

BAB 2

LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai landasan teori yang digunakan sebagai acuan dalam analisis dan pengembangan perangkat lunak manajemen kinerja karyawan berbasis web. Pembahasannya meliputi pengertian sistem informasi, aplikasi berbasis web, manajemen kinerja karyawan, dan segala sesuatu yang berhubungan dengan pengimplementasian program tersebut.

2.1. Tinjauan Pustaka

Perangkat lunak yang telah dibuat pada saat ini adalah "Pembangunan Perangkat Lunak Pelaporan Hasil Belajar Siswa Berbasis Web" yang dibuat oleh Elisabeth Thea Melianta, 2008. Perangkat lunak tersebut dibuat untuk membantu melakukan penghitungan nilai akhir dan pengisian raport tingkat SMA, dan dibangun dengan menggunakan Java dan MySQL. Aplikasi ini telah berhasil membantu guru dalam penghitungan nilai akhir dengan komponen nilai tugas, ulangan, uts, dan uas dengan bobot nilai untuk masing-masing komponen yang telah ditentukan sebelumnya oleh setiap guru yang bersangkutan dan membantu wali kelas dalam pembuatan data raport lengkap untuk setiap mata pelajaran, nilai kompetensi, pengembangan diri (kegiatan ekstrakurikuler), absensi, dan kepribadian.

Selain itu terdapat "Pembangunan Web Komunitas Band Indie Jogja Berbasis Teknologi AJAX (JogIn)" yang dibuat oleh I Wayan Handika, 2008. Perangkat lunak tersebut

dibuat untuk membantu komunitas band di Jogja yang bergerak pada jalur indie label. JogIn dibangun dengan menggunakan bahasa pemrograman ASP.NET, C# sebagai *code behind*, ASP.NET AJAX Extensions 1.0 sebagai framework AJAX, dan SQL Server 2005 sebagai basis datanya. Aplikasi yang dibuat telah berhasil mengimplementasikan teknologi AJAX untuk membuat sebuah aplikasi web komunitas band indie Jogja.

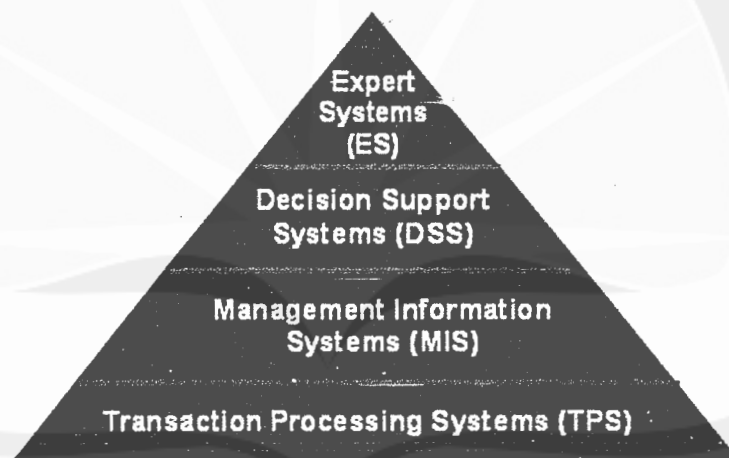
2.2. Sistem Informasi

Sistem bisa dikatakan sebagai kumpulan dari obyek atau elemen yang berinteraksi satu sama lain untuk mencapai tujuan tertentu (Mudjihartono, 1998). Sedangkan kata informasi didefinisikan sebagai data dalam konteks sedang digunakan atau dimanfaatkan atau data yang sudah matang dan bermanfaat bagi pengguna (Mudjihartono, 1998). Data terdiri dari kenyataan dan gambar yang secara relatif tak mempunyai arti bagi pengguna. Jika data ini diproses, maka data ini dapat diubah menjadi informasi.

Informasi dapat diperoleh dari sistem informasi yang disebut juga dengan *processing systems* atau *information processing systems*. Sistem informasi dapat didefinisikan sebagai berikut : Suatu sistem informasi adalah suatu kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar untuk mengambil keputusan yang cerdas.

Sistem informasi terdiri dari beberapa komponen yang saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai suatu tujuan tertentu. Komponen-komponen dari sistem informasi dapat disebut dengan istilah blok bangunan (*building block*), yaitu : blok masukan/*input*, blok proses/*model*, blok keluaran/*ouput*, blok basis data, blok kontrol/kendali, dan blok teknologi.

Sistem informasi dikembangkan untuk tujuan yang berbeda-beda, tergantung pada kebutuhan bisnis. Sistem Informasi kategorial berdasar tingkat kegunaan (Mudjihartono, 1998) dapat ditunjukkan oleh gambar 2.1 :



Gambar 2.1 Sistem Informasi Kategorial Berdasar Tingkat Kegunaan

1. *Transaction Processing Systems / Data Processing Systems (TPS/DPS)*: mendukung kegiatan organisasi sehari-hari dan memelihara sebagian besar data internal organisasi, memproses sejumlah besar data untuk transaksi bisnis rutin. Terdapat data entry ke sistem dan laporan transaksi dengan sedikit membutuhkan keputusan.

2. *Management Information Systems (MIS)*: menyusun laporan secara periodik untuk kontrol, perencanaan dan membuat keputusan. Laporan bersifat umum bagi semua bagian perusahaan.
3. *Decision Support Systems (DSS)*: mendukung pengambil keputusan dengan menyusun informasi pada kebutuhan tertentu. Program yang dibuat khusus untuk orang tertentu atau *group* tertentu pengambil keputusan.
4. *Expert Systems (ES)* dan *Artificial Intellegente (AI)*: menangkap keahlian tiruan dari pembuat keputusan dalam menyelesaikan masalah. *Expert Systems* menggunakan pendekatan AI untuk menyelesaikan masalah secara beralasan sesuai pemikiran manusia.

