

# Membuat Catatan Online dengan CherryPy, Zurb Foundation dan MySQL

---



Sabtu, 23 November 2013

Pukul 12.30 – 16.30 WIB

Laboratorium Umum Ilmu Komputer

---

Pemberdayaan Open Source Software  
Universitas Pendidikan Indonesia (POSS – UPI)  
Lab Rekayasa Perangkat Lunak, Gedung Ilmu Komputer  
Jln Setiabudhi no 21, Bandung



# Apa yang akan kita Gunakan ?

---



**TinyMCE**

**Jinja2**

# Cara menggunakannya ?

---

Untuk menggunakan MySQL di python kita membutuhkan konektor python ke MySQL. Dapatkan di [http:// dev.mysql.com](http://dev.mysql.com).

Setelah mendapatkan konektornya, ekstrak bundle dari konektor tersebut. Masuk kedalam foldernya dan ketik perintah :

```
python setup.py install
```

Hal ini hanya berlaku di Linux atau Unix :D. Kalau Windows harus download sesuai installer khususnya.

# Cara menggunakannya ?

---

Untuk menggunakan CherryPy dan Jinja2 di python kita harus menginstall CherryPy dan Jinja2. Dapatkan CherryPy di <http://cherrypy.org> dan Jinja2 di <http://jinja2.org>

Setelah mendapatkan CherryPy dan Jinja2, ekstrak kedua bundle tersebut. Masuk kedalam foldernya masing – masing dan ketik perintah :

```
python setup.py install
```

# Cara menggunakannya ?

---

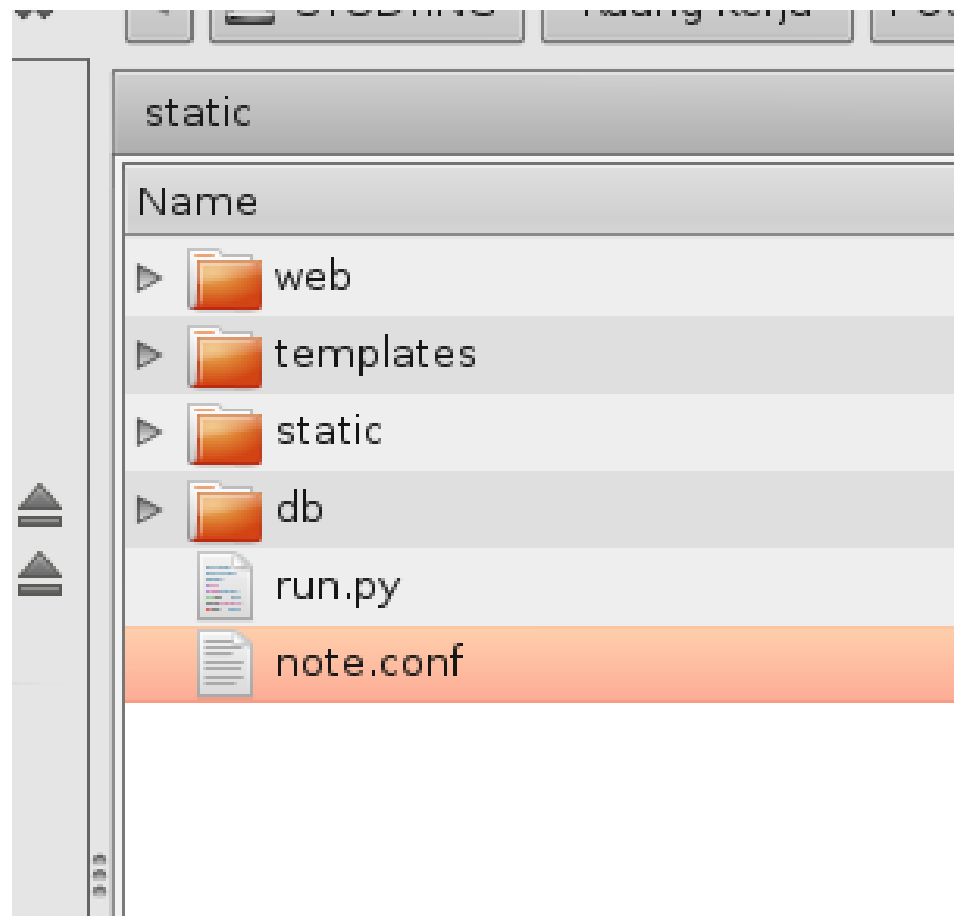
Untuk menggunakan Zurb Foundation dan TinyMCE, Anda bisa mendapatkan bundlenya di <http://foundation.zurb.com> dan <http://tinymce.org> .

Simpan hasil ekstraknya di folder static pada folder projek web Anda, atau tempatkan sesuai kebutuhan Anda. Kemudian panggil dari file HTML atau template yang akan ditampilkan

# Apa yang akan kita lakukan ?

---

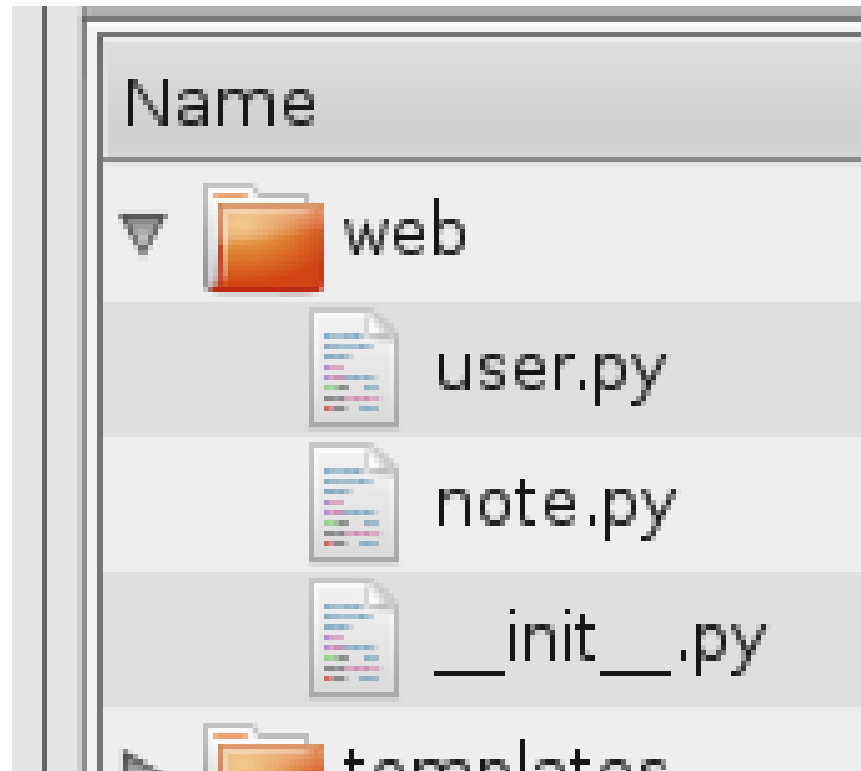
Buatlah folder dan file seperti pada gambar dibawah ini.



