harus melakukan perhitungan apapun.

6. SPP

Permasalahan administrasi SPP selam ini dilakukan secara manual sehingga proses pemantauan siswa yang belum membayar SPP sangat sulit dilakukan.

Solusi dari permasalahan ini adalah dengan mengimplementasikan sistem dan formatisasi yang didesain khusus untuk *handle* permasalahan ini.

7. BP

Pencatatan aktifitas bimbingan dan konseling yang dilakukan oleh guru BP selama ini dilakukan secara manual, sehingga *retrieve* dengan hasil konseling sangat sulit dilakukan termasuk latar belakang siswa.

Solusi dari permasalahan ini adalah dengan mengimplementasikan sistem otomatisasi.

3.4. System Tools

Peralatan yang digunakan dalam implementasi sistem adalah perangkat keras (*hardware*), sistem operasi dan sistem informasi pendidikan. *System Tools* ini didefinisikan dengan mengacu pada kriteria terendah dari *hardware* dan sistem operasi hasil survey.

3.5. System Plans.

Dengan mengimplementasikan sistem informasi pendidikan ini maka diharapkan maka dalam jangka pendek segala proses manualisasi sistem informasi pendidikan dapat diganti dalam proses otomatisasi sehingga dapat meningkatkan efektifitas dan efisiensi dalam pengelolaan sistem pendidikan. Sedangkan untuk jangka panjangnya diberlakukan sistem otomatisasi ini maka proses belajar mengajar

dapat dimonitor dengan baik sehingga dapat meningkatkan proses belajar mengajar sekolah.



BAB IV

DESAIN SISTEM

Setelah *real system* dideskripsikan secara detail ke dalam bahasa formal maka fase selanjutnya adalah membuat desain system dengan mengacu deskripsi *system* yang dibuat dalam fase analisis *system*. Tujuan dari fase desain sistem ini adalah membuat *blue print* yang lengkap sehingga dapat dijadikan *guideline* dalam fase-fase selanjutnya terutama dalam fase *coding*. Fase desain ini merupakan fase translasi kepentingan-kepentingan *customer* yang dirangkum dalam deskripsi sistem ke dalam bahasa spesifik yang dimengerti oleh *programmer*. Sehingga diperlukan keahlian khusus dari sistem analis dalam mengeksplorasi spesifikasi sistem dan menselaraskannya dengan potensi yang dimiliki oleh *programmer*.

Menurut Fairley ada 3 tahap dalam fase desain sistem yang harus dilakukan oleh *software engineer*, yaitu : *external design*, *Architectural design* dan *detail design*. Tiga tahap ini merupakan suatu proses abstrak yang sulit didefinisikan batasan kronologisnya sehingga pada pelaksanaannya tahapan ini hanya didasarkan pada *goal* dari setiap tahapan desain sistem saja.

Dalam desain sistem informasi pendidikan ini mekanisme desain sistem mengacu pada tiga tahap desain sistem di atas, yaitu :

4.1. Tahap *External Design*

Pada tahap *external design* ini seorang *software engineer* harus menyusun, merencanakan dan menentukan karakteristik *external* dari suatu produk *software*.

Karakteristik external suatu produk *software* meliputi : desain tampilan program, desain format report, sistem penyimpanan data, karakteristik fungsional dan *performance requirements*.

Dalam tahap *external* desain dari fase desain sistem informasi pendidikan ada ini mengacu pada teoritika yang dikemukakan oleh fairley, yaitu dengan merumuskan hal-hal sebagai berikut :

Desain Tampilan

4.1.1. Tampilan Program Utama

Tampilan utama (menu utama) program system informasi pendidikan menggunakan *pulldown menu*. Dengan menggunakan *pulldown menu* semua sub modul yang didesain yang berisi semua fasilitas system *dibinding*, sehingga *user* tinggal melakukan *click* terhadap salah satu menu untuk mengaktifkan suatu fasilitas.

Dalam menu utama ada beberapa informasi penting yang harus ditampilkan di antaranya : judul program (system informasi pendidikan), desainer program dan tanggal system. Informasi ini diharapkan dapat mendeskripsikan system dan dapat dijadikan legitimasi penggunaan system.

Ilustrasi *lay out* tampilan program utama sistem informasi pendidikan dapat dilihat pada Gambar 4.1 Lay Out Program Utama.

4.1.2. Tampilan PSB (Penerimaan Siswa Baru)

Dalam sub system Penerimaan Siswa Baru ada beberapa fasilitas yang *included* dalam system di antaranya :

❖ Tampilan Fasilitas Input PSB

Tampilan Input PSB merupakan fasilitas input data-data yang berhubungan dengan penerimaan siswa baru. Data-data ini digunakan dalam proses seleksi penerimaan siswa baru maupun pada fasilitas-fasilitas selanjutnya. Mengingat data PSB merupakan data awal dari semua proses di sekolah. Data-data yang digunakan dalam proses PSB ini di antaranya :

- Data Pribadi Siswa

Data pribadi siswa ini meliputi : nomor pendaftaran, nama calon siswa, jenis kelamin, tempat dan tanggal lahir, alamat, anak keberapa, jumlah saudara, agama, asal sekolah, alamat sekolah, tahun lulus, prestasi calon siswa, pengalaman dan hobby calon siswa.

- Data prestasi akademik

Data prestasi akademik meliputi : asal sekolah, nilai UAN, nilai hasil test.

- Data Orang tua dan Wali

Data orang tua dan wali meliputi : nama orang tua atau wali, pekerjaan, pendidikan, penghasilan per bulan dan alamat orang tua atau wali.

- Data tambahan

Data tambahan ini merupakan “optional” (tidak setiap sekolah menggunakan data tambahan ini) yang berisi : Data besar sumbangan

pendidikan dan besar SPP per bulan maupun data prestasi non akademik seperti prestasi olahraga dan seni.

❖ Tampilan Fasilitas Edit

Pada dasarnya tampilan fasilitas Edit sama dengan fasilitas Input hanya saja pada fasilitas edit ada tampilan seluruh data dan ada fasilitas select yang digunakan untuk memilih data yang akan diedit. Sedangkan untuk fasilitas simpan pada dasarnya menggunakan fasilitas hapus dan kemudian menyimpan data baru.

❖ Tampilan Data PSB dan Hasil Seleksi

Data PSB ditampilkan dengan menggunakan dua macam tampilan, yaitu tampilan jurnal dan tampilan keseluruhan seperti data pribadi calon siswa, data prestasi akademik dan data orang tua atau wali. Tampilan jurnal digunakan khusus bagi sekolah yang menggunakan UAN sebagai "variabel penentu" hasil seleksi PSB. Sehingga data yang ditampilkan dalam jurnal merupakan deskripsi dari rangkuman data calon siswa yang mendaftar. Deskripsi itu memuat 3 informasi, yaitu : Range UAN, Jumlah, dan Frekuensi kumulatif. Disamping itu sebagai informasi tambahan ditampilkan juga : UAN tertinggi, UAN terendah, UAN rata-rata dan jumlah pendaftar. Untuk seleksi menggunakan beberapa variable seperti test, nilai ijazah, nilai UAN dan nilai sumbangan diperlukan suatu formulasi matematis untuk menentukan bobot dari setiap variable.

Tampilan yang akan digunakan untuk merepresentasikan semua data calon siswa dan semua data variable penentu seleksi penerimaan siswa baru digunakan komponen visual *DataGrid* yang akan di-*binding*-kan dengan bahasa SQL (*structured Query Language*). Ilustrasi Lay Out Jurnal dapat dilihat pada Gambar 4.2 Lay Out Tampilan Jurnal.

4.1.3. Tampilan Distribusi Kelas

Tampilan distribusi kelas hanya pada tampilan hasil proses distribusi kelas, sedangkan tampilan input tidak begitu prinsip karena hanya memasukkan jumlah kelas dan kapasitas setiap kelas. Sedangkan untuk hasil dari proses distribusi kelas maka data yang ditampilkan nomor absen, nomor induk dan nama siswa untuk setiap kelas. Ilustrasi Tampilan Distribusi Kelas dapat dilihat pada Gambar 4.3 Lay Out Tampilan Distribusi Kelas.

4.1.4. Tampilan Pemantauan Proses Belajar Mengajar

Pada fasilitas Pemantauan proses Belajar mengajar tampilan yang didesain meliputi :

❖ Tampilan Input

Tampilan input nilai dalam fasilitas ini menggunakan *looping* untuk setiap siswa dalam satu kelas. Sehingga *user* tinggal memasukkan nilai dari setiap pelajaran tanpa harus memasukkan nomor induk atau nomor absen masing-masing siswa. Tampilan Input ini juga digunakan pada fasilitas Edit. Ilustrasi *Lay Out* Input Nilai dapat dilihat pada Gambar 4.4 Lay Out tampilan Input Nilai Ulangan Harian

❖ Tampilan *Output* Hasil Pemantauan Proses Belajar Mengajar

Ada beberapa macam tampilan sebagai *output* dari hasil pemantauan proses belajar, yaitu : Tampilan nilai ulangan setiap siswa untuk seluruh mata pelajaran, Tampilan daya serap setiap kelas untuk setiap pelajaran dan grafik daya serap. Ilustrasi Tampilan nilai ulangan setiap siswa dapat dilihat pada Gambar 4.5 Lay Out Tampilan Input Nilai Ulangan Harian Per Matapelajaran.

Sedangkan Ilustrasi Tampilan Input Nilai Ulangan Harian Per MataPelajaran dapat dilihat pada Gambar 4.6 Lay Out Tampilan Input Nilai Ulangan Harian Per Kelas

4.1.5. Tampilan Raport

Ada beberapa tampilan dalam fasilitas raport, diantaranya : tampilan input nilai ulangan akhir (test sumatif) dan nilai tugas. Tampilan ini pada prinsipnya sama dengan tampilan fasilitas input pemantauan proses belajar mengajar. Sedangkan tampilan *Output* fasilitas raport menampilkan informasi perhitungan raport dengan menggunakan formula matematis tertentu.

Ilustrasi *Lay Out* Raport dapat dilihat pada Gambar 4.7. Lay Out Nilai Raport.

4.1.6. Tampilan Pemilihan Jurusan

Tampilan Pemilihan jurusan dikonsentrasikan pada tampilan input yang terdiri dari 2 input : input test IQ dan input minat. Dua form ini sebenarnya

menggunakan format desain yang sama dengan input nilai ulangan harian pada fasilitas pemantauan hasil proses belajar mengajar. Sedangkan tampilan pemilihan jurusan dilakukan dengan cara manual dengan mempertimbangkan data yang ditampilkan oleh komputer. Ilustrasi *Lay Out* Proses pemilihan Jurusan dapat dilihat pada Gambar 4.8. Lay Out Proses Penjurusan.

4.1.7. Tampilan SPP

Pada fasilitas tampilan SPP difokuskan pada tampilan output dari hasil representasi data statistika pembayaran SPP meliputi :

- ❖ Pembayaran SPP pada tanggal tertentu
- ❖ Daftar siswa yang telah membayar SPP sampai bulan tertentu
- ❖ Daftar siswa yang belum membayar SPP sampai bulan tertentu
- ❖ Jurnal pembayaran SPP dalam bulan tertentu

Sedangkan tampilan input pembayaran SPP menggunakan komponen visual dasar seperti *combobox* dan *edit text*. Koneksi data yang digunakan dengan menggunakan fasilitas komponen visual SQL.

4.1.8. Tampilan BP

Pada fasilitas tampilan BP difokuskan pada tampilan output dari hasil representasi data statistika proses bimbingan dan konseling yang dilakukan oleh guru BP seperti data statistika kasus dengan frekuensinya. Sedangkan untuk fasilitas input dilengkapi dengan track record siswa sehingga

diharapkan ada kesinambungan dalam setiap penanganan permasalahan siswa.

4.2 Tahap Architectural Design

Pada tahap *Architectural design* ini seorang *software engineer* harus menyusun, merencanakan dan menentukan perancangan data base, fungsi-fungsi internal yang berhubungan dengan fasilitas program, dekomposisi *high level function* ke sub fungsi, dan menentukan tingkat *interdependensi* antar sub fungsi.

Dalam tahap *Architectural design* dari fase desain sistem informasi pendidikan ini, diperlukan implementasi teoritis *architectural design* yang digunakan sebagai landasan dalam fase selanjutnya (*fase coding*). Ada beberapa hal yang harus didesain dalam tahap *architectural design* ini di antaranya :

4.2.1 Desain Data Base

Dalam desain data base sistem informasi pendidikan ini, mengacu pada fasilitas-fasilitas yang akan dikembangkan dalam sistem. Ini disebabkan setiap fasilitas yang akan dikembangkan mempunyai struktur data dan fungsi yang berbeda (hubungan antar satu fasilitas dengan fasilitas yang lain sangat kecil).

Keadaan ini menyebabkan perlunya satu data base atau lebih untuk satu fasilitas.

Data base yang digunakan dalam program system informasi pendidikan degenerate dengan menggunakan SQL dengan mengacu pada format database DBASE 4 dan Paradox

Pada tahap desain data base ini data yang diperlukan dalam sistem informasi pendidikan di antaranya :

4.2.1.1 Fasilitas PSB

Pada fasilitas PSB ini data dibagi menjadi 7 bentuk data yaitu :

a) Data pribadi calon siswa

Data pribadi calon siswa ini disimpan dalam suatu file dengan system penamaan : PSBSiswaT+Tahun+.dbf, data ini akan dipakai lagi sewaktu calon siswa tersebut lulus seleksi.hanya saja data pribadi siswa yang lulus seleksi disimpan dalam *file* DataSiswa.dbf walaupun *field-fieldnya* sama kecuali NoDaftar diganti NoInduk.

Data Pribadi Calon Siswa dapat dilihat pada Tabel 4.1.Data Pribadi Calon Siswa

b) Data Nilai Ujian Akhir Nasional

Data Nilai Ujian Akhir Nasional (UAN) ini dijadikan satu *file* khusus dan disimpan dengan format penamaan : PSBNEMT+Tahun+.dbf. Bagi calon siswa yang lulus seleksi maka data UAN ini akan disimpan dalam file dengan nama DataNEM.dbf dengan mengganti *field* NoDaftar menjadi NoInduk.

Data Nilai Ujian Akhir Nasional Calon Siswa dapat dilihat pada Tabel 4.2 Data Nilai Ujian Akhir Nasional Calon Siswa.

c) Data Nilai Ijazah Calon Siswa

Data Nilai ijazah calon siswa ini dijadikan satu file khusus dan disimpan dengan *format* penamaan : PSBIjazahT+Tahun+.dbf. Bagi calon siswa yang lulus seleksi maka data ijazah ini akan disimpan dalam *file* dengan nama DataIjazah.dbf dengan mengganti *field* NoDaftar menjadi NoInduk.

Data Nilai Ijazah Calon Siswa dapat dilihat pada table 4.3 Data Nilai Ijazah Calon Siswa.

d) Data Test Akademik

Data test akademik calon siswa merupakan data yang dijadikan variable utama dalam proses seleksi penerimaan siswa baru. Data test akadeik ini disimpan dalam *file* dengan format penamaan : PSBTestT+Tahun+.Db.

Field-field dalam file test akademik ini Dapat dilihat Pada Tabel Tabel 4.4 Data Kemampuan Akademis Calon Siswa.

e) Data Prestasi Calon Siswa

Data prestasi non akademik calon siswa merupakan pertimbangan dalam seleksi penerimaan siswa baru, prestasi non akademik ini bias berupa prestasi olahraga maupun seni. Fomat penamaan untuk file ini adalah : PSBPestasiT+Tahun+.dbf sedangkan fiel-fieldnya Dapat dilihat pada Tabel 4.5 Data Prestasi Non Akademik Calon Siswa

f) Data Orang Tua Siswa

Data orang tua ini berisi data pribadi orang tua calon siswa, sebenarnya pada proses PSB data orang tua siswa kurang begitu relevan untuk dijadikan salah satu variabel penentu keputusan. Tetapi momen pencatatan data orang tua siswa itu sangat tepat dilakukan sewaktu proses PSB. Dan ini mengacu pada *form-form* PSB yang berlaku di setiap sekolah (berdasarkan hasil *survey*). Hasil pencatatan data orang tua calon siswa ini disimpan dalam file PSBOrtuT+Tahun+.dbf. Bagi calon siswa yang lulus seleksi maka data orang tua calon siswa akan disimpan dalam *file* dengan nama DataOrtu.dbf dengan mengganti field NoDaftar menjadi NoInduk.

Adapun data orang tua siswa dapat dilihat pada Tabel 4.6 Data Orang Tua Calon Siswa.

g) Data Hasil Seleksi & Pendaftaran Ulang

Data calon siswa yang dinyatakan lulus seleksi penerimaan siswa baru dimasukkan dalam data 'pengumuman dan pendaftaran ulang'. site mini digunakan untuk menjaga supaya hanya siswa yang telah dinyatakan lulus saja yang bisa melakukan daftar ulang.

Data Pendaftaran Ulang dapat dilihat pada Tabel 4.7 Data pengumuman Hasil Seleksi PSB dan Pendaftaran Ulang

4.2.1.2 Fasilitas Distribusi Kelas

Pada fasilitas distribusi kelas data yang digunakan ada dua kelompok data, yaitu data kelompok siswa baru (kelas 1) dan data kelompok siswa Lama (kelas 2 dan Kelas 3)

a. Kelompok siswa baru

Data yang digunakan dalam pembagian kelas bagi siswa baru adalah data pendaftaran ulang dan data pribadi siswa (khususnya yang berhubungan dengan potensi akademik seperti UAN dan test).

b. Kelompok Siswa Lama

Data yang digunakan dalam pembagian kelas bagi siswa lama adalah data raport. Khusus bagi kelas 3 SMU ditambah dengan data hasil pembagian jurusan.

Sedangkan data *Output* dari fasilitas PSB ini akan disimpan dalam file SiswaAktif.dbf , Adapun field-field dalam file SiswaAktif.dbf dapat dilihat pada Tabel 4.8. Data Siswa Aktif. :

4.2.1.3 Fasilitas Pemantauan Proses Belajar Mengajar

Pada fasilitas pemantauan proses belajar mengajar berdasarkan pada data dari hasil proses pembagian kelas dan data mata pelajaran. Sedangkan data *output* dari fasilitas belajar mengajar disimpan dalam *file* dengan format penamaan : NUT+Tahun+Semester+.dbf. Adapun

field-field file nilai ulangan harian itu Dapat dilihat pada Tabel 4.9.

Data Nilai Ulangan Harian.

4.2.1.4 Fasilitas Raport

Pada fasilitas pembuatan raport data yang digunakan adalah data yang didapat dari proses distribusi kelas dan data ulangan. Sedangkan data *output* dari fasilitas pembuatan raport disimpan dalam file dengan format penamaan RaportT+Tahun+Semester+.dbf sedangkan field-fieldnya dapat dilihat Pada Tabel 4.10. Data Nilai Raport.:

4.2.1.5 Data Absensi

Data Absensi adalah data kehadiran siswa dalam satu tahapan proses pendidikan (triwulan atau semester). Data ini diinput setiap hari dan disimpan dalam 2 *file*, yaitu *file* yang digunakan untuk mencatat *track record* absensi siswa dan *file* yang digunakan untuk proses rekapitulasi data absensi siswa. File yang digunakan untuk mencatat *track record* absensi siswa menggunakan format penamaan : AbsenT+tahun+ Semester+.dbf dan struktur datanya Dapat dilihat Pada Tabel 4.11. Data Absensi.

Sedangkan file yang digunakan untuk menyimpan data rekapitulasi absensi siswa menggunakan format penamaan : TotalAbsen+Tahun+ Semester+.dbf

Struktur datanya dapat dilihat Pada Tabel 4.12. Data Rekapitulasi Absensi.

4.2.1.6 Fasilitas Penentuan Jurusan

Data yang digunakan dalam fasilitas ini adalah data yang didapat dari hasil fasilitas pembuatan Raport (nilai raport) dan data minat siswa terhadap jurusan tertentu dan data hasil test IQ. Sedangkan data *output* merupakan data yang digunakan dalam proses distribusi kelas yaitu disimpan dalam file SiswaAktif.dbf.

Data minat siswa disimpan dalam file dengan format penamaan : Minat+Tahun+.dbf. Struktur datanya dapat dilihat pada Tabel 4.13.

Data Minat Siswa Terhadap Jurusan.:

Sedangkan untuk data test IQ disimpan dalam file dengan format penamaan : TestIQ+Tahun+.dbf .

Struktur datanya dapat dilihat pada Tabel 4.14. Data Test IQ Siswa.

4.2.1.7 Data SPP

Data SPP adalah data pembayaran SPP yang didokumentasikan dalam satu tahun ajaran. Data ini diinput setiap hari dan disimpan dalam 2 *file*, yaitu *file* yang digunakan untuk mencatat pembayaran SPP siswa dan *file* yang digunakan untuk proses rekapitulasi data SPP siswa. *File* yang digunakan untuk mencatat pembayaran SPP siswa menggunakan format penamaan : SPPT+tahun+.dat. Struktur datanya dapat dilihat pada Tabel 4.15. Data Pembayaran SPP.

