Jurnal Informatika, Vol. 12, No. 2, November 2014, 63-69 DOI: 10.9744/informatika.12.2.63-69
ISSN 1411-0105

DATA MODEL CUSTOMIZATION FOR YII BASED ERP APPLICATION

Andre Leander[1], Adi Wibowo[1*], Lily Puspa Dewi[1]
[1] Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121–131, Surabaya 60236. Telp. (031) 2983455, Fax. (031) 8417658
E-mail: leeshienwen@gmail.com, adiw@petra.ac.id, lily@petra.ac.id
*Korespondensi penulis

Abstrak: Semakin berkembangnya bisnis perusahaan UD. Logam Utama, memicu kebutuhan akan informasi secara cepat dan akurat guna meningkatkan kinerja, efisiensi, kontrol dan *value* perusahaan. Perusahaan membutuhkan sebuah sistem dimana setiap fungsional perusahaan saling terintegrasi. ERP merupakan aplikasi yang memiliki sistem yang mampu memusatkan *database* dan mampu dikonfigurasi sesuai dengan proses bisnis perusahaan. Sebelum memulai pengembangan aplikasi, dilakukan analisa dan desain terhadap proses bisnis perusahaan. Tahap desain menghasilkan sejumlah pemodelan yang nantinya digunakan untuk membuat aplikasi. Hasil akhir dari pengembangan aplikasi ini adalah sebuah aplikasi ERP yang mampu dikonfigurasi dengan proses bisnis perusahaan. ERP yang dikembangkan mencakup modul pergudangan dan produksi, modul pembelian, modul penjualan, dan modul akuntansi.

Kata kunci: *Enterprise Resource Planning (ERP), Yii Framework, Business Architecture, Technology Architecture, Software, Agricultural Company.*

Abstract: *As UD. Logam Utama's business grow, trigger the need of fast and accurate information in order to improve performance, efficiency, control and company's values. The company needs a system that can integrate each functional area. ERP has centralized database and able to be configured, according to company's business processes.First phase of application development is analysis and design the company's business processes. The design phase produce a number of models that will be used to created application.The final result of application development is an ERP application that can be configured with the company's business process. The ERP application consist of warehouse or production module, purchasing module, sales module, and accounting module.*

Keywords*: Enterprise Resource Planning (ERP), Yii Framework, Business Architecture, Technology Architecture, Software, Agricultural Company.*

PENDAHULUAN

Perkembangan teknologi yang sangat pesat dewasa ini memberikan pengaruh yang besar dalam bidang perekonomian terutama terhadap sejumlah perusahaan dalam persaingannya di dunia bisnis. Penerapan sistem informasi merupakan salah satu langkah yang digunakan untuk membantu kinerja perusahaan agar menjadi lebih baik, lebih efisien, dan lebih terkontrol dalam hal mengurangi kesalahan ketika menjalankan bisnis. Hal ini sangat penting karena dengan penerapan sistem informasi, perusahaan dapat meningkatkan value kepada pembeli dan menjadi dukungan bagi perusahaan untuk bersaing dengan perusahaan lainnya.

UD Logam Utama merupakan sebuah perusahaan yang mengelola hasil bumi terutama hasil pertanian di Luwuk, Sulawesi Tengah. Dalam menjalankan proses bisnis operasionalnya, perusahaan masih melakukannya secara manual. Hal ini menyebabkan pekerjaan menjadi lambat dan kurangnya kontrol menyebabkan proses pencatatan dan perhitungan sangat rentan terjadi kesalahan. Sebagai contoh sering kali perusahaan mengalami kesulitan dalam mengurusi para penjual yang sangat banyak dan membuat penjual tersebut menunggu terlalu lama untuk proses pembayaran. Terdapat juga kejadian dimana perusahaan salah menghitung nota sehingga pihak penjual merasa dirugikan.

Selain itu tidak adanya konsistensi terhadap data terbaru dan lambatnya aliran informasi mengakibatkan keterlambatan pengambilan keputusan atau juga bisa membuat keputusan yang salah.