Dalam hal ini, perangkat lunak yang akan dibangun oleh penulis merupakan *Transaction Processing Systems*.

2.3. Aplikasi Berbasis Web

Dalam rekayasa perangkat lunak, suatu aplikasi web (bahasa Inggris: *web application* atau sering disebut *webapp*) adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti internet atau intranet. Sehingga hanya dengan memiliki akses internet kita telah dapat menggunakan aplikasi berbasis web tersebut.

2.3.1. Pengantar Aplikasi Berbasis Web

Saat ini web tidak sekedar menjadi media untuk menyajikan informasi, tetapi telah menjadi platform aplikasi. Kemudahan dalam penggunaan *browser* web untuk merambah informasi yang ada di internet, telah menginspirasi pemrogram aplikasi untuk menggunakan media

browser ini sebagai platform aplikasi. Aplikasi tidak lagi menjadi rumit untuk dideploy, diimplementasi, dan dioperasionalkan. Tidak diperlukan pemasangan aplikasi pada setiap komputer. Setiap pengguna komputer cukup memiliki akses ke server aplikasi, maka pengguna akan dapat melakukan akses dan menggunakan aplikasi.

2.3.2. Arsitektur Client-Server

Client-server merupakan sebuah paradigma dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi ke dalam dua pihak, yaitu pihak client dan pihak server.

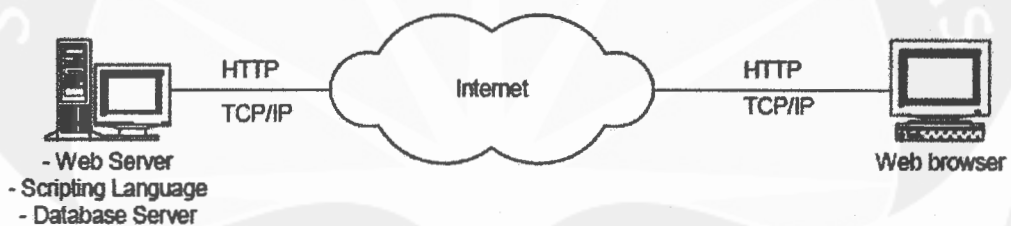
Dalam model client-server, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, tetapi masih merupakan sebuah kesatuan yakni komponen client dan komponen server. Komponen client juga sering disebut sebagai *front-end*, sementara komponen server disebut sebagai *back-end*. Komponen client dari aplikasi dijalankan dalam sebuah *workstation* dan menerima masukan data dari pengguna. Komponen client tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan teknologi pemrosesan tertentu dan mengirimkannya kepada komponen server yang dijalankan di atas mesin server, umumnya dalam bentuk *request* terhadap beberapa layanan yang dimiliki oleh server. Komponen server akan menerima request dari client, kemudian memprosesnya dan mengembalikan hasil pemrosesan tersebut kepada client.

Sebuah contoh dari aplikasi client-server sederhana adalah aplikasi web yang didesain dengan menggunakan Active Server Pages (ASP) atau PHP. Script PHP atau ASP akan dijalankan di dalam web server (Apache atau Internet

Information Services), sementara script yang berjalan di pihak client akan dijalankan oleh web browser pada komputer client. Client-server merupakan penyelesaian masalah pada software yang menggunakan database sehingga setiap komputer tidak perlu diinstall database, dengan metode client-server database dapat diinstall pada suatu komputer sebagai server dan aplikasinya diinstall pada client.

2.3.3. Infrastruktur Aplikasi Berbasis Web

Aplikasi berbasis web membutuhkan infrastruktur seperti ditunjukkan oleh gambar 2.2 (Hartono, 2004) :



Gambar 2.2 Infrastruktur Aplikasi Berbasis Web

1. Web server

Web server merupakan servis utama dalam sebuah aplikasi berbasis web. Web server digunakan untuk mempublish dokumen-dokumen sehingga client dapat mengaksesnya melalui protokol HTTP. Contoh web server seperti Apache, IIS, Tomcat dan lain sebagainya.

2. Scripting Language

Scripting language digunakan sebagai bahasa pemrograman dalam membangun sebuah aplikasi berbasis web. Terdapat banyak pilihan bahasa pemrograman dalam web, antara lain : HTML, XML, Perl, PHP, ASP, Java, dll.

3. Database server

Database server digunakan untuk mendukung sebuah aplikasi berbasis web yang membutuhkan proses penyimpanan secara permanen. Contoh database server seperti MySQL, MS SQL server, PostGre SQL, Oracle dan lain sebagainya.

4. TCP/IP

TCP/IP digunakan sebagai protokol standar dalam transmisi paket-paket data.

5. HTTP (*Hypertext Transfer Protocol*)

HTTP merupakan protokol yang digunakan sebagai layanan web.

6. *Web browser*

Web broser digunakan untuk merender teks-teks dalam format HTML sehingga menjadi tampilan sesuai desain pengembangnya. Beberapa jenis *web browser* seperti Internet Explorer, Opera, Mozilla, Netscape dan lain sebagainya.

2.4. Visual Web Developer 2008 Express SP1

Visual Web Developer Express adalah suatu *freeware* tool pengembangan web untuk membuat ASP.NET websites. Active Server Pages .NET (disingkat sebagai ASP.NET) dapat didefinisikan sebagai platform teknologi yang didesain untuk mengembangkan aplikasi berbasis web. Saat ini .Net Framework sudah sampai versi 3.5 yang sudah mendukung Vista untuk versi .Net Framework 3.0 ke atas dan mendukung ASP.NET AJAX yang membuat pengembangan web lebih efisien dan interaktif. .Net Framework merupakan komponen penghubung antara aplikasi dengan lingkungan Windows yang kita miliki, .Net Framework bersifat AddOn atau komponen tambahan.

