

## **BAB II**

### **LANDASAN TEORI**

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

Sistem Informasi dibutuhkan oleh suatu organisasi untuk mengelola data agar menghasilkan informasi yang dapat membantu proses pengambilan keputusan. Untuk memahami arti dari sistem informasi, kita hendaknya terlebih dulu memahami dua kata penyusunnya yaitu sistem dan informasi.

##### **2.1.1 Pengertian Sistem**

Menurut Hartini(2006), sistem dapat didefinisikan dengan dua buah sudut pandang. Yang pertama adalah melihat suatu sistem sebagai sebuah metode atau prosedur, dan yang kedua adalah melihat suatu sistem sebagai sekumpulan komponen. Sebagai sebuah prosedur, sistem didefinisikan sebagai jaringan kerja dari prosedur-prosedur yang berhubungan dan berkumpul bersama-sama untuk melakukan kegiatan untuk mencapai tujuan tertentu. Sebagai sekumpulan komponen, sistem didefinisikan sebagai kumpulan dari elemen-elemen atau objek-objek yang saling berinteraksi untuk mencapai tujuan tertentu.

##### **2.1.2 Karakteristik Sistem**

###### **1. Komponen Sistem**

Berupa subsistem atau bagian-bagian dari sistem yang menjalankan fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan sistem yang lebih besar.

## 2. Batas Sistem

Daerah yang membatasi antara suatu sistem dengan sistem yang lain.

## 3. Lingkungan Luar Sistem

Segala sesuatu di luar batas sistem yang mempengaruhi operasi sistem.

## 4. Penghubung Sistem

Media Penghubung antara satu subsistem dengan subsistem lainnya supaya dapat berintegrasi.

## 5. Masukan Sistem

Energi yang dimasukkan ke sistem.

## 6. Keluaran Sistem

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

## 7. Pengolah Sistem

Mengubah masukan menjadi keluaran.

## 8. Sasaran Sistem

Tujuan atau sasaran yang akan dicapai.

### **2.1.3 Pengertian Informasi**

Informasi didefinisikan sebagai data dalam konteks sedang digunakan atau dimanfaatkan atau data yang sudah matang dan bermanfaat bagi pengguna (Mudjihartono, 1998).

Informasi merupakan sesuatu yang nyata yang dapat mengurangi derajat ketidakpastian tentang suatu keadaan atau kejadian. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata, seperti tempat, benda, orang yang ada dan terjadi. Dengan demikian

informasi dapat pula dikatakan sebagai data yang telah dimanipulasi sehingga dapat berguna bagi seseorang.

#### **2.1.4 Kualitas Informasi**

Menurut Jogiyanto (1999), kualitas suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu:

##### 1. Informasi harus akurat

Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat berarti informasi harus jelas mencerminkan maksudnya. Komponen akurasi yaitu:

- a. *Completeness*, berarti informasi yang dihasilkan atau dibutuhkan harus memiliki kelengkapan yang baik, karena bila informasi yang dihasilkan sebagian-sebagian akan mempengaruhi dalam pengambilan keputusan.
- b. *Correctness*, berarti informasi yang dihasilkan atau dibutuhkan harus memiliki kebenaran.
- c. *Security*, berarti informasi yang dihasilkan atau dibutuhkan harus memiliki keamanan.

##### 2. Informasi harus tepat waktu

Hal ini berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan didalam pengambilan keputusan.

##### 3. Informasi harus relevan

Hal ini berarti informasi tersebut harus mempunyai manfaat bagi pengguna sistem.

Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan biaya mendapatkannya.

### **2.1.5 Pengertian Sistem Informasi**

Sistem Informasi adalah suatu sistem untuk mengatur, mengolah, dan mengelola data yang masuk ke dalam sistem sehingga akan menjadi suatu informasi yang berguna bagi pengguna sistem untuk suatu keperluan tertentu.

Menurut Jogiyanto (1991), sistem informasi adalah sekumpulan prosedur organisasi atau perusahaan yang pada saat dilaksanakan akan memberikan informasi bagi pemakai untuk mengambil keputusan atau untuk mengendalikan organisasi. Sedangkan kumpulan objek atau elemen yang berinteraksi satu sama lain untuk mencapai tujuan tertentu merupakan suatu pengertian dari sistem.