Sedangkan *file* yang digunakan untuk menyimpan data rekapitulasi pembayaran SPP siswa menggunakan format penamaan :

DataSPP+Tahun+Semester+.dbf. Struktur datanya dapat dilihat pada Tabel 4.16. Data Rekapitulasi SPP.

4.2.1.8 Data Guru

Data guru digunakan untuk menentukan guru pengampu setiap matapelajaran. Data guru ini disimpan dalam file dengan nama : pengajar.dbf. Adapun data guru dapat dilihat pada Tabel 4.17. Data Pengajar.

4.2.1.9 Data Matapelajaran

Data matapelajaran merupakan data yang digunakan untuk menampung semua data yang berhubungan dengan matapelajaran. Data matapelajaran ini disimpan dalam file dengan nama Pelajaran.dbf. Struktur data yang digunakan dalam file pelajaran.dbf Dapat dilihat pada Tabel 4.18. Data Matapelajaran.

4.2.2 Saling Ketergantungan (Interdependency)

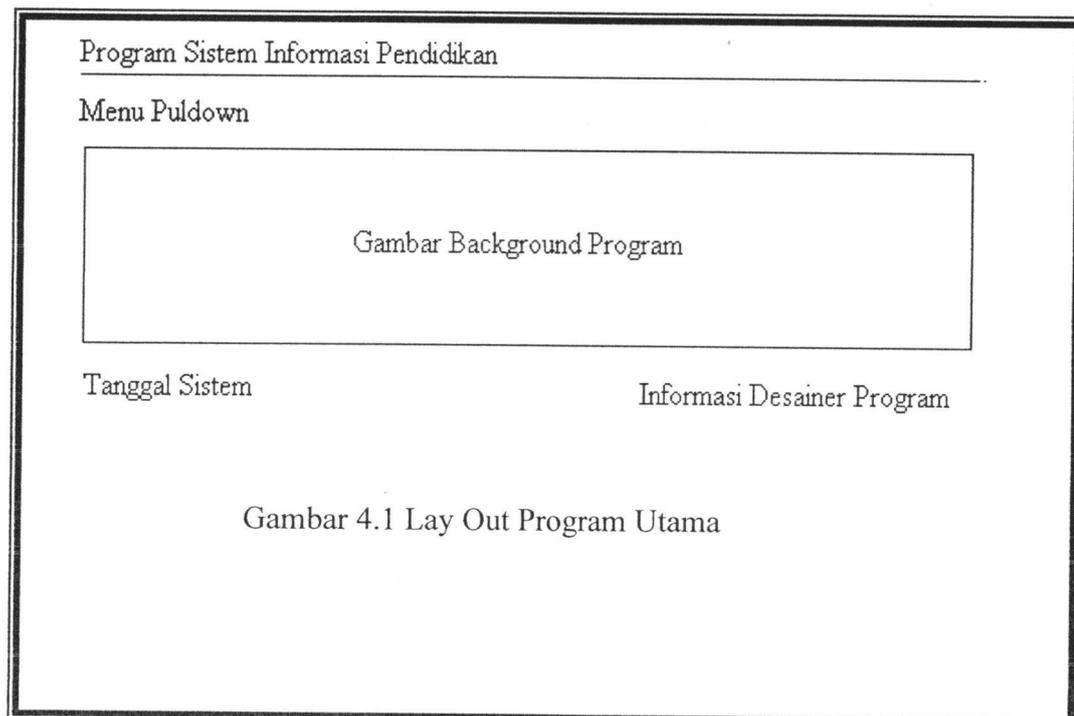
Saling ketergantungan adalah suatu keadaan dimana suatu proses, function atau data membutuhkan proses, *function* atau data lain. Kondisi ini mempengaruhi tingkat kronologis suatu proses, *function* atau data dalam fase *coding* dan fase *testing*.

Pada fase desain sistem menentukan derajat ketergantungan sangat mutlak dilakukan apalagi jika *project* yang dikerjakan mempunyai *size* yang besar dan dikerjakan secara paralel. Dalam desain sistem informasi pendidikan penentuan derajat ketergantungan didasarkan pada ketergantungan data. Dari

uraian data di atas dapat disimpulkan berbentuk Tabel interpedensi. Tabel ini dapat dilihat pada tabel 4.19 Interdepedensi.:

4.3 Desain Detail

Tahap desain detail adalah tahap akhir dalam fase desain sistem. Dalam tahap ini seorang *software engineer* harus menerjemahkan setiap bentuk komponen sistem yang dihasilkan pada tahap dekomposisi ke dalam bentuk yang mudah dipahami oleh *programmer*. Ada dua bentuk yang representatif untuk mendeskripsikan sistem dalam tahap desain detail ini, yaitu : bentuk flowchart dan algoritma program. Kedua bentuk ini yang akan dijadikan pedoman programmer dalam fase *coding* (programmer tinggal menerjemahkan dua bentuk desain ini ke dalam *syntack* bahasa pemrograman yang digunakan dalam pengembangan sistem).



Gambar 4.1 Lay Out Program Utama

Jurnal Penerimaan Siswa Baru

Nama Sekolah

Tahun Ajaran

No	Batas UAN	Jumlah	Frekuensi Kumulatif
1.			
2.			
3.			
4.			
5.			
Jumlah			

UAN Tertinggi :

UAN Terendah :

UAN Rata-rata :

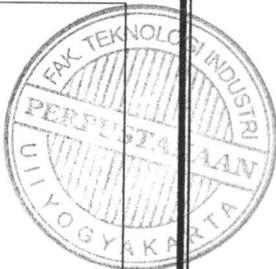
Gambar 4.2 Lay Out Tampilan Jurnal

Nama Sekolah dan Alamat

Nama Kelas

No.	No Induk	Nama Siswa
1.
2.
3.
4.

Gambar 4.3 Lay Out Tampilan Distribusi Kelas



Input Nilai Ulangan Harian Kelas
 Matapelajaran :
 Ulangan ke :

No. Absen	:	
No. Induk	:	
Nama	:	
Nilai	:	

Gambar 4.4 Lay Out tampilan Input Nilai Ulangan Harian

No Induk	Nama Siswa	Ulangan Ke				
		I	II	III	IV	V
....
....
....
....
Rata-Rata	

Rata-rata Total :

Gambar 4.5 Lay Out Tampilan Input Nilai Ulangan Harian Per Matapelajaran

No	Mata Pelajaran	Kelas					
		I-1	I-2	I-3	I-4	...	Rerata
1.	Matematika					
2.	Fisika					
3.	Biologi					
4.
	Rata-Rata					

Gambar 4.6 Lay Out Tampilan Input Nilai Ulangan Harian Per Kelas

Kelas : _____ Matapelajaran : _____

Wali Kelas : _____ Guru Bidang Studi : _____

NIS	Nama	UH	Tugas	Sumatif	Raport	Katrol

Gambar 4.7 Lay Out Nilai Raport

Nomor Induk :

Nama :

IPA	Nilai
Matematika
Fisika
Biologi
Kimia
Rata-Rata
Rangking

IPS	Nilai
Antropologi
Akuntansi
Ekonomi
.....	
Rata-Rata
Rangking

Bahasa	Nilai
B. Indonesia
B. Inggris
.....
.....
Rata-Rata
Rangking

Minat I :

Minat II :

Daya Tampung IPA

Daya Tampung IPS

Daya Tampung Bahasa

Gambar 4.8 Lay Out Proses Penjurusan

Tabel 4.1 Data Pribadi Calon Siswa

No	Field	Tipe Data	Keterangan
1.	Tanggal	Date	Tanggal pendaftaran
2.	NoDaftar	String[10]	Nomor pendaftaran
3.	Nama	String[20]	Nama calon siswa
4.	Kelamin	String[6]	Jenis kelamin calon siswa
5.	Tempat	String[30]	Tempat lahir calon siswa
6.	TglLahir	Date	Tanggal lahir calon siswa
7.	Agama	String[19]	Agama calon siswa
8.	Alamat	String[35]	Alamat calon siswa
9.	Anak	String[2]	Calon siswa dalam keluarga anak keberapa
10.	Saudara	String[2]	Jumlah saudara calon siswa
11.	Hobby	String[50]	Hobby calon siswa
12.	Pengalaman	String[50]	Pengalaman calon siswa (organisasi)
13.	Prestasi	String[50]	Prestasi yang pernah diraih calon siswa
14.	Sekolah	String[50]	Sekolah calon siswa
15.	AlSekolah	String[50]	Alamat sekolah calon siswa
16.	Lulus	String[4]	Tahun kelulusan

Tabel 4.2 Data Nilai Ujian Akhir Nasional Calon Siswa

No	Field	Tipe Data	Keterangan
1.	NoDaftar	String[5]	Nomor pendaftaran
2.	Mapel	Float(5,2)	Jumlah Matapelajaran
3.	PPKN	Float(5,2)	Nilai pelajaran PPKN
4.	Indo	Float(5,2)	Nilai pelajaran bahasa Indonesia
5.	Mat	Float(5,2)	Nilai pelajaran Matematika
6.	IPA	Float(5,2)	Nilai pelajaran IPS
7.	IPS	Float(5,2)	Nilai pelajaran IPS
8.	Ingg	Float(5,2)	Nilai pelajaran Bahasa Inggris
9.	Jumlah	Float(5,2)	Jumlah Nilai UAN
10.	Rata	Float(5,2)	Rata-Rata Nilai UAN

Tabel 4.3 Data Nilai Ijazah Calon Siswa

No	Field	Tipe Data	Keterangan
1.	NoDaftar	String[5]	Nomor pendaftaran
2.	NoIjazah	String[10]	Nomor Ijazah
3.	Mapel	Float(5,2)	Jumlah Matapelajaran
4.	PPKN	Float(5,2)	Nilai pelajaran PPKN
5.	Agama	Float(5,2)	Nilai pelajaran Agama
6.	Indo	Float(5,2)	Nilai pelajaran Bahasa Indonesia
7.	Mat	Float(5,2)	Nilai pelajaran Matematika
8.	IPA	Float(5,2)	Nilai pelajaran IPA
9.	IPS	Float(5,2)	Nilai pelajaran IPS
10.	Ingg	Float(5,2)	Nilai pelajaran Bahasa Inggris
11.	Orkes	Float(5,2)	Nilai pelajaran Olahraga & Kesehatan
12.	Trampil	Float(5,2)	Nilai pelajaran Ketrampilan
13.	Seni	Float(5,2)	Nilai pelajaran Kesenian
14.	Lokal	Float(5,2)	Nilai pelajaran Muatan Lokal
15.	Jumlah	Float(5,2)	Jumlah Nilai ijazah
16.	Rata	Float(5,2)	Rata-Rata Nilai Ijazah

Tabel 4.4 Data Kemampuan Akademis Calon Siswa

No	Field	Tipe Data	Keterangan
1.	NoDaftar	String[5]	Nomor pendaftaran
2.	Pertama	Float(5,2)	Nilai Test Hari Pertama (PPKN & Bahasa Indonesia)
3.	Kedua	Float(5,2)	Nilai Test Hari Kedua (Matematika & IPA)
4.	Jumlah	Float(5,2)	Jumlah nilai Test hari pertama dan kedua
5.	Nilai	Float(5,2)	Nilai dari test dengan penggunaan perbandingan kontribusi hari pertama dan hari kedua

Tabel 4.5 Data Prestasi Non Akademik Calon Siswa

No	Field	Tipe Data	Keterangan
1.	NoDaftar	String[5]	Nomor pendaftaran
2.	Prestasi	String[150]	Prestasi yang pernah diraih oleh calon siswa
3.	Tingkat	String[50]	Tingkat Kejuaraan yang diikuti
4.	Nilai	Float(5,2)	Skor dari prestasi yang diraih siswa

Tabel 4.6 Data Orang Tua Calon Siswa

No	Field	Tipe Data	Keterangan
1.	NoDaftar	String[10]	Nomor pendaftaran
2.	Nama	String[30]	Nama orang tua calon siswa
3.	TglLahir	Date	Tanggal Lahir orang tua calon siswa
4.	Pendidikan	String[20]	Latar belakang pendidikan orang tua
5.	Pekerjaan	String[20]	Pekerjaan orang tua
6.	Agama	String[19]	Agama orang tua calon siswa
7.	Gaji	String[14]	Penghasilan orang tua calon siswa
8.	Alamat	String[50]	Alamat orang tua calon siswa
9.	Ortu	String[1]	Tanda data Bapak, Ibu atau wali

Tabel 4.7 Data pengumuman Hasil Seleksi PSB dan Pendaftaran Ulang

No	Field	Tipe Data	Keterangan
1.	NoDaftar	String[10]	Nomor urut Pendaftaran Ulang
2.	Keputusam	String[10]	Hasil Seleksi Penerimaan siswa Baru
3.	DafUlang	String[1]	Validasi pendaftaran ulang
4.	Tanggal	Date	Tanggal Pendaftaran Ulang

Tabel 4.8 Data Siswa Aktif

No	Field	Tipe Data	Keterangan
1.	NoInduk	String[10]	Nomor Induk Siswa
2.	Nama	String[30]	Nama Siswa
3.	Kelamin	String[9]	Jenis Kelamin Siswa
4.	Kelas	String[1]	Kelas Siswa
5.	Jurusan	String[8]	Jurusan Siswa
6.	Indek	String[2]	Indek Kelas Siswa

Tabel 4.9 Data Nilai Ulangan Harian

No	Field	Tipe Data	Keterangan
1.	NoInduk	String[5]	Nomor Induk Siswa
2.	Mapel	String[20]	Mata pelajaran
3.	I	Float(5,2)	Nilai Ulangan I
4.	II	Float(5,2)	Nilai Ulangan II
5.	III	Float(5,2)	Nilai Ulangan III
6.	IV	Float(5,2)	Nilai Ulangan IV
7.	V	Float(5,2)	Nilai Ulangan V
8.	Ke	Float(5,2)	Counter Ulangan Harian
9.	Jumlah	Float(5,2)	Jumlah Ulangan
10.	Rata	Float(5,2)	Rata-Rata Ulangan Harian

Tabel 4.10 Data Nilai Raport

No	Field	Tipe Data	Keterangan
1.	NoInduk	String[5]	Nomor Induk Siswa
2.	Kode	String[20]	Kode Mata pelajaran
3.	Matapelajaran	String[15]	Matapelajaran
4.	Asli	Float(5,2)	Nilai raport hasil dari perhitungan
5.	Katroi	Float(5,2)	Nilai Katroi yang diberikan oleg guru bidang studi
6.	Nilai	Float(5,2)	Nilai raport

Tabel 4.11 Data Absensi

No	Field	Tipe Data	Keterangan
1.	Tanggal	Date	Tanggal Absent
2.	NoInduk	String[10]	Nomor Induk Siswa
3.	Absen	String[7]	Jenis Absensi siswa dengan 3 kategori (Sakit,Ijin dan Alpa)

Tabel 4.12 Data Rekapitulasi Absen

No	Field	Tipe Data	Keterangan
1.	NoInduk	String[10]	Nomor Induk Siswa
2.	Sakit	Integer	Counter Absen jenis Sakit
3.	Ijin	Integer	Counter Absen jenis Ijin
4.	Alpha	Integer	Counter Absen jenis Alpha

Tabel 4.13 Data Minat Siswa Terhadap Jurusan

No	Field	Tipe Data	Keterangan
1.	NoInduk	String[5]	Nomor Induk Siswa
2.	MinatI	String[20]	Minat pertama terhadap jurusan
3.	MinatII	String[20]	Minat kedua terhadap jurusan

Tabel 4.14 Data Test IQ Siswa

No	Field	Tipe Data	Keterangan
1.	NoInduk	String[5]	Nomor Induk Siswa
2.	IO	String[4]	Hasil test IO

Tabel 4.15 Data Pembayaran SPP

No	Field	Tipe Data	Keterangan
1.	Tanggal	Date	Tanggal Pembayaran SPP
2.	NoInduk	String[10]	Nomor Induk Siswa
3.	Nama	String[30]	Nama Siswa
4.	Keias	String[15]	Keias Siswa
5.	Bulan	String[15]	Bulan yang dibayar
6.	Besar	LongInt	Besar SPP

Tabel 4.16 Data Rekapitulasi SPP

No	Field	Tipe Data	Keterangan
1.	NoInduk	String[10]	Nomor Induk Siswa
2.	Nama	String[30]	Nama Siswa
3.	Kelas	String[15]	Kelas Siswa
4.	Jul	String[1]	Pembayaran Bulan Juli
5.	Ags	String[1]	Pembayaran Bulan Agustus
6.	Sept	String[1]	Pembayaran Bulan September
7.	Okt	String[1]	Pembayaran Bulan Oktober
8.	Nov	String[1]	Pembayaran Bulan November
9.	Des	String[1]	Pembayaran Bulan Desember
10.	Jan	String[1]	Pembayaran Bulan Januari
11.	Feb	String[1]	Pembayaran Bulan Februari
12.	Mar	String[1]	Pembayaran Bulan Maret
13.	Apr	String[1]	Pembayaran Bulan April
14.	Mei	String[1]	Pembayaran Bulan Mei
15.	Jun	String[1]	Pembayaran Bulan Juni
16.	Bulan	String[1]	Bulan terakhir yang dibayar
17.	Akhir	String[1]	Counter Bulan

Tabel 4.17 Data Pengajar

No	Field	Tipe Data	Keterangan
1.	NIP	String[15]	Nomor ID
2.	Nama	String[30]	Nama guru
3.	Tempat	String[15]	Tempat Lahir
4.	TglLahir	Date	Tanggal lahir
5.	Kelamin	String[10]	Jenis kelamin
6.	Agama	String[18]	Agama
7.	Nikah	String[11]	Status pernikahan
8.	Pendidikan	String[10]	Jenjang pendidikan formal
9.	Jurusan	String[20]	Jurusan pendidikan formal
10.	Almamater	String[30]	Nama Almamater
11.	Lulusan	String[4]	Tahun lulusan
12.	StatPeg	String[25]	Status kepegawaian
13.	Jabatan	String[25]	Jabatan
14.	Pangkat	String[15]	Pangkat
15.	Gol	String[6]	Golongan
16.	Dinas	Date	Tanggal mulai dinas
17.	Alamat	String[30]	Alamat

Tabel 4.18 Data Matapelajaran

No	Field	Tipe Data	Keterangan
1.	Kode	String[5]	Kode Matapelajaran
2.	Pelajaran	String[15]	Nama Matapelajaran
3.	Kelas	String[1]	Kelas
4.	Jurusan	String[6]	Jurusan
5.	SKS	Byte	Jam per minggu
6.	Status	String[11]	Status matapelajaran (dicantumkan diraport dan diajarkan, diajarkan saja, raport saja)

Tabel 4.19 Interdependensi

No	Nama Data	Ketergantungan
1.	Data pribadi calon siswa	-
2.	Data UAN calon siswa	-
3.	Data ijazah calon siswa	-
4.	Data pengumuman	-
5.	Data test seleksi	-
6.	Data prestasi calon siswa	-
7.	Data orang tua calon siswa	-
8.	Data guru	-
9.	Data Matapelajaran	-
10.	Data pendaftaran ulang	Data pribadi siswa (menentukan hasil seleksi)
11.	Data pribadi, Ortu, Ijazah, UAN, SPP siswa	Semua data calon siswa dengan data pendaftaran ulang
12.	Data pembagian kelas	Data pendaftaran ulang, data test dan data raport
13.	Data pemantauan pro-ses belajar	Data pembagian kelas & data mata pelajaran
14.	Data test	Data pembagian kelas & data mata pelajaran
15.	Data tugas	Data pembagian kelas & data mata pelajaran
16.	Data Absensi	Data Pembagian kelas
17.	Data Rapor	Data pembagian kelas, data mata pelajaran, data pemantauan proses belajar, data test, data tugas dan data absensi
18.	Data SPP	Data pembagian kelas
19.	Data Minat	Data pembagian kelas
20.	Data IQ	Data pembagian
21.	Data Jurusan	Data pembagian kelas

BAB V

FASE IMPLEMENTASI

Fase implementasi merupakan fase realisasi desain system ke dalam bentuk intruksi (*script*) yang dipahami oleh computer. Proses *translasi* ini di dunia computer lazim dikenal dengan istilah *coding*. Dalam proses *coding* yang perlu diperhatikan adalah bagaimana membuat suatu *script* yang efisien dan efektif sehingga penggunaan *resource* seperti memori maupun *space* dapat dilakukan secara optimal.

