



# **Praktik Kerja Lapangan Universitas Muhammadiyah Prof. Dr Hamka 2023**

## **LAPORAN PERKEMBANGAN WEBSITE PKL**

*Disusun Oleh :*

Nama : Bintang Gede Hartono

NISN : 0055863048

Kelas : XII Rekayasa Perangkat Lunak 2

Berikut adalah Laporan mengenai Perkembangan Website Praktik Kerja Lapangan (PKL) UHAMKA per Kamis, 12 Oktober 2023 yang memiliki dua role, admin & user dengan fitur sebagai berikut:

#### Non-User Function: Login

##### I. User

- a) Index (Halaman Utama, Syarat PKL, Alur PKL)
- b) Profile User
- c) Edit Profile User
- d) Logout

##### II. Admin

- a) Index (Halaman Utama, Syarat PKL, Alur PKL)
- b) Profile Admin
- c) Edit Profile Admin
- d) Data Peserta PKL
- e) Regist User/Admin
- f) Edit & Delete Data User/Admin
- g) Logout

#### Upcoming Features:

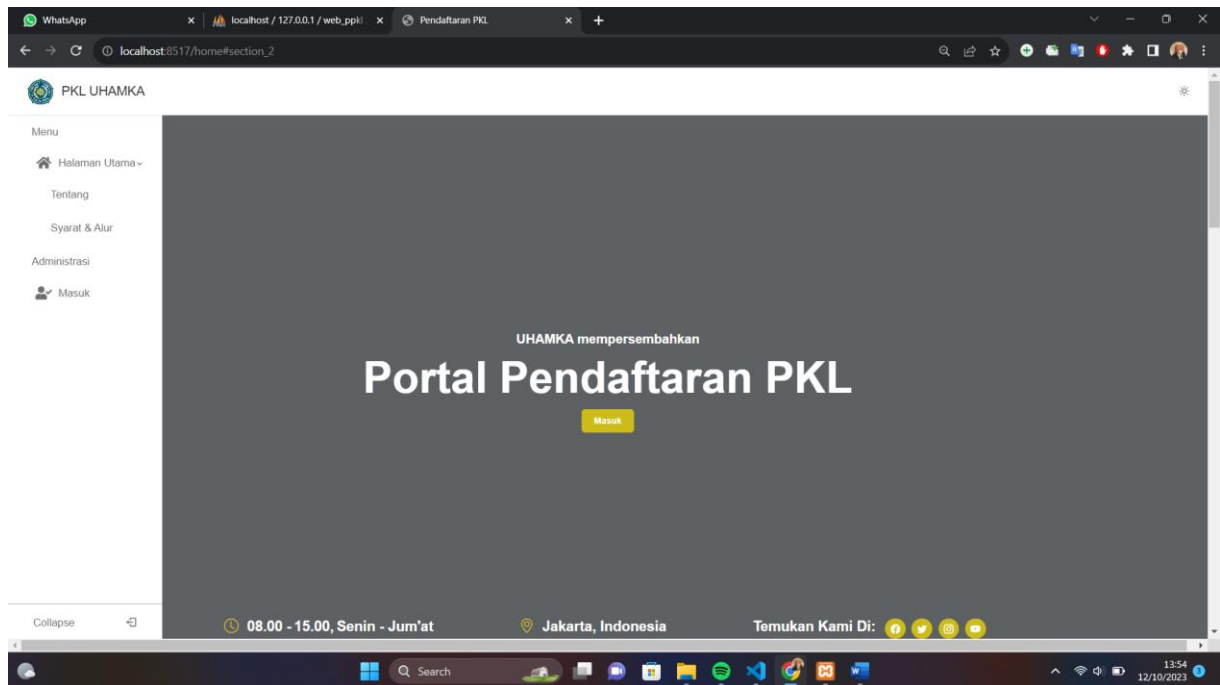
- a) View The Uploaded File:
  - User: Melihat file/surat yang ia upload serta status surat tersebut (diterima/ditolak/pending)
  - Admin: Melihat dan memberikan respon atas surat yang masuk dari user (accepted/rejected/pending)

#### Part-Dividing:

Bintang: All function in User, backing up in admin func

Najwa: All function in Admin, backing up in user func

## 1. Index



WhatsApp localhost / 127.0.0.1 / web\_pkl / Pendaftaran PKL

localhost:8517/home#section\_3

## PKL UHAMKA

### "Kembangkan Kemampuan Diri dengan Magang di UHAMKA"

### Alur Magang

NO	Kegiatan	Diagram Alur			Mutu Baku	Output
		Komisi Magang	Maha siswa	Pembimbing Perusahaan / supervisor		
1	Pendaftaran				Perencanaan UHAMKA ac 1 1 minggu	Materi magang Daftar Kandidat Terpilih
2	Seleksi oleh Perusahaan				Surat pengantar UHAMKA ac 1 1 minggu	
3	Observasi kelengkapan syarat administrasi & Orientasi				Surat Pengantar	
4	Pelaksanaan Magang				Dokumen Hasil seleksi	3 - 6 bulan Proyek dan program magang
5	Monitoring & Evaluasi oleh pembimbing DU/DI				Bukti bayar SPP Bukti lolos seleksi	4 x 1 bulan Laporan kemajuan proyek & Penilaian
6	Mahasiswa Selesai Magang					1 hari Surat Selesai Magang
7	Mahasiswa Mengupload Surat Keterangan Selesai Magang					1 hari