Terdapat fleksibilitas pada Visual Web Developer 2008 karena mendukung *multi-targeting project*, yakni dapat memilih *framework* mana yang akan digunakan, baik ASP.NET 2.0, ASP.NET 3.0, maupun ASP.NET 3.5. Visual Web Developer 2008 Express SP1 mendukung baik class library maupun web application project, yang tidak didukung di VS2005 Express. Pada Visual Web Developer 2008 juga terdapat peningkatan pada *visual designer* (mode Split View), fitur IntelliSense dan *debugging* pada JavaScript, serta integrasi AJAX Extensions yang sudah *built-in* pada ASP.NET 3.5 tanpa perlu *men-download* secara terpisah.

Versi 2008 ini menyediakan fitur-fitur yang dibangun di atas teknologi terakhir yang mencakup AJAX, CSS, LINQ dan Javascript IntelliSense dan juga menyediakan HTML dan CSS *editing tools*.

2.5. AJAX

AJAX adalah kependekan dari *Asynchronous JavaScript and XML*. AJAX merupakan suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web interaktif. Dengan menggunakan AJAX, JavaScript yang ada pada suatu halaman web dapat berkomunikasi langsung dengan *server* dengan menggunakan objek XMLHttpRequest. Dengan objek ini, JavaScript dapat mengambil dan mengirimkan data ke *server* tanpa perlu melakukan *reload* atau *refresh* pada halaman web. Hal ini disebabkan AJAX melakukan transfer data (*request Http*) secara *asynchronous* antara browser dengan web server sehingga memungkinkan untuk mengambil sebagian data dari server tanpa perlu mengambil semua isi halaman web tersebut. Dengan demikian, aplikasi web menjadi lebih cepat, lebih interaktif, dan lebih *user-friendly*.

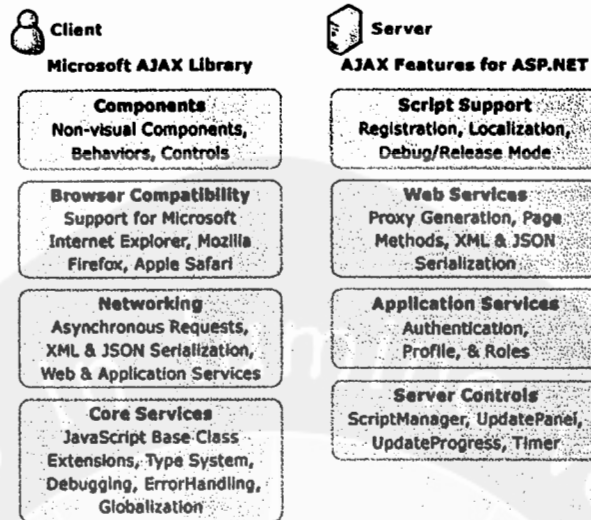
2.5.1. ASP.NET AJAX

ASP.NET AJAX merupakan nama resmi framework AJAX dari Microsoft yang sebelumnya memiliki nama ATLAS. ASP.NET AJAX merupakan framework yang terdiri dari pustaka *script client* dan komponen *server* yang terintegrasi. ASP.NET AJAX juga didukung dengan ASP.NET AJAX Control Toolkit dan beberapa fitur tambahan lainnya (Masykur, 2007).

Beberapa keunggulan aplikasi web menggunakan ASP.NET AJAX dibandingkan dengan aplikasi web yang secara penuh berbasis server antara lain (Masykur, 2007) :

1. Lebih efisien karena sebagian proses dilakukan di browser.
2. Elemen UI yang lebih familiar seperti indikator proses, tooltips dan jendela pop-up.
3. Update sebagian halaman yang hanya mengganti sebagian dari halaman web (*partial rendering*).
4. Framework yang mudah disesuaikan dengan kebutuhan sebagaimana *server-control*.
5. Mendukung sebagian besar browser terkenal termasuk Microsoft Internet Explorer, Mozilla Firefox dan Apple Safari.

Arsitektur ASP.NET AJAX dibagi menjadi dua bagian yakni arsitektur pada client dan arsitektur pada server seperti ditunjukkan oleh gambar 2.3 berikut :



Gambar 2.3 Arsitektur Server dan Client ASP.NET AJAX

Arsitektur Server ASP.NET AJAX

Komponen server ASP.NET AJAX terdiri dari kontrol-kontrol web server ASP.NET dan komponen-komponen untuk mengatur UI dan alur aplikasi, juga mengatur *serialization*, validasi, pengembangan *control server*, dan lain sebagainya.

Kontrol-kontrol Server ASP.NET AJAX

Berikut adalah kontrol-kontrol server pada ASP.NET AJAX :

1. ScriptManager

Berfungsi untuk mengatur sumberdaya script komponen-komponen client, merender bagian dari halaman, localization, globalization, dan custom user script. Script manager dibutuhkan oleh UpdatePanel, UpdateProgress dan kontrol Timer.

2. UpdatePanel

Digunakan untuk membuat rendering sebagian halaman (*partial rendering*), bukan keseluruhan halaman. Hanya

bagian yang ada di dalam UpdatePanel yang akan di-reload oleh web server.

3. UpdateProgress

Berfungsi untuk menyediakan informasi status bagian dari halaman web yang sedang di-update dalam suatu kontrol UpdatePanel.

