

BAB III PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Dalam melaksanakan kegiatan kerja profesi di PT. Aseries Informix Solusi Teknologi (Nakama Id), praktikan bekerja pada tim *Information Technology* khususnya sebagai Back-end Developer. Tim ini memiliki berbagai tugas yang berperan penting untuk mendukung sistem operasional perusahaan, yaitu mengelola infrastruktur teknologi yang mencakup jaringan komputer, server, dan perangkat keras yang digunakan oleh seluruh tim Nakama Id.

Sebagai seorang Back-end Developer, praktikan ditugaskan untuk dapat mengembangkan, memelihara, dan mengelola bagian server serta database yang dimiliki oleh Nakama Id. *Back-end* developer merupakan sebuah bidang yang bekerja untuk mengembangkan dan membangun sebuah perangkat lunak pada operasi bagian belakang aplikasi. Biasanya bidang pekerjaan ini sangat berhubungan dengan logika dan operasi basis data (Prasetyo, Baehaki, Suharjono, & Ananda P, 2022). Selain itu, praktikan juga harus berkolaborasi dengan bagian Front-end untuk melakukan integrasi data yang sudah disiapkan sebelumnya ke dalam komponen tampilan aplikasi yang sudah dibuat. Selain itu, praktikan juga bertanggung jawab untuk membuat fitur Employee Hub yang berada pada proyek sistem absensi Nakama Id.

Fitur Employee Hub yang dibuat serta dikembangkan oleh praktikan, merupakan sebuah portal yang dikembangkan menggunakan *framework* Codeigniter 4 berbasis website. Dengan demikian, proyek tersebut sangat relevan dengan program studi yang diampu oleh praktikan karena sesuai dengan kurikulum yang sudah dipelajari sebelumnya selama praktikan menempuh masa studi di Universitas Pembangunan Jaya pada Program Studi Informatika. Praktikan juga memperoleh beberapa ilmu baru, yaitu pemahaman perancangan *fitur* dan keamanan sistem pada sebuah website (*Website Security*), pemecahan masalah saat mengembangkan

sebuah platform dan penguasaan beberapa library serta bahasa pemrograman yang belum pernah dipelajari sebelumnya.

3.2 Pelaksanaan Kerja

Pelaksanaan kerja dalam membangun platform berbasis *website* yang dilakukan oleh praktikan terdiri dari beberapa tahap agar *fitur* tersebut memiliki hasil yang baik dan kenyamanan dari sisi penggunanya. Tahapan utama, yaitu dimulai dengan analisis dan perancangan aplikasi meliputi, pembuatan struktur program, perancangan *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*. Setelah tahap tersebut selesai dilakukan, maka dilanjutkan dengan implementasi pembuatan operasi *Back-end* dan *Front-end* pada fitur *employee hub*.



Gambar 3. 1 Tahapan Perancangan Fitur

Gambar 3.1 merupakan langkah pelaksanaan kerja sistematis yang dilakukan oleh praktikan untuk membuat proyek dalam melaksanakan kegiatan kerja profesi di Nakama Id. Selama membuat proyek tersebut, praktikan juga diawasi dan diarahkan secara langsung oleh pembimbing eksternal agar perancangan proyek dapat terealisasi dengan maksimal. Timeline pelaksanaan proyek diatas dapat dilihat secara detail pada **Tabel 3.1**.

Tabel 3. 1 Linimasa Pelaksanaan Kerja

Aktivitas	Maret				April				Mei			
	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
Analisis kebutuhan Fitur	■	■										
Pembuatan Mockup			■									
Perancangan <i>Back-end</i>				■	■							
Perancangan <i>Front-end</i>					■	■						
Implementasi <i>Back-end</i>					■	■	■					
Implementasi dan Integrasi <i>Front-end</i>							■	■	■			
Refactoring Program											■	■

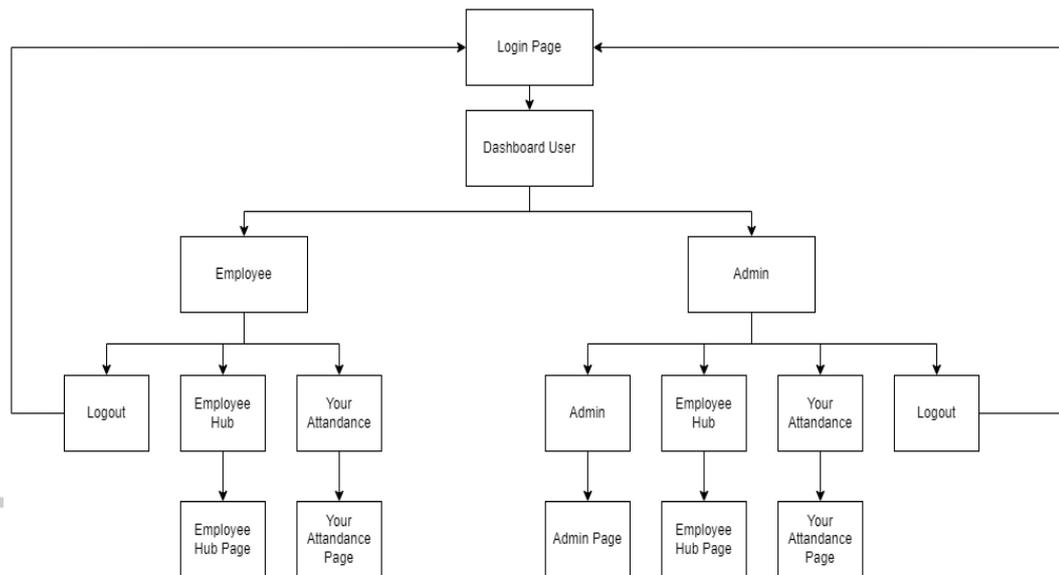
Berdasarkan tabel pelaksanaan proyek diatas, sampai saat ini praktikan baru mengerjakan proyek tersebut dalam kurun waktu 3 bulan dan akan terus berlanjut sesuai dengan tahap pengembangan fitur yang sudah ditentukan sebelumnya. Fitur tersebut pun berkemungkinan akan selalui diperbarui dan dikembangkan dengan fitur lainnya. Seluruh tahap yang dikerjakan doleh praktikan akan selalu di monitoring dan dibimbing oleh pembimbing eksternal secara langsung guna mendapatkan hasil yang maksimal.

3.2.1 Tahap Perancangan Fitur

Pada tahap perancangan, praktikan diberikan arahan untuk membuat dan melakukan pengembangan fitur *employee hub* pada sistem absensi Nakama Id. Fitur tersebut dibuat khusus untuk seluruh karyawan agar dapat melihat keseluruhan informasi mengenai *check in* dan *check out* beserta *list to do* yang mereka kerjakan. Pembimbing eksternal memberikan alur *fitur* dan kemudian praktikan ubah menjadi sebuah alur struktur program. Struktur tersebut mencakup

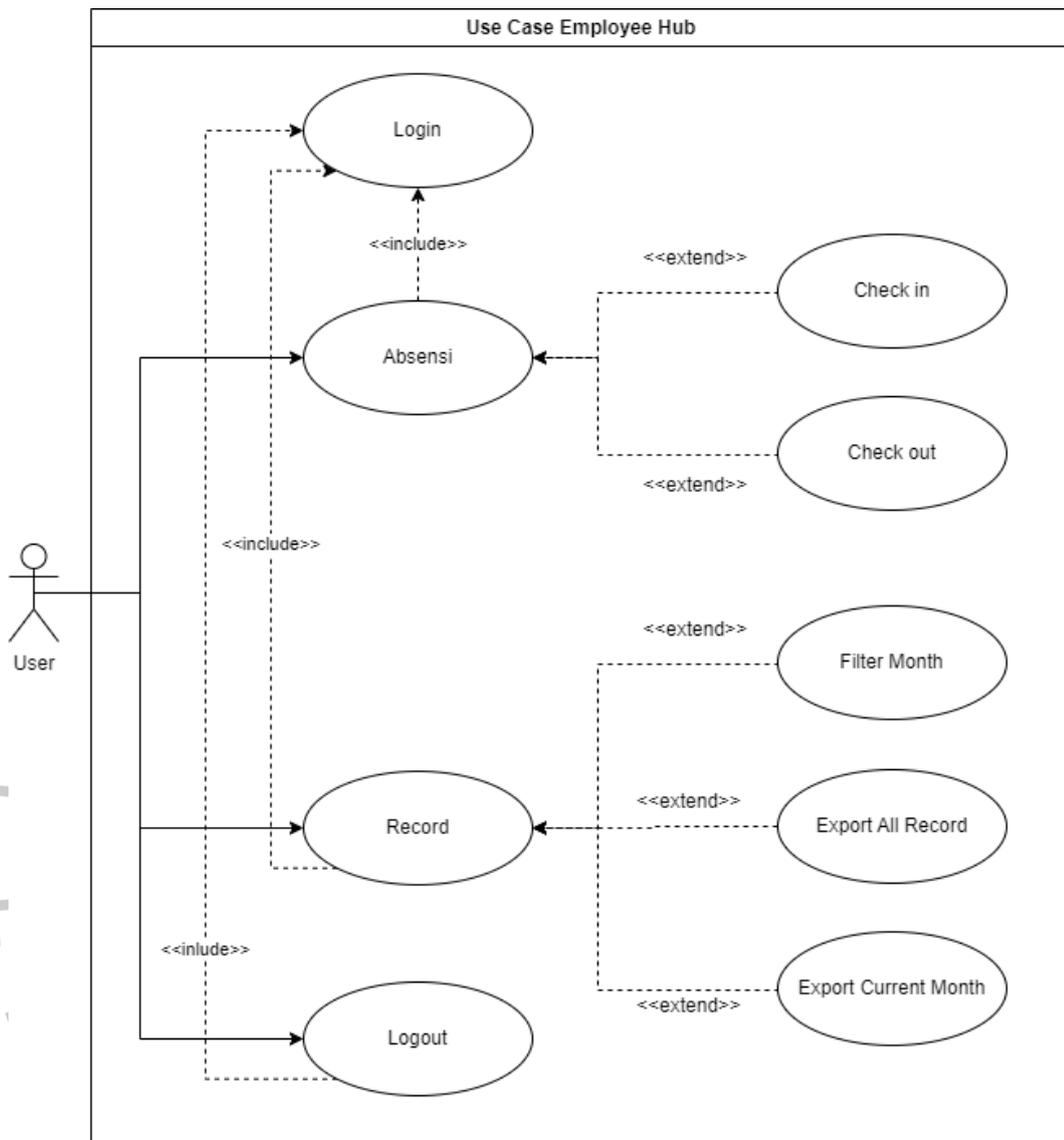
seluruh informasi mengenai jalannya fitur *employee hub* agar dapat dipahami oleh seluruh *stakeholder* yang menggunakan fitur tersebut.

Struktur Program Fitur Employee Hub



Gambar 3. 2 Struktur Program *Employee Hub*

Gambar 3.2 merupakan struktur program yang terdapat fitur *employee hub*. Akan tetapi, fokus pengembangan dan pembuatan yang dilakukan oleh praktikan terdapat pada struktur *Employee Hub* yang sangat berhubungan dengan alur lainnya, yaitu *Your Attendance*.