TINJAUAN PUSTAKA

Enterprise Resource Planning

Enterprise Resource Planning merupakan sistem perusahaan dimana antara satu sistem fungsional dengan sistem fungsional yang lain saling berkaitan dan bekerja sama. Hal ini dimungkinkan

dengan menggunakan *software* yang mampu mengintegrasikan semua modul di masing-masing fungsional/divisi di dalam sebuah perusahaan [1]. *Software* membantu perusahaan untuk mengelola dan menampilkan informasi secara real time antar divisi yang berbeda dengan menggunakan database tertentu. Selain itu, *software* ERP haruslah mampu untuk dikonfigurasi sehingga dapat digunakan diberbagai komputer hardware dan *network* yang berbeda. *Software* ERP sangatlah mahal dan sangat kompleks oleh karena itu *software* ini memang dikhususkan untuk perusahaan [1].

Dalam penerapan ERP bisa saja tidak meningkatkan produktivitas dan keuntungan kompetitif apabila penerapan sistem tidak dijalankan dengan baik [2]. Penerapan ERP di dalam perusahaan memiliki siklus hidup yang disebut sebagai Siklus Hidup Pengembangan Sistem ERP Fase yaitu fase perencanaan, fase analisis, fase desain, fase implementasi, dan fase dukungan teknis [3]. Terdapat delapan faktor yang dapat mempengaruhi implementasi ERP yaitu: data-data yang diberikan (*Data Provided*), sistem parallel (Parallel Systems), pelatihan dan pengujian (Training and Testing), harapan dari sistem ERP (Expectation from the ERP System), kualitas pekerja (Employee Retention), desain dan pengujian (Design and Testing), kustomisasi ERP harus kurang dari 30%, dan semua *stakeholder* harus diidentifikasi pada tahap awal pengembangan termasuk pelanggan dan vendor [4].

Organizational Data, Master Data, and Transaction Data

Organizational data digunakan untuk merepresentasikan struktur dari sebuah perusahaan. Contoh struktur dari perusahaan seperti perusahaan, divisi, pabrik, gudang, area penyimpanan, dan area penjualan.

Master data merupakan representasi entitas terkait dengan berbagai macam proses. Beberapa entitas yang termasuk master data di dalam sistem ERP antara lain data *supplier/vendor*, data *customer*, dan data material/produk. Data material/produk merupakan master data yang paling umum dan paling banyak digunakan di berbagai macam proses seperti proses pembelian, proses penjualan, dan proses produksi. Oleh karena itu data material menjadi data yang paling kompleks dan luas dibandingkan master data yang lain. Data material di dalam sistem ERP tersusun *atas basic data*, *material group*, *material type*, dan data terkait dengan proses tertentu (*purchasing data*, *sales data*, *warehouse data*, *management accounting data*, dan lain-lain) [5].

Transaction data mencerminkan hasil dari pengeksekusian proses atau transaksi. Contoh t*ransaction* data antara lain tanggal, jumlah, harga, dan pembayaran serta waktu pengiriman. Sehingga *transaction data* merupakan kombinasi dari *organizational data*, *master data*, dan *situational data*. *Situational data* adalah data yang sifatnya spesifik terhadap kondisi pengeksekusian proses, contoh siapa (*who*), apa (*what*), kapan (*when*), dan di mana (*where*) [5].

XML

eXtensible Markup Language atau XML merupakan sebuah dokumen yang didesain dan dibuat untuk menyusun data, sebagai media pengirim, dan penyimpan data. Berbeda dengan HTML dimana memiliki tag yang sudah ditentukan, dalam XML anda bisa mendefinisikan *tag* anda sendiri. Contoh isi dari dokumen XML dapat dilihat pada Gambar 2.39.

Yii Framework

Yii ("Yes It Is!") merupakan sebuah *framework open source* yang disusun dengan menggunakan PHP5 untuk pengembangan aplikasi berbasis web. Yii sering digunakan untuk pengembangan sebuah aplikasi yang sifatnya *rapid development* dan juga sangat membantu untuk menghasilkan sebuah aplikasi yang *extremely efficient*, *extensible*, dan *maintainable*.

