

Penerapan Sistem Absensi Kinerja Pengabdian Tri Dharma secara Online pada Website Berbasis Yii Framework

The Application of a System of Absentee Performance Tri Dharma Devotion Online on websites based on Yii Framework

Qurotul Aini¹, Untung Rahardja², Femi Allamiah³

¹Dosen STMIK Raharja Jurusan Sistem Informasi, ²Dosen STMIK Raharja Jurusan Sistem Informasi, ³Mahasiswa STMIK Raharja Jurusan Sistem Informasi.

[¹aini@raharja.info](mailto:aini@raharja.info), [²untung@raharja.info](mailto:untung@raharja.info), [³femi@raharja.info](mailto:femi@raharja.info)

Abstrak

Di era globalisasi saat ini perkembangan teknologi sudah semakin canggih dan mudah diakses. yang di mana semua kebutuhan manusia bisa dilakukan dengan suatu akses yang secara online. Karena perkembangan teknologi semakin canggih maka dari itu pun segala sesuatu yang ada harus bisa dijadikan mudah. Saat ini belum kebanyakan sistem absensi di perguruan tinggi diakses lebih mudah dengan cara online masih dengan sesuatu yang manual dan sering kali bisa terjadi adanya penitipan absensi dengan teman sesama yang menjalani Pengabdian Tri Dharma. Pengabdian Tri Dharma di Perguruan Tinggi memang berbeda-beda salah satunya yang ada di Perguruan Tinggi Raharja ini bahwa ada Tri Dharma di dalam kampus seperti menjadi Asisten Lab. Saat ini penilaian Tri Dharma terkadang kurang objektif karena kurangnya data penilaian yang kurang ditata baik sehingga kurang menjadikan sebuah penilaian menjadi Objektif. Maka dari itu dengan adanya suatu perkembangan yang ada perancangan sebuah sistem absensi untuk kinerja Pengabdian Tri Dharma dengan secara online akan mencegah terjadi adanya kecurangan dalam sebuah penitipan absensi. Dengan adanya nya sistem absensi tersebut bisa diakses dengan mudah dengan hanya scan Qrcode saja. yang akan dirancang sedemikian rupa agar bisa absensi dengan mudah. Dengan menggunakan sebuah Website yang tersedia yang dirancang berbasis Yii Framework, sudah tersedia dengan lebih mudah dengan hanya akses Website Pensil (Penilaian Asisten Lab) guna mempermudah dan memperlancar Pengabdian Tri Dharma Absensi dengan efesien.

Kata Kunci—Absensi Online, Framework, Yii Framework dan Pensil (Penilaian Asisten Lab)

Abstract

In the current globalization era technology development already is getting advanced and easily accessible. That is where all human needs can be done with an online access. Due to the development of increasingly sophisticated technology thus everything else must be made easy. Currently most systems attendance in College are accessed more easily by means of the online manual with something still and often can happen the existence of day care attendance with fellow friends who lead the Tri Dharma Devotion. Tri Dharma devotion in the College indeed varies one existing in college that there's this Raharja Tri Dharma in the campus as a Lab Assistant. The current assessment Tri Dharma sometimes less objective because of lack of data less well-appointed assessment so that an evaluation be made less objective. Thus the existence of a development of the existing design of a system of absentee for the performance of the Tri Dharma Devotion with online will prevent the existence of cheating occurred in a day care attendance. With existence of the attendance system can be accessed easily by just scan

Qrcode only. that will be designed in such a way that the attendance could easily. By using a Website available that are designed based Yii Framework, is already available with easier access the Website with only Pencil (Lab Assistant Assessment) in order to facilitate Attendance by Tri Dharma Devotion efficiently.

Keywords—*Online course, Yii Framework, Framework and a pencil (assessment of Lab Assistant)*

1. PENDAHULUAN

Merancang sebuah sistem Absensi yang dapat diakses dengan mudah dan cepat lalu bisa melihat kinerja siapa yang paling aktif dan pasif. karena dengan sebuah rancangan sistem tersebut maka tidak perlu di khawatirkan lagi akan terjadinya sebuah kecurangan sistem menitipkan absensi kepada sesama kinerja pengabdian tri dharma.

Karena masih dilihat kurang efektif dan efisien jika sebuah penilaian dalam absensi masih manual sedangkan perkembangan jaman sewaktu waktu terus maju dan berkembang dengan baik. Karena sebuah kinerja tri dharma kurang diperhatikan, Sejauh ini sistem absensi didalam perkembangan sebuah penilaian absensi untuk pengguna kinerja tri dharma kurang terlihat penilaianya, dengan adanya perancangan sebuah sistem yang menghasilkan penilaian tri dharma menjadi lebih akurat dimilai maka ada sebuah penilaian untuk kinerja tri dharma.

Di perguruan tinggi sebenarnya sangat membutuhkan sebuah sistem yang segalanya serba cepat untuk dinilai keaktifan nya, karna sebuah pengabdian tri dharma sangat penting sebagai sebuah pembelajaran agar semakin disiplin jika sudah memasuki dunia kerja yang nyata.

Dengan adanya perancangan sebuah sistem absensi yang dapat dilihat keaktifan nya maka dari itu ada sebuah pencapaian yang hanya dengan *scan* menggunakan Qr Code yang tersedia maka akan memudahkan semua pengabdian tri dharma menjadi lebih bertanggung jawab untuk penilaianya itu sendiri.

Berhubung sistem sudah online maka dari itu untuk login sudah bisa dengan menggunakan SSO (*Single Sign On*) agar terlihat lebih cepat dalam mengakses sistem tersebut.

Penelitian ini sudah dilakukan oleh Wulan Lestari [1] dibuatlah *system Single Sign On* (SSO) untuk lebih mempermudah anggota dalam melakukan login tanpa harus menggunakan *password*. Dan dengan merumuskan penggunaan SSO pada register atau login RhjFox hal ini dimungkinkan dengan menggunakan metode SWOT untuk pencapaian sebuah masalah. Diharapkan dengan menggunakan *system Single Sign On* (SSO) dapat lebih mempermudah anggota untuk mengakses maupun mencari informasi kegiatan RhjFox yang ada di database melalui viewbord serta memiliki Rank yang cukup ideal. Hal ini dapat mempermudah pengguna dalam menggunakan aplikasi yang ada, sehingga keamanan data pengguna lebih terjamin.

Penelitian ini sudah dilakukan oleh Fitria Nursetianingsih [2] Adapun beberapa faktor yang mempengaruhi kinerja pegawai diantaranya adalah kepribadian, kemampuan bekerja sama, kedisiplinan dan prestasi kerja. Absensi merupakan salah satu bentuk kedisiplinan pegawai yang juga membantu meningkatkan motivasi disetiap aktivitas perusahaan. *Profesionalisme* adalah komitmen para anggota suatu profesi untuk meningkatkan kemampuannya secara terus menerus. Dan dalam hal ini absensi merupakan salah satu tolok ukur suatu *profesionalisme* pegawai. Seseorang yang memiliki jiwa *profesionalisme* senantiasa mendorong dirinya untuk mewujudkan kerja yang profesional juga. Dalam segi kedisiplinan, data absensi setiap pegawai dapat diolah dan ditampilkan pada *dashboard* yang efektif agar penyajian informasi memberikan kejelasan pada penggunanya. *Dashboard* dapat dijadikan untuk melihat secara visual berupa grafik-grafik yang interaktif berdasarkan data yang ada. Dari grafik tersebut pihak perusahaan dapat melihat performa absensi seluruh pegawai.