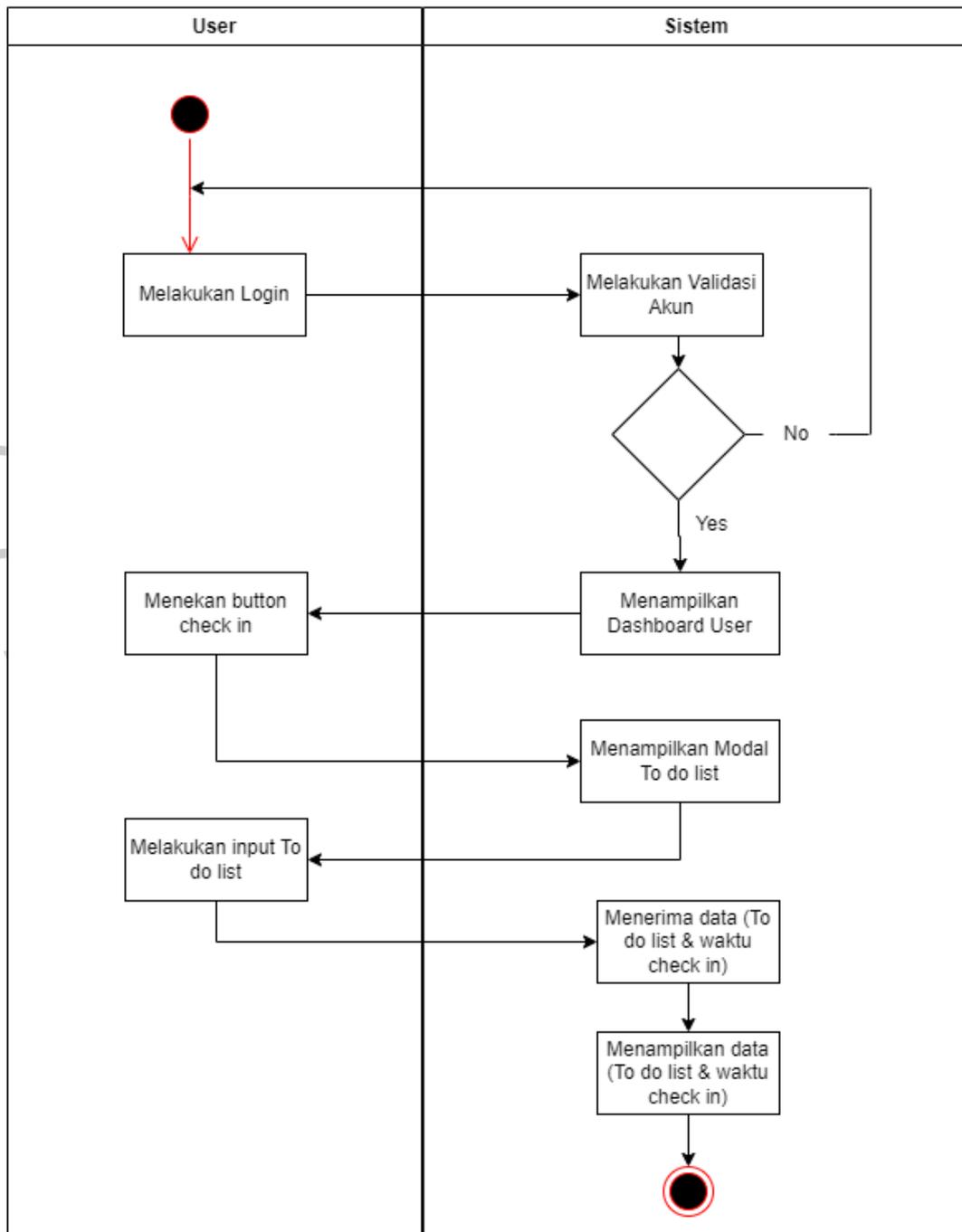
Gambar 3. 3 Rancangan Use Case Diagram *Employee Hub*

Gambar 3.3 adalah *Use Case Diagram* pada fitur *Employee Hub*. Fitur ini memiliki tiga *use case* utama, yaitu absensi, *record*, dan *logout*. *Use case diagram* adalah rancangan fungsional dalam suatu sistem yang memiliki tujuan untuk menggambarkan interaksi antara sistem dengan pelakunya atau yang biasa disebut dengan *actor*. (Rahmatuloh & Rizky Revanda, 2022)

Setelah perancangan *use case diagram* selesai dibuat, kemudian praktikan merancang *activity diagram* sesuai dengan seluruh aktivitas yang tercakup pada *use case diagram employee hub*. *Activity diagram* berfungsi untuk

mendeskripsikan sebuah alur aktivitas kerja dari sebuah proses ataupun sistem perangkat lunak. Pada dasarnya diagram ini menggambarkan sebuah rangkaian aktivitas. (Musthofa & Adiguna, 2022)

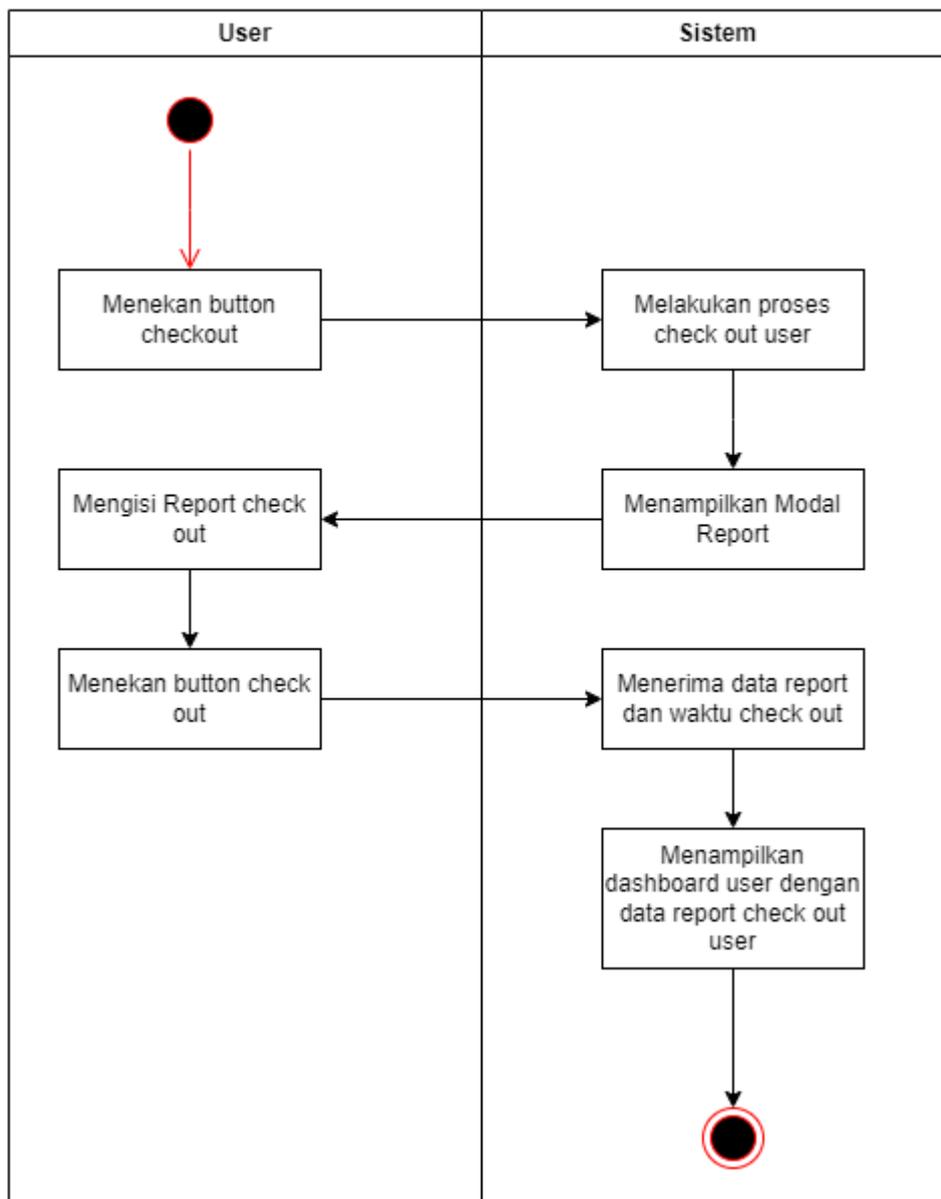
Activity Diagram Melakukan Absensi Masuk



Gambar 3. 4 Rancangan Activity Diagram Absensi *Check in*

Gambar 3.4 merupakan rancangan aktivitas ketika *user* melakukan absensi masuk. Data *to do list* yang diisi akan otomatis tersimpan pada *database* dan dapat dilihat sesuai keinginan *user*.

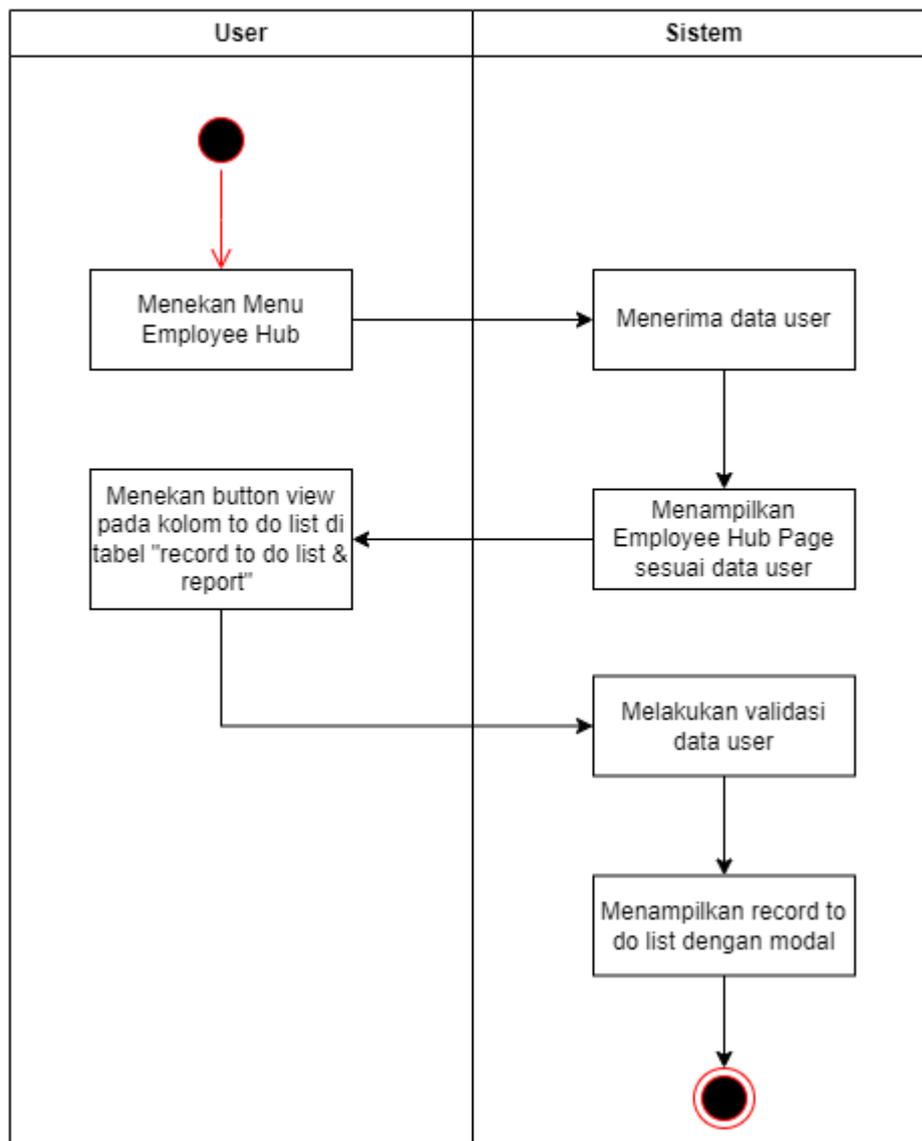
Melakukan Absensi Keluar



Gambar 3. 5 Rancangan Activity Diagram Absensi *Check out*

Gambar 3.5 merupakan rancangan aktivitas ketika user melakukan absensi keluar. Data *report* yang diisi oleh simpan akan disimpan secara otomatis ke dalam *database* dan dapat dilihat serta di cetak sesuai keinginan *user*. Aktivitas tersebut dapat dilihat dalam rancangan diagram aktivitas lain pada fitur ini.

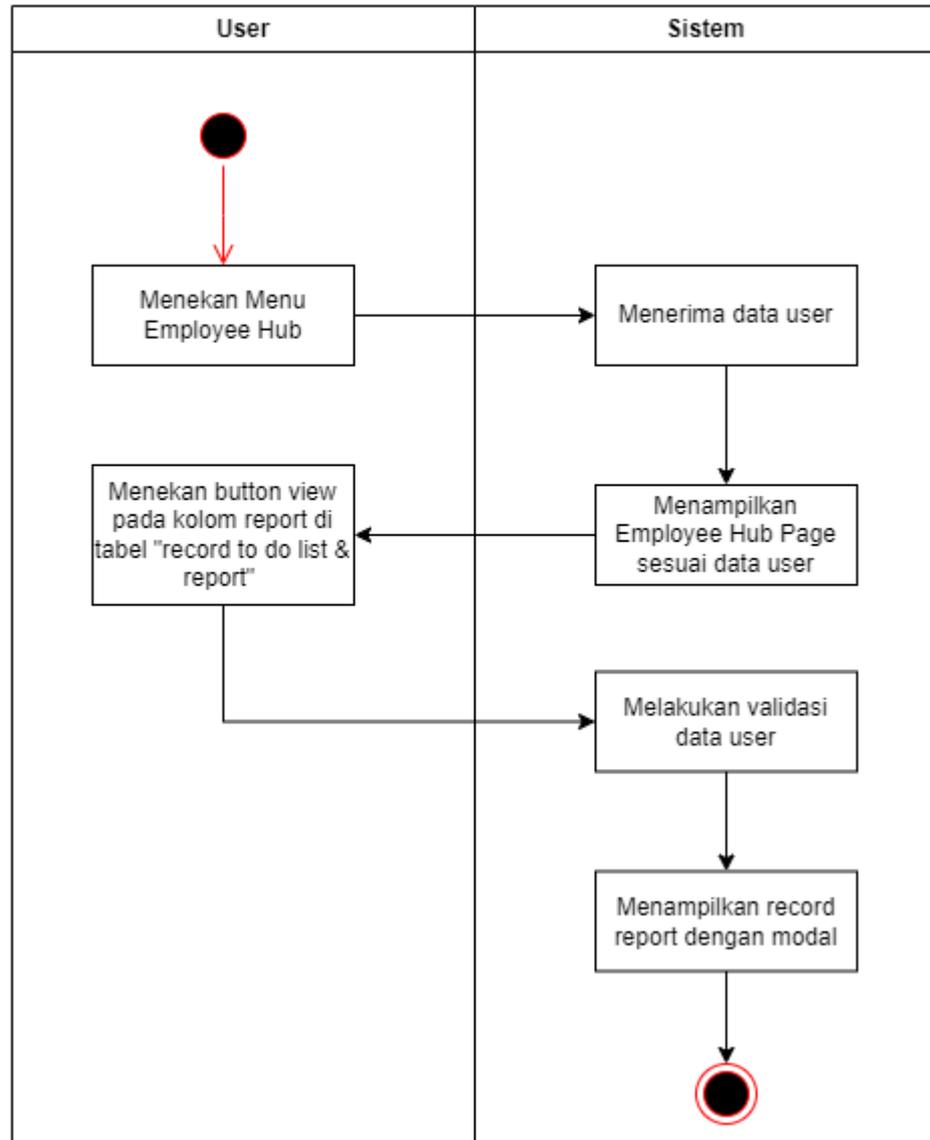
Melihat Record To Do List



Gambar 3. 6 Rancangan Activity Diagram *Record Todolist*

Gambar 3.6 merupakan rancangan aktivitas ketika user ingin melihat *record* (riwayat) terkait informasi *to do list* yang disimpan ketika melakukan *check in*.

