



Continuing Education Program  
Center for Computing and Information Technology  
Fakultas Teknik Universitas Indonesia



**Proyek Rapid Application Development with Sharia Compliance**  
**“Aplikasi Penilaian Siswa Sekolah Menengah Pertama”**

Disusun Oleh:  
Wafa Hasnaghina Ulfah

**Kelas : 3 TIPS 3**

**Faculty : Indah Ayu Yuliani, ST., MM.**

**CEP-CCIT FTUI BEKERJASAMA DENGAN**  
**PUSBANGSITEK UIN SYARIF HIDAYATULLAH JAKARTA**  
**2013**

**KATA PENGANTAR**

Puji dan syukur penulis panjatkan kepada Allah SWT. Yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan makalah yang berjudul “Aplikasi Penilaian Siswa Sekolah Menengah Pertama” ini tepat pada waktu yang telah ditentukan oleh pihak *faculty* untuk memenuhi tugas proyek semester 3 (Tiga).

Dalam penyusunan makalah ini penulis banyak mengalami kesulitan, namun berkat adanya bantuan dari teman-teman, bimbingan dari *faculty*, dan dari beberapa situs internet untuk menambah informasi dalam pembuatan makalah yang penulis buat, maka pada akhirnya penyusunan makalah ini dapat terselesaikan dengan baik dan tepat pada waktu yang telah ditentukan.

Tidak lupa penulis ucapkan terima kasih kepada seluruh *faculty* yang telah memberikan tugas ini kepada penulis, karena melalui tugas ini penulis dapat mengetahui lebih banyak lagi mengenai teknologi yang telah dikembangkan. Penulis tidak lupa berterima kasih kepada orang tua dan rekan-rekan kelompok maupun rekan kelompok lain yang telah mendukung baik secara moril maupun material sehingga tugas ini dapat diselesaikan dengan baik dan tepat pada waktunya.

Di dalam penyusunan laporan ini, penulis menyadari bahwa laporan ini masih banyak kekurangan dan jauh dari kesempurnaan disebabkan oleh beberapa faktor diantaranya adalah karena waktu dan kemampuan penulis yang terbatas. Oleh karena itu, penulis mengharapkan saran dan kritik yang bersifat membangun demi perbaikan tugas ini dan untuk tugas-tugas selanjutnya. Atas kerjasamanya, penulis mengucapkan terima kasih.

Depok, Oktober 2013

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	iii
DAFTAR GAMBAR.....	vi
DAFTAR TABLE .....	viii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan Penulisan .....	2
1.4 Batasan Masalah .....	2
1.5 Metodologi Penulisan .....	2
1.6 Sistematika Penulisan .....	2
BAB II LANDASAN TEORI.....	4
2.1 <i>Rapid Application Development</i> .....	4
2.1.1 Definisi RAD .....	4
2.1.2 Sejarah RAD.....	4
2.2 C# (C Sharp) .....	5
2.2.1 Sejarah C Sharp .....	5
2.3 Visual Basic .Net .....	7
2.3.1 .Net Framework .....	7
2.4 Basis Data .....	8
2.4.1 Konsep DBMS.....	9
2.4.2 Entity Relationship Diagram .....	10
2.4.3 Komponen Entity Relationship Diagram.....	10
2.4.4 SQL Server 2005 .....	11
2.5 Unified Modeling Language (UML) .....	12
2.5.1. <i>Use case diagram</i> .....	12
BAB III ANALISA PERMASALAHAN.....	14
3.1 Gambaran Umum Aplikasi .....	14

3.2	Analisa Sistem .....	14
3.2.1	ER Diagram Aplikasi Penilaian Siswa SMP .....	15
3.2.2	Use Case Diagram Aplikasi Penilaian Siswa SMP .....	16
	Gambar 3. 2 Use Case Aplikasi Penilaian Siswa SMP .....	16
3.2.3	Desain Menu Utama .....	17
3.2.5	Desain Form Update .....	23
3.2.6	Desain Form Lihat .....	24
3.2.7	Desain Laporan .....	25
	BAB IV IMPLEMENTASI SISTEM.....	26
4.1	Form Login .....	26
4.2	Form Tambah Akun.....	29
4.3	File .....	32
4.3.1	Form Ganti Password .....	32
4.3.2	Logout.....	33
4.3.3	Exit .....	33
4.4	Help.....	33
4.5	Menu Administrator.....	35
4.5.1	Form Input Data Guru .....	35
4.5.2	Form Update Data Guru .....	37
4.5.3	Form Lihat Data Guru .....	39
4.5.4	Laporan Data Guru .....	40
4.5.5	Form Lihat Audit .....	41
4.6	Menu Tata Usaha.....	41
4.6.1	Form Input Data Siswa .....	42
4.6.2	Form Lihat Data Siswa .....	44
4.6.3	Form Update Data Siswa .....	45
4.7	Menu Guru.....	46
4.7.1	Form Input Nilai Siswa.....	47
4.7.2	Form Ekstrakurikuler Siswa .....	49
4.7.3	Form Aktifitas Siswa .....	50

4.7.4	Nilai Raport .....	52
4.7.5	Aktifitas Siswa.....	53
4.8	Menu Bagian Kurikulum .....	54
4.8.1	Form Data Mata Pelajaran .....	54
4.8.2	Form Data Kelas .....	56
4.8.3	Form Data Ekstrakurikuler .....	57
4.8.4	Form Data Wali Kelas .....	58
4.9	Menu Kepala Sekolah.....	59
4.10	Laporan .....	59
BAB V PENUTUP .....		61
5.1	Kesimpulan .....	61
5.2	Saran .....	61
DAFTAR PUSTAKA.....		62

## DAFTAR GAMBAR

Gambar 3. 1 ER Diagram Aplikasi Penilaian Siswa SMP .....	15
Gambar 3. 2 Use Case Aplikasi Penilaian Siswa SMP .....	16
Gambar 3. 3 Desain Menu Administrator .....	17
Gambar 3. 4 Desain Menu Tata Usaha.....	17
Gambar 3. 5 Desain Menu Bagian Kurikulum .....	18
Gambar 3. 6 Desain Menu Guru.....	18
Gambar 3. 7 Desain Menu Kepala Sekolah.....	18
Gambar 3. 8 Desain Form Input Data Siswa .....	19
Gambar 3. 9 Desain Form Simpan Data Guru.....	20
Gambar 3. 10 Desain Form Data Ekstrakurikuler .....	20
Gambar 3. 11 Desain Form Nilai Siswa .....	21
Gambar 3. 12 Desain Form Aktifitas Siswa .....	22
Gambar 3. 13 Desain Form Ektrakurikuler Siswa.....	23
Gambar 3. 14 Desain Form Update Data Siswa .....	24
Gambar 3. 15 Desain Form Lihat Data Siswa .....	25
Gambar 3. 16 Desain Laporan.....	25
Gambar 4. 1 Tampilan Form Login.....	26
Gambar 4. 2 Form Tambah Akun.....	30
Gambar 4. 3 Form Ganti Password .....	32
Gambar 4. 4 Form How Do I – Tab Icons.....	34
Gambar 4. 5 Form How Do I – Tab Key Press .....	34
Gambar 4. 6 Menu Administrator.....	35
Gambar 4. 7 Form Input Data Guru .....	36
Gambar 4. 8 Form Update Data Guru .....	38
Gambar 4. 9 Form Lihat Data Guru .....	40
Gambar 4. 10 Laporan Data Guru .....	40
Gambar 4. 11 Form Lihat Audit .....	41
Gambar 4. 12 Form Menu Tata Usaha .....	42

Gambar 4. 13 Form Data Siswa.....	43
Gambar 4. 14 Form Lihat Data Siswa .....	44
Gambar 4. 15 Form Update Siswa.....	45
Gambar 4. 16 Menu Guru.....	47
Gambar 4. 17 Form Nilai Mata Pelajaran.....	47
Gambar 4. 18 Form Ekstrakurikuler Siswa .....	49
Gambar 4. 19 Form Aktifitas Siswa .....	51
Gambar 4. 20 Form Lihat Nilai Raport .....	53
Gambar 4. 21 Form Lihat Aktifitas Siswa.....	53
Gambar 4. 22 Form Menu Bagian Kurikulum .....	54
Gambar 4. 23 Form Data Mata Pelajaran .....	55
Gambar 4. 24 Form Data Kelas .....	56
Gambar 4. 25 Form Data Ekstrakurikuler .....	57
Gambar 4. 26 Form Data Wali Kelas .....	58
Gambar 4. 27 Menu Kepala Sekolah.....	59
Gambar 4. 28 Form Laporan .....	60

## DAFTAR TABLE

Table 2. 1 Simbol-Simbol ERD.....	11
Table 2. 2 Komponen <i>Use case diagram</i> .....	13



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Semakin berkembangnya kemajuan ilmu pengetahuan dan teknologi saat ini, kini komputer digunakan di perusahaan, instansi dan sekolah. Digunakan dalam berbagai aktivitas dan pekerjaan. Dengan komputer pemberian layanan dalam berbagai bidang menjadi lebih baik, cepat dan efisien.

Dalam kegiatan pendidikan pun diperlukan adanya suatu sistem komputerisasi dalam berbagai keperluan. Khususnya dalam pengolahan nilai raport di sebuah Sekolah Tingkat Tinggi Pertama atau Sekolah Menengah Pertama (SLTP atau SMP).

Di banyak SLTP atau SMP di Indonesia, komputer telah diperkenalkan dan dipergunakan untuk dan pembelajaran praktek bagi siswa-siswi dan pekerjaan administrasi. Dalam pekerjaan administrasi, komputer berguna untuk memudahkan dalam pelayanan untuk siswa dan wali kelas atau pun orang tua siswa. Dalam pelayanan untuk siswa dan wali kelas atau pun orang tua siswa. Dalam hal ini, pengolahan nilai raport masih menggunakan Microsoft Excel. Upaya dalam pengolahan data dan informasi akan berhasil dengan perubahan yang lebih baik pada sistem yang ada (*Microsot Excel*). Sehingga apabila terjadi kesalahan dan keterlambatan saat perhitungan pengolahan nilai yang banyak menyita waktu dan membutuhkan banyak tenaga dapat diperkecil.

Berdasarkan masalah di atas, penulis ingin membuat sistem aplikasi penilaian siswa khususnya nilai raport agar memberi kemudahan, kecepatan dan ketepatan dalam pengolahan data-data dan nilai siswa oleh guru sehingga mendapatkan informasi dan laporan penilaian siswa secara akurat dan tepat waktu dengan menggunakan bahasa pemrograman C# dan SQL Server 2005 sebagai basis data.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah disebutkan diatas maka dapat disimpulkan rumusan masalah sebagai berikut:

1. Bagaimana analisa dan rancangan sebuah aplikasi penilaian siswa SMP ?

2. Bagaimana implementasi aplikasi penilaian siswa SMP secara terkomputerisasi ?

### 1.3 Tujuan Penulisan

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Menjelaskan analisa dan rancangan aplikasi penilaian siswa SMP.
2. Menjelaskan implementasi Aplikasi Penilaian Siswa SMP berbasis *desktop*.

### 1.4 Batasan Masalah

Agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka diperlukan batasan-batasan terhadap pembahasan makalah ini. Batasan-batasan dalam penelitian ini adalah sebagai berikut:

1. Menganalisa dan merancang sebuah aplikasi penilaiann siswa SMP dengan menggunakan *Entity Relationship Diagram* dan *Use Case Diagram*;
2. Mengimplementasikan aplikasi penilaian siswa SMP dengan menggunakan bahasa pemrograman C# dan SQL Server 2005 sebagai basis data;

### 1.5 Metodologi Penulisan

Metode yang digunakan dalam penulisan makalah ini adalah metode pustaka yaitu metode yang dilakukan dengan mempelajari dan mengumpulkan data dari pustaka yang berhubungan dengan alat, baik berupa buku maupun informasi di internet. Dan juga Metode Eksperimen yaitu melakukan percobaan-percobaan atau penelitian langsung ke tempat perkara yang nantinya diteliti terlebih dahulu, sebelum membuat dan menulis makalah rangkaian ini.

### 1.6 Sistematika Penulisan

Sistematika dalam penulisan makalah ini adalah :

1. BAB I Pendahuluan  
Bab ini menguraikan latar belakang penulisan, tujuan penulisan, batasan masalah, dan sistematika penulisan.

2. BAB II Landasan Teori

Bab ini menjelaskan acuan dari makalah ini dalam menganalisa sistem dan aplikasi penilaian siswa sekolah menengah pertama menggunakan C#, Visual Studio .Net 2010 dan SQL Server 2005.