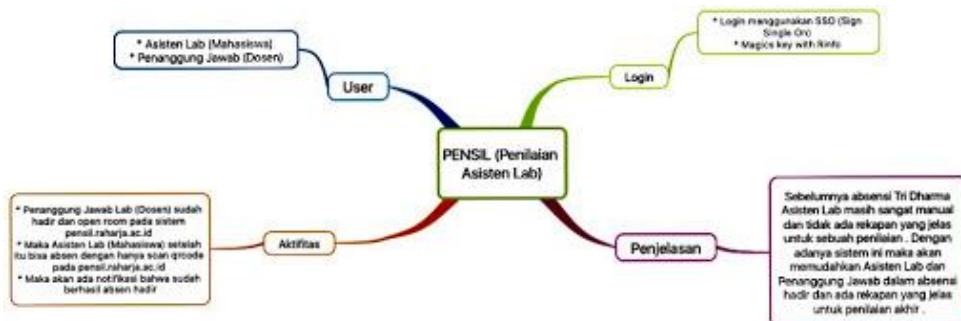
Penelitian ini sudah dilakukan oleh Miftahul Jannah [3] modern seperti sekarang ini, di mana hampir semua orang memiliki *smartphone*-nya sendiri, tidak menutup kemungkinan bahwa *smartphone* tersebut dapat dimanfaatkan untuk sistem absensi di perguruan tinggi. Salah

satu fitur dari *smartphone* yang menarik adalah kamera. Ide yang muncul adalah memanfaatkan QR Code dan kamera pada *smartphone* Android untuk menjadi sistem absensi.

Penelitian ini sudah dilakukan oleh R. Rizal Isnanto [4] Masalah yang terjadi dalam sistem yang telah diterapkan terletak pada pangumpulan data-data hadir mahasiswa yang dilakukan dalam kelas, di mana sistem presensi yang berjalan saat ini masih dilakukan secara manual, yaitu dengan pemanggilan atau pengisian tanda tangan secara langsung. Di era modern seperti sekarang ini, tidak menutup kemungkinan bahwa telepon selular dapat dimanfaatkan untuk sistem presensi di perguruan tinggi. Sebuah ponsel yang memiliki sistem operasi di dalamnya dan dapat mengakses internet adalah salah satu ciri utama dari *smartphone*. Smartphone merupakan pengembangan dari ponsel yang menyediakan fitur-fitur seperti pada komputer. Salah satu fitur dari *smartphone* yang menarik adalah kemampuannya untuk mengambil, menyimpan, serta menampilkan gambar dengan format JPEG karena sebagian besar *smartphone* memiliki kamera. Ide yang muncul adalah untuk memanfaatkan QR Code dan *smartphone* Android untuk menjadi sistem presensi. Dengan memanfaatkan QR Code, data mahasiswa dapat disimpan dalam bentuk gambar QR Code yang kemudian disimpan di dalam ponsel mahasiswa. Selanjutnya gambar dapat digunakan untuk identifikasi mahasiswa pada sistem presensi.

Penelitian ini sudah dilakukan oleh Dahrul Syah [5] Pendidikan merupakan salah satu faktor penting yang dibutuhkan manusia untuk mendapatkan kualitas hidup yang lebih baik. Pendidikan di Indonesia terus berkembang baik dalam kurikulum maupun sarana dan prasarana. Pendidikan dapat ditempuh melalui lembaga pendidikan formal dan informal. Salah satu lembaga pendidikan formal adalah perguruan tinggi. Perguruan tinggi sebagai salah satu pondasi pendidikan di suatu negara memiliki tugas dan tanggung jawab dalam membentuk pemimpin-pemimpin bangsa di masa depan yang memiliki intelektualitas tinggi serta budi pekerti yang luhur. Pencapaian tugas tersebut tidak lepas dari peran serta dan kontribusi sumber daya pendidik serta sarana pendidikan yang juga harus berkualitas.

Penelitian ini sudah dilakukan oleh Siti Ria Zuliana [6] Dalam penyampaian informasi saat ini dibutuhkan sistem yang cepat dan mudah untuk diakses secara online anywhere and anytime yang diberikan harus selalu up to date. Perkembangan internet berkembang begitu pesat dan terus berinovasi dan menciptakan terobosan yang baru. Dalam meningkatkan kualitas sistem pembelajaran pada saat bimbingan belajar berlangsung yaitu sistem layanan absensi scanning qrcode untuk merekam kehadiran mahasiswa dengan baik.



Gambar 1. *Mind Mapping* Penilaian Asisten Lab

Terdapat gambar *Mind Mapping* guna memperlancar semua arah tujuan yang akan dijalankan pada sebuah sistem tersebut. Ada sebuah *User* (1) yang di mana terdapat Asisten Lab (Mahasiswa) dan Penanggung Jawab Lab (Dosen). *Login* (2), Dengan menggunakan SSO (*Single Sign On*) dan Menggunakan *Magics Key with Rinfo*. *Aktifitas* (3) Penanggung Jawab Lab (Dosen) sudah hadir dan *Open Room* pada sistem *pensil.raharja.ac.id*, Maka Asisten Lab (Mahasiswa) Setelah itu berhasil Absensi dengan hanya *Scan Qr Code* pada *pensil.raharja.ac.id*,

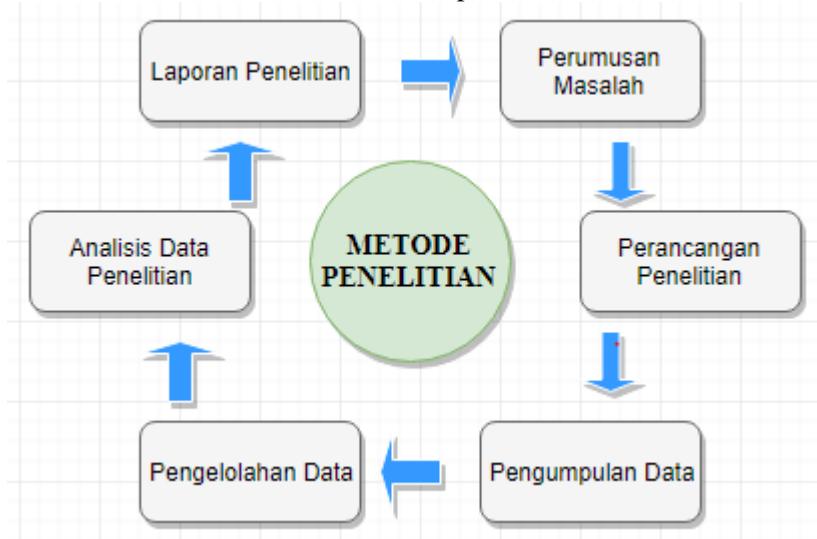
Maka akan ada notifikasi bahwa sudah berhasil absen hadir. Penjelasan (4) Sebelumnya Absensi Tri Dharma Asisten Lab masih sangat manual dan tidak ada rekapan yang jelas untuk sebuah penilaian. Dengan adanya sistem ini maka akan memudahkan Asisten Lab dan Penanggung Jawab dalam Absensi hadir dan ada rekapan yang jelas untuk Penilaian Akhir.