13:56 12/10/2023

WhatsApp localhost / 127.0.0.1 / web\_pkl / Pendaftaran PKL

localhost:8517/home#section\_3

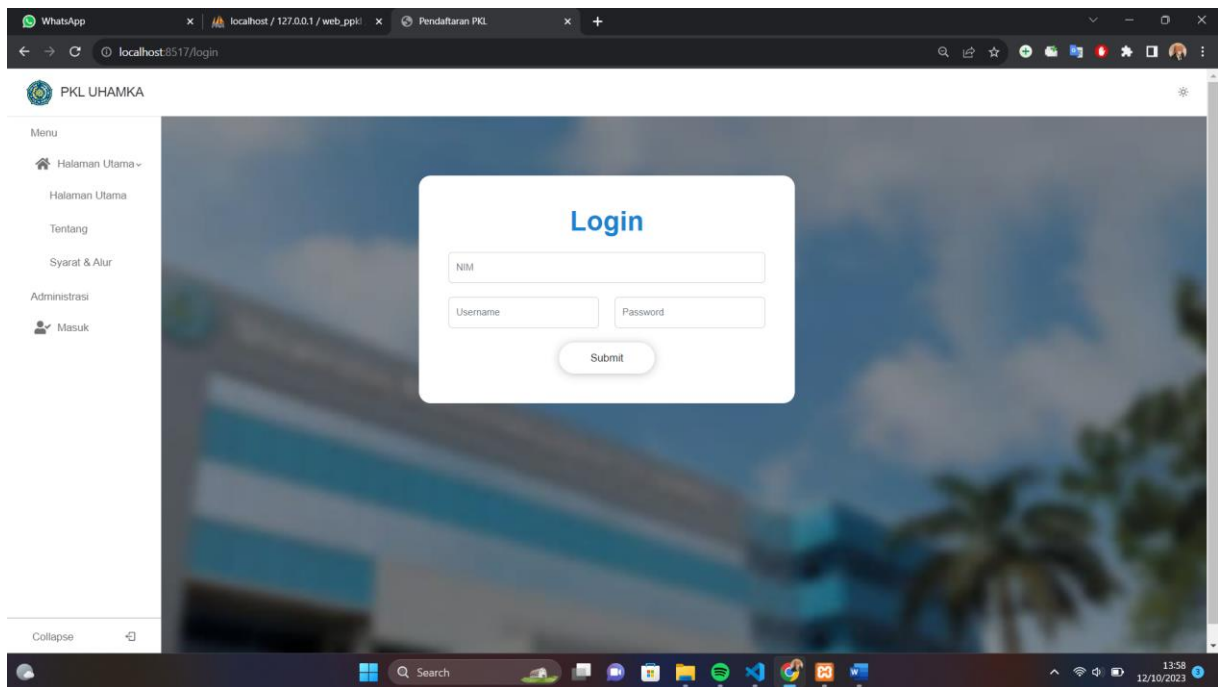
## PKL UHAMKA

### Alur Magang

NO	Kegiatan	Diagram Alur			Mutu Baku	Output
		Komisi Magang	Maha siswa	Pembimbing Perusahaan / supervisor		
1	Pendaftaran				Perencanaan UHAMKA ac 1 1 minggu	Materi magang Daftar Kandidat Terpilih
2	Seleksi oleh Perusahaan				Surat pengantar UHAMKA ac 1 1 minggu	
3	Observasi kelengkapan syarat administrasi & Orientasi				Surat Pengantar	
4	Pelaksanaan Magang				Dokumen Hasil seleksi	3 - 6 bulan Proyek dan program magang
5	Monitoring & Evaluasi oleh pembimbing DU/DI				Bukti bayar SPP Bukti lolos seleksi	4 x 1 bulan Laporan kemajuan proyek & Penilaian
6	Mahasiswa Selesai Magang					1 hari Surat Selesai Magang
7	Mahasiswa Mengupload Surat Keterangan Selesai Magang					1 hari