Dalam pengembangan system informasi pendidikan ini, fase *coding* dibagi menjadi beberapa tahapan berdasarkan banyaknya fasilitas sistem informasi pendidikan, yaitu :

5.1 Program utama

Dalam fase desain system, program utama terdiri dari : menu utama dengan menggunakan menu pulldown, gambar sebagai latar belakang, judul program, informasi desainer dan tanggal system. Dengan menggunakan Borland Delphi 7.0, program utama dapat dikembangkan dengan menggunakan komponen visual yang ada dalam IDE (*integrated Development Environment*) Borland Delphi 7.0. Hasil dari pengembangan ini dapat dilihat sebagai berikut :



Gambar 5.1 Tampilan Program Utama Sistem Informasi Pendidikan

untuk merealisasikan desain tadi diperlukan langkah-langkah sebagai berikut :

- ❖ Pembuatan judul program, untuk membuat judul program dilakukan dengan merubah property *caption* dalam *form* utama menjadi judul yang diinginkan.
- ❖ Pembuatan menu utama, untuk membuat menu utama dalam bentuk *pull down* menu diperlukan komponen visual Main menu. Dalam IDE Delphi 7.0 komponen visual ini terletak dalam tab standard. Dengan melakukan *double click* pada komponen visual maka secara otomatis akan masuk dalam main menu editor dari fasilitas ini tinggal dilakukan input menu utama dan sub-sub menu.
- ❖ Pembuatan latar belakang program, latar belakang program dibuat dengan menggunakan komponen visual *Image*, dengan mengganti property

picture dengan *path* suatu gambar maka akan otomatis image akan menampilkan gambar yang terdapat dalam *path*. Untuk menampilkan gambar sesuai dengan ukuran yang diinginkan property *stretch* harus diisi dengan nilai *true*.

- ❖ Untuk menampilkan tanggal system diperlukan komponen visual panel dengan merubah property *caption* pada komponen visual panel dengan statement sebagai berikut :

```
Panel2.Caption := FormatDateTime('dddd, dd - mmmm - yyyy', Now);
```

- ❖ Untuk membuat tulisan desainer program digunakan komponen visual label dengan merubah property *caption*, sedangkan untuk membuat animasinya diperlukan komponen timer dengan mengeset property *interval = 1* dan membuat *procedure* events *OnTimer* sebagai berikut :

```
procedure TProgUtama.Timer1Timer(Sender: TObject);
begin
  If Label1.Left = -350 Then Label1.Left := 780
  Else Label1.Left := Label1.Left - 2;
  If Panel5.Visible Then
    Begin
      If Image5.Top >= -800 Then Image5.Top := Image5.Top - 1
      Else Image5.Top := 450;
    End;
  end;
```

5.2 Fasilitas Penerimaan Siswa Baru

Dalam fasilitas penerimaan siswa baru ini ada dua sitem penerimaan siswa baru yaitu system penerimaan siswa baru secara regular melalui seleksi penerimaan siswa baru setiap tahun ajaran baru dan system penerimaan siswa

baru non regular sewaktu ada mutasi siswa dari sekolah lain. Pada prinsipnya dua sistem di atas sama, hanya saja untuk non regular langsung masuk data file siswaaktif.dbf yaitu file yang digunakan untuk menampung siswa yang aktif dari kelas 1 sampai kelas 3. Tampilan PSB untuk regular seperti di bawah ini :

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho...

.....Pendaftaran Siswa Baru.....

Data Pribadi Siswa Baru

No. Pendaftaran : _____

Nama : _____

Tempat Lahir : _____

Tanggal Lahir : 15-06-2003 ▾

Jenis Kelamin : Laki-Laki
 Perempuan

Agama : _____ ▾

Alamat : _____

Siswa
Akademik
Orang Tua
Sumbangan

Terus
Batal

Tampil **Simpan** **Keluar**

Gambar 5.2 Tampilan Input Data PSB

Dalam fasilitas PSB ini, pertama yang dilakukan adalah membuat file secara otomatis meliputi file di bawah ini :

- ❖ PSBSiswaT+AmbilTahun+.dbf (dengan format data base)
- ❖ PSBOrtuT+AmbilTahun+.dbf (dengan format data base)
- ❖ PSBIjazahT+AmbilTahun+.dbf (dengan format data base)
- ❖ PSBNEMT+AmbilTahun+.dbf (dengan format data base)
- ❖ PSBSMBT+AmbilTahun+.dbf (dengan format data base)

- ❖ PSBUmumT+AmbilTahun+.dbf (dengan format data base)
- ❖ PSBPrestasiT+AmbilTahun+.dbf (dengan format data base)
- ❖ PSBTestT+AmbilTahun+.db (dengan format paradoks)

Sedangkan detail databasenya (*field*) untuk setiap file-file di atas mengacu pada desain data base yang telah ditetapkan dalam fase desain sistem (bab IV)

Pembuatan file secara otomatis ini dilakukan sewaktu file dibuat (create) sehingga harus mencantumkan script dalam bahasa SQL dalam event OnCreate seperti di bawah ini :

```

Procedure TDaftar.FormCreate(Sender: TObject);
Var
  NamaFile : String;
begin
  NamaFile := 'c:\DataSispen\PSBSiswaT'+Tahun+'.dbf';
  If Not FileExists(NamaFile) Then
    Begin
      Q.SQL.Clear;
      Q.SQL.Add(Format('Create Table "%s"',[NamaFile]));
      Q.SQL.Add('(Tanggal Date,');
      Q.SQL.Add('NoDaftar Character(10),');
      Q.SQL.Add('Nama Character(30),');
      Q.SQL.Add('Kelamin Character(10),');
      Q.SQL.Add('Tempat Character(20),');
      Q.SQL.Add('TglLahir Date,');
      Q.SQL.Add('Agama Character(19),');
      Q.SQL.Add('Alamat Character(35),');
      Q.SQL.Add('Anak Character(2),');
      Q.SQL.Add('Saudara Character(2),');
      Q.SQL.Add('Hobby Character(50),');
      Q.SQL.Add('Pengalaman Character(50),');
      Q.SQL.Add('Prestasi Character(50),');
      Q.SQL.Add('Sekolah Character(50),');
      Q.SQL.Add('ALSekolah Character(50),');
      Q.SQL.Add('Lulus Character(10)');
      Q.ExecSQL;
    End;
  End;

```

```

Q.SQL.Clear;
Q.SQL.Add(Format('Create Index Nama on "%s" (Nama) ',[NamaFile]));
Q.ExecSQL;
End;
End;

```

File-file yang lain juga dibuat sewaktu *event OnCreate* dengan mengacu pada *field-field* yang ditetapkan dalam fase desain data base. sedangkan desain *form* untuk PSB menggunakan komponen visual di antaranya :

- 1) Edit untuk menerima input berupa *text (string)* seperti yang digunakan untuk menerima input nomor pendaftaran, nama calon siswa, tempat lahir, alamat dan sebagainya.
- 2) *ComboBox* untuk menerima input yang berupa *text* tetapi telah ditentukan seperti untuk input agama, asal sekolah dan pekerjaan orang tua.
- 3) *DateTimePicker* untuk menerima input berupa tanggal seperti yang digunakan untuk menerima input untuk tanggal lahir.
- 4) *RadioButton* untuk input yang berupa *optional* seperti input jenis kelamin, penggunaan *RadioButton* hanya untuk opsi input yang relative sedikit karena akan memakan tempat dan agak sulit untuk menindak lanjuti input dari *RadioButton* dalam pengolahan data.
- 5) *Button tombol* untuk melakukan komunikasi dengan user seperti tombol simpan, batal, tampil dan sebagainya. Untuk tombol-tombol ini biasanya akan dihubungkan pada *event OnClick* seperti pada tombol



tampil (button20) *event OnClick*nya merupakan *procedure* yang berisi perintah SQL untuk menampilkan data PSBSiswaT+AmbilTahun+.dbf

```
procedure TDaftar.Button20Click(Sender: TObject);
begin
  Q.SQL.Clear;
  Q.SQL.Add(Format('Select Tanggal,NoDaftar>Nama,Kelamin,Sekolah From
"%s",[c:\DataSispen\PSBSiswaT'+AmbilTahun+'.dbf']));
  Q.Open;
  If not Q.IsEmpty Then
    Begin
      Panel7.Visible := True;
      Panel14.Visible := False;
    End
  Else
    MessageDlg('Tidak Ada Data Calon Siswa Baru...!',mtInformation,[mbOK],0);
end;
```

- 6) *GroupBox* dan *Panel* merupakan komponen visual yang digunakan untuk permaianan layar karena tidak semua data tertampung dalam satu layer maka harus dipecah dalam paket-paket data kemudian dengan menggunakan properti *visible* dari komponen *GroupBox* dan *Panel*, data dapat ditampilkan dan dihilangkan. Properti *visible* diisi dengan *true* untuk menampakkan komponen dan mengisi propoerti *visible* dengan *false* untuk menghilangkan komponen (tak tampak).

Selain *event OnClick* dan *OnCreate* fasilitas PSB ini juga menggunakan *event-event* lain di antaranya :

- A. *Event OnDbClick* yaitu *event* yang akan dieksekusi jika user melakukan *double click* pada komponen. Dalam fasilitas PSB event *OnDbClick* digunakan untuk melakukan pendataan asal sekolah. Seperti *script* di bawah ini ;

```

procedure TDaftar.ComboBox11DbfClick(Sender: TObject);
begin
  Panel13.Visible := True;
  Edit50.SetFocus;
end;

```

- B. *Event OnExit* yaitu *event* yang akan dieksekusi jika user meninggalkan komponen. Dalam fasilitas PSB *event OnExit* digunakan sewaktu pencarian alamat asal sekolah seperti *script* di bawah ini :

```

procedure TDaftar.ComboBox11Exit(Sender: TObject);
begin
  Q.SQL.Clear;
  Q.SQL.Add(Format('Select      Distinct(Alsekolah)      From      "%s"',
['c:\DataSispen\PSBSiswaT'+AmbilTahun+'.dbf']));
  Q.SQL.Add('Where Sekolah = :Daftar');
  Q.Prepare;
  Q.Params[0].AsString := ComboBox11.Text;
  Q.Open;
  If not Q.IsEmpty Then Edit6.Text := Q['AlSekolah'];
end;

```

Untuk melakukan penyimpanan digunakan *event OnClick* pada tombol

Simpan (Button19) dengan *procedure* di bawah ini :

```

procedure TDaftar.Button19Click(Sender: TObject);
Var
  Tahun,NamaFile : String;
  Ada : Boolean;
begin
  Tahun := AmbilTahun;
  NamaFile := 'c:\DataSispen\PSBSiswaT'+Tahun+'.dbf';
  Q.SQL.Clear;
  Q.SQL.Add(Format('Select * From "%s"',[NamaFile]));
  Q.SQL.Add('Where NoDaftar = :Daftar');
  Q.Prepare;
  Q.Params[0].AsString := Edit1.Text;
  Q.Open;
  If not Q.IsEmpty Then Ada := True Else Ada := False;
  If (Edit1.Text <> "") And (Edit2.Text <> "") And (not Ada) Then
    Begin
      Q.SQL.Clear;
      Q.SQL.Add(Format('Insert Into "%s" (',[NamaFile]));
  Q.SQL.Add('Tanggal,NoDaftar>Nama,Kelamin,Tempat,TglLahir,Agama,Alamat,Anak,
Saudara,Hobby,Pengalaman,Prestasi,Sekolah,AlSekolah,Lulus)');

```

```

    Q.SQL.Add('
Values(:Tanggal,:NoDaftar,:Nama,:Kelamin,:Tempat,:kTglLahir,:Agama,:Alamat,:An
ak,:Saudara,:Hobby,:Pengalaman,:Prestasi,:Sekolah,:ALSekolah,:Lulus)');
    Q.Prepare;
    Q.Params[0].AsDate := Now;
    Q.Params[1].AsString := InputKata(Edit1.Text);
    Q.Params[2].AsString := InputKata(Edit2.Text);
    If not (RadioButton1.Checked Or RadioButton2.Checked) Then
Q.Params[3].AsString := '-';
    If RadioButton1.Checked Then Q.Params[3].AsString :=
RadioButton1.Caption;
    If RadioButton2.Checked Then Q.Params[3].AsString :=
RadioButton2.Caption;
    Q.Params[4].AsString := InputKata(Edit3.Text);
    Q.Params[5].AsDate := DateTimePicker1.Date;
    Q.Params[6].AsString := InputKata(ComboBox1.Text);
    Q.Params[7].AsString := InputKata(Edit4.Text);
    Q.Params[8].AsString := InputKata(Edit9.Text);
    Q.Params[9].AsString := InputKata(Edit8.Text);
    Q.Params[10].AsString := InputKata(Edit10.Text);
    Q.Params[11].AsString := InputKata(Edit11.Text);
    Q.Params[12].AsString := InputKata(Edit12.Text);
    Q.Params[13].AsString := InputKata(Combobox11.Text);
    Q.Params[14].AsString := InputKata(Edit6.Text);
    Q.Params[15].AsString := InputKata(Edit7.Text);
    Q.ExecSQL;

    { dengan cara yang sama untuk file-file yng lain}

End
Else
Begin
    If Ada Then MessageDlg('Nomor Pendaftaran Sudah Ada...!', mtInformation,
[mbOK],0)
    Else MessageDlg('Nomor Dan Nama Siswa Masih Kosong... ', mtInformation,
[mbOK],0)
    End;
end;

```

Sedangkan untuk melakukan edit data PSB dilakukan proses *delete* terhadap record yang dipilih kemudian dilakukan *resaving* seperti script di bawah ini :

```

procedure TEditDaftar.Button1Click(Sender: TObject);
Var
  Tanggal : TDate;
  NamaFile : String;
  Nomor : String;
begin
  If not T.IsEmpty Then
    Begin
      Tanggal := T['Tanggal'];
      Nomor := T['NoDaftar'];
      NamaFile := 'c:\DataSispen\PSBSiswaT'+AmbilTahun+'.dbf';
      A.SQL.Clear;
      A.SQL.Add(Format('Delete From "%s"', [NamaFile]));
      A.SQL.Add('Where NoDaftar = :Kode');
      A.Prepare;
      A.Params[0].AsString := Nomor;
      A.ExecSQL;
      {Hapus file-file lain}
      NamaFile := 'c:\DataSispen\PSBSiswaT'+AmbilTahun+'.dbf';
      If (Edit3.Text <> '') And (Edit4.Text <> '') Then
        Begin
          A.SQL.Clear;
          A.SQL.Add(Format('Insert Into "%s" (',[NamaFile]));
          A.SQL.Add('Tanggal,NoDaftar>Nama,Kelamin,Tempat,TglLahir,Agama,Alamat,
          Anak,Saudara,Hobby,Pengalaman,Prestasi,Sekolah,ALSekolah,Lulus)');
          A.SQL.Add('
          Values(:Tanggal,:NoDaftar,:Nama,:Kelamin,:Tempat,:TglLahir,:Agama,:Alama
          t,:Anak,:Saudara,:Hobby,:Pengalaman,:Prestasi,:Sekolah,:ALSekolah,:Lulus)');
          A.Prepare;
          A.Params[0].AsDate := Tanggal;
          A.Params[1].AsString := InputKata(Edit1.Text);
          A.Params[2].AsString := InputKata(Edit2.Text);
          If RadioButton1.Checked Then A.Params[3].AsString :=
          InputKata(RadioButton1.Caption);
          If RadioButton2.Checked Then A.Params[3].AsString :=
          InputKata(RadioButton2.Caption);
          A.Params[4].AsString := InputKata(Edit3.Text);
          A.Params[5].AsDate := DateTimePicker1.Date;
          A.Params[6].AsString := InputKata(ComboBox1.Text);
          A.Params[7].AsString := InputKata(Edit4.Text);
          A.Params[8].AsString := InputKata(Edit9.Text);
          A.Params[9].AsString := InputKata(Edit8.Text);
          A.Params[10].AsString := InputKata(Edit10.Text);
          A.Params[11].AsString := InputKata(Edit11.Text);
          A.Params[12].AsString := InputKata(Edit12.Text);
          A.Params[13].AsString := InputKata(ComboBox11.Text);

```

```

A.Params[14].AsString := InputKata(Edit6.Text);
A.Params[15].AsString := InputKata(Edit7.Text);
A.ExecSQL;
{Simpan file-file lain}
End;;
End;

```

Sedangkan untuk pembuatan jurnal digunakan *scripts* SQL sebagai berikut :

```

Q.SQL.Clear;
Q.SQL.Add(Format('Delete From "%s"', [C:\DataSispen\Statistika.dbf]));
Q.ExecSQL;
NamaFile := 'c:\DataSispen\PSBSiswaT'+AmbilTahun+'.dbf';
Kumulatif := 0;
Q.SQL.Clear;
Q.SQL.Add(Format('Select Count(NoDaftar) As Jumlah From "%s"', [NamaFile]));
Q.Open;
If not Q.IsEmpty Then Total := Q['Jumlah'] Else Total := 0;
If Total > 0 Then
  Begin
    Total := Q['Jumlah'];
    Edit1.Text := RealToStr(Total,0);
    Tanggal := DateTimePicker1.Date;
    Repeat
      Q.SQL.Clear;
      Q.SQL.Add(Format('Select Count(NODaftar) As Jumlah From
"%s"', [NamaFile]));
      Q.SQL.Add('Where Tanggal = :Besarl');
      Q.Prepare;
      Q.Params[0].AsDate := Tanggal;
      Q.Open;
      If not Q.IsEmpty Then
        Begin
          Jumlah := Q['Jumlah'];
          Kumulatif := Kumulatif + Jumlah;
          Q.SQL.Clear;
          Q.SQL.Add(Format('Insert Into "%s" (',[C:\DataSispen\Statistika.dbf]));
          Q.SQL.Add('Interval,Jumlah,Prosentase,Kumulatif)');
          Q.SQL.Add(' Values(:Interval,:Jumlah,:Prosentase,:Kumulatif)');
          Q.Prepare;
          Q.Params[0].AsString := FormatDateTime('dd mmmm yyyy',Tanggal);
          Q.Params[1].AsFloat := Jumlah;
          Q.Params[2].AsFloat := (Jumlah / Total) * 100;
          Q.Params[3].AsFloat := Kumulatif;

```

```

    Q.ExecSQL;
  End;
  Tanggal := Tanggal + 1;
  Until Tanggal > DateTimePicker2.Date;
End;

```

Sedangkan untuk proses *sortir* sebenarnya hanya dengan menggunakan perintah tampil dalam bahasa SQL dengan menggunakan *select * from NamaFile* dengan menambahkan baris instruksi *Order By Test, UAN, Ijazah*

Untuk proses pengumuman diperlukan *script* SQL Update untuk mengubah nilai *field* pengumuman dari “-” menjadi “Diterima”, “Tidak Diterima” atau “Cadangan” seperti di bawah ini :

```

A.SQL.clear;
A.SQL.Add(Format('Update   "%s"   ', [c:\DataSispen\PSBUmumT'+AmbilTahun+'.dbf']));
A.SQL.Add('Set Keputusan = "Tidak Diterima"');
A.ExecSQL;
Q.First;
If StrToInt(Edit14.Text) > 0 Then
  Begin
    A.SQL.clear;
    A.SQL.Add(Format('Update
"%s", [c:\DataSispen\PSBUmumT'+AmbilTahun+'.dbf]));
    A.SQL.Add('Set Keputusan = "Diterima"');
    A.SQL.Add('Where NoDaftar = :Daftar');
    Repeat
      Kode := Q['NoDaftar'];
      A.Prepare;
      A.Params[0].AsString := Kode;
      A.ExecSQL;
      Q.Next;
      Inc(I);
    Until I >= StrToInt(Edit14.Text);
  End;
  I := 0;
  If StrToInt(Edit15.Text) > 0 Then
    Begin
      A.SQL.clear;

```

```

A.SQL.Add(Format('Update
"%s"', ['c:\DataSispen\PSBUmumT'+AmbilTahun+'.dbf']));
A.SQL.Add('Set Keputusan = "Cadangan");
A.SQL.Add('Where NoDaftar = :Daftar');
Repeat
  Kode := Q['NoDaftar'];
  A.Prepare;
  A.Params[0].AsString := Kode;
  A.ExecSQL;
  Q.Next;
  Inc(I);
Until I >= StrToInt(Edit15.Text);
End;

```

5.3 Fasilitas Distribusi Kelas

Dalam implementasi distribusi kelas, selain mempertimbangkan kemampuan akademik, jenis kelamin harus dipertimbangkan juga, supaya hasil proses distribusi jumlah kelamin berimbang. Proses distribusi kelas ini diawali dari menentukan kelas dan jurusan, jumlah kelas (ruangan) dalam kelas dan jurusan tadi serta menentukan metode yang akan digunakan. Setelah itu baru memasukkan jumlah maksimal siswa dalam masing-masing kelas seperti tampak di bawah ini:

The screenshot shows a window titled ".... Fasilitas Pembagian Kelas....". Inside, there is a section "Data Distribusi Kelas" with the following fields:

- Kelas : 1
- Jurusan : Belum Penjurusan
- Jumlah Kelas : 6
- Metode : Homogen

Below this is a "Kapasitas Kelas" section with a text box containing "40". At the bottom of the window, there are four buttons: "Tampil", "Next", "Eksekusi", and "Keluar".