3. BAB III Analisa Permasalahan

Bab ini menjelaskan mengenai perancangan sistem, mencakup analisa dan pemecahan rumusan masalah.

4. BAB IV Implementasi Sistem

Bab ini menjelaskan mengenai implementasi sistem yang sebelumnya sudah dirancang pada tahap perancangan dan desain.

5. BAB IV Penutup

Bab ini menjelaskan tentang kesimpulan dan saran dari penelitian.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Rapid Application Development***

##### **2.1.1 Definisi RAD**

*Rapid application development* (RAD) atau *rapid prototyping* adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. *Rapid application development* menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) *user* dan selanjutnya disingkirkan. *Working model* digunakan kadang-kadang saja sebagai basis desain dan implementasi sistem final.

##### **2.1.2 Sejarah RAD**

RAD adalah istilah yang awalnya digunakan untuk menggambarkan suatu proses pengembangan *software* yang dikembangkan oleh James Martin pada tahun 1991. Metodologi Martin melibatkan pengembangan iteratif dan pembangunan *prototype*. Baru-baru ini, istilah dan singkatan ini digunakan dalam arti yang lebih luas yang mencakup berbagai teknik yang ditujukan untuk mempercepat pengembangan aplikasi, seperti penggunaan kerangka kerja aplikasi web dan jenis lain dari kerangka kerja perangkat lunak.

Pengembangan aplikasi cepat merupakan tanggapan terhadap proses *non-rapid* yang dikembangkan pada 1970-an dan 1980-an, seperti Analisis dan Desain Sistem Terstruktur dan model *Waterfall* lainnya. Satu masalah dengan metodologi sebelumnya adalah bahwa aplikasi begitu lama untuk dibangun sehingga persyaratan telah berubah sebelum system itu selesai, sehingga *system* yang dibangun tidak memadai atau bahkan tidak dapat digunakan. Masalah lain adalah asumsi bahwa tahap analisis kebutuhan metodologi saja akan mengidentifikasi semua persyaratan penting. Fakta membuktikan bahwa hal ini jarang terjadi, bahkan untuk proyek dengan para profesional yang sangat berpengalaman di semua tingkat.

Dimulai dengan ide-ide dari Brian Gallagher, Alex Balchin, Barry Boehm dan Scott Shultz, James Martin mengembangkan pendekatan *Rapid Application Development* selama tahun 1980 di IBM dan akhirnya diresmikan dengan menerbitkan bukunya pada tahun 1991, *Rapid Application Development*.

## 2.2 C# (C Sharp)

C# (dibaca: C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berdasarkan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar *ECMA-334 C# Language Specification*, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam *keyboard* standar.

### 2.2.1 Sejarah C Sharp

Pada akhir dekade 1990-an, Microsoft membuat program Microsoft Visual J++ sebagai sebuah langkah percobaan untuk menggunakan Java di dalam sistem operasi Windows untuk meningkatkan antarmuka dari Microsoft Component Object Model (COM). Akan tetapi, akibat masalah dengan pemegang hak cipta bahasa pemrograman Java, Sun Microsystems, Microsoft pun menghentikan pengembangan J++, dan beralih untuk membuat pengganti J++, kompilernya dan mesin virtualnya sendiri dengan menggunakan sebuah bahasa pemrograman yang bersifat *general-purpose*. Untuk menangani proyek ini, Microsoft merekrut Anders Helsberg, yang merupakan mantan karyawan Borland yang membuat bahasa Turbo Pascal, dan Borland Delphi, yang juga mendesain Windows Foundation Classes (WFC) yang digunakan di dalam J++. Sebagai hasil dari usaha tersebut, C# pun pertama kali diperkenalkan pada bulan Juli 2000 sebagai sebuah bahasa pemrograman modern berorientasi objek yang menjadi sebuah bahasa

pemrograman utama di dalam pengembangan di dalam *platform* Microsoft .NET Framework.

Pengalaman Helsberg sebelumnya dalam pendesain bahasa pemrograman seperti Visual J++, Delphi, Turbo Pascal) dengan mudah dilihat dalam sintaksis bahasa C#, begitu pula halnya pada inti *Common Language Runtime* (CLR). Dari kutipan atas *interview* dan makalah-makalah teknisnya ia menyebutkan kelemahan-kelemahan yang terdapat pada bahasa pemrograman yang umum digunakan saat ini, misalnya C++, Java, Delphi, ataupun Smalltalk. Kelemahan-kelemahan yang dikemukakannya itu yang menjadi basis CLR sebagai bentukan baru yang menutupi kelemahan-kelemahan tersebut, dan pada akhirnya memengaruhi desain pada bahasa C# itu sendiri. Ada kritik yang menyatakan C# sebagai bahasa yang berbagi akar dari bahasa-bahasa pemrograman lain. Fitur-fitur yang diambilnya dari bahasa C++ dan Java adalah desain berorientasi objek, seperti *garbage collection*, *reflection*, akar kelas (*root class*), dan juga penyederhanaan terhadap pewarisan jamak (*multiple inheritance*). Fitur-fitur tersebut di dalam C# kini telah diaplikasikan terhadap iterasi, properti, kejadian (*event*), *metadata*, dan konversi antara tipe-tipe sederhana dan juga objek.

C# didisain untuk memenuhi kebutuhan akan sintaksis C++ yang lebih ringkas dan *Rapid Application Development* yang 'tanpa batas' (dibandingkan dengan RAD yang 'terbatas' seperti yang terdapat pada Delphi dan Visual Basic).

Agar mampu mempromosikan penggunaan besar-besaran dari bahasa C#, Microsoft, dengan dukungan dari Intel Corporation dan Hewlett-Packard, mencoba mengajukan standardisasi terhadap bahasa C#. Akhirnya, pada bulan Desember 2001, standar pertama pun diterima oleh *European Computer Manufacturers Association* atau *Ecma International* (ECMA), dengan nomor standar ECMA-334. Pada Desember 2002, standar kedua pun diadopsi oleh ECMA, dan tiga bulan kemudian diterima oleh *International Organization for Standardization* (ISO), dengan nomor standar ISO/IEC 23270:2006.

## 2.3 Visual Basic .Net

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk *platform* Microsoft .NET Framework 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform .NET Framework 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk .NET Framework 1.1, dan Visual Studio 2002 ditujukan untuk .NET Framework 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998).

### 2.3.1 .Net Framework

*Microsoft .NET Framework* (dibaca *Microsoft Dot Net Framework*) atau lebih dikenal dengan singkatan dot net merupakan sebuah perangkat lunak kerangka kerja yang berjalan utamanya pada sistem operasi Microsoft Windows, saat ini .NET Framework umumnya telah terintegrasi dalam distribusi standar Windows (mulai dari Windows Server 2003 dan versi-versi Windows yang lebih baru). Kerangka kerja ini menyediakan sejumlah besar pustaka pemrograman komputer dan mendukung beberapa bahasa pemrograman serta

interoperabilitas yang baik sehingga memungkinkan bahasa-bahasa tersebut berfungsi satu dengan lain dalam pengembangan sistem. Berbeda halnya dengan tipikal aplikasi konvensional umumnya, program yang ditulis dengan memanfaatkan .NET Framework berjalan pada lingkungan perangkat lunak melalui *Common Language Runtime*, dan bukan perangkat keras secara langsung. Hal ini memungkinkan aplikasi yang dibuat di atas .NET secara teoritis dapat berjalan pada perangkat keras apapun yang didukung oleh .NET Framework. Perangkat lunak ini adalah kunci penawaran utama dari Microsoft, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk *platform Windows*.

Pada dasarnya, .NET Framework memiliki 2 komponen utama: CLR dan .NET Framework Class Library. Program-program yang ditulis untuk .NET Framework dijalankan pada suatu lingkungan *software* yang mengatur persyaratan-persyaratan *runtime program*. *Runtime environment* ini, yang juga merupakan suatu bagian dari .NET Framework, dikenal sebagai *Common Language Runtime (CLR)*. CLR menyediakan penampilan dari *application virtual machine*, sehingga para programmer tidak perlu mengetahui kemampuan CPU tertentu yang akan menjalankan program. CLR juga menyediakan layanan-layanan penting lainnya seperti jaminan keamanan, pengaturan memori, *garbage collection* dan *exception handling* / penanganan kesalahan pada saat *runtime*. *Class library* dan CLR ini merupakan komponen inti dari .NET Framework. Kerangka kerja itu pun dibuat sedemikian rupa agar para *programmer* dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan juga komputer dari beberapa ancaman keamanan. CLR adalah turunan dari CLI (*Common Language Infrastructure*) yang saat ini merupakan standar ECMA. Pada berbagai literatur dan referensi di Internet, .NET Framework seringkali disingkat menjadi .NET saja.

## 2.4 Basis Data

Basis Data adalah kumpulan data yang tersusun secara sistematis sehingga akan memudahkan pengguna untuk mengakses dan mengatur sehingga akan menghasilkan sebuah informasi yang efektif dan efisien.



*Database* dapat berdiri-sendiri dan juga dapat digabung menjadi satu kesatuan. Menggabung dua atau lebih jenis *database* disebut mengkorelasikan. Sistem manajemen *database* merupakan *software* yang digunakan untuk mengolah *database* sampai pengaturan hak akses.

Sebagai contoh sederhana dari *database* adalah buku telepon yang mungkin sering dilihat. Bagaimana halnya dengan *database* dengan sistem *database* dengan menggunakan komputer? Hal tersebut sama saja seperti *database* yang sifatnya manual (seperti contoh buku telepon di atas) hanya saja dengan adanya komputer maka informasi yang ada di dalam *database* akan sangat mudah untuk di-*update* dan sangat cepat untuk dicari.

*Software* atau aplikasi yang bertugas untuk mengatur, menyimpan, memodifikasi data disebut dengan *software database engine* dan lebih resminya disebut dengan DBMS (*Database Management System*). Ada banyak sekali aplikasi DBMS ini mulai yang berjalan di komputer *personal* (PC) sampai ke komputer skala *mainframe*. Contoh-contoh dari aplikasi *database engine* misalnya seperti:

1. SQL Server, dibuat oleh Microsoft.
2. MS Access, dibuat oleh Microsoft.
3. Oracle *Database* , dibuat oleh Oracle.
4. MySQL, dibuat oleh MySQL AB.
5. Firebird, dibuat oleh komunitas open source berdasarkan dari kode Interbase.
6. PostgreSQL, dibuat oleh komunitas open source.
7. DB2, dibuat oleh IBM.
8. Masih banyak lagi sederet nama *database engine* yang ada di dunia ini.

Basis data adalah himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.

#### **2.4.1 Konsep DBMS**

*Database Management System* adalah sistem pengorganisasian dan pengolahan data pada komputer. Sistem ini dirancang mampu memelihara informasi dan membuat

informasi tersedia saat dibutuhkan dengan beberapa referensi data yang sama. *Database Management System* merupakan kumpulan perangkat lunak yang digunakan untuk menangani semua pengaksesan ke *database*. Mempunyai fasilitas membuat, mengakses, memanipulasi, dan memelihara *database*. Bertujuan untuk efisiensi dan kenyamanan dalam memperoleh menyimpan informasi dalam *database*.

Ketika akan membuat *database* ada dua cara yang digunakan yaitu:

*Top down approach* adalah cara pembuatan DBMS yang melalui proses perancangan terlebih dahulu, seperti pembuatan *entity relationship diagram*.

*Bottom up approach* adalah cara pembuatan DBMS tanpa melalui proses perancangan terlebih dahulu, namun proses perancangan dapat dilakukan jika dibutuhkan. Proses *bottom up approach* biasanya dilakukan dengan cara melakukan normalisasi terlebih dahulu untuk pembuatan *database* tanpa menggunakan *entity relational diagram*.


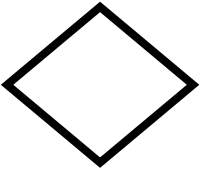
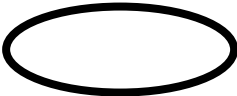

#### 2.4.2 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan relasi data.

#### 2.4.3 Komponen Entity Relationship Diagram

1. Entitas, suatu kumpulan obyek atau sesuatu yang dapat dibedakan atau dapat diidentifikasi secara unik. Kumpulan entitas yang sejenis disebut *entity set*.
2. *Relationship*, hubungan yang terjadi antara satu entitas atau lebih.
3. Atribut, kumpulan elemen data yang membentuk suatu entitas. Jenis-jenis Atribut :
  - a. Key adalah atribut yang digunakan utk menentukan suatu entity secara unik.
  - b. *Atribut simple* adalah atribut yang bernilai tunggal.
  - c. *Atribut Multi Value* adalah atribut yang memiliki sekelompok nilai untuk setiap instan entity.