Di dalam sistem ini merancang sebuah sistem Pensil (Penilaian Asisten Lab) dengan menggunakan Yii adalah framework pemrograman umum Web yang bisa dipakai untuk mengembangkan semua jenis aplikasi Web. Dikarenakan sangat ringan dan dilengkapi dengan mekanisme caching yang canggih, Yii sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, forum, sistem manajemen konten (CMS), sistem *e-commerce*, dan lain-lain.

Yii melampaui framework PHP lain dalam hal efisiensi, kekayaan-fitur, dan kejelasan dokumentasi. Yii di desain dengan hati-hati dari awal agar sesuai untuk pengembangan aplikasi Web secara serius. Yii bukan berasal dari produk pada beberapa proyek maupun konglomerasi pekerjaan pihak-ketiga. [7] Yii adalah hasil dari pengalaman karya para pembuat pada pengembangan aplikasi Web dan investigasi framework pemrograman Web dan aplikasi yang paling populer.

2. METODE PENELITIAN

Berdasarkan Metode Penelitian yang ada saat ini akan membahas berbagai masalah yang ada, pada pembahasan saat ini yang ada di dalam Metode Penelitian yang dijalankan ada 6 Metode Penelitian yaitu, Perumusan Masalah, Perancangan Penelitian, Pengumpulan Data, Pengelolahan Data, Analisis Data Penelitian dan Laporan Penelitian.



Gambar 2. Metode Penelitian

Keterangan Metode Penelitian :

- a. Perumusan Masalah
Menyusun kerangka masalah dengan melakukan sebuah observasi sebelum melakukan sebuah penelitian.
- b. Perancangan Penelitian

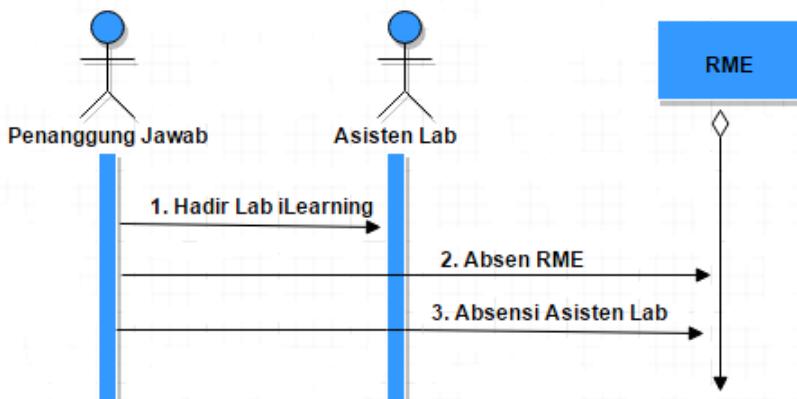
- Membuat sebuah Perancangan dengan membuat *Prototype* pada *blueprint* sehingga dapat teralisasi saat sudah dijalankan dengan sistem.
- c. Pengumpulan Data
Baiknya saat kita menjalankan sebuah sistem, hal pertama yang kita ketahui adalah menumpulkan sebuah data yang akan menjadi sebuah reverensi untuk mengola data berikutnya.
 - d. Pengelolahan Data
Saat sudah mengumpulkan data yang akurat maka tahap selanjutnya akan ada sebuah pengelolahan data yang lebih akurat dan lebih real.
 - e. Analisis Data Penelitian
Saat sudah mengumpulkan data dan mengelola data hal yang harus dicapai adalah menganalisa sebuah data tersebut menjadi data yang dapat diakses dengan baik dan akurat.
 - f. Laporan Penelitian
Laporan Penelitian hal yang mendekati sebuah uji coba atau implementasi yang akan dijalankan dalam sebuah sistem yang sudah siap dilaporkan sebagai pencapaian data yang akurat

3. HASIL DAN PEMBAHASAN

Dalam sebuah Hasil dan Pembahasan yang akan dijelaskan adalah sebuah penelitian sistem yang memudahkan bagi penggunaanya dalam melakukan sebuah akses dalam Absensi dengan hal yang mudah dan menarik dengan hanya *Scan* *Qr Code*, dalam sebuah pembahasan yang akan dibahas berikut ini adalah :

3.1 Analisa Permasalahan

Pada perguruan tinggi yang sudah mempunyai sebuah sistem pembelajaran *iLearning*, sepatutnya segala sesuatu sistem harus ikut dalam sistem yang ikut maju dan berkembang pesat. yang di mana harus terdapat sistem yang bisa dinilai baik bagi perguruan tinggi tersebut. Jika sistem pembelajaran sudah bisa jauh lewati tahap sudah *Online* maka sistem yang masih manual harus dirubah menjadi sistem yang mudah untuk perkembangan perguruan tinggi tersebut. permasalahan yang pertama sistem penilaian untuk pengabdian tri dharma masih sangat manual dan yang kedua masih ada yang kurang bertanggung jawab dalam menjalankan sebuah pengabdian tersebut.



Gambar 3. *Sequence Diagram*

Berikut ini adalah sebuah penjelasan gambar diatas yang di mana terdapat Penanggung Jawab dan Asisten Lab. diharuskan kepada Asisten Lab untuk menjalankan sebuah Tri Dharma yang di mana penilaian tersebut diawali dengan sebuah kinerja yang baik. Penilaian yang wajib diikuti oleh Asisten Lab dengan disiplin dalam menjalankan Tri Dharma seperti penilaian Absensi yang di mana itu adalah sebuah penilaian yang di mana permasalahan yang ada saat

Asisten Lab hadir maka harus membuka sebuah RME (Raharja Multimedia Edutainment) yang terdapat di Perguruan Tinggi lalu bisa Absensi. Tetapi itu kurang efektif dan efisien di mana masih ada terjadinya sebuah kecurangan penitipan absen pada sesama Asisten Lab jika hanya dengan memberi sebuah NIM lalu masuk kedalam web RME (Raharja Multimedia Edutainment) tersebut.

3.2 Pemecahan Masalah

Untuk mengatasi berbagai macam masalah tersebut, maka diperlukan sistem yang efektif dan efisien dalam melakukan penilaian absensi secara online. Dari rumusan masalah tersebut menghasilkan satu pemecahan masalah yaitu dengan menggunakan sistem yang dapat di akses dengan mudah dan cepat dengan hanya login pada sebuah website yang tersedia maka para kinerja pengabdian tri dharma tersebut bisa terneutralisir dengan baik.



Gambar 4. Berhasil Scan Qr Code pada Sistem

Dalam pembuatan perancangan sistem tersebut maka bisa dilihat dan disimpulkan bahwa ada sistem yang bisa dengan mudahnya untuk diakses karena dengan hanya dengan membuka sistem yang tersedia login dengan menggunakan SSO akan lebih cepat dan efektif untuk dijalankan. Dalam sistem ini ada sebuah penanggung jawab untuk membuka sistem yang ada guna memperlancar dan mencegah terjadinya kecurangan dalam membuka ruangan tersebut.

