Upload image dan rotate image

```
package main

import (
	"fmt"
	"image"
	"image/color"
	"image/jpeg"
	"net/http"
)

var (
	images = make(map[string]image.Image)
)

func HandleRoot(w http.ResponseWriter, r *http.Request) {
	fmt.Fprintf(w, `
		<html><body>
			<form method="post" enctype="multipart/form-data"
action="/upload" name="upload">
				<label for="file">Choose image:</label>
				<input type="file" name="image" /><br/>
				<input type="submit" name="submit"
value="Upload" />
			</form>
		</body></html>
	`)
}

func HandleUpload(w http.ResponseWriter, r *http.Request) {
	file, header, _ := r.FormFile("image")
	image, _, _ := image.Decode(file)
	images[header.Filename] = image
	http.Redirect(w, r, "/editor?name="+header.Filename, 303)
}

func HandleImage(w http.ResponseWriter, r *http.Request) {
	imageName := r.FormValue("name")
	image := images[imageName]
	jpeg.Encode(w, image, &jpeg.Options{Quality:
jpeg.DefaultQuality})
}

func HandleEditor(w http.ResponseWriter, r *http.Request) {
	fmt.Fprintf(w, `
		<html><body>
			<a href="/">Home</a><br/>
```

```
                <img src="/image?name=%s" width="300" /><br/>
                Choose action:
                     <a href="/invert?name=%s">Invert colors</a>
                    <a href="/rotate?name=%s">Rotate clockwise</a>
            </body></html>
     `, r.FormValue("name"), r.FormValue("name"),
r.FormValue("name"))
}

func invert(c color.Color) color.Color {
     R, G, B, A := c.RGBA()
     return color.NRGBA{R: 255 - uint8(R), G: 255 - uint8(G), B:
255 - uint8(B), A: uint8(A)}
}

func HandleInvert(w http.ResponseWriter, r *http.Request) {
     name := r.FormValue("name")
     fmt.Println("Starting inversion: ", name)
     i := images[name]
     inverted := image.NewRGBA(i.Bounds())
     for y := i.Bounds().Min.Y; y < i.Bounds().Max.Y; y++ {
          for x := i.Bounds().Min.X; x < i.Bounds().Max.X; x++ {
               inverted.Set(x, y, invert(i.At(x, y)))
          }
     }
     images[name] = inverted.SubImage(i.Bounds())
     defer http.Redirect(w, r, "/editor?name="+name, 303)
}

// untuk memutar image
func HandleRotate(w http.ResponseWriter, r *http.Request) {
     name := r.FormValue("name")
     fmt.Println("Starting rotation: ", name)
     i := images[name]
     newBounds := image.Rectangle{
          i.Bounds().Min,
          image.Point{i.Bounds().Max.Y, i.Bounds().Max.X},
     }
     rotated := image.NewRGBA(newBounds)
     for y := i.Bounds().Min.Y; y < i.Bounds().Max.Y; y++ {
          for x := i.Bounds().Min.X; x < i.Bounds().Max.X; x++ {
               rotated.Set(y, x, i.At(x, newBounds.Max.X-y))
          }
     }
     images[name] = rotated.SubImage(newBounds)
     http.Redirect(w, r, "/editor?name="+name, 303)
}

func main() {
     http.HandleFunc("/", HandleRoot)
```

```
	http.HandleFunc("/upload", HandleUpload)
	http.HandleFunc("/image", HandleImage)
	http.HandleFunc("/invert", HandleInvert)
	http.HandleFunc("/editor", HandleEditor)
	http.HandleFunc("/rotate", HandleRotate)

	http.ListenAndServe(":8000", nil)
}
```