Table 2. 1 Simbol-Simbol ERD

Simbol	Nama
	<b>Entitas</b> , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	<b>Relasi</b> , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	<b>Atribut</b> , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah).
	<b>Garis</b> , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

#### 2.4.4 SQL Server 2005

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai *driver* JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data *mirroring* dan *clustering*. Pada versi

sebelumnya, MS SQL Server 2000 terserang oleh cacing komputer SQL Slammer yang mengakibatkan kelambatan akses internet pada tanggal 25 Januari 2003.

Perintah SQL terbagi atas 3 (tiga) kelompok yaitu :

1. Bahasa Defenisi Data (*Data Defenition Language*)

DDL adalah hal terpenting dalam pembetulan *database* dengan objek utamanya adalah tabel, indeks, mengubah tabel , menghapus tabel dan membuat domain. Salah satu tipe dasarnya yaitu *create database* yang merupakan perintah untuk membuat *database*.

*File database* yang berisikan informasi tentang data dan objek data base. *File* data lebih lanjut dapat dibagi menjadi dua macam, yaitu *Primary Data file* (*File* data primer), berisikan informasi awal bagi data base dan bernilai bagi *file-file* lain dalam *database*. Dan *Secondary data file* (*file* data sekunder), berisikan data pengguna. *File* ini dapat digunakan untuk memperluas data ke banyak media penyimpanan (disk) dengan meletakan masing-masing *file* pada disk yang berbeda.

2. Bahasa Manipulasi Data (*Data Manipulation Language*)

Hal terpenting DML adalah melakukan operasi atau manipulasi terhadap *database* dengan objek utama *insert*, *delete*, *selection*, *view* dan *update*.

3. Bahasa Kendali Data (*Data Control Language*)

DCL dapat melakukan konfirmasi penyimpanan *database*, menunda perubahan dan menghapus perintah yang diberikan. Pada beberapa *database* juga dapat diatur dengan *database space* dan *user*.

## 2.5 Unified Modeling Language (UML)




*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

### 2.5.1. Use case diagram

*Use case diagram* digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *Use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

*Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *Use case*.

Table 2. 2 Komponen *Use case diagram*

Nama Komponen	Keterangan	Simbol
<i>Use case</i>	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nama <i>Use case</i> dituliskan didalam elips tersebut.	
<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> .	
<i>Association</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>Use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>Actor</i> dengan <i>Use case</i> .	

## **BAB III**

### **ANALISA PERMASALAHAN**

#### **3.1 Gambaran Umum Aplikasi**

Aplikasi Penilaian Siswa SMP, dengan menggunakan aplikasi ini guru dapat melihat data siswa dan aktifitas yang dilakukan siswa di sekolah dan staff dapat mengelola data siswa dan data guru yang ada di sekolah.

Konsep dari aplikasi penilaian siswa ini adalah sebagai berikut :

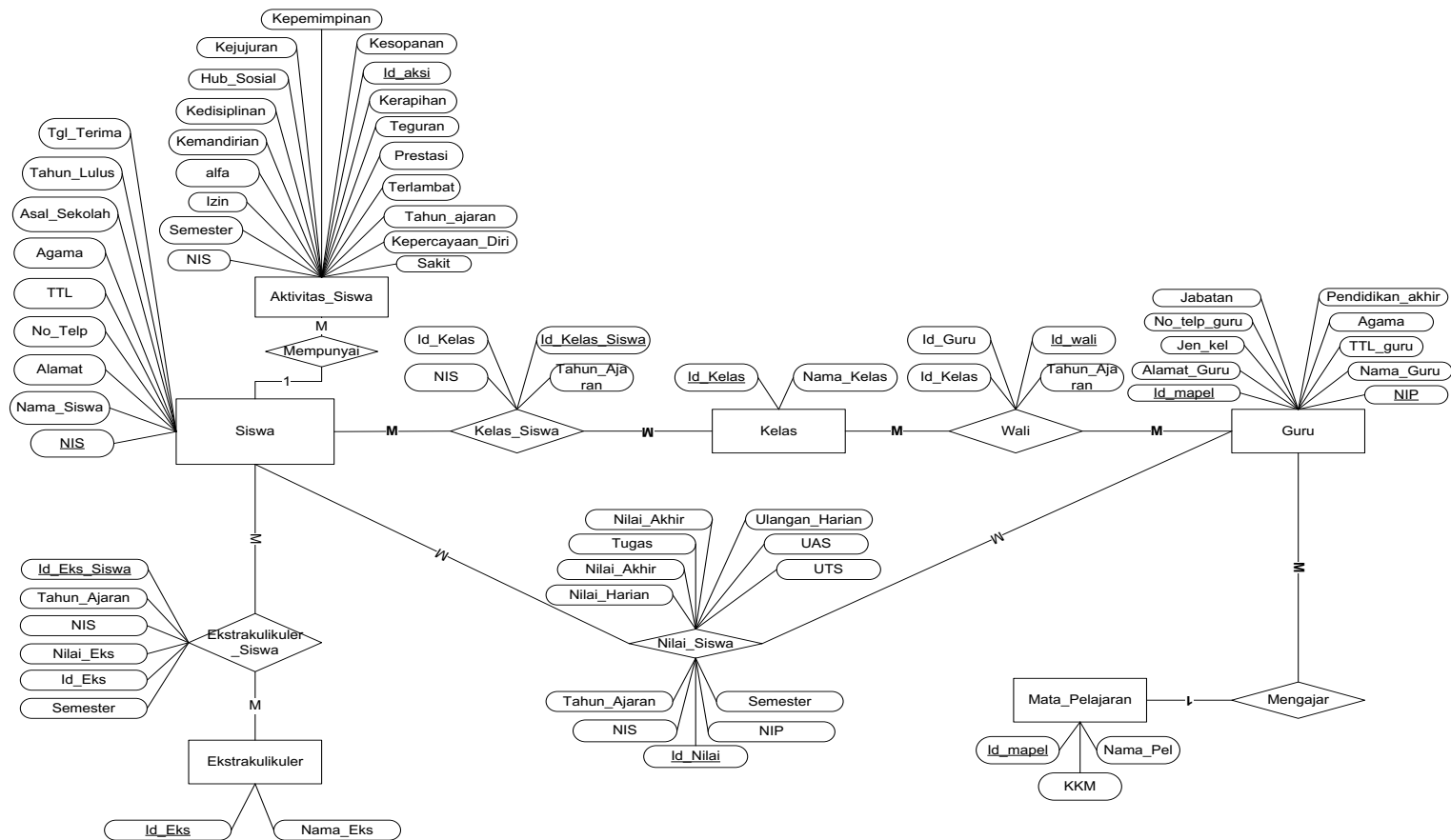
1. Data siswa yang diinputkan oleh staff (tata usaha) dapat terintegrasi secara langsung dengan basis data sehingga data sudah terintegrasi, tertata dan memudahkan penggunaan;
2. Data nilai siswa diinputkan oleh guru dan aplikasi bisa langsung memproses data sehingga data nilai siswa dapat diberikan dengan akurat dan benar;
3. Memiliki banyak fitur untuk memudahkan mengakses data, seperti simpan, hapus, lihat, dan edit data;
4. Memiliki Laporan Penilaian Siswa.

#### **3.2 Analisa Sistem**

Aplikasi penilaian siswa SMP ini dirancang dan dibangun menggunakan bahasa pemrograman C#, menggunakan aplikasi Visual Studio.Net 2010, dan menggunakan Microsoft SQL Server 2005 sebagai basis data. Aplikasi ini berjalan di atas sistem operasi Windows, minimum Windows XP.

### 3.2.1 ER Diagram Aplikasi Penilaian Siswa SMP

Dalam pembangunan Aplikasi Penilaian Siswa SMP ini mengacu pada ER Diagram Aplikasi Penilaian Siswa SMP yang sebelumnya sudah dirancasng terlebih dahulu.



Gambar 3. 1 ER Diagram Aplikasi Penilaian Siswa SMP





### 3.2.3 Desain Menu Utama

Desain ini berguna untuk menampilkan menu yang digunakan *user* sesuai dengan jabatan yang dimiliki.

#### a. Desain Menu Administrator

Desain Menu Administrator digunakan untuk menampilkan menu hanya untuk administrator berupa pengolahan data guru, data akun dan data audit.

Menu Administrator				
File	Data Guru	Akun	Audit	Help
Ganti Password	Masukkan Data Guru	Tambah Data Akun	Lihat Data Audit	How Do I
Logout	Perbaharui Data Guru	Lihat Data Akun	Laporan Data Audit	About
Exit	Lihat Data Guru	Laporan Data Akun		Help
	Laporan Data Guru			

Gambar 3. 3 Desain Menu Administrator

#### b. Desain Menu Tata Usaha

Desain Menu Tata Usaha digunakan untuk menampilkan menu hanya untuk staff tata usaha berupa mengolah data siswa dan data guru.

Menu Tata Usaha		
File	Data Siswa	Help
Ganti Password	Masukkan Data Siswa	How Do I
Logout	Perbaharui Data Siswa	About
Exit	Lihat Data Siswa	Help
	Laporan Data Siswa	

Gambar 3. 4 Desain Menu Tata Usaha

#### c. Desain Menu Bagian Kurikulum

Desain Menu Bagian Kurikulum digunakan untuk menampilkan menu hanya untuk staff bagian kurikulum berupa mengolah data mata pelajaran, data kelas, data ekstrakurikuler dan data walikelas.

Menu Bagian Kurikulum					
File	Mata Pelajaran	Kelas	Ekstrakurikuler	Walikelas	Help
Ganti Password	Olah Mata Pelajaran	Olah Data Kelas	Olah Data Ekstrakurikuler	Olah Data Walikelas	How Do I
Logout	Perbaharui Mata Pelajaran	Perbaharui Data Kelas	Perbaharui Data Ekstrakurikuler	Perbaharui Data Walikelas	About
Exit	Hapus Mata Pelajaran	Hapus Data Kelas	Hapus Data Ekstrakurikuler	Hapus Data Walikelas	Help
	Lihat Mata Pelajaran	Lihat Data Kelas	Lihat Data Ekstrakurikuler	Lihat Data Walikelas	
	Laporan Mata Pelajaran	Laporan Data Kelas	Laporan Data Ekstrakurikuler	Laporan Data Walikelas	

Gambar 3. 5 Desain Menu Bagian Kurikulum

d. Desain Menu Guru

Desain Menu Guru digunakan untuk menampilkan menu hanya untuk guru berupa pengolahan nilai mata pelajaran, nilai ekstrakurikuler, dan aktifitas siswa.

Menu Guru					
File	Nilai Mata Pelajaran	Nilai Ekstrakurikuler	Aktifitas Siswa	Nilai Raport	Help
Ganti Password	Olah Nilai Mata Pelajaran	Olah Nilai Ekstrakurikuler	Olah Aktifitas Siswa	Lihat Nilai Raport	How Do I
Logout	Perbaharui Nilai Mata Pelajaran	Perbaharui Nilai Ekstrakurikuler	Perbaharui Aktifitas Siswa	Laporan Nilai Raport	About
Exit	Hapus Nilai Mata Pelajaran	Hapus Nilai Ekstrakurikuler	Hapus Aktifitas Siswa		Help
	Lihat Nilai Mata Pelajaran	Lihat Nilai Ekstrakurikuler	Lihat Aktifitas Siswa		
	Laporan Nilai Mata Pelajaran	Laporan Nilai Ekstrakurikuler	Laporan Aktifitas Siswa		

Gambar 3. 6 Desain Menu Guru

e. Desain Menu Kepala Sekolah

Desain Menu Kepala Sekolah digunakan untuk menampilkan menu hanya untuk kepala sekolah berupa menu untuk melihat data dan laporan data-data.

Menu Kepala Sekolah			
File	View	Laporan	Help
Ganti Password	Lihat Data Siswa	Laporan Data Siswa	How Do I
Logout	Lihat Data Guru	Laporan Data Guru	About
Exit	Lihat Data Mata Pelajaran	Laporan Data Mata Pelajaran	Help
	Lihat Data Ekstrakurikuler	Laporan Data Ekstrakurikuler	

Gambar 3. 7 Desain Menu Kepala Sekolah

### 3.2.4 Desain Input

Desain ini berguna untuk menginputkan data-data yang diperlukan untuk pengolahan nilai raport. Beberapa desain pendataan tersebut sebagai berikut :

a. Desain Form Input Data Siswa

Form Input Data Siswa digunakan untuk menginputkan data-data siswa yang diperlukan yang terdiri dari identitas siswa dan tanggal masuk siswa ke sekolah baru.