3.3 Listing Program

```
/*
 * public function actionCreate()
 * {
 *     $model = new Users();
 *
 *     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 *         return $this->redirect(['manage']);
 *     } else {
 *         return $this->render('create', [
 *             'model' => $model,
 *         ]);
 *     }
 *
 *     /**
 *     * Updates an existing Users model.
 *     * If update is successful, the browser will be redirected to the 'view' page.
 *     * @param string $id
 *     * @return mixed
 *     */
 *     public function actionUpdate($id)
 *     {
 *         $model = $this->findModel($id);
 *
 *         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 *             return $this->redirect(['manage']);
 *         } else {
 *             return $this->render('update', [
 *                 'model' => $model,
 *             ]);
 *         }
 *     }
 * }
```

Gambar 5. Konfigurasi pada Qr Code dan Scan untuk Absensi

Pada Script tersebut proses untuk memasukkan Script QR Code Supports QR Code *versions Exports to PNG, JPEG images, also exports as bit-table* agar menghasilkan sebuah QR Code

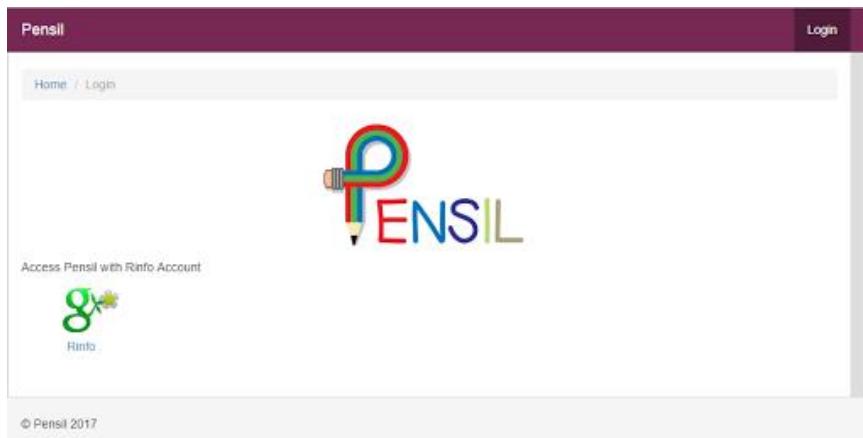


```

 31     /**
 32      * Updates an existing OpenRoom model.
 33      * If update is successful, the browser will be redirected to the 'view' page.
 34      * @param integer $id
 35      * @return mixed
 36      */
 37     public function actionUpdate($id)
 38     {
 39         $model = $this->findModel($id);
 40
 41         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 42             return $this->redirect(['view', 'id' => $model->openId]);
 43         } else {
 44             return $this->render('create', [
 45                 'model' => $model,
 46             ]);
 47         }
 48     }
 49
 50     /**
 51      * Deletes an existing OpenRoom model.
 52      * If delete is successful, the browser will be redirected to the 'index' page.
 53      * @param integer $id
 54      * @return mixed
 55      */
 56     public function actionDelete($id)
 57     {
 58         $model = $this->findModel($id);
 59
 60         if ($model->delete()) {
 61             return $this->redirect(['index']);
 62         } else {
 63             return $this->render('index', [
 64                 'searchModel' => $searchModel,
 65                 'dataProvider' => $dataProvider,
 66             ]);
 67         }
 68     }
 69
 70     /**
 71      * Displays a single Absensi model.
 72      * @param integer $id
 73      * @return mixed
 74      */
 75     public function actionView($id)
 76     {
 77         return $this->render('view', [
 78             'model' => $this->findModel($id),
 79         ]);
 80     }
 81 }
 82
 83 /**
 84  * Lists all Absensi models.
 85  * @return mixed
 86  */
 87 public function actionIndex()
 88 {
 89     $searchModel = new AbsensiSearch();
 90     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 91
 92     return $this->render('index', [
 93         'searchModel' => $searchModel,
 94         'dataProvider' => $dataProvider,
 95     ]);
 96 }
 97
 98 /**
 99  * Displays a single Absensi model.
100  * @param integer $id
101  * @return mixed
102  */
103 public function actionCreate()
104 {
105     $model = new OpenRoom();
106
107     if ($model->load(Yii::$app->request->post()) && $model->save()) {
108         return $this->redirect(['view', 'id' => $model->openId]);
109     } else {
110         return $this->render('create', [
111             'model' => $model,
112         ]);
113     }
114 }
 115
 116 /**
 117  * Updates an existing OpenRoom model.
 118  * If update is successful, the browser will be redirected to the 'view' page.
 119  * @param integer $id
 120  * @return mixed
 121  */
 122     public function actionUpdate($id)
 123     {
 124         $model = $this->findModel($id);
 125
 126         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 127             return $this->redirect(['view', 'id' => $model->openId]);
 128         } else {
 129             return $this->render('create', [
 130                 'model' => $model,
 131             ]);
 132         }
 133     }
 134
 135     /**
 136      * Deletes an existing OpenRoom model.
 137      * If delete is successful, the browser will be redirected to the 'index' page.
 138      * @param integer $id
 139      * @return mixed
 140      */
 141     public function actionDelete($id)
 142     {
 143         $model = $this->findModel($id);
 144
 145         if ($model->delete()) {
 146             return $this->redirect(['index']);
 147         } else {
 148             return $this->render('index', [
 149                 'searchModel' => $searchModel,
 150                 'dataProvider' => $dataProvider,
 151             ]);
 152         }
 153     }
 154
 155     /**
 156      * Displays a single Absensi model.
 157      * @param integer $id
 158      * @return mixed
 159      */
 160     public function actionView($id)
 161     {
 162         return $this->render('view', [
 163             'model' => $this->findModel($id),
 164         ]);
 165     }
 166 }
 167
 168 /**
 169  * Lists all Absensi models.
 170  * @return mixed
 171  */
 172 public function actionIndex()
 173 {
 174     $searchModel = new AbsensiSearch();
 175     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 176
 177     return $this->render('index', [
 178         'searchModel' => $searchModel,
 179         'dataProvider' => $dataProvider,
 180     ]);
 181 }
 182
 183 /**
 184  * Displays a single Absensi model.
 185  * @param integer $id
 186  * @return mixed
 187  */
 188 public function actionCreate()
 189 {
 190     $model = new Absensi();
 191
 192     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 193         return $this->redirect(['view', 'id' => $model->id]);
 194     } else {
 195         return $this->render('create', [
 196             'model' => $model,
 197         ]);
 198     }
 199 }
 200
 201 /**
 202  * Updates an existing Absensi model.
 203  * If update is successful, the browser will be redirected to the 'view' page.
 204  * @param integer $id
 205  * @return mixed
 206  */
 207     public function actionUpdate($id)
 208     {
 209         $model = $this->findModel($id);
 210
 211         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 212             return $this->redirect(['view', 'id' => $model->id]);
 213         } else {
 214             return $this->render('create', [
 215                 'model' => $model,
 216             ]);
 217         }
 218     }
 219
 220     /**
 221      * Deletes an existing Absensi model.
 222      * If delete is successful, the browser will be redirected to the 'index' page.
 223      * @param integer $id
 224      * @return mixed
 225      */
 226     public function actionDelete($id)
 227     {
 228         $model = $this->findModel($id);
 229
 230         if ($model->delete()) {
 231             return $this->redirect(['index']);
 232         } else {
 233             return $this->render('index', [
 234                 'searchModel' => $searchModel,
 235                 'dataProvider' => $dataProvider,
 236             ]);
 237         }
 238     }
 239
 240     /**
 241      * Displays a single Absensi model.
 242      * @param integer $id
 243      * @return mixed
 244      */
 245     public function actionView($id)
 246     {
 247         return $this->render('view', [
 248             'model' => $this->findModel($id),
 249         ]);
 250     }
 251 }
 252
 253 /**
 254  * Lists all Absensi models.
 255  * @return mixed
 256  */
 257 public function actionIndex()
 258 {
 259     $searchModel = new AbsensiSearch();
 260     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 261
 262     return $this->render('index', [
 263         'searchModel' => $searchModel,
 264         'dataProvider' => $dataProvider,
 265     ]);
 266 }
 267
 268 /**
 269  * Displays a single Absensi model.
 270  * @param integer $id
 271  * @return mixed
 272  */
 273 public function actionCreate()
 274 {
 275     $model = new Absensi();
 276
 277     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 278         return $this->redirect(['view', 'id' => $model->id]);
 279     } else {
 280         return $this->render('create', [
 281             'model' => $model,
 282         ]);
 283     }
 284 }
 285
 286 /**
 287  * Updates an existing Absensi model.
 288  * If update is successful, the browser will be redirected to the 'view' page.
 289  * @param integer $id
 290  * @return mixed
 291  */
 292     public function actionUpdate($id)
 293     {
 294         $model = $this->findModel($id);
 295
 296         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 297             return $this->redirect(['view', 'id' => $model->id]);
 298         } else {
 299             return $this->render('create', [
 300                 'model' => $model,
 301             ]);
 302         }
 303     }
 304
 305     /**
 306      * Deletes an existing Absensi model.
 307      * If delete is successful, the browser will be redirected to the 'index' page.
 308      * @param integer $id
 309      * @return mixed
 310      */
 311     public function actionDelete($id)
 312     {
 313         $model = $this->findModel($id);
 314
 315         if ($model->delete()) {
 316             return $this->redirect(['index']);
 317         } else {
 318             return $this->render('index', [
 319                 'searchModel' => $searchModel,
 320                 'dataProvider' => $dataProvider,
 321             ]);
 322         }
 323     }
 324
 325     /**
 326      * Displays a single Absensi model.
 327      * @param integer $id
 328      * @return mixed
 329      */
 330     public function actionView($id)
 331     {
 332         return $this->render('view', [
 333             'model' => $this->findModel($id),
 334         ]);
 335     }
 336 }
 337
 338 /**
