

## **BAB II**

### **LANDASAN TEORI**

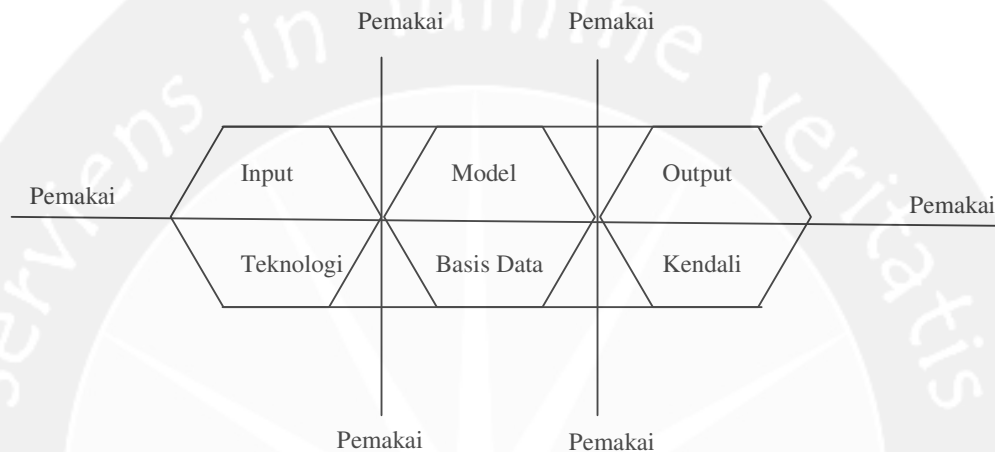
#### **2.1 Konsep Dasar Sistem Informasi**

Dalam dunia teknologi informasi dan komunikasi, sistem informasi memegang peranan yang sangat penting bagi suatu manajemen di dalam pengambilan keputusan. Untuk memahami definisi dari sistem informasi, terlebih dahulu kita harus mengerti dua kata yang menyusunnya yaitu sistem dan informasi. Sistem adalah kumpulan objek 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 (Margianti & Suryadi, 1995).

Sistem informasi merupakan sebuah sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan bagi pihak luar tertentu.

Sistem informasi terdiri dari komponen-komponen yang saling berhubungan dan bergantung satu sama lain. Komponen yang tidak saling berhubungan tidak akan membentuk sebuah sistem. Dalam membentuk sebuah sistem informasi yang berbasis komputer, ada beberapa komponen

yang saling terkait yang disebutnya dengan istilah "Building Block" (Gambar 2.1.), yaitu: blok masukan/*input*, blok proses/*model*, blok keluaran/*ouput*, blok basis data, blok kontrol/*kendali*, dan blok teknologi.



**Gambar 2.1. Blok Sistem Informasi  
(Mudjihartono, 1998)**

### **2.1.1 Input**

Input (masukan) adalah sekumpulan data yang bisa terjadi dari transaksi, kejadian, *order* atau permintaan. Data tersebut pada umumnya berupa teks, angka, simbol khusus, dan lebih spesifik lagi bisa berupa citra, suara, atau signal analog yang akan diubah ke signal digital lewat transduser.

### **2.1.2 Proses/Model**

Proses/Model adalah sebetuk abstraksi atau representasi dari realita. Proses biasanya merupakan bentuk penyederhanaan daru hal yang nyata/kenyataan. Proses atau model diklasifikasikan ke dalam berbagai cara, misalnya:

1. Model Prosedural, adalah seperangkat pernyataan deklaratif. Setiap pernyataan bersifat *action oriented* yang berisi aksi yang harus diambil dan sebuah objek yang dikenai aksi.
2. Model Logika, hal mendasar yang ada pada model logika adalah aturan/*rule*. Sebuah aturan mengandung kondisi yang harus diuji, dan bergantung kepada hasil kondisi. Sebuah aturan juga mengandung aksi yang harus diambil.
3. Model Matematika, adalah representasi kuantitatif dari realitas.

### **2.1.3 Output**

Output (keluaran) adalah komponen yang bertanggung jawab terhadap penyajian bentuk hasil sistem yang berupa informasi yang akan dikirimkan ke luar sistem. Efektivitas keluaran bergantung kepada kualitas dan tingkat kegunaan informasi yang disajikannya.

