

Terbit online pada laman web jurnal: <http://jurnal.iaii.or.id>

JURNAL RESTI

(Rekayasa Sistem dan Teknologi Informasi)

Vol. 4 No. 3 (2020) 521 - 530

ISSN Media Elektronik: 2580-0760

Aplikasi *Scheduller Team Meeting* Berbasis *Mobile* Dengan Menggunakan *Push Notification*

Masmur Tarigan¹, Adven Kristianata²^{1,2}Teknik Informatika, Ilmu Komputer, Universitas Esa Unggul¹Masmur.tarigan@esaunggul.ac.id, ²advenkris37@gmail.com

Abstract

Each team / individual in a company has its own internal and external meetings. At present PT NTX Solusi Teknologi does not have a system that can schedule meetings and track its employees who carry out meetings, it affects the time management for each individual or team that collaborates with each other. The purpose of making this application is for scheduling, notification of meetings, monitoring the team that is conducting the meeting, and also for data collection by the company admin. Because employees use a variety of devices, applications are created using Web Responsive that can adjust the appearance of employee devices and have the same functionality for all devices. The method used in making this application is the Prototype method for gathering needs, designing and evaluating programs. The problem analysis method uses Fishbone analysis to obtain specific problems. The program will be tested with the Black Box method to ensure the system runs well. With this conclusion the scheduler application can be used to make the schedule, tracking members who carry out external meetings, used by various devices and has a responsive display. And based on testing Black Box the application can run well.

Keywords : web responsive, meeting, schedule, notification, team management

Abstrak

Setiap team/perorangan dalam sebuah perusahaan masing-masing mempunyai jadwal pertemuan internal maupun eksternal. Saat ini PT NTX Solusi Teknologi belum mempunyai sistem yang bisa menjadwalkan pertemuan dan pelacakan terhadap karyawannya yang melaksanakan pertemuan, hal itu berpengaruh terhadap pengelolaan waktu untuk masing-masing individu maupun team yang saling berkolaborasi. Tujuan pembuatan aplikasi ini adalah untuk penjadwalan, pemberian notifikasi pertemuan, pemantauan team yang sedang melaksanakan pertemuan, dan juga untuk pendataan oleh admin perusahaan. Karena karyawan memakai berbagai macam perangkat, maka aplikasi dibuat dengan menggunakan *Web Responsive* yang bisa menyesuaikan tampilan perangkat karyawan dan mempunyai fungsionalitas yang sama untuk semua perangkat. Metode yang digunakan dalam pembuatan aplikasi ini adalah metode *Prototype* untuk pengumpulan kebutuhan, perancangan dan evaluasi program. Metode analisis masalah menggunakan analisis *Fishbone* untuk memperoleh permasalahan yang spesifik. Program akan diuji coba dengan metode *Black Box* untuk memastikan sistem berjalan dengan baik. Dengan ini kesimpulan aplikasi *scheduller* dapat digunakan untuk melakukan pendwalan, pelacakan anggota yang melaksanakan pertemuan eksternal, digunakan oleh berbagai perangkat dan mempunyai tampilan yang responsif. Dan berdasarkan pengujian *Black Box* aplikasi dapat berjalan baik.

Kata Kunci : responsif web, pertemuan, penjadwalan, notifikasi, manajemen team

1. Pendahuluan

Setiap pertemuan yang dilaksanakan oleh sebuah team/personal dalam sebuah perusahaan memerlukan berbagai spesialisasi untuk menciptakan kinerja yang baik dan saling bekerja sama.

sebuah manajemen yang baik sehingga tercipta sebuah produktivitas kerja yang baik seperti Tim lintas fungsi yaitu [1] tim lintas fungsi telah digunakan di banyak perusahaan sebagai cara untuk mengatur karyawan dari Pertemuan dalam perusahaan ini ada 2 jenis yaitu internal dan eksternal. Pertemuan internal dilaksanakan dalam lingkup karyawan perusahaan saja dan jika eksternal dilaksanakan di luar perusahaan dengan *client*.

Diterima Redaksi : 16-04-2020 | Selesai Revisi : 16-06-2020 | Diterbitkan Online : 20-06-2020

Dalam pelaksanaannya, pertemuan hanya mengundang orang-orang/team yang mempunyai kepentingan dalam urusan tersebut dan tidak semua diundang [2]. Terkadang undangan pertemuan tidak terbaca karena banyaknya *chat* yang masuk. Juga tidak melakukan konfirmasi terhadap penjadwalan pertemuan.