13:57 12/10/2023

## 2. Login

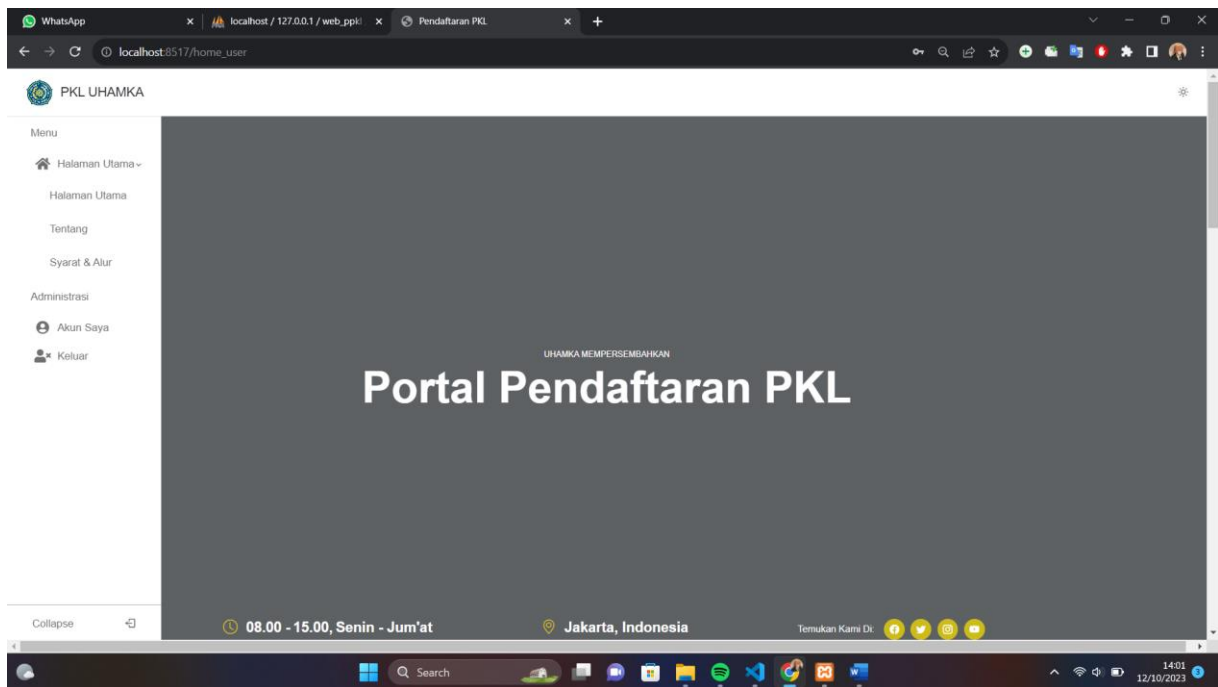


- Fungsi Login

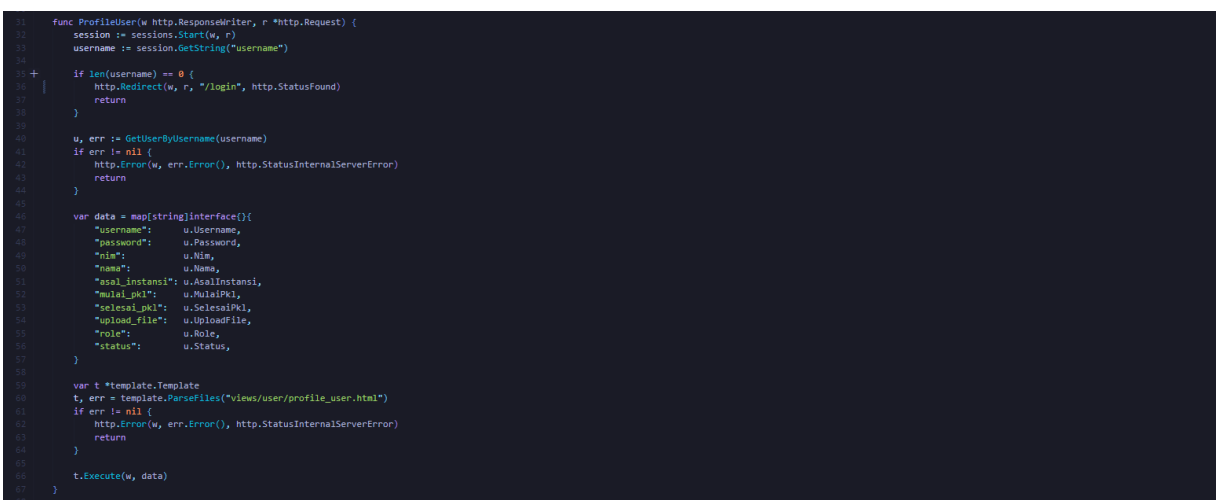
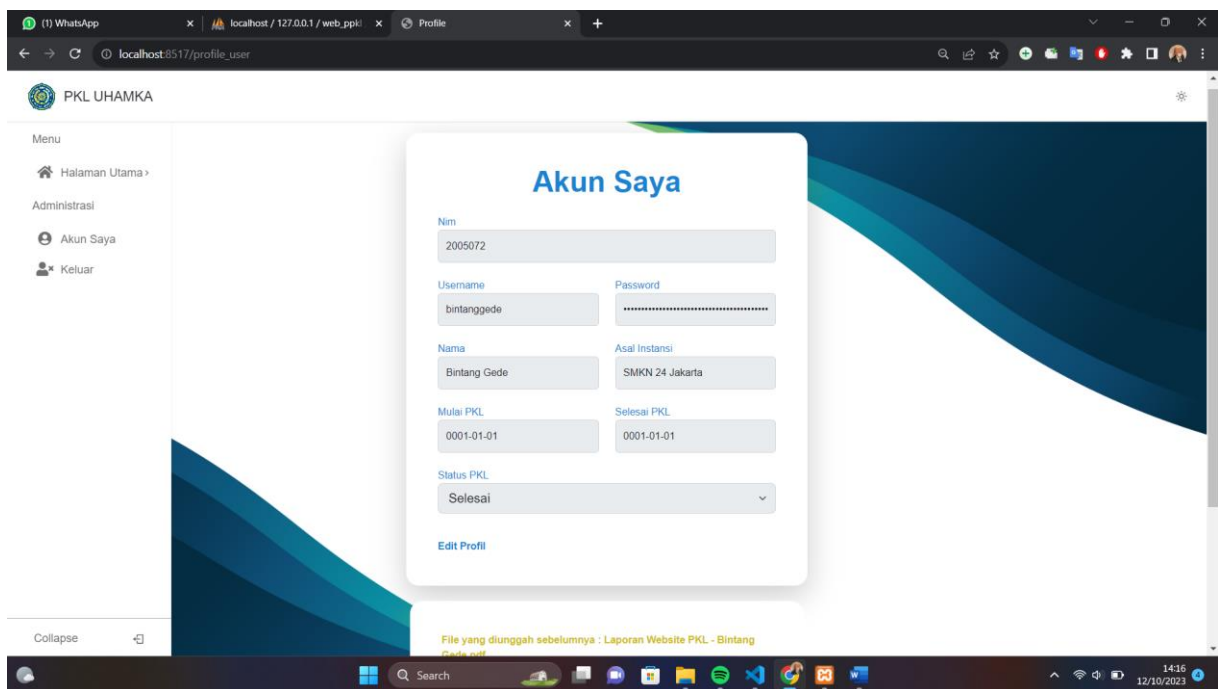
```
1 package logincontroller
2
3 import (
4     "fmt"
5     "net/http"
6     "text/template"
7
8     "github.com/kataras/go-sessions"
9     "github.com/naajwa28/web_organized/entities"
10    "golang.org/x/crypto/bcrypt"
11)
12
13 func login(w http.ResponseWriter, r *http.Request) {
14     session := sessions.Start(w, r)
15     if len(session.GetString("username")) != 0 {
16         http.Redirect(w, r, "/", http.StatusFound)
17         return
18     }
19     if r.Method != "POST" {
20         ttmpl, err := template.ParseFiles("views/home/login.html")
21         if err != nil {
22             http.Error(w, err.Error(), http.StatusInternalServerError)
23             return
24         }
25         ttmpl.Execute(w, nil)
26         return
27     }
28     username := r.FormValue("username")
29     password := r.FormValue("password")
30     fmt.Println(username)
31     fmt.Println(password)
32
33     users := entities.QueryUser(username)
34     admins := entities.QueryAdmin(username)
```

```
35
36     var passwordMatch bool
37     var role int
38
39     if (entities.User()) != users {
40         passwordMatch = (bcrypt.CompareHashAndPassword([]byte(users.Password), []byte(password)) == nil)
41         role = users.Role
42     } else if (entities.Admin()) != admins {
43         passwordMatch = (bcrypt.CompareHashAndPassword([]byte(admins.Password), []byte(password)) == nil)
44         role = 1
45     }
46
47     if passwordMatch {
48         // login success
49         session := sessions.Start(w, r)
50         session.Set("username", username)
51         session.Set("role", role)
52         if role == 1 {
53             // admin login
54             http.Redirect(w, r, "/home_admin", http.StatusSeeOther)
55             fmt.Println("Sukses")
56         } else {
57             // user login
58             http.Redirect(w, r, "/home_user", http.StatusSeeOther)
59             fmt.Println("Sukses")
60         }
61     } else {
62         // login failed
63         fmt.Println("Gagal, username atau password salah")
64         fmt.Println(w, "Gagal")
65         http.Redirect(w, r, "/login", http.StatusFound)
66     }
67 }
68 }
```