Kualitas keluaran harus memiliki sifat-sifat seperti:

1. Akurasi, yaitu keluaran bebas dari kesalahan dan gangguan berdasarkan data mentah yang diperiksa.
2. *Timeliness*, yaitu keluaran merefleksikan situasi dan data yang paling aktual dan ketersediaan informasi setiap saat.
3. Relevansi, yaitu keluaran harus mempunyai hubungan yang kuat dengan hal yang ditangani dan calon keputusan yang akan diambil).

### **2.1.4 Basis Data**

Basis data merupakan kumpulan dari elemen-elemen data. Elemen data tersebut terbagi ke dalam tabel-

tabel, dimana tabel-tabel itu mempunyai sejumlah kolom dan baris. Suatu tabel akan mempunyai kolom kunci yang digunakan untuk membangun hubungan dengan tabel lain. Kolom inilah yang mengidentifikasi secara unik setiap baris di dalam sebuah tabel. Kolom kunci ini disebut juga dengan kunci primer. Semua kolom dalam tabel dapat dibuat sebagai kunci primer, selama memenuhi ketentuan berikut:

1. Setiap baris harus memiliki nilai kunci primer (kolom tidak boleh bernilai kosong atau null).
2. Kolom yang berisikan nilai kunci primer tidak pernah dapat dimodifikasi dan diperbaharui.
3. Nilai kunci primer tidak dapat digunakan kembali (jika baris tersebut sudah dihapus dari dalam tabel, kunci primernya tidak dapat diberikan kepada baris-baris berikutnya atau baris baru).

Hubungan atau relasi antar tabel melalui kunci primer inilah yang disebut dengan basis data relasional. Semakin banyak tabel yang ada dalam suatu basis data, semakin banyak pula relasi yang diperlukan untuk menghubungkan semua tabel. Tujuan dibangunnya basis data relasional adalah sebagai berikut:

1. Membuat derajat kebebasan yang tinggi.
2. Menyediakan landasan yang kokoh yang berhubungan dengan masalah yang berkaitan dengan semantik data, konsistensi data, dan redundansi data.

#### **2.1.5 Kontrol**

Kontrol merupakan komponen sistem informasi yang didesain untuk memastikan adanya proteksi, integritas, dan kelancaran operasi dari suatu sistem informasi.

### **2.1.6 Teknologi**

Teknologi merupakan alat bantu untuk bekerjanya sistem informasi. Teknologi menangkap masukan, menjalankan model, menyimpan data yang diakses, menghasilkan dan mentransmisikan keluaran, dan membantu mengendalikan seluruh sistem. Teknologi mengandung tiga area utama, yaitu: *Brainware*, *Software*, dan *Hardware*.

#### 1. *Brainware*

*Brainware* merupakan sumber daya manusia atau bisa dikatakan juga sebagai teknisi yang terkait secara langsung dengan sistem informasi.

#### 2. *Software*

*Software* atau perangkat lunak merupakan bagian dari komputer yang secara fisik tidak terlihat. Namun, sangat penting bagi kelangsungan hidup komputer.

#### 3. *Hardware*

*Hardware* atau perangkat keras merupakan bagian dari komputer yang secara fisik terlihat. *Hardware* dibagi menjadi empat bagian, yaitu: piranti input, piranti pemrosesan, piranti keluaran, dan piranti penyimpan luar/eksternal.

### **2.2. Siklus Hidup Pengembangan Sistem**

Pengembangan sistem informasi harus dilakukan dengan melalui berbagai tahapan, mulai dari perencanaan sistem sampai penerapan, pengoperasian, dan pemeliharaan sistem. Apabila pengoperasian sistem yang dikembangkan masih menimbulkan permasalahan-permasalahan serta sistem belum mencapai tahap pemeliharaan sistem, maka perlu dikembangkan kembali dan proses ini kembali ke tahap awal, yakni tahap perencanaan sistem. Siklus pengembangan sistem

informasi ini disebut dengan *System Life Cycle*. Tiap-tiap tahapan dalam pengembangan sistem ini mempunyai karakteristik tersendiri. Tahapan utama siklus hidup sistem terdiri dari:

1. Perencanaan Sistem (*System Planning*)

Pada tahap ini, akan dibuat suatu perencanaan atau garis-garis besar sistem yang nantinya akan dianalisis. Tahap ini merupakan tahapan pertama. Segala kemungkinan/ide-ide dapat dimasukkan dalam tahap ini. Selanjutnya, baru diproses di tahap berikutnya yaitu Analisis Sistem.