 339  * Lists all Absensi models.
 340  * @return mixed
 341  */
 342 public function actionIndex()
 343 {
 344     $searchModel = new AbsensiSearch();
 345     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 346
 347     return $this->render('index', [
 348         'searchModel' => $searchModel,
 349         'dataProvider' => $dataProvider,
 350     ]);
 351 }
 352
 353 /**
 354  * Displays a single Absensi model.
 355  * @param integer $id
 356  * @return mixed
 357  */
 358 public function actionCreate()
 359 {
 360     $model = new Absensi();
 361
 362     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 363         return $this->redirect(['view', 'id' => $model->id]);
 364     } else {
 365         return $this->render('create', [
 366             'model' => $model,
 367         ]);
 368     }
 369 }
 370
 371 /**
 372  * Updates an existing Absensi model.
 373  * If update is successful, the browser will be redirected to the 'view' page.
 374  * @param integer $id
 375  * @return mixed
 376  */
 377     public function actionUpdate($id)
 378     {
 379         $model = $this->findModel($id);
 380
 381         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 382             return $this->redirect(['view', 'id' => $model->id]);
 383         } else {
 384             return $this->render('create', [
 385                 'model' => $model,
 386             ]);
 387         }
 388     }
 389
 390     /**
 391      * Deletes an existing Absensi model.
 392      * If delete is successful, the browser will be redirected to the 'index' page.
 393      * @param integer $id
 394      * @return mixed
 395      */
 396     public function actionDelete($id)
 397     {
 398         $model = $this->findModel($id);
 399
 400         if ($model->delete()) {
 401             return $this->redirect(['index']);
 402         } else {
 403             return $this->render('index', [
 404                 'searchModel' => $searchModel,
 405                 'dataProvider' => $dataProvider,
 406             ]);
 407         }
 408     }
 409
 410     /**
 411      * Displays a single Absensi model.
 412      * @param integer $id
 413      * @return mixed
 414      */
 415     public function actionView($id)
 416     {
 417         return $this->render('view', [
 418             'model' => $this->findModel($id),
 419         ]);
 420     }
 421 }
 422
 423 /**
 424  * Lists all Absensi models.
 425  * @return mixed
 426  */
 427 public function actionIndex()
 428 {
 429     $searchModel = new AbsensiSearch();
 430     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 431
 432     return $this->render('index', [
 433         'searchModel' => $searchModel,
 434         'dataProvider' => $dataProvider,
 435     ]);
 436 }
 437
 438 /**
 439  * Displays a single Absensi model.
 440  * @param integer $id
 441  * @return mixed
 442  */
 443 public function actionCreate()
 444 {
 445     $model = new Absensi();
 446
 447     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 448         return $this->redirect(['view', 'id' => $model->id]);
 449     } else {
 450         return $this->render('create', [
 451             'model' => $model,
 452         ]);
 453     }
 454 }
 455
 456 /**
 457  * Updates an existing Absensi model.
 458  * If update is successful, the browser will be redirected to the 'view' page.
 459  * @param integer $id
 460  * @return mixed
 461  */
 462     public function actionUpdate($id)
 463     {
 464         $model = $this->findModel($id);
 465
 466         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 467             return $this->redirect(['view', 'id' => $model->id]);
 468         } else {
 469             return $this->render('create', [
 470                 'model' => $model,
 471             ]);
 472         }
 473     }
 474
 475     /**
 476      * Deletes an existing Absensi model.
 477      * If delete is successful, the browser will be redirected to the 'index' page.
 478      * @param integer $id
 479      * @return mixed
 480      */
 481     public function actionDelete($id)
 482     {
 483         $model = $this->findModel($id);
 484
 485         if ($model->delete()) {
 486             return $this->redirect(['index']);
 487         } else {
 488             return $this->render('index', [
 489                 'searchModel' => $searchModel,
 490                 'dataProvider' => $dataProvider,
 491             ]);
 492         }
 493     }
 494
 495     /**
 496      * Displays a single Absensi model.
 497      * @param integer $id
 498      * @return mixed
 499      */
 500     public function actionView($id)
 501     {
 502         return $this->render('view', [
 503             'model' => $this->findModel($id),
 504         ]);
 505     }
 506 }
 507
 508 /**
 509  * Lists all Absensi models.
 510  * @return mixed
 511  */
 512 public function actionIndex()
 513 {
 514     $searchModel = new AbsensiSearch();
 515     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 516
 517     return $this->render('index', [
 518         'searchModel' => $searchModel,
 519         'dataProvider' => $dataProvider,
 520     ]);
 521 }
 522
 523 /**
 524  * Displays a single Absensi model.
 525  * @param integer $id
 526  * @return mixed
 527  */
 528 public function actionCreate()
 529 {
 530     $model = new Absensi();
 531
 532     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 533         return $this->redirect(['view', 'id' => $model->id]);
 534     } else {
 535         return $this->render('create', [
 536             'model' => $model,
 537         ]);
 538     }
 539 }
 540
 541 /**
 542  * Updates an existing Absensi model.
 543  * If update is successful, the browser will be redirected to the 'view' page.
 544  * @param integer $id
 545  * @return mixed
 546  */
 547     public function actionUpdate($id)
 548     {
 549         $model = $this->findModel($id);
 550
 551         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 552             return $this->redirect(['view', 'id' => $model->id]);
 553         } else {
 554             return $this->render('create', [
 555                 'model' => $model,
 556             ]);
 557         }
 558     }
 559
 560     /**
 561      * Deletes an existing Absensi model.
 562      * If delete is successful, the browser will be redirected to the 'index' page.
 563      * @param integer $id
 564      * @return mixed
 565      */
 566     public function actionDelete($id)
 567     {
 568         $model = $this->findModel($id);
 569
 570         if ($model->delete()) {
 571             return $this->redirect(['index']);
 572         } else {
 573             return $this->render('index', [
 574                 'searchModel' => $searchModel,
 575                 'dataProvider' => $dataProvider,
 576             ]);
 577         }
 578     }
 579
 580     /**
 581      * Displays a single Absensi model.
 582      * @param integer $id
 583      * @return mixed
 584      */
 585     public function actionView($id)
 586     {
 587         return $this->render('view', [
 588             'model' => $this->findModel($id),
 589         ]);
 590     }
 591 }
 592
 593 /**
 594  * Lists all Absensi models.
 595  * @return mixed
 596  */
 597 public function actionIndex()
 598 {
 599     $searchModel = new AbsensiSearch();
 600     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 601
 602     return $this->render('index', [
 603         'searchModel' => $searchModel,
 604         'dataProvider' => $dataProvider,
 605     ]);
 606 }
 607
 608 /**
 609  * Displays a single Absensi model.
 610  * @param integer $id
 611  * @return mixed
 612  */
 613 public function actionCreate()
 614 {
 615     $model = new Absensi();
 616
 617     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 618         return $this->redirect(['view', 'id' => $model->id]);
 619     } else {
 620         return $this->render('create', [
 621             'model' => $model,
 622         ]);
 623     }
 624 }
 625
 626 /**
 627  * Updates an existing Absensi model.
 628  * If update is successful, the browser will be redirected to the 'view' page.
 629  * @param integer $id
 630  * @return mixed
 631  */
 632     public function actionUpdate($id)
 633     {
 634         $model = $this->findModel($id);
 635
 636         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 637             return $this->redirect(['view', 'id' => $model->id]);
 638         } else {
 639             return $this->render('create', [
 640                 'model' => $model,
 641             ]);
 642         }
 643     }
 644
 645     /**
 646      * Deletes an existing Absensi model.
 647      * If delete is successful, the browser will be redirected to the 'index' page.
 648      * @param integer $id
 649      * @return mixed
 650      */
 651     public function actionDelete($id)
 652     {
 653         $model = $this->findModel($id);
 654
 655         if ($model->delete()) {
 656             return $this->redirect(['index']);
 657         } else {
 658             return $this->render('index', [
 659                 'searchModel' => $searchModel,
 660                 'dataProvider' => $dataProvider,
 661             ]);
 662         }
 663     }
 664
 665     /**
 666      * Displays a single Absensi model.
 667      * @param integer $id
 668      * @return mixed
 669      */
 670     public function actionView($id)
 671     {
 672         return $this->render('view', [
 673             'model' => $this->findModel($id),
 674         ]);
 675     }
 676 }
 677
 678 /**
 679  * Lists all Absensi models.
 680  * @return mixed
 681  */
 682 public function actionIndex()
 683 {
 684     $searchModel = new AbsensiSearch();
 685     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 686
 687     return $this->render('index', [
 688         'searchModel' => $searchModel,