4. Timer

Berfungsi untuk melakukan postback ke server berdasarkan interval waktu yang telah ditentukan. Kontrol Timer dapat digunakan untuk melakukan postback keseluruhan halaman atau sebagian halaman yang ada dalam UpdatePanel.

Web Services ASP.NET AJAX

ASP.NET AJAX menyediakan *web services* yang dapat digunakan oleh script client dan dapat bekerja bersama dengan *application services* ASP.NET. Dengan menggunakan ASP.NET AJAX, pemanggilan web service menggunakan script client dapat dilakukan dengan mudah karena telah disediakan komponen network di dalamnya.

Arsitektur Client ASP.NET AJAX

Pustaka script client ASP.NET AJAX terdapat pada beberapa file (.js). Beberapa layer yang ada pada pustaka ASP.NET AJAX antara lain :

1. Layer *browser compatibility*. Layer ini menyediakan kompatibilitas sebagian besar browser populer.
2. Service inti ASP.NET AJAX, yang mengandung komponen-komponen inti seperti *classes*, *namespaces*, *event handling*, *inheritance*, tipe data, dan serialisasi *object*.

3. Pustaka class ASP.NET AJAX, berisi komponen-komponen seperti *string builder* dan penanganan *error* tingkat lanjut.
4. Layer *network* yang menangani komunikasi antara *service client web* dengan aplikasi dan mengatur pemanggilan *method* secara asinkron.
5. Dukungan pustaka Javascript yang terkandung dalam *assembly* atau berada dalam file (.js). Penyatuan script Javascript ke dalam *assembly* dapat mempermudah pemasangan aplikasi (*deployment*) dan memecahkan masalah *versioning*.
6. Dukungan atas pengaksesan *server-based forms authentication* dan informasi *profile* dalam *script client*.
7. Dukungan atas mode *release* dan *debug* serta mendukung *localization* yang tertanam dalam *assembly* maupun terpisah dalam file JavaScript.

Debugging

Arsitektur ASP.NET menyediakan model untuk mode *release* dan *debug*. Pada mode *release*, pemeriksaan *error* dan penanganan *exception* telah dioptimasi untuk *performance* dengan script yang minimum. Pada mode *debug*, menyediakan lebih banyak fitur seperti pengecekan tipe dan argument.

Dalam class *debug helper* (Sys.Debug) tersedia *method-method* untuk menampilkan objek-objek ke dalam form yang mudah dibaca pada bagian bawah halaman web. *Helper* ini juga menampilkan pesan *trace* untuk melakukan *assertions* dan *break* ke debugger.

Localization dan Globalization

Arsitektur ASP.NET AJAX server dan client tersedia mekanisme model *localization* dan *globalization* pada script client. Dengan fitur ini, kita dapat merancang aplikasi dengan berbagai macam *locale* UI (bahasa dan kultur) menggunakan satu basis kode. Sebagai contoh, untuk menampilkan tanggal atau angka, ASP.NET AJAX akan memformat tampilan sesuai dengan *setting culture* pada browser pengguna tanpa harus melakukan *postback* ke server.

2.5.2. AJAX Control Toolkit

AJAX Control Toolkit merupakan sekumpulan komponen gratis (dan *open source*) yang dikeluarkan secara resmi oleh Microsoft untuk membantu para *programmer* web dalam melakukan pemrograman berbasis web dengan menggunakan teknik AJAX.

Toolkit versi 3.5.20820 hanya untuk pengembang yang membangun aplikasi di atas .NET Framework 3.5 SP1 menggunakan Visual Studio 2008 SP1. Untuk .NET Framework 2.0 dan Visual Studio 2005 menggunakan Toolkit versi 1.0.20229.

2.6. DXperience v8.3 (versi 2008 volume 3)

DXperience adalah kumpulan komponen, *library*, dan IDE tools (CodeRush and Refactor!) dalam satu *package* yang dibuat oleh Developer Express. DXperience menyediakan kontrol baik untuk aplikasi desktop maupun aplikasi berbasis web. DXperience terdiri dari empat edisi yakni DXperience Universal, DXperience Enterprise, DXperience WinForms, dan DXperience ASP.NET. DXperience

Enterprise dan Universal meliputi semua produk dari Developer Express untuk Visual Studio dan .NET Framework dan menyertakan *source code* dari semua komponen Windows Forms dan ASP.NET. DXperience WinForms meliputi semua komponen untuk .NET Windows Forms tetapi tidak menyertakan *source code*-nya, sedangkan DXperience ASP.NET meliputi semua komponen untuk ASP.NET tetapi tidak menyertakan *source code*-nya.

DXperience ASP.NET v8.3

DXperience ASP.NET adalah kumpulan komponen dan library ASP.NET dalam satu *package* yang telah mendukung AJAX, memiliki kemampuan untuk mengupdate isi kontrol menggunakan *callbacks* sehingga tidak memerlukan *reload* keseluruhan halaman. DXperience ASP.NET meliputi ASPxperience Suite, ASPxGauges Suite, ASPxGridView and Editors Suite, ASPxHtmlEditor Suite, ASPxPivotGrid Suite, ASPxScheduler Suite, ASPxSpellChecker Suite, ASPxTreeList Suite, XtraCharts Suite, XtraReports Suite. Seluruh komponen pada DXperience ASP.NET mendukung semua browser utama mencakup Internet Explorer 5.5+, Mozilla, Mozilla Firefox, Netscape 7.2+, Opera 8+, dan Apple Safari.

2.6.1. ASPxGridView and Editors Suite

ASPxGridView merupakan web grid untuk ASP.NET yang dapat melakukan pemrosesan pada sisi server secara asinkron menggunakan teknologi *callback* berbasis AJAX.