# Apa yang akan kita lakukan ?

---

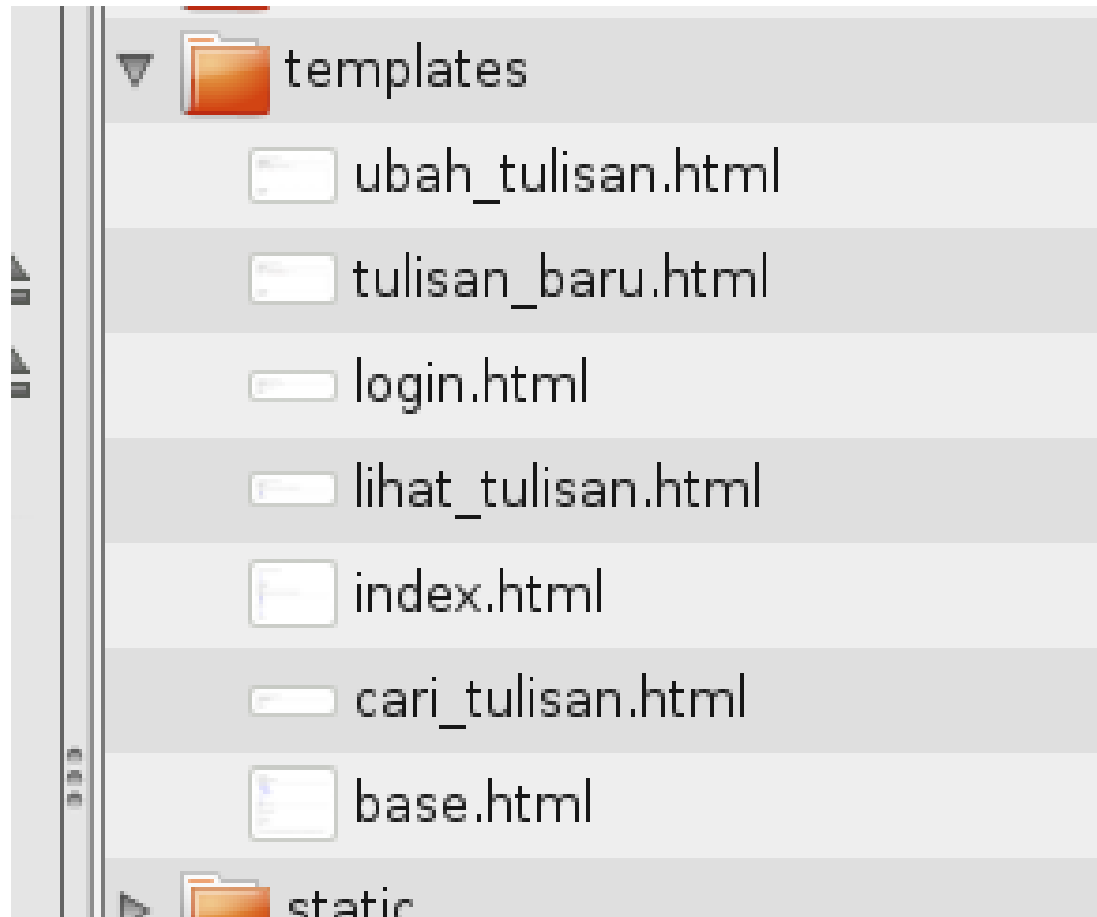
Buatlah folder dan file seperti pada gambar dibawah ini.



# Apa yang akan kita lakukan ?

---

Buatlah folder dan file seperti pada gambar dibawah ini.