### 3. User



- Profil User





- Ubah Profil User

PKL UHAMKA

Menu

- Halaman Utama

Administrasi

- Akun Saya
- Keluar

Collapse

## Ubah Profil

NIM  
2005072

Username  
bintanggede

Password  
Password/Password baru

Nama  
Bintang Gede

Asal Instansi  
SMKN 24 Jakarta

Mulai PKL  
0001-01-01

Selesai PKL  
0001-01-01

Status PKL  
Selesai

File yang diunggah sebelumnya : Laporan Website PKL - Bintang Gede.pdf  
(Unggah file sebelumnya jika tidak ingin mengunggah file baru)

Choose File No file chosen

- Fungsi Ubah User

```

70 func EditProfileUser(w http.ResponseWriter, r *http.Request) {
71     db := config.Connect_db()
72     session := sessions.Start(w, r)
73     username := session.GetString("username")
74
75     if len(username) == 0 {
76         http.Redirect(w, r, "/profile_user", http.StatusFound)
77         return
78     }
79     u, err := GetUserByUsername(username)
80     if err != nil {
81         http.Error(w, err.Error(), http.StatusInternalServerError)
82         return
83     }
84
85     if r.Method == http.MethodPost {
86         newUsername := r.FormValue("new_username")
87         newPassword := r.FormValue("new_password")
88
89         if newUsername != "" {
90             _, err := db.Exec("UPDATE users SET username = ? WHERE username = ?", newUsername, username)
91             if err != nil {
92                 http.Error(w, err.Error(), http.StatusInternalServerError)
93                 return
94             }
95             username = newUsername
96             session.Set("username", newUsername)
97         }
98
99         if newPassword != "" {
100             hashedPassword, err := bcrypt.GenerateFromPassword([]byte(newPassword), bcrypt.DefaultCost)
101             if err != nil {
102                 http.Error(w, err.Error(), http.StatusInternalServerError)
103                 return
104             }
105             _, err = db.Exec("UPDATE users SET password = ? WHERE username = ?", hashedPassword, username)
106             if err != nil {
107                 http.Error(w, err.Error(), http.StatusInternalServerError)
108                 return
109             }
110         }
111     }
112 }

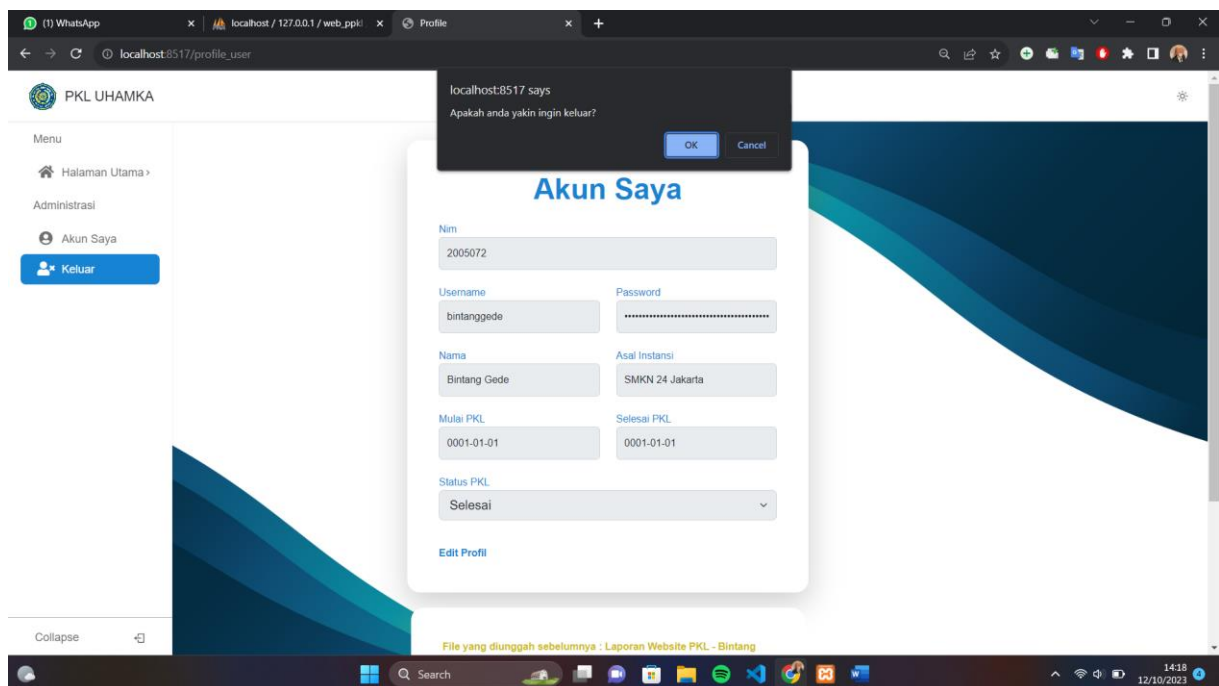
```

```

113 var data = map[string]interface{}{
114     "username": u.Username,
115     "password": u.Password,
116     "nim":      u.Nim,
117     "nama":     u.Name,
118     "asal_instansi": u.AsalInstansi,
119     "mulai_pkl": u.MulaiPKL,
120     "selesai_pkl": u.SelesaiPKL,
121     "upload_file": u.UploadFile,
122     "role":      u.Role,
123     "status":    u.Status,
124 }
125
126 var t *template.Template
127 t, err = template.ParseFiles("views/user/edit_profile_user.html")
128 if err != nil {
129     http.Error(w, err.Error(), http.StatusInternalServerError)
130     return
131 }
132
133 t.Execute(w, data)
134 }
135 }