Berikut beberapa fitur yang dimiliki ASPxGridView and Editors Suite :

1. Mendukung AJAX

ASPxGridView memiliki kemampuan mengupdate isi dari kontrol grid melalui *callback*, sehingga tidak memerlukan ASP.NET AJAX update panel untuk memungkinkan kemampuan ini. Update panel hanya berfungsi apabila ingin mensinkronkan isi grid dengan kontrol yang lain. Dengan menempatkan semua kontrol yang terkait pada sebuah panel dan memaksa grid untuk mengirim via *postback* dalam menanggapi aksi dari pengguna.

2. Data Editing

ASPxGridView menyediakan tiga *built-in edit mode* :

1) Edit Form

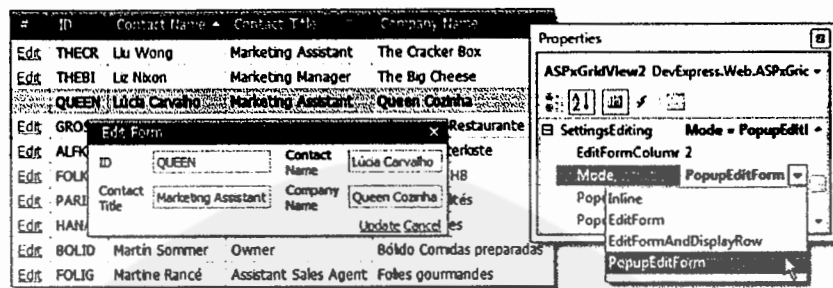
Form edit ditampilkan saat ASPxGridView berada pada *edit mode*. Edit form menampilkan edit cells, tiap cell mempresentasikan masing-masing kolom data dan dapat diubah nilainya pada baris yang diedit. Contoh edit form ditunjukkan pada gambar 2.4 :

#	First Name	Last Name	Birth Date	Title	Hire Date
	Janet	Levering	8/30/1963	Sales Representative	4/1/1992
First Name	<input type="text" value="Janet"/>	Title	<input type="text" value="Sales Representative"/>	Last Name	<input type="text" value="Levering"/>
Birth Date	<input type="text" value="8/30/1963"/>	<input type="checkbox"/>	Hire Date	<input type="text" value="4/1/1992"/>	<input type="checkbox"/>
Notes	<p>Janet has a BS degree in chemistry from Boston College (1984). She has also completed a certificate program in food retailing management. Janet was hired as a sales associate in 1991 and promoted to sales representative in February 1992.</p>				
					Update Cancel

Gambar 2.4 Edit Form

2) Popup Edit Form

Pada popup edit form, form edit ditampilkan dalam jendela pop up. Contoh popup edit form ditunjukkan pada gambar 2.5 :



Gambar 2.5 PopUp Edit Form

3) In-Line Editing

Pada in-line editing, baris edit ditampilkan langsung di dalam baris ASPxGridView. Edit cell ditampilkan bersesuaian dengan kolom yang ditampilkan. Contoh in-line editing ditunjukkan pada gambar 2.6 :

#	Contact Name	Company Name
Edit	Maria Anders	Alfreds Futterkiste
Update Cancel	Ana Trujillo	Ana Trujillo Emparedados y helados

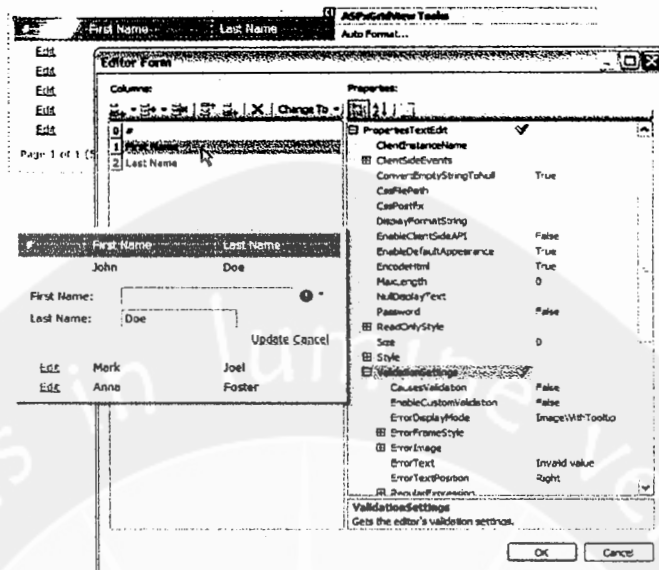
Gambar 2.6 In-line Editing

3. Edit Form Validation

Data yang diinputkan oleh user pada edit form dapat divalidasi pada sisi klien atau sisi server.

1) Client-side validation

Validasi di sisi klien menghindarkan mengirim *callback* ke server yang akan meningkatkan performansi. Pengaturan validasi pada editor dapat dilakukan melalui `PropertiesTextEdit.ValidationSettings` property. Jika inputan data tidak valid, ditampilkan error icons di samping editor yang nilainya tidak valid. Contoh *client side validation* ditunjukkan oleh gambar 2.7 :



Gambar 2.7 Client-Side Validation

Client-side validation tidak dapat berfungsi jika mendesign edit form sendiri dengan menambahkan editor pada Template Container di Edit Form atau Edit Item Template Container di kolom grid. Custom editor yang digunakan pada template container tidak dapat dikenali, menyebabkan validasi pada sisi klien diabaikan ketika tombol Update ditekan sehingga 'UPDATEEDIT' callback dikirim ke server. Oleh karena itu pada kondisi ini validasi harus dilakukan pada sisi server.