Gambar 5.3 Tampilan Distribusi Kelas

Dalam pemrograman di atas penulis menggunakan *array* berdimensi dua untuk menampung informasi tentang jumlah maksimal siswa dan jumlah siswa. Jumlah maksimal siswa dalam kelas ditampung dengan *array* dengan nilai kolom = 1 seperti program di bawah ini :

```

If I <= StrToInt(Edit2.Text) Then
  Begin
    Jumlah[I,1] := Round(StrToReal(Edit3.Text));
    Inc(I);
    If I <= StrToInt(Edit2.Text) Then
      Begin
        If Jur = '-' Then Label4.Caption := 'Kapasitas Kelas '+Edit1.Text+ ' - '
+IntToStr(I)
        Else Label4.Caption := 'Kapasitas Kelas '+Edit1.Text+ ' - '+ Jur +' - '
+IntToStr(I)
        End;
        Edit3.Text := "";
        Edit3.SetFocus;
      End
    Else MessageDlg('Jumlah Kelas Telah Melebihi, Click Tombol
Eksekusi',mtInformation,[mbOK],0);
  End

```

Sedangkan kolom = 2 digunakan untuk menampung data jumlah siswa berkelamin laki-laki dalam satu kelas. Penghitungan ini berdasarkan prosentase dari jumlah kelamin laki-laki dan wanita seperti di bawah ini :

```

P.SQL.Clear;
P.SQL.Add(Format('Select      Count(NoInduk)      As      Jumlah      From
"%s"', ['c:\dataSispen\SiswaAktif.dbf']));
P.SQL.Add('Where Kelamin = "Laki-Laki"');
P.SQL.Add('And Kelas = "1"');
P.Open;
If P['Jumlah'] > 0 Then Pria := P['Jumlah'] Else Pria := 1;
P.SQL.Clear;
P.SQL.Add(Format('Select      Count(NoInduk)      As      Jumlah      From
"%s"', ['c:\dataSispen\SiswaAktif.dbf']));
P.SQL.Add('Where Kelamin = "Perempuan"');
P.SQL.Add('And Kelas = "1"');
P.Open;

```

```

If P['Jumlah'] > 0 Then Wanita := P['Jumlah'] Else Wanita := 1;
Prosentase := Pria / (Pria + Wanita);
P.SQL.Clear;
P.SQL.Add(Format('Select B.NoInduk,B.Kelamin,A.Nilai as Test from "%s" A, "%s"
B',
['c:\dataSispen\Datatest.dbf',
'c:\dataSispen\SiswaAktif.dbf']));
P.SQL.Add('Where B.NoInduk = A.NoInduk');
P.SQL.Add('And B.Kelamin = "Laki-Laki"');
P.SQL.Add('And B.Kelas = "1"');
P.SQL.Add('Order By A.Nilai Desc');
P.Open;
W.SQL.Clear;
W.SQL.Add(Format('Select B.NoInduk,B.Kelamin,A.Nilai as Test from "%s" A, "%s"
B',
['c:\dataSispen\Datatest.dbf',
'c:\dataSispen\SiswaAktif.dbf']));
W.SQL.Add('Where A.NoInduk = B.NoInduk');
W.SQL.Add('And B.Kelamin = "Perempuan"');
W.SQL.Add('And B.Kelas = "1"');
W.SQL.Add('Order By A.Nilai Desc');
W.Open;
For J := 1 to I Do
    Jumlah[J,2] := Round(Prosentase * Jumlah[J,1]);

```

Setelah jumlah maksimal siswa dalam satu kelas dan jumlah kelamin laki-laki dalam satu kelas diketahui maka langkah selanjutnya adalah membagi berdasarkan metode yang ditetapkan apakah dengan metode homogen (yang mempunyai kemampuan yang sama dikelompokkan dalam satu kelas) atau metode heterogen (pembagian kelas tidak memperhitungkan kemampuan siswa). Seperti *scripts* di bawah ini :

```

If Combobox2.Text = 'Homogen' Then
    Begin
        P.First;
        W.First;
        For J := 1 to I-1 Do
            Begin
                K := I;
                Repeat

```

```

T.SQL.Clear;
T.SQL.Add(Format("Update "%s"",[c:\dataSispen\SiswaAktif.dbf]));
T.SQL.Add('Set Indek = :Indek');
T.SQL.Add('Where NoInduk = :Induk');
T.Prepare;
T.Params[0].AsString := IntToStr(J);
T.Params[1].AsString := P['NoInduk'];
T.ExecSQL;
P.Next;
K := K + 1;
Until K > Jumlah[J,2];
K := 1;
Repeat
T.SQL.Clear;
T.SQL.Add(Format("Update "%s"",[c:\dataSispen\SiswaAktif.dbf]));
T.SQL.Add('Set Indek = :Indek');
T.SQL.Add('Where NoInduk = :Induk');
T.Prepare;
T.Params[0].AsString := IntToStr(J);
T.Params[1].AsString := W['NoInduk'];
T.ExecSQL;
W.Next;
K := K + 1;
Until K > (Jumlah[J,1] - Jumlah[J,2]);
End;
Repeat
T.SQL.Clear;
T.SQL.Add(Format("Update "%s"",[c:\dataSispen\SiswaAktif.dbf]));
T.SQL.Add('Set Indek = :Indek');
T.SQL.Add('Where NoInduk = :Induk');
T.Prepare;
T.Params[0].AsString := IntToStr(I);
T.Params[1].AsString := P['NoInduk'];
T.ExecSQL;
P.Next;
Until P.EOF;
Repeat
T.SQL.Clear;
T.SQL.Add(Format("Update "%s"",[c:\dataSispen\SiswaAktif.dbf]));
T.SQL.Add('Set Indek = :Indek');
T.SQL.Add('Where NoInduk = :Induk');
T.Prepare;
T.Params[0].AsString := IntToStr(I);
T.Params[1].AsString := W['NoInduk'];
T.ExecSQL;
W.Next;

```

```

    Until W.EOF;
End;

If ComboBox2.Text = 'Heterogen' Then
Begin
    P.First;
    W.First;
    J := 1;
    Repeat

        If Jumlah[J,2] > 0 Then
        Begin
            T.SQL.Clear;
            T.SQL.Add(Format('Update "%s"', ['c:\dataSispen\SiswaAktif.dbf']));
            T.SQL.Add('Set Indek = :Indek');
            T.SQL.Add('Where NoInduk = :Induk');
            T.Prepare;
            T.Params[0].AsString := IntToStr(J);
            T.Params[1].AsString := P['NoInduk'];
            T.ExecSQL;
        End;

        If J > (I) Then J := 1 Else J := J + 1;
        P.Next;
    Until P.EOF;
    J := 1;
    Repeat

        If (Jumlah[J,1] - Jumlah[J,2]) > 0 Then
        Begin
            T.SQL.Clear;
            T.SQL.Add(Format('Update "%s"', ['c:\dataSispen\SiswaAktif.dbf']));
            T.SQL.Add('Set Indek = :Indek');
            T.SQL.Add('Where NoInduk = :Induk');
            T.Prepare;
            T.Params[0].AsString := IntToStr(J);
            T.Params[1].AsString := W['NoInduk'];
            T.ExecSQL;
        End;

        If J > (I - 1) Then J := 1 Else J := J + 1;
        W.Next;
    Until P.EOF;
End;
End;

```

5.4 Absen

Fasilitas Absen ini sebenarnya merupakan pencatatan data absen ke dalam dua file yaitu :

- a. File Absen+AmbilTahun+AmbilSemester+.dbf

file ini berfungsi untuk mencatat *track record* absensi pada tanggal kapan dan jenis absensi seperti *script* di bawah ini :

```
A.SQL.Clear;
A.SQL.Add(Format('Insert                                     Into
"%s"(', [c:\DataSispen\AbsenT'+AmbilTahun+AmbilSemester+'.dbf']));
A.SQL.Add('Tanggal,NoInduk,Absent) Values (:Tanggal,:NoInduk,:Absent)');
A.Prepare;
A.Params[0].AsDate := DateTimePicker1.Date;
A.Params[1].AsString := Label6.Caption;
If RadioButton1.Checked Then A.Params[2].AsString := 'Sakit';
If RadioButton2.Checked Then A.Params[2].AsString := 'Ijin';
If RadioButton3.Checked Then A.Params[2].AsString := 'Alpa';
A.ExecSQL;
```

- b. File TotalAbsen+AmbilTahun+AmbilSemester+.dbf

file ini berfungsi untuk mencatat rekapitulasi absent siswa. File ini dipergunakan pada proses statistika absensi siswa seperti siswa yang telah absen melebihi jumlah hari tertentu. Pencatatan dalam file seperti dalam *scripts* ini :

```
A.SQL.Clear;
A.SQL.Add(Format('Select * from "%s"', [c:\DataSispen\TotalAbsenT'+
AmbilTahun+AmbilSemester+'.dbf']));
A.SQL.Add('Where NoInduk = :NoInduk');
A.Prepare;
A.Params[0].AsString := Label6.Caption;
A.Open;

If not A.IsEmpty Then
```

```

Begin
  A.SQL.Clear;
  A.SQL.Add(Format('Update
"%s"', ['c:\DataSispen\TotalAbsenT'+AmbilTahun+AmbilSemester+'.dbf']));
  If RadioButton1.Checked Then A.SQL.Add('Set Sakit = Sakit + 1');
  If RadioButton2.Checked Then A.SQL.Add('Set Ijin = Ijin + 1');
  If RadioButton3.Checked Then A.SQL.Add('Set Alpa = Alpa + 1');
  A.SQL.Add('Where NoInduk = :NoInduk');
  A.Prepare;
  A.Params[0].AsString := Label6.Caption;
  A.ExecSQL;
End
Else

  Begin
    A.SQL.Clear;
    A.SQL.Add(Format('Insert
"%s"', ['c:\DataSispen\TotalAbsenT'+AmbilTahun+AmbilSemester+'.dbf']));
    A.SQL.Add('NoInduk,Sakit,Ijin,Alpa) Values (:NoInduk,:Sakit,:Ijin,:Alpa)');
    A.Prepare;
    A.Params[0].AsString := Label6.Caption;

    Into

    If RadioButton1.Checked Then
      Begin
        A.Params[1].AsFloat := 1;
        A.Params[2].AsFloat := 0;
        A.Params[3].AsFloat := 0;
      End;

    If RadioButton2.Checked Then
      Begin
        A.Params[1].AsFloat := 0;
        A.Params[2].AsFloat := 1;
        A.Params[3].AsFloat := 0;
      End;

    If RadioButton3.Checked Then
      Begin
        A.Params[1].AsFloat := 0;
        A.Params[2].AsFloat := 0;
        A.Params[3].AsFloat := 1;
      End;
    A.ExecSQL;
  End;

```

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho

...Fasilitas Absensi Siswa...

Data SPP Siswa

Nomor Absen : 14
 Nomor Induk : 00114
 Nama : YUSRIN AROFAH Z
 Kelas : 1 - 3

Data Pembayaran SPP

Tanggal : 07/07/2005
 Jenis Absensi : Sakit Ijin Alpa

Gambar 5.4 Tampilan Absensi Siswa

5.5 Fasilitas Ulangan Harian

Merupakan fasilitas entry data ulangan harian kemudian merekapnya dengan bentuk statistika tertentu. Sebelum entry data diperlukan input data berupa kelas, jurusan, indeks kelas, matapelajaran dan ulangan yang ke berapa?. Seperti tampilan dibawah ini :

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho

...Input Nilai Ulangan Harian...

Data Kelas dan Ulangan

Kelas : 1
 Jurusan : Belun Penjurusau
 Indeks : 1
 Matapelajaran : Matematika
 Ulangan Ke : 2

Adobe Photoshop - [Untitled-1 @ 50% (Layer 1, RGB/8)]

.Gambar 5.5 Tampilan Awal Input Nilai Ulangan Harian

Dalam tampilan di atas penulis menggunakan *event OnExit* yaitu sebuah *event* yang akan dikerjakan sewaktu suatu komponen (combobox1) ditinggalkan (keluar dari komponen combobox1). Procedure itu berupa pembacaan semua nama matapelajaran yang ada dalam file pelajaran.dbf seperti *scripts* berikut ini :

```

procedure TUIHar.ComboBox1Exit(Sender: TObject);
Var
  jur : String;
begin
  If (Edit5.Text <> "") And (ComboBox1.Text <> "")Then
    Begin
      Case ComboBox1.ItemIndex Of
        0 : Jur := '-';
        1 : Jur := 'IPA';
        2 : Jur := 'IPS';
        3 : Jur := 'Bahasa';
      End;

      A.SQL.Clear;
      A.SQL.Add(Format('Select * from "%s"', [c:\DataSispen\Pelajaran.dbf]));
      A.SQL.Add('Where Kelas = :Kelas');
      A.SQL.Add('And Jurusan = :Jurusan');
      A.Prepare;
      A.Params[0].AsString := Edit5.Text;
      A.Params[1].AsString := Jur;
      A.Open;

      If not A.IsEmpty Then
        Begin
          A.First;
          ComboBox2.Clear;
          Repeat
            ComboBox2.Items.Add(A['Pelajaran']);
          A.Next;
          Until A.Eof;
        End;
      End;
    End;
end;

```

Setelah semua data awal dientry computer akan melakukan proses pencarian siswa yang berada pada kelas, jurusan dan indek yang telah ditentukan oleh *user* dengan event pemicu *event OnClick* pada Button10(Button Input) seperti scripts di bawah ini :

```

procedure TUIHar.Button10Click(Sender: TObject);
Var
  Jur : String;
begin
  Case ComboBox1.ItemIndex Of
    0 : Jur := '-';
    1 : Jur := 'IPA';
    2 : Jur := 'IPS';
    3 : Jur := 'Bahasa';
  End;

  Q.SQL.Clear;
  Q.SQL.Add(Format('Select * from "%s"', [c:\dataSispen\SiswaAktif.dbf]));
  Q.SQL.Add('Where Kelas = :Kelas');
  Q.SQL.Add('And Jurusan = :Jurusan');
  Q.SQL.Add('And Indek = :Indek');
  Q.Prepare;
  Q.Params[0].AsString := Edit5.Text;
  Q.Params[1].AsString := Jur;
  Q.Params[2].AsString := Edit6.Text;
  Q.Open;
  If not Q.IsEmpty Then
    Begin
      Q.First;
      Label17.Caption := IntToStr(Q.RecNo);
      Label12.Caption := Q['NoInduk'];
      Label11.Caption := Q['Nama'];
      Edit1.Text := '';
      Panel6.Visible := False;
      Panel2.Visible := True;

      End
    Else MessageDlg('Belum Ada Data Siswa Dengan Kelas Di
atas...!', mtInformation, [mbOK], 0);
  end;

```

Setelah procedure di atas dieksekusi maka akan tampil panel seperti di bawah ini :

Gambar 5.6 Tampilan Proses Input Nilai Ulangan Harian

Setelah nilai ulangan harian diisi user tinggal menekan button Simpan maka dengan event `OnClick` maka nilai itu akan disimpan ke dalam file `NUT'+AmbilTahun+AmbilSemester+'.dbf'` seperti scripts di bawah ini ;

```

procedure TUIHar.Button1Click(Sender: TObject);
Var
  Kode : String;
begin
  If Edit1.Text <> " Then
    Begin
      A.SQL.Clear;
      A.SQL.Add(Format("Select NoInduk from
'%s'",[c:\DataSispen\NUT'+AmbilTahun+AmbilSemester+'.dbf']));
      A.SQL.Add("Where NoInduk = :NoInduk");
      A.SQL.Add('And Mapel = :Matpel');
      A.Prepare;
      A.Params[0].AsString := Label12.Caption;
      A.Params[1].AsString := Label31.Caption;
      A.Open;

      If not A.IsEmpty Then
    
```

```

Begin
  A.SQL.Clear;
  A.SQL.Add(Format('Update
"%s"', ['c:\DataSispen\NUT'+AmbilTahun+AmbilSemester+'.dbf']));
  Case StrToInt(Label9.Caption) Of
    1 : A.SQL.Add('Set I = :I,');
    2 : A.SQL.Add('Set II = :I,');
    3 : A.SQL.Add('Set III = :I,');
    4 : A.SQL.Add('Set IV = :I,');
    5 : A.SQL.Add('Set V = :I,');
  End;
  If not Edit Then
    Begin
      A.SQL.Add('Jumlah = Jumlah + :Tot,');
      A.SQL.Add('Ke = Ke + 1');
    End
  Else
    Begin
      A.SQL.Add('Jumlah = Jumlah + :Tot');
    End;
    A.SQL.Add('Where NoInduk = :NoID');
    A.SQL.Add('And Mapel = :Mapel');
    A.Prepare;
    A.Params[0].AsFloat := StrToReal(Edit1.Text);
    If Edit Then A.Params[1].AsFloat := StrToReal(Edit1.Text) - Nilai Else
  A.Params[1].AsFloat := StrToReal(Edit1.Text);
    A.Params[2].AsString := Label12.Caption;
    A.Params[3].AsString := Label31.Caption;
    A.ExecSQL;

  End
Else
  Begin
    A.SQL.Clear;
    A.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\Pelajaran.dbf']));
    A.SQL.Add('Where Kelas = :Kelas');
    A.SQL.Add('And Pelajaran = :Jurusan');
    A.Prepare;
    A.Params[0].AsString := Edit5.Text;
    A.Params[1].AsString := ComboBox2.Text;
    A.Open;
    Kode := A['Kode'];

    A.SQL.Clear;
    A.SQL.Add(Format('Insert Into "%s"
(['c:\DataSispen\NUT'+AmbilTahun+AmbilSemester+'.dbf']));

```

```

A.SQL.Add('NoInduk,Kode,MaPel,I,II,III,IV,V,Ke,Jumlah,Rata');
A.SQL.Add('Values
(:NoInduk,:Kode,:MaPel,:I,:II,:III,:IV,:V,:Ke,:Jumlah,:Rata)');
A.Prepare;
A.Params[0].AsString := Label12.Caption;
A.Params[1].AsString := Kode;
A.Params[2].AsString := Label31.Caption;
A.Params[3].AsFloat := StrToReal(Edit1.Text);
A.Params[4].AsFloat := 0;
A.Params[5].AsFloat := 0;
A.Params[6].AsFloat := 0;
A.Params[7].AsFloat := 0;
A.Params[8].AsFloat := 1;
A.Params[9].AsFloat := StrToReal(Edit1.Text);
A.Params[10].AsFloat := StrToReal(Edit1.Text);
A.ExecSQL;
End;

Q.Next;
If not Q.EOF Then
Begin
Label17.Caption := IntToStr(Q.RecNo);
Label12.Caption := Q['NoInduk'];
Label11.Caption := Q['Nama'];
Edit1.Text := '';
Edit1.SetFocus;
End
Else
MessageDlg('Input Data Sudah Selesai',mtInformation,[mbOK],0);
End;
end;

```

5.6 SPP

Fasilitas SPP sebenarnya hamper sama dengan fasilitas absent hanya saja dalam fasilitas ini penulis ingin menonjolkan koneksitas dan kompatibilitas antara database pascal klasik (file bertipe) dengan data base bawaan Delphi yaitu file-file database IV. Tidak seperti file-file database, file bertipe (database pascal klasik) membutuhkan deklarasi nama field dan tipe

maupun panjang masing-masing field.pendeklarasian ini dicantumkan dalam deklarasi *type* sebelum deklarasi *form* utama sehingga hasil dari deklarasi ini dapat dibaca oleh deklarasi *form* utama terutama untuk mendeklarasikannya dalam bagian variable *private* seperti *scripts* di bawah ini :

```

type
  SPP      = Record
    Tanggal : TDate;
    NoInduk : String[18];
    Nama    : String[18];
    Kelas   : String[18];
    Besar   : String[8];
  End;

  RekapSPP = Record
    NoInduk : String[18];
    Nama    : String[18];
    Kelas   : String[18];

    Jul    : String[1];
    Ags    : String[1];
    Sept   : String[1];
    Okt    : String[1];
    Nov    : String[1];
    Des    : String[1];
    Jan    : String[1];
    Feb    : String[1];
    Mar    : String[1];
    Apr    : String[1];
    Mei    : String[1];
    Jun    : String[1];
    Bulan  : String[1];
    Akhir  : Byte;
  End;

TISPP = class(TForm)
  {Deklarasi komponen visual yg digunakan }

  DateTimePicker2: TDateTimePicker;
  procedure Button1Click(Sender: TObject);
  procedure Button3Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure Button8Click(Sender: TObject);

```

```

procedure Edit5Change(Sender: TObject);
procedure Button20Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure DBGrid1CellClick(Column: TColumn);

private
  { Private declarations }

  FSPP : File Of SPP;
  TSPP : SPP;
  TRekap : File Of RekapSPP;
  RSPP : RekapSPP;
public
  { Public declarations }
end;
var
  ISPP: TISPP;

```



Setelah dideklarasikan maka file yang akan dibuat harus di hubungkan dengan nama file *internal* dengan statement *AssignFile(TSPP, 'c:\dataSispen\SPPT'+AmbilTahun+'.dat');* dan *AssignFile(TRekap, 'c:\dataSispen\DataSPPT'+AmbilTahun+'.dat');*

Kemudian dilakukan baru dilakukan check keberadaan kedua file tersebut jika sudah ada maka kedua file tersebut harus dibuka dengan intruksi *Reset(TSPP)* dan *Reset(TRekap)* tetapi kalau belum ada maka instruksi yang digunakan adalah *Rewrite(TSPP)* dan *Rewrite(TRekap)* dengan intruksi *Rewrite* maka secara otomatis file tersebut akan *dcreate* oleh computer. Tampilan dari fasilitas pembayaran SPP sebagai berikut :

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho...