 689         'dataProvider' => $dataProvider,
 690     ]);
 691 }
 692
 693 /**
 694  * Displays a single Absensi model.
 695  * @param integer $id
 696  * @return mixed
 697  */
 698 public function actionCreate()
 699 {
 700     $model = new Absensi();
 701
 702     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 703         return $this->redirect(['view', 'id' => $model->id]);
 704     } else {
 705         return $this->render('create', [
 706             'model' => $model,
 707         ]);
 708     }
 709 }
 710
 711 /**
 712  * Updates an existing Absensi model.
 713  * If update is successful, the browser will be redirected to the 'view' page.
 714  * @param integer $id
 715  * @return mixed
 716  */
 717     public function actionUpdate($id)
 718     {
 719         $model = $this->findModel($id);
 720
 721         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 722             return $this->redirect(['view', 'id' => $model->id]);
 723         } else {
 724             return $this->render('create', [
 725                 'model' => $model,
 726             ]);
 727         }
 728     }
 729
 730     /**
 731      * Deletes an existing Absensi model.
 732      * If delete is successful, the browser will be redirected to the 'index' page.
 733      * @param integer $id
 734      * @return mixed
 735      */
 736     public function actionDelete($id)
 737     {
 738         $model = $this->findModel($id);
 739
 740         if ($model->delete()) {
 741             return $this->redirect(['index']);
 742         } else {
 743             return $this->render('index', [
 744                 'searchModel' => $searchModel,
 745                 'dataProvider' => $dataProvider,
 746             ]);
 747         }
 748     }
 749
 750     /**
 751      * Displays a single Absensi model.
 752      * @param integer $id
 753      * @return mixed
 754      */
 755     public function actionView($id)
 756     {
 757         return $this->render('view', [
 758             'model' => $this->findModel($id),
 759         ]);
 760     }
 761 }
 762
 763 /**
 764  * Lists all Absensi models.
 765  * @return mixed
 766  */
 767 public function actionIndex()
 768 {
 769     $searchModel = new AbsensiSearch();
 770     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 771
 772     return $this->render('index', [
 773         'searchModel' => $searchModel,
 774         'dataProvider' => $dataProvider,
 775     ]);
 776 }
 777
 778 /**
 779  * Displays a single Absensi model.
 780  * @param integer $id
 781  * @return mixed
 782  */
 783 public function actionCreate()
 784 {
 785     $model = new Absensi();
 786
 787     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 788         return $this->redirect(['view', 'id' => $model->id]);
 789     } else {
 790         return $this->render('create', [
 791             'model' => $model,
 792         ]);
 793     }
 794 }
 795
 796 /**
 797  * Updates an existing Absensi model.
 798  * If update is successful, the browser will be redirected to the 'view' page.
 799  * @param integer $id
 800  * @return mixed
 801  */
 802     public function actionUpdate($id)
 803     {
 804         $model = $this->findModel($id);
 805
 806         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 807             return $this->redirect(['view', 'id' => $model->id]);
 808         } else {
 809             return $this->render('create', [
 810                 'model' => $model,
 811             ]);
 812         }
 813     }
 814
 815     /**
 816      * Deletes an existing Absensi model.
 817      * If delete is successful, the browser will be redirected to the 'index' page.
 818      * @param integer $id
 819      * @return mixed
 820      */
 821     public function actionDelete($id)
 822     {
 823         $model = $this->findModel($id);
 824
 825         if ($model->delete()) {
 826             return $this->redirect(['index']);
 827         } else {
 828             return $this->render('index', [
 829                 'searchModel' => $searchModel,
 830                 'dataProvider' => $dataProvider,
 831             ]);
 832         }
 833     }
 834
 835     /**
 836      * Displays a single Absensi model.
 837      * @param integer $id
 838      * @return mixed
 839      */
 840     public function actionView($id)
 841     {
 842         return $this->render('view', [
 843             'model' => $this->findModel($id),
 844         ]);
 845     }
 846 }
 847
 848 /**
 849  * Lists all Absensi models.
 850  * @return mixed
 851  */
 852 public function actionIndex()
 853 {
 854     $searchModel = new AbsensiSearch();
 855     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 856
 857     return $this->render('index', [
 858         'searchModel' => $searchModel,
 859         'dataProvider' => $dataProvider,
 860     ]);
 861 }
 862
 863 /**
 864  * Displays a single Absensi model.
 865  * @param integer $id
 866  * @return mixed
 867  */
 868 public function actionCreate()
 869 {
 870     $model = new Absensi();
 871
 872     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 873         return $this->redirect(['view', 'id' => $model->id]);
 874     } else {
 875         return $this->render('create', [
 876             'model' => $model,
 877         ]);
 878     }
 879 }
 880
 881 /**
 882  * Updates an existing Absensi model.
 883  * If update is successful, the browser will be redirected to the 'view' page.
 884  * @param integer $id
 885  * @return mixed
 886  */
 887     public function actionUpdate($id)
 888     {
 889         $model = $this->findModel($id);
 890
 891         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 892             return $this->redirect(['view', 'id' => $model->id]);
 893         } else {
 894             return $this->render('create', [
 895                 'model' => $model,
 896             ]);
 897         }
 898     }
 899
 900     /**
 901      * Deletes an existing Absensi model.
 902      * If delete is successful, the browser will be redirected to the 'index' page.
 903      * @param integer $id
 904      * @return mixed
 905      */
 906     public function actionDelete($id)
 907     {
 908         $model = $this->findModel($id);
 909
 910         if ($model->delete()) {
 911             return $this->redirect(['index']);
 912         } else {
 913             return $this->render('index', [
 914                 'searchModel' => $searchModel,
 915                 'dataProvider' => $dataProvider,
 916             ]);
 917         }
 918     }
 919
 920     /**
 921      * Displays a single Absensi model.
 922      * @param integer $id
 923      * @return mixed
 924      */
 925     public function actionView($id)
 926     {
 927         return $this->render('view', [
 928             'model' => $this->findModel($id),
 929         ]);
 930     }
 931 }
 932
 933 /**
 934  * Lists all Absensi models.
 935  * @return mixed
 936  */
 937 public function actionIndex()
 938 {
 939     $searchModel = new AbsensiSearch();
 940     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 941
 942     return $this->render('index', [
 943         'searchModel' => $searchModel,
 944         'dataProvider' => $dataProvider,
 945     ]);
 946 }
 947
 948 /**
 949  * Displays a single Absensi model.
 950  * @param integer $id
 951  * @return mixed
 952  */
 953 public function actionCreate()
 954 {
 955     $model = new Absensi();
 956
 957     if ($model->load(Yii::$app->request->post()) && $model->save()) {
 958         return $this->redirect(['view', 'id' => $model->id]);
 959     } else {
 960         return $this->render('create', [
 961             'model' => $model,
 962         ]);
 963     }
 964 }
 965
 966 /**
 967  * Updates an existing Absensi model.
 968  * If update is successful, the browser will be redirected to the 'view' page.
 969  * @param integer $id
 970  * @return mixed
 971  */
 972     public function actionUpdate($id)
 973     {
 974         $model = $this->findModel($id);
 975
 976         if ($model->load(Yii::$app->request->post()) && $model->save()) {
 977             return $this->redirect(['view', 'id' => $model->id]);
 978         } else {
 979             return $this->render('create', [
 980                 'model' => $model,
 981             ]);
 982         }
 983     }
 984
 985     /**
 986      * Deletes an existing Absensi model.
 987      * If delete is successful, the browser will be redirected to the 'index' page.
 988      * @param integer $id
 989      * @return mixed
 990      */
 991     public function actionDelete($id)
 992     {
 993         $model = $this->findModel($id);
 994
 995         if ($model->delete()) {
 996             return $this->redirect(['index']);
 997         } else {
 998             return $this->render('index', [
 999                 'searchModel' => $searchModel,
 1000                'dataProvider' => $dataProvider,
 1001            ]);
 1002        }
 1003    }
 1004
 1005    /**
 1006      * Displays a single Absensi model.
 1007      * @param integer $id
 1008      * @return mixed
 1009      */
 1010     public function actionView($id)
 1011     {
 1012         return $this->render('view', [
 1013             'model' => $this->findModel($id),
 1014         ]);
 1015     }
 1016 }
 1017
 1018 /**
 1019  * Lists all Absensi models.
 1020  * @return mixed
 1021  */
 1022 public function actionIndex()
 1023 {
 1024     $searchModel = new AbsensiSearch();
 1025     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
 1026
 1027     return $this->render('index', [
 1028         'searchModel' => $searchModel,
 1029         'dataProvider' => $dataProvider,
 1030     ]);
 1031 }
 1032
 1033 /**
 1034  * Displays a single Absensi model.
 1035  * @param integer $id
 1036  * @return mixed
 1037  */
 1038 public function actionCreate()
 1039 {
 1040     $model = new Absensi();
 1041
 