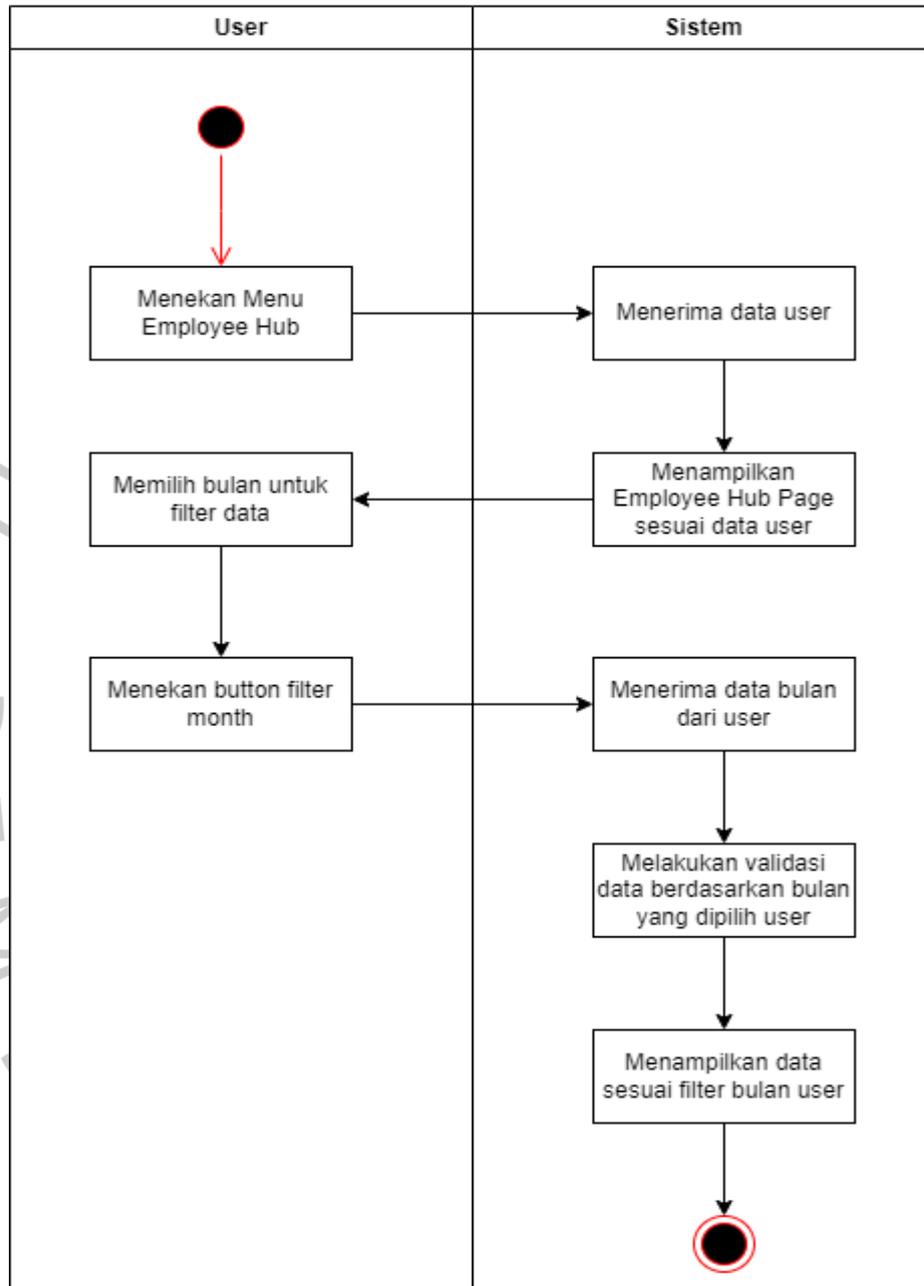
Melihat Record Report



Gambar 3. 7 Rancangan Activity Diagram *Record Report*

Gambar 3.7 merupakan rancangan aktivitas ketika user ingin melihat *record* (riwayat) terkait informasi *report* yang disimpan ketika melakukan *check out*.

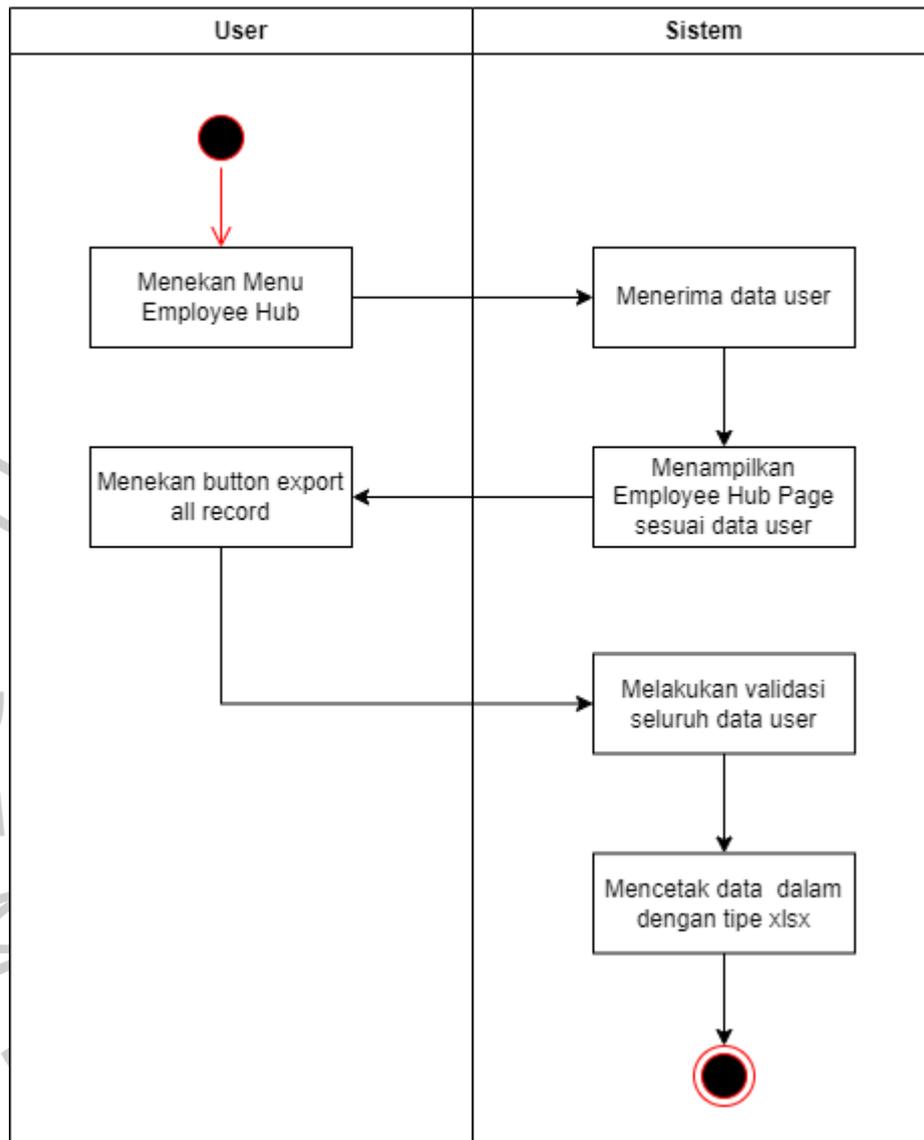
Melakukan filter Record (To do list dan Report) Berdasarkan Bulan



Gambar 3. 8 Rancangan Activity Diagram Record Filter by Month

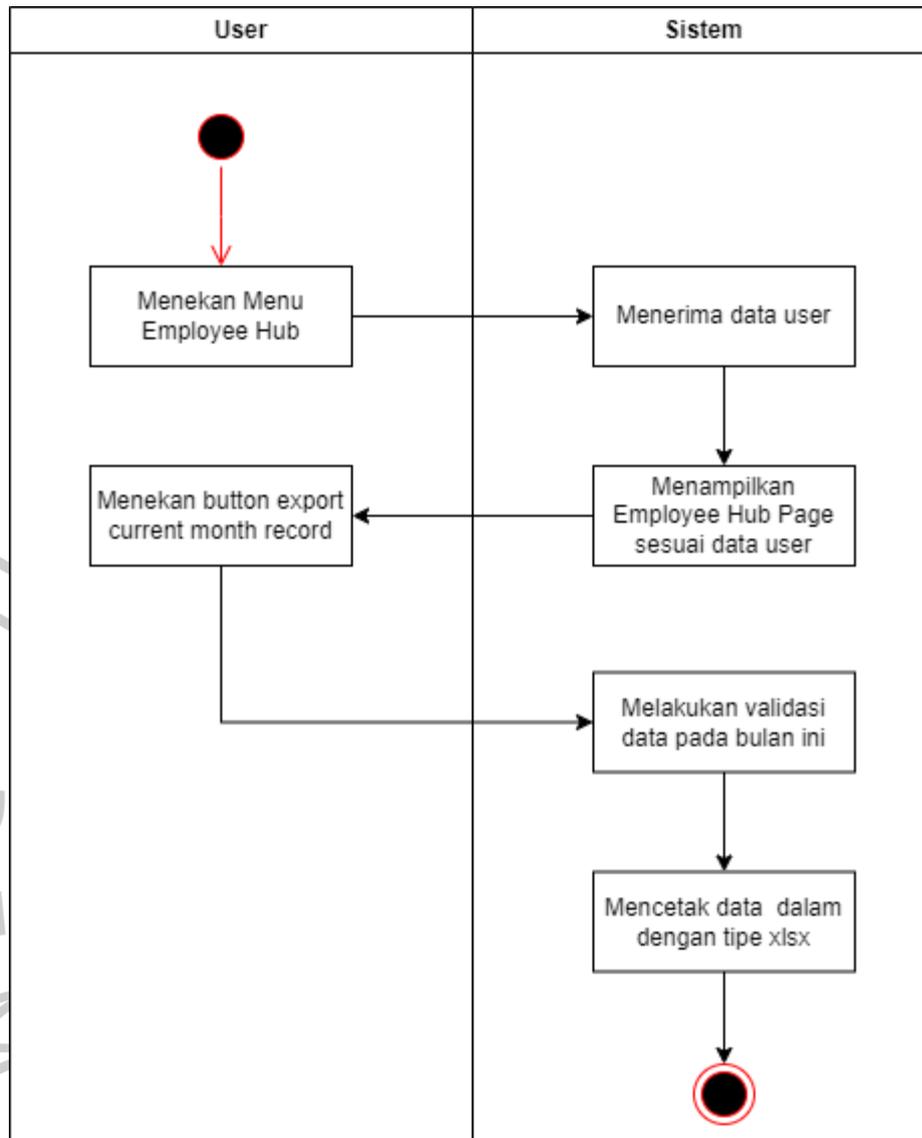
Gambar 3.8 merupakan rancangan aktivitas ketika user ingin melihat data absensi berdasarkan bulan yang dipilih. Data tersebut mencakup *to do list* dan *report* yang *disimpan* ketika melakukan *chek in* dan *check out*.