...Fasilitas Pembayaran SPP...

Data SPP Siswa

Nomor Induk : 0001
 Nama : Agus Waskito Setyo Budi
 Kelas : 1 - 1
 Besar SPP : 0
 Bulan Terakhir : Juli

Data Pembayaran SPP

Tanggal : 07/07/2005
 Bulan : Agustus
 Pembayaran SPP : 35.000

Gambar 5.7 Tampilan Proses Input SPP

Setelah *user* mengisi bulan dan nominal SPP, dengan mengclick *button* Simpan data SPP akan disimpan dalam dua file diatas. Adapun *procedure* penyimpanannya sebagai berikut :

```

procedure TISPP.Button3Click(Sender: TObject);
Var
  Akhir : Real;
  Bulan : String;
  I : Integer;
  Ketemu : Boolean;
begin
  If StrToReal(Edit1.Text) > 0 Then
    Begin

      Seek(FSPP,FileSize(FSPP));
      TISPP.Tanggal := DateTimePicker1.Date;
      TISPP.NoInduk := Label3.Caption;
      TISPP>Nama := Label6.Caption;
    End;
  End;

```

```

TSPP.Kelas := label9.Caption;
TSPP.Bulan := ComboBox1.Text;
TSPP.Besar := StrToReal(HapusTitik(Edit1.Text));
Write(FSPP,TSPP);
I:=0;
Ketemu := False;
Repeat
  Seek(TRekap,I);
  Read(TRekap,RSPP);
  If RSPP.NoInduk = Label3.Caption Then Ketemu := True
  Else Inc(I);
Until (I = FileSize(TRekap)) Or Ketemu;
If Ketemu Then
  Begin
    Akhir := RSPP.Akhir;
    If Akhir = 0 Then Bulan := 'Juli';
    If Akhir = 1 Then Bulan := 'Agustus';
    If Akhir = 2 Then Bulan := 'September';
    If Akhir = 3 Then Bulan := 'Oktober';
    If Akhir = 4 Then Bulan := 'November';
    If Akhir = 5 Then Bulan := 'Desember';
    If Akhir = 6 Then Bulan := 'Januari';
    If Akhir = 7 Then Bulan := 'Februari';
    If Akhir = 8 Then Bulan := 'Maret';
    If Akhir = 9 Then Bulan := 'April';
    If Akhir = 10 Then Bulan := 'Mei';
    If Akhir = 11 Then Bulan := 'Juni';
    If Akhir = 0 Then RSPP.Jul := 'V';
    If Akhir = 1 Then RSPP.Ags := 'V';
    If Akhir = 2 Then RSPP.Sept := 'V';
    If Akhir = 3 Then RSPP.Okt := 'V';
    If Akhir = 4 Then RSPP.Nov := 'V';
    If Akhir = 5 Then RSPP.Des := 'V';
    If Akhir = 6 Then RSPP.Jan := 'V';
    If Akhir = 7 Then RSPP.Feb := 'V';
    If Akhir = 8 Then RSPP.Mar := 'V';
    If Akhir = 9 Then RSPP.Apr := 'V';
    If Akhir = 10 Then RSPP.Mei := 'V';
    If Akhir = 11 Then RSPP.Jun := 'V';
    RSPP.Akhir := Akhir+ 1;
    RSPP.Bulan := Bulan;

    Write(TRekap,RSPP);
  End
Else
  Begin

```

```

RSPP.Akhir := 1;
RSPP.Bulan := 'Juli';
RSPP.Jul := 'V';
RSPP.Ags := 'X';
RSPP.Sept := 'X';
RSPP.Okt := 'X';
RSPP.Nov := 'X';
RSPP.Des := 'X';
RSPP.Jan := 'X';
RSPP.Feb := 'X';
RSPP.Mar := 'X';
RSPP.Apr := 'X';
RSPP.Mei := 'X';
RSPP.Jun := 'X';
RSPP.NoInduk := Label3.Caption;
RSPP>Nama := Label6.Caption;
RSPP.Kelas := Label9.Caption;
Seek(TRekap,FileSize(TRekap));
Write(TRekap,RSPP);
End;

Label3.Caption := '-';
Label6.Caption := '-';
Label9.Caption := '-';
Label14.Caption := '-';
ComboBox1.Text := "";
Label24.Caption := '-';
Edit1.Text := "";
Panel5.Visible := True;
Panel2.Visible := False;
End;
end;

```

5.7 Fasilitas BP

Fasilitas ini berfungsi untuk melakukan pencatatan data konseling dan bimbingan siswa yang dilakukan oleh guru BP. Data konseling maupun bimbingan siswa disimpan dalam file 'c:\dataSispen\BP.dbf' data BP ini dapat diretrieve kembali menjadi track record kasus-kasus siswa yang masuk ke BP.

Pada prinsipnya ada dua fasilitas dalam modul BP ini yaitu fasilitas pendataan kasus baru dan menampilkan track record kasus siswa. Adapun tampilan fasilitas BP sebagai berikut :

The screenshot shows a window titled "Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho". The main header is "...Bimbingan Dan Konseling Siswa...". Below the header, there is a section for "Data Pribadi Siswa" with the following details:

Data Pribadi Siswa		Kelas : 1 - 1
Nomor Induk :	0001	<input type="button" value="Selesai"/> <input type="button" value="Sebelum"/> <input type="button" value="Data BP"/> <input type="button" value="Batal"/>
Nama :	Agus Waskito Setyo Budi	
Tempat Lahir :	Pemalang	
Tanggal Lahir :	13/04/1988	
Jenis Kelamin :	Laki-Laki	
Agama :	Islam	
Anak Ke :	1	
Jumlah Saudara :	4	
Alamat :	Jl. Kartini No. 4 Petarukan	

Gambar 5.7 Tampilan Proses Input Data BP

Dengan mengclick buton Data BP maka akan tampak form entry data untuk kasus BP sebagai berikut :

The screenshot shows the same application window, but with the "Data BP" button clicked. The form now displays case details:

Data Pribadi Siswa			
Tanggal :	07-07-2005	<input type="button" value="Track Record"/> <input type="button" value="Simpan"/> <input type="button" value="Batal"/>	
Kriteria :	Kriteria Permasalahan <input type="radio"/> Konseling Siswa <input checked="" type="radio"/> Bimbingan Siswa		
Jenis Permasalahan :	Narkoba		
Detail Permasalahan :	Kedapatan membawa ganja di sekolah		
Solusi Permasalahan :	Pembinaan dengan melakukan koordinasi dengan orang tua (pemberian surat peringatan I dan skorrsing 1 minggu)		
Catatan :	Jika melakukan kesalahan yang sama langsung dikeluarkan		
			<input type="button" value="Keluar"/>

Gambar 5.8 Tampilan Input Kasus BP

Dengan memanfaatkan event OnClick pada Button Simpan dengan procedure di bawah ini :

```

Procedure TBPS.Button9Click(Sender: TObject);
Var
  Nama : String;
begin
  If ComboBox1.Text <> " Then
    Begin
      Nama := 'c:\dataSispen\BP.dbf';
      Q.SQL.Clear;
      Q.SQL.Add(Format('Insert Into "%s"(', [Nama]));
      Q.SQL.Add('Tanggal, Kriteria, Jenis, Masalah, Solusi, Catatan, NoInduk) Values
(:Tanggal, :Kriteria, :Jenis, :Masalah, :Solusi, :Catatan, :NoInduk)');
      Q.Prepare;
      Q.Params[0].AsDate := DateTimePicker1.Date;
      If RadioButton1.Checked Then Q.Params[1].AsString := 'Konseling';
      If RadioButton2.Checked Then Q.Params[1].AsString := 'Bimbingan';
      Q.Params[2].AsString := ComboBox1.Text;
      Q.Params[3].AsString := Memo1.Lines.Text;
      Q.Params[4].AsString := Memo2.Lines.Text;
      Q.Params[5].AsString := Memo3.Lines.Text;
      Q.Params[6].AsString := Label16.Caption;
      Q.ExecSQL;
    End;
    RadioButton1.Checked := False;
    RadioButton2.Checked := False;
    ComboBox1.Text := '';
    Memo1.Lines.Clear;
    Memo2.Lines.Clear;
    Memo3.Lines.Clear;
end;

```

Dan retrieve data BP dalam wujud track record kasus siswa digunakan scripts di bawah ini :

```

procedure TBPS.Button8Click(Sender: TObject);
begin
  Q.SQL.Clear;
  Q.SQL.Add(Format('Select * From "%s"', ['c:\dataSispen\BP.dbf']));
  Q.SQL.Add('Where NoInduk = :No');
  Q.Prepare;
  Q.Params[0].AsString := Label16.Caption;
  Q.Open;

```

```

If not Q.IsEmpty Then
  Begin
    Q.First;
    Label130.Caption := DateToStr(Q['Tanggal']);
    Label128.Caption := Q['Kriteria'];
    Label129.Caption := Q['Jenis'];
    Memo6.Lines.Text := Q['Masalah'];
    Memo5.Lines.Text := Q['Solusi'];
    Memo4.Lines.Text := Q['Catatan'];
    Panel3.Visible := False;
    Panel8.Visible := True;
  End
Else MessageDlg('Tidak Ada Data Konseling Maupun
Bimbingan..',mtInformation,[mbOK],0);
end;

```

5.8 Fasilitas Raport

Fasilitas raport merupakan fasilitas pembuatan nilai raport dari nilai ulangan harian, nilai tugas dan nilai ulangan sumatif. Maka langkah pertama dalam mengentry nilai raport adalah mengentry nilai tugas dan nilai sumatif (nilai ulangan harian sudah dientry menggunakan fasilitas ulangan harian). Ada form khusus untuk untuk mengentry data nilai sumatif seperti di bawah ini :

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho

...Input Nilai Ulangan Umum (Test Sumatif)...

Data Ulangan Siswa

Nomor Absent : 1
 Nomor Induk : 0001
 Nama : Agus Waskito Setyo Budi
 Kelas : 1 - 1
 Matapelajaran : Matematika

Input Nilai Ulangan Sumatif

Nilai Ulangan Sumatif : 85

Sebelum
 Setelah
 Simpan
 Close

Tampil Keluar

Gambar 5.9 Tampilan Input Nilai Ulangan Sumatif

Scripts untuk mengentry data sumatif ini sama dengan sripts yang digunakan sewaktu mengentry data ulangan harian. Hal yang sama berlaku untuk fasilitas mengentry nilai tugas. Setelah nilai ulangantugas dan nilai sumatif telah dientry maka user tinggal melakukan prosentase masing-masing variable di atas seperti form di bawah ini :

The screenshot shows a software window titled "...Penentuan Nilai Raport...". Inside, there is a form with the following fields and controls:

- Data Kelas Dan Matapelajaran:**
 - Kelas :
 - Jurusan :
 - Matapelajaran :
- Rumus Nilai Raport:**
 - Nilai = $\frac{1 \text{ Ulangan} + 1 \text{ Tugas} + 2 \text{ Sumatif}}{4}$
 - Buttons: , ,

Gambar 5.10 Tampilan Penentuan Rumus Nilai Raport

Dengan menggunakan event OnClick pada Button Nilai Raport maka prrosedure dibawah ini akan dieksekusi :

```

Procedure TRap.Button1Click(Sender: TObject);
Var
  Nilai : Real;
  jUR,Kode : String;
begin
  If Label7.Caption <> '0' Then
  Begin
    Case ComboBox1.ItemIndex Of
      0 : Jur := '-';
      1 : Jur := 'IPA';
    end
  end

```

```

2 : Jur := 'IPS';
3 : Jur := 'Bahasa';
End;
A.SQL.Clear;
A.SQL.Add(Format('Select Kode From "%s"', [c:\dataSispen\Pelajaran.dbf]));
A.SQL.Add('Where Pelajaran = :Pelajaran');
A.SQL.Add('And Kelas = :Kel');
A.SQL.Add('And Jurusan = :Kel1');
A.Prepare;
A.Params[0].AsString := ComboBox2.Text;
A.Params[1].AsString := Edit4.Text;
A.Params[2].AsString := Jur;
A.Open;
If not A.IsEmpty Then Kode := A['Kode'];
A.SQL.Clear;
A.SQL.Add(Format('Select A.NoInduk,B.Nilai As Sumatif,C.Ke,C.Jumlah,D.Nilai
As Tugas From "%s" A, "%s" B, "%s" C, "%s" D ',
[c:\dataSispen\SiswaAktif.dbf], 'c:\DataSispen\NSMITT'+AmbilTahun+AmbilSemester
+'.dbf',
'c:\DataSispen\NUT'+AmbilTahun+AmbilSemester+'.dbf', 'c:\DataSispen\NTugasT'+A
mbilTahun+AmbilSemester+'.dbf']));
A.SQL.Add('Where A.NoInduk = B.NoInduk');
A.SQL.Add('And A.NoInduk = C.NoInduk');
A.SQL.Add('And A.NoInduk = D.NoInduk');
A.SQL.Add('And A.Kelas = :Kel');
A.SQL.Add('And B.Mapel = C.Mapel');
A.SQL.Add('And B.Mapel = D.Mapel');
A.SQL.Add('And B.Mapel = :Mapel');
A.SQL.Add('And A.Jurusan = :Mapel1');
A.Prepare;
A.Params[0].AsString := Edit4.Text;
A.Params[1].AsString := ComboBox2.Text;
A.Params[2].AsString := Jur;
A.Open;
If not A.IsEmpty Then
Begin
A.First;
Repeat
Nilai := ((StrToWord(Edit1.Text) * (A['Jumlah'] / A['Ke'])) +
(StrToWord(Edit2.Text) * A['Tugas']))
+ (StrToWord(Edit3.Text) * A['Sumatif']) /
StrToWord(Label7.Caption);
Q.SQL.Clear;
Q.SQL.Add(Format('Insert Into "%s"
(', [c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
Q.SQL.Add('Kode, MatPel, Asli, Katrol, Nilai, NoInduk');

```

```

Q.SQL.Add('Values (:Kode,:MatPel,:Asli,:Katrol,:Nilai,:NoInduk)');
Q.Prepare;
Q.Params[0].AsString := Kode;
Q.Params[1].AsString := ComboBox2.Text;
Q.Params[2].AsFloat := Nilai;
Q.Params[3].AsFloat := 0;
Q.Params[4].AsFloat := Round(Nilai);
Q.Params[5].AsString := A['NoInduk'];
Q.ExecSQL;
A.Next;
Until A.EOF;
MessageDlg('Proses Pembuatan Raport Telah
Selesai...',mtInformation,[mbOK],0);
End
Else MessageDlg('Data Raport Tidak Dijumpai...',mtInformation,[mbOK],0);
End;
end;

```

5.8 Fasilitas Pemilihan Jurusan

Fasilitas pemilihan jurusan ini merupakan fasilitas yang menggunakan data minat siswa pada suatu jurusan tertentu dan data kemampuan akademik yang diperoleh dari nilai raport. Data minat diperoleh dari fasilitas input minat siswa pada jurusan tertentu seperti di bawah ini :

The screenshot shows a software window titled "Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho..." with a sub-header "...Input Minat Siswa Terhadap Jurusan...". The window contains two main sections: "Data Siswa Pemilih Jurusan" and "Data Prestasi Siswa".

Data Siswa Pemilih Jurusan

Nomor Absent	: 1
Nomor Induk	: 0001
Nama	: Agus Waskito Setyo Budi
Kelas	: 1 - 1

Data Prestasi Siswa

Pilihan Pertama	: Ilmu Pengetahuan Alam
Pilihan Kedua	: Ilmu Pengetahuan Sosial

On the right side of the window, there are four buttons: "Sebelum", "Seguduh", "Simpan", and "Close". At the bottom of the window, there are two buttons: "Tampil" and "Keluar".

Gambar 5.11 Tampilan Input Minat Siswa

Setelah semua minat siswa pada jurusan tertentu di entry kemudian baru dilakukan pemilihan matapelajaran yang dijadikan variable penentu dalam pemilihan jurusan seperti form di bawah ini ;

Gambar 5.12 Tampilan Penentuan Matapelajaran Penentu Proses Penjurusan

Dengan entry data yang dilakukan oleh user dalam fasilitas ini maka computer akan melakukan searching nilai raport untuk setiap matapelajaran yang di pilih dalam procedure diatas kemudian melakukan proses perhitungan rata-rata, rangking dan entry daya tampung untuk masing-masing jurusan. Scripts yang digunakan untuk menangani masalah di atas adalah ;

```

procedure TJurus.Button6Click(Sender: TObject);
Var
  Tamp : Array[1..6] Of Real;
begin

```

```

Q.SQL.Clear;
Q.SQL.Add(Format('Select * from "%s",[c:\dataSispen\SiswaAktif.dbf]'));
Q.SQL.Add('Where Kelas = :Kelas');
Q.SQL.Add('Order By NoInduk');

Q.Prepare;
Q.Params[0].AsString := '3';
Q.Open;

If not Q.IsEmpty Then
  Begin
    Q.First;
    Repeat
      If IPA[1] <> " Then
        Begin
          A.SQL.Clear;
          A.SQL.Add(Format('Select * from
"%s",[c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf]'));
          A.SQL.Add('Where NoID = :Kelas');
          A.SQL.Add('And Matpel = :Kelas1');

          A.Prepare;
          A.Params[0].AsString := Q['NoID'];
          A.Params[1].AsString := IPA[1];
          A.Open;

          If Not A.IsEmpty Then Tamp[1] := A['Nilai'] Else Tamp[1] := 0;
        End;

      If IPA[2] <> " Then
        Begin
          A.SQL.Clear;
          A.SQL.Add(Format('Select * from
"%s",[c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf]'));
          A.SQL.Add('Where NoID = :Kelas');
          A.SQL.Add('And Matpel = :Kelas1');

          A.Prepare;
          A.Params[0].AsString := Q['NoID'];
          A.Params[1].AsString := IPA[2];
          A.Open;

          If Not A.IsEmpty Then Tamp[2] := A['Nilai'] Else Tamp[2] := 0;
        End;

      If IPA[3] <> " Then

```

```

Begin
  A.SQL.Clear;
  A.SQL.Add(Format('Select                                *                                from
"%s"', ['c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
  A.SQL.Add('Where NoID = :Kelas');
  A.SQL.Add('And Matpel = :Kelas1');

  A.Prepare;
  A.Params[0].AsString := Q['NoID'];
  A.Params[1].AsString := IPA[3];
  A.Open;

  If Not A.IsEmpty Then Tamp[3] := A['Nilai'] Else Tamp[3] := 0;
End;

If IPA[4] <> " Then
  Begin
    A.SQL.Clear;
    A.SQL.Add(Format('Select                                *                                from
"%s"', ['c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
    A.SQL.Add('Where NoID = :Kelas');
    A.SQL.Add('And Matpel = :Kelas1');

    A.Prepare;
    A.Params[0].AsString := Q['NoID'];
    A.Params[1].AsString := IPA[4];
    A.Open;

    If Not A.IsEmpty Then Tamp[4] := A['Nilai'] Else Tamp[4] := 0;
  End;

If IPA[5] <> " Then
  Begin
    A.SQL.Clear;
    A.SQL.Add(Format('Select                                *                                from
"%s"', ['c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
    A.SQL.Add('Where NoID = :Kelas');
    A.SQL.Add('And Matpel = :Kelas1');

    A.Prepare;
    A.Params[0].AsString := Q['NoID'];
    A.Params[1].AsString := IPA[5];
    A.Open;

    If Not A.IsEmpty Then Tamp[5] := A['Nilai'] Else Tamp[5] := 0;
  End;

```

```

Tamp[6] := (Tamp[1] + Tamp[2] + Tamp[3] + Tamp[4] + Tamp[5]) / I;

A.SQL.Clear;
A.SQL.Add(Format('Insert Into "%s"(', [c:\DataSispen\TampIPA.dbf]));
A.SQL.Add('NoInduk,MP1,MP2,MP3,MP4,MP5,Rata,Rangking)
Values
(:NoInduk,:MP1,:MP2,:MP3,:MP4,:MP5,:Rata,:Rangking)');
A.Prepare;
A.Params[0].AsString := Q['NoID'];
A.Params[1].AsFloat := Tamp[1];
A.Params[1].AsFloat := Tamp[2];
A.Params[2].AsFloat := Tamp[3];
A.Params[3].AsFloat := Tamp[4];
A.Params[4].AsFloat := Tamp[5];
A.Params[5].AsFloat := Tamp[6];

A.ExecSQL;

If IPS[1] <> " Then
Begin
  A.SQL.Clear;
  A.SQL.Add(Format('Select
"%s"', [c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
  A.SQL.Add('Where NoID = :Kelas');
  A.SQL.Add('And Matpel = :Kelas1');

  A.Prepare;
  A.Params[0].AsString := Q['NoID'];
  A.Params[1].AsString := IPS[1];
  A.Open;

  If Not A.IsEmpty Then Tamp[1] := A['Nilai'] Else Tamp[1] := 0;
End;

If IPS[2] <> " Then
Begin
  A.SQL.Clear;
  A.SQL.Add(Format('Select
"%s"', [c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
  A.SQL.Add('Where NoID = :Kelas');
  A.SQL.Add('And Matpel = :Kelas1');

  A.Prepare;
  A.Params[0].AsString := Q['NoID'];
  A.Params[1].AsString := IPS[2];