2. Analisis Sistem (*System Analysis*)

Pada tahap ini, sistem informasi akan diuraikan ke dalam bagian komponen-komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, dan hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya. Tahap analisis merupakan tahapan kritis dan sangat penting, karena kesalahan pada tahap ini juga akan menyebabkan kesalahan pada tahap selanjutnya.

3. Perancangan Sistem (*System Design*)

Tahap ini menyangkut konfigurasi dari komponen-komponen *software* dan *hardware* dari suatu sistem sehingga instalasi sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. Komponen sistem informasi yang dirancang adalah model, *output*, *input*, basis data, teknologi, dan kontrol.

#### 4. Implementasi Sistem (*System Implementation*)

Tahap ini akan mewujudkan sistem sesuai dengan rancangan yang telah dibuat pada tahap sebelumnya yaitu tahap Perancangan Sistem. Sistem diimplementasikan sesuai dengan bahasa pemrograman dan basis data yang diinginkan.

#### 5. Pemeliharaan Sistem (*System Maintenance*)

Pada tahap ini sistem yang telah selesai diimplementasikan akan dipelihara untuk menjaga agar kinerja sistem selalu efektif dan efisien. Pada tahap ini juga dimungkinkan untuk mengembangkan sistem jika ternyata di kemudian hari terdapat suatu perubahan yang tidak mengubah keseluruhan sistem.

### 2.3 Sistem Informasi Berbasis Web

Seiring dengan perkembangan teknologi web dan informasi, maka dikembangkan model sistem informasi yang dikenal dengan nama *web based information system* atau sistem informasi berbasis web. Sistem informasi berbasis *web* merupakan sistem informasi yang berjalan di *internet*. Dimana basis datanya diletakan di *web server*. Sehingga untuk mengaksesnya harus menggunakan halaman web. *User* dapat dengan mudah mengakses informasi melalui *internet browser* dalam bentuk halaman web. Keuntungan menggunakan sistem informasi berbasis web antara lain user dapat mengaksesnya dari berbagai tempat melalui komputer yang terkoneksi dengan *internet* dan kecepatan penyampaian informasi kepada user.

Dewasa ini seiring dengan perkembangan teknologi web, berkembang pula teknologi *mobile*. Handphone keluaran terbaru telah menyediakan *web browser* pada

perangkatnya sebagai fasilitas untuk mengakses halaman web. Dengan demikian, *user* dapat mengakses informasi dari berbagai tempat melalui handphone yang memiliki *web browser* dengan berbagai koneksitas seperti GPRS, 3G, 3.5G atau melalui *hotspot* diberbagai tempat.

## **2.4 PRADO Framework**

### **2.4.1 Pengenalan PRADO**

PRADO adalah kerangka pemrograman berbasis-komponen dan kendali-event untuk mengembangkan aplikasi web dalam PHP 5. PRADO merupakan singkatan dari PHP Rapid Application Development Object-oriented.

Tujuan utama dari PRADO adalah menghidupkan reusabilitas secara maksimum dalam pemrograman web (Xue et al, 2009). Dengan reusabilitas, dapat diartikan tidak hanya menggunakan ulang kode yang dimiliki seseorang, tetapi juga menggunakan ulang kode orang lain dalam cara yang mudah. Yang terakhir lebih penting karena menghemat usaha penciptaan program dan menghemat waktu pengembangan. Untuk mencapai tujuan di atas, PRADO menetapkan sebuah protokol atas penulisan dan pemakaian komponen untuk membentuk aplikasi web. Komponen adalah unit software yang mengandung-dirinya sendiri dan dapat dipakai ulang dengan kustomisasi mudah. Komponen baru dapat dibuat dengan komposisi sederhana terhadap komponen-komponen yang sudah ada.

Untuk menjembatani interaksi dengan komponen, PRADO menerapkan paradigma pemrograman kendali-event yang memperbolehkan delegasi aturan yang bisa diperpanjang ke komponen. Aktivitas *end-user* seperti mengklik pada tombol kirim, ditangkap sebagai event



server. Metode atau fungsi dapat disertakan pada event ini agar ketika event terjadi, metode atau fungsi dipanggil secara otomatis untuk merespon event. Dibandingkan dengan pemrograman web tradisional di mana para pengembang harus berhadapan dengan variabel kasar *POST* atau *GET*, pemrograman kendali-event membantu para pengembang lain lebih berfokus pada logika yang diperlukan dan mengurangi secara drastis pengkodean berulang kali.