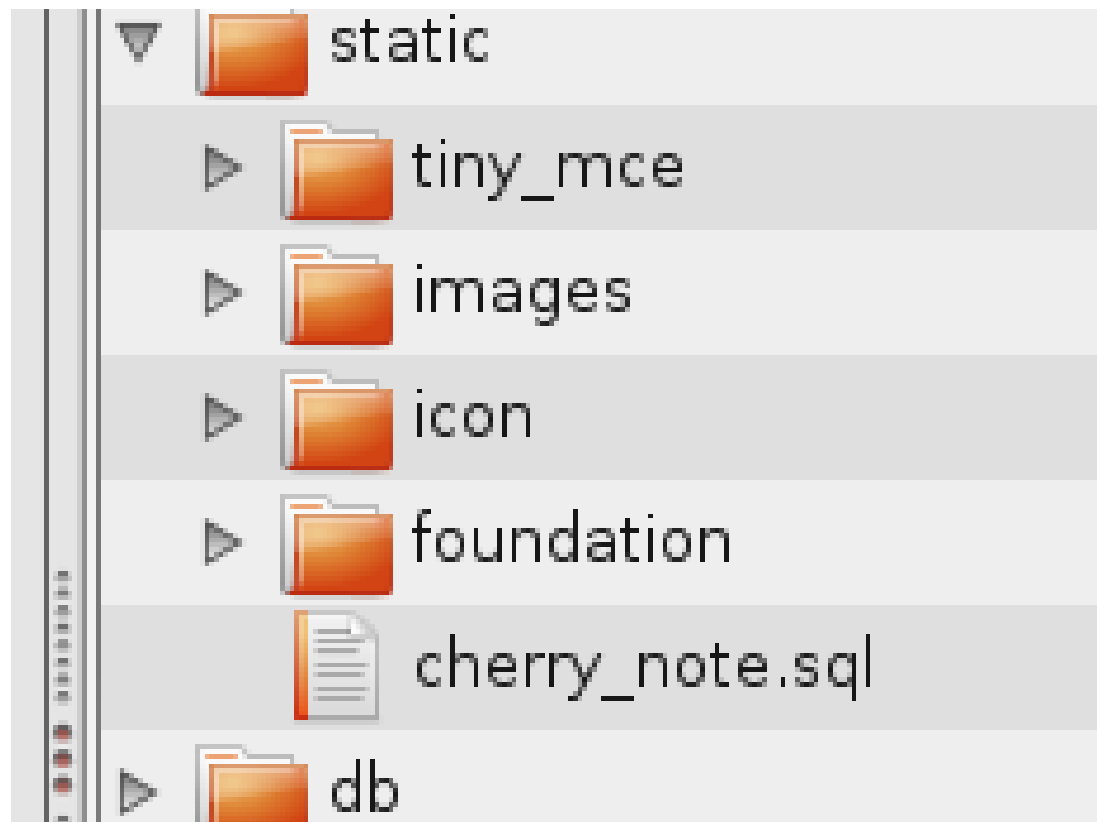




# Apa yang akan kita lakukan ?

---

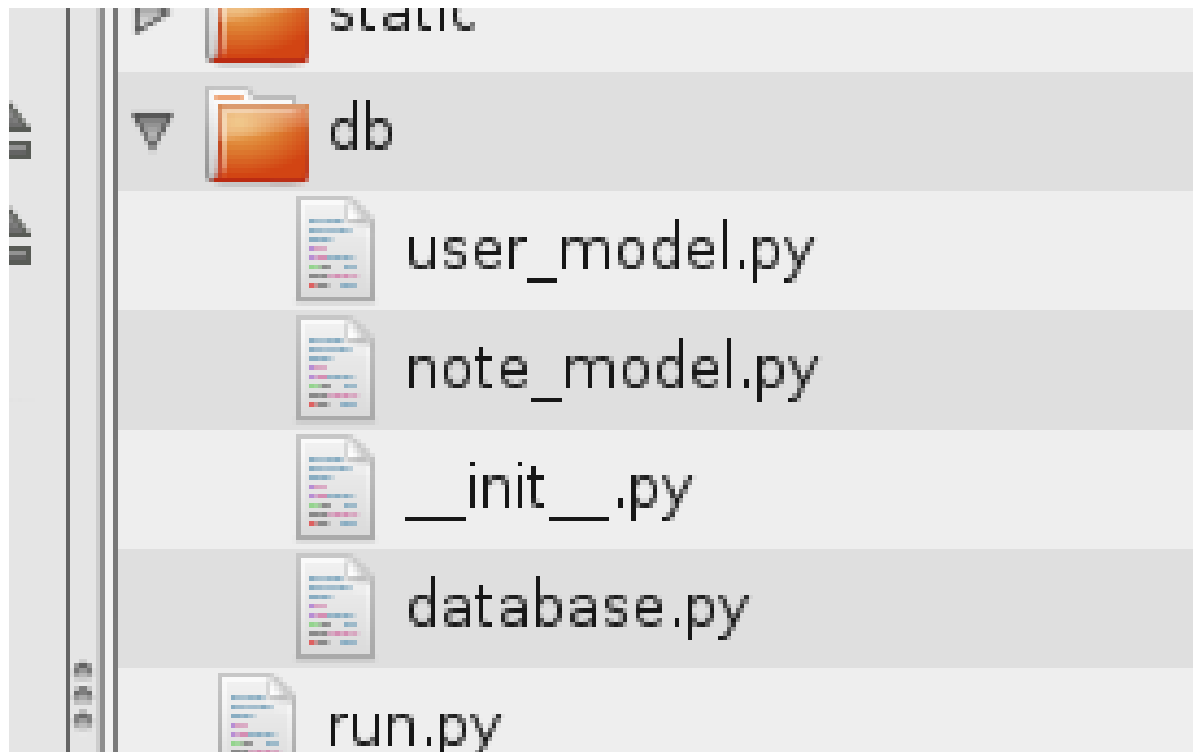
Copy bahan – bahan tambahan yang sudah diberikan kedalam folder static.



# Apa yang akan kita lakukan ?

---

Buatlah folder dan file seperti pada gambar dibawah ini.



# Apa yang akan kita lakukan ?

---

Kita akan membangun aplikasi catatan ini dimulai dengan tahapan – tahapan berikut :

- Membuat database
- Membuat file konfigurasi
- Membuat mekanisme autentikasi
- Membuat proses pengisian catatan baru ke database
- Menampilkan detail catatan
- Memperbaharui salah satu catatan
- Menghapus catatan
- Membuat sistem pencarian catatan

# Membuat database

---

Buka PHPMysqlAdmin, kemudian buat database baru dengan nama `cherry_note`

Import file **`cherry_note.sql`** yang terdapat di folder static kedalam database `cherry_note`

Database sudah siap digunakan dan kita akan membuat aplikasinya :D. Tambah sesuai selera jumlah user dengan akun baru untuk ujicoba login berikutnya