```

```

A.Open;

If Not A.IsEmpty Then Tamp[2] := A['Nilai'] Else Tamp[2] := 0;
End;

If IPS[3] <> " Then
Begin
A.SQL.Clear;
A.SQL.Add(Format('Select * from
"%s",[c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf]'));
A.SQL.Add('Where NoID = :Kelas');
A.SQL.Add('And Matpel = :Kelas1');

A.Prepare;
A.Params[0].AsString := Q['NoID'];
A.Params[1].AsString := IPS[3];
A.Open;

If Not A.IsEmpty Then Tamp[3] := A['Nilai'] Else Tamp[3] := 0;
End;

If IPS[4] <> " Then
Begin
A.SQL.Clear;
A.SQL.Add(Format('Select * from
"%s",[c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf]'));
A.SQL.Add('Where NoID = :Kelas');
A.SQL.Add('And Matpel = :Kelas1');

A.Prepare;
A.Params[0].AsString := Q['NoID'];
A.Params[1].AsString := IPS[4];
A.Open;

If Not A.IsEmpty Then Tamp[4] := A['Nilai'] Else Tamp[4] := 0;
End;

If IPS[5] <> " Then
Begin
A.SQL.Clear;
A.SQL.Add(Format('Select * from
"%s",[c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf]'));
A.SQL.Add('Where NoID = :Kelas');
A.SQL.Add('And Matpel = :Kelas1');

A.Prepare;

```

```

A.Params[0].AsString := Q['NoID'];
A.Params[1].AsString := IPS[5];
A.Open;

If Not A.IsEmpty Then Tamp[5] := A['Nilai'] Else Tamp[5] := 0;
End;

Tamp[6] := (Tamp[1] + Tamp[2] + Tamp[3] + Tamp[4] + Tamp[5]) / 1;

A.SQL.Clear;
A.SQL.Add(Format('Insert Into "%s"(', [c:\DataSispen\TampIPS.dbf]));
A.SQL.Add('NoInduk,MP1,MP2,MP3,MP4,MP5,Rata,Rangking)
Values
(:NoInduk,:MP1,:MP2,:MP3,:MP4,:MP5,:Rata,:Rangking)');
A.Prepare;
A.Params[0].AsString := Q['NoID'];
A.Params[1].AsFloat := Tamp[1];
A.Params[2].AsFloat := Tamp[2];
A.Params[3].AsFloat := Tamp[3];
A.Params[4].AsFloat := Tamp[4];
A.Params[5].AsFloat := Tamp[5];
A.Params[6].AsFloat := Tamp[6];
A.ExecSQL;

If BHS[1] <> " Then
Begin
A.SQL.Clear;
A.SQL.Add(Format('Select
*
"%s"', [c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
A.SQL.Add('Where NoID = :Kelas');
A.SQL.Add('And Matpel = :Kelas1');

A.Prepare;
A.Params[0].AsString := Q['NoID'];
A.Params[1].AsString := BHS[1];
A.Open;

If Not A.IsEmpty Then Tamp[1] := A['Nilai'] Else Tamp[1] := 0;
End;

If BHS[2] <> " Then
Begin
A.SQL.Clear;
A.SQL.Add(Format('Select
*
"%s"', [c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
A.SQL.Add('Where NoID = :Kelas');

```

```

A.SQL.Add('And Matpel = :Kelas1');

A.Prepare;
A.Params[0].AsString := Q['NoID'];
A.Params[1].AsString := BHS[2];
A.Open;

If Not A.IsEmpty Then Tamp[2] := A['Nilai'] Else Tamp[2] := 0;
End;

If BHS[3] <> " Then
Begin
  A.SQL.Clear;
  A.SQL.Add(Format('Select * from
"%s",['c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
  A.SQL.Add('Where NoID = :Kelas');
  A.SQL.Add('And Matpel = :Kelas1');

  A.Prepare;
  A.Params[0].AsString := Q['NoID'];
  A.Params[1].AsString := BHS[3];
  A.Open;

  If Not A.IsEmpty Then Tamp[3] := A['Nilai'] Else Tamp[3] := 0;
End;

If BHS[4] <> " Then
Begin
  A.SQL.Clear;
  A.SQL.Add(Format('Select * from
"%s",['c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
  A.SQL.Add('Where NoID = :Kelas');
  A.SQL.Add('And Matpel = :Kelas1');

  A.Prepare;
  A.Params[0].AsString := Q['NoID'];
  A.Params[1].AsString := BHS[4];
  A.Open;

  If Not A.IsEmpty Then Tamp[4] := A['Nilai'] Else Tamp[4] := 0;
End;

If BHS[5] <> " Then
Begin
  A.SQL.Clear;

```

```

        A.SQL.Add(Format('Select * from
"%s",[c:\DataSispen\RaportT'+AmbilTahun+AmbilSemester+'.dbf']));
        A.SQL.Add('Where NoID = :Kelas');
        A.SQL.Add('And Matpel = :Kelas1');

        A.Prepare;
        A.Params[0].AsString := Q['NoID'];
        A.Params[1].AsString := BHS[5];
        A.Open;

        If Not A.IsEmpty Then Tamp[5] := A['Nilai'] Else Tamp[5] := 0;
        End;

        Tamp[6] := (Tamp[1] + Tamp[2] + Tamp[3] + Tamp[4] + Tamp[5]) / I;

        A.SQL.Clear;
        A.SQL.Add(Format('Insert Into "%s"(',[c:\DataSispen\TampBHS.dbf']));
        A.SQL.Add('NoInduk,MP1,MP2,MP3,MP4,MP5,Rata,Rangking)
(:NoInduk,:MP1,:MP2,:MP3,:MP4,:MP5,:Rata,:Rangking)');
        A.Prepare;
        A.Params[0].AsString := Q['NoID'];
        A.Params[1].AsFloat := Tamp[1];
        A.Params[2].AsFloat := Tamp[2];
        A.Params[3].AsFloat := Tamp[3];
        A.Params[4].AsFloat := Tamp[4];
        A.Params[5].AsFloat := Tamp[5];
        A.Params[6].AsFloat := Tamp[6];
        A.ExecSQL;

        Q.Next;
        Until Q.EOF;

        Q.SQL.Clear;
        Q.SQL.Add(Format('Select * from "%s",[c:\DataSispen\TampIPA.dbf']));
        Q.SQL.Add('Order By Rata Desc');
        Q.Open;

        If not Q.IsEmpty Then
        Begin
            Q.First;
            Repeat
                A.SQL.Clear;
                A.SQL.Add(Format('Update "%s",[c:\DataSispen\TampIPA.dbf']));
                A.SQL.Add('Set Rangking = :Rangking');
                A.Prepare;
                A.Params[0].AsFloat := Q.RecNo;

```

```
A.ExecSQL;

Q.Next;
Until Q.Eof;
End;

Q.SQL.Clear;
Q.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\TampIPS.dbf']));
Q.SQL.Add('Order By Rata Desc');
Q.Open;

If not Q.IsEmpty Then
  Begin
    Q.First;
    Repeat
      A.SQL.Clear;
      A.SQL.Add(Format('Update "%s"', ['c:\DataSispen\TampIPS.dbf']));
      A.SQL.Add('Set Rangking = :Rangking');
      A.Prepare;
      A.Params[0].AsFloat := Q.RecNo;
      A.ExecSQL;

      Q.Next;
    Until Q.Eof;
  End;

Q.SQL.Clear;
Q.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\TampBHS.dbf']));
Q.SQL.Add('Order By Rata Desc');
Q.Open;

If not Q.IsEmpty Then
  Begin
    Q.First;
    Repeat
      A.SQL.Clear;
      A.SQL.Add(Format('Update "%s"', ['c:\DataSispen\TampBHS.dbf']));
      A.SQL.Add('Set Rangking = :Rangking');
      A.Prepare;
      A.Params[0].AsFloat := Q.RecNo;
      A.ExecSQL;

      Q.Next;
    Until Q.Eof;
  End;
```

```

Q.SQL.Clear;
Q.SQL.Add(Format('Select * from "%s"', ['c:\dataSispen\SiswaAktif.dbf']));
Q.SQL.Add('Where Kelas = :Kelas');
Q.SQL.Add('Order By NoInduk');

Q.Prepare;
Q.Params[0].AsString := '3';
Q.Open;

If not Q.IsEmpty Then
  Begin

    Q.First;
    Label3.Caption := Q['NoInduk'];
    Label5.Caption := Q['Nama'];
    Label11.Caption := '2 - '+Q['Indek'];

    A.SQL.Clear;
    A.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\MinatT'+AmbilTahun+'.dbf']));
    A.SQL.Add('Where NoInduk = :Kelas');
    A.Prepare;
    A.Params[0].AsString := Label3.Caption;
    A.Open;

    If not A.IsEmpty Then
      Begin
        Label16.Caption := A['Minat1'];
        Label17.Caption := A['Minat2'];
      End
    Else
      Begin
        Label16.Caption := '-';
        Label17.Caption := '-';
      End;

    A.SQL.Clear;
    A.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\TestIQ.dbf']));
    A.SQL.Add('Where NoInduk = :Kelas');
    A.Prepare;
    A.Params[0].AsString := Label3.Caption;
    A.Open;

    If not A.IsEmpty Then

```

```

Begin
  Label12.Caption := A['IQ'];
End
Else
  Begin
    Label12.Caption := '-';
  End;
A.SQL.Clear;
A.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\TampIPA.dbf']));
A.SQL.Add('Where NoInduk = :NoInduk');
A.Prepare;
A.Params[0].AsString := Label3.Caption;
A.Open;
If not A.IsEmpty Then
  Begin
    Label25.Caption := RealToStr(A['MP1'],1);
    Label26.Caption := RealToStr(A['MP2'],1);
    Label27.Caption := RealToStr(A['MP3'],1);
    Label34.Caption := RealToStr(A['MP4'],1);
    Label35.Caption := RealToStr(A['MP5'],1);
    Label36.Caption := RealToStr(A['Rata'],2);
    Label41.Caption := IntToStr(Round(A['Rangking']));
    Label40.Caption := IntToStr(TIPA);
  End;
A.SQL.Clear;
A.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\TampIPS.dbf']));
A.SQL.Add('Where NoInduk = :NoInduk');
A.Prepare;
A.Params[0].AsString := Label3.Caption;
A.Open;
If not A.IsEmpty Then
  Begin
    Label54.Caption := RealToStr(A['MP1'],1);
    Label53.Caption := RealToStr(A['MP2'],1);
    Label52.Caption := RealToStr(A['MP3'],1);
    Label51.Caption := RealToStr(A['MP4'],1);
    Label50.Caption := RealToStr(A['MP5'],1);
    Label63.Caption := RealToStr(A['Rata'],2);
    Label62.Caption := IntToStr(Round(A['Rangking']));
    Label61.Caption := IntToStr(TIPS);
  End;
A.SQL.Clear;
A.SQL.Add(Format('Select * from "%s"', ['c:\DataSispen\TampBHS.dbf']));
A.SQL.Add('Where NoInduk = :NoInduk');
A.Prepare;
A.Params[0].AsString := Label3.Caption;

```

```

A.Open;
If not A.IsEmpty Then
  Begin
    Label78.Caption := RealToStr(A['MP1'],1);
    Label77.Caption := RealToStr(A['MP2'],1);
    Label76.Caption := RealToStr(A['MP3'],1);
    Label75.Caption := RealToStr(A['MP4'],1);
    Label74.Caption := RealToStr(A['MP5'],1);
    Label87.Caption := RealToStr(A['Rata'],2);
    Label86.Caption := IntToStr(Round(A['Rangking']));
    Label84.Caption := IntToStr(TBHS);
  End;
End;
End
Else MessageDlg('Belum Ada Data Siswa Dengan Kelas Di
atas...!',mtInformation,[mbOK],0);
end;

```

Setelah itu baru dilakukan komparasi antara kemampuan setiap jurusan dengan minat dan daya tampung untuk setiap jurusan seperti form di bawah ini :

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho....

...Seleksi Pemilihan Jurusan...

Data Pribadi Siswa		Data Minat Siswa Terhadap Jurusan	
No. Induk : 0024		Minat I : Ilmu Pengetahuan Alam	
Nama : Candraningrum		Minat II : Ilmu Pengetahuan Sosial	
Kelas : 2 - 2		IQ : 124	

Jurusan Ilmu Pengetahuan Alam	Jurusan Ilmu Pengetahuan Sosial	Jurusan Ilmu Bahasa
Matematika : 8	Ekonomi : 7	Bahasa Indonesia : 7
Fisika : 9	Sejarah : 7	Bahasa Inggris : 8
Kimia : -		
Biologi : -		

Keputusan
Ilmu Pengetahuan Alam
Simpan
By Pass

<u>Rata-Rata</u> : 7,75	<u>Rata-Rata</u> : 6,75	<u>Rata-Rata</u> : 7,5
<u>Rangking</u> : 19	<u>Rangking</u> : 54	<u>Rangking</u> : 21
<u>Daya Tampung</u> : 90	<u>Daya Tampung</u> : 90	<u>Daya Tampung</u> : 45

Keputusan

Daya Tampung

Matapelajaran

Keluar

Gambar 5.13 Tampilan Proses Penjurusan

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho

...Fasilitas Absensi Siswa...

Data SPP Siswa

Nomor Absen : 14
 Nomor Induk : 00114
 Nama : YUSRIN AROFAH Z
 Kelas : 1 - 3

Data Pembayaran SPP

Tanggal : 07/07/2005
 Jenis Absensi : Sakit Ijin Alpa

Gambar 5.4 Tampilan Absensi Siswa

5.5 Fasilitas Ulangan Harian

Merupakan fasilitas entry data ulangan harian kemudian merekapnya dengan bentuk statistika tertentu. Sebelum entry data diperlukan input data berupa kelas, jurusan, indeks kelas, matapelajaran dan ulangan yang ke berapa?. Seperti tampilan dibawah ini :

Program Sistem Informasi Pendidikan Designed By Fuad Ridlo Nugroho

...Input Nilai Ulangan Harian...

Data Kelas dan Ulangan

Kelas : 1
 Jurusan : Belia Penjurusan
 Indeks : 1
 Matapelajaran : Matematika
 Ulangan Ke : 2

Gambar 5.5 Tampilan Awal Input Nilai Ulangan Harian

Setelah procedure di atas dieksekusi maka akan tampil panel seperti di bawah ini :

Gambar 5.6 Tampilan Proses Input Nilai Ulangan Harian

Setelah nilai ulangan harian diisi user tinggal menekan button Simpan maka dengan event `OnClick` maka nilai itu akan dsimpan ke dalam file `NUT'+AmbilTahun+AmbilSemester+'.dbf'` seperti scripts di bawah ini ;

```

procedure TUIHar.Button1Click(Sender: TObject);
Var
  Kode : String;
begin
  If Edit1.Text <> " " Then
    Begin
      A.SQL.Clear;
      A.SQL.Add(Format('Select NoInduk from
"%s"', [c:\DataSispen\NUT'+AmbilTahun+AmbilSemester+'.dbf']));
      A.SQL.Add('Where NoInduk = :NoInduk');
      A.SQL.Add('And Mapel = :Matpel');
      A.Prepare;
      A.Params[0].AsString := Label12.Caption;
      A.Params[1].AsString := Label31.Caption;
      A.Open;

      If not A.IsEmpty Then
    
```

Bab VI

INTEGRASI PROGRAM DAN FASE TESTING

6.1 Integrasi Program

Integrasi program merupakan tahap akhir dari fase *coding*, setelah semua sub program dinyatakan bebas dari kesalahan (error) secara syntack maupun semantik, maka dilakukan penggabungan sub-sub program menjadi program utama. Proses penggabungan ini lazim disebut **proses integrasi**. Dalam proses integrasi semua unit dan form yang digunakan dalam program sistem informasi pendidikan dengan menggunakan IDE (Integrated Development Environment) Borland DELPHI 7.0 digunakan instuksi standard :

```
NamaForm := TNamaForm.Create(Self);  
NamaForm.ShowModal;
```

Kedua instuksi diatas berfungsi untuk membuat suatu *form* dan mengalokasikan dalam memori (*TNamaForm.Create(Self)*) dan form ini akan dikenal dengan mana *NamaForm*, sedangkan untuk mengaktifkannya digunakan *keyword Showmodal*. Intruksi ini harus ~~harus~~ *dibindingkan* menu yang dijadikan *activating point*. *Form* yang diaktifkan dalam ~~suatu~~ *program* utama harus dikenal oleh program utama. Cara mendeklarasikan suatu *form* dalam program utama digunakan fasilitas *add project* (atau menekan Shift + F11), ~~kemudian Nama file harus dideklarasikan~~ dalam block program utama setelah *keyword Uses*. Seperti dalam ~~program~~ *program* utama sistem informasi pendidikan di bawah ini :

```
unit SIPendidikan;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls, StdCtrls, Menus, jpeg, DB, DBTables;
```

```
type
```

```
TProgUtama = class(TForm)  
  MM: TMainMenu;  
  Image1: TImage;  
  {deklarasi komponen visual lainnya}  
  procedure FormCreate(Sender: TObject);  
  {deklarasi prosedur lainnya}  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

```
var
```

```
ProgUtama: TProgUtama;
```

```
implementation
```

```
Uses
```

```
UserbaUsahaUnit,  
HapusPassword,  
{Deklarasi nama unit yang digunakan dalam program utama}
```

```
{SR *.dfm}
```

Sedangkan untuk mengaktifkan suatu *form* menggunakan menu

OnClick dari komponen visual MainMenu seperti *scripts* di bawah ini :