Menurut Robert A. Leitch dan K. Roscoe Davis, sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

### **2.1.6 Komponen Sistem Informasi**

Sistem informasi terdiri dari beberapa komponen yang saling berhubungan dan bergantung satu sama lain yaitu:

#### **1. *Brainware***

*Brainware* merupakan sumber data manusia atau orang yang terkait langsung dengan sistem informasi.

#### **2. *Software***

*Software* atau perangkat lunak merupakan bagian dari komputer yang sangat penting bagi kendali komputer.

Perangkat lunak dibagi menjadi 3 (tiga) kategori, yaitu:

- a. Perangkat lunak sistem operasi, misalnya: DOS, WINDOWS, LINUX, dan lain-lain.
- b. Perangkat lunak bahasa pemrograman, misalnya: Visual Basic, PHP, C#, Java, dan lain-lain.
- c. Perangkat lunak aplikasi, misalnya: Dbase, Lotus, Matlab, dan lain-lain.

### 3. *Hardware*

*Hardware* atau perangkat keras merupakan bagian dari komputer yang secara fisik terlihat. *Hardware* dibagi menjadi 4 (tiga) kategori, yaitu:

- a. Alat masukan (*Input Hardware*), misalnya: keyboard, scanner, mouse, digital camera, bar code reader, dan lain-lain.
- b. Alat pemrosesan (*Processing Hardware*), misalnya: processor.
- c. Alat keluaran (*Output Hardware*), misalnya: monitor, speaker, printer, plotterm, dan lain-lain.
- d. Alat penyimpanan (*Storage Hardware*), misalnya: disket, flashdisk, optical disk, multimedia card, dan lain - lain.

### 4. *Data*

*Data* merupakan fakta mentah yang diperlukan sebagai sumber daya masukan sistem dan akan diproses oleh sistem untuk menghasilkan output berupa informasi yang berguna.

### 5. *Network*

*Network* atau jaringan merupakan piranti yang digunakan sebagai sarana komunikasi yang menghubungkan semua komponen sistem informasi.

### 2.1.7 Siklus Hidup Pengembangan Sistem Informasi

Pengembangan sistem informasi harus melalui tahapan-tahapan tertentu karena sistem informasi membutuhkan banyak sumber daya yang kompleks. Proses pengembangan sistem melewati beberapa tahapan mulai dari sistem direncanakan sampai sistem tersebut diterapkan, dioperasikan, dan dipelihara. Bila sistem yang dikembangkan masih menimbulkan kembali permasalahan-permasalahan yang kritis serta tidak dapat ditangani dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ke tahap pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (*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 atau ide-ide dapat dimasukkan dalam tahap ini. Selanjutnya, baru diproses di tahap berikutnya yaitu tahap analisa 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. Langkah-langkah dalam tahap ini adalah sebagai berikut:

1. *Identity* (mengidentifikasi masalah)
2. *Understand* (memahami dari kerja sistem yang ada)
3. *Analyze* (menganalisis sistem)
4. *Report* (membuat laporan hasil analisa)

### 3. Perancangan Sistem (*System Design*)

Tahap ini menyangkut konfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga instalasi dari 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.

### **2.1.8 Sistem Informasi Berbasis Web**

Sistem informasi berbasis web atau *web based information system* adalah sistem informasi yang berjalan di jaringan internet dan basis datanya diletakan di *web server*. Pengguna dapat mengakses sistem informasi ini melalui halaman web dengan internet *browser*. Keunggulan sistem informasi berbasis web yaitu dapat diakses dari berbagai tempat di seluruh belahan dunia melalui komputer yang terhubung dengan jaringan internet.

### **2.2. Object Oriented Programming (OOP)**

Seiring perkembangan dunia pemrograman, metode pemrograman terstruktur mulai ditinggalkan karena dianggap kurang efisien. Kini banyak orang menggunakan metode pemrograman berorientasi objek yang lebih modular, praktis dan efisien.

#### **2.2.1 Konsep Dasar Object Oriented**

Pada bahasa pemrograman berorientasi objek, algoritma dan struktur data dibungkus bersama sebagai sebuah objek, yang mempunyai sekumpulan atribut-atribut dan *method-method* yang memiliki konsep-konsep dasar antara lain:

##### **1. Objek**

Objek adalah unit dari bahasa pemrograman yang memiliki dua karakteristik utama yaitu:

- a. Tiap objek memiliki atribut yang menyimpan data, dalam bahasa teknisnya atribut ini disebut sebagai variabel.