Ringkasnya, mengembangkan aplikasi Web PRADO sebagian besar menyangkut penurunan tipe komponen yang sudah dibuat sebelumnya, mengkonfigurasinya dengan menyetel propertinya, merespon ke event-nya dengan menuliskan fungsi pengendali, dan mengaturnya ke dalam halaman untuk aplikasi.

#### **2.4.2 Sejarah PRADO**

Inspirasi original PRADO berasal dari Apache Tapestry. Selama desain dan implementasi, ide pengembangan PRADO meminjam ide dari Borland Delphi dan Microsoft ASP.NET. Versi pertama PRADO keluar dalam bulan Juni 2004 dan ditulis dalam PHP 4. Karena adanya kontes kode Zend PHP 5, maka pengembangan menulis ulang PRADO dalam PHP 5, yang membuktikan menjadi perkembangan yang lebih baik. PRADO memenangkan hadiah utama dalam kontes Zend, memperoleh pilihan tertinggi baik dari umum dan panelis juri.

Pada bulan Agustus 2004, PRADO mulai ditampung pada SourceForge sebagai proyek sumber terbuka. Segera setelah itu, situs proyek [xisc.com](http://xisc.com) diumumkan kepada publik. Dengan dukungan fantastis dari tim pengembang

dan pengguna PRADO, PRADO berkembang ke versi 2.0 di pertengahan tahun 2005. Dalam versi ini, Wei Zhuo berkontribusi ke PRADO dengan dukungan I18N dan L10N yang istimewa. Kemudian pada bulan Mei 2005, diputuskan untuk menulis ulang sepenuhnya kerangka kerja PRADO untuk memecahkan beberapa isu dasar dalam versi 2.0 dan untuk menangkap dengan beberapa fitur dalam Microsoft ASP .NET 2.0. Setelah hampir setahun kerja keras dengan lebih dari 50000 baris kode baru, versi 3.0 akhirnya dirilis pada bulan April 2006.

Mulai PRADO versi 3.0, usaha signifikan dialokasikan untuk memastikan kualitas dan stabilitas PRADO. Dapat dikatakan PRADO versi 2.x dan versi 1.x adalah pekerjaan bukti-konsep, sedangkan PRADO versi 3.x telah berkembang menjadi proyek yang layak untuk pengembangan aplikasi bisnis serius. *Framework* PRADO versi terbaru saat ini adalah PRADO versi 3.1.4 yang dirilis pada 11 Januari 2009.

#### **2.4.3 Fitur-Fitur Utama PRADO**

PRADO merupakan sebuah kerangka kerja (*framework*) yang unik, sehingga dapat mengalihkan pemrograman PHP yang membosankan menjadi lebih cepat dan menyenangkan. Berikut ini fitur-fitur utama PRADO yakni (Xue et al, 2009):

##### *1. Reusability*

Kode mengikuti protokol komponen yang dapat digunakan secara berulang-ulang. Hal ini menguntungkan tim pengembang selama bekerja, karena dapat menggunakan kembali pekerjaan sebelumnya

serta mengintegrasikan pekerjaan pihak lain dengan mudah.

2. Pemrograman kendali *event*

Aktivitas yang dilakukan oleh pengguna akhir, seperti mengklik pada tombol kirim, dianggap sebagai sebuah *event*. Dengan begitu para pengembang lebih terfokus dengan bagaimana mengendalikan *event* yang terjadi akibat aktivitas pengguna.

3. Integrasi Tim

Tampilan (*presentation*) dan logika (*logic*) disimpan secara terpisah sehingga membuat pekerjaan tim menjadi lebih mudah. Seorang pengembang tampilan hanya perlu tahu bagaimana membuat tampilan yang baik, sedangkan logika atas tampilan tersebut bisa dikerjakan oleh pengembang lain.

4. *Powerful web controls*

PRADO hadir dengan set komponen berkaitan dengan antarmuka pengguna web. Halaman web interaktif dapat dibuat dengan beberapa baris kode. Sebagai contoh, menggunakan komponen *datagrid*, seseorang bisa dengan cepat membuat sebuah halaman yang menyajikan tabel data yang membolehkan banyak halaman, pengurutan, penyuntingan, dan penghapusan baris data.

5. Dukungan database yang lengkap

Sejak versi 3.1, PRADO telah dilengkapi dengan dukungan database lengkap yang secara alami ditulis dan selanjutnya cocok dengan kerangka kerja PRADO lainnya. Berdasarkan kompleksitas dari obyek bisnis, seseorang dapat memilih untuk menggunakan akses data berbasis-PDO, atau rekaman aktif yang

lebih umum dikenal, atau skema pemetaan obyek bisnis lengkap SqlMap.