```

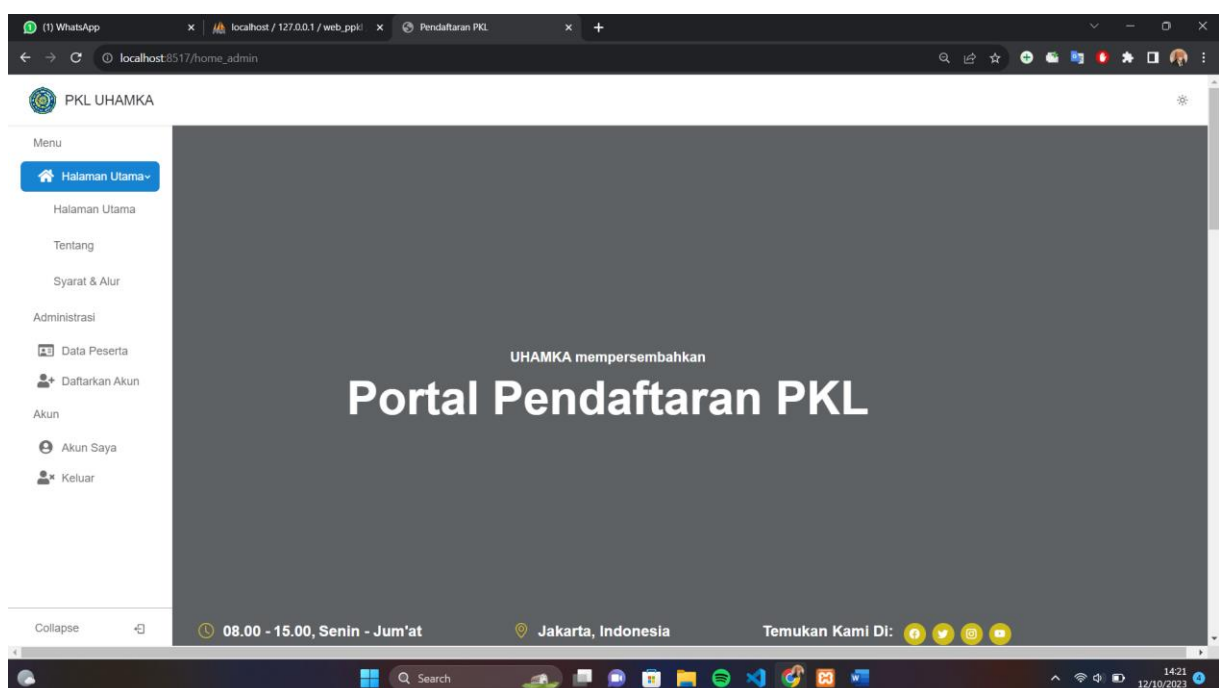
## 4. Logout



- Fungsi Logout

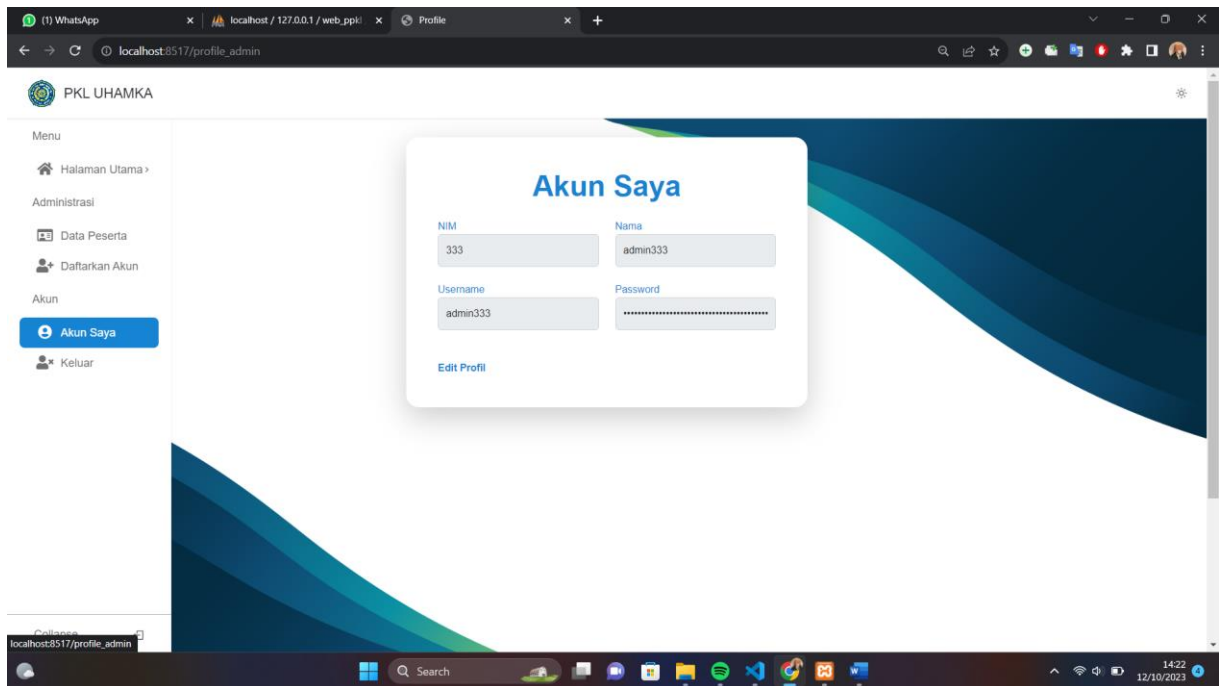
```
1 package logoutcontroller
2
3 import (
4     "net/http"
5
6     "github.com/kataras/go-sessions"
7 )
8
9 func Logout(w http.ResponseWriter, r *http.Request) {
10     session := sessions.Start(w, r)
11     session.Clear()
12     sessions.Destroy(w, r)
13     http.Redirect(w, r, "/home", http.StatusFound)
14 }
15 }
```