Form Data Siswa

Data Siswa

NIS :

Nama :

Alamat :

Nomor Telepon :

Jenis Kelamin : ☒ Pria ☐ Wanita

Tempat Lahir :

Tanggal Lahir :

Agama :

Asal Sekolah :

Tahun Lulus :

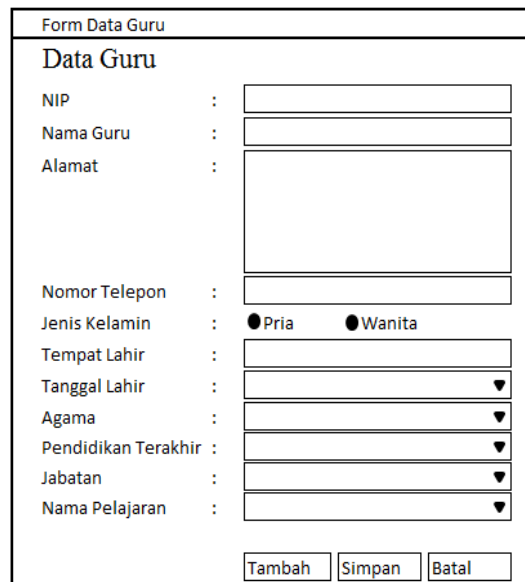
Tanggal Masuk :

Tambah Simpan Batal

Gambar 3. 8 Desain Form Input Data Siswa

b. Desain Input Data Guru

Form Input Data Guru digunakan untuk menginputkan data-data guru yang diperlukan yang terdiri dari identitas guru, pendidikan terakhir dan jabatan di sekolah.



Form Data Guru

**Data Guru**

NIP :

Nama Guru :

Alamat :

Nomor Telepon :

Jenis Kelamin : ☒ Pria ☐ Wanita

Tempat Lahir :

Tanggal Lahir :

Agama :

Pendidikan Terakhir :

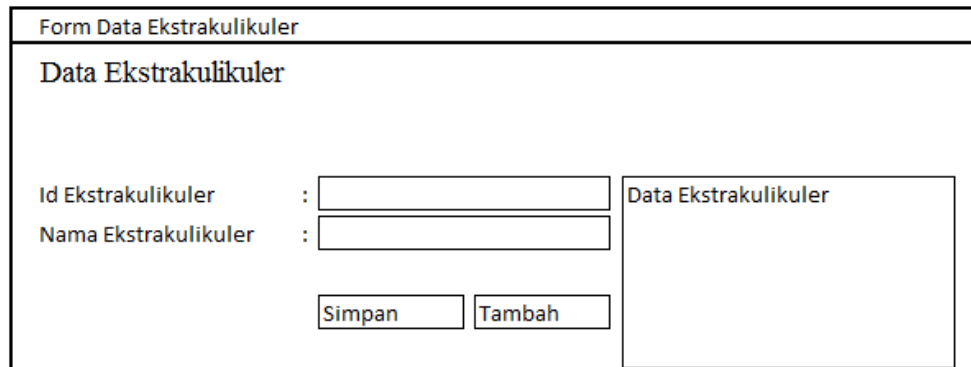
Jabatan :

Nama Pelajaran :

Gambar 3. 9 Desain Form Simpan Data Guru

c. Desain Form Data Ekstrakurikuler

Form Data Ekstrakurikuler digunakan untuk menginputkan dan melihat data ekstrakurikuler yang ada di sekolah. Desain form seperti ini digunakan juga untuk Form Data Kelas, dan Form Data Mata Pelajaran.



Form Data Ekstrakurikuler

**Data Ekstrakurikuler**

Id Ekstrakurikuler :

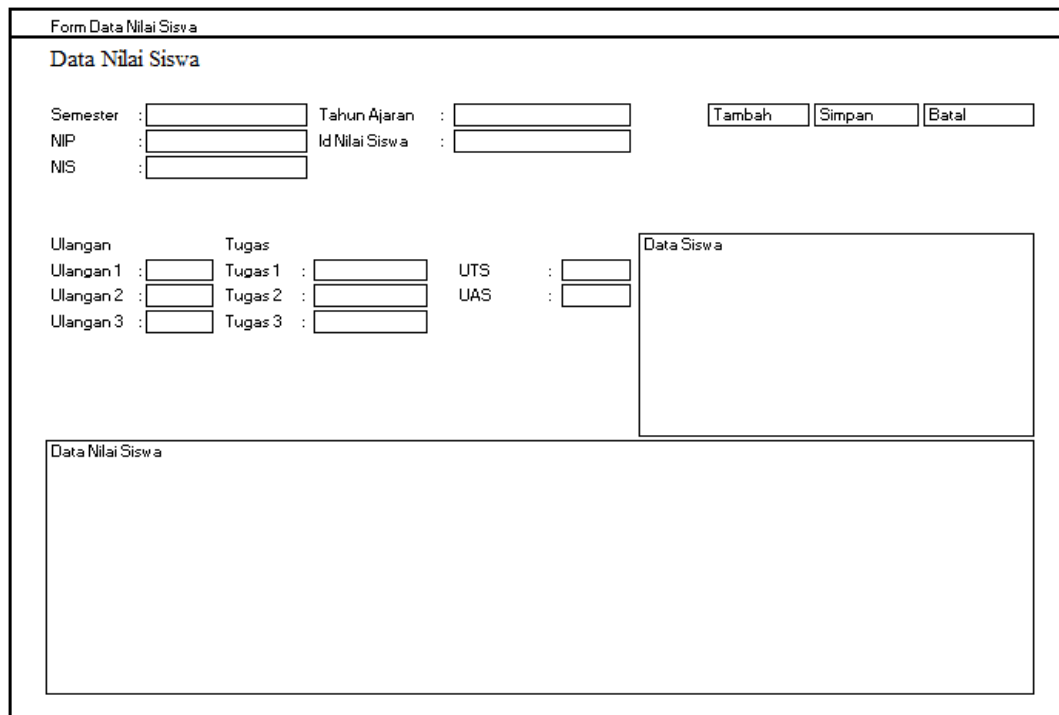
Nama Ekstrakurikuler :

Data Ekstrakurikuler

Gambar 3. 10 Desain Form Data Ekstrakurikuler

d. Desain Input Data Nilai Siswa

Form Nilai Siswa digunakan untuk menginputkan dan melihat nilai ulangan, tugas, ujian tengah semester dan ujian akhir semester siswa dan siswi berdasarkan mata pelajaran.



The form is titled "Form Data Nilai Siswa" and contains the following sections:

- Data Nilai Siswa**: A header section for the main data entry area.
- Input Fields**:
  - Semester :
  - Tahun Ajaran :
  - NIP :
  - Id Nilai Siswa :
  - NIS :
- Buttons**: Three buttons labeled "Tambah", "Simpan", and "Batal" are located to the right of the input fields.
- Grading Section**:
  - Ulangan (Retest) section with three rows:
    - Ulangan 1 :
    - Ulangan 2 :
    - Ulangan 3 :
  - Tugas (Assignment) section with three rows:
    - Tugas 1 :
    - Tugas 2 :
    - Tugas 3 :
  - UTS (Midterm Exam) :
  - UAS (Final Exam) :
- Data Siswa**: A large rectangular box on the right side of the form, intended for student information.
- Data Nilai Siswa**: A large rectangular box at the bottom of the form, intended for the final grade calculation and display.

Gambar 3. 11 Desain Form Nilai Siswa

e. Desain Input Data Kepribadian Siswa

Form Input Data Kepribadian Siswa digunakan untuk memasukan dan melihat data kepribadian berdasarkan Nomor Induk Siswa.

Form Aktivitas Siswa			
<b>Data Kepribadian Siswa</b>			
<b>Id Aktivitas Siswa :</b>	<input type="text"/>	<b>Semester :</b>	<input type="text"/>
<b>NIS :</b>	<input type="text"/>	<b>Tahun Ajaran :</b>	<input type="text"/>
<b>Nama Siswa :</b>	<input type="text"/>		
<b>Aktivitas</b>	<b>Kepribadian</b>		
Izin : <input type="text"/>	Kerapihan : <input type="text"/>	Hubungan Sosial : <input type="text"/>	
Alfa : <input type="text"/>	Kedisiplinan : <input type="text"/>	Kejujuran : <input type="text"/>	
Sakit : <input type="text"/>	Kepercayaan Diri : <input type="text"/>	Kesopanan : <input type="text"/>	
Telat : <input type="text"/>	Kemandirian : <input type="text"/>	Kepemimpinan : <input type="text"/>	
<b>Data Siswa</b>		<b>Data Aktivitas Siswa</b>	
<input type="text"/>		<input type="text"/>	
<input type="button" value="Tambah"/>	<input type="button" value="Simpan"/>	<input type="button" value="Batal"/>	

Gambar 3. 12 Desain Form Aktifitas Siswa

f. Desain Form Input Data Nilai Ekstrakurikuler Siswa

Form Input Data Nilai Ekstrakurikuler Siswa digunakan untuk memasukan nilai Ekstrakurikuler sesuai dengan Ekstrakurikuler yang siswa atau siswi geluti berdasarkan Nomor Induk Siswa.

Form Ekstrakurikuler Siswa			
<b>Ekstrakurikuler Siswa</b>			
Tahun Ajaran	:	<input type="text"/>	Semester
Id Ekstra Siswa	:	<input type="text"/>	Id Ekstra
NIS	:	<input type="text"/>	Nilai
Data Ekstrakurikuler		Data Siswa	
<div></div>		<div></div>	
Data Ekstrakurikuler Siswa			
<div></div>			
Tambah		Simpan	Batal

Gambar 3. 13 Desain Form Ekstrakurikuler Siswa

### 3.2.5 Desain Form Update

Form Update Data Siswa digunakan untuk meng-*update* data siswa jika terjadi kesalahan ataupun terjadi perubahan data siswa berdasarkan Nomor Induk Siswa atau NIS.

Form Update Data Siswa

Data-Data Siswa

Data Siswa

NIS :

Nama :

Alamat :

Nomor Telepon :

Jenis Kelamin :

Tempat Lahir :

Tanggal Lahir :

Agama :

Asal Sekolah :

Tahun Lulus :

Tanggal Masuk :

Gambar 3. 14 Desain Form Update Data Siswa

### 3.2.6 Desain Form Lihat

Form Lihat Data Siswa digunakan untuk melihat atau mencari data siswa berdasarkan Nomor Induk Siswa atau NIS.



Form Lihat Data Siswa

Data-Data Siswa

Cari Refresh

Data Siswa

NIS :  Tempat Lahir :

Nama :  Tanggal Lahir :

Alamat :  Agama :

Asal Sekolah :

Tahun Lulus :

Nomor Telepon :  Tanggal Masuk :

Jenis Kelamin :

Gambar 3. 15 Desain Form Lihat Data Siswa

### 3.2.7 Desain Laporan

Desain Laporan ini digunakan untuk menunjukkan desain laporan yang nantinya akan diaplikasikan ke dalam bentuk *Crystal Report*.

Gambar

Laporan nama\_laporan

Tanggal Print

Nama Kolom Nama Kolom Nama Kolom

Isi Kolom Isi Kolom Isi Kolom

Isi Kolom Isi Kolom Isi Kolom

Halaman

Gambar 3. 16 Desain Laporan

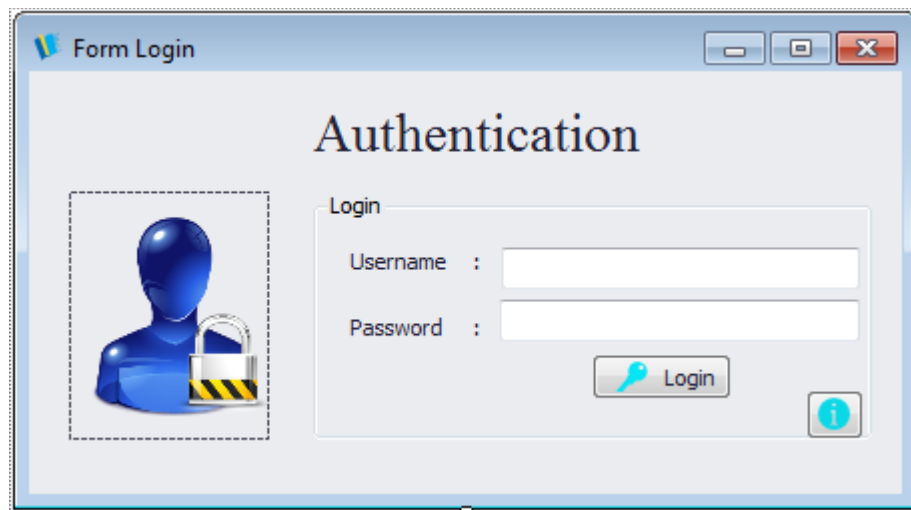
## BAB IV

### IMPLEMENTASI SISTEM

Implementasi program Aplikasi Penilaian Raport Siswa Sekolah Menengah Pertama dalam Bahasa Pemrograman C# dan Microsoft SQL Server 2005 sebagai basis data.