6. Dukungan AJAX

Menggunakan AJAX dalam PRADO jauh lebih mudah dengan kontrol aktif inovatif yang diperkenalkan sejak versi 3.1. Dengan mudah kita dapat menulis aplikasi AJAX-enabled tanpa menuliskan satu baris pun kode javascript. Kenyataannya, menggunakan kontrol aktif tidak jauh berbeda dengan menggunakan kontrol web reguler non-AJAX.

7. Dukungan Multibahasa (*I18N and L10N support*)

PRADO menyertakan dukungan lengkap untuk membangun aplikasi dengan multi bahasa dan lokal. Dengan dukungan komponen *I18N* dan *L10N*, membuat halaman web yang mendukung multi bahasa menjadi sangat mudah sekali.

8. *XHTML* Standar

Halaman web yang dibuat menggunakan PRADO sesuai dengan aturan standar *XHTML*.

9. *Accommodation of existing work*

PRADO adalah kerangka kerja umum dengan fokus pada lapisan *presentation*. Hal itu tidak mengecualikan para pengembang dari pemakaian *library* kelas atau kit piranti yang sudah ada. Sebagai contoh, seseorang dapat memakai AdoDB atau Creole untuk berhadapan dengan DB dalam aplikasi PRADO-nya.

10. Fitur lainnya

Penanganan kesalahan serta pencatatan pesan, cache generik dan cache output selektif, penanganan kesalahan yang dapat dikustomisasi serta dilokalisasi, otorisasi dan otentikasi yang

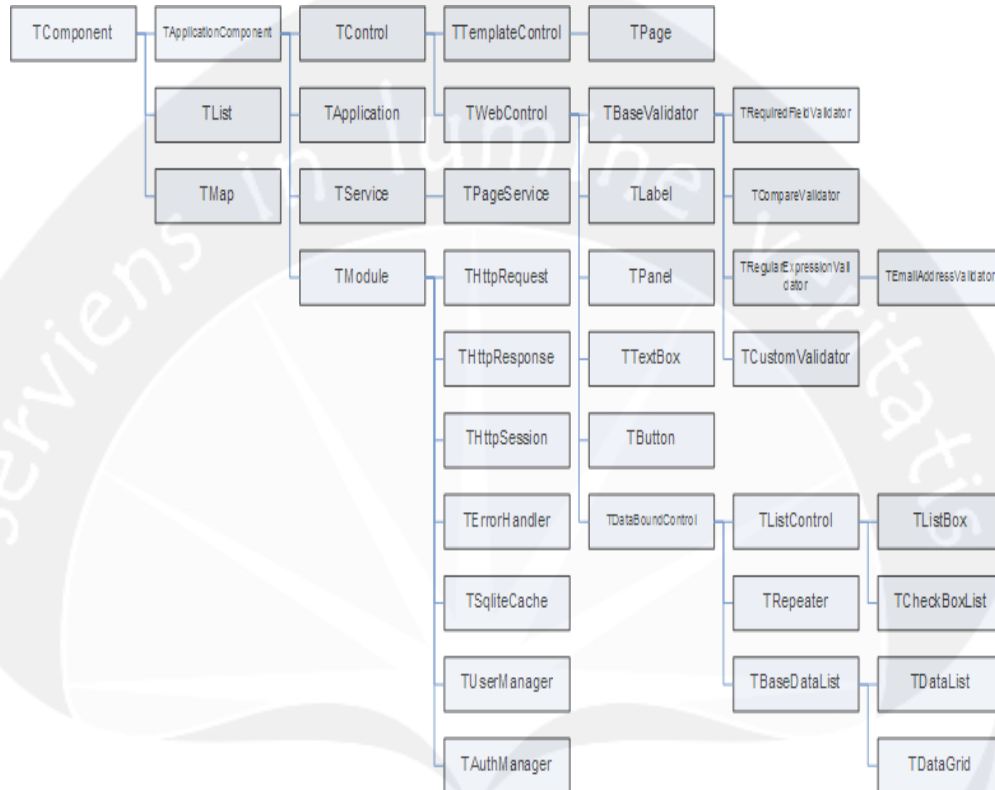
dapat diperluas, langkah pengamanan seperti pencegahan *cross-site script* (XSS), proteksi cookie, dll.