2) Server-side validation

Validasi baris pada sisi server diimplementasikan melalui event `ASPGridView.RowValidating` yang secara otomatis dijalankan ketika baris mengalami update. Untuk menjalankan validasi baris secara manual dilakukan dengan memanggil method `ASPGridView.DoRowValidation`. `RowError` property digunakan untuk menampilkan error text pada baris error (*error row*). Baris error otomatis ditampilkan di

bawah baris yang diedit jika RowError property tidak diset string kosong. Contoh *client side validation* ditunjukkan oleh gambar 2.8 :

#	First Name	Last Name
	John	Doe

First Name:

Last Name:

[Update](#) [Cancel](#)

Error Row [Empty strings are not allowed.

Edit	Mark	Joel
Edit	Anna	Foster

Gambar 2.8 Server-Side Validation

4. Data Shaping and Manipulation

1) Data Summaries

ASPxGridView mendukung dua tipe summary :

a. Total Summaries

Nilai yang dihasilkan adalah hasil penghitungan dari seluruh baris yang ada pada ASPxGridView dan ditampilkan pada footer.

b. Group Summaries

Nilai yang dihasilkan adalah hasil penghitungan dari seluruh baris yang ada pada satu group dan ditampilkan pada baris group atau pada group footer.

Contoh *data summaries* ditunjukkan oleh gambar 2.9 :

Company Name	Country	Region	Unit Price	Total	Quantity	Total	
City: Warszawa (\$3,531.95, Count=16)							
City: Wala Wala (\$357.00, Count=2)							
Lazy K Kountry Store	USA	WA	14.7	\$147.00			
Lazy K Kountry Store	USA	WA	21	\$210.00	10	\$190.00	
City: Versailles (\$1,992.05, Count=11)							
City: Toulouse (\$10,272.35, Count=31) (Continued on the next page)							
La maison d'Asie	France		28.8	\$288.00	20	\$800.00	
La maison d'Asie	France		13.6	\$136.00	10	\$60.00	
					21	\$714.00	
					16	\$200.00	
Count=2155				\$1,354,458.59	40	\$1,392.00	
Page 1 of 12 (114 items)						20	\$260.00
					12	\$228.00	
Ricardo Adocados	Rio de Janeiro		\$12.00		35	\$420.00	
QUICK-Stop	Cunewalde		\$22.00		50	\$1,100.00	
Count=2155					Min=1		
					Avg=23.81	\$1,354,458.59	
					Max=130		

Gambar 2.9 Data Summaries

2) Filtering

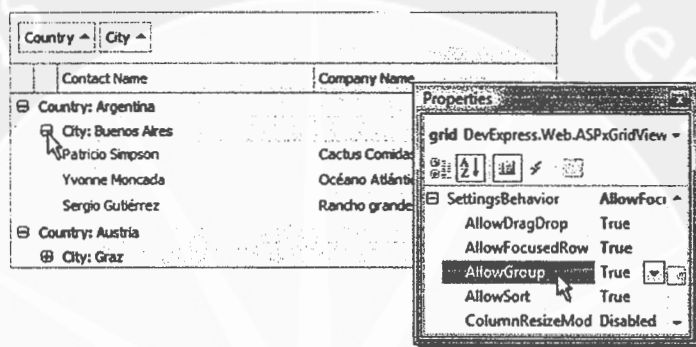
Filtering adalah menampilkan record data dari data source yang memenuhi kriteria yang diinputkan. Pengguna dapat memfilter data yang ditampilkan pada ASPxGridView dengan menginputkan teks pada Filter Row. Selain itu, kolom header ASPxGridView dapat menampilkan tombol untuk *data filtering*. Dengan mengklik tombol filter pada header kolom, maka akan ditampilkan drop down list yang berisi nilai unik yang ada pada kolom sehingga kita dapat dengan mudah menyaring data berdasarkan kolom tersebut. Contoh filter data menggunakan tombol filter pada header kolom ditunjukkan pada gambar 2.10 :

Company Name	City	Region	Country
(All)		Isle of Wight	UK
Alfreds Futterkiste		Isle of Wight	UK
Ana Trujillo Emparedados y helados		Isle of Wight	UK
Antonio Moreno Taquería		Isle of Wight	UK
Around the Horn		Isle of Wight	UK
Berglunds snabbköp			Italy
Blauer See Delikatessen			Italy
Reggiani Caseifici	Reggio Emilia		Italy

Gambar 2.10 Header Filter Button

3) Grouping

ASPxGridView dapat melakukan grouping data berdasarkan satu atau beberapa kolom data. Untuk mengelompokkan data berdasarkan kolom yang diinginkan dilakukan dengan men-drag header dari kolom tersebut ke group panel yang terletak di atasnya. Gambar 2.11 berikut menunjukkan contoh grouping data berdasarkan kolom country dan kolom city :



Gambar 2.11 Data Grouping

4) Sorting

Pengurutan berdasarkan kolom data secara ascending atau descending.

5. Representasi data dengan Master-Detail

ASPxGridView menyediakan pilihan Master-Detail dalam menyajikan data, dimana terdapat suatu grid sebagai Master dan grid lain yang bersarang dalam grid Master sebagai Detail-nya. Dengan hampir tanpa kode, kita dapat membangun *master-detail layout* dengan segala kompleksitas, dengan berapapun jumlah tingkatan bersarang dan berapapun jumlah detail pada masing-masing tingkatan. Tidak ada batasan dalam menampilkan data pada detail grid.

2.6.2. XtraReports Suite

XtraReports adalah platform *reporting* untuk para developer Windows Forms dan ASP.NET, memiliki *report designer* yang mudah digunakan dan terintegrasi penuh dengan MS Visual Studio IDE. XtraReports menggunakan *code editor* yang sama (C#, VB.NET atau J#), komponen toolbox yang sama, dan menggunakan semua *resources* yang tersedia pada Visual Studio IDE.