#### 4.1 Form Login

Tampilan Form Login pada aplikasi Penilaian Siswa SMP. Terdapat kolom untuk memasukan username dan password. Jika login berhasil maka user akan masuk ke form selanjutnya sesuai dengan Jabatan.



Gambar 4. 1 Tampilan Form Login

Tombol login pada form login digunakan untuk memproses login. Berikut *code* pada tombol login :

```
public void cekLogin()
{
    String cek = cekText();
    if (cek == "isi")
    {
        String valid = new Class_login().cekLogin(textUsername.Text, textPass.Text);
        if (valid == "blokir")
        {
            MessageBox.Show("Username dan Password Terblokir",
                "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
            kosongtext();
        }
    }
}
```

```

    }
    else
        if (valid == "Tidak Aktif")
        {
            MessageBox.Show("Username dan Password Tidak Aktif",
                "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
            kosongtext();
        }
        else
            if (valid == "Tidak Terdaftar")
            {
                MessageBox.Show("Username dan Password Tidak Terdaftar",
                    "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
                kosongtext();
            }
            else
                if (valid == "gagal")
                {
                    MessageBox.Show("Username dan Password Tidak Valid",
                        "Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    kosongtext();
                }
                else
                {
                    MessageBox.Show("Selamat Datang "
                        +textUsername.Text, "Berhasil Login Sebagai " +valid, MessageBoxButtons.OK,
                        MessageBoxIcon.Information);

                    if (valid == "Administrator")
                    {
                        this.Hide();
                        MenuAdmin admin = new MenuAdmin();
                        admin.Show();
                    }
                    if (valid == "Staff Tata Usaha")
                    {
                        this.Hide();
                        MenuTU tu = new MenuTU();
                        tu.Show();
                    }
                    if (valid == "Guru")
                    {
                        this.Hide();
                        MenuGuru guru = new MenuGuru();
                        guru.Show();
                    }
                    if (valid == "Staff Kurikulum")
                    {
                        this.Hide();
                        MenuBK bk = new MenuBK();
                        bk.Show();
                    }
                    if (valid == "Kepala Sekolah")
                    {

```

```

        this.Hide();
        MenuKepSek ks = new MenuKepSek();
        ks.Show();
    }
}
else
{
    MessageBox.Show(cek);
}
}
}

```

Fungsi tersebut akan memanggil Class\_login() untuk mencari username dan password pada basis data dan akan mencocokkannya dengan username dan password yang diberikan user. Jika username dan password telah ditemukan maka akan di cari level atau jabatan sehingga user akan masuk ke dalam menu aplikasi sesuai dengan jabatan. Berikut merupakan Class\_login() untuk mencari data pada table user :

```

public String cekLogin(String usr, String pwd)
{
    String result = "";
    koneksirad.query = "Select * from tb_user where username = @username";
    koneksirad.param("@username", usr);
    koneksirad.execr();
    if (koneksirad.read())
    {
        String idp = koneksirad.dr["nip"].ToString();
        String user = koneksirad.dr["username"].ToString();
        String pass = koneksirad.dr["password"].ToString();
        String lev = koneksirad.dr["level"].ToString();
        String stat = koneksirad.dr["status"].ToString();
        String itung = koneksirad.dr["coba_pass"].ToString();
        int count = Convert.ToInt32(itung);

        String enc = EnkripMD5(pwd);
        String status = "Tidak Aktif";
        String status1 = "blokir";

        if (status.Equals(stat))
        { //cek apakah akun tidak aktif
            result = "Tidak Aktif";
        }
        else
        {
            if (status1.Equals(stat))
            { //cek apakah akun terblokir
                result = "blokir";
            }
            else
            {
                if (enc.Equals(pass))
                {
                    if (lev.Equals("Administrator"))

```

```

        {
            result = "Administrator";
        }
        if (lev.Equals("Staff Tata Usaha"))
        {
            result = "Staff Tata Usaha";
        }
        if (lev.Equals("Guru"))
        {
            result = "Guru";
        }
        if (lev.Equals("Staff Kurikulum"))
        {
            result = "Staff Kurikulum";
        }
        if (lev.Equals("Kepala Sekolah"))
        {
            result = "Kepala Sekolah";
        }

        //reset salah password
        count = 3;
        salah_password(count, usr);

        auditLogin(idp, user, "Login Ke Sistem");
    }
    else
    {
        result = "gagal";
        //hitung coba pass tersisa
        count--;
        salah_password(count, usr);
        if (count >= 0)
        {
            MessageBox.Show("Maksimal salah password tersisa " +
count + " kali lagi");
        }
    }
}
else
{
    result = "Tidak Terdaftar";
}
return result;
}

```

## 4.2 Form Tambah Akun

Form tambah akun digunakan untuk menambahkan akun sesuai dengan jabatan untuk dapat menjalankan aplikasi sesuai dengan tugas masing-masing.

Gambar 4. 2 Form Tambah Akun

Tombol cari yang ada pada form account digunakan untuk mencari Level Pegawai, Username, dan Password sesuai NIP yang dimiliki staff dan guru yang ada di sekolah. Berikut *code* yang ada dalam tombol cari :

```
koneksirad.query = "select * from tb_guru where nip =" + textCari.Text + "";
koneksirad.execr();
if (koneksirad.read())
{
    textNIP.Text = koneksirad.dr["nip"].ToString();
    textLevel.Text = koneksirad.dr["jabatan"].ToString();
}
else
{
    MessageBox.Show("Data Tidak Ketemu");
}
```

Tombol add atau tambah pada form account digunakan untuk mendapatkan password secara otomatis. Berikut merupakan *code* yang dipakai pada tombol add :

```
public void random()
{
    int r, k;
    string password = "";
    char[] upperCase = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
    int[] numbers = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };
    Random rRandom = new Random();
```

```

        for (int i = 0; i < 5; i++)
        {
            r = rRandom.Next(2);
            if (r == 0)
            {
                k = rRandom.Next(0, 25);
                password += upperCase[k];
            }
            else if (r == 1)
            {
                k = rRandom.Next(0, 9);
                password += numbers[k];
            }
        }
        textPass.Text = password;
    }

```

Tombol simpan pada form account ini digunakan untuk menyimpan data account pada basis data. Berikut *code* pada tombol simpan :

```

String status = "Aktif";
cobapass = 3;
if (cek == "isi")
{
    koneksiрад.query = "insert into tb_user (nip, username, password, level, status,
coba_pass) values (@nip, @user, @pass, @level, @status, @coba_pass)";
    koneksiрад.param("@nip", NIP);
    koneksiрад.param("@user", user);
    koneksiрад.param("@pass", pass);
    koneksiрад.param("@level", level);
    koneksiрад.param("@status", status);
    koneksiрад.param("@coba_pass", cobapass);
    int i = koneksiрад.exec();
    if (i != 0)
    {
        MessageBox.Show("Data Berhasil Disimpan", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("Data Gagal Disimpan", "Data Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
    kosongtext();
}
else
{

```

```

        MessageBox.Show(cek);
    }

```

### 4.3 File

Menu strip File digunakan untuk memberikan fasilitas berupa mengganti password, logout untuk kembali ke menu login dan juga exit untuk menutup seluruh aplikasi. Berikut merupakan form ganti password yang digunakan untuk mengganti password sesuai dengan username.

#### 4.3.1 Form Ganti Password

Form ganti password digunakan untuk mengganti password sesuai dengan username dan pengecekan password lama yang sebelumnya didapatkan otomatis dari sistem.



Gambar 4. 3 Form Ganti Password

Tombol update pada form ganti password ini digunakan untuk meng-update password lama menjadi password baru. Berikut merupakan *code* pada tombol update :

```

if (cek == "isi")
{
    string usr = textUsername.Text;
    String valid = new ClassGantiPass().cekPassword(usr, passLama, passBaru);
    if (valid == "Valid")
    {
        passBaru = new Class_login().EnkripMD5(passBaru);
        String simpan = new ClassGantiPass().updatePassword(usr, passBaru);
    }
}

```



```

        if (simpan == "Password Berhasil Di Update" + "\nPengalihan aplikasi...")
        {
            MessageBox.Show(simpan);
            FormLogin fl = new FormLogin();
            fl.Show();
        }
        else
        {
            MessageBox.Show(simpan);
        }
    }
    else
    {
        MessageBox.Show("Password Lama Salah", "Gagal", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        kosongtext();
    }
}
else
{
    MessageBox.Show(cek);
}
}

```

#### 4.3.2 Logout

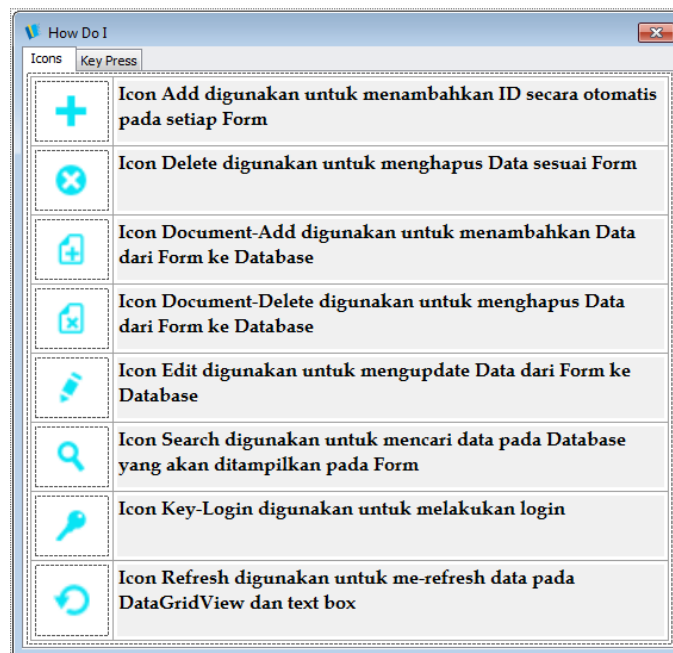
Menu strip logout digunakan untuk keluar dari Main Menu, menutup sesi, dan menampilkan form login.

#### 4.3.3 Exit

Menu strip exit digunakan untuk keluar dari aplikasi, menutup sesi dan tidak memunculkan apapun.

