

# Create Custom Modul in Odoo with Python

Rochana P H

Typical case, modul development dalam Odoo adalah memodifikasi atau meng-extend modul yg sudah tersedia.

Yang akan kita coba,

- membuat modul baru dengan dependency sesedikit mungkin
- modul bukan hasil modifikasi atau extend dari modul yg sudah ada

# Modul Basic Structure

`__init__.py` → python importable

`__manifest__.py` → descriptor file

`model.py`

`view.xml`

# Membuat aplikasi To-Do Task

Buku: Odoo 10 Development Essentials PACKT

# Aktivasi mode developer Odoo

Masuk ke dev.mode:

Settings → Activate the developer mode (bagian kanan bawah) → Apps

# Buat folder modul aplikasi dalam directory addons

Buat folder *todo\_app*

Catatan:

Nama folder merupakan nama teknis (warna merah pada Odoo dev. mode)

Nama teknis → harus valid diidentifikasi Python

- Dimulai dengan huruf alfabet
- Hanya boleh memuat alfabet, angka, dan underscore

Sementara, buat file `__init__.py`  
kosong

# 1. Init file

Buat file, `__manifest__.py`

## 2.Descriptor File



# Descriptor file

Harus memuat sebuah Python dictionary dengan beberapa key descriptor yg memungkinkan. Minimal yg dibutuhkan ada key *name*.

```
{  
    'name': 'To-Do Application',  
    'description': 'Manage your personal To-Do  
tasks.',  
    'author': 'Daniel Reis',  
    'depends': ['base'],  
    'application': True,  
}
```

Key *depends* valuenya list dari dependency modul2 yg dibutuhkan.

Kalau modul ini diinstall → yg di dalam depends juga diinstall.

Kalau modul yg dibuat tidak memerlukan modul lain, isikan value dengan *base*.

# Cek list aplikasi

- Simpan file.
- Update list apps Odoo.

Apps → Update Apps List → Reload

- Cek apakah aplikasi berhasil dibuat, ada di daftar list apps
- Install.

Action ▼



## To-Do Application

By Rochana P H

Upgrade

Uninstall

Information

Technical Data

Installed Features

Website

<https://www.odoo.com>

Category

Uncategorized

Summary

Technical Name

new\_todo\_list

License

LGPL Version 3

Latest Version

11.0.1.0

Modul to manage personal To-Do Tasks

Diimplementasikan dengan Python  
class yang diambil dari template  
class Odoo

## 3. Model Layer

# Model layer

Aplikasi yg akan kita buat:

- Aplikasi untuk mencatatkan to-do tasks.
- Akan dibuat single text field untuk deskripsi task
- Akan ada checkbox untuk menandai yg sudah selesai dilakukan
- Akan ditambahkan tombol clean untuk menghapus tugas2 yg sudah selesai

# Data model

Buat file, `todo_model.py`

```
1  # -*- coding: utf-8 -*-
2  from odoo import models, fields
3
4  class TodoTask(models.Model):
5      _name = 'todo.task'
6      _description = 'To-do Task'
7      name = fields.Char('Description', required=True)
8      is_done = fields.Boolean('Done?')
9      active = fields.Boolean('Active', default=True)
```

# Data model

- Baris 1** – special marker, memberi info ke Python interpreter bahwa file tsb pakai format UTF-8 sehingga dapat handle non-ASCII characters.
- Baris 2** – import statement, mengimport objek *models* dan *fields* dari Odoo core
- Baris 4** – deklarasi model baru kita, yakni class yg diambil dari *models.Model*
- Baris 5** – atribut *\_name* merupakan identifier yg akan digunakan Odoo utk merefer ke model ini (*todo.task*).
- Baris 6** – atribut *\_description*, tidak mandatory, tapi demi model record yg user-friendly untuk user messages.
- Baris 7:9** – mendefinisikan field yg dibutuhkan model

Nah setelah file `todo_model.py` dibuat, file `init` harus dirubah agar dapat digunakan oleh modul. Maka harus di load di `__init__.py`, ubah menjadi

```
1  from . import todo_model
2
```



# Upgrade

- Stop-start server Odoo (jika diperlukan)
- Masuk Odoo dev mode
- Update list apps Odoo
  - Apps → Update Apps List → Reload
- Upgrade To-Do Application

# Cek technical menu

Sampai di sini, belum ada menu di dashboard yg bisa kita gunakan terkait todo apps, karena kita belum membuat viewnya. Untuk memastikan apa modelnya sudah terbuat, cek di:

- Setting → Database Structure (bagian kiri, di bawah technical) → Models
- Search todo.task → klik
- Cek apakah atribut2 dlm class TodoTask sudah ada apa belum. Tampilan di todo.task harusnya spt ini:

Edit

Create

Print ▼

Action ▼

Model Description

To-do Task

Type

Base Object

Model

todo.task

In Apps

new\_todo\_list

Transient Model

☐

Mail Thread

☐

Fields

Access Rights

Notes

Views

Field Name	Field Label	Field Type	Required	Readonly	Indexed	Type
__last_update	Last Modified on	datetime	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
active	Active	boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
create_date	Created on	datetime	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
create_uid	Created by	many2one	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
display_name	Display Name	char	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
id	ID	integer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
is_done	Done?	boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
name	Description	char	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
write_date	Last Updated on	datetime	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
write_uid	Last Updated by	many2one	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field

# Atribut/fields lain bawaan Odoo

Di bawah ini adalah reserved fields Odoo, otomatis ditambahkan setiap bikin model baru

- `id` is a unique numeric identifier for each record in the model.
- `create_date` and `create_uid` specify when the record was created and who created it respectively.
- `write_date` and `write_uid` confirm when the record was last modified and who modified it respectively.
- `__last_update` is a helper that is not actually stored in the database. It is used for concurrency checks.

Mendeskripsikan user interface dengan XML.

## 4. View Layer

# View layer

Buat sub-directory *views*, file XML diletakkan di directory tsb.

Jadi pathnya → *todo\_app/views*

# Menambahkan Menu

Di poin 3 kita sudah membuat model utk menyimpan data. Selanjutnya buat interface berupa menu pada dashboard Odoo.

Buat file, `todo_menu.xml`

```
1  <?xml version='1.0'?>
2  <odoo>
3      <!-- Action to open To-do Task list -->
4      <act_window id="action_todo_task"
5          name="To-do Task"
6          res_model="todo.task"
7          view_mode="tree,form"/>
8      <!-- Menu item to open To-do Task list -->
9      <menuitem id="menu_todo_task"
10         name="Todos"
11         action="action_todo_task"/>
12  </odoo>
```

Yang di atas tadi adalah Odoo data file, bisa dilihat ada 2 item utk ditambahkan ke dashboard Odoo:

- `<act_window>` mendefinisikan action yg akan memanggil model `todo.task` yg kita buat sebelumnya (di file `todo_model.py`).  
dengan tree view dan form view respectively.
- `<menuitem>` mendefinisikan menu (yg di tab atas Odoo) dan memanggil `action_todo_task` yg sdh dibuat di atasnya tag ini.





Modul kita belum tahu adanya file data XML yg barusan kita buat.

Maka, tambahkan key `"data"` pada file `__manifest__.py`

```
1  {
2      'name': 'To-Do Application',
3      'version': '1.0',
4      'description': """Modul to manage personal To-Do Tasks""",
5      'author': 'Rochana P H',
6      'depends': ['base'],
7      'data': ['views/todo_menu.xml'],|
8      'installable': True,
9      'application': True,
10     'autoinstall': False
11
12 }
```

Lalu, upgrade modul.

Todos Apps Settings  Administrator (todo) 

### To-do Task / New

Description	<input type="text"/>	Done?	<input type="checkbox"/>
Active?	<input checked="" type="checkbox"/>		

Form view

To-do Task		Search...
<input type="button" value="Create"/>	<input type="button" value="Import"/>	
<input type="checkbox"/>	Description	
<input type="checkbox"/>	RABU  KKPM Single Layer Perceptron SGD	
<input type="checkbox"/>	RABU  KKPM Backpropagation slide	
<input type="checkbox"/>	SENIN  Fuzzy slide	
<input type="checkbox"/>	RABU  AST Korelasi data.go.id	
<input type="checkbox"/>	RABU  ERP view.xml	
<input type="checkbox"/>	JUMAT  ERP finishing custom modul	

Tree/list view

# Tugas terakhir

- Mencari tutorial custom modul Odoo → implement (boleh desain sendiri juga)
- Tiap kelompok tidak boleh sama
- Jumat 18 Mei, **presentasi judul-ide**
- Jumat 25 Mei, **laporan dikumpulkan saat responsi**
- **Poin plus** → kalau bisa extend/mengkombinasikan modul yg sudah ada