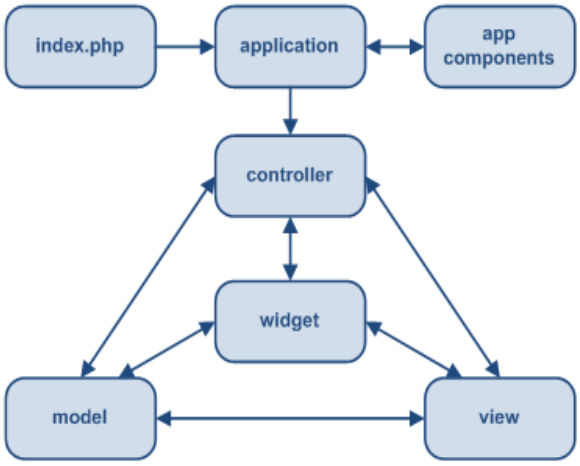
Model-View-Controller

Yii mengimplementasi pola desain *model, view,* dan *controller* (MVC) dimana telah diadobsi secara luas oleh sejumlah pemrograman Web. MVC bertujuan untuk memisahkan logika bisnis (*business logic*) dari tampilan aplikasi (*user interface*), sehingga para pengembang aplikasi dapat lebih mudah untuk mengubah masing-masing bagian tanpa mempengaruhi yang lain. Dalam MVC, *model* menggambarkan informasi (data) dan aturan (*business rules*). *View* berisikan elem-elemn yang digunakan untuk membentuk tampilan aplikasi (*user interface*) seperti teks atau kolom inputan. Sedangkan *controller* berfungsi untuk memanajemen komunikasi antara *model* dan *view*. Pada Gambar 1 mengilustrasikan struktur statis dari sebuah aplikasi Yii.

Gii

Yii dilengkapi dengan dengan sebuah *Web-based code generator tool* yang disebut Gii. Gii merupakan sebuah *tool* yang berfungsi untuk

membuat *model*, *controller*, *form*, dan *view* melalui tampilan aplikasi, sehingga pengembang tidak perlu membuat MVC secara manual. Gii dapat di kembangkan secara bebas oleh pengguna Yii seperti mengubah isi dari *script* yang dihasilkan dengan menggunakan *generator* dasar Yii atau dengan membuat *generator* yang baru.



Gambar 1. Struktur Statis dari Aplikasi Yii

ANALISA DAN DESAIN

Fitur kustomisasi yang dimiliki oleh aplikasi ERP didesain untuk menggunakan *file* XML sebagai media penyimpanan data kustomisasi dan pola struktur data. Pada tahap ini perlu dilakukan analisa terhadap struktur data yaitu dengan mendaftarkan data-data apa saja yang dibutuhkan, *rules* untuk data-data tersebut, dan bagaimana data dapat digunakan ketika fitur kustomisasi dijalankan. Struktur data tersebut disimpan di dalam XML *schema* dan digunakan ketika membuat file XML yang menyimpan data kustomisasi. File XML yaitu instance dari XML Schema selanjutnya digunakan pada *table generator*, *model generator*, dan *form generator*.

Desain XML Schema

XML *schema* merupakan sebuah *file* XML yang digunakan sebagai panduan penulisan *Instance* XML. XML *table extension* adalah sebuah *file Instance* XML dimana aplikasi membaca isi dari XML tersebut dan membuatkan *table* baru di *database* sesuai dengan isi dari XML. Isi dari XML *schema* dapat dilihat pada Gambar 2.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="table">
        <xs:complexType>
        <xs:sequence>
             <xs:element name="table_name" type="xs:string"/>
             <xs:element name="field" minOccurs="1"
                              maxOccurs="unbounded">
     <xs:complexType>
                  <xs:sequence>
                  <xs:element name="field_name"
                              type="xs:string"/>
                  <xs:element name="data_type">
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                           <xs:enumeration value="varchar"/>
                           <xs:enumeration value="text" />
                           <xs:enumeration value="date"/>
                           <xs:enumeration value="timestamp"/>
                           <xs:enumeration value="smallint"/>
                           <xs:enumeration value="integer"/>
                           <xs:enumeration value="bigint"/>
                           <xs:enumeration value="numeric"/>
                           <xs:enumeration value="boolean"/>
                           <xs:enumeration value="serial"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="length" minOccurs="0"
                  type="xs:integer"/>
                  <xs:element name="scale" minOccurs="0"
                  type="xs:integer"/>
                  <xs:element name="comment" type="xs:string"/>
                  <xs:element name="type" type="xs:string"/>
                  <xs:elememt name="default_value"
                  type="xs:string"/>
                  <xs:element name="null_able"
                  type="xs:boolean"/>
                  <xs:element name="array_value" minOccurs="0"
                  maxOccurs="unbounded">
                  <xs:complexType>
                <xs:sequence>
                <xs:element name="key" type="xs:string"/>
                <xs:element name="value" type="xs:string"/>
                     </xs:sequence>
                </xs:complexType>
                </xs:element>
              <xs:element name="table_value" minOccurs="0"
              maxOccurs="1">
              <xs:complexType>
              <xs:sequence>
              <xs:element name="table_name" type="xs:string"/>
              <xs:element name="key_field" type="xs:string"/>
              <xs:element name="value_field" type="xs:string"/>
                  </xs:sequence>
              </xs:complexType>
                </xs:element>
              </xs:sequence>
              </xs:complexType>
              </xs:element>
              </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Gambar 2. XML Schema