## 5. Admin



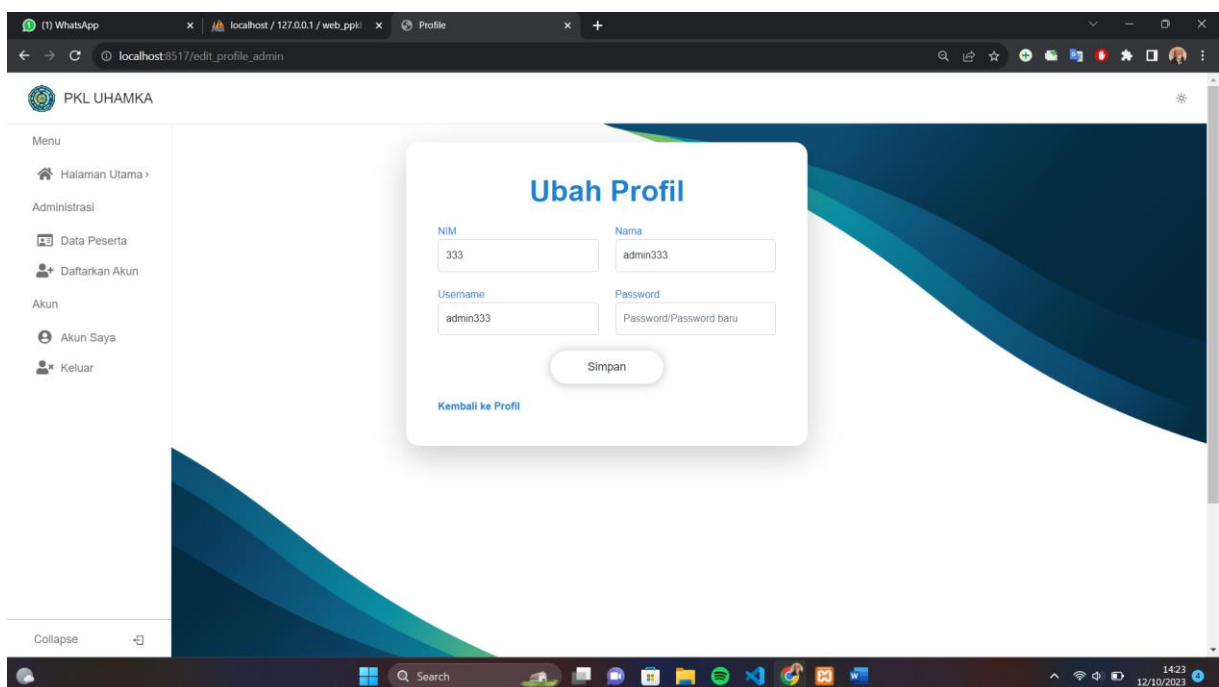


- Profil Admin



```
139 func ProfileAdmin(w http.ResponseWriter, r *http.Request) {
140     session := sessions.Start(w, r)
141     username := session.GetString("username")
142
143     if len(username) == 0 {
144         http.Redirect(w, r, "/login", http.StatusFound)
145         return
146     }
147     u, err := GetUserByUsername(username)
148     if err != nil {
149         http.Error(w, err.Error(), http.StatusInternalServerError)
150         return
151     }
152
153     var data = map[string]interface{}{
154         "username": u.Username,
155         "password": u.Password,
156         "nim":      u.Nim,
157         "nama":     u>Nama,
158     }
159
160     var t *template.Template
161     t, err = template.ParseFiles("views/admin/profile_admin.html")
162     if err != nil {
163         http.Error(w, err.Error(), http.StatusInternalServerError)
164         return
165     }
166
167     t.Execute(w, data)
168 }
```

- Ubah Profil Admin



- Fungsi Ubah Admin

```

190 func EditProfileAdmin(w http.ResponseWriter, r *http.Request) {
191     db := config.Connect_DB()
192     session := sessions.Start(w, r)
193     username := session.GetString("username")
194
195     if len(username) == 0 {
196         http.Redirect(w, r, "/profile_admin", http.StatusFound)
197         return
198     }
199
200     u, err := GetUserByUsername(username)
201     if err != nil {
202         http.Error(w, err.Error(), http.StatusInternalServerError)
203         return
204     }
205
206     if r.Method == http.MethodPost {
207         newPassword := r.FormValue("new_password")
208         newUsername := r.FormValue("new_username")
209
210         if newUsername != "" {
211             _, err := db.Exec("UPDATE users SET username = ? WHERE username = ?", newUsername, username)
212             if err != nil {
213                 http.Error(w, err.Error(), http.StatusInternalServerError)
214                 return
215             }
216             username = newUsername
217             session.Set("username", newUsername)
218         }
219
220         if newPassword != "" {
221             hashedPassword, err := bcrypt.GenerateFromPassword([]byte(newPassword), bcrypt.DefaultCost)
222             if err != nil {
223                 http.Error(w, err.Error(), http.StatusInternalServerError)
224                 return
225             }
226             _, err := db.Exec("UPDATE users SET password = ? WHERE username = ?", hashedPassword, username)
227             if err != nil {
228                 http.Error(w, err.Error(), http.StatusInternalServerError)
229                 return
230             }
231         }
232     }
233
234     var data = map[string]interface{}{
235         "username": u.Username,
236         // "password": u.Password,
237         "nim": u.Nim,
238         "nama": u.Name,
239     }
240
241     var t *template.Template
242     t, err = template.ParseFiles("views/admin/edit_profile_admin.html")
243     if err != nil {
244         http.Error(w, err.Error(), http.StatusInternalServerError)
245         return
246     }
247
248     t.Execute(w, data)
249 }