PRADO sering disebut sebagai kerangka kerja (*framework*) yang unik. Keunikannya terutama terletak pada paradigma pemrograman berbasis-komponen dan kendali-event (*component-based and event-driven programming paradigm*). Meskipun paradigma pemrograman ini bukan hal baru dalam pemrograman aplikasi desktop dan bukan hal baru dalam beberapa bahasa pemrograman web. Seperti kebanyakan *framework* PHP lainnya, PRADO berfokus pada pemisahan penyajian dan logika serta mempromosikan pola desain MVC (*model-view-controller*). PRADO melakukannya dengan cara pemisahan logika disimpan dalam kelas dan penyajian dalam *template* atau *page*. PRADO melakukan lebih banyak aspek tidak hanya MVC. PRADO mengisi banyak area kosong dalam pemrograman web PHP dengan paradigma pemrograman berbasis-komponen, kontrol web yang kaya, dukungan database yang handal, fitur penanganan kesalahan yang fleksibel serta fitur pencatatan, dan masih banyak lagi.

#### **2.4.5. Arsitektur PRADO**

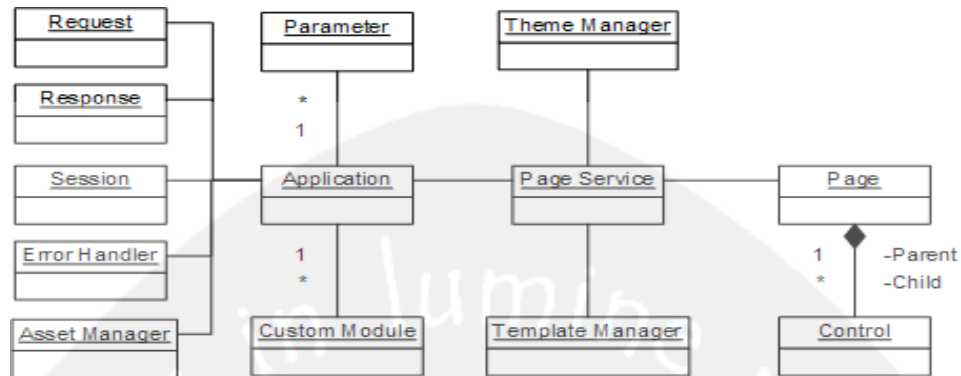
PRADO merupakan sebuah kerangka kerja presentasi utama (*primarily a presentational framework*). Kerangka kerja terfokus pada pembuatan pemrograman Web, yang banyak berkaitan dengan interaksi pengguna, menjadi berbasis-komponen (*component-based*) dan kendali-event (*event-driven*) agar pengembang bisa lebih produktif.

Berikut ini adalah susunan kelas yang menggambarkan beberapa kelas utama yang disediakan oleh PRADO:



**Gambar 2.2. Arsitektur Kelas Utama pada PRADO**  
(Xue et al, 2009)

Ketika aplikasi PRADO memproses permintaan halaman (*page request*), diagram obyek statisnya dapat dilihat sebagai berikut:



Gambar 2.3. Diagram Obyek Statis Ketika Aplikasi PRADO memproses *Page Request* (Xue et al, 2009)

#### 2.4.6 Halaman (*Page*) pada PRADO

Halaman (*page*) merupakan salah satu control yang ada pada PRADO. Presentasi halaman ditampilkan secara langsung ke pengguna (*end-user*). Pengguna mengakses halaman dengan mengirimkan permintaan layanan halaman.

Setiap halaman harus mempunyai file template. Akhiran (*file extensions*) dari nama file harus *.page*. Nama file (tanpa akhiran) adalah nama halaman. PRADO akan mencoba mencari file kelas halaman di bawah direktori yang berisi file template halaman. File kelas halaman tersebut harus mempunyai nama file yang sama (diakhiri dengan *file extensions .php*) seperti file template.

Kelas-kelas komponen yang disediakan PRADO yang akan dipakai didefinisikan di dalam halaman (*page*), misalnya untuk membuat komponen text box maka dengan cara sebagai berikut:

```
<com:TTextBox ID="txtPassword" TextMode="Password" Columns="20">
</com:TTextBox>
```

#### 2.4.7 Modul

Modul adalah turunan dari kelas yang menerapkan antarmuka `IModule`. Umumnya sebuah modul didesain untuk fungsionalitas tertentu yang dapat disertakan ke dalam aplikasi PRADO dan berbagi dengan semua komponen dalam aplikasi.