Berikut beberapa fitur yang terdapat pada XtraReports :

1. *All in one*

XtraReports bekerja pada Windows *projects* maupun ASP.NET *applications*. Dengan membuat satu laporan dapat digunakan pada kedua platform yang berbeda tersebut.

2. Integrasi penuh dengan Visual Studio .NET

Report designer terintegrasi penuh ke dalam Visual Studio .NET. XtraReports *publish* secara langsung ke dalam kode .NET dan mendukung VS.NET *command menus* dan *bars*.

3. Standard .NET data objects, mendukung *IList* interface, dan XML data objects

XtraReports bekerja dengan semua objek data yang didukung oleh Visual Studio .NET, dapat mem-*bind report* ke data XML atau object data yang mengimplementasikan *IList* interface.

4. XtraReports *controls*

Terdapat banyak kontrol dari XtraReports yang meliputi *Line*, *BarCode*, *CheckBox*, *PageInfo*, *Panel*, *PictureBox*, *PageBreak*, *Table*, *ZipCode*, dan lain sebagainya.

5. Mendukung Master-Detail report

Master-detail report dapat dibuat hanya dengan menggunakan satu report dan datasource tunggal dengan relasi data ADO.NET. Dengan mendefinisikan relasi DataSet kemudian menambahkan Detail Report band yang menggambarkan relasi tersebut.

6. Kaya akan *exporting format*

Report yang dihasilkan dapat di-export ke dalam format PDF, HTML, MHT, RTF, TXT, CSV dan MS Excel. Dapat di-export pula ke dalam *image* format seperti BMP, EMF, GIF, JPEG, PNG, TIFF dan WMF.

2.6.3. XtraCharts Suite

XtraCharts Suite merupakan *charting control* yang menyediakan berbagai tipe grafik 2D dan 3D. XtraCharts Suite menyediakan berbagai palletes untuk pewarnaan grafik secara otomatis. XtraCharts bekerja pada Windows *projects* maupun ASP.NET *applications*. Untuk Windows Forms project menggunakan ChartControl sedangkan untuk membuat grafik pada halaman web menggunakan WebChartControl. WebChartControl mendukung AJAX *callback* untuk mengupdate isi dari kontrol *chart*.

2.6.4. ASPxperience

2.6.4.1 ASPxNavBar

ASPxNavBar merupakan komponen navigasi situs ASP.NET dengan dukungan *built-in* template untuk membuat tampilan navigasi yang berbeda sesuai keinginan kepada *end user*. ASPxNavBar dapat di-bind dari data source file XML untuk menampilkan informasi group dan item.

2.6.4.2 ASPxTabControl

ASPxTabControl merupakan komponen untuk menciptakan tampilan berupa tab pada web site. ASPxTabControl terdiri dari dua komponen, yakni ASPxTabControl yang digunakan hanya untuk menampilkan tab atau ASPxPageControl yang digunakan untuk membuat isi halaman tertentu pada setiap tabnya. Kedua kontrol dibangun dengan HTML code yang minimal, bekerja pada sisi *client*, cepat dan mendukung penuh AJAX pada permintaan halaman melalui *callbacks*.

2.6.4.3 ASPxRoundPanel

ASPxRoundPanel merupakan komponen untuk membuat panel dengan sudut tumpul. Komponen ini dilengkapi dengan *designer dialog* yang memungkinkan kita untuk menetapkan radius sudut, panel, warna perbatasan, *header visibility* dan warna.

2.7. Manajemen Kinerja Karyawan

2.7.1 Definisi Manajemen Kinerja Karyawan

Penilaian kinerja kerja karyawan adalah masalah penting bagi seluruh pengusaha. Namun demikian, kinerja yang memuaskan tidak terjadi secara otomatis dimana hal ini cenderung akan makin terjadi dengan menggunakan sistem penilaian manajemen yang baik. Sistem manajemen kinerja terdiri dari proses-proses untuk mengidentifikasi, mendorong, mengukur, mengevaluasi, meningkatkan dan memberi penghargaan terhadap kinerja para karyawan yang dipekerjakan (Mathis & Jackson, 2002).

Kinerja pada dasarnya adalah apa yang dilakukan atau tidak dilakukan karyawan. Kinerja karyawan adalah yang

mempengaruhi seberapa banyak mereka memberi kontribusi kepada organisasi yang antara lain termasuk kuantitas *output*, kualitas *output*, jangka waktu *output*, kehadiran di tempat kerja, dan sikap kooperatif.

Peningkatan kinerja dapat dilakukan melalui penilaian kinerja, penilaian kinerja adalah proses evaluasi seberapa baik karyawan mengerjakan pekerjaan mereka ketika dibandingkan dengan satu set standar, dan kemudian mengkomunikasikannya dengan para karyawan. Penilaian kinerja (*performance appraisal*) pada dasarnya merupakan faktor kunci guna mengembangkan suatu organisasi secara efektif dan efisien, karena adanya kebijakan atau program yang lebih baik atas sumber daya manusia yang ada dalam organisasi. Penilaian dapat dilakukan oleh atasan, karyawan, kelompok kerja, pihak luar, atau kombinasi dari para penilai.

Standar kinerja menjelaskan tingkat-tingkat kinerja yang diharapkan dan merupakan "bahan perbandingan", atau "tujuan", atau "target" tergantung dari pendekatan yang diambil. Standar kinerja yang realistis, terukur, dan mudah dipahami menguntungkan baik bagi organisasi maupun karyawan. Dalam artian, standar kinerja mendefinisikan tentang pekerjaan yang tergolong memuaskan.