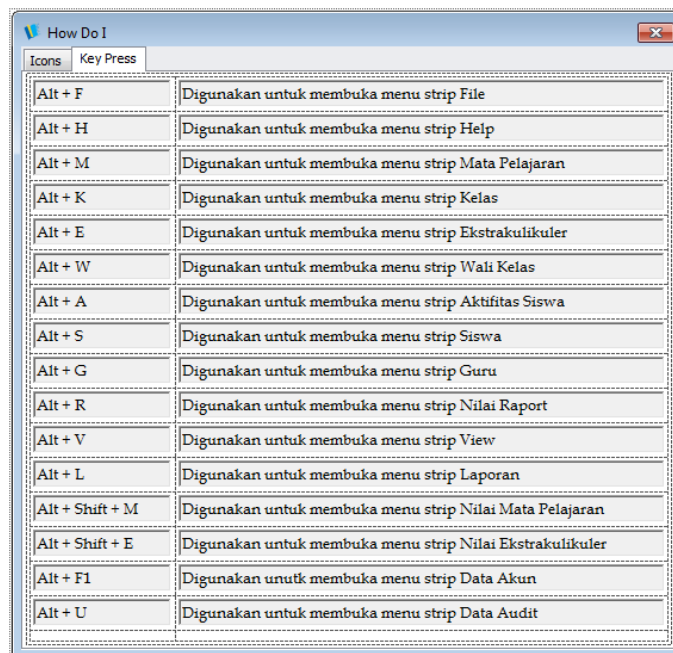
#### 4.4 Help

Menu strip help digunakan untuk menampilkan penjelasan tentang aplikasi Penilaian Siswa Sekolah Menengah Pertama dan penjelasan mengenai beberapa ikon atau gambar, dan fungsi tombol yang ada pada aplikasi tersebut. Tab Icon digunakan untuk menampilkan dan menjelaskan kegunaan dari icon yang digunakan pada aplikasi Penilaian Siswa Sekolah Menengah Pertama



Gambar 4. 4 Form How Do I – Tab Icons

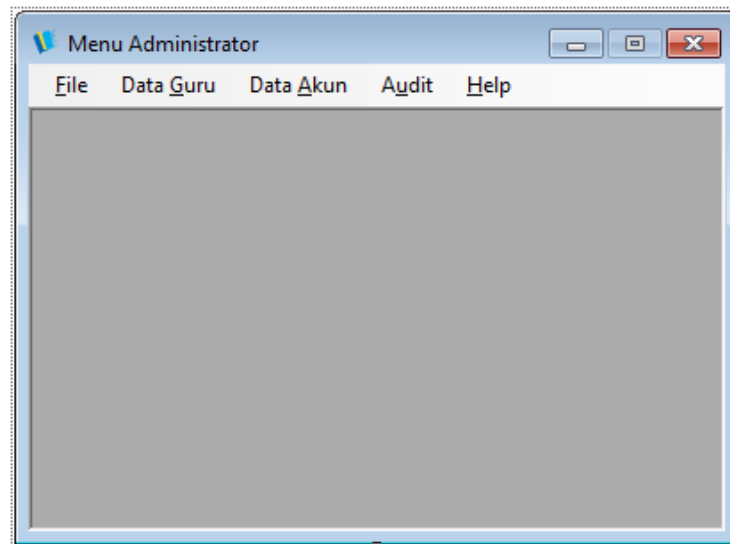
Sedangkan, tab Key Press digunakan untuk menjelaskan kegunaan fungsi tombol keyboard yang terhubung dengan aplikasi.



Gambar 4. 5 Form How Do I – Tab Key Press

## 4.5 Menu Administrator

Menu Administrator ini ditampilkan hanya untuk user dengan jabatan Administrator, fasilitas yang ada pada menu administrator ini adalah mengolah data guru, data akun, dan melihat data audit serta mengolah laporan dari pengolahan data-data tersebut seperti laporan data guru, data akun dan data audit.



Gambar 4. 6 Menu Administrator

### 4.5.1 Form Input Data Guru

Form Input Data Guru digunakan untuk memasukkan data dari form ke basis data. Di fasilitasi dengan tombol Tambah untuk menambahkan ID secara otomatis, tombol Simpan untuk memasukkan data guru dari form ke baris data dan tombol Batal untuk membatalkan penginputan.

*Code untuk tombol simpan*

```
String cek = cekText();
    if (cek == "isi")
    {
        koneksi.query = "insert into tb_guru (nip, nama_guru,
alamat_guru, no_telp, jen_kel_guru, tmp_lhr_guru, tgl_lhr_guru, " +
            "agama_guru, pen_akhir, jabatan, nama_pel) values (@nip,
@nama, @alamat, @notelp, @jenkel, @tl, @tgl, @agama, @penakhir, @jab, @nampel)";
        koneksi.param("@nip", nip);
        koneksi.param("@nama", nama);
        koneksi.param("@alamat", alamat);
        koneksi.param("@notelp", telp);
        koneksi.param("@jenkel", jenkel);
```

```

koneksirad.param("@t1", tmptlhr);
koneksirad.param("@tgl", tgl1hr);
koneksirad.param("@agama", agama);
koneksirad.param("@penakhir", pen_akhir);
koneksirad.param("@jab", jabatan);
koneksirad.param("@nampe1", nama_pel);

int i = koneksirad.exec();
if (i != 0)
{
    MessageBox.Show("Data Berhasil Disimpan", "Information",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    String idp = Class_login.Ses_id_peg;
    String usr = Class_login.Ses_username;
    String act = "Simpan Data Guru " + nip;
    new Class_login().auditLogin(idp, usr, act);
}
else
{
    MessageBox.Show("Data Gagal Disimpan", "Data Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
kosongtext();
}
else
{
    MessageBox.Show(cek);
}

```

Gambar 4. 7 Form Input Data Guru

#### 4.5.2 Form Update Data Guru

Form Update Data Guru digunakan untuk memperbaharui data dari form ke basis data yang sebelumnya sudah dimasukkan ke dalam basis data. Di fasilitasi dengan tombol Cari untuk mencari data yang ada pada basis data, tombol Refresh untuk membuat data form ke keadaan kosong, tombol Update digunakan untuk memperbaharui data yang sebelumnya sudah pernah dimasukkan ke dalam basis data, tombol Batal untuk membatalkan pengisian data pada form.

*Code* untuk tombol Update :

```
String cek = cekText();
    if (cek == "isi")
    {

        koneksiad.query = "update tb_guru set nama_guru = @nama,
alamat_guru = @alamat, no_telp = @notelp," +
                        "jen_kel_guru = @jenkel, tmp_t_lhr_guru =
@tl, tgl_lhr_guru = @tgl, agama_guru = @agama," +
                        "pen_akhir = @penakhir, jabatan = @jab,
nama_pel = @napel where nip = @nip";
        koneksiad.param("@nip", nip);
        koneksiad.param("@nama", nama);
        koneksiad.param("@alamat", alamat);
        koneksiad.param("@notelp", telp);
        koneksiad.param("@jenkel", jenkel);
        koneksiad.param("@tl", tmp_t_lhr);
        koneksiad.param("@tgl", tgl_lhr);
        koneksiad.param("@agama", agama);
        koneksiad.param("@penakhir", pen_akhir);
        koneksiad.param("@jab", jabatan);
        koneksiad.param("@napel", napel);

        DialogResult a = MessageBox.Show("Anda yakin ingin Update
Data?", "Information", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
        if (a == DialogResult.OK)
        {
            int i = koneksiad.exec();
            if (i != 0)
            {
                MessageBox.Show("Data Berhasil Diupdate", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                String idp = Class_login.Ses_id_peg;
                String usr = Class_login.Ses_username;
                String act = "Update Data Guru " +nip;
                new Class_login().auditLogin(idp, usr, act);
            }
        }
    }
}
```

```

    }
    else
    {
        MessageBox.Show("Data Gagal Diupdate", "Data Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    kosongtext();
}
else
{
    kosongtext();
}
}
else
{
    MessageBox.Show(cek);
}

```

The screenshot shows a Windows application window titled "Form Update Data Guru". The window contains a section titled "Data-Data Guru" which includes a search bar with "Cari" and "Refresh" buttons. Below this is a table with the following columns: NIP, Nama, Alamat, No Telp, Jenis Kelamin, and Tempat Lahir. The table is currently empty. Below the table are several input fields for updating a teacher's data: NIP, Nama, Alamat, Nomor Telepon, Jenis Kelamin, Tempat Lahir, Tanggal Lahir (with a calendar icon), Agama, Pendidikan Terakhir, Jabatan, and Nama Pelajaran. At the bottom right of the form are "Update" and "Cancel" buttons.

Gambar 4. 8 Form Update Data Guru

### 4.5.3 Form Lihat Data Guru

Form Lihat Data Guru digunakan untuk melihat data-data guru. Pada form ini user dapat menggunakan fasilitas pencarian untuk mencari spesifikasi data sesuai dengan Nomor Induk Pegawai atau NIP dan tombol Refresh untuk mengembalikan keadaan form ke keadaan semula.

*Code* untuk tombol Cari berdasarkan NIP :

```
koneksirad.query = "select * from tb_guru where nip = @nip";
koneksirad.param("nip", textCari.Text);
koneksirad.execr();

if (koneksirad.read())
{
    textNIP.Text = koneksirad.dr["nip"].ToString();
    textNama.Text = koneksirad.dr["nama_guru"].ToString();
    textAlamat.Text = koneksirad.dr["alamat_guru"].ToString();
    textNoTelp.Text = koneksirad.dr["no_telp"].ToString();
    textJenKel.Text = koneksirad.dr["jen_kel_guru"].ToString();
    textTL.Text = koneksirad.dr["tmpt_lhr_guru"].ToString();
    textTgl.Text = koneksirad.dr["tgl_lhr_guru"].ToString();
    textAgama.Text = koneksirad.dr["agama_guru"].ToString();
    textPenAkhir.Text = koneksirad.dr["pen_akhir"].ToString();
    textJab.Text = koneksirad.dr["jabatan"].ToString();
    textNapel.Text = koneksirad.dr["nama_pel"].ToString();
}
else
{
    MessageBox.Show("Data Tidak Ketemu");
}
```

Gambar 4. 9 Form Lihat Data Guru

#### 4.5.4 Laporan Data Guru

Laporan Data Guru digunakan untuk melihat dan mencetak laporan data guru yang didapatkan dari hasil penginputan data oleh user.

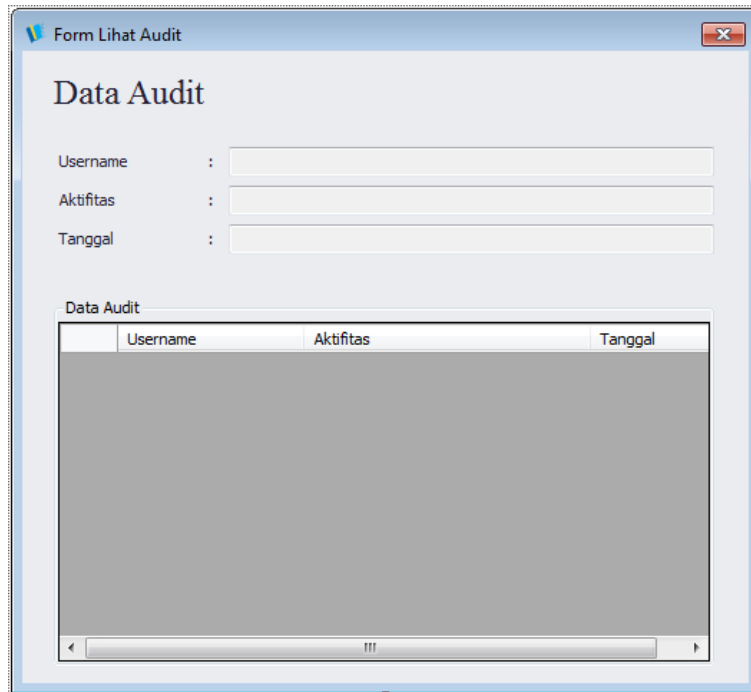
NIP	Nama Guru	Pendidikan Terakhir	Jabatan
NIP001	Superadministrator	Doktor	Administrator
NIP002	Nabila Anundita Humair	Doktor	Staff Tata Usaha
NIP003	Wildatunnisa Fachriah	Doktor	Staff Kurikulum
NIP004	Tari Tri Utami	Sarjana	Kepala Sekolah
NIP005	Nenden Sri Mayangsari	Doktor	Guru

Gambar 4. 10 Laporan Data Guru



#### 4.5.5 Form Lihat Audit

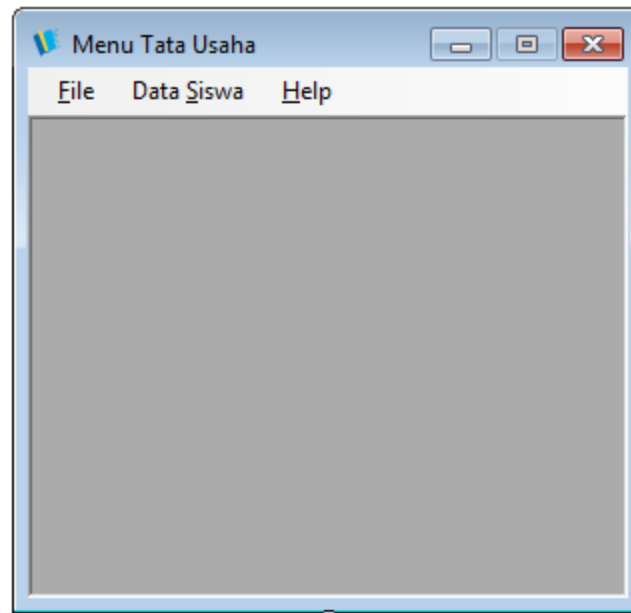
Form lihat audit digunakan untuk melihat hasil audit yang secara otomatis dimasukkan ke dalam basis data untuk mengawasi penginputan data berdasarkan username.