PRADO menggunakan konfigurasi untuk menetapkan apakah mengambil sebuah modul, mengambil modul jenis apa, dan bagaimana untuk menginisialisasi modul yang diambilnya. Para pengembang dapat mengganti modul inti dengan implementasinya sendiri melalui konfigurasi aplikasi, atau mereka dapat menulis modul baru guna menyediakan fungsionalitas tambahan. Sebagai contoh, modul dapat dikembangkan untuk menyediakan logika database umum pada satu atau beberapa halaman.

Ada tiga modul inti yang diambil secara standar kapan saja aplikasi dijalankan. Ketiganya adalah modul permintaan (*request module*), modul respon (*response module*), dan modul pengendali kesalahan (*error handler module*). Sebagai tambahan, modul sesi (*session module*) diambil saat ia digunakan dalam aplikasi. PRADO menyediakan implementasi standar untuk semua modul ini. Modul custom (*custom module*) dapat dikonfigurasi atau dikembangkan guna mengganti atau menambah modul inti ini. Modul kustom dan modul inti dapat dikonfigurasi melalui konfigurasi.

##### 1. Modul Permintaan

Modul permintaan menyediakan penyimpanan dan skema akses untuk permintaan pengguna yang dikirim melalui HTTP. Permintaan pengguna datang dari beberapa sumber, termasuk URL, data post, data sesi, data cookie, dll. Data ini semuanya dapat



diakses melalui modul permintaan. Standarnya PRADO menggunakan `THttpRequest` sebagai modul permintaan. Modul permintaan dapat diakses melalui properti `Request` dari aplikasi dan kontrol.

#### 2. Modul Respon

Modul respon menerapkan mekanisme untuk mengirimkan output ke pengguna klien. Modul respon dapat dikonfigurasi guna mengontrol bagaimana output di-cache pada sisi klien. Standarnya PRADO menggunakan `THttpResponse` sebagai modul respon. Modul respon dapat diakses melalui properti `Response` dari aplikasi dan kontrol.

#### 3. Modul Pengendali Kesalahan

Modul pengendali kesalahan dipakai untuk menangkap dan memproses semua kondisi kesalahan dalam sebuah aplikasi. PRADO menggunakan `TErrorHandler` sebagai modul pengendali kesalahan. Ia menangkap semua peringatan PHP dan eksepsi, dan ditampilkan dalam bentuk yang tepat bagi *end-user*. Modul pengendali kesalahan dapat diakses melalui properti `ErrorHandler` dari turunan aplikasi.

#### 4. Modul Sesi

Modul sesi melapisi fungsionalitas terkait dengan penanganan sesi pengguna. Modul sesi secara otomatis diambil saat aplikasi menggunakan sesi. Standarnya PRADO menggunakan `THttpSession` sebagai modul sesi, yang merupakan pelapis sederhana dari fungsi sesi yang disediakan oleh PHP. Modul sesi dapat diakses melalui properti `Session` dari aplikasi dan kontrol.

#### 5. Modul Custom

PRADO dirilis dengan beberapa modul lebih disamping modul inti. Modul custom ini meliputi modul cache (TSqliteCache dan TMemCache), modul manajemen pengguna (TUserManager), modul otentikasi dan otorisasi (TAuthManager), dll. Ketika TPageService diminta, ia juga mengambil modul tertentu untuk layanan halaman, termasuk manajer asset (TAssetManager), manajer template (TTemplateManager), manajer tema/skin (TThemeManager).

#### **2.4.8 Konfigurasi Aplikasi**

Konfigurasi aplikasi digunakan untuk menetapkan perilaku global atas aplikasi, termasuk spesifikasi dari alias path (*path aliases*), pemakaian namespace, konfigurasi modul dan layanan, serta parameter.

Konfigurasi untuk aplikasi disimpan dalam sebuah file XML bernama `application.xml`, yang harus ditempatkan di bawah path basis aplikasi (`protected`).

#### **2.4.9 Konfigurasi Halaman**

Konfigurasi halaman sebagian besar dipakai oleh TPageService untuk memodifikasi atau menambahkan konfigurasi aplikasi. Seperti ditunjukkan oleh namanya, konfigurasi halaman dikaitkan dengan direktori yang menyimpan beberapa file halaman. Konfigurasi halaman disimpan sebagai file XML bernama `config.xml`.