Penilaian kinerja karyawan memiliki dua penggunaan yang umum di dalam organisasi, yakni (Mathis & Jackson, 2002) :

1. Penggunaan administratif

Penilaian kinerja digunakan untuk menentukan kompensasi (gaji) yang diterima berdasarkan bagaimana mereka melaksanakan pekerjaan mereka. Penggunaan administratif lainnya dari penilaian kinerja adalah

seperti keputusan untuk promosi, pemecatan, pengurangan, dan penugasan pindah tugas.

Penilaian kinerja adalah penting ketika organisasi memberhentikan, mempromosikan, atau membayar orang-orang secara berbeda, karena hal-hal ini membutuhkan pembelaan yang kritis jika karyawan menuntut keputusan yang ada.

2. Penggunaan untuk pengembangan

Penilaian kinerja dapat menjadi sumber informasi utama dan umpan balik untuk karyawan, yang merupakan kunci bagi pengembangan mereka di masa mendatang. Penilaian kinerja digunakan untuk mengidentifikasi kelemahan, potensi, dan kebutuhan pelatihan untuk karyawan.

2.7.2 Format dan Perumusan Penilaian

Berikut perumusan penilaian kinerja karyawan yang digunakan dalam perangkat lunak yang dikembangkan :

- a. Indikator penilaian memiliki bobot yang berbeda pada masing-masing posisi kerja. Besar kecilnya bobot untuk suatu posisi kerja mencerminkan pentingnya indikator penilaian pada posisi kerja tersebut.
- b. Total bobot untuk satu posisi sama dengan 100 (dalam persen).
- c. Setiap indikator penilaian memiliki nilai maksimum yang diperbolehkan.
- d. Penilaian mingguan adalah rekap nilai rata-rata mingguan dari evaluasi kerja harian.
- e. Penyusunan target kerja adalah salah satu dari indikator penilaian yang nilainya didapat dari tanggal penyusunan target kerja per periode satu bulan.

Rumus yang digunakan : tidak menyerahkan = 0; menyerahkan lebih dari tanggal 20 = 60; menyerahkan tanggal 15 sampai dengan 20 = 70; menyerahkan tanggal 7 sampai 14 = 80; menyerahkan kurang dari tanggal 7 = 90.

- f. LHKK adalah kependekan dari Laporan Harian Kinerja Karyawan yang merupakan lembaran *check list* penilaian kinerja harian dari karyawan.

Evaluasi LHKK adalah salah satu dari indikator penilaian yang nilainya didapat dari tanggal penyerahan lembar harian kinerja karyawan per periode satu bulan.

Rumus yang digunakan : tidak menyerahkan = 0; menyerahkan lebih dari tanggal 20 = 60; menyerahkan tanggal 15 sampai dengan 20 = 70; menyerahkan tanggal 7 sampai 14 = 80; menyerahkan kurang dari tanggal 7 = 90.

- g. IMTKP adalah kependekan dari Ijin Meninggalkan Tempat Kerja Pribadi, yaitu jumlah jam tidak masuk kerja karena ijin untuk keperluan pribadi selama jam kerja, meliputi *job site* untuk keperluan pribadi, ijin datang terlambat dan pulang lebih awal.

IMTKP adalah salah satu dari indikator penilaian yang mencatat jumlah jam dan frekuensi dari ijin meninggalkan tempat kerja untuk kepentingan pribadi dalam periode satu semester. Nilai didapatkan dari jumlah jam IMTKP, sedangkan frekuensi IMTKP akan mengurangi perolehan nilai akhir.

Rumus yang digunakan : 0 jam = 100; 0.1 - 3.0 jam = 95; 3.1 - 6.0 jam = 90; 6.1 - 9.0 jam = 80; 9.1 - 12.0

jam = 70; 12.1 - 15.0 jam = 60; 15.1 - 18.0 jam = 50;
> 18.0 jam = 0.

Frekuensi IMTKP mengurangi nilai akhir :

1 kali = -0	6 kali = -0.75	11 kali = -3.0
2 kali = -0	7 kali = -1.0	12 kali = -3.5
3 kali = -0	8 kali = -1.5	13 kali = -4.0
4 kali = -0.25	9 kali = -2.0	14 kali = -4.5
5 kali = -0.50	10 kali = -2.5	15 kali = -5.0

dan seterusnya bertambah (-0.5).

- h. Absensi/Ketidakhadiran adalah jumlah hari tidak hadir di Perusahaan yang dapat mengurangi kontribusi terhadap produktivitas Perusahaan, meliputi semua hari tidak hadir kerja, ijin sakit karena haid, tidak termasuk cuti tahunan, cuti khusus, ijin sakit opname di rumah sakit, dan ijin sakit karena mengalami kecelakaan saat menjalankan tugas/pekerjaan.

Absensi adalah salah satu dari indikator penilaian yang mencatat banyaknya jumlah absen dalam hari dalam periode satu semester. Nilai didapatkan dari jumlah hari absensi.

Rumus yang digunakan : 0 hari = 100; 1 hari = 85;
2 hari = 75; 3 hari = 65; 4 hari = 55; 5 hari = 45;
6 hari = 35; 7 hari = 25; > 7 hari = 0.

- i. Tukar shift adalah tidak hadir sesuai jadwal shift yang ditetapkan oleh perusahaan.

Jumlah tukar shift dalam satu semester akan mengurangi perolehan nilai akhir. Rumus yang digunakan :

1 kali = -0.25	5 kali = -1.25	9 kali = -3.00
2 kali = -0.50	6 kali = -1.50	10 kali = -3.50
3 kali = -0.75	7 kali = -2.00	11 kali = -4.00
4 kali = -1.00	8 kali = -2.50	12 kali = -4.50