Mencetak Seluruh Data Record (To do list dan Report) dalam Excel



Gambar 3. 9 Rancangan Activity Diagram *Export All Record*

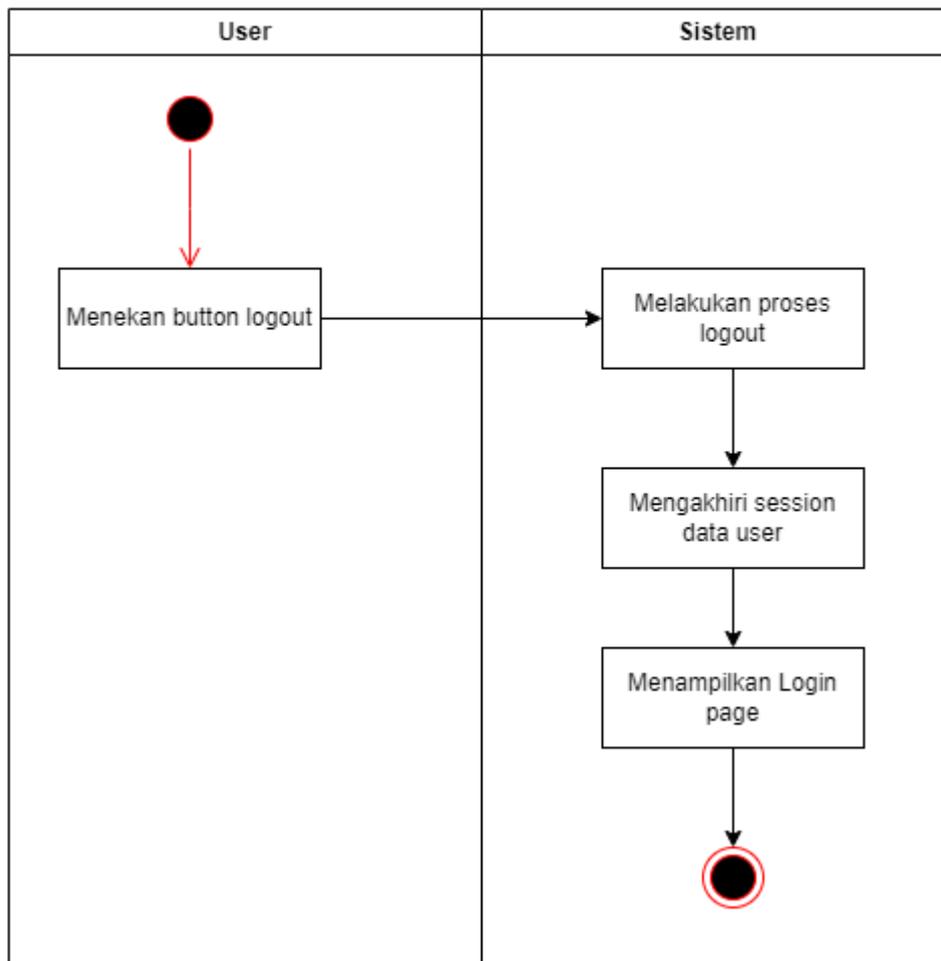
Gambar 3.9 merupakan rancangan aktivitas ketika user ingin mencetak seluruh data *record* absensi, baik *to do list* maupun *report* dan waktu *check in* serta *check out* yang tersimpan dalam database. Data tersebut akan dicetak dalam tipe *file xlsx*.



Gambar 3. 10 Rancangan Activity Diagram *Export by Month*

Gambar 3.10 merupakan rancangan aktivitas ketika user ingin mencetak seluruh data *record* absensi berdasarkan bulan yang dipilih, baik *to do list* maupun *report* dan waktu *check in* serta *check out* yang tersimpan dalam database. Data tersebut akan dicetak dalam tipe *file xlsx*.

Melakukan Logout

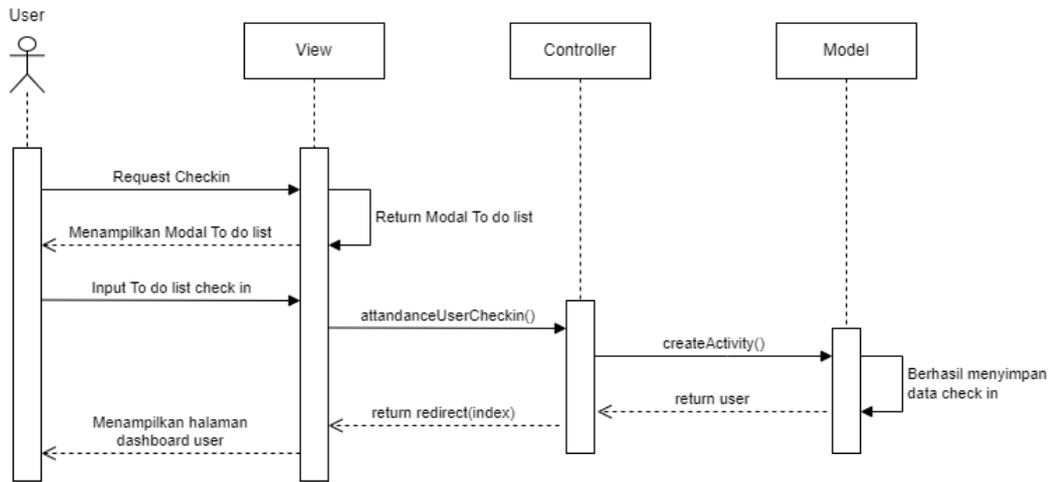


Gambar 3. 11 Rancangan Activity Diagram *Logout*

Gambar 3.11 adalah rancangan aktivitas ketika user ingin melakukan *logout*. Sistem akan mengakhiri data *session user* secara otomatis dan mengembalikan *user* pada halaman *login*.

Setelah seluruh *activity* diagram selesai dirancang. Kemudian praktikan membuat *sequence* diagram sesuai dengan rancangan *use case* sebelumnya. *Sequence* diagram berfungsi untuk mendeskripsikan hubungan antara objek dengan sistem melalui sebuah pesan (*message*) dalam satu kurun waktu yang tertentu. Diagram ini mencakup dimensi horizontal dan vertikal (Kurniawan & Syarifuddin, 2020).

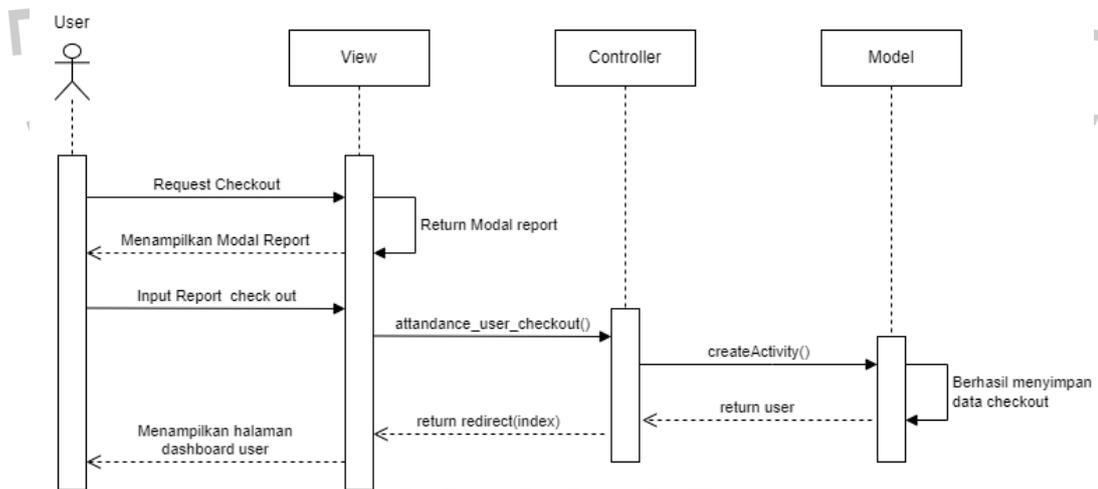
Sequence Diagram Absensi Check in



Gambar 3. 12 Rancangan Sequence Diagram Absensi *Check in*

Gambar 3.12 diatas merupakan rancangan *sequence* diagram ketika user ingin melakukan absensi *check in*.

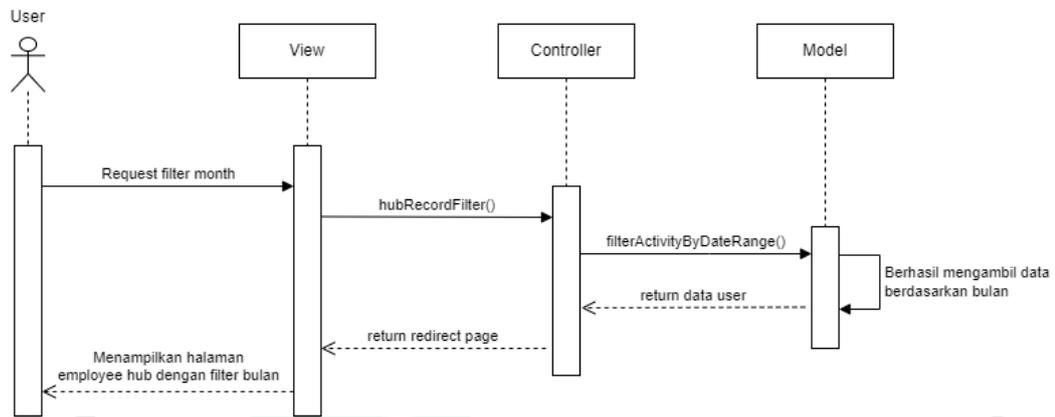
Sequence Diagram Absensi Checkout



Gambar 3. 13 Rancangan Sequence Diagram Absensi *Check out*

Gambar 3.13 merupakan rancangan *sequence* diagram ketika user ingin melakukan absensi *check out*.

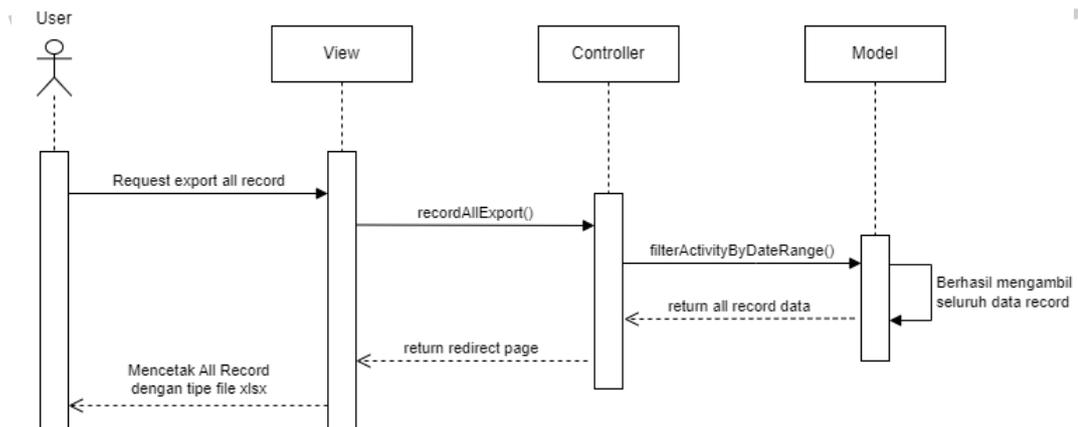
Sequence Diagram Record Filter Month



Gambar 3. 14 Rancangan Sequence Diagram *Record Filter Month*

Gambar 3.14 merupakan rancangan *sequence* diagram ketika user ingin melakukan filter data record untuk melihat data absensi berdasarkan bulan tertentu.

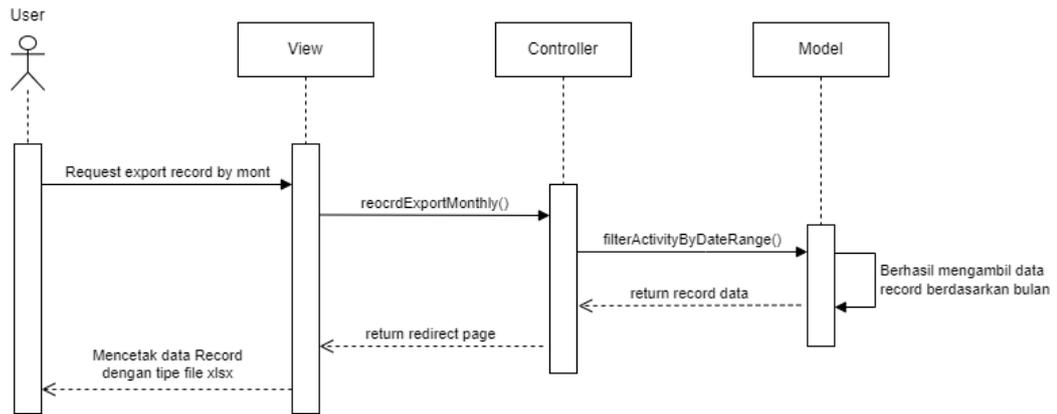
Sequence Diagram Record Export All



Gambar 3. 15 Rancangan Sequence Diagram *Export All Record*

Gambar 3.15 merupakan rancangan *sequence* diagram ketika user ingin mencetak data seluruh absensi yang tersimpan pada database. Data tersebut akan dicetak dengan tipe *file* xlsx.