Gambar 4. 11 Form Lihat Audit

#### 4.6 Menu Tata Usaha

Menu Tata Usaha ini ditampilkan hanya untuk user dengan jabatan Staff Tata Usaha, fasilitas yang ada pada menu tata usaha ini adalah mengolah data siswa, mengolah data guru, dan mengolah laporan dari pengolahan data-data seperti laporan data guru dan laporan data siswa.



Gambar 4. 12 Form Menu Tata Usaha

#### 4.6.1 Form Input Data Siswa

Form Input Data Siswa digunakan untuk memasukkan data dari form ke basis data. Di fasilitasi dengan tombol Tambah untuk menambahkan ID secara otomatis, tombol Simpan untuk memasukkan data dari form ke baris data dan tombol Batal untuk membatalkan penginputan.

Gambar 4. 13 Form Data Siswa

Code untuk simpan data siswa

```
String cek = cekText();
    if (cek == "isi")
    {
        koneksi.query = "insert into tb_siswa (nis, nama_siswa,
alamat, notelp, jen_kel, tempat_lahir, tgl_lahir, " +
        "agama, asal_sekolah, tahun_lulus, tgl_masuk) values (@nis,
@nasis, @alsis, @notelp, @jenkel, @tlsis, @tglsis, @agama, @asal, @tahun, @tglmsk)";
        koneksi.param("@nis", nis);
        koneksi.param("@nasis", nasis);
        koneksi.param("@alsis", alsis);
        koneksi.param("@notelp", notelp);
        koneksi.param("@jenkel", jenkel);
        koneksi.param("@tlsis", tlsis);
        koneksi.param("@tglsis", tglsis);
        koneksi.param("@agama", agama);
        koneksi.param("@asal", asal);
        koneksi.param("@tahun", tahunlulus);
        koneksi.param("@tglmsk", tglmsk);
```

```

int i = koneksi.exec();
if (i != 0)
{
    MessageBox.Show("Data Berhasil Disimpan", "Information",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    String idp = Class_login.Ses_id_peg;
    String usr = Class_login.Ses_username;
    String act = "Simpan Data Siswa" + nis;
    new Class_login().auditLogin(idp, usr, act);
}
else
{
    MessageBox.Show("Data Gagal Disimpan", "Data Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
kosongtext();
}
else
{
    MessageBox.Show(cek);
}
}

```

#### 4.6.2 Form Lihat Data Siswa

Form Lihat Data Siswa digunakan untuk melihat data-data siswa. Pada form ini user dapat menggunakan fasilitas pencarian untuk mencari spesifikasi data sesuai dengan Nomor Induk Siswa atau NIS dan tombol refresh untuk mengembalikan keadaan form ke keadaan semula.

The screenshot shows a Windows application window titled "Form Lihat Data Siswa". The window contains a search bar with a "Cari" button and a "Refresh" button. Below the search bar is a table titled "Data Siswa" with columns: NIS, Nama, Alamat, No Telp, Jenis Kelamin, and Tempat Lahir. The table is currently empty. Below the table, there are input fields for NIS, Nama, Alamat, Nomor Telepon, Jenis Kelamin, Tempat Lahir, Tanggal Lahir, Agama, Asal Sekolah, Tahun Lulus, and Tanggal Masuk.

Gambar 4. 14 Form Lihat Data Siswa

### 4.6.3 Form Update Data Siswa

Form Update Data Siswa digunakan untuk mengubah data jika terjadi kesalahan pada saat memasukkan data dari form ke basis data.

Gambar 4. 15 Form Update Siswa

Code untuk perbaharui data siswa

```
String cek = cekText();
if (cek == "isi")
{
    koneksi.koneksi.query = "update tb_siswa set nama_siswa = @nasis, alamat = @alsis,
notelp = @notelp, jen_kel = @jenkel, tempat_lahir = @tllsis, tgl_lahir = @tglsis,
agama = @agama, asal_sekolah = @asal, tahun_lulus = @tahun, tgl_masuk = @tglmsk
where nis = @nis";
    koneksi.koneksi.param("@nis", nis);
    koneksi.koneksi.param("@nasis", nasis);
    koneksi.koneksi.param("@alsis", alsis);
    koneksi.koneksi.param("@notelp", notelp);
    koneksi.koneksi.param("@jenkel", jenkel);
    koneksi.koneksi.param("@tllsis", tllsis);
    koneksi.koneksi.param("@tglsis", tglsis);
    koneksi.koneksi.param("@agama", agama);
    koneksi.koneksi.param("@asal", asal);
    koneksi.koneksi.param("@tahun", tahunlulus);
    koneksi.koneksi.param("@tglmsk", tglmsk);
}
```

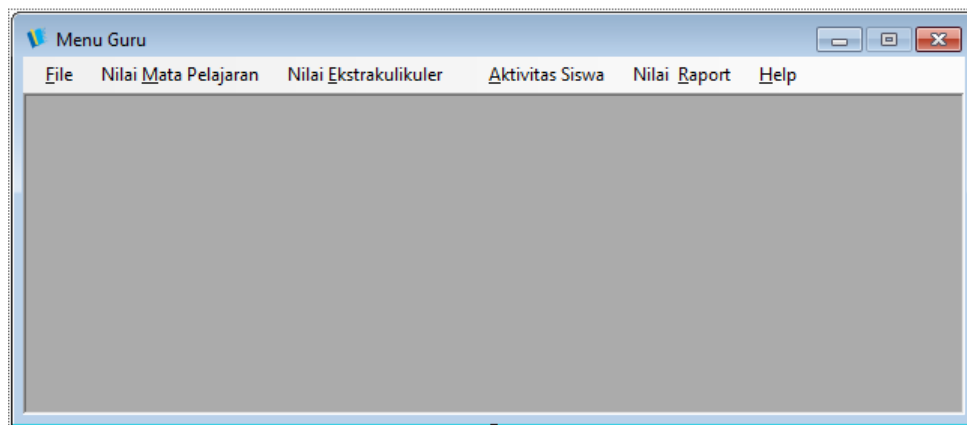
```

DialogResult a = MessageBox.Show("Anda yakin ingin Update Data?", "Information",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
if (a == DialogResult.OK)
{
    int i = koneksi.exec();
    if (i != 0)
    {
        MessageBox.Show("Data Berhasil Diupdate", "Information", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        String idp = Class_login.Ses_id_peg;
        String usr = Class_login.Ses_username;
        String act = "Update Data Siswa " + nis;
        new Class_login().auditLogin(idp, usr, act);
    }
    else
    {
        MessageBox.Show("Data Gagal Diupdate", "Data Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
    kosongtext();
}
else
{
    kosongtext();
}
}
else
{
    MessageBox.Show(cek);
}
}

```

#### 4.7 Menu Guru

Menu Guru ini ditampilkan hanya untuk user dengan jabatan Guru, fasilitas yang ada pada menu guru ini adalah mengolah nilai mata pelajaran, mengolah nilai ekstrakurikuler, mengolah data aktifitas siswa dan mengolah laporan dari pengolahan data-data seperti laporan nilai mata pelajaran, laporan nilai ekstrakurikuler dan laporan aktifitas siswa. Windows State pada Menu Guru diubah ke Maximized agar ketika aplikasi dijalankan Menu Guru akan tampil memenuhi layar.



Gambar 4. 16 Menu Guru

#### 4.7.1 Form Input Nilai Siswa

Form nilai siswa digunakan untuk mengolah nilai siswa sesuai dengan mata pelajaran yang didapat dari sesi login dan nantinya akan mendapatkan Nilai Raport.

**Data Nilai Siswa**

Semester :  Tahun Ajaran :

NIP :  Id Nilai Siswa :

NIS :

Ulangan Tugas

Ulangan 1 :  Tugas 1 :  UTS :  NIS

Ulangan 2 :  Tugas 2 :  UAS :

Ulangan 3 :  Tugas 3 :

**Data Siswa**

NIS	Nama

**Data Nilai Siswa**

Id Nilai Siswa	Semester	Tahun Ajaran	NIP	NIS	Ulangan 1	Ulangan 2	U

Gambar 4. 17 Form Nilai Mata Pelajaran

*Code* untuk simpan nilai mata pelajaran

```

String cek = cekText();
    if (cek == "isi")
    {
        koneksi.query = "insert into tb_nilaisiswa (id_nilaisiswa,
semester, tahun_ajaran, nip, nis, ul_1, ul_2, ul_3, nru, tu_1, tu_2, tu_3, nrt, uts,
uas, nilai_harian, nilai_raport)" +
            "values (@idnisis, @semester, @tajar, @nip, @nis, @u1, @u2,
@u3, @nru, @t1, @t2, @t3, @nrt, @uts, @uas, @nh, @nr)";
        koneksi.param("@idnisis", idnisis);
        koneksi.param("@semester", semester);
        koneksi.param("@tajar", tajar);
        koneksi.param("@nip", nip);
        koneksi.param("@nis", nis);
        koneksi.param("@u1", u1);
        koneksi.param("@u2", u2);
        koneksi.param("@u3", u3);
        koneksi.param("@nru", nru);
        koneksi.param("@t1", t1);
        koneksi.param("@t2", t2);
        koneksi.param("@t3", t3);
        koneksi.param("@nrt", nrt);
        koneksi.param("@uts", uts);
        koneksi.param("@uas", uas);
        koneksi.param("@nh", nh);
        koneksi.param("@nr", nr);

        int i = koneksi.exec();
        if (i != 0)
        {
            MessageBox.Show("Data Berhasil Disimpan", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            String idp = Class_login.Ses_id_peg;
            String usr = Class_login.Ses_username;
            String act = "Simpan Nilai Siswa " + idnisis;
            new Class_login().auditLogin(idp, usr, act);
        }
        else
        {
            MessageBox.Show("Data Gagal Disimpan", "Data Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        kosongtext();
    }
    else
    {
        MessageBox.Show(cek);
    }
}

```



#### 4.7.2 Form Ekstrakurikuler Siswa

Form Ekstrakurikuler Siswa digunakan untuk memberikan nilai ekstrakurikuler sesuai dengan ekstrakurikuler yang dipilih siswa.

Gambar 4. 18 Form Ekstrakurikuler Siswa

*Code* untuk simpan ekstrakurikuler siswa

```
String cek = cekText();
if (cek == "isi")
{
    koneksiрад.query = "insert into tb_ekssiswa (id_ekssiswa,
id_eks, nis, tahun_ajaran, semester, nilai_eks) values (@ideksis, @ideks, @nis,
@tajar, @semester, @nilsis)";
    koneksiрад.param("@ideksis", idEksis);
    koneksiрад.param("@ideks", idEks);
    koneksiрад.param("@nis", NIS);
    koneksiрад.param("@tajar", tajar);
    koneksiрад.param("@semester", semester);
    koneksiрад.param("@nilsis", NilSis);
}
```

```

        int i = koneksi.exec();
        if (i != 0)
        {
            MessageBox.Show("Data Berhasil Disimpan", "Information",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            String idp = Class_login.Ses_id_peg;
            String usr = Class_login.Ses_username;
            String act = "Simpan Data Ekstrakurikuler Siswa" + idEksis;
            new Class_login().auditLogin(idp, usr, act);
        }
        else
        {
            MessageBox.Show("Data Gagal Disimpan", "Data Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        kosongtext();
    }
    else
    {
        MessageBox.Show(cek);
    }
}

```

#### 4.7.3 Form Aktifitas Siswa

Form aktifitas siswa digunakan untuk memberikan penilaian aktifitas siswa di sekolah dan juga kepribadian siswa selama di sekolah atau di kelas.