Dalam pertemuan eksternal, team yang melaksanakan pertemuan di luar area kantor hanya mengirimkan *share location* melalui whatsapp atau telegram kepada *admin* yang akan mendata kegiatan mereka. Sehingga bukti mereka melakukan pertemuan bisa saja hilang jika riwayat *chat* yang dikirimkan terhapus.

Untuk sebuah pertemuan juga diperlukan detail pertemuan seperti agenda pertemuan, lokasi, tanggal, waktu dan dokumen yang diperlukan. Ini menjadi masalah tersendiri jika dokumen yang diperlukan lupa dibawa atau tidak tersedia pada perangkat yang dibawa.

Berdasarkan latar belakang diatas, perusahaan mempunyai beberapa masalah sebagai berikut:

Pertama, perusahaan belum mempunyai aplikasi untuk mendata pertemuan yang akan dilaksanakan oleh masing-masing karyawan, ini sangat penting untuk perusahaan dimana mereka mengelola sumber daya manusia dan pendataan terhadap kegiatan masing-masing karyawan.

Kedua, perusahaan belum mempunyai aplikasi yang bisa melakukan penjadwalan terhadap pertemuan yang akan dilaksanakan secara internal maupun eksternal dan memberikan notifikasi terhadap karyawan yang telah diikuti dalam pertemuan tersebut.

Ketiga, belum ada sistem pelacakan karyawan yang melaksanakan pertemuan di luar kantor dengan *client*. Sistem ini juga digunakan untuk memastikan karyawan dalam kondisi baik ketika melakukan pertemuan di luar kantor.

Tujuan sekaligus manfaat untuk perusahaan pada penelitian ini adalah untuk membuat sebuah sistem penjadwalan pertemuan; membuat sistem untuk pendataan admin; membuat sistem pelacakan terhadap karyawan menggunakan *exif* data pada foto dengan menggunakan *Longitude* dan *Latitude* [3]; membuat fungsi *push* notifikasi untuk memberi tahu pertemuan diterima di setiap perangkat [4]; membuat fungsi konfirmasi kehadiran pada jadwal yang sudah dibuat sehingga team/admin mengetahui anggota yang bisa/tidak bisa hadir.

Aplikasi ini dibangun menggunakan *Web Responsive* yang bertujuan untuk mencakup berbagai perangkat yang digunakan oleh pengguna tetapi memiliki fungsionalitas yang sama, dengan menggunakan *front-end framework* untuk tampilannya yaitu Bootstrap [5] Bootstrap merupakan *framework* untuk membangun desain web secara responsif.

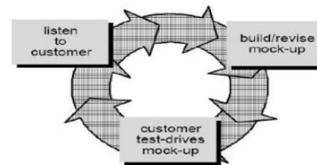
Dalam penelitian ini, pengembangan menggunakan metode *Prototype*, dengan 3 tahapan yaitu pengumpulan

kebutuhan, perancangan serta evaluasi program. Metode ini sangat cocok untuk mengetahui kebutuhan pengguna, sehingga dalam pengembangan maupun evaluasi tidak mengalami kesulitan karena sudah mengetahui masalah yang mereka punya. Metode analisa yang digunakan yaitu *Fishbone*, metode ini untuk mengetahui rincian masalah yang ada. Dalam melakukan validasi sistem menggunakan metode *Black Box*, digunakan sebagai pengujian sistem untuk mendapatkan hasil yang sesuai harapan.

2. Metode Penelitian

Untuk mendapatkan hasil yang sesuai harapan, penelitian ini menggunakan metode pengembangan *Prototype*. Model *Prototype* [6] dapat dilihat pada gambar 1.

MODEL PROTOTYPE



Gambar 1. Metode *Prototype*

Keterangan pada tahap pengembangan dengan metode *Prototype*:

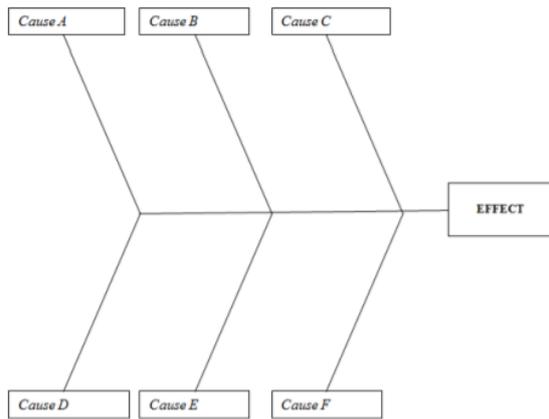
Tahap pertama dalam pengembangan dengan metode *prototype* adalah mendengarkan pengguna (pengumpulan kebutuhan). Pada tahap ini, dilakukan dengan melakukan wawancara pada *Product Manager* dari NTX Solusi Teknologi, wawancara ini meliputi kekurangan dalam perusahaan dan permasalahan yang ada, sehingga bisa menawarkan sebuah solusi dari permasalahan yang ada.