- b. Tiap objek memiliki fungsionalitas, dalam bahasa teknisnya fungsionalitas ini disebut sebagai *method*.

## 2. *Class*

*Class* adalah cetak biru yang mendefinisikan variabel-variabel dan *method-method* secara umum. Objek merupakan instansiasi dari *class*, dimana masing-masing objek memiliki nilai variabel yang mungkin berbeda.

Baik atribut maupun *method* dalam suatu *class* dapat memiliki salah satu *access modifier* untuk menentukan hak akses terhadap atribut atau *method* tersebut. *Access modifier* itu adalah:

- a. *Private*

Yang berarti atribut atau *method* hanya dapat diakses oleh *class* pemiliknya.

- b. *Protected*

Yang berarti atribut atau *method* dapat diakses oleh *class* pemilik dan *class-class* yang diturunkan dari *class* pemilik.

- c. *Friend*

Yang berarti atribut atau *method* dapat diakses oleh *class* pemilik, *class-class* yang diturunkan dari *class* pemilik, dan *class-class* lain yang berada dalam satu *package*.

- d. *Public*

Yang berarti semua atribut atau *method* dapat diakses oleh semua *class*.

### 2.2.3 *Use Case Diagram*

*Use Case Diagram* merupakan model yang digunakan untuk menggambarkan sistem secara logika tanpa

memperhatikan lingkungan fisik dimana user berinteraksi dengan sistem.

Simbol-simbol yang digunakan dalam *Use Case Diagram* adalah:

### 1. Use Case

*Use case* merupakan deskripsi fungsionalitas yang diberikan sistem kepada pemakai. *Use case* merupakan deskripsi sederetan aksi yang dilakukan sistem untuk mendapatkan hasil tertentu bagi aktor tertentu. Simbol *use case* adalah bentuk oval dengan nama *use case* berada di dalam atau di bawahnya.

Simbol untuk *use case* adalah sebagai berikut:



**Gambar 2.1. Notasi Use Case**

### 2. Aktor

Aktor merupakan orang atau sistem lain yang berhubungan langsung dengan sistem yang sedang dikembangkan. Aktor menyatakan peranan yang dimainkan *user* saat melakukan interaksi dengan sistem.

Simbol untuk aktor adalah sebagai berikut:



**Gambar 2.2. Notasi Aktor**

### 3. Asosiasi antara Aktor dan Use Case

Asosiasi antara aktor dengan *use case* berarti aktor akan menjalankan *use case* dan memperoleh hasil dari *use case* tersebut. Asosiasi antar aktor dan *use case* dapat digambarkan dengan garis, dengan anak

panah opsional pada ujung yang bersentuhan dengan *use case*, di mana anak panah opsional tersebut mengarah ke *use case*.

Asosiasi aktor dan *use case* dapat dilihat pada gambar berikut:



**Gambar 2.3 Asosiasi Aktor dan Use Case**

## 2.3 Basis Data

### 2.3.1 Teori Basis Data

Basis data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media dan tidak perlu suatu kerangkapan data dengan cara-cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali; dapat digunakan oleh satu atau lebih program aplikasi secara optimal; data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya; data disimpan sedemikian rupa sehingga penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*). Agar suatu perangkat lunak mudah untuk dikembangkan atau dirawat, maka perlu memperhatikan beberapa konsep dalam sebuah manajemen basis data.

Terdapat beberapa aturan yang harus dipatuhi pada file basis data agar dapat memenuhi kriteria sebagai

suatu basis data. Beberapa aturan itu berhubungan dengan:

1. Kerangkapan data  
yaitu munculnya data-data yang sama secara berulang-ulang pada file basis data yang semestinya tidak diperlukan.
2. Inkonsistensi data  
yaitu munculnya data yang tidak konsisten pada medan yang sama untuk beberapa file dengan kunci yang sama.
3. Data terisolasi  
disebabkan oleh pemakaian beberapa file basis data. Program aplikasi yang digunakan tidak dapat mengakses file tertentu dalam sistem basis data tersebut, kecuali bila program aplikasi diubah atau ditambah sehingga seolah-olah ada file yang terpisah atau terisolasi terhadap file yang lain.
4. Keamanan data  
berhubungan dengan masalah keamanan data dalam sistem basis data. Pada prinsipnya file basis data hanya boleh digunakan oleh pemakai tertentu yang mempunyai wewenang untuk mengakses.
5. Integritas data  
berhubungan dengan unjuk kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh.

