

TWISTAngular CHEAT SHEETS

- **Redirect ke halaman**

Controller
<pre>\$app->redirect(...); // ubah ... dengan url tujuan</pre>

- **Fetch data dari database (nama kolom jadi property object)**

Model
<pre>public function getTest() { \$query = \$this->db->query("SELECT * FROM test"); \$result = array(); foreach (\$query as \$val) { \$result[] = array('id' => \$val->ID_TEST, 'data' => \$val->DATA_TEST); } return \$result; }</pre>
Controller
<pre>\$ctr->model('test'); // sesuai nama file model, contoh: test_model.php \$ctr->TestModel->getTest(); // nama model titlecase nama file, spasi dibuang</pre>

- **Fetch data pertama dari tabel**

Model
<pre>public function cariTest() { \$query = \$this->db->query("SELECT * FROM test", true); // hanya berisi 1 object return \$query->ID_TEST; }</pre>

- **Fetch data jadikan array (nama kolom jadi index array)**

Model
<pre>\$query = \$this->db->query("SELECT * FROM test", NULL, true); return \$result['ID_TEST'];</pre>

- **Menampilkan View**

Controller
<pre>\$ctr->view('profil', array()); // akan mencari profil.html di folder view</pre>

- **Menampilkan data dari Controller ke View**

Controller
<pre>\$ctr->view('profil', array('nama' => 'Fulan bin Fulan'));</pre>
View (profil.html)
<pre><h1>Selamat Datang {{ \$nama }}</h1></pre>

- Menampilkan data dari Model ke HTML View

Model
<pre>public function get_test() { \$r = array(); foreach (\$this->db->query("SELECT * FROM test") as \$val) { \$r[] = array('id' => \$val->ID_TEST, 'data' => \$val->DATA_TEST); } return \$r; }</pre>
Controller
<pre>\$ctr->model('test'); \$datatest = \$ctr->TestModel->get_test(); \$ctr->view('testpage', array('tests' => \$datatest));</pre>
View
<pre><h1>Data Test</h1> <table class="table"> @foreach (\$tests as \$test) <tr> <td>{{ \$test['id'] }}</td> <td>{{ \$test['nama'] }}</td> </tr> @endforeach </table></pre>

- Mendapatkan parameter dari URL

Controller
<pre>\$app->get('/hello/:nama', function(\$nama) use(\$app, \$ctr) { echo 'Halo ', \$nama; // \$nama berisi data dari URL });</pre>

- Hanya menerima nilai parameter URL yang sesuai (* dengan Regular Expression)

Controller
<pre>\$app->get('/pegawai/:id', function(\$id) use(\$app, \$ctr) { echo \$id; // hanya menerima URL dengan bagian :id berupa angka })->conditions(array('id' => '[\d]+'));</pre>

- Grouping URL

Controller
<pre>\$app->group('/api', function() use(\$app, \$ctr) { // menangkap URL /api/pegawai/:id \$app->get('/pegawai/:id', function(\$id) use(\$app, \$ctr) { echo \$id; // hanya menerima URL dengan bagian :id berupa angka })->conditions(array('id' => '[\d]+')); // menangkap URL /api/pimpinan/:id \$app->post('/pimpinan/:id', function(\$id) use(\$app, \$ctr) { ... }); });</pre>

- Mengubah DATE/DATETIME di database menjadi Tanggal Indonesia

Controller
<pre>... \$ctr->helper('date'); // masukkan helper, panggil sebelum \$ctr->model() ...</pre>
Model
<pre>\$tanggal = datedb_to_tanggal(\$tanggalDabatase, 'd F Y'); // output contoh: 11 Januari 2016 // format tanggal d, F, Y sesuai dengan di http://php.net/manual/en/function.date.php</pre>

- Mendapatkan parameter GET, POST, PUT di Model

Model
<pre>// method GET \$value = \$this->get_vars(array('nama', 'alamat')); // jika tidak diset, maka terisi NULL // method POST \$value = \$this->post_vars(array('nama', 'alamat')); // method PUT \$value = \$this->put_vars(array('nama', 'alamat'));</pre>

- Controller output berupa JSON

Controller
<pre>json_output(\$app, \$data); // data berupa array untuk json</pre>

- Menghentikan aplikasi dengan error 40x, pilih salah satu

Controller
<pre>halt_400(\$app); // bad request halt_401(\$app); // unauthorized halt_403(\$app); // forbidden halt_404(\$app); // not found</pre>

-