# Membuat database : isi cherry\_note.sql

---

```
DROP TABLE IF EXISTS `note`;  
CREATE TABLE `note` (  
  `id_note` int(11) NOT NULL AUTO_INCREMENT,  
  `id_user` int(11) NOT NULL,  
  `judul` varchar(200) NOT NULL,  
  `tag` varchar(200) NOT NULL,  
  `isi` text NOT NULL,  
  `tanggal_dibuat` datetime NOT NULL,  
  `tanggal_diperbaharui` datetime NOT NULL,  
  PRIMARY KEY (`id_note`)  
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `users`;  
CREATE TABLE `users` (  
  `id_users` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(45) NOT NULL,  
  `password` varchar(45) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `facebook` varchar(45) DEFAULT NULL,  
  `twitter` varchar(45) DEFAULT NULL,  
  `deskripsi` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`id_users`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

```
INSERT INTO `users` VALUES (1,'root','root','ridwanbejo@gmail.com','',' ',NULL);
```

# Membuat file konfigurasi

---

File konfigurasi ini berisi pengaturan aplikasi yang akan kita gunakan. Biasanya berisi pengaturan lokasi asset website kita seperti gambar, stylesheet, javascript, dan plugin website.

Selain diatas, biasanya dalam file konfigurasi ini terdapat pengaturan session management, lokasi server, port server, favicon website kita, dan lainnya

Editlah file konfigurasi **note.conf** dengan teks editor yang Anda gunakan. Kemudian isi file tersebut dengan baris konfigurasi pada slide berikutnya.

# Membuat file konfigurasi : isi note.conf

---

```
[global]
server.socket_host = "127.0.0.1"
server.socket_port = 8080
server.thread_pool = 10

[/]
tools.staticdir.root = "/home/ridwan/Project/CherryPy Project/note"

[/static]
tools.staticdir.on = True
tools.staticdir.dir = "static"
```

# Membuat Autentikasi

---

Dari beberapa fitur yang akan kita bangun. Autentikasi merupakan fitur krusial dari sebuah aplikasi web agar tidak sembarangan orang menggunakan aplikasi web tersebut.

Biasanya terdiri dari login dan logout. User akan diberikan sebuah form login. Kemudian aplikasi akan mencari akun yang dicari. Jika ada akunnya, sistem akan menyimpan data user yang login tersebut dalam session, jika tidak ada maka akan diberikan informasi error bahwa akun tidak ada.



# Membuat Autentikasi

---

Karena kita mendekati paradigma Model View Controller (MVC). Kode program yang berisi kode presentasi (HTML, CSS, JS) akan disimpan di folder template,

Kode program yang berisi pengaturan akses kontrol terhadap URL dan proses tertentu disimpan di folder web.

Kode program yang berisi akses terhadap database disimpan di file folder db. Dan konfigurasi ke database disimpan di **database.py**

File css, js, gambar, icon, dan plugin web disimpan di folder static. Agar file – file tadi tidak tercampur dengan file – file yang sudah dikelompokkan tadi.

# Membuat Autentikasi

---

**Ayo kita mulai :D**

# Membuat Autentikasi

---

Buat file **database.py** dan simpan di folder db

```
import MySQLdb

db = MySQLdb.connect('localhost', 'root', 'root',
'cherry_note')
cursor = db.cursor()
```

File database.py ini akan digunakan oleh setiap file model untuk mencegah pengetikan kode akses ke database secara berulang

# Membuat Autentikasi

---

Selanjutnya kita buat kelas model untuk akses tabel User dengan nama **user\_model.py**

```
from database import *

class UserModel:
    def __init__(self):
        self.db = db
        self.cursor = cursor

    def get_user(self, username, password):
        sql = "select * from users where username='%s' and password = '%s'" % (username, password)
        results = {}
        try :
            self.cursor.execute(sql)
            row = self.cursor.fetchone()
            results.update({'id_users':row[0], 'username':row[1], 'email':row[3], 'facebook':row[4], 'twitter':row[5], 'deskripsi':row[6]})
            return results
        except:
            print "Error : tidak bisa mengambil data"
            return results
```

# Membuat Autentikasi

---

Dalam mengelompokkan sebuah file kode program. Di python dikenal sebuah file bernama **\_\_init\_\_.py** yang berfungsi untuk mengatur file apa saja yang bisa digunakan dari isi folder tersebut.

Dalam file `__init__.py` yang akan kita tulis, terdapat pemanggilan terhadap model note, model user, dan konfigurasi database.

```
from note_model import NoteModel
from user_model import UserModel
from database import *
```

# Membuat Autentikasi

---

Nah dalam Cherrypy sebuah controller dinyatakan dengan kelas. Kelas ini memiliki method – method (function) yang bisa diakses lewat url dengan menggunakan `@cherrypy.expose` atau menjadi method yang tidak bisa diakses dan hanya digunakan secara internal.

Controller User ini memiliki dua method yang bisa diakses oleh user yaitu login dan logout. Controller User ini mengakses model `user_model` untuk mendapatkan data – data user.

Selain itu session harus dinyalakan di controller ini. Dan kita juga harus mengimport library cherrypy untuk menggunakan fitur – fitur yang ada di cherrypy.

# Membuat Autentikasi

---

Simpan file **user.py** ini di folder web

```
#!/usr/bin/env python
from db import UserModel
import cherrypy

class User:

    _cp_config = {'tools.sessions.on': True}

    def __init__(self):
        self.usermdl = UserModel()

    @cherrypy.expose
    def login(self, username=None, password=None):
        data_user = self.usermdl.get_user(username, password)
        if len(data_user) != 0:
            cherrypy.session['data_user'] = data_user
        else:
            cherrypy.session['error_login_msg'] = 'Akun tidak ditemukan'
            raise cherrypy.HTTPRedirect('/')