```
procedure TProgUtama.PendaftaranReguler1Click(Sender: TObject);  
Var  
  Daftar : Tdaftar;  
begin  
  Ddaftar := TDaftar.Create(Self);  
  Daftar.ShowModal;  
  Daftar.Free;  
end;
```



6.2 Testing

Testing merupakan suatu fase dalam 5 *life cycles* yang berfungsi untuk mengecek apakah *software* sudah terbebas dari kesalahan sintack maupun semantik

Ada beberapa tahapan *testing* yang lazim dalam industri *software*, yaitu :

6.2.1 Alpha Test

Yaitu suatu *test* yang dilakukan kepada suatu *product software* dengan mengacu pada kriteria yang ditetapkan sendiri oleh seorang *programmer*.

Dalam *alpha test* terjadi beberapa tahapan *test* di antaranya :

❖ *Testing* Modul

Yaitu melakukan *test* pada masing-masing modul program apakah suatu modul program terbebas dari kesalahan sintack maupun semantik. Biasanya *testing* modul ini dilakukan setiap suatu modul program telah selesai dibuat.

❖ *Testing* Integrasi

Yaitu *test* yang dilakukan suatu modul diintegrasikan, biasanya *test* ini hanya melakukan *test* terhadap kecepatan program dan penggunaan *resource* seperti penggunaan memori dan *space*. Tetapi biasanya *programmer* jarang melakukan *test* ini karena sewaktu program telah diintegrasikan dan *dcompile* maka masalah ini menjadi wilayah IDE DELPHI.

❖ *Test Program Secara Menyeluruh*

Yaitu *test* yang dilakukan untuk memastikan apakah suatu *product software* sesuai dengan deskripsi sistem yang telah dijabarkan dalam fase desain sistem. Titik berat test ini tidak lagi pada masalah sintack tetapi lebih menitik beratkan pada masalah semantik.

Dalam fase *test* sistem informasi pendidikan ini kriteria penerimaan (*acceptance criteria*) yang ditetapkan adalah :

1. Terbebas dari kesalahan sintack bahasa pemrograman Delphi
2. Terbebas dari kesalahan logika terutama dalam penggunaan operasi repetisi maupun operasi logik seperti penggunaan **IF ... Then Else** maupun **Case ... Of**
3. Terbebas dari kesalahan konversi maupun pembagian dengan nol.
4. Terbebas dari kesalahan penyimpanan data maupun pembacaan data.
5. Hasil dari program yang dikembangkan sesuai dengan deskripsi sistem yang telah ditetapkan dalam fase desain sistem.

6.2.2 *Betha Test*

Yaitu suatu *test* yang dilakukan oleh *user* kepada suatu *product software* dengan mengacu pada kriteria yang ditetapkan oleh *user*. Test ini berfungsi untuk menyamakan persepsi antara pembuat sistem dengan *customer*. Pada test ini dititikberatkan pada hasil suatu *product software* apakah sama dengan yang diinginkan oleh *customer*

atau tidak. Program sistem informasi pendidikan ini telah ditest dengan menggunakan kriteria customer di SMA DIPONEGORO Pemalang.

BAB VII

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari Penelitian dan pembahasan dari bab I sampai dengan bab VI dapat ditarik beberapa kesimpulan yaitu :

1. Untuk membuat proyek *software* yang berukuran besar misalnya Sistem Informasi Pendidikan diperlukan suatu teknik *design* program yang mengacu pada kaidah *5 life cycles*.
2. Selain menggunakan kerangka *5 life cycles* yang diperlukan untuk teknik dokumentasi sebagai media komunikasi antar termin.