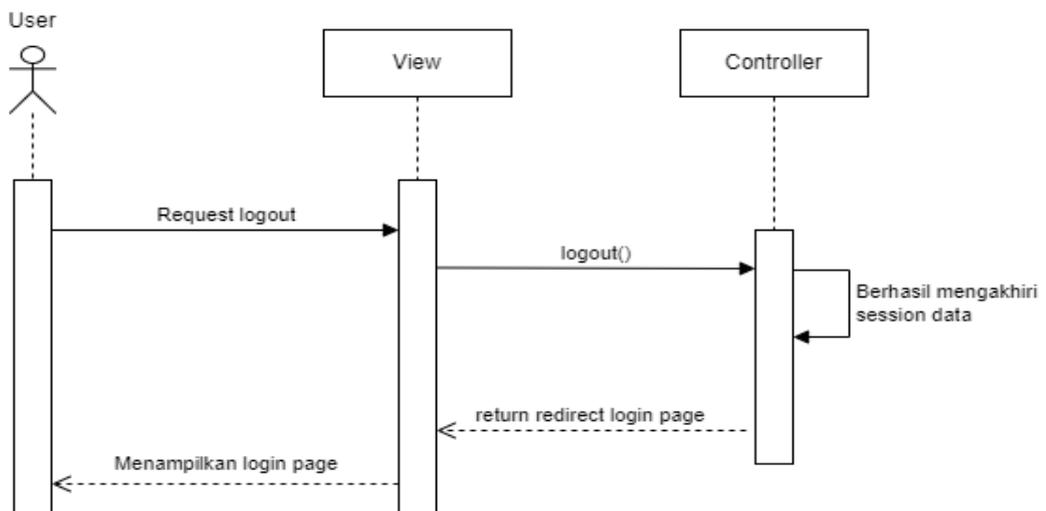
Sequence Diagram Record Export By Month



Gambar 3. 16 Rancangan Sequence Diagram *Export by Month*

Gambar 3.16 merupakan rancangan *sequence* diagram ketika user ingin mencetak data absensi berdasarkan bulan tertentu. Data tersebut juga akan dicetak dengan tipe *file* *xlsx*.

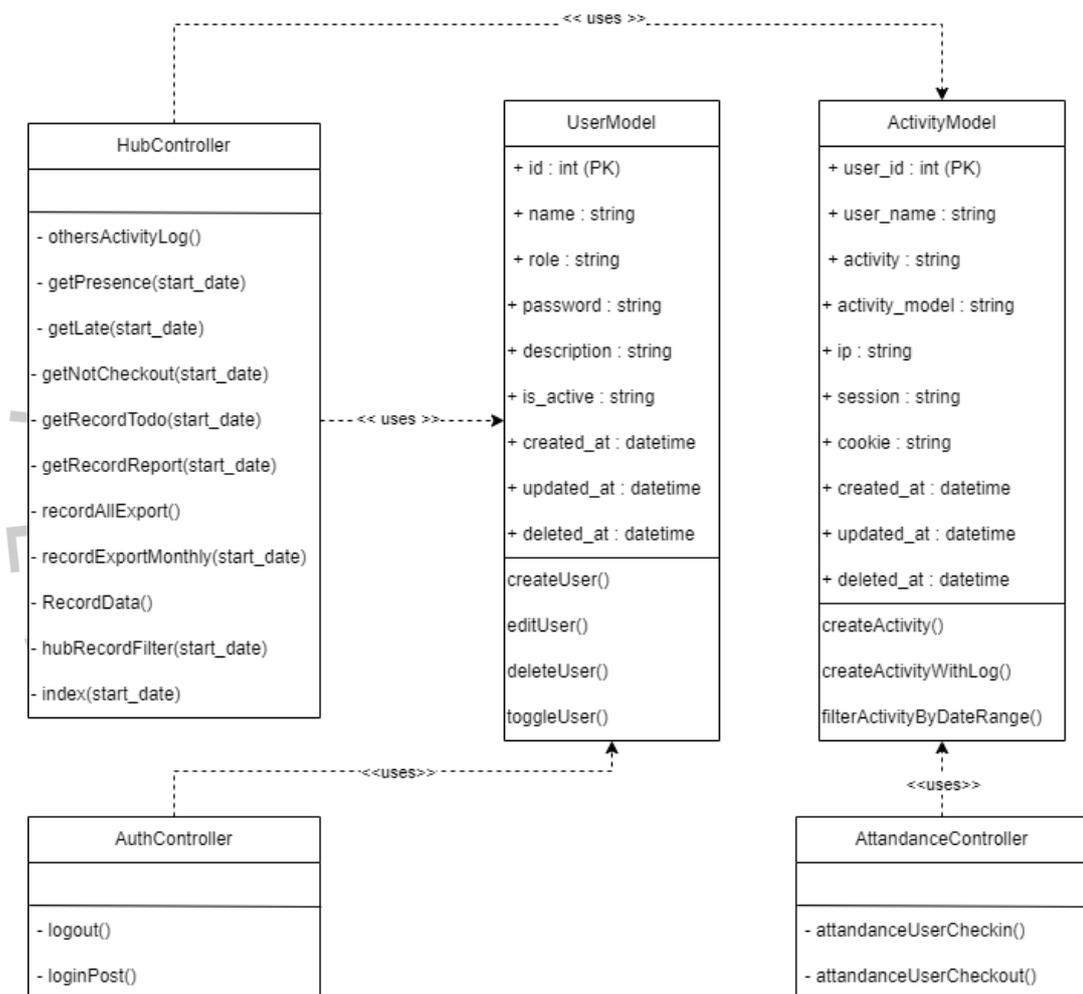
Sequence Diagram Logout



Gambar 3. 17 Rancangan Sequence Diagram Logout

Gambar 3.17 merupakan rancangan *sequence* diagram ketika user ingin melakukan *logout*. Ketika proses tersebut berhasil dieksekusi maka, sistem akan mengembalikan user kepada *view login*.

Class Diagram MVC Employee Hub

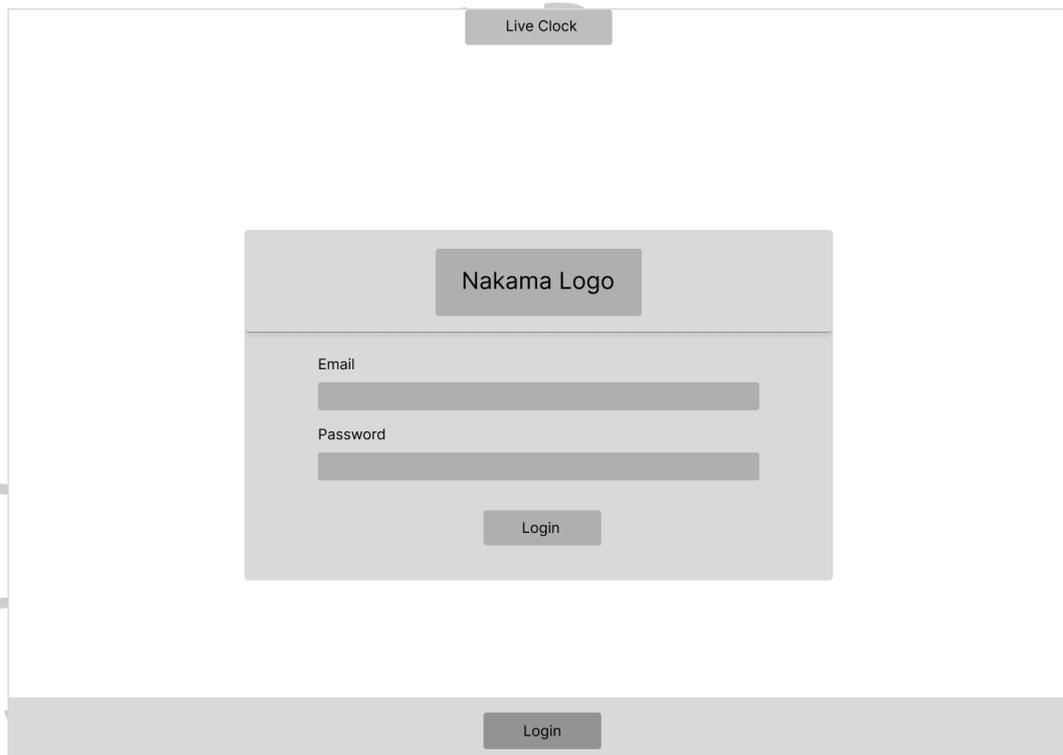


Gambar 3. 18 Class Diagram *Employee Hub*

Gambar 3.18 menjelaskan hubungan antara *controller* dan model yang memiliki asosiasi terhadap model ketika melaksanakan eksekusi program. Seluruh fungsi yang terdapat pada *controller* akan digunakan pada proses *back-end* maupun *front-end*.

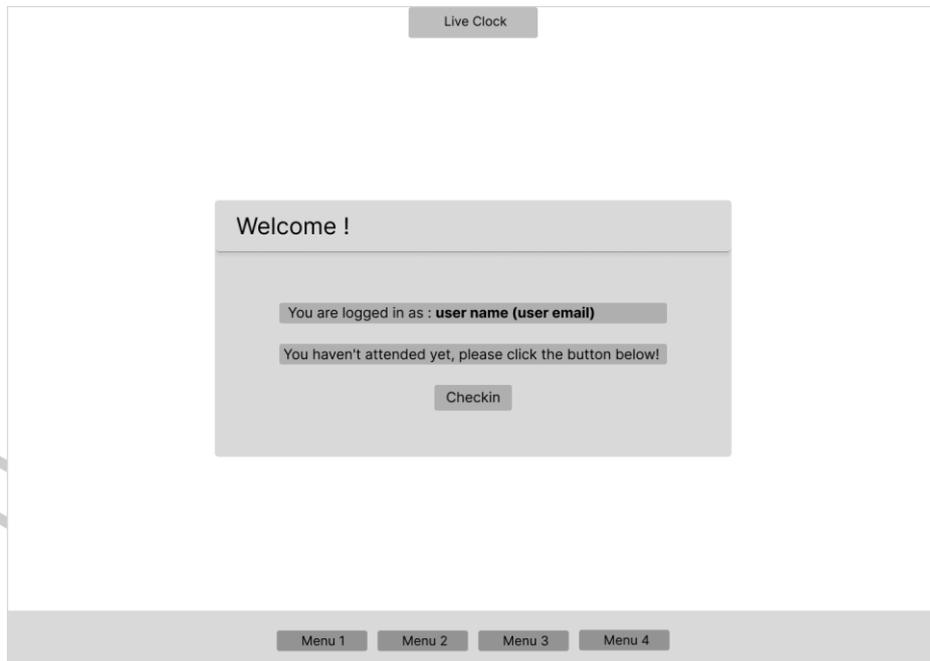


Dalam membangun fitur *employee hub* tentunya dibutuhkan rancangan *user interface* agar implementasi program lebih mudah diatur dan pengguna dapat menggunakan fitur tersebut dengan mudah. Praktikan membuat tampilan *user interface* fitur ini dalam bentuk *mockup* sebelum melakukan implementasi program.



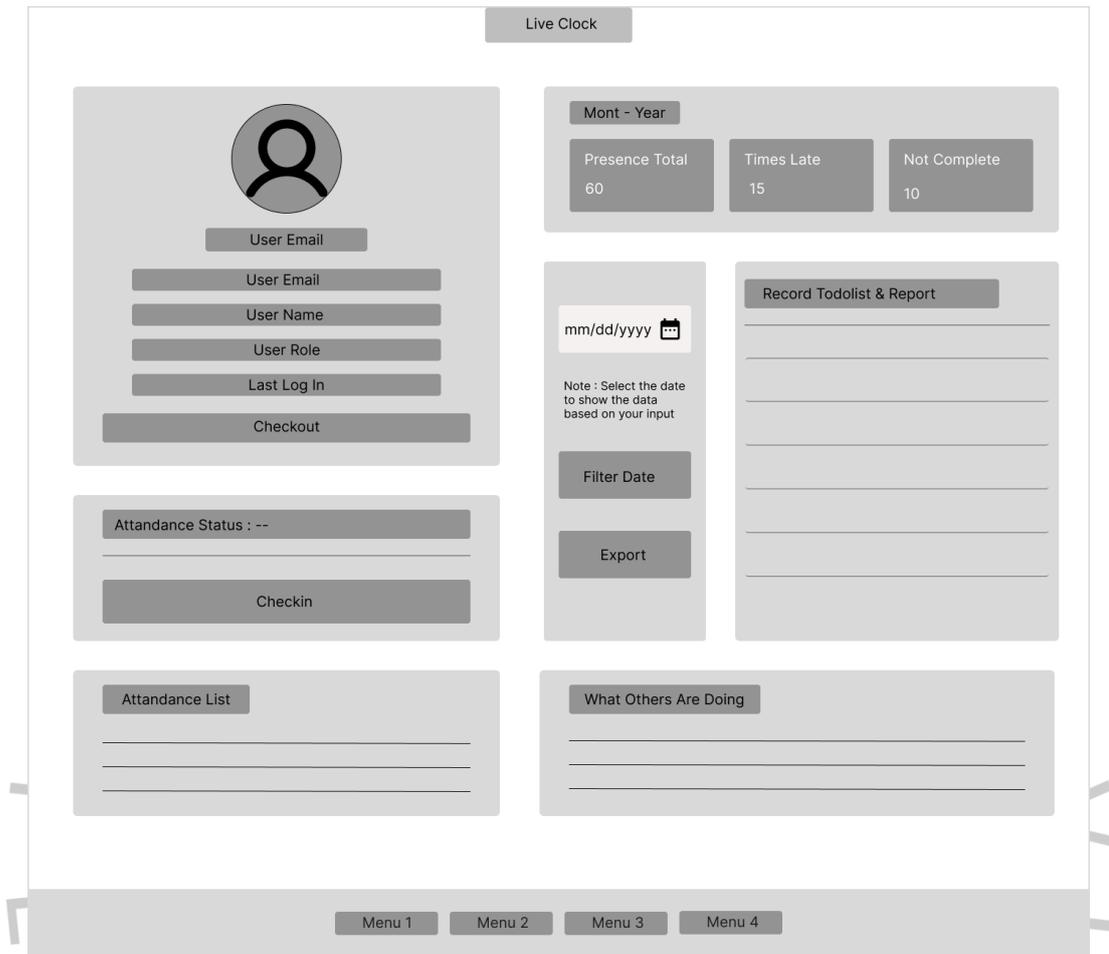
Gambar 3. 19 Mockup Login Page

Gambar 3.19 adalah desain mockup pada halaman *login*. Pada sistem absensi Nakama Id setiap karyawan akan memasuki halaman *login* terlebih dahulu dan akan diminta melakukan input *email* dan *password* untuk dilakukan validasi data.