    @cherrypy.expose
    def logout(self):
        if cherrypy.session.get('data_user'):
            del cherrypy.session['data_user']
            raise cherrypy.HTTPRedirect('/')
        else:
            raise cherrypy.HTTPRedirect('/')
```

# Membuat Autentikasi

---

Fungsi login dan logout di **user.py** tidak bisa berdiri sendiri. Melainkan harus digunakan oleh controller Note yang ada di **note.py**. Dengan demikian walaupun kita membuat controller lain yang berisi dari pengembangan aplikasi kita. Penggunaan controller User bisa digunakan di controller lain tersebut. Jadi cukup sekali tulis dipanggil darimanapun yang membutuhkan.

Controller Note ini mempunyai banyak fitur. Tapi untuk tahapan ini kita batasi dulu sampai fitur untuk melihat halaman index dan halaman login. Selanjutnya tinggal menambahkan kode tambahan di controller Note.

Controller Note harus menjalankan session, meload folder templates untuk meload file – file template yang kita buat (HTML), kemudian meload model Note yang akan digunakan untuk mengakses tabel Note.

Di method index ini Jika Anda belum pernah login maka akan tampil form login. Jika gagal login maka akan muncul peringatan. Jika berhasil akan tampil ke halaman index aplikasi catatan online yang kita bangun.



# Membuat Autentikasi

---

Simpan file **note.py** ini di folder web

```
#!/usr/bin/env python
from db import NoteModel
from jinja2 import Environment, FileSystemLoader
import cherrypy

env = Environment(loader=FileSystemLoader('templates'))

class Note:

    _cp_config = {'tools.sessions.on': True}

    def __init__(self):
        self.notemdl = NoteModel()
```

---

# Membuat Autentikasi

---

Simpan file **note.py** ini di folder web

```
@cherry.py.expose
def index(self, tag=None):
    if cherry.py.session.get('data_user'):
        tmpl = env.get_template('index.html')
        if tag:
            posts = self.notemdl.get_post_by_tag(tag)
        else:
            if tag is None:
                posts = self.notemdl.all_post()
            elif tag=="":
                posts = []
        return tmpl.render(posts=posts, data_user=cherry.py.session.get('data_user'))
    else:
        tmpl = env.get_template('login.html')
        if cherry.py.session.get('error_login_msg'):
            error_login_msg = cherry.py.session.get('error_login_msg')
            del cherry.py.session['error_login_msg']
            return tmpl.render(error_login_msg=error_login_msg)
        else:
            return tmpl.render()
```

# Membuat Autentikasi

---

**Kapan bisa  
dijalankannya :D ?**

**Kalem yah kita baru  
membuat model dan  
controller tinggal template :D**

# Membuat Autentikasi

---

Sebelum menuju pembuatan template kita harus membuat dua buah file penting lainnya yaitu **note\_model.py** yang berisi fitur – fitur untuk mengakses tabel Note, dan **\_\_init\_\_.py** untuk ditempatkan di folder web.

Buat dulu **\_\_init\_\_.py** dan tempatkan di folder web

```
from note import Note
from user import User
```

# Membuat Autentikasi

---

Simpan file **note\_model.py** ini di folder db

```
from database import *
import datetime

class NoteModel:

    def __init__(self):
        self.db = db
        self.cursor = cursor

    def all_post(self):
        sql = "select * from note where id_user = 1 order by tanggal_diperbaharui desc"
        try :
            self.cursor.execute(sql)
            temp_results = self.cursor.fetchall()
            results = []
            i = 1
            for row in temp_results:
                num = i
                results.append({'num':num, 'id_note':row[0], 'judul':row[2], 'tag':row[3],
'isi':row[4], 'tanggal_dibuat':row[5], 'tanggal_diubah':row[6]})
            return results
        except:
            print "Error : tidak bisa mengambil data"
```

# Membuat Autentikasi

---

Nah telah sampailah kita untuk membuat file template yang akan digunakan untuk menampilkan data yang kita inginkan.

Template Engine yang akan kita gunakan adalah Jinja2. Dengan menggunakan template kita bisa menggunakan template inheritance agar file header dan footer tidak disalin kedalam file html baru. Kemudian kita juga bisa menggunakan template syntax pada Jinja2 agar kode template tidak dicampuri oleh kode serverside.

Nanti kita akan membuat sebuah file **base.html** yang akan digunakan ulang oleh template turunannya seperti **login.html**. Dengan demikian kode html di **login.html** hanya berisi kode html bagian form login saja. Untuk header dan footer menggunakan kode html yang ada di **base.html**

Template ini akan dipanggil oleh method – method yang ada di controller. Di dalam rendering template tersebut kita juga bisa melewatkan parameter – parameter yang bisa digunakan di dalam template yang dirender.

# Membuat Autentikasi

---

Simpan file **base.html** ini di folder templates

```
<html>
<head>
  <title>Cherry Note</title>
  <!-- Included JS Files (Compressed) -->

  <script src="/static/foundation/javascripts/jquery.js"></script>
  <script src="/static/foundation/javascripts/foundation.min.js"></script>

  <!-- Initialize JS Plugins -->
  <script src="/static/foundation/javascripts/app.js"></script>

  <script src="/static/foundation/javascripts/modernizr.foundation.js"></script>
  <script type="text/javascript" src="/static/tiny_mce/tiny_mce.js"></script>
  <script type="text/javascript">
    tinyMCE.init({
      mode : "textareas",
      theme : "advanced",
      theme_advanced_toolbar_location : "top",
      theme_advanced_toolbar_align : "left",
    });
  </script>
  <link rel="stylesheet" href="/static/foundation/stylesheets/foundation.min.css">
  <link rel="stylesheet" href="/static/foundation/stylesheets/app.css">

</head>
```

---

# Membuat Autentikasi

Terusan dari file **base.html** ini di folder templates

```
<body>
  <div class="row" id="header">
    <div class="twelve columns">
      <h2>Cherry Note</h2>
      <p>Sebuah Catatan Online berbasis <b>Python</b></p>
      <hr />
      {% if data_user %}
      <nav style="" class="top-bar hide-for-small">
        <ul class="title-area">
          <!-- Title Area -->
          <li class="name">
            <h1><a href="/">Beranda</a></h1>
          </li>
        </ul>
        <!-- Right Nav Section -->
        <ul class="left">
          <li class="divider"></li>
          <li class="has-dropdown"><a href="#">Catatan</a>
            <ul class="dropdown">
              <li><a href="/tulisan_baru">Buat Catatan Baru</a></li>
              <li><a href="/ekspor">Ekspor Catatan</a></li>
            </ul>
          </li>
          <li class="divider"></li>
          <li class=""><a href="/cari">Cari</a></li>
          <li class="divider"></li>
        </ul>
      </nav>
    </div>
  </div>
```



# Membuat Autentikasi

---

Terusan dari file **base.html** ini di folder templates