```

- Daftarkan Akun User

The screenshot shows a web browser window with the URL `localhost:8517/register_admin`. The page title is 'PKL UHAMKA'. On the left, there is a sidebar menu with the following items: 'Menu', 'Halaman Utama', 'Administrasi', 'Data Peserta', 'Daftarkan Akun' (highlighted with a blue button), 'Akun', 'Akun Saya', and 'Keluar'. The main content area displays a form titled 'Daftarkan Akun'. The form contains the following fields and controls:

- NIM: A text input field.
- Username: A text input field.
- Password: A text input field.
- Nama: A text input field.
- Asal Instansi: A text input field.
- Upload Surat: A section with a 'Choose File' button and the text 'No file chosen'.
- Mulai PKL: A date input field showing 'dd/mm/yyyy'.
- Selesai PKL: A date input field showing 'dd/mm/yyyy'.
- Role: A dropdown menu showing the value '2'.
- Status PKL: A dropdown menu showing the value 'Tidak Aktif'.
- Submit: A button at the bottom of the form.

- Fungsi Daftarkan Akun

```

13 func RegisterAdmin(w http.ResponseWriter, r *http.Request) {
14     db := config.Connect_DB()
15
16     if r.Method == "GET" {
17         temp, err := template.ParseFiles("views/admin/register_admin.html")
18         if err != nil {
19             panic(err)
20         }
21         temp.Execute(w, nil)
22         return
23     }
24
25     username := r.FormValue("username")
26     password := r.FormValue("password")
27     nim := r.FormValue("nim")
28     nama := r.FormValue("nama")
29     asal_instansi := r.FormValue("asal_instansi")
30     mulai_pkl := r.FormValue("mulai_pkl")
31     selesai_pkl := r.FormValue("selesai_pkl")
32     upload_file := r.FormValue("upload_file")
33     role := r.FormValue("role")
34     status := r.FormValue("status")
35
36     users := entities.QueryUser(username)
37
38     if (entities.User()) == users {
39         hashedPassword, err := bcrypt.GenerateFromPassword([]byte(password), bcrypt.DefaultCost)
40         if len(hashedPassword) != 0 && util.CheckErr(w, r, err) {
41             stat, err := db.Prepare("INSERT INTO users SET username=?, password=?, nim=?, nama=?, asal_instansi=?, mulai_pkl=?, selesai_pkl=?, upload_file=?, role=?, status=?")
42             if err == nil {
43                 _, err := stat.Exec(&username, &hashedPassword, &nim, &nama, &asal_instansi, &mulai_pkl, &selesai_pkl, &upload_file, &role, &status)
44                 if err != nil {
45                     http.Error(w, err.Error(), http.StatusInternalServerError)
46                     return
47                 }
48                 http.Redirect(w, r, "/data_peserta", http.StatusSeeOther)
49                 fmt.Println("Berhasil mendaftarkan akun")
50                 return
51             }
52         }
53     } else {
54         http.Redirect(w, r, "/register_admin", http.StatusSeeOther)
55         fmt.Println("Gagal mendaftarkan akun (username sudah digunakan)")
56         return
57     }
58 }

```

- Tabel Data Peserta PKL

PKL UHAMKA

Menu

- Halaman Utama
- Administrasi
  - Data Peserta
  - Daftarkan Akun
- Akun
  - Akun Saya
  - Keluar

## Data Peserta PKL

NIM/NISN	Username	Nama	Asal Instansi	Mulai	Selesai	Status	Role	Surat
333	admin333	admin333	smkn 24	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	Admin	
777	user2	suyu	smkn 24	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	User	
444	pkkp	dem	den	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	User	RUNDOW
2005072	bintanggede	Bintang Gede	SMKN 24 Jakarta	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	User	Laporan Website PKL

PKL UHAMKA

Menu

- Halaman Utama
- Administrasi
  - Data Peserta
  - Daftarkan Akun
- Akun
  - Akun Saya
  - Keluar

## Data Peserta PKL

asal Instansi	Mulai	Selesai	Status	Role	Surat	Ubah & Hapus data
smkn 24	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	Admin		Ubah Hapus
smkn 24	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	User		Ubah Hapus
den	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	User	RUNDOWN1.pdf	Ubah Hapus
SMKN 24 Jakarta	0001-01-01 00:00:00 +0000 UTC	0001-01-01 00:00:00 +0000 UTC	Tidak Aktif	User	Laporan Website PKL - Bintang Gede.pdf	Ubah Hapus