3. Dengan adanya Sistem Informasi Pendidikan ini, permasalahan seputar administrasi pendidikan manual yang sarat dengan keterbatasan manusiawi dapat dieliminasi dengan menggunakan sistem komputerisasi.
4. Dalam desain sebuah data base menggunakan system database pascal klasik memiliki keunggulan dari segi keamanan (data tidak bisa dibaca oleh program lain) tetapi kelemahannya akses data lambat dan pembuatan scripts relative lebih sulit. sedangkan dengan menggunakan data base yang include dalam Delphi maka pembuatan scripts lebih mudah, akses data cepat tetapi mempunyai keaamanan data yang rendah.

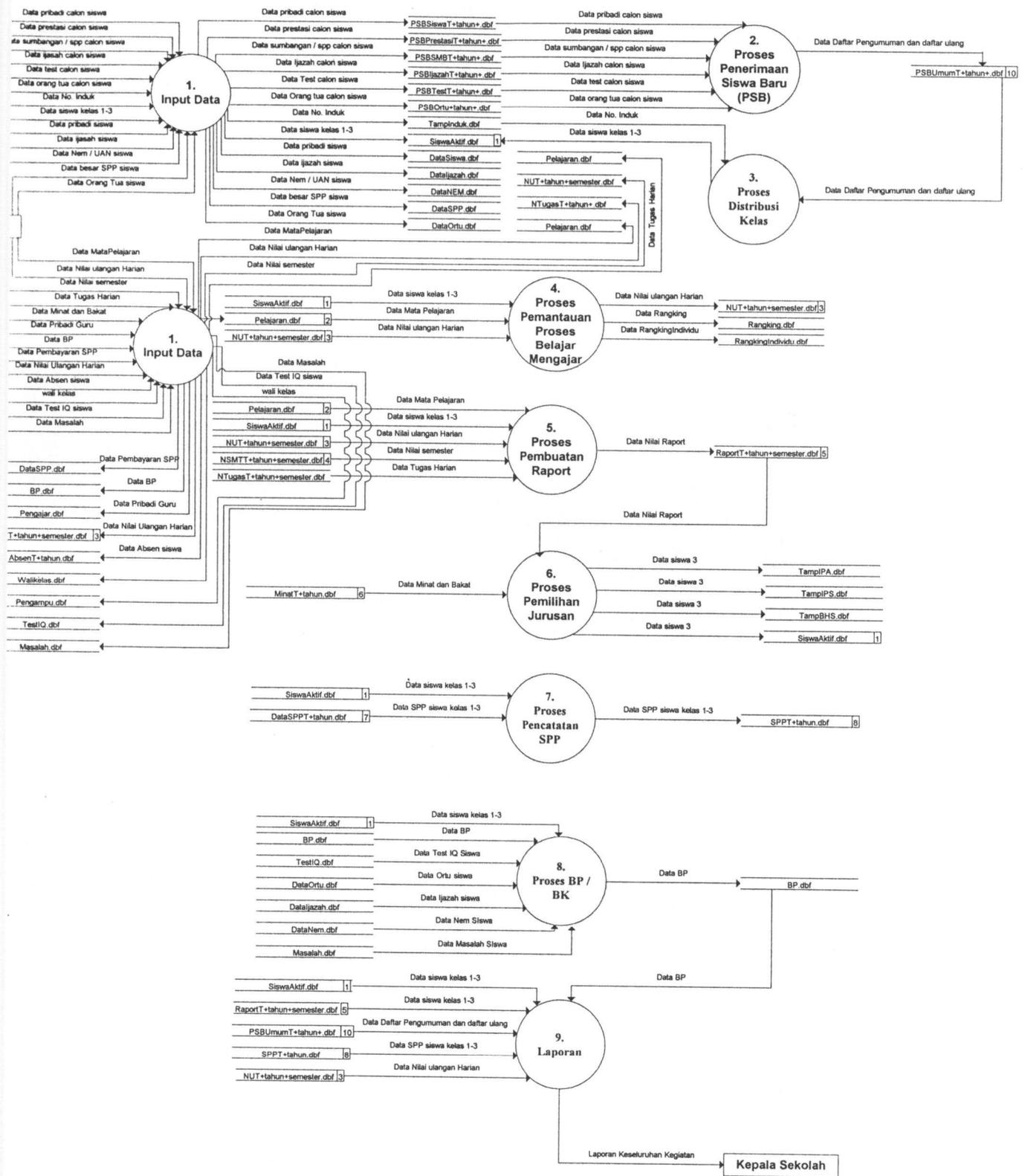
B. Saran

1. *Software* Sistem Informasi Pendidikan ini perlu dikembangkan lagi di antaranya masalah keamanan data. Ini dapat diselesaikan dengan memprotek password dengan menggunakan format .dat melalui bahasa pemrograman pascal.
2. Perlunya pengembangan dari menu-menu tampilan yang lebih komplit misalnya adanya fasilitas menu pembuatan jadwal mata pelajaran.
3. Perlu adanya Penambahan Fasilitas Tampilan Grafik yang dapat dicetak sehingga proses pelaporan Grafik dari sistem Informasi Pendidikan ini lebih baik dan mudah dikontrol.

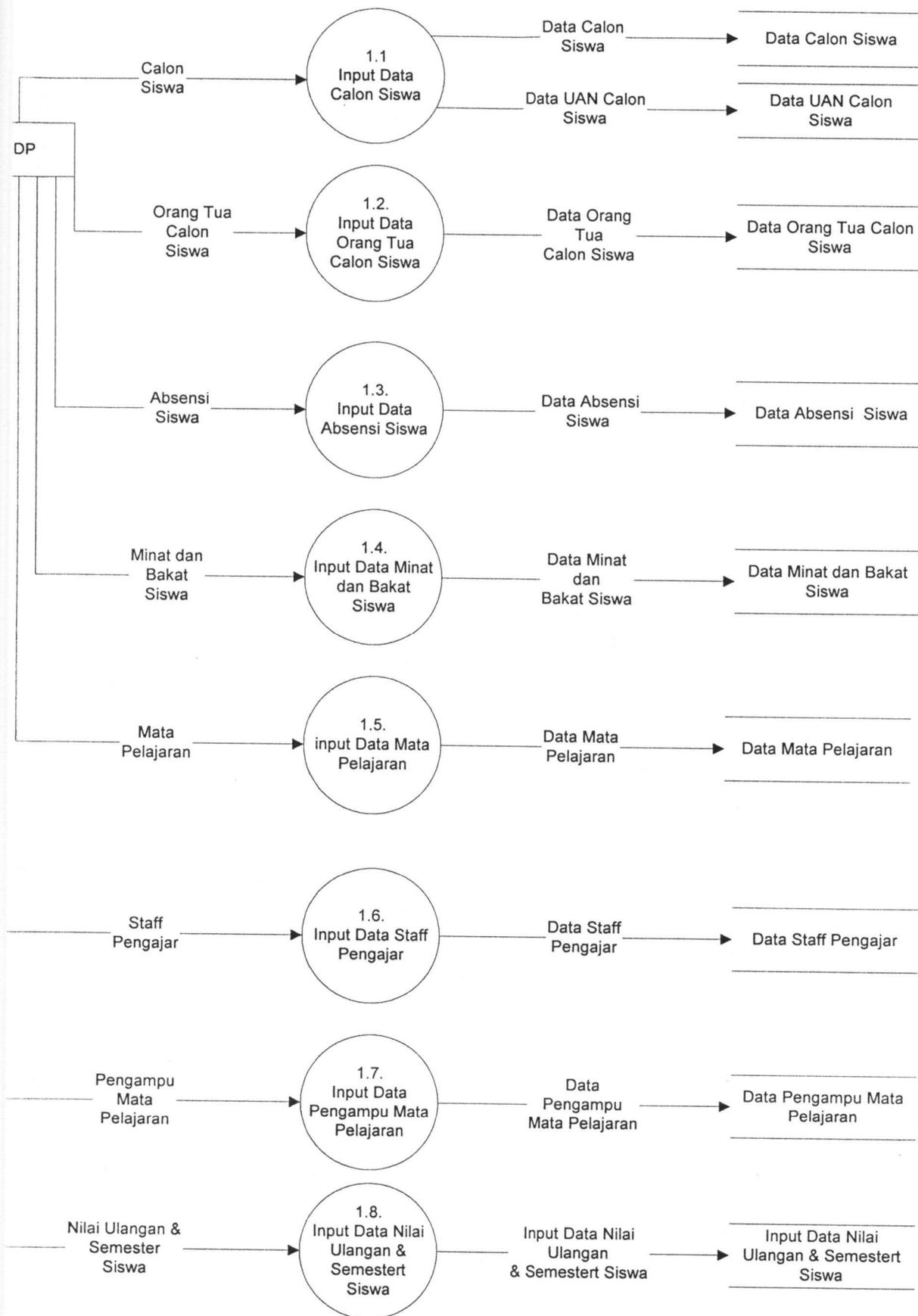
DAFTAR PUSTAKA

- [ADISI] Henry c.lucas Jr, Analisis. Desain dan Implementasi Sistem Informasi, Penerbit Erlangga, Jakarta, 1993
- [PAPPL] Tavri D Mahyuzir, Pengantar Analisis dan Perancangan Perangkat Lunak, PT. Elex Media komputindo, Jakarta,1991
- [MACSOPO]Leman, Mengembangkan Aplikasi Client/Server dengan Oracle Power Objects, PT. Elex Media komputindo, Jakarta,1997
- [AICTA] Yoshiaki Shirai, Artificial Intellegence, Concepts, Techniques and Aplications, John Wiley & Sons Ltd, Singapore,1984
- [SPSA] Stewart V Hoover – Ronald F. Perry, Simulation Problem Solving Approach, Addison – Weshley Publishing Company, Canada, 1989
- [PMD] Jon Matcho – David R. Faulkner, Panduan Menggunakan Delphi, Andi Offset , Yogjakarta, 1997
- [PPD] Wahana Komputer Semarang, Pemrograman Praktis Dengan Delphi, Andi Yogyakarta, Yogyakarta,1997
- [MD] Marco Cantu, Mastering Delphi, Sybex. Inc, Amerika,1999

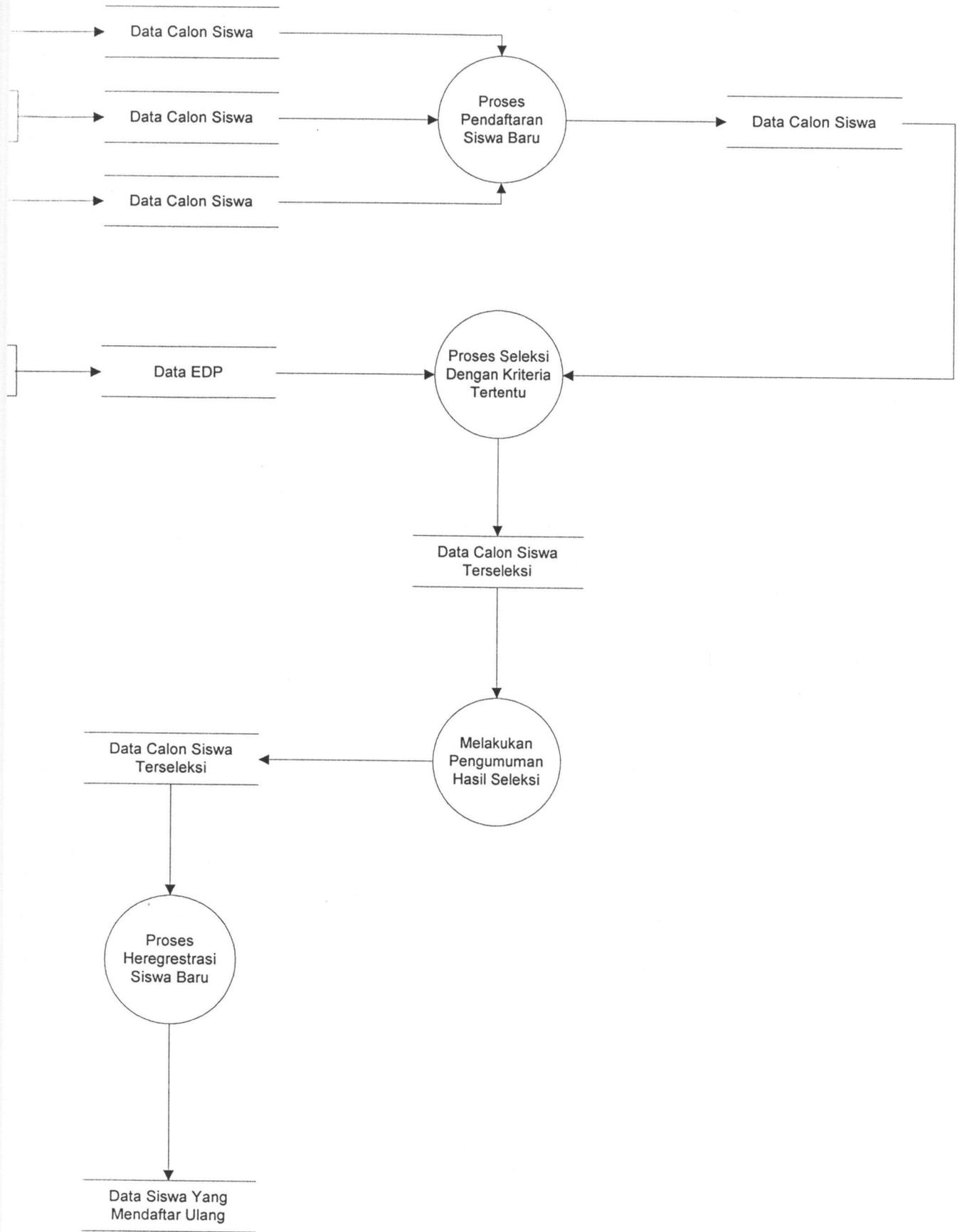
LAMPIRAN



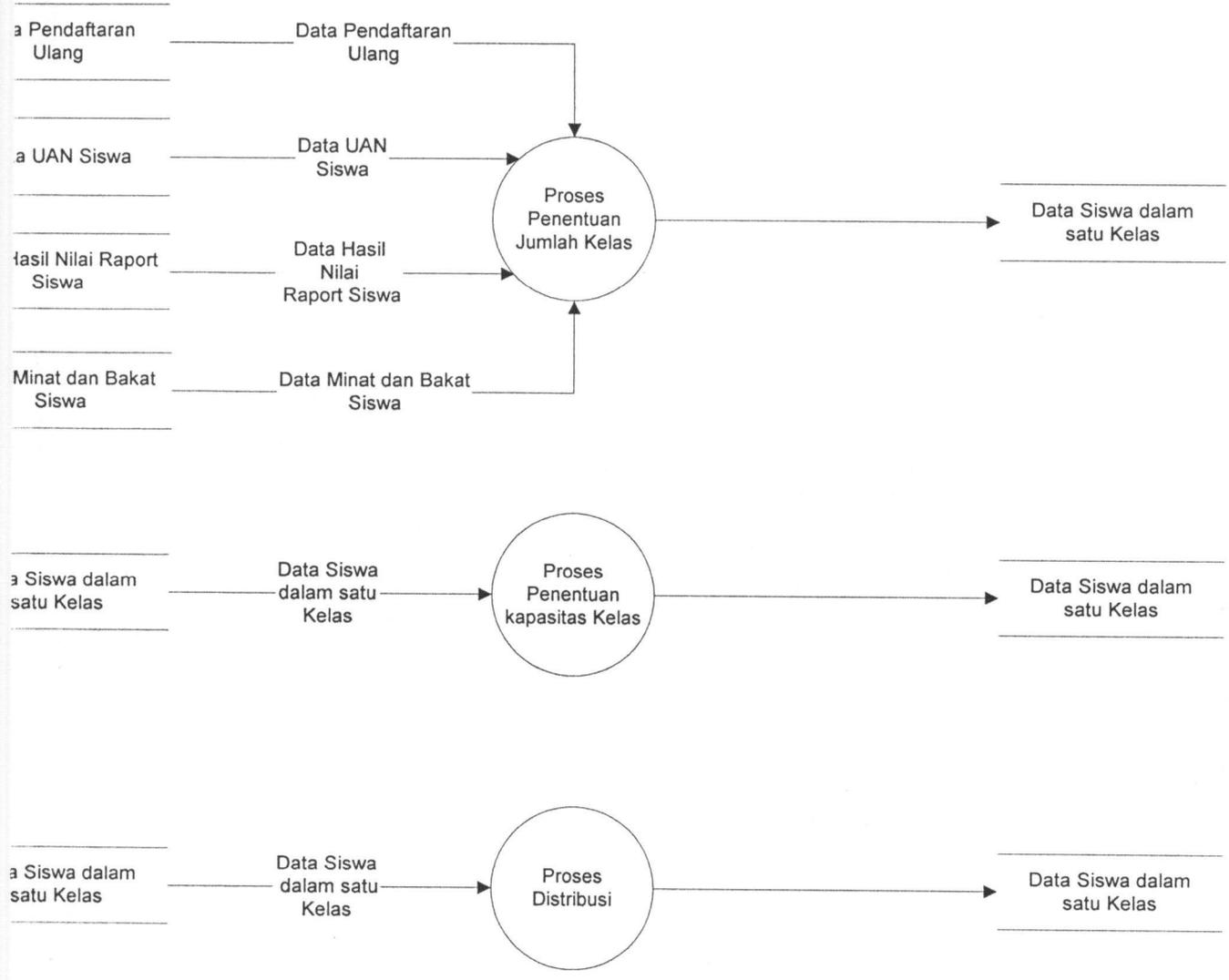
Gambar 4.12. DFD Level-1 Sistem Informasi Pendidikan



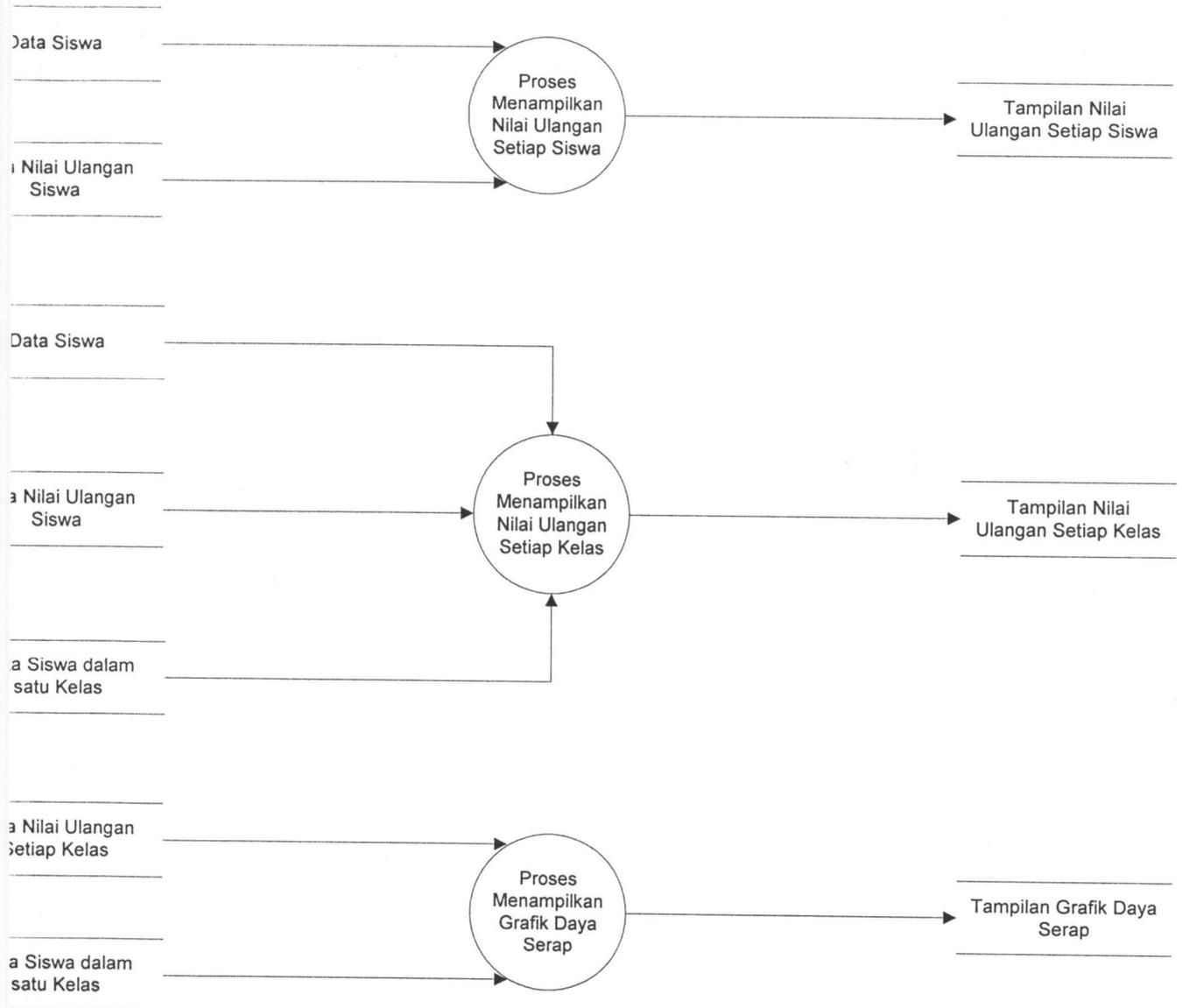
Gambar 4.13.DFD Level 2 Input data



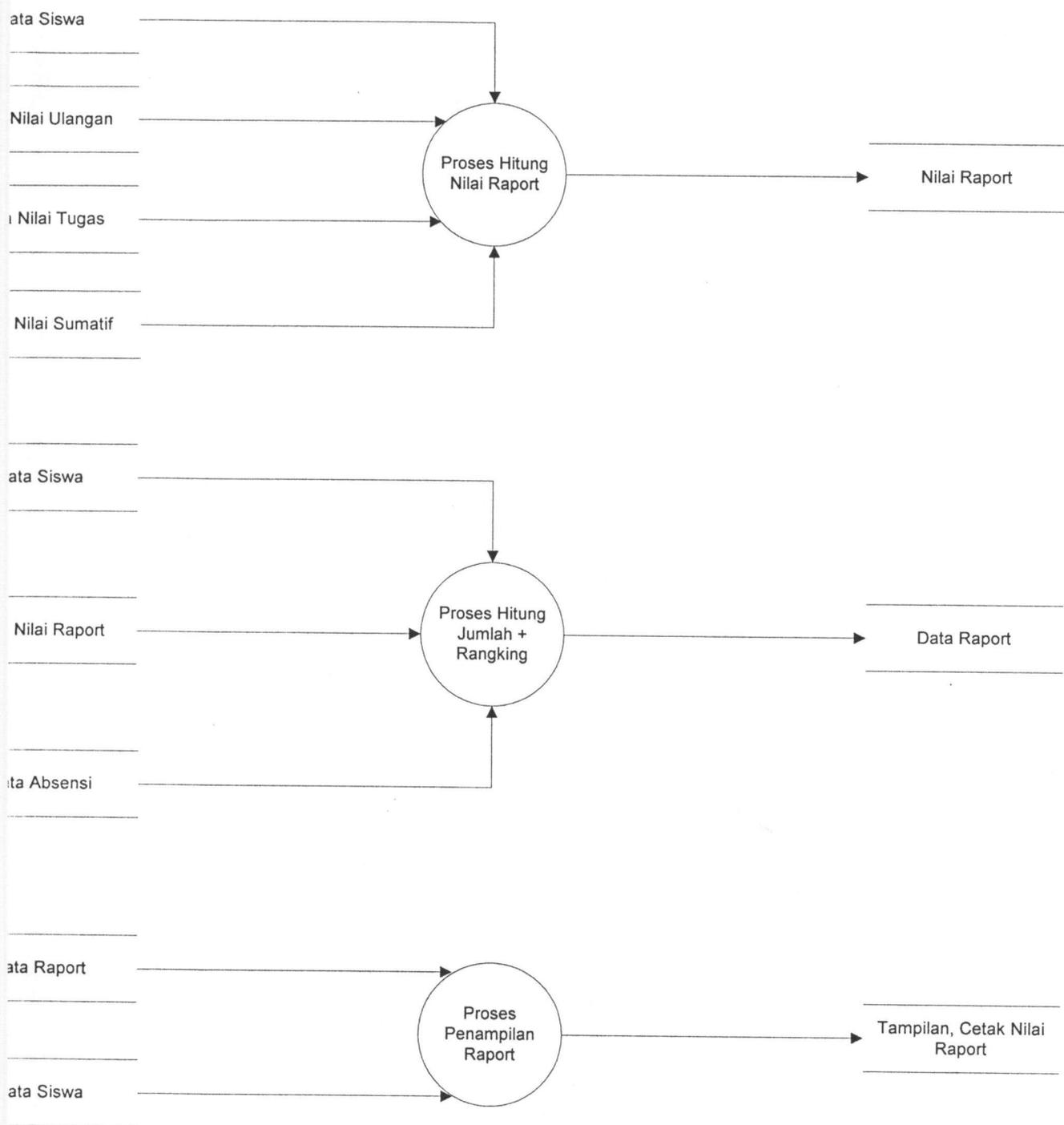
Gambar 4.14. DFD Level 2 Penerimaan Siswa Baru (PSB)



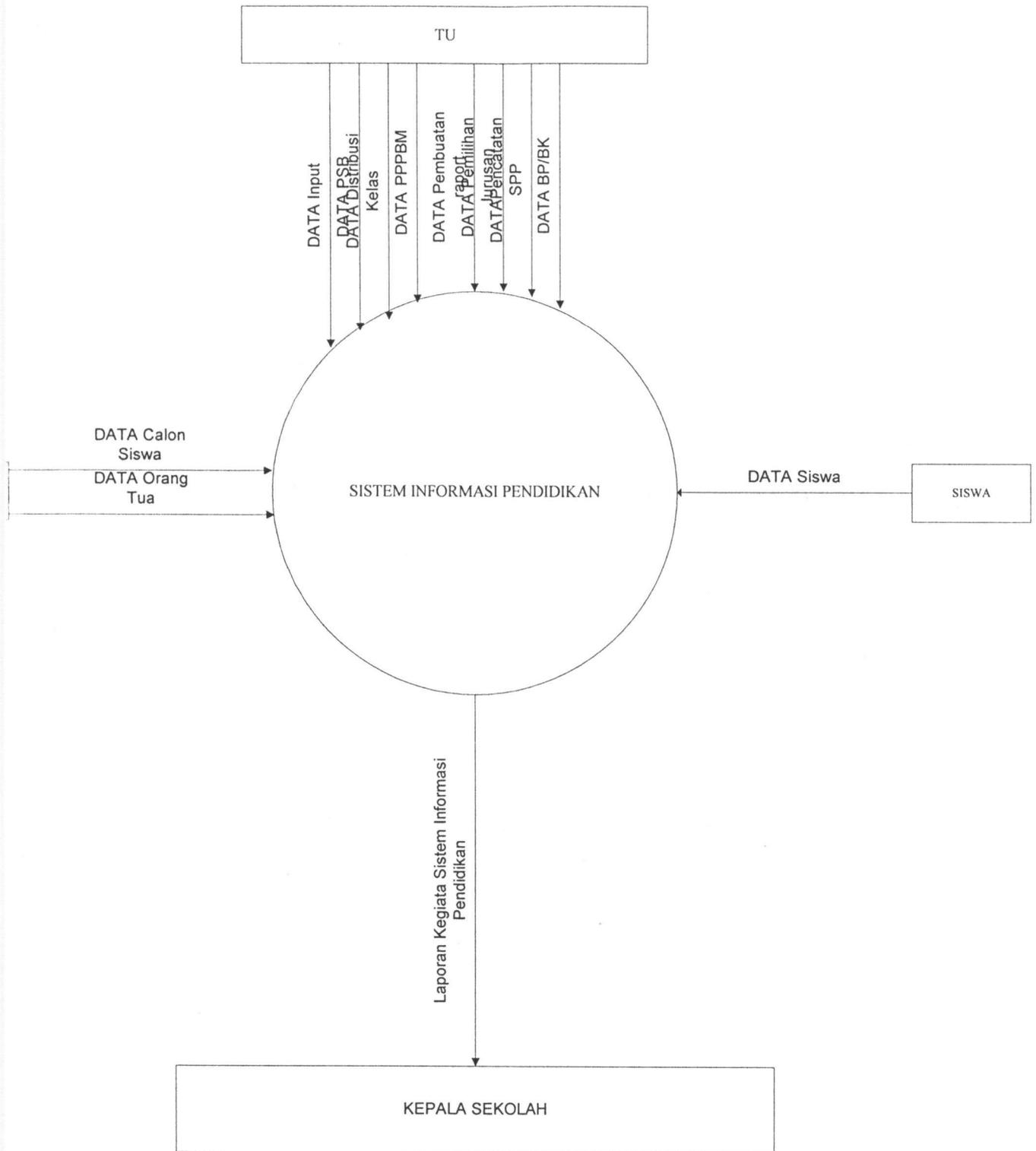
Gambar 4.15. DFD Level 2 Proses Distribusi Kelas



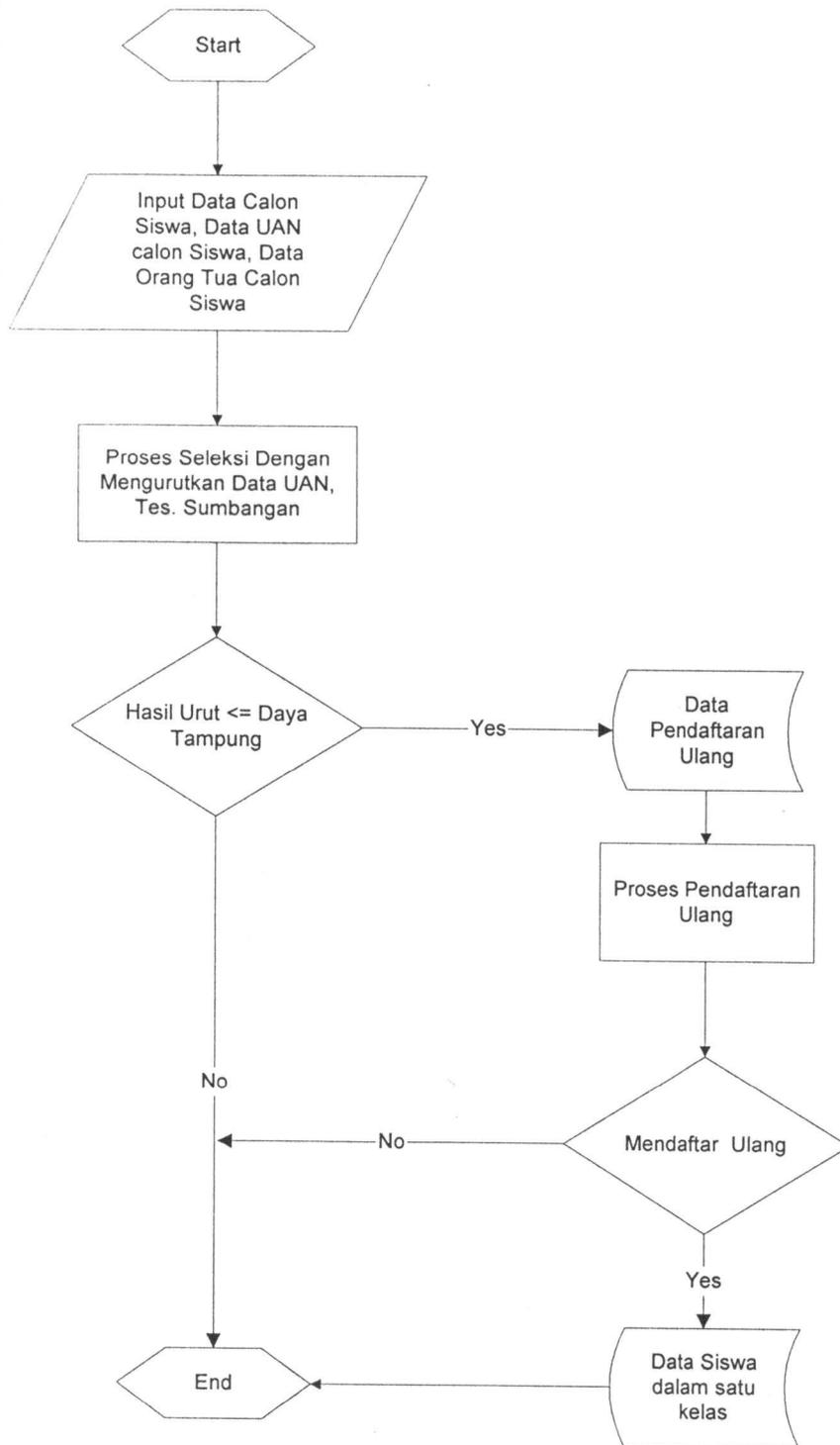
Gambar 4.16. DFD Level 2 Proses Pemantauan Belajar Mengajar



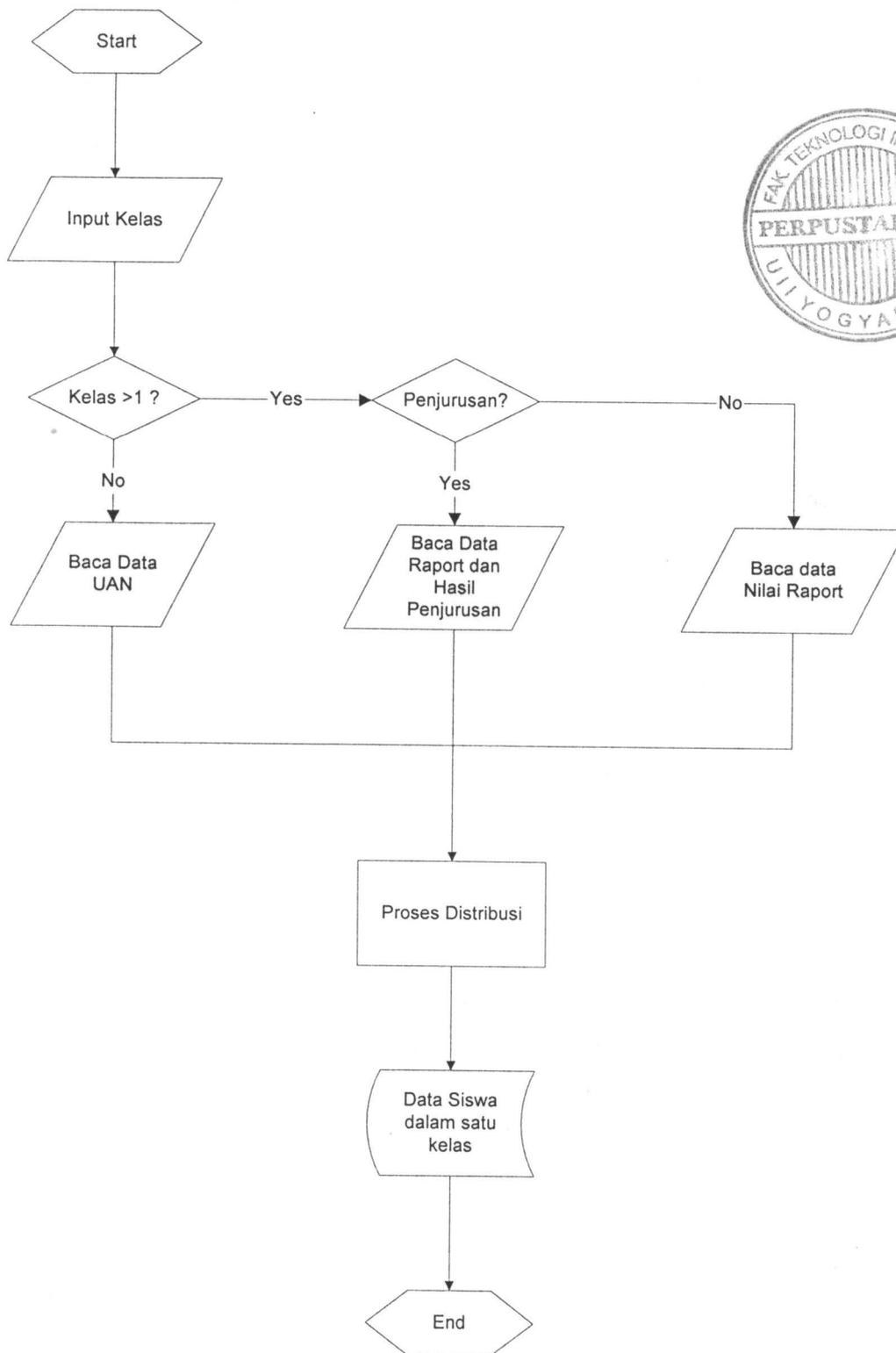
Gambar 4.17. DFD Level 2 Proses Pembuatan Raport



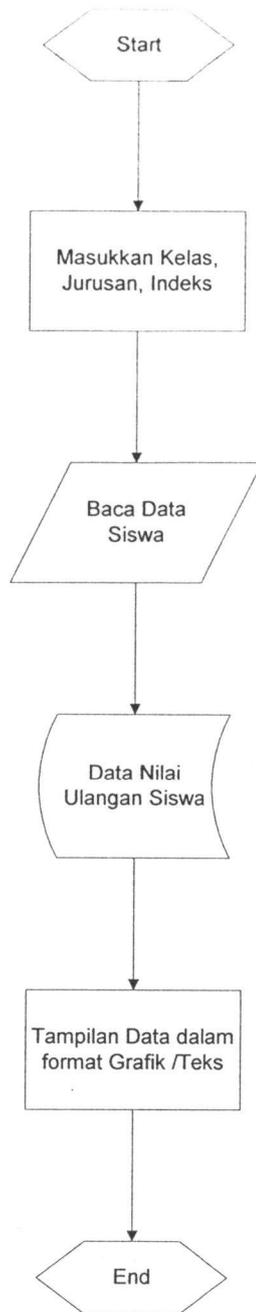
Gambar 4.18. DIAGRAM KONTEKS



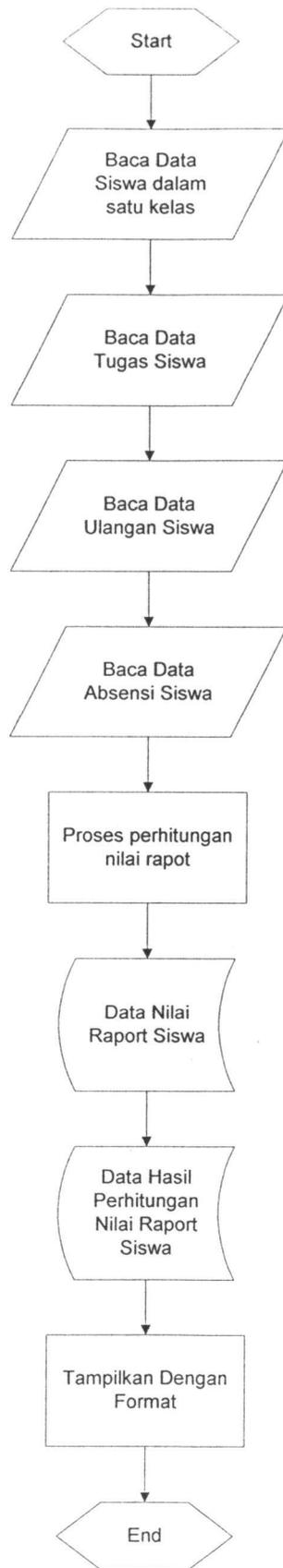
Gambar 4.1. Flowchart Pendaftaran Siswa Baru



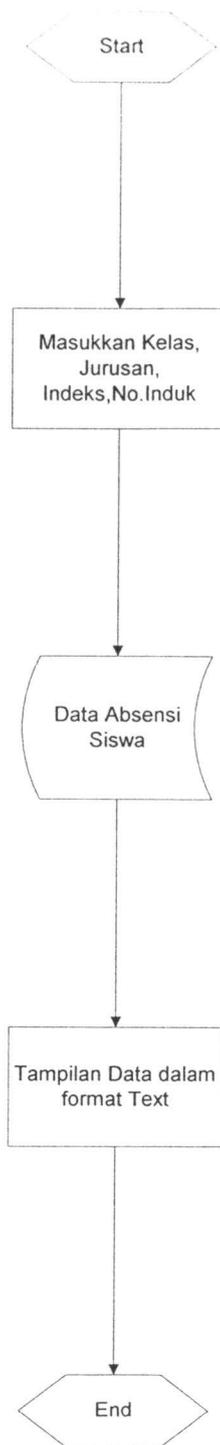
Gambar 4.2. Flowchart Distribusi Kelas



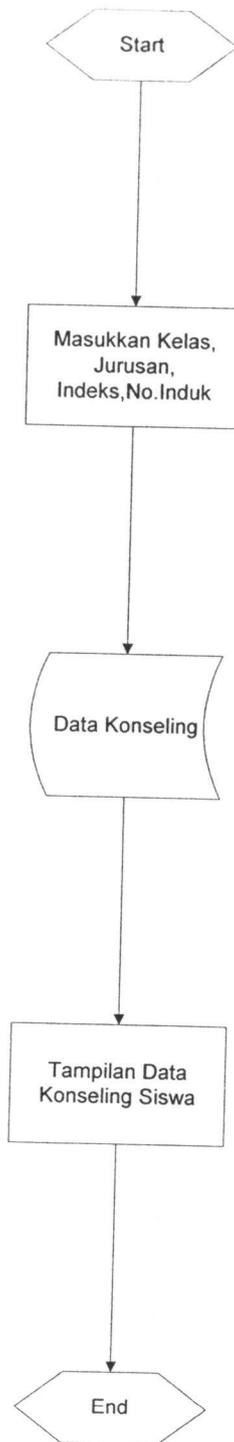
Gambar 4.3. Flowchart Proses Belajar Mengajar



Gambar 4.4. Flowchart Pembuatan Raport



Gambar 4.5. Flowchart Proses Pembayaran SPP



Gambar 4.6. Flowchart Proses BP/
BK