PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

RUBY ON RAILS HANDBOOK

SESSION 2

1. RAILS

Rails adalah framework aplikasi web open source yang dapat mengoptimalkan bahasa pemograman Ruby.

Rails dirancang untuk membuat sebuah aplikasi web lebih mudah. Rails memungkinkan kita untuk menulis atau mebuat sebuah kode menjadi lebih simple dibandingankan menggunakan framework lainnya.

2. Architecture of Rails application (MVC Concept)

2.1 Model

Model berfungsinya untuk menghubungkan database dengan aplikasi. Oleh karena itu semua manipulasi yang akan kita lakukan terhadap data yang akan kita pakai sebaiknya ditulis dalam model. Misalnya, untuk mem-validasi sebuah username agar tidak lebih dari 30 karakter, validasi harus dilakukan dalam model. Agar setiap data yang akan masuk kedalam database otomatis tervalidasi dan tidak mungkin terlewat.

2.2 View

View merupakan bagian yang fungsinya untuk membuat user interface. Misalnya HTML, XML, AJAX dll. Frontend Interface dibuat dalam view. View ini hanya bertugas untuk menampilkan data yang telah disiapkan dalam controller dulu sebelumnya.

2.3 Controller

Controller merupakan bagian yang fungsinya menerima "event" dari luar (user input) dan bertugas untuk menghubungkan model dengan view. Misalnya, untuk mengambil data sebuah user "A". Ambil datanya dari model, siapkan di controller, lalu view bertugas menampilkannya dalam halaman web.

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

3. Rails Installations

3.1 Install Gem Rails

Dalam membuat sebuah aplikasi Ruby on Rails pastikan gem rails sudah terinstall, contoh :

Install Rails 3.2.14

gem install rails -v=3.2.14

3.2 Create Application

Berikut step by step dalam setup aplikasi Ruby On rails :

1. Untuk membuat awal sebuah apliaksi Ruby on Rails, jalankan perintah di bawah ini di terminal :

   rails new {nama_project} –d mysql

2. Setting konfigurasi database di, config/database.yml :

```
development:
  adapter: mysql2
  encoding: utf8
  reconnect: false
  database: ror_training
  pool: 5
  username: root
  password:
  socket: /var/run/mysqld/mysqld.sock
```

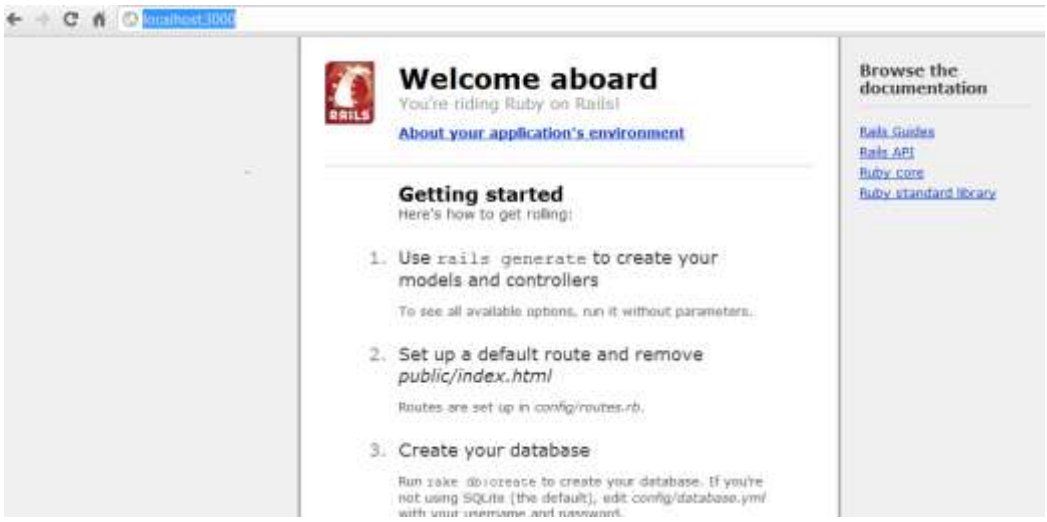
3. Sebelum membuka aplikasi Ruby On Rails di browser, jalankan perintah dibawah ini, untuk mengaktifkan server Rails :

   rails s (in app directory) atau

   rails s -p <port number> (in app directory)