Penjelasan setiap elemen yang terdapat pada Gambar 2 dapat dilihat pada Tabel 1.

Tabel 1. Tabel Penjelasan Elemen-Elemen XML dari XML*Schema*

No	Nama Elemen	Keterangan
1	table_name	Menyimpan nama *table exstension*
2	field	Menyimpan daftar *attribute* atau *field* yang dimiliki oleh *table extension*
3	field_name	Menyimpan nama *attribute* atau *field*
4	data_type	Menyimpan tipe data dari *attribute* atau *field*
5	length	Menyimpan panjang dari *attribute* atau *field*
6	scale	Menyimpan nilai belakang koma dari *attribute* atau *field* yang memiliki tipe data *numeric, decimal, money,* atau *serial*
7	comment	Menyimpan komentar untuk *attribute* atau *field*
8	type	Menyimpan jenis dari *attribute* atau *field* seperti *Primary Key, Foreign Key,* dan *Normal Field*
9	default_value	Menyimpan nilai awal dari *attribute* atau *field* apabila tidak memiliki nilai
10	null_able	Menyimpan pengaturan apakah *attribute* atau *field* boleh dibiarkan kosong atau tidak.
11	array_value	Menyimpan daftar pilihan nilai *attribute* atau *field*
12	key	Menyimpan nilai dari array_value
13	value	Menyimpan data yang ditampilkan oleh key array_value
14	table_value	Menyimpan daftar pilihan nilai dimana nilai berasal dari *table* lain.
15	table_name	Menyimoan nama table yang menjadi table_value
16	key_field	Menyimpan *attribute* atau *field* kunci dari *table* yang menjadi table_value
17	value_field	Menyimpan *attribute* atau *field* dari table_value yang menjadi referensi nilai

Desain Template Class Model, Template Form, dan *Template View*

Template class model, template form, dan *template view* adalah *file-file* PHP yang berisikan *script code* yang digunakan sebagai pola dasar pembuatan *file model, form,* dan *view extension*. Pada Gambar 3, Gambar 4, dan Gambar 5 merupakan desain *template class model*, *template form,* dan *template view*.

```
class <?php echo $modelClass; ?> extends <?php echo $this->baseClass."\n"; ?>
{
    public function tableName()
    {
        return '<?php echo $tableName; ?>';
    }

    public function rules()
    {
        return array(
                <?php foreach($rules as $rule): ?>
                    <?php echo $rule.",\n"; ?>
                <?php endforeach; ?>
    array('<?php echo implode(', ', array_keys($columns)); ?>', 'safe', 'on'=>'search'),
        );
    }

    public function relations()
    {
        return array(
                <?php foreach($relations as $name=>$relation): ?>
                    <?php echo "'$name' => $relation,\n"; ?>
                <?php endforeach; ?>
        );
    }

    public function attributeLabels()
    {
        return array(
        <?php foreach($labels as $name=>$label): ?>
                <?php echo "'$name' => '$label',\n"; ?>
        <?php endforeach; ?>
        );
    }

    public function search()
    {
        $criteria=new CDbCriteria;
        <?php
        foreach($columns as $name=>$column)
        {
            if($column->type==='string')
                echo "\t\t\$criteria->compare('$name',\$this->$name,true);\n";
            else
                echo "\t\t\$criteria->compare('$name',\$this->$name);\n";
        }
        ?>
        return new CActiveDataProvider($this, array(
                'criteria'=>$criteria,
        ));
    }
```