- Fungsi GetDataPeserta

```

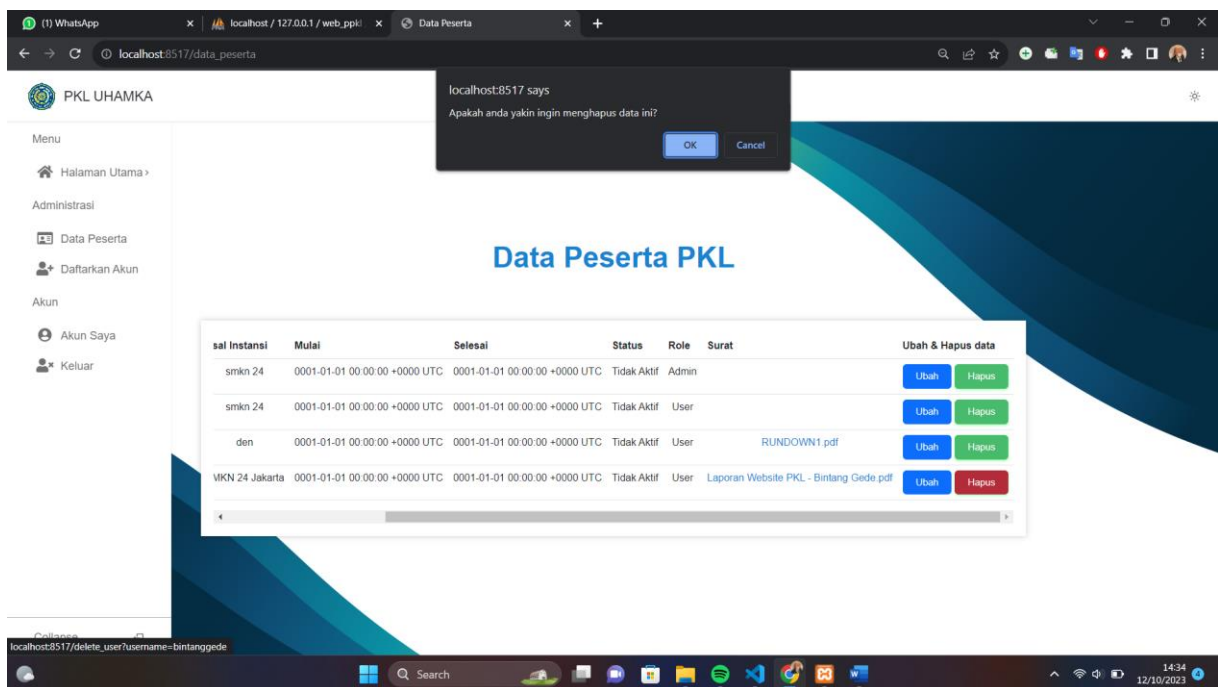
202 func GetDataPeserta(w http.ResponseWriter, r *http.Request) {
203     db := config.Connect_DB()
204     var u entities.User
205     rows, err := db.Query("SELECT * FROM users")
206     if err != nil {
207         log.Fatal(err)
208     }
209     defer rows.Close()
210     usefx := []entities.User{}
211     for rows.Next() {
212         err := rows.Scan(&u.ID, &u.Username, &u.Password, &u.Nim, &u>Nama, &u.AsalInstansi,
213             &u.HasilPkl, &u.SelesaiPkl, &u.UploadFile, &u.Role, &u.Status)
214         if err != nil {
215             log.Fatal(err)
216         }
217         usefx = append(usefx, u)
218     }
219     if err := rows.Err(); err != nil {
220         log.Fatal(err)
221     }
222     tmpl, err := template.ParseFiles("views/admin/data_peserta.html")
223     if err != nil {
224         log.Fatal(err)
225     }
226     data := struct {
227         DtEmp []entities.User
228     }{
229         DtEmp: usefx,
230     }
231     err = tmpl.Execute(w, data)
232     if err != nil {
233         log.Fatal(err)
234     }
235 }

```

- Ubah data user sebagai admin

- Ubah data admin

- Hapus data



- Fungsi hapus data

```

434 func DeleteUser(w http.ResponseWriter, r *http.Request) {
435     db := config.Connect_DB()
436     username := r.URL.Query().Get("username")
437     _, err := db.Exec("DELETE FROM users WHERE username = ?", username)
438     if err != nil {
439         http.Error(w, err.Error(), http.StatusInternalServerError)
440         return
441     }
442     http.Redirect(w, r, "/data_peserta", http.StatusSeeOther)
443 }
444
445
446

```

## 6. Fungsi Main

```

package main

import (
    "log"
    "net/http"

    _ "github.com/go-sql-driver/mysql"
    "github.com/naajwa28/web_organized/config"
    "github.com/naajwa28/web_organized/controller/admincontroller"
    "github.com/naajwa28/web_organized/controller/homecontroller"
    "github.com/naajwa28/web_organized/controller/logincontroller"
    "github.com/naajwa28/web_organized/controller/logoutcontroller"
    "github.com/naajwa28/web_organized/controller/profileadmin"
    "github.com/naajwa28/web_organized/controller/registeradmin"
    "github.com/naajwa28/web_organized/controller/registeruser"
    "github.com/naajwa28/web_organized/models/adminmodels"
    "github.com/naajwa28/web_organized/models/usermodels"
)

func main() {
    // Create connection
    config.Connect_DB()

    http.HandleFunc("/uploads", http.StripPrefix("/uploads", http.FileServer(http.Dir("uploads"))))
    http.HandleFunc("/css", http.StripPrefix("/css", http.FileServer(http.Dir("css"))))
    http.HandleFunc("/js", http.StripPrefix("/js", http.FileServer(http.Dir("js"))))
    http.HandleFunc("/fonts", http.StripPrefix("/fonts", http.FileServer(http.Dir("fonts"))))
    http.HandleFunc("/images", http.StripPrefix("/images", http.FileServer(http.Dir("images"))))
    http.HandleFunc("/video", http.StripPrefix("/video", http.FileServer(http.Dir("video"))))

    // Routes
    // Home
    http.HandleFunc("/home", homecontroller.Home)

    // Admin
    http.HandleFunc("/home_admin", admincontroller.HomeAdmin)
    http.HandleFunc("/data_peserta", admincontroller.GetDataPeserta)
    http.HandleFunc("/edit_user", admincontroller.EditUser)
    http.HandleFunc("/edit_admin", admincontroller.EditAdmin)
    http.HandleFunc("/register_admin", admincontroller.RegisterAdmin)
    http.HandleFunc("/profile_admin", admincontroller.ProfileAdmin)
    http.HandleFunc("/edit_profile_admin", admincontroller.EditProfileAdmin)
    http.HandleFunc("/delete_user", admincontroller.DeleteUser)

    http.HandleFunc("/save_profile_admin", adminmodels.SaveProfileAdmin)
    http.HandleFunc("/save_user", adminmodels.SaveUser)
    http.HandleFunc("/save_admin", adminmodels.SaveAdmin)

    // User
    http.HandleFunc("/home_user", usercontroller.HomeUser)
    http.HandleFunc("/profile_user", usercontroller.ProfileUser)
    http.HandleFunc("/edit_profile_user", usercontroller.EditProfileUser)
    http.HandleFunc("/save_profile_user", usermodels.SaveProfileUser)

    // Logout
    http.HandleFunc("/logout", logoutcontroller.Logout)

    // Login
    http.HandleFunc("/login", logincontroller.Login)

    // Run server
    log.Println("Server running on port: 8517")
    log.Fatal(http.ListenAndServe(":8517", nil))
}

```