### **2.3.2 Structured Query Language (SQL)**

*Structured Query Language* (SQL) adalah bahasa yang dirancang secara khusus untuk berkomunikasi dengan

basis data. SQL diciptakan dengan sedikit kata dan dirancang untuk melakukan suatu hal dan melakukannya dengan baik serta memberikan efisiensi kepada pengguna untuk membaca dan menulis data dari suatu basis data.

SQL dapat digunakan dengan dua cara, yaitu menggunakan SQL secara interaktif, dimana pengguna memasukkan perintah SQL pada sebuah komputer, atau menyimpan atau memasukkan perintah SQL ke dalam bahasa procedural.

Perintah-perintah SQL yang sering digunakan antara lain:

1. **SELECT**, untuk menyatakan field, konstanta dan ekspresi yang akan ditampilkan sebagai hasil query.
2. **FROM**, berisikan tabel-tabel yang berisi data yang akan ditampilkan.
3. **WHERE**, untuk menyatakan syarat dari hasil query
4. **ORDER BY**, untuk mengurutkan hasil query.
5. **INSERT**, untuk menambah baris tunggal pada tabel.
6. **DELETE**, untuk menghapus satu atau lebih baris dari tabel.
7. **UPDATE**, untuk meng-update atau mengubah satu atau lebih baris didalam tabel.
8. **DISTINCT**, untuk membuang baris yang duplikasi dan dipakai sekali untuk sebuah klausa **SELECT**.
9. **COUNT()**, untuk menghasilkan nomor baris pada sebuah kolom.
10. **MAX()**, untuk menghasilkan nilai tertinggi dari kolom.
11. **SUM()**, untuk menghasilkan jumlah nilai sebuah kolom.

## **2.4 PRADO Framework**

### **2.4.1 Pengenalan PRADO**

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

Tujuan utama dari PRADO adalah menggunakan reusabilitas secara maksimum dalam pemrograman *web* (Xue dan Zhuo, 2008). Dengan adanya reusabilitas, pengembang tidak hanya menggunakan ulang kode yang dimiliki seseorang, tetapi juga menggunakan ulang kode orang lain dalam cara yang mudah sehingga menghemat usaha penciptaan dan waktu pengembangan program. Untuk mencapai tujuan tersebut, 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 dari 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*. Pemrograman kendali *event* membantu para pengembang

untuk lebih berfokus pada logika yang diperlukan dan mengurangi secara drastis pengkodean berulang kali.

Ringkasnya, pengembangan aplikasi *web* menggunakan PRADO *Framework* sebagian besar menyangkut penurunan tipe komponen yang sudah dibuat sebelumnya, mengkonfigurasinya dengan mengeset 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 diambil 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. 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 *open source*. Segera setelah itu, situs proyek *xisc.com* diumumkan kepada publik. Dengan dukungan fantastis dari tim pengembang dan pengguna PRADO, PRADO berkembang ke versi 2.0 pada pertengahan tahun 2005. Dalam versi ini, Wei Zhuo berkontribusi ke PRADO dengan dukungan I18N dan L10N yang istimewa. Pada bulan Mei 2005, kerangka kerja PRADO ditulis ulang untuk memecahkan beberapa isu dasar dalam versi 2.0 dan untuk menangkap beberapa fitur dalam Microsoft ASP .NET 2.0. Setelah hampir setahun

kerja keras dengan lebih dari 50000 baris kode baru, PRADO 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 tanggal 11 Januari 2008.

#### **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 dan Zhuo, 2008):

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 dapat 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 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, dan lain-lain.