```
<ul class="right">
  <li class="divider"></li>
  <li class=""><a href="/user/logout">Logout</a></li>
</ul>
</section>
</nav>
{% endif %}
</div>
</div>

<div class="row" id="content">
  {% block content %}
  {% endblock %}
</div>
```

# Membuat Autentikasi

---

Terusan dari file **base.html** ini di folder templates

```
<div class="row" id="footer">
<hr />
<section class="eight columns">
  <h6>Powered by :</h6>
  <div class="four columns">
    
    <p>
      A Minimalist Python Web Framework
    </p>
  </div>
  <div class="four columns">
    
    <p>
      Linux for Human Being
    </p>
  </div>

  <div class="four columns">
    
    <p>
      Start Here Build Everywhere
    </p>
  </div>
</section>
```

# Membuat Autentikasi

---

Terusan dari file **base.html** ini di folder templates

```
<section class="four columns panel">
  <h6>Supported by :</h6>
  
  <p> Projek ini dimulai 30 November 2013. Didukung oleh
  Pemberdayaan Open Source Software -
  Universitas Pendidikan Indonesia (POSS - UPI), 2013 </p>
</section>
</div>
</body>
</html>
```

# Membuat Autentikasi

---

**Panjang banget yah  
HTML nya :D**

**Tapi berikutnya kita tidak akan  
menulis sepanjang itu lagi**

# Membuat Autentikasi

---

Di template – template berikutnya seperti **login.html** dan **index.html** kita hanya akan menulis sedikit kode html karena kita akan menggunakan extends terhadap **base.html**

Coba perhatikan pada `{% block content %}` `{% endblock %}` pada **base.html**. Kedua tag tersebut merupakan penanda bahwa pada block tersebut kontennya bisa diganti sesuai dengan yang diperintahkan aplikasi. Jadi bisa saja nantinya berisi form login atau berisi daftar catatan.

Penamaan block bisa Anda sesuaikan dengan kebutuhan.

# Membuat Autentikasi

---

Simpan file **login.html** ini di folder templates

```
{% extends "base.html" %}

{% block content %}

<div class="four columns centered">
  <div class="panel" style="">
    <h4>Silahkan login</h4>
    <hr />
    {% if error_login_msg %}
    <small class="error">{{error_login_msg}}</small>
    {% endif %}
    <form method="post" action="/user/login">
      <label>Username : </label>
      <br />
      <input type="text" name='username' />
      <label>Password : </label>
      <br />
      <input type="password" name='password' />
      <br />
      <br />
      <input type="submit" value="Login" class="small button" />
    </form>
  </div>
</div>

{% endblock %}
```

# Membuat Autentikasi

Simpan file **index.html** ini di folder templates

```
{% extends "base.html" %}

{% block content %}

    {% for post in posts %}
    <div class="panel" style="">
        <h4>{{ post['judul'] }}</h4>
        <h6> {{post['tag']}}</h6>
        <p>tanggal dibuat : {{ post['tanggal_dibuat'] }} | tanggal_diubah : {{ post['tanggal_diubah'] }}</p>
        <p>{{ post['isi'] }}</p>
        <hr />
        <ul class="inline-list">
            <li><a href='/lihat_tulisan/{{post.id_note}}'></a></li>
            <li><a href='/ubah_tulisan/{{post.id_note}}'></a></li>
            <li><a href='/hapus_tulisan/{{post.id_note}}'></a></li>
        </ul>
    </div>
    {% endfor %}

</div>

{% endblock %}
```

# Membuat Autentikasi

---

Sekarang kita akan jalankan autentikasi kita melalui satu buah file bernama **run.py**. Di dalam file ini kita memanggil semua controller yang dibutuhkan, menentukan hirarki URL dari aplikasi kita, membaca konfigurasi aplikasi kita, dan memanggil built-in server yang dimiliki cherrypy.

```
#!/usr/bin/env python
from web import Note, User
import cherrypy
import os.path

root = Note()
root.user = User()

tutconf = os.path.join(os.path.dirname(__file__), 'note.conf')

if __name__ == '__main__':
    cherrypy.quickstart(root, config=tutconf)
else:
    cherrypy.tree.mount(root, config=tutconf)
```



# Membuat Autentikasi

---

Kemudian dari terminal a.k.a konsol kita jalankan file run.py dengan perintah dibawah ini :

```
python run.py
```

Mari kita lihat hasil yang pekerjaan kita :D

# Membuat Autentikasi

Tampilan saat belum login :



## Cherry Note

Sebuah Catatan Online berbasis Python

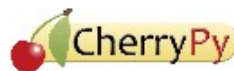
Silahkan login

Username :

Password :

Login

Powered by :



A Minimalist Python Web Framework



Linux for Human Being



Start Here Build Everywhere

Supported by :



Projek ini dimulai 30 November 2013.  
Didukung oleh Pemberdayaan Open Source  
Software - Universitas Pendidikan Indonesia

# Membuat Autentikasi

---

Tampilan saat sudah login :



## Cherry Note

Sebuah Catatan Online berbasis Python

Beranda

Catatan ▾

Cari

Logout

### bejo ganteng

bejo ganteng

tanggal dibuat : 2013-11-23 07:12:00 | tanggal\_diubah : 2013-11-23 07:12:00

Tulis apa yang sedang Anda pikirkan :D



# Menampilkan Detail Catatan

---

Pada pekerjaan sebelumnya kita hanya sampai menampilkan daftar catatan. Itupun kalau diisi terlebih dahulu dengan beberapa catatan. Sekarang kita akan menampilkan detail dari salah satu catatan

Dengan membubuhkan id catatan pada link lihat tulisan yang diganti dengan gambar kaca pembesar, link tersebut akan diproses oleh method **lihat\_tulisan** kemudian mengambil detail catatan dengan **get\_post\_by\_id** dan dirender ke template **lihat\_tulisan.html**

Lebih lengkapnya kita lanjutkan pekerjaan kita :D

# Menampilkan Detail Catatan

---

Terusan file `note_model.py`