4. Buka di browser dengan url http://localhost:3000, maka akan tampil :

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id



4. Active Record Introdaction

ActiveRecord adalah ORM(Object Relational Mapping) yang disediakan oleh Rails. Fungsinya untuk menjambatani RDBMS dan OO(Object Oriented). Dengan kata lain Active Record adalah M (Model) di MVC, mengapa disebut demikian, karena model merupakan sebuah lapisan dari struktur MVC yang bertanggung jawab dalam proses logika dan bisnis data.

Object Relational Mapping atau sering disebut ORM, adalah teknik yang menghubungkan sebuah objek didalam aplikasi dengan tabel yang terdapat didalam sebuah relasi sistem manajemen database. Dengan menggunakan ORM, hubungan atau sifat objek dalam sebuah aplikasi dapat dengan mudah disimpan dan diambil dari database tanpa perlu membuat kode SQL secara langsung.

Active Record mempunya beberapa mekanisme, yaitu :

1. Mewakili sebuah model dan tabel di database (data).
2. Mewakili asosiai antar model
3. Mewakili hirarki antar model yang saling terkait.
4. Validasi model sebelum data akan masuk ke database.
5. Melakukan operasi database dalam mode berorientasi objek

Dengan adanya ActiveRecord dalam Rails, maka :

Tables di Database > menjadi Class model

Rows di Database > menjadi Objects dari class model tersebut

Columns di rows > menjadi Attributes / Methods dalam object tersebut

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

ex:

* tabel users --> model User
* 1 record tabel di users --> user = select * from users limit 1 (Object)
* name (field table users) --> user.name

4.1 Active Record Naming Conventions

Dengan menggunakan Active Record terdapat sebuah aturan dalam penulisan sebuah penamaan model dan database. Bila menggunakan nama kelas yang terdiri dari dua kata atau lebih, nama kelas model harus mengikuti konvensi Ruby, menggunakan formulir CamelCase, sedangkan nama tabel harus berisi kata-kata yang dipisahkan dengan garis bawah. contoh:

- **Database Tabel** - Plural dengan garis bawah yang memisahkan kata (misalnya, book_clubs)
- **Model Class** - Singular dengan huruf pertama dari setiap kata dikapitalisasi (misalnya, bookclub)

4.2 Schema Conventions

Active Record menggunakan aturan penamaan untuk sebuah kolom dalam tabel database.

- **Foreign keys** - Field ini harus diberikan sebuah nama dengan mengikuti pola singularized_tabel_name_id (ex. user_id, article_id), dengan penamaan seperti ini active record akan otomatis mencari sebuah tabel yang memiliki relasi
- **Primary keys** - Secara default, Active record akan menggunakan integer untuk kolom id sebagai primary key table.

5. Action Controller

Controller atau ActionController fungsinya adalah untuk menjembatani antara view dan model. Artinya di dalam controller ini kita harus mempersiapkan semua data dari database yang akan di tampilkan di dalam view. Untuk menampungnya tentu kita membutuhkan variable, di dalam controller kita bisa menggunakan 2 jenis variable :

1. instance variable/global variable, cirinya : diawali dengan tanda @,
   contoh @product, @products.

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

sesuai namanya variable ini dapat kita gunakan di dalam method lain dan juga dapat dibawa ke dalam view.

2. local variable, contoh : product, products (tanpa @)
   variable ini hanya bisa dipakai didalam method yang membuatnya dan tidak bisa dibawa ke dalam view.