Tahap kedua yaitu pembangunan *mock-up* (perancangan). Pada tahap ini, perancangan berdasarkan kebutuhan sistem, fitur, *input/output*, fungsionalitas dari sistem yang dibuat, dibuat menggunakan UML (*Unified Modelling Language*). Sebelum masuk kepada tahapan pengkodean, tahapan ini harus dijabarkan secara detail untuk digunakan pada tahap pengkodean.

Tahap ketiga yaitu pengujian *mock-up* untuk pengguna (evaluasi), pengguna akan menganalisa cara kerja dari rancangan *mock-up* yang telah dibuat untuk mengetahui fungsionalitas dan memperjelas kebutuhan aplikasi.

Metode analisis masalah menggunakan metode *Fishbone* [7] Diagram tulang ikan (juga disebut diagram Ishikawa atau diagram sebab-akibat) adalah teknik grafis untuk menunjukkan beberapa penyebab peristiwa atau fenomena tertentu, bisa dilihat pada gambar 2.

Pada tahapan analisa ini, dilakukan pengelompokan masalah berdasarkan kesimpulan masalah yang di dapat pada proses wawancara. Hasil analisa masalah akan di kelompokkan dalam tabel 1.



Gambar 2. Metode Fishbone

Tabel 1. Analisis Data

Faktor	No	Hasil Analisis
Data	1	Jadwal pertemuan eksternal tidak terpantau team lain
	2	Jadwal pertemuan belum tersusun

Tabel 2. Analisis Man (Manusia)

Faktor	No	Hasil Analisis
Man	1	Lupa Jadwal pertemuan
	2	Lupa akan jadwal yang bersamaan.

Tabel 3. Analisis Tools (Aplikasi)

Faktor	No	Hasil Analisis
Tools	1	Belum ada aplikasi
	2	Penjadwalan masih belum terstruktur

Tabel 4. Analisis Metode

Faktor	No	Hasil Analisis
Data	1	Penjadwalan masih melalui Chatting
	2	Kerjasama antar team

Tahapan selanjutnya adalah pengkodean program, setelah di dapatkan semua kebutuhan yang disetujui oleh pengguna tentang fungsionalitas dari *mock-up*. Cara kerja program haruslah sesuai dengan rancangan dari *mock-up* yang telah dibuat.

Metode pengujian menggunakan *Black Box testing* [8] pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. Pengujian ini tidak melihat dan menguji *source code* program.

3. Hasil dan Pembahasan

Berdasarkan dari hasil analisa permasalahan yang ada dan pengembangan dengan metode *prototype*, maka dapat diketahui solusi yang didapatkan. Berikut hasil dari analisa dengan menggunakan metode *Prototype*:

3.1 Pengumpulan Kebutuhan

Pertama, yaitu melakukan wawancara untuk mengetahui permasalahan secara detail dari perusahaan dan kebutuhan sistem. Secara garis besar merupakan hasil dari wawancara yang telah dilaksanakan, kebutuhan fungsional sebagai berikut:

Halaman *login Admin* dan karyawan/*user*, untuk mengakses data sesuai kebutuhan mereka; *Push Notification*, digunakan untuk mengirimkan notifikasi jadwal pertemuan yang telah dibuat dan juga sebagai pengingat pertemuan; *Document Uploader*, digunakan untuk mengunggah dokumen yang dibutuhkan pada saat melaksanakan pertemuan; Unggah foto, digunakan untuk melacak karyawan, dengan memanfaatkan *Data Exif Photo (long-lat)* yang tersimpan pada saat mengambil foto, juga sebagai bukti bahwa telah sampai pada pertemuan.

Fitur yang dimiliki oleh *admin* pada aplikasi ini adalah: Fitur pertama adalah *Form Login*, terdiri atas *Username* dan *Password*, digunakan sebagai akses admin menuju *dashboard* utama aplikasi *scheduler*; Fitur kedua *Dashboard admin*, terdiri atas total *schedule*, total *user*, daftar rencana pertemuan terbaru, *pie chart* jadwal, kalender yang terdapat jadwal di dalam tanggalnya untuk memudahkan pengecekan jadwal; Fitur ketiga adalah *Users*, terdiri atas tabel pengguna dan membuat pengguna, digunakan sebagai pengelolaan *user* terhadap akses halaman pengguna, juga terdapat fungsi menambahkan user, hapus user dan *edit user*; Fitur keempat adalah *schedule*, terdiri atas *table* yang memuat informasi jadwal pertemuan, membuat jadwal dan detail jadwal. Fungsional terdiri atas membuat jadwal, menghapus jadwal, mengedit jadwal dan melihat rincian detail jadwal pertemuan.