```
def get_post_by_id(self, id_note):
    sql = "select * from note where id_user = 1 and id_note = %d order by
tanggal_diperbaharui desc" % (id_note)
    try :
        self.cursor.execute(sql)
        row = self.cursor.fetchone()
        results = {}
        results.update({'id_note':row[0], 'judul':row[2], 'tag':row[3], 'isi':row[4],
'tanggal_dibuat':row[5], 'tanggal_diubah':row[6]})
        return results
    except:
        print "Error : tidak bisa mengambil data"
```

# Menampilkan Detail Catatan

---

Terusan file **note.py**

```
-----  
  
@cherry.py.expose  
def lihat_tulisan(self, id_note):  
    if cherry.py.session.get('data_user'):  
        tmpl = env.get_template('lihat_tulisan.html')  
        post = self.notemdl.get_post_by_id(int(id_note))  
        print post  
        return tmpl.render(post=post, data_user=cherry.py.session.get('data_user'))  
    else:  
        raise cherry.py.HTTPRedirect('/')
```

# Menampilkan Detail Catatan

Simpan file `lihat_tulisan.html` di folder `templates`

```
{% extends "base.html" %}

{% block content %}

<div class="twelve columns">
  <div class="panel" style="">
    <h4>{{ post['judul'] }}</h4>
    <h6> {{post['tag']}}</h6>
    <p>tanggal dibuat : {{ post['tanggal_dibuat'] }} | tanggal_diubah :
    {{ post['tanggal_diubah'] }}</p>
    <p>{{ post['isi'] }}</p>
    <hr />
    <ul class="inline-list">
      <li><a href='/ubah_tulisan/{{post.id_note}}'></a></li>
      <li><a href='/hapus_tulisan/{{post.id_note}}'></a></li>
    </ul>
  </div>
</div>

{% endblock %}
```

# Membuat Catatan Baru

---

Di fitur membuat catatan baru ini kita akan mengisikan judul, tag, dan isi catatan yang akan dibuat oleh user yang login saat itu.

Dengan menggunakan TinyMCE maka proses editing catatan menjadi lebih mudah dan kaya fitur karena ada bullet and numbering, aligning, pengisian gambar, dan penambahan link

Lebih lengkapnya kita lanjutkan pekerjaan kita :D



# Membuat Catatan Baru

---

## Terusan file `note_model.py`

```
def insert_post(self, judul, tag, isi):
    temp_tanggal = datetime.datetime.now()
    sql = "insert into note (id_user, judul, tag, isi, tanggal_dibuat, tanggal_diperbaharui) values
(%d, '%s', '%s', '%s', '%s', '%s')" % (1, judul, tag, isi, temp_tanggal.strftime('%Y-%m-%d %H-%M-%S'), temp_tanggal.strftime('%Y-%m-%d %H-%M-%S'))
    try:
        self.cursor.execute(sql)
        db.commit()
        print "Info : data berhasil diisikan.."
    except:
        db.rollback()
        print "Error : pengisian data gagal.."
```

# Membuat Catatan Baru

---

## Terusan file `note.py`

```
-----  
  
@cherry.py.expose  
def tulisan_baru(self):  
    if cherry.py.session.get('data_user'):  
        tmpl = env.get_template('tulisan_baru.html')  
        return tmpl.render(data_user=cherry.py.session.get('data_user'))  
    else:  
        raise cherry.py.HTTPRedirect('/')
```

```
  
@cherry.py.expose  
def proses_tulisan_baru(self, judul=None, tag=None, elm1=None):  
    if cherry.py.session.get('data_user'):  
        self.notemdl.insert_post(judul, tag, elm1)  
        raise cherry.py.HTTPRedirect("/")  
    else:  
        raise cherry.py.HTTPRedirect('/')
```

# Membuat Catatan Baru

---

Simpan file `tulisan_baru.html` di folder templates

```
{% extends "base.html" %}

{% block content %}

<div class="twelve columns">
  <div class="panel" style="">
    <h3>Tulisan Baru</h3>
    <hr />
    <form method="post" action="/proses_tulisan_baru">
      <label>Judul :</label>
      <input type="text" name='judul' placeholder="isi judulnya disini ..." />
      <br />
      <label>Tag (contoh : linux, programming, python, cherry.py) : </label>
      <input type="text" name='tag' placeholder="isi tagnya disini ..." />
      <br />
      <label>Isi Catatan :</label>
      <textarea id="elm1" name="elm1" rows="15" cols="80" style="width: 100%">
        Tulis apa yang sedang Anda pikirkan :D
      </textarea>

      <br />
      <input type="submit" value="Submit" class="small button"/>
      <input type="reset" value="Reset" class="small button"/>
    </form>
  </div>
</div>

{% endblock %}
```

# Memperbaharui salah satu catatan

---

Hampir sama dengan melihat detail catatan. Pada proses ini catatan yang akan diperbaharui setiap datanya akan diisikan di field pada form update catatan dan kita bisa memperbaiki setiap data yang keliru.

Lebih lengkapnya kita lanjutkan pekerjaan kita :D

# Memperbaharui salah satu catatan

---

## Terusan file `note_model.py`

```
def update_post(self, judul, tag, isi, id_note):
    temp_tanggal = datetime.datetime.now()
    sql = "update note set judul='%s', tag='%s', isi='%s', tanggal_diperbaharui='%s' where id_note=%d" %
(judul, tag, isi, temp_tanggal.strftime('%Y-%m-%d %H-%M-%S'), int(id_note))
    try:
        self.cursor.execute(sql)
        db.commit()
        print "Info : data berhasil diubah.."
    except:
        db.rollback()
        print "Error : pengubahan data gagal.."
```

# Memperbaharui salah satu catatan

---

## Terusan file `note.py`