Selain itu juga controller berfungsi sebagai penyimpan daftar halaman yang dapat diakses oleh user.
Contoh sebuah controller dengan nama products :

```
class ProductsController < ApplicationController
  def index
   @welcome =  "Hello, I am from controller"
 end

 def show
   @product =  Product.first # first -> attribute untuk mengambil data pertama
 end

end
```

Note : sebuah controller pasti akan berelasi dengan sebuah folder view yang namanya sama dengan nama controller yang bersangkutan.
Contoh : controller dengan nama products (app_name/app/controllers/products_controller.rb) pasti akan memiliki folder view dengan nama products (app_name/app/views/products).
Setiap method yang ada di dalam controller pasti akan memberikan respond/tampilan untuk user.

Berdasarkan contoh controller di atas maka variable @welcome dapat digunakan di dalam html yang dibuat ada folder app_name/app/views/products/index.html.erb.
Karena method index tersebut tidak me-render atau redirect ke halaman lain maka kita harus menyiapkan halaman dengan nama index.html.erb.

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

6. Action View

ActionView atau view ini adalah halaman yang ber-basic html sehingga kita dapat isi halaman view ini dengan kode-kode html.

Untuk dapat menggunakan kode-kode rails di dalam view ini kita harus menggunakan tag *<%= %>.*

Contoh : berdasarkan kode diatas, maka untuk menampilkan isi variable *@welcome* di dalam view (app/views/products/index.html.erb.) dengan cara :

```
<%= @welcome %>
```

Selain tag *<%= %>*, di dalam view kita bisa menggunakan tag *<% %>* tanpa 'sama dengan' (=). Perbedaannya adalah jika menggunakan tag yang menggunakan tanda = artinya apa yang ada di hasilkan dalam tag tersebut akan di tampilkan ke interface, jika tanpa tanda = artinya apa yang dihasilkan oleh tag tersebut tidak akan di tampilkan ke interface. Tag tanpa = biasa dipakai untuk logic proses, contoh :

```
<% [2,4,6,8,4,2,5].each do |number| %>
  <%= number %>
<% end %>
```

6. Database Migration

Migration adalah sebuah fungsi yang disediakan oleh rails untuk mempermudah developer dalam memanage struktur databasenya. Semua perubahan struktur dalam database seperti penambahan table, kolom, perubahan datatype WAJIB dilakukan menggunakan migration untuk mempermudah tracking history.

6.1 Create Migration

Migration ini dapat di buat dengan 2 jenis generator, yaitu :

1. *rails g model article*

   Perintah di atas sebetulanya untuk men-generate sebuah model. Tapi harap di ingat sebuah model biasanya berelasi dengan sebuah table, sehingga pada waktu kita membuat model kita di haruskan membuat table nya juga.

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

Perintah ini akan menghasilkan beberapa file seperti berikut :

```
    create    db/migrate/20111213071626_create_articles.rb
    create    app/models/article.rb
    invoke    test_unit
    create      test/unit/article_test.rb
    create      test/fixtures/articles.yml
```

yang di maksud dengan flle migration adalah *20111213071626_create_articles.rb* file ini selalu disimpan di folder app_name/db/migrate isi dari file ini adalah :

```
class CreateArticles < ActiveRecord::Migration
 def change
  create_table :articles do |t|
   t.timestamps
  end
 end
end
```

Didalam file ini kita daftarkan column-column yang akan dimiiliki oleh table artiles.

```
class CreateArticles < ActiveRecord::Migration
 def change
  create_table :articles do |t|
   t.string :title
   t.text   :content
   t.timestamps
  end
 end
end
```

Pada contoh di atas kita mempersiapkan 2 buah column, title dengan type string dan content dengan type text.