Gambar 3. *Template Class Model*

```
<?php echo "<?php\n"; ?>
/* @var $this <?php echo $this->getModelClass(); ?>Controller */
/* @var $model <?php echo $this->getModelClass(); ?> */
/* @var $form CActiveForm */
?>

<?php
    $class = $this->getModelClass();
    $model = new $class;
    $tableName = $model->tableName();
    $file = simplexml_load_file(yii::app()-
            >basePath."/data/ERPGEnerator/".$tableName.".xml");
    $attributes = $this->getModelAttributes();
?>

<?php foreach($file->field as $i=>$attribute): ?>
<div class="form-group ">
  <?php echo "<?php echo \$form->labelEx($class::model(),'".strtolower($attribute-
        >field_name)."',array('class'=>'col-md-3 col-lg-3 control-label','for'=>'')); ?>\n"; ?>
  <div class="col-md-9 col-lg-9">

<?php
        if($attribute->component=="textField"){
          echo "<?php echo \$form->textField(
          $class::model(),'".strtolower($attribute->field_name)."'
          ,array('class'=>'form-control')); ?>\n";
          echo "<?php echo \
          $form->error($class::model(),'".strtolower($attribute->field_name)."',
          array('class'=>'badge pull-left')); ?>\n";
          }
          else if($attribute->component=="textArea")
          {
?>
…. /* another code */
```

Gambar 4. *Template File Form*

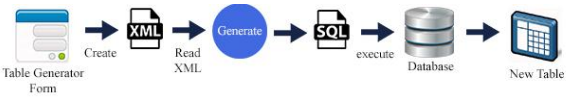
```
<div class="col-lg-6 col-md-6 col-sm-12 col-xs-12">
<table class="table table-responsive table-hover table-condensed table-bordered">
<tbody>
<?php
$fields = $this->getXMLTableField();
foreach($fields as $field): ?>
    <tr>
    <td class="success" width="30%">
        <?php
        echo "<?php echo CHtml::encode(\$ext->getAttributeLabel('".strtolower((string)
        $field->field_name)."')); ?>";
        ?>
    </td>
         <?php if($this->checkForeignKey(strtolower((string)$field->field_name)))
         {
                ?>
    <td>
        <?php
        echo "<?php echo CHtml::encode(\$ext->".
        $this->getRelationNameAttribute(strtolower((string)
        $field->field_name))."->".$field->table_value->value_field."); ?>"; ?></td>
        <?php
              }
              else
              {
         ?>
     <td>
        <?php
                echo "<?php echo CHtml::encode(\$ext->".
                strtolower((string)$field->field_name)."); ?>"; ?>
        </td>
        <?php
                }
         ?>
       </tr>
            <?php endforeach; ?>
    </tbody>
    </table>
</div>
```

Gambar 5. *Template File View*

Cara Kerja ERP Generator

ERP *generator* terbagi atas tiga bagian yaitu *table generator*, *model generator*, dan *form generator*. *Table generator* berfungsi untuk membuat *table* baru di *database* yang merupakan ekstensi dari *table* dasar yang telah tersedia. *Table* baru tersebut kemudian dibuat *class model* dengan menggunakan *model generator*. Model yang dihasilkan dari *model generator* kemudian dipakai untuk membuat *file form* tampilan untuk mengisi data dan *file view* untuk menampilkan data dengan menggunakan *form generator*.

Table generator bekerja dengan pertama-tama membuat *file* XML *table extension*. Isi dari *file* XML disesuaikan dengan data yang diberikan oleh pengguna. Dari *file* XML *table extension* tersebut kemudian dibuat *table* baru di database dengan mengeksekusi perintah SQL secara otomatis oleh aplikasi. Sebelum membuat *table* baru, nama *table* tersebut diperiksa terlebih dahulu apakah telah dipakai atau tidak terpakai. Ilustrasi cara kerja *table generator* dapat dilihat pada Gambar 6.