```

Absensi yang lebih akurat. dengan mengakses sebuah data penilaianya pun akan lebih cepat dan akurat dalam penilaian akhir.

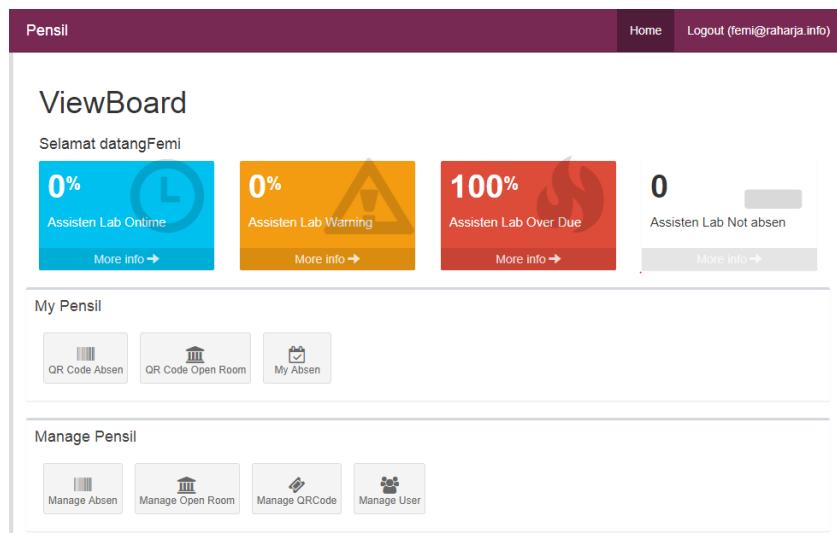
a. Halaman utama



Gambar 8. Halaman utama pada website

Pada gambar diatas adalah halaman utama sebagai menu utama pada website Pensil (Penilaian Asisten Lab) yaitu pensil.raharja.ac.id