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

2. *rails g migration add_rating_to_articles*

Perintah ini adalah perintah untuk membuat sebuah file migration saja tanpa file modelnya. File migration ini tidak akan terikat dengan suatu table atau model manapun yang artinya kita dapat menamai file migration ini dengan bebas (nama file migration dengan yang dikerjakannya berbeda), TAPI sebaiknya nama file migration mencerminkan apa yang akan di kerjakan oleh file migration tersebut. Isi dari file migration adalah:

```
class AddRatingToArticles < ActiveRecord::Migration
 def up
 end

 def down
 end
end
```

File migration ini berbeda dengan file migration yang di hasilkan oleh dari generator model. Pada dasarnya sebuah file migration memiliki 2 buah method di dalamnya (up dan down), tapi sekarang kita memiliki 1 buah method baru yaitu change. Method change ini sebetulnya memiliki method down juga tapi rails yang akan membuatkannya untuk kita.
Bolehkah kita selalu menggunakan method change di file migration?Jawabannya adalah boleh, tapi method change ini memiliki kelemahan yaitu tidak bisa di pakai untuk me-remove column (untuk men-remove sebuah column kita harus menggunakan method up dan down).

Berikut adalah contoh file migration yang sudah dilengkapi :

```
class AddRatingToArticles < ActiveRecord::Migration
 def up
  add_column :articles, :rating, :integer
 end

 def down
  remove_column :articles, :rating
 end
end
```

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

atau

```
class AddRatingToArticles < ActiveRecord::Migration
 def change
  add_column :articles, :rating, :integer
 end
end
```

Pada contoh di atas kita akan menambah sebuah column baru, rating dengan type integer ke table articles.

Note : Sebuah file migration pasti diawali dengan bebrapa digit angka, itu adalah versionnya. Angka itu di dapat dari waktu file tersebut di-generate, jadi sudah pasti angka tersebut unik.

6.2 Run Migration

Apakah semua file migration ini akan otomatis akan menghasilkan/memanipulasi table yang ada?Tentu saja tidak, kita harus mengeksekusi file migration tersebut.

Berikut adalah command-command yang membantu kita dalam mengeksekusi file migration :

1. *rake db:migrate*

   --> menjalankan seluruh file migration yang belum di eksekusi/dijalankan dan mencatatnya ke schema.rb dan table schema_migrations yang ada di databases yang bersangkutan (table ini hanya mencatat version mana saja yang sudah dijalankan).

2. *rake db:rollback*

   --> memundurkan migration ke satu versi sebelumnya. Dan file tersebut dianggap belum di eksekusi. Sebetulnya pada waktu kita menjalankan perintah ini, file migration bagian method down yang dieksekusi, oleh karena itu method down wajib di isi kecuali kita menggunakan method change.

3. *rake db:rollback STEP=n*

   --> memundurkan migration ke n version sebelumnya. Dan file-file tersebut dianggap belum di eksekusi.

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

4. *rake db:migrate:down VERSION=20080601000010*

   --> menjalankan method down yang ada di file migration dengan version 20080601000010.

5. *rake db:migrate:up VERSION=20080906120000*

   --> menjalankan method up yang ada di file migration dengan version 2008096120000.

6.3 General Manipulation Commands

Berikut adalah command-command yang dapat digunakan dalam file migration.

1. *drop_table :products*

   --> untuk menghapus tabel dengan nama products.

2. *add_column :users, :time_birthday_at, :datetime*

   --> menambahkan field time_birthday_at dengan type datetime pada table users.

3. *remove_column :products, :price*

   --> menghapus kolom price di tabel products.

4. *change_column :users, :age, :integer*

   --> merubah tipe field age di table users menjadi integer.

5. *rename_column :users, :umur, :age*

   --> merubah nama field dari 'umur' menjadi 'age' di table users.

6. *add_index :products, :part_number*

   --> menambah index part_number di tabel products.

7. *remove_index :products, :part_number*

   --> menghapus index part_number di tabel products.

PT Walden Global Services
Jl. Sukarno-Hatta no. 104
Bandung 40222 Indoensia
www.wgs.co.id

6.4 Various of Datatypes

Datatype yang akan terbaca dalam migration adalah :

- *:string,*
- *:text,*
- *:integer,*
- *:float,*
- *:decimal,*
- *:datetime,*
- *:timestamp,*
- *:time,*
- *:date,*
- *:binary,*
- *:boolean.*