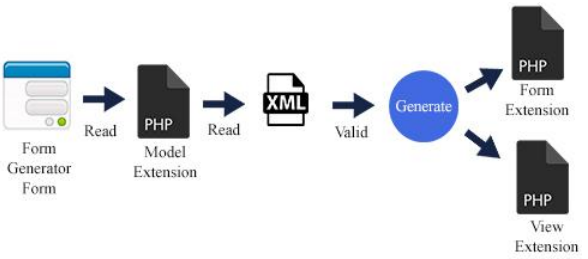


Gambar 6. Ilustrasi cara kerja *table generator*

Model generator bekerja dengan membaca *file* XML *table extension* dan *table extension* yang ada di dalam *database*. *Model generator* menggunakan *template class model* sebagai dasar untuk membuat model baru. *Model generator* membaca nama *table parent*, nama *table* ekstensi, daftar *attributes* dan nama *label attributes*, daftar relasi, dan *rule attributes* dari *file* XML *table extension* dan *table extension* di *database*, kemudian menggunakannya pada *template class model*. Setelah membaca data, *model generator* membuat *file* baru pada *folder* model di modul yang dipilih oleh pengguna. Ilustrasi cara kerja *model generator* dapat dilihat pada Gambar 7.

Gambar 7. Ilustrasi cara kerja *model generator*

Form generator bekerja dengan membaca model ektensi dan membuat dua file baru yaitu *file form* dan *file view*. *Form generator* menggunakan panduan pembuatan yaitu dengan *template form*. *Form generator* terlebih dahulu membaca isi dari *class model* dan membuat daftar komponen yang disesuaikan dengan daftar komponen yang tersimpan file XML *table extension*. Dengan menggunakan fungsi yang dimiliki *form generator*, aplikasi membuat 2 *file* baru dengan lokasi penyimpanan pada *folder view* dari modul yang dipilih pengguna. Isi dari 2 *file* baru tersebut disesuaikan dengan isi template *form* dan *view*. Ilustrasi cara kerja *form generator* dapat dilihat pada Gambar 8.



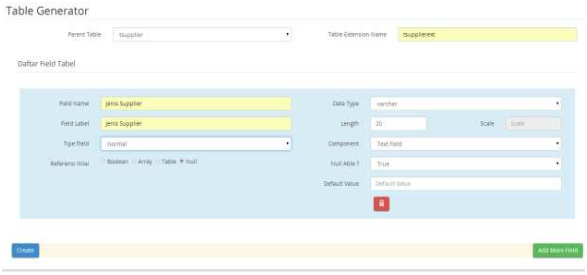
Gambar 8. Ilustrasi cara kerja *form generator*

HASIL

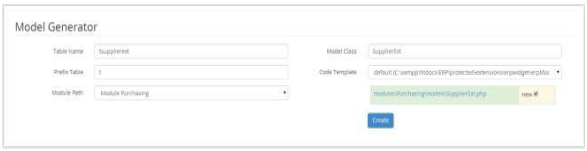
Hasil aplikasi berupa program ERP yang dapat dikonfigurasi. Konfigurasi meliputi penambahan *table ekstensi* dari table yang sudah ada sebelumnya. Selain itu konfigurasi meliputi pembuatan *class model* baru dan *file view* sebagai *form* dari *model*. Tampilan *form* yang digunakan untuk membuat *table* baru dapat dilihat pada Gambar 9.

Selain itu juga terdapat *form* untuk membuat *model* baru sesuai dengan *table* yang dibuat sebelumnya. Tampilan *form model generator* dapat dilihat pada Gambar 10.

Hasil eksekusi dari *model generator* pada Gambar 10 menghasilkan *file model extension* baru dimana disimpan pada *folder* khusus (ext) pada modul yang dipilih pengguna. Isi dari file model extension dapat dilihat pada Gambar 11.