Gambar 3. 20 Mockup Dashboard User

Gambar 3.20 adalah kerangka halaman *user interface dashboard* utama ketika user berhasil login sebagai karyawan. Pada halaman ini user dapat mengakses 4 menu yang terdapat pada bagian *navigation bar* di bawah tampilan *dashboard*. Halaman tersebut dapat diakses ketika user sudah berhasil melakukan login. *Controller* akan melakukan operasi back-end untuk melakukan autentikasi terkait data *email* dan *password* yang dimiliki oleh user. Jika data tersebut berhasil ditemukan, maka *controller* akan mengembalikan *view dashboard* kepada *user*. Akan tetapi, jika data *user* tidak ditemukan pada *database*, maka *controller* akan mengembalikan user kepada *view login*.



Gambar 3. 21 Mockup Employee Hub Page

Gambar 3.21 merupakan tampilan halaman *employee hub* dengan hak akses karyawan. Pada halaman tersebut karyawan dapat melihat berbagai informasi, seperti informasi akun, *record list* yang mencakup *list to do* ketika *check in* dan *report* ketika *check out*, dan informasi keseluruhan statistik karyawan (jumlah hadir, jumlah terlambat, dan jumlah tidak melakukan *check out*).

3.2.2 Tahap Pembuatan Fitur

Employee Hub merupakan sebuah fitur yang dibuat dan dikembangkan oleh praktikan untuk seluruh karyawan Nakama Id guna melihat berbagai informasi terkait absensi dan dapat di akses oleh berbagai *role*, baik “admin” maupun “employee” berbasis *website*. Adapun informasi tersebut mencakup, total seluruh karyawan yang melakukan absensi (*check out dan check out*), informasi mengenai list to do yang diisi oleh seluruh karyawan ketika melakukan *check out* serta *report* yang diisi ketika mereka melakukan *check out*.

Terdapat spesifikasi perangkat lunak yang digunakan oleh praktikan dalam membuat fitur tersebut. Informasi detail perangkat lunak terdapat pada tabel dibawah ini.

Tabel 3. 2 Kebutuhan Perangkat Lunak

No.	Detail Perangkat Lunak	Kebutuhan Perangkat Lunak
1.	<i>Operating System</i>	Windows 11
2.	<i>Local Web Server</i>	Apache 2.4.4.7
3.	<i>Front-end & Back-end Framework</i>	Codeigniter 4
4.	<i>Database</i>	MySQL 5.7.33
5.	IDE	Visual Studio Code
6.	<i>Scripting Language</i>	PHP 7.4.33
7.	<i>Development Server</i>	Laragon 5.0.0

Adapun informasi detail spesifikasi perangkat keras yang digunakan oleh praktikan dalam membuat fitur tersebut tercakup pada **tabel 3.3**.

Tabel 3. 3 Kebutuhan Perangkat Keras

No.	Detail Perangkat Keras	Kebutuhan Perangkat Keras
1.	<i>Processor</i>	AMD Ryzen 5 3550H
2.	VGA	Nvidia GTX 1650 4 GB
3.	<i>Memory</i>	16 GB

Spesifikasi perangkat keras yang tercakup pada tabel diatas mencakup *processor*, *VGA*, dan *memory* yang digunakan oleh praktikan dalam mengembangkan fitur *employee hub*.

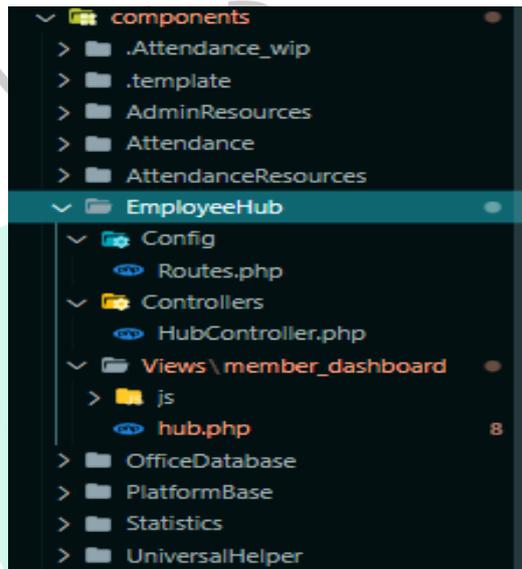
Pembuatan dan implementasi *Back-end* serta *Front-end* pada kedua fitur tersebut menggunakan sebuah *framework*. *Framework* adalah kumpulan aturan-aturan yang membentuk sebuah interaksi antara fungsi atau perintah dalam pembuatan aplikasi dengan tujuan mempermudah *programmer* dalam merancang kode program. (Nabila & Amnur, 2021).

Adapun *Framework* yang digunakan dalam membuat dan mengembangkan fitur ini, yaitu menggunakan Codeigniter 4. *Framework* ini bersifat fleksibel dalam operasi *Back-end* maupun *Front-end* untuk mengembangkan aplikasi berbasis website. Codeigniter merupakan sebuah *framework* yang dibuat dengan tujuan untuk memudahkan programmer membangun sebuah aplikasi berbasis web. *Framework* ini dibangun menggunakan bahasa pemrograman PHP dan memiliki waktu eksekusi yang cepat. (Sallaby & Indra, 2020).

Penyimpanan yang digunakan praktikan dalam membuat fitur tersebut, yaitu menggunakan basis data MySQL. Basis data adalah serangkaian data dan informasi secara terstruktur yang dapat diakses menggunakan kode program pada suatu perangkat komputer. (Helmud, 2021). Sedangkan MySQL adalah salah satu sistem penyimpanan data yang bersifat dinamis dan berbasis web yang dapat dimanajemen secara relasional atau tergolong ke dalam RDBMS (Relational Database Management Sistem). Selain itu, basis data ini juga memiliki struktur kode (*query*) yang sederhana atau yang biasa dikenal dengan SQL (Structured Query Language) (Hidayat, Yani, Rusidi, & Saadulloh, 2019). Sistem basis data tersebut dibangun terlebih dahulu dengan tujuan untuk menyimpan data yang akan dioperasikan melalui *Back-end* dan *fetching* data dari sisi *Front-end*. Selain itu, praktikan juga melakukan pencadangan *database* pada server lokal yang tersimpan pada *device* (perangkat) pribadi sebagai backup jika suatu saat terjadi masalah pada server.

Penerapan *framework* Codeigniter 4 yang digunakan dalam mengembangkan fitur ini, yaitu menggunakan struktur *code module*. *Code module* merupakan sebuah komponen modular yang digunakan untuk mengelola dan mengorganisir kerangka kerja pada sebuah aplikasi di dalam *framework* tersebut. Modul-modul tersebut dipisah dengan tujuan untuk mempermudah pengembangan dan pemeliharaan aplikasi. Adapun beberapa manfaat lain

penerapan *code module* dalam pembuatan aplikasi menggunakan *framework* ini, yaitu organisasi yang lebih baik, reusabilitas (dapat diterapkan pada komponen yang berbeda), skalabilitas (modifikasi modul yang mudah), dan pengujian yang lebih baik. Seluruh komponen modular dalam fitur Employee Hub tercakup dalam folder “components”.



Gambar 3. 22 Implementasi Code Module

Gambar 3.22 merupakan daftar komponen modular yang digunakan dalam sistem absensi, akan tetapi fokus pengembangan fitur *Employee Hub* berada dalam folder “*EmployeeHub*” dan akan saling berhubungan dengan komponen lain, seperti “*PlatformBase*” yang berfungsi sebagai komponen autentikasi (login dan register), “*Attendance*” dan “*AttendanceResources*” sebagai komponen utama yang menyimpan seluruh data absensi.

Tahapan utama dalam membuat fitur *Employee Hub*, yaitu dengan mempersiapkan dan membuat migrasi *database* terlebih dahulu, sebagai penyimpanan utama data dalam fitur ini. Informasi detail mengenai daftar tabel yang digunakan dalam membuat fitur *Employee Hub* tertera pada **Gambar 3.23**.

Name ^	Rows	Size	Created	Updated	Engine	Comment	Type
activities	6	32.0 KiB	2023-05-09 10:22:29		InnoDB		Table
assets	0	16.0 KiB	2023-05-09 10:22:29		InnoDB		Table
ci_sessions	4	16.0 KiB	2023-05-09 10:22:29		InnoDB		Table
migrations	0	16.0 KiB	2023-05-09 10:22:29		InnoDB		Table
statistics_logs	0	32.0 KiB	2023-05-09 10:22:29		InnoDB		Table
users	3	32.0 KiB	2023-05-09 10:22:29		InnoDB		Table
vendors	0	16.0 KiB	2023-05-09 10:22:29		InnoDB		Table

Gambar 3. 23 Daftar Tabel Employee Hub

Gambar di atas merupakan daftar tabel yang digunakan dalam membuat fitur Employee Hub. Seluruh tabel tersebut memiliki relasi dalam operasional data untuk mencegah terjadinya redundant pada suatu atribut/kolom. Akan tetapi, tabel utama yang digunakan dalam membuat fitur ini, yaitu “activities” dan “users”. Fungsional utama tabel “activities”, yaitu untuk menyimpan keseluruhan aktivitas user, seperti informasi *list to do* dan *report* yang dilakukan ketika *check in* dan *check out* serta waktu *user* melakukan aktivitas tersebut. Sedangkan fungsi tabel “users”, yaitu untuk menyimpan informasi akun pribadi, seperti nama, *email*, *password*, dan informasi waktu pembuatan setiap akun yang dilakukan oleh admin.

Beberapa implementasi operasi Back-end yang dilakukan pada fitur ini, yaitu untuk kebutuhan sistem *login* dan menyimpan informasi *list to do* ketika karyawan melakukan *check in* serta *report* ketika karyawan melakukan *check out*. Seluruh operasi *Back-end* tersebut akan sangat berhubungan dengan setiap atribut di dalam tabel yang sudah dirancang sebelumnya.

```

1  <?php
2
3  namespace Components\Attendance\Models;
4
5  use CodeIgniter\Model;
6
7  use function App\Helpers\getIP;
8
9  15 references | 0 implementations
10 class ActivityModel extends Model
11 {
12     3 references
13     protected $DBGroup          = 'default';
14     18 references
15     protected $table            = 'activities';
16     29 references
17     protected $primaryKey       = 'id';
18     6 references
19     protected $useAutoIncrement = true;
20     7 references
21     protected $insertID         = 0;
22     9 references
23     protected $returnType       = 'array';
24     11 references
25     protected $useSoftDeletes   = false;
26     2 references
27     protected $protectFields    = true;
28     5 references
29     protected $allowedFields    = [
30         'user_id',
31         'user_name',
32         'activity',
33         'activity_log',
34         'ip',
35         'session',
36         'cookie',
37         'created_at',
38         'updated_at',
39         'deleted_at',
40     ];
41

```

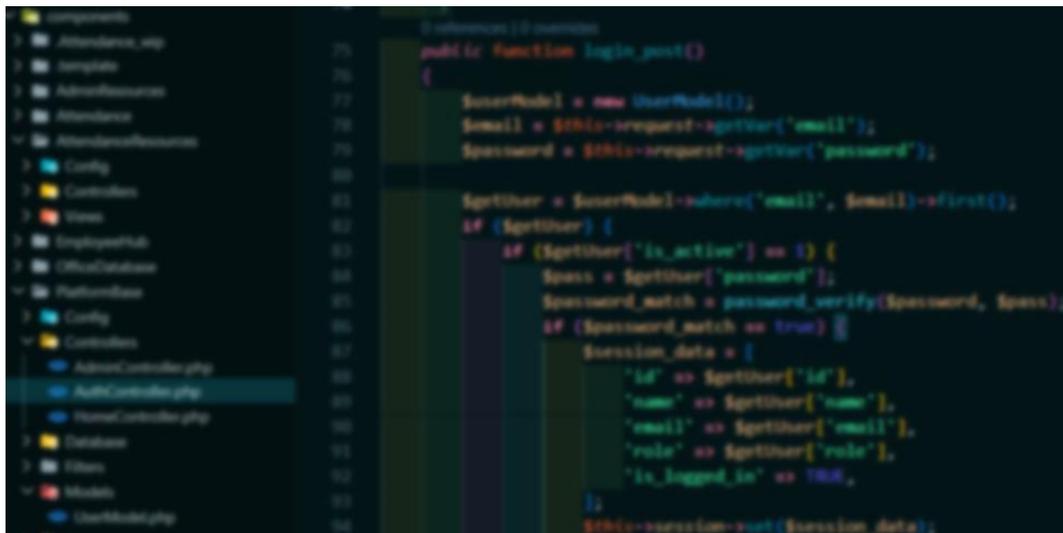
Gambar 3. 24 Activity Model Employee Hub