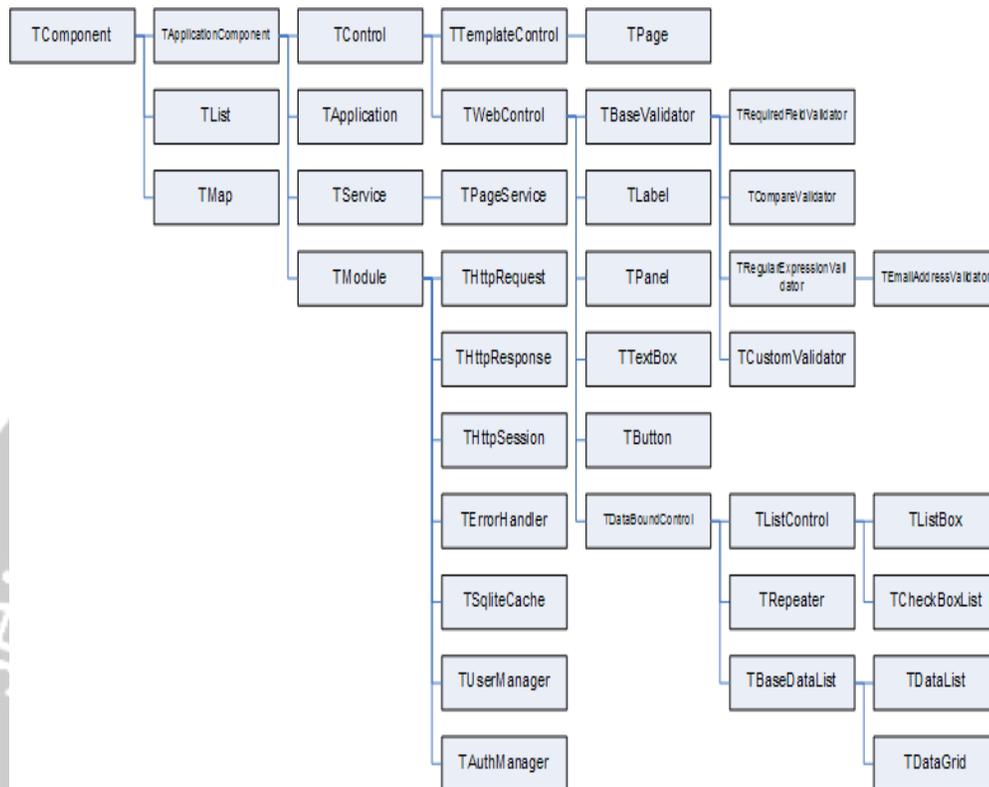
Keunikan PRADO *Framework* terletak pada paradigma pemrograman berbasis-komponen dan kendali-event (*component-based and event-driven programming paradigm*). Meskipun paradigma pemrograman ini bukan hal

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*) yang terfokus pada pembuatan pemrograman Web yang banyak berkaitan dengan interaksi pengguna. Kerangka kerja PRADO 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 2.4. Arsitektur Kelas Utama pada PRADO**  
(Xue dan Zhuo, 2008)

#### 2.4.6 Halaman (Page) pada PRADO

Halaman (*page*) merupakan salah satu kontrol 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.

#### 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, dan lain-lain. 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), dan lain-lain. 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 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`.

## 2.5 SQL Server 2005

SQL Server 2005 merupakan sistem basis data relasional yang diproduksi oleh Microsoft. SQL Server banyak digunakan dalam sistem yang berskala kecil maupun besar. Pada bulan Oktober 2005, Microsoft mengeluarkan SQL Server 2005 sebagai penyempurnaan dari SQL Server 2000. Keunggulan yang dimiliki oleh SQL Server 2005 antara lain:

### 1. Skalabilitas dan Ketersediaan

SQL Server 2005 dapat digunakan dalam berbagai platform. Selain itu, SQL Server 2005 juga dilengkapi dengan fasilitas seperti penggabungan server, integrasi data untuk pelaporan manual, dan kapasitas memori yang besar yang berguna untuk meningkatkan level performansi yang diperlukan.

### 2. Keunggulan Basis Data Tingkat Enterprise

SQL Server 2005 sebagai basis data relasional mendukung permintaan pengolahan data yang lebih cepat dan efisien untuk mendukung kompleksitas permintaan data pada tingkat enterprise.

### 3. Kemudahan Instalasi dan Packages.

Fitur ini dimiliki oleh SQL Server 2005 karena sehingga memberikan kemudahan yang diharapkan dapat mempercepat proses pembuatan aplikasi.

### 4. Data Warehousing

Dalam pengolahan data, SQL Server 2005 dapat digunakan dalam menganalisa suatu aliran data yang sedang aktif.