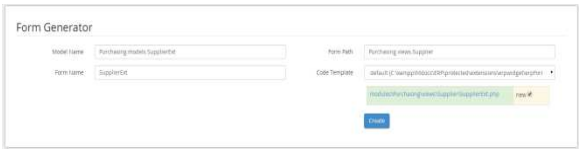
Gambar 9. *Form Table Generator*



Gambar 10. Tampilan *Model Generator*

```
class SupplierExt extends CActiveRecord
{
  public function tableName()
  {
      return 'tsupplierext';
  }

  public function rules()
  {
      return array(
            array('id_supplier,
                  jenis_supplier','default',
                 'setOnEmpty' => true,'value'=>null),
            array('id_supplier',
                   'numerical', 'integerOnly'=>true),
            array('jenis_supplier',
                   'length', 'max'=>20),
            array('id_supplier,
                   jenis_supplier', 'safe', 'on'=>'search'),
      );
  }

  public function relations()
  {
      return array(
            'Supplier' =>
             array(self::BELONGS_TO, 'Supplier', 'id_supplier'),
      );
  }

  public function attributeLabels()
  {
      return array(
            'id_supplier' => 'Supplier',
            'jenis_supplier' => 'Jenis Supplier',
      );
  }

  public function search()
  {
      $criteria=new CDbCriteria;

      $criteria->compare('id_supplier',$this->id_supplier);
      $criteria->compare('jenis_supplier',
      $this->jenis_supplier,true);

      return new CActiveDataProvider($this, array(
            'criteria'=>$criteria,
      ));
  }

  public static function model($className=__CLASS__)
  {
      return parent::model($className);
  }
}
```

Gambar 12. Hasil *Form Generator*



Gambar 13. Tampilan *Form Generator*

```
<?php
/* @var $this SupplierExtController */
/* @var $model SupplierExt */
/* @var $form CActiveForm */
?>


<div class="form-group ">
    <?php echo
$form->labelEx(SupplierExt::model(),'jenis_supplier'
,array('class'=>'col-md-3 col-lg-3 control-label','for'=>'')); ?>

    <div class="col-md-9 col-lg-9">
      <?php echo $form->textField(SupplierExt::model(),
'jenis_supplier',array('class'=>'form-control')); ?>

<?php echo $form->error(SupplierExt::model(),
'jenis_supplier',array('class'=>'badge pull-left')); ?>

    </div>
 </div>
```

Gambar 14. *File Form* Hasil *Form Generator*

```
<div class="col-lg-6 col-md-6 col-sm-12 col-xs-12">
<table class="table table-responsive table-hover table-condensed
table-bordered">
  <tbody>
   <tr>
      <td class="success" width="30%">
      <?php echo CHtml::encode($ext->getAttributeLabel
      ('jenis_supplier')); ?></td>
      <td>
      <?php echo CHtml::encode($ext->jenis_supplier); ?>
      </td>
   </tr>
  </tbody></table>
 </div>
```

Gambar 15. *File Form* Hasil *Form Generator*

File model extension yang dihasilkan *model generator* digunakan pada *form generator* sebagai dasar pembuatan *file form* dan *file* view. Tampilan *form generator* dapat dilihat pada dan Gambar 13 dan hasil dari *form generator* menghasilkan *file form* dan *file view* yang isinya dapat dilihat pada Gambar 14 dan Gambar 15.

KESIMPULAN

Dari hasil perancangan dan pembuatan aplikasi ERP yang diterapkan pada perusahaan UD. Logam utama, dapat diambil kesimpulan antara lain:

- Semua fungsional perusahaan mampu diintegrasian ke dalam satu aplikasi ERP dan satu database terpusat. Hal ini menghasilkan penyebaran informasi yang cepat, efisien dan terkontrol.
- Aplikasi yang dikembangkan ERP menyediakan sejumlah besar fungsi yang mampu menangani proses bisnis yang dimiliki perusahaan. Fungsi-fungsi yang sediakan mampu menyelesaikan permasalahan dan memenuhi kebutuhan perusahaan di setiap aktivitas bisnis perusahaan.

DAFTAR PUSTAKA

[1] Monk, E. B. *Concepts in Enterprise Resource Planning 4th Edition*. United States of America: Course Technology, Cengage Learning. 2013.

[2] Tenkorang, R. A., Helo. P. *Enterprise Resource Planning* (ERP): *A Review Literature Report*. 2011.

[3] Yanis, F. *Pentingnya Sistem Enterprise Resource Planning (ERP) Dalam Rangka untuk Membangun Sumber Data pada Suatu Perusahaan*. 2013.

[4] Dixit. A. Kr., Prakash, O. *A Study of Issues Affecting ERP Implementation in SMEs*. 2011.

[5] Magal, S. & Word, J. *Integrated Business Process with ERP Systems*. United State: John Wiley & Sons, Inc. 2012.