Gambar 3.24 mencakup model “ActivityModel” yang digunakan untuk merepresentasikan seluruh atribut/kolom yang tertera pada tabel “activities” yang tersimpan pada *database*. Seluruh atribut tersebut harus di deklarasi pada sebuah model sebelum digunakan pada sebuah *controller*.

```
1 <?php
2
3 namespace Components\PlatformBase\Models;
4
5 use CodeIgniter\Model;
6
7 class UserModel extends Model
8 {
9     protected $DBGroup = 'default';
10    protected $table = 'users';
11    protected $primaryKey = 'id';
12    protected $useAutoIncrement = true;
13    protected $insertID = 0;
14    protected $returnType = 'array';
15    protected $useSoftDeletes = true;
16    protected $protectFields = true;
17    protected $allowedFields = [
18        'name',
19        'email',
20        'role',
21        'password',
22        'description',
23        'is_active',
24    ];
```

Gambar 3. 25 User Model Employe Hub

Gambar 3.25 mencakup model “*UserModel*” yang digunakan untuk merepresentasikan seluruh atribut/kolom yang tertera pada tabel “*users*” yang tersimpan pada *database*. Seluruh atribut tersebut juga harus di deklarasikan pada sebuah model sebelum digunakan pada sebuah *controller*. Fungsional atribut tersebut akan sangat berfungsi pada operasi autentikasi yang akan dilakukan pada bagian *Back-end* dan penarikan data pada bagian *Front-end*, melalui berbagai halaman yang sudah dibuat oleh praktikan sebelumnya.



```
public function login_post()
{
    $searchModel = new UserModel();
    $email = $this->request->get('email');
    $password = $this->request->get('password');

    $getfinder = $searchModel->where('email', $email)->first();
    if ($getfinder) {
        if ($getfinder['is_active'] == 1) {
            $pass = $getfinder['password'];
            $password_match = password_verify($password, $pass);
            if ($password_match == true) {
                $session_data = [
                    'id' => $getfinder['id'],
                    'name' => $getfinder['name'],
                    'email' => $getfinder['email'],
                    'role' => $getfinder['role'],
                    'is_logged_in' => true,
                ];
                $this->session->put($session_data);
            }
        }
    }
}
```

Gambar 3. 26 Back-end Controller Method Login

Gambar 3.26 merupakan method “login_post” yang terletak pada *controller* “AuthController”. Method ini berfungsi untuk melakukan autentikasi pada sebuah akun, apabila akun tersebut sudah terdaftar dan memiliki status aktif serta sesuai antara email dan password.



```
} else {
    return redirect()->to($this->login_redirect)->with('msg', 'Password is incorrect.');
```

```
} else {
    return redirect()->to($this->login_redirect)->with('msg', 'Account is not approved/activated yet.');
```

```
} else {
    return redirect()->to($this->login_redirect)->with('msg', 'Email does not exists.');
```

```
}
```

Gambar 3. 27 Back-end Respon Method Login

Gambar 3.27 merupakan sebuah respon yang akan dikembalikan kedalam Front-end sebuah view, jika login tersebut berhasil ataupun gagal ketika dieksekusi pada operasi Back-end.

```

1  * @php
2
3  namespace Components\EmployeeHub\Controllers;
4
5  use App\Controllers\BaseController;
6  use Components\Attendance\Models\ActivityModel;
7  use Components\PlatformBase\Models\UserModel;
8
9  /**
10   * Attendance List Source
11   */
12  class HubController extends BaseController
13  {
14      /**
15       * Attendance List Source
16       */
17      public function attendanceListSource()
18      {
19          $this->month = date("m");
20          $this->day = date("d");
21
22          $activityModel = new ActivityModel();
23          $userModel = new UserModel();
24
25          $getUsers = $activityModel->
26              ->where('created_at <=', '2021-' . $this->month . '-' . $this->day . ' 00:00:00') // start hari ini jam 00:00:00
27              ->where('created_at >', '2021-' . $this->month . '-' . $this->day . ' 23:59:00') // akhir hari ini jam 23:59:00
28              ->select('user_id')->distinct('user_id')
29              ->findAll();
30
31          //PREPARE SQL DATA FROM TABLE
32          //Prepared() data check in dan check out pada setiap input activity di database
33          $getUsersData = [];
34          foreach ($getUsers as $user) {
35              //Prepared() data check in pada table attendance, please let the activity check in with when $user($created_at)
36          }
37      }
38  }

```

Gambar 3. 28 Back-end Controller Method attendanceListSource

Gambar 3.28 merupakan method “attendanceListSource” yang terletak pada controller “HubController”. Adapun fungsi method tersebut untuk menarik seluruh data karyawan yang telah melakukan check out dan check in .

```

77
78  public function userAttendanceDataSource()
79  {
80      $SESSION["checked_in"] = 0;
81      $activityModel = new ActivityModel();
82      $this->month = date("m");
83      $this->day = date("d");
84
85      $data = [
86          "checked_in" => 0,
87      ];
88
89      # Melakukan check user checkin
90      $getUserCheckIn = $activityModel
91          ->where('user_id', $SESSION["id"])
92          ->where('activity', 'check_in')
93          ->where('created_at <=', '2021-' . $this->month . '-' . $this->day . ' 00:00:00')
94          ->where('created_at >', '2021-' . $this->month . '-' . $this->day . ' 23:59:00')
95          ->select('created_at as checked_in_time, activity_log')
96          ->first();
97
98      # ambil report kemarin
99      $getYesterdayReport = $activityModel
100         ->where('user_id', $SESSION["id"])
101         ->where('activity', 'check_out')
102         ->where('created_at >', $this->month . '-' . $this->day . ' 00:00:00')
103         ->orderBy('created_at', 'DESC')
104         ->first();
105
106      $data["previous_report"] = "";
107      if (!empty($getYesterdayReport)) {
108          $data["previous_report"] = $getYesterdayReport["activity_log"];
109      }
110
111      if (!empty($getUserCheckIn)) {
112          $data["checked_in"] = 1;
113          $data["today_log"] = $getUserCheckIn["activity_log"];
114          $data["checked_in_time"] = $getUserCheckIn["checked_in_time"];
115      }
116  }

```

Gambar 3. 29 Back-end Controller Method userAttendanceDataSource

Gambar 3.29 merupakan method “userAttendanceDataSource” yang terletak pada controller “HubController”. Adapun beberapa fungsi method tersebut, yaitu untuk melihat report yang telah dikerjakan oleh karyawan pada hari sebelumnya, dan melakukan validasi, jika karyawan telah melakukan check out maka tidak dapat melakukan check out kembali di hari yang sama.

```

public function userActivityLog()
{
    $activityModel = new ActivityModel();
    $this->day = date("d");
    $this->month = date("m");

    $query = $activityModel
        ->where('activity', 'check_in')
        ->where('created_at <=', '2023-' . $this->month . '-' . $this->day . ' 00:00:00')
        ->where('created_at >=', '2023-' . $this->month . '-' . $this->day . ' 23:59:59')
        ->select('user_name, activity_log')
        ->find();

    return $query->getResult();
}

```

Gambar 3. 30 Back-end Controller Method userActivityLog

Gambar 3.30 merupakan *method* “userActivityLog” yang terletak pada *controller* “HubController”. Fungsional utama *method* tersebut, yaitu untuk menarik data *list to do* yang diisi oleh karyawan pada hari yang sama, sehingga memudahkan admin dalam memonitoring data *list to do* tersebut.

```

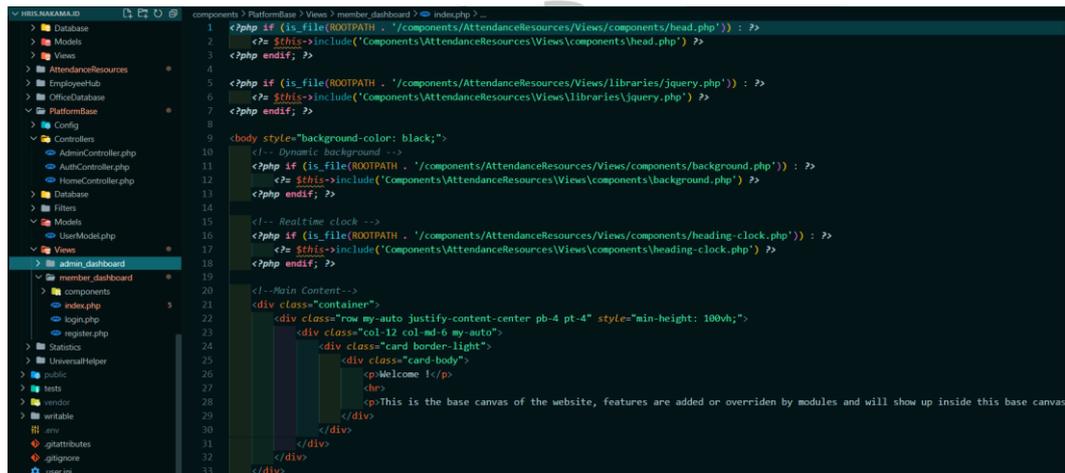
0 references | 0 overrides
public function index()
{
    $data = [
        'Activity' => $this->userActivityLog(),
        'attendance_data' => [],
        'getPresence' => 0,
        'getLates' => 0,
        'getNotCheckOut' => 0,
        'get_all_list' => $this->attendanceListSource(),
        'activity_data' => $this->userActivityLog(),
    ];
    return view('Components\EmployeeHub\Views\member_dashboard\hub', $data);
}

```

Gambar 3. 31 Back-end Controller Method Index

Gambar 3.31 merupakan *method* *index* yang berfungsi menampung seluruh respon dari setiap *method* yang terdapat di *controller* “HubController”. Respon yang diberikan akan disimpan pada variabel “data” dalam bentuk *array*.

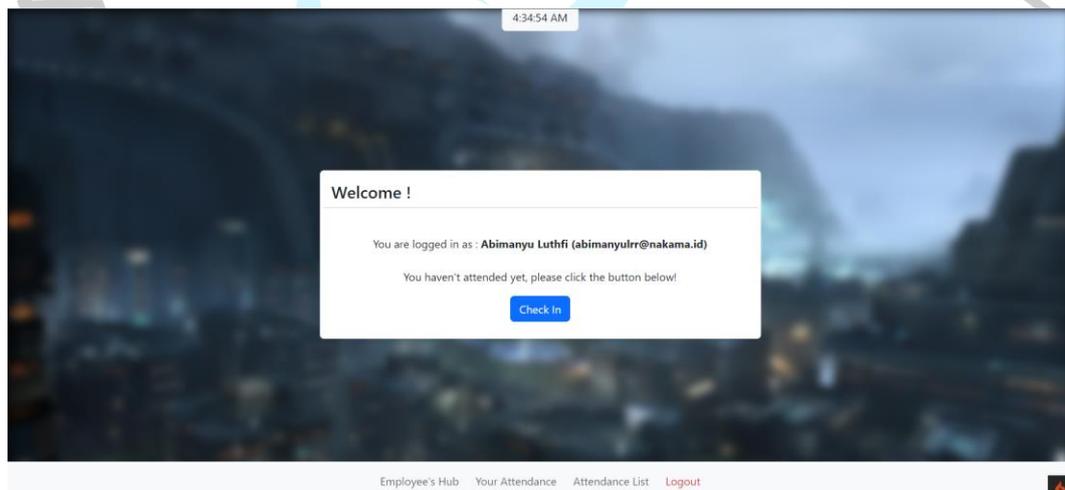
Implementasi operasi *Front-end* pada fitur ini akan menerima data yang diolah pada proses *Back-end* terlebih dahulu melalui sebuah *controller* karena framework ini bersifat *Model View Controller* (MVC).



```
1 <?php if (is_file(ROOTPATH . "/components/AttendanceResources/Views/components/head.php")) : ?>
2 <?= $this->include("Components\AttendanceResources\Views\components\head.php") ?>
3 <?php endif; ?>
4
5 <?php if (is_file(ROOTPATH . "/components/AttendanceResources/Views/libraries/jquery.php")) : ?>
6 <?= $this->include("Components\AttendanceResources\Views\libraries\jquery.php") ?>
7 <?php endif; ?>
8
9 <body style="background-color: black;">
10 <!-- Dynamic background -->
11 <?php if (is_file(ROOTPATH . "/components/AttendanceResources/Views/components/background.php")) : ?>
12 <?= $this->include("Components\AttendanceResources\Views\components\background.php") ?>
13 <?php endif; ?>
14
15 <!-- Real time clock -->
16 <?php if (is_file(ROOTPATH . "/components/AttendanceResources/Views/components/heading-clock.php")) : ?>
17 <?= $this->include("Components\AttendanceResources\Views\components\heading-clock.php") ?>
18 <?php endif; ?>
19
20 <!-- Main Content -->
21 <div class="container">
22 <div class="row my-auto justify-content-center pb-4 pt-4" style="min-height: 100vh;">
23 <div class="col-12 col-md-6 my-auto">
24 <div class="card border-light">
25 <div class="card-body">
26 <p>Welcome !</p>
27 <hr>
28 <p>This is the base canvas of the website, features are added or overridden by modules and will show up inside this base canvas.</p>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
```

Gambar 3. 32 Penggalan Kode Front-end 1

Gambar 3.32 merupakan penggalan kode, jika pengguna berhasil login dengan hak akses “*employee*” atau karyawan. *View* tersebut akan menerima data login yang sudah diolah pada *controller* “*AuthController*”.