Sejak PRADO versi 3.1.1, atribut `id` dalam elemen `<pages>` bisa berupa path halaman relatif yang mengarah ke halaman di dalam subdirektori pada direktori yang berisi konfigurasi halaman. Sebagai contoh,

id="admin.Home" merujuk ke halaman Home di bawah direktori admin. Atribut id juga bisa berisi wildcard '\*' untuk menemukan seluruh halaman di bawah direktori yang ditetapkan. Sebagai contoh, id="admin.\*" merujuk ke seluruh halaman di bawah direktori admin dan sub-direktorinya. Peningkatan ini membolehkan para pengembang untuk memusatkan konfigurasi halamannya.

## **2.5 Kegiatan Perkuliahan**

Kegiatan perkuliahan adalah proses pembelajaran yang meliputi kegiatan tatap muka di kelas, penyelenggaraan percobaan dan pemberian tugas akademik lain. Untuk menunjang kegiatan perkuliahan tersebut dapat diadakan seminar, simposium, diskusi panel, lokakarya, dan kegiatan ilmiah lain. (Buku Pedoman UAJY, 2005).

Sistem perkuliahan pada Program Pascasarjana UAJY menggunakan sistem semester, dalam tiap semester mencakup 4 bulan. Sistem ini diambil atas pertimbangan untuk mempersingkat waktu studi tanpa mengurangi jumlah tatap muka tetapi dengan mengurangi libur antar semester. Tiap semester terdiri atas 16 minggu untuk kegiatan perkuliahan termasuk ujian tengah semester dan ujian akhir semester. Jumlah jam tatap muka untuk setiap SKS adalah 50 menit tiap minggu. Pada akhir semester diberi liburan satu minggu. (Buku Pedoman UAJY, 2005).

### **2.5.1 Pengisian Kartu Rencana Studi**

Seorang mahasiswa berhak mengikuti kegiatan perkuliahan apabila pada semester yang bersangkutan

sudah melakukan herregistrasi dan mendaftarkan diri dengan mengisi Kartu Rencana Studi (KRS). (Buku Pedoman UAJY, 2005).

### 2.5.2 Sistem Penilaian

Aspek yang dievaluasi/dinilai untuk setiap matakuliah meliputi (Buku Pedoman UAJY, 2005):

- Partisipasi kelas
- Pengerjaan tugas/kasus/proyek/presentasi
- Ujian tengah semester
- Ujian akhir semester

Penilaian prestasi mahasiswa dilakukan dengan menggunakan sistem huruf, yaitu: A, B, C, D, dan E, dengan sebutan sebagai berikut:

A = sangat baik

B = baik

C = cukup

D = kurang

E = gagal

T = tidak lengkap

Pada huruf-huruf tersebut dapat ditambahkan tanda "+" atau "-". Untuk menghitung indeks prestasi nilai-nilai huruf tersebut dikonversi sebagai berikut:

A = 4,0

A- = 3,7

B+ = 3,3

B = 3,0

B- = 2,7

C+ = 2,3

C = 2,0

C- = 1,7

D = 1,0

E = 0

Penilaian tidak lengkap (T) diberikan kepada mahasiswa yang belum menyelesaikan seluruh tugas/kasus/proyek yang merupakan aspek yang dievaluasi. Nilai tidak lengkap dapat diubah statusnya setelah mahasiswa menyelesaikan tugas/kasus/proyek tersebut dan menyerahkan kepada dosen yang bersangkutan melalui bagian administrasi. Apabila sampai batas waktu yang telah ditentukan tidak diselesaikan maka nilai tidak lengkap menjadi E.

Indeks prestasi (IP) adalah alat untuk mengukur kinerja mahasiswa untuk tiap semester ataupun secara kumulatif. IP dihitung dengan rumus:

$$IP = \frac{\sum_{i=1}^n (J_i \cdot N_i)}{\sum_{i=1}^n J_i}$$

Keterangan :

$J_i$  = jumlah sks matakuliah  $i$

$N_i$  = nilai matakuliah  $i$

**Gambar 2.4. Rumus Perhitungan Indeks Prestasi (Buku Pedoman UAJY, 2005)**

### 2.5.3 Laporan Prestasi dan Transkrip Akademik

Setiap akhir semester kepada mahasiswa diberikan Laporan Hasil Studi (LHS) yang berisi laporan prestasi mahasiswa pada semester yang bersangkutan serta prestasi kumulatif mahasiswa sampai dengan semester yang bersangkutan. (Buku Pedoman UAJY, 2005).