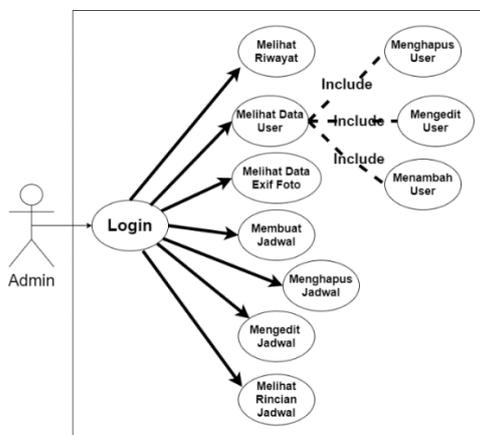
Fitur yang dimiliki oleh pengguna pada aplikasi ini adalah: Fitur pertama adalah *Form Login*, terdiri atas *Email* dan *Password*, digunakan sebagai akses menuju *dashboard* pengguna dan melihat data pertemuan berdasarkan pengguna yang *login*; Fitur kedua adalah *Dashboard*, terdiri *card* semua jadwal, jadwal hari ini, *complete* atau yang telah di selesaikan dan jadwal yang sudah dibuat di hari selanjutnya; Fitur ketiga adalah *form riwayat*, digunakan untuk melihat riwayat pertemuan yang telah terjadwal/dilakukan; Fitur keempat adalah jadwal, fitur ini menggunakan *card* sebagai pengelompokan jadwal yang telah dibuat, fungsionalitas yang terdapat pada *card* ini adalah pengunduh dokumen, mengunggah foto dan pengubah status kehadiran pada pertemuan tersebut; Fitur kelima adalah form tambah jadwal, digunakan untuk membuat jadwal pertemuan yang terdiri dari *form* agenda, deskripsi, status pertemuan (internal/eksternal) tanggal waktu mulai dan berakhirnya pertemuan, pilih anggota yang mengikuti pertemuan, lokasi pertemuan dan unggah dokumen pertemuan.

3.2 Perancangan dan Evaluasi

Tahapan ini adalah perancangan dan analisis solusi untuk dari aplikasi *scheduler*, perancangan ini menggunakan UML (*Unified Modelling Language*) untuk mendapatkan rincian fungsionalitas sebelum ke dalam tahap pengkodean program.

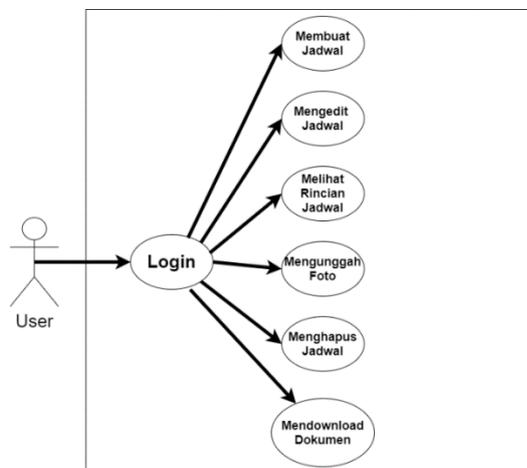
Untuk menggambarkan fungsionalitas yang akan dibuat, maka digambarkan dengan Diagram *Use Case*.

Use Case yang pertama adalah untuk admin sebagai pengelola data karyawan, pengelola data, pengelola jadwal, melihat detail jadwal user dan *exif* data dari foto yang di unggah oleh pengguna. Fungsi ini untuk memenuhi dari perancangan yang diajukan sebagai solusi, dievaluasi oleh pengguna terkait apakah sudah sesuai dengan kebutuhan yang mereka setuju bisa dilihat pada gambar 3.