```
-----  
  
@cherry.py.expose  
def ubah_tulisan(self, id_note):  
    if cherry.py.session.get('data_user'):  
        tmpl = env.get_template('ubah_tulisan.html')  
        post = self.notemdl.get_post_by_id(int(id_note))  
        return tmpl.render(post=post, data_user=cherry.py.session.get('data_user'))  
    else:  
        raise cherry.py.HTTPRedirect('/')  
  
@cherry.py.expose  
def proses_ubah_tulisan(self, judul=None, tag=None, elm1=None, id_note=None):  
    if cherry.py.session.get('data_user'):  
        self.notemdl.update_post(judul, tag, elm1, id_note)  
        raise cherry.py.HTTPRedirect("/")  
    else:  
        raise cherry.py.HTTPRedirect('/')
```

# Memperbaharui salah satu catatan

---

Simpan file **ubah\_tulisan.html** di folder templates

```
{% extends "base.html" %}

{% block content %}

<div class="twelve columns">
  <div class="panel" style="">
    <h3>Mengubah Tulisan</h3>
    <hr />
    <form method="post" action="/proses_ubah_tulisan">
      <label>Judul :</label>
      <input type="text" name='judul' placeholder="isi judulnya disini ..." value="{{post.judul}}"/>
      <br />
      <label>Tag (contoh : linux, programming, python, cherry py) : </label>
      <input type="text" name='tag' placeholder="isi tagnya disini ..." value="{{post.tag}}"/>
      <br />
      <label>Isi Catatan :</label>
      <textarea id="elm1" name="elm1" rows="15" cols="80" style="width: 100%">
        {{ post.isi }}
      </textarea>
      <input type="hidden" name="id_note" value="{{ post.id_note }}" />
    </form>
  </div>
</div>
```

---

# Memperbaharui salah satu catatan

---

Terusan file `ubah_tulisan.html` di folder templates

```
-----  
  
    <br />  
    <br />  
    <input type="submit" value="Submit" class="small button"/>  
    <input type="reset" value="Reset" class="small button"/>  
  </form>  
</div>  
  
</div>  
  
{% endblock %}
```



# Menghapus catatan

---

Hampir sama dengan melihat detail catatan. Pada proses ini catatan yang akan dihapus tinggal pilih icon tong sampah pada setiap catatan.

Pada proses hapus ini tidak dibutuhkan template karena begitu catatan dihapus langsung kembali ke halaman utama

Lebih lengkapnya kita lanjutkan pekerjaan kita :D

# Menghapus catatan

---

Terusan file `note_model.py`

```
def delete_post(self, id_note):
    sql = "delete from note where id_note=%d" % (int(id_note))
    try:
        self.cursor.execute(sql)
        db.commit()
        print "Info : data berhasil dihapus.."
    except:
        db.rollback()
        print "Error : penghapusan data gagal.."
```

# Menghapus catatan

---

## Terusan file `note.py`

```
-----  
@cherrypy.expose  
def hapus_tulisan(self, id_note):  
    if cherrypy.session.get('data_user'):  
        self.notemdl.delete_post(id_note)  
        raise cherrypy.HTTPRedirect("/")  
    else:  
        raise cherrypy.HTTPRedirect('/')
```

# Mencari catatan

---

Hampir sama dengan melihat detail catatan. Pada proses ini catatan yang akan dihapus tinggal pilih icon tong sampah pada setiap catatan.

Pada proses hapus ini tidak dibutuhkan template karena begitu catatan dihapus langsung kembali ke halaman utama

Lebih lengkapnya kita lanjutkan pekerjaan kita :D

# Mencari catatan

---

## Terusan file `note_model.py`

```
def get_post_by_tag(self, tag):
    sql = "select * from note where tag like '%%%s%%'" % (tag)
    try :
        self.cursor.execute(sql)
        temp_results = self.cursor.fetchall()
        results = []
        i = 1
        for row in temp_results:
            num = i
            results.append({'num':num, 'id_note':row[0], 'judul':row[2], 'tag':row[3], 'isi':row[4],
'tanggal_dibuat':row[5], 'tanggal_diubah':row[6]})
            i = i + 1
        return results
    except:
        print "Error : tidak bisa mengambil data"
```

# Mencari catatan

---

Terusan file **note.py**

```
-----  
  
@cherry.py.expose  
def cari(self):  
    if cherry.py.session.get('data_user'):  
        tmpl = env.get_template('cari_tulisan.html')  
        return tmpl.render(data_user=cherry.py.session.get('data_user'))  
    else:  
        raise cherry.py.HTTPRedirect('/')
```

# Mencari catatan

---

Simpan file **cari\_tulisan.html** di folder templates

```
{% extends "base.html" %}

{% block content %}

<div class="twelve columns">
  <div class="panel" style="">
    <h4>Cari catatan Anda disini :</h4>
    <hr />
    <form method="get" action="/">
      <input type="text" name='tag' placeholder="masukkan nama catatan yang ingin dicari ..." />
      <br />
      <br />
      <input type="submit" value="Cari" class="small button"/>
    </form>
  </div>
</div>

{% endblock %}
```

# Apa yang telah kita lakukan :D

---

Akhirnya kita sudah membuat sebuah aplikasi catatan online yang meliputi fitur :

- autentikasi
- membuat catatan baru
- melihat detail catatan
- memperbaharui salah satu catatan
- menghapus catatan
- mencari catatan

Dan yang belum lengkap di fitur ini adalah pendaftaran User baru, pembaharuan profil user, lupa password user, dan Halaman manajemen admin agar bisa menjaga konten dari sistem kita dari hal yang berbau SARA dan Pornografi



# Follow mereka di :

---



**@cherrypy**



**@foundationzurb**



**@MySQL**

# Dapatkan juga :

---

**File – file static :**

<http://github.com/ridwanbejo/cherrynote>

**Info pembahasan cherrypy terbaru :**

<http://www.poss-upi.org>