Gambar 3. 33 Tampilan Halaman Dashboard

Gambar 3.33 adalah antarmuka *user interface* utama karyawan ketika berhasil melakukan login dan harus melakukan *check in* terlebih dahulu

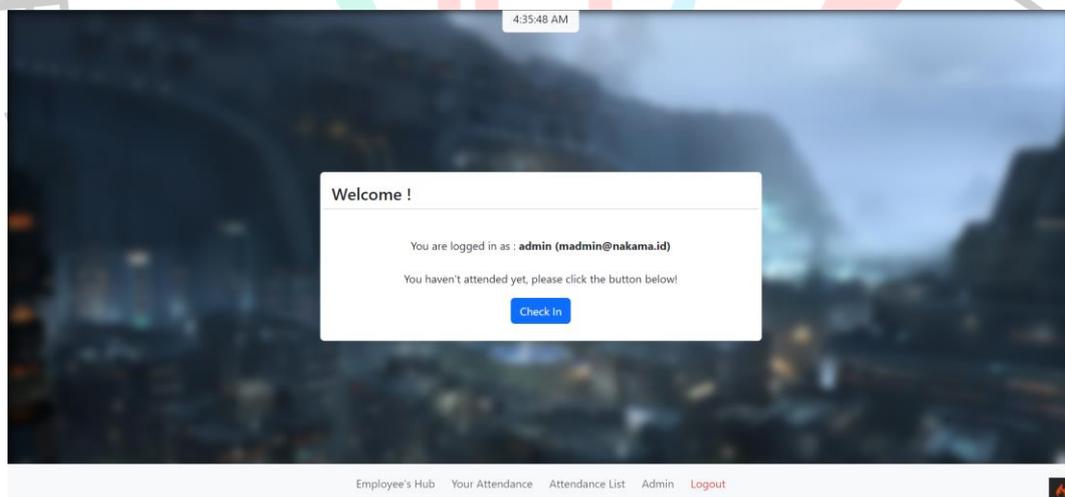
```

1 <!DOCTYPE html>
2
3 <?php if(is_file(ROOTPATH . '/components/AdminResources/Views/components/head.php')): ?>
4 <?= $this->include('Components/AdminResources/Views/components/head.php') ?>
5 <?php endif; ?>
6
7 <body class="hold-transition sidebar-mini">
8 <div class="wrapper">
9 <?php if(is_file(ROOTPATH . '/Components/PlatformBase/Views/admin_dashboard/components/navbar.php')): ?>
10 <?= $this->include('Components/PlatformBase/Views/admin_dashboard/components/navbar.php') ?>
11 <?php endif; ?>
12
13 <?php if(is_file(ROOTPATH . '/Components/PlatformBase/Views/admin_dashboard/components/sidebar.php')): ?>
14 <?= $this->include('Components/PlatformBase/Views/admin_dashboard/components/sidebar.php') ?>
15 <?php endif; ?>
16
17 <!-- Content Wrapper. Contains page content -->
18 <div class="content-wrapper">
19 <!-- Content Header (Page header) -->
20 <div class="content-header">
21 <div class="container">
22
23 <div class="row">
24 <div class="col-12">
25 <!-- Buttons Horizontal Grid -->
26 <div class="card">
27 <div class="card-header">
28 <span><strong>Welcome !</strong></span>
29 </div>
30 <div class="card-body">
31 <p>Here is the admin dashboard, used to put panel or pages for features which requires admin management.</p>
32 <p>Buttons are defined and validated inside this module, while extra pages goes to the respective modules.</p>
33 </div>
34 </div>
35 </div>
36 </div>

```

Gambar 3. 34 Penggalan Kode Front-end 2

Gambar 3.34 merupakan penggalan kode, jika pengguna berhasil login dengan hak akses “admin”. View tersebut akan menerima data login yang sudah diolah pada controller “AuthController” dan dikembalikan kepada class index yang terletak pada file “admin_dashboard”.



Gambar 3. 35 Tampilan Dashboard Admin

Gambar 3.35 mencakup tampilan login dengan hak akses sebagai “admin”. Terdapat beberapa panel yang dapat diakses oleh admin, akan tetapi fokus fitur yang dikembangkan oleh praktikan, yaitu “Employee Hub”.

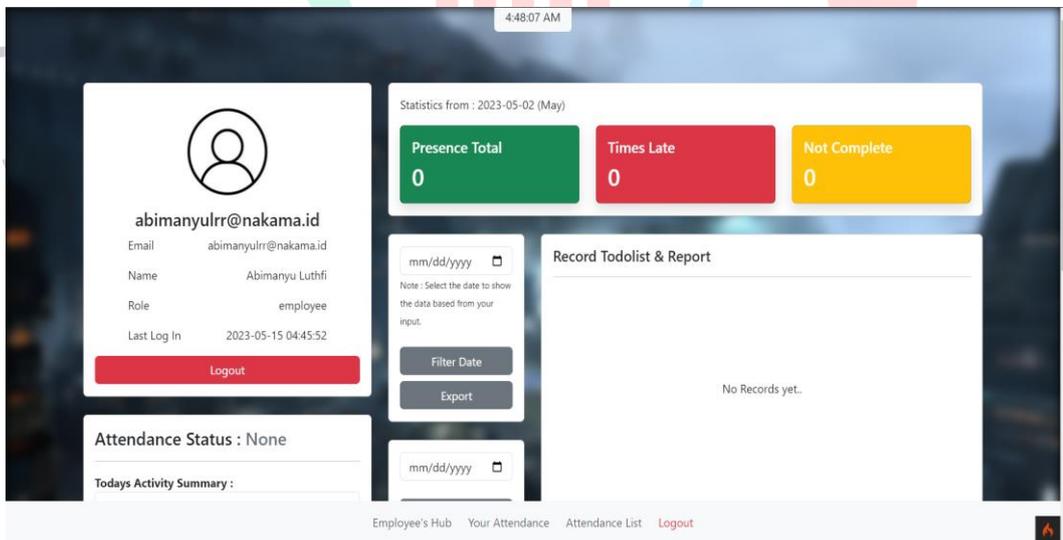
```

1 <?php if (is_file(ROOTPATH . '/components/AttendanceResources/Views/components/head.php')) : ?>
2 <?= $this->include('Components\AttendanceResources\Views\components\head.php') ?>
3 <?php endif; ?>
4
5 <?php if (is_file(ROOTPATH . '/components/AttendanceResources/Views/libraries/jquery.php')) : ?>
6 <?= $this->include('Components\AttendanceResources\Views\libraries\jquery.php') ?>
7 <?php endif; ?>
8
9 <body style="background-color: black;">
10 <!-- Dynamic background -->
11 <?php if (is_file(ROOTPATH . '/components/AttendanceResources/Views/components/background.php')) : ?>
12 <?= $this->include('Components\AttendanceResources\Views\components\background.php') ?>
13 <?php endif; ?>
14
15 <!-- Realtime clock -->
16 <?php if (is_file(ROOTPATH . '/components/AttendanceResources/Views/components/heading-clock.php')) : ?>
17 <?= $this->include('Components\AttendanceResources\Views\components\heading-clock.php') ?>
18 <?php endif; ?>
19
20 <!-- Main Content -->
21 <div class="container-xl mb-4 d-flex justify-content-center" style="min-height: 100vh;">
22 <div class="row my-auto">
23 <div class="col-12">
24 <div class="row mb-4" style="margin-top: 100px;">
25 <div class="col-12 col-lg-4 mb-2">
26 <div class="row mb-4">
27 <!-- (PlatformBase) User Profile -->
28 <div class="col-12">

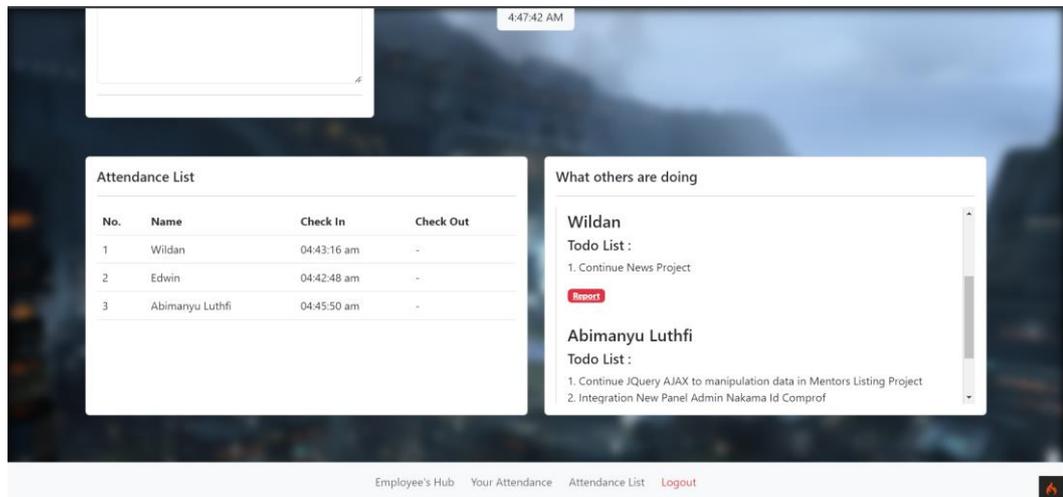
```

Gambar 3. 36 Penggalan Kode Front-end 3

Gambar 3.36 merupakan penggalan kode untuk menampilkan halaman "Employee Hub". View tersebut akan dieksekusi ketika kondisi pada Back-end controller "HubController" Terpenuhi.



Gambar 3. 37 Tampilan User Interface Halaman Employee Hub



Gambar 3. 38 Tampilan User Interface Fitur Employee Hub

Gambar 3.37 dan 3.38 adalah tampilan halaman “Employee Hub” ketika pengguna menekan panel tersebut.

3.3 Kendala Yang Dihadapi

Selama melaksanakan kegiatan kerja profesi di PT. Aseries Informix Solusi Teknologi (Nakama Id), praktikan memiliki beberapa kendala yang ditemukan, baik dalam perancangan fitur maupun aktivitas keseharian kerja profesi. Berikut merupakan beberapa kendala yang dihadapi oleh praktikan:

1. Kesibukan waktu antara praktikan dan pembimbing eksternal dalam bekerja, sehingga terhambatnya waktu dalam berkonsultasi mengenai fitur yang sedang dirancang.
2. Selama melaksanakan kegiatan kerja profesi, terkadang praktikan ditugaskan untuk membantu divisi lain dalam melaksanakan sebuah *event*. Hal tersebut menyebabkan terganggunya jadwal dan prioritas perancangan fitur yang sudah ditetapkan sebelumnya.
3. Suasana ruang lingkup kerja yang kurang kondusif, sehingga praktikan merasa kurang fokus dalam melaksanakan tugas yang sudah ditentukan sebelumnya.

3.4 Cara Mengatasi Kendala

Berikut merupakan beberapa cara yang dilakukan oleh praktikan dalam menghadapi kendala selama melaksanakan kegiatan kerja profesi di PT. Aseries Informix Solusi Teknologi (Nakama Id) :

1. Melakukan penjadwalan khusus dalam konsultasi dan pelaporan terkait kemajuan pembuatan fitur dan tugas yang diberikan oleh pembimbing eksternal.
2. Melakukan manajemen waktu dan menggunakan skala prioritas pada tugas yang lebih mendesak untuk dikerjakan.
3. Berdiskusi dengan pembimbing eksternal untuk fleksibel dalam perpindahan lokasi dalam mengembangkan fitur dan tugas yang diberikan sebelumnya.

3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Selama melaksanakan kegiatan kerja profesi di Nakama Id, praktikan memperoleh berbagai pembelajaran, antara lain:

1. Kemampuan menganalisis dalam perancangan dan pembuatan fitur berbasis website.
2. Pemahaman mengenai keamanan sistem berbasis website (*website security*) dari ancaman pihak ketiga (*third party*).
3. Melatih kemampuan dalam kerjasama tim untuk membuat suatu proyek, sehingga hasil yang diperoleh dapat lebih maksimal.
4. Praktikan menjadi peka terhadap perubahan teknologi dan berbagai *library* pada sebuah bahasa pemrograman yang sebelumnya belum di pernah dipelajari selama menempuh perkuliahan.
5. Praktikan memiliki beberapa pemahaman dan kemampuan baru yang diperoleh ketika membantu divisi lain mengurus suatu event.