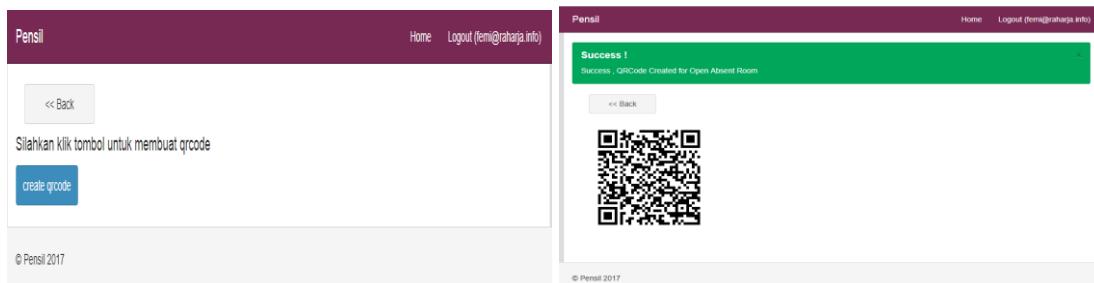
b. Tampilan website



Gambar 9. Halaman untuk absensi

Halaman berikut ini untuk Penanggung Jawab dan Asisten Lab buka absensi ruangan dan absensi untuk asisten lab. Yang di mana halaman tersebut jika Penanggung Jawab ingin membuka absensi ruangan maka hanya dengan klik button QR Code Open Room saja. Lalu dengan Asisten Lab hanya dengan klik button QR Code Absen.

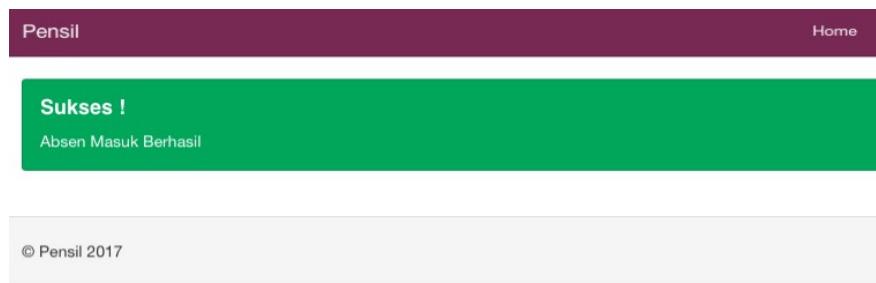
c. Tampilan menu Qr Code



Gambar 10. Tampilan menu Qr Code untuk *scanning*

Pada gambar diatas untuk Penanggung Jawab dan Asisten Lab agar mendapatkan QR Code dengan tahapan pertama klik *button create Qr Code* maka akan muncul gambar QR Code tersebut untuk dapat di *scanning*.

d. Tampilan Notifikasi berhasil masuk absensi



Gambar 11. Berhasil masuk absensi

Gambar tersebut adalah notifikasi bahwa sudah berhasil absensi pada website pensil.raharja.ac.id

Sangat mempermudah bagi Asisten Lab (Mahasiswa) dalam menjalankan sebuah Absensi dengan hanya akses pensil.raharja.ac.id. dan memperlancar Absensi yang benar benar sangat mudah di akses dan lebih akurat.

4. KESIMPULAN

Berdasarkan Penelitian diatas bisa disimpulkan bahwa sebuah Penilaian Asisten Lab didalam Tri Dharma harus dibuat dengan sangat mudah dan lebih Objektif. karena setiap performa mahasiswa sangatlah berbeda yang di mana setiap mahasiswa ada yang hadir tepat waktu, terlambat atau izin yang di mana sebuah Penilaian ini harus sangat objektif. Dengan adanya sistem ini sangat dibutuhkan sekali, guna memperlancar dan memudahkan lebih efesien juga bagi dosen yang menilai sebuah penilaian Tri Dharma yang harusnya diberi penilaian yang harus Objektif. Dengan melihat *Dashboard* yang tersedia untuk melihat sebuah penilaian pada Asisten Lab sangat mempermudah untuk diakses demi kelancaran sebuah penilaian. Dan Asisten Lab pun menjadi sangat lebih mudah dalam Absensi dengan hanya *Scan Qr Code* maka akan mempermudah dalam Absensi dan juga mengantisipasi terjadinya sebuah kecurangan yang di mana akan ada menitipkan absensi kepada teman dan Absensi ini hanya bisa diakses dan di *scan* hanya didalam Lab iLearning perguruan tinggi raharja saja. menyimpulkan bahwa dalam

penelitian ini memudahkan Asisten Lab (Mahasiswa) dalam melakukan sebuah Absensi yang lebih cepat dan akurat dengan hanya *Scan* *Qr Code*. dan Memudahkan Penanggung Jawab Lab (Dosen) juga lebih akurat dalam melihat data hadir yang lebih akurat dan *real*.

5. SARAN

Dari hasil penelitian ini masih terdapat kekurangan dan dari kekurangan itu memungkinkan akan dilakukan pengembangan lebih lanjut, saran yang perlu penulis sampaikan adalah untuk Asisten lab tidak absen manual kembali. Sistem yang dibuat akan memudahkan Penanggung jawab untuk menilai asisten lab tersebut. Dengan hanya mengakses sebuah website pensil.raharja.ac.id yang sudah tersusun dengan rapih dan memudahkan Asisten Lab dalam absensi secara online, mudah, dan cepat.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih sebesar besarnya atas arahan dan bimbingan kepada penulis. saat penulisan jurnal berlangsung. tidak adanya arahan kepada penulis, maka penulis tidak akan dapat menyelesaikan jurnal ini dengan baik.

DAFTAR PUSTAKA

- [1] Rahardja, U., Fadillah, I. and Lestari, W., 2017, August. Penggunaan System Single Sign On (SSO) Dalam Mendukung Kemajuan Pada Forum RHJFOX di Perguruan Tinggi Raharja. In SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi (Vol. 5, No. 1).
- [2] Rahardja, U., Sholeh, O. and Nursetianingsih, F., 2015. Penggunaan Dashboard untuk Mengontrol Kinerja Absensi Pegawai Guna Meningkatkan Profesionalisme Pegawai Pada PT. Sinarmas Land Property. SEMNASTEKNOMEDIA Online, 3(1).
- [3] Heroza, R.I. and Jannah, M., 2017. Pengembangan Sistem Absensi Menggunakan QR Code Reader Berbasis Android (Studi Kasus: Fakultas Ilmu Komputer Jurusan Sistem Informasi UNSRI). KNTIA, 4.
- [4] Setyawan, A.H., Satoto, K.I. and Isnanto, R.R., 2010. Perancangan Aplikasi Sistem Presensi Mahasiswa Menggunakan Qr Code Pada Sistem Operasi Android. Diponegoro: Universitas diponegoro.
- [5] Nurkhaerani, R., Wijayanto, H. and Syah, D., 2013. Eksplorasi Kinerja Dosen Tersertifikasi dalam Melaksanakan Tridharma Perguruan Tinggi di Indonesia. Xplore: Journal of Statistics, 1(2).
- [6] Aini, Q., Graha, Y.I. and Zuliana, S.R., 2017. Penerapan Absensi QRCode Mahasiswa Bimbingan Belajar pada Website berbasis YII Framework. SISFOTENIKA, 7(2), pp.207-218.
- [7] Framework, Y., 2014. About Yii| Yii PHP Framework.