Gambar 4. 19 Form Aktifitas Siswa

Code untuk simpan aktifitas siswa

```
String cek = cekText();
if (cek == "isi")
{
    koneksi.query = "insert into tb_aksi (id_aksi, nis, semester,
tahun_ajaran, izin, alfa, sakit, telat, kerapihan, kedisiplinan, kepercayaan_diri,
kemandirian," +
        "hubsos, kejujuran, kesopanan, kepemimpinan) values "+
        "(@idaksi, @nis, @semester, @tajar, @izin, @alfa, @sakit,
        @telat, @rapi, @disiplin, @pd, @mandiri, @husos, @jujur, @sopan, @pemimpin)";
    koneksi.param("@idaksi", idaksi);
    koneksi.param("@nis", nis);
    koneksi.param("@semester", semester);
    koneksi.param("@tajar", tajar);
    koneksi.param("@izin", izin);
    koneksi.param("@alfa", alfa);
    koneksi.param("@sakit", sakit);
    koneksi.param("@telat", telat);
    koneksi.param("@rapi", rapi);
    koneksi.param("@disiplin", disiplin);
    koneksi.param("@pd", pd);
```

```

koneksirad.param("@mandiri", mandiri);
koneksirad.param("@husos", husos);
koneksirad.param("@jujur", jujur);
koneksirad.param("@sopan", sopan);
koneksirad.param("@pemimpin", pemimpin);

int i = koneksirad.exec();
if (i != 0)
{
    MessageBox.Show("Data Berhasil Disimpan", "Information",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    String idp = Class_login.Ses_id_peg;
    String usr = Class_login.Ses_username;
    String act = "Simpan Data Aktifitas Siswa " + idaksi;
    new Class_login().auditLogin(idp, usr, act);
}
else
{
    MessageBox.Show("Data Gagal Disimpan", "Data Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
kosongtext();
}
else
{
    MessageBox.Show(cek);
}

```

#### 4.7.4 Nilai Raport

Form Lihat Nilai Raport ini digunakan untuk melihat nilai raport berdasarkan Nomor Induk Siswa dan nilai mata pelajaran yang didapatkan dari pengolahan nilai mata pelajaran oleh guru.

Gambar 4. 20 Form Lihat Nilai Raport

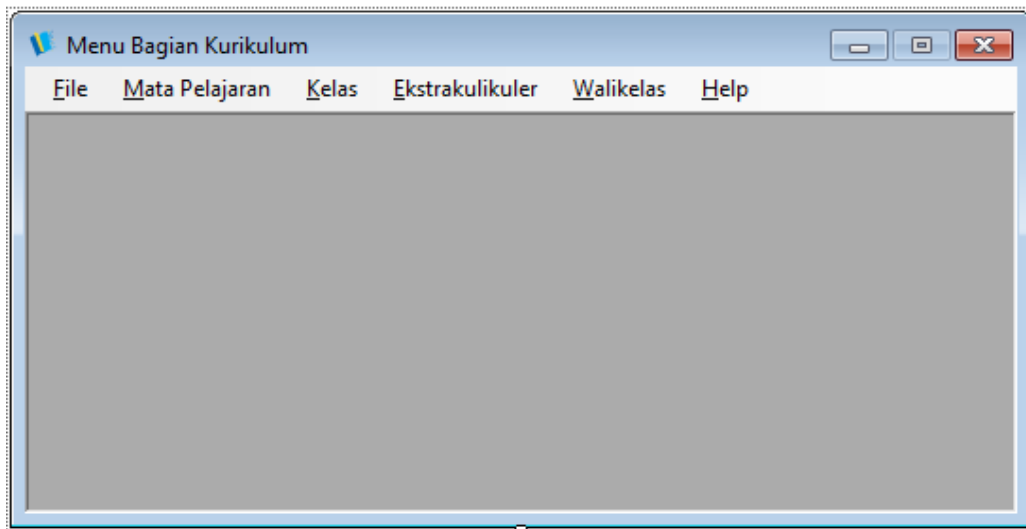
#### 4.7.5 Aktivitas Siswa

Form Lihat Aktivitas Siswa digunakan untuk melihat data-data aktivitas siswa.

Gambar 4. 21 Form Lihat Aktivitas Siswa

#### 4.8 Menu Bagian Kurikulum

Menu Bagian Kurikulum ini ditampilkan hanya untuk user dengan jabatan Staff Bagian Kurikulum, fasilitas yang ada pada menu bagian kurikulum ini adalah mengolah data mata pelajaran, mengolah data kelas, mengolah data ekstrakurikuler, dan mengolah data walikelas. Windows State pada Menu Wali Kelas diubah ke Maximized agar ketika aplikasi dijalankan Menu Wali Kelas akan tampil memenuhi layar.



Gambar 4. 22 Form Menu Bagian Kurikulum

##### 4.8.1 Form Data Mata Pelajaran

Form data mata pelajaran digunakan untuk mengolah data mata pelajaran yang diterapkan dalam kegiatan mengajar oleh guru kepada siswa.

Gambar 4. 23 Form Data Mata Pelajaran

Code untuk simpan mata pelajaran

```
String cek = cekText();
if (cek == "isi")
{
    koneksi.query = "insert into tb_matpel (id_matpel, nama_pel,
kkm) values (@idmatpel, @namapel, @kkm)";
    koneksi.param("@idmatpel", idmatpel);
    koneksi.param("@namapel", namapel);
    koneksi.param("@kkm", kkm);

    int i = koneksi.exec();
    if (i != 0)
    {
        MessageBox.Show("Data Berhasil Disimpan", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        String idp = Class_login.Ses_id_peg;
        String usr = Class_login.Ses_username;
        String act = "Simpan Data Mata Pelajaran " + idmatpel;
        new Class_login().auditLogin(idp, usr, act);
    }
    else
    {
        MessageBox.Show("Data Gagal Disimpan", "Data Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    kosongtext();
}
else
{
    MessageBox.Show(cek);
}
```

#### 4.8.2 Form Data Kelas

Form data kelas digunakan untuk mengolah data kelas atau ruangan yang dipakai sebagai kelas pada sekolah.

Gambar 4. 24 Form Data Kelas

Code untuk simpan data kelas

```
String cek = cekText();
    if (cek == "isi")
    {
        koneksi.query = "insert into tb_kelas (id_kelas, nama_kelas)
values (@idkelas, @namakelas)";
        koneksi.param("@idkelas", idkelas);
        koneksi.param("@namakelas", namakelas);

        int i = koneksi.exec();
        if (i != 0)
        {
            MessageBox.Show("Data Berhasil Disimpan", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            String idp = Class_login.Ses_id_peg;
            String usr = Class_login.Ses_username;
            String act = "Simpan Data Kelas" + idkelas;
            new Class_login().auditLogin(idp, usr, act);
        }
        else
        {
            MessageBox.Show("Data Gagal Disimpan", "Data Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        kosongtext();
    }
    else
    {
        MessageBox.Show(cek);
    }
}
```



### 4.8.3 Form Data Ekstrakulikuler

Form data ekstrakulikuler digunakan untuk mengolah data ekstrakulikuler yang ada di sekolah.

Gambar 4. 25 Form Data Ekstrakulikuler

*Code* untuk simpan data ekstrakulikuler

```
String cek = cekText();
    if (cek == "isi")
    {
        koneksi.query = "insert into tb_ekstra (id_eks, nama_eks)
values (@ideks, @namaeks)";
        koneksi.param("@ideks", ideks);
        koneksi.param("@namaeks", namaeks);

        int i = koneksi.exec();
        if (i != 0)
        {
            MessageBox.Show("Data Berhasil Disimpan", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            idp = Class_login.Ses_id_peg;
            usr = Class_login.Ses_username;
            act = "Simpan Data Ekstrakulikuler " + ideks;
            new Class_login().auditLogin(idp, usr, act);
        }
        else
        {
            MessageBox.Show("Data Gagal Disimpan", "Data Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        kosongtext();
    }
    else
    {
        MessageBox.Show(cek);
    }
}
```

}

#### 4.8.4 Form Data Wali Kelas

Form data wali kelas digunakan untuk mengolah data wali kelas per kelas.

Gambar 4. 26 Form Data Wali Kelas

*Code untuk simpan data wali kelas*

```
String cek = cekText();
    if (cek == "isi")
    {
        koneksi.query = "insert into tb_wali (id_wali, id_kelas,
id_guru, tahun_ajaran) values (@idwali, @idkelas, @nip, @tajar)";
        koneksi.param("@idwali", wali);
        koneksi.param("@idkelas", kelas);
        koneksi.param("@nip", nip);
        koneksi.param("@tajar", tajar);

        int i = koneksi.exec();
        if (i != 0)
        {
            MessageBox.Show("Data Berhasil Disimpan", "Information",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            String idp = Class_login.Ses_id_peg;
```

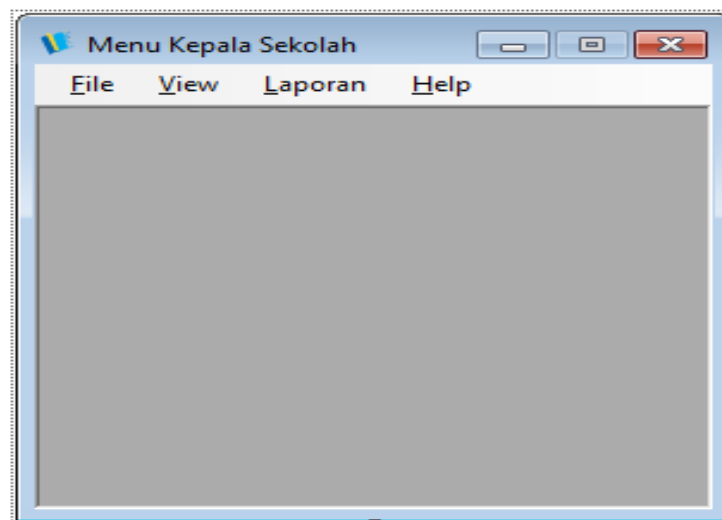
```

        String usr = Class_login.Ses_username;
        String act = "Simpan Data Walikelas " +wali;
        new Class_login().auditLogin(idp, usr, act);
    }
    else
    {
        MessageBox.Show("Data Gagal Disimpan", "Data Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    kosongtext();
}
else
{
    MessageBox.Show(cek);
}
}

```

#### 4.9 Menu Kepala Sekolah

Menu Kepala Sekolah ini ditampilkan hanya untuk user dengan jabatan Kepala Sekolah, fasilitas yang ada pada menu kepala sekolah ini adalah melihat data mata pelajaran, melihat data kelas, melihat data ekstrakurikuler, dan melihat data walikelas serta laporan dari data-data tersebut. Windows State pada Menu Kepala Sekolah diubah ke Maximized agar ketika aplikasi dijalankan Menu Wali Kelas akan tampil memenuhi layar.



Gambar 4. 27 Menu Kepala Sekolah

#### 4.10 Laporan

Form laporan digunakan untuk melihat atau mencetak data secara keseluruhan atau berdasarkan Id tertentu dalam bentuk laporan sehingga didapatkan data dalam bentuk *hard copy*. Berikut contoh form laporan :

**Seasky English**  
Hold The World in Your Hand!

12/26/2013

Semester	Tahun Ajaran	NIP	Nama Pelajaran	NIS	Nilai Raport
2	2009-2010	NIP005	Bahasa Indonesia	NIS001	84.93
2	2012-2013	NIP005	Bahasa Indonesia	NIS002	70.00

Current Page No.: 1      Total Page No.: 1      Zoom Factor: 100%

Gambar 4. 28 Form Laporan

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Aplikasi “Sistem Penilaian Siswa Sekolah Menengah Pertama (SMP)” dibuat dengan menggunakan metode pengembangan *Rapid Application Development* pada Bahasa Pemrograman C# menggunakan Microsoft Visual Studio 2010 untuk memberikan kemudahan pegawai atau guru di sekolah untuk mengolah data-data, seperti data siswa, data pelajaran, data kelas, dan lain sebagainya sehingga diharapkan dapat mempercepat, mempermudah, mengefisien dan efektifkan proses pengolahan data di sekolah. Aplikasi ini didukung dengan fitur-fitur pembantu seperti penginputan data, perbaharuan dan penghapusan data, serta melihat dan mencetak data dalam bentuk laporan.

#### **5.2 Saran**

1. Dibutuhkan konsep yang matang dalam merancang dan membangun aplikasi Sistem Penilaian Siswa Sekolah Menengah Pertama ini sehingga saat pengimplementasian kedalam program dapat dijalankan sesuai rencana dan fungsi masing-masing fitur.
2. Dibutuhkan pelatihan dalam pengimplementasian dan penerapan aplikasi Sistem Penilaian Siswa Sekolah Menengah Pertama ini sehingga para user tidak kebingungan dalam menjalankan aplikasi ini.

## DAFTAR PUSTAKA

- <http://rapidapplicationdevelopmentrad.blogspot.com/> diunggah Tanggal 14-9-2013
- <http://yuliagroups.wordpress.com/pengertian-prototype/> diunggah Tanggal 14-9-2013
- <http://3ka-09.digimon.tv/t9-pengertian-visual-basic> diunggah Tanggal 14-9-2013
- <http://lehaqueenzy.wordpress.com/about/> diunggah Tanggal 14-9-2013
- <http://ceritaaufklarung.wordpress.com/2012/10/17/pengertian-c-net/> diunggah Tanggal 14-9-2013
- Rapid Application Development with Sharia Compliance didokumentasi Tanggal diunggah Tanggal 14-9-2013