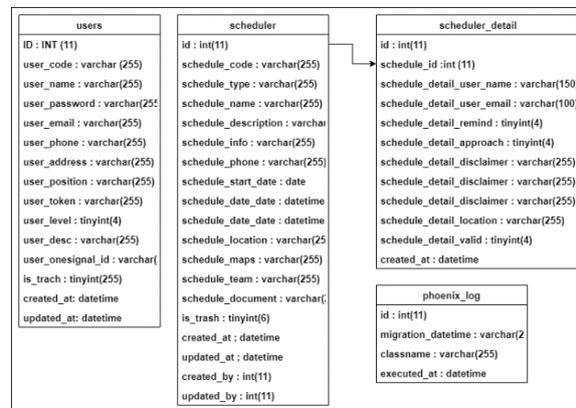
Gambar 3. Diagram *Use Case* Admin

Use Case yang kedua adalah untuk pengguna/*user*, user setelah *login*, maka akan tampil milik mereka di dalam aplikasi *scheduler*. Fungsional seperti membuat jadwal, mengedit jadwal, melihat rincian jadwal, mengunggah foto, menghapus jadwal dan mengunggah dokumen. Semua kebutuhan ini untuk membuat fungsional program yang dibuat, sehingga terarah dan terstruktur dengan baik, bisa dilihat pada gambar 4.



Gambar 4. Diagram *Use Case* User

Selanjutnya adalah desain dari *database* untuk aplikasi ini, *database* digunakan untuk menyimpan data yang akan digunakan pada aplikasi *scheduler*, bisa dilihat pada gambar 5.



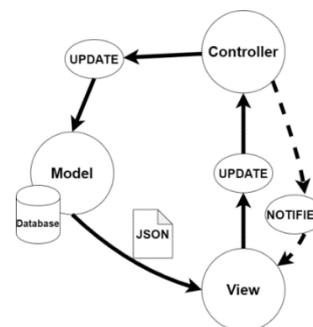
Gambar 5. Desain *Database*

Setelah semua dievaluasi dan memenuhi kebutuhan pengguna, maka masuk ke dalam pengkodean program.

3.3 Pengkodean Program

Pengkodean program ini berdasarkan kebutuhan dan diagram yang telah dibuat, sebagai acuan dari analisa masalah yang ada. Pengkodean atau bisa disebut *deployment* sudah dalam pembuatan program.

Tahapan pengkodean program ini menggunakan Bootstrap sebagai *Framework Front-End* yang bisa merealisasikan tampilan *multi-device* sesuai perangkat pengguna, *Back-End* menggunakan *PHP Symfony* [9]. Untuk menjamin keamanan komunikasi antar data, aplikasi ini dibangun dengan menerapkan sistem MVC (*Model View Controller*) [10] memisahkan program dalam tiga bagian terpisah yang saling terhubung sesuai dengan namanya, yaitu: *Model*, *View*, dan *Controller*, bisa dilihat pada gambar 6.



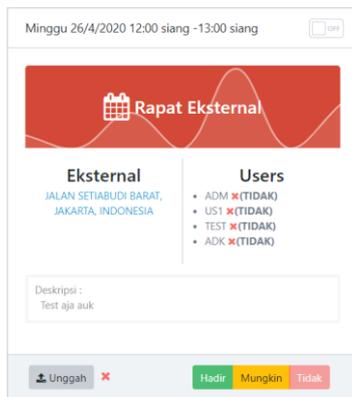
Gambar 6. Model MVC

Dalam penerapannya, semua data yang di request oleh *Front-End*, akan diproses oleh API dan akan di berikan kembali dalam bentuk format JSON (*JavaScript Object Notation*) [11] format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer.

Contoh *payload response* yang didapatkan dan ditampilkan oleh front-end ditampilkan pada gambar 7.

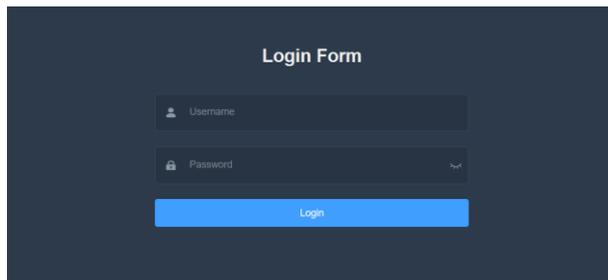
Contoh response JSON

```
{
  "schedule_code":2003270017,
  "schedule_type":1,
  "schedule_name":"Rapat Eksternal",
  "schedule_description":"Main Main aja",
  "schedule_info":null,
  "schedule_phone":8111118282,
  "schedule_start_date":"2020-03-27 15:00:00",
  "schedule_end_date":"2020-03-27 16:00:00",
  "schedule_location":"Jalan Setiabudi Barat, Jakarta, Indonesia",
  "schedule_maps":"106.824567,-6.209305",
  "schedule_team":null,
  "schedule_document":"static/img/Doc.",
  "created_at":"2020-03-27 14:14:39",
  "updated_at":null,
  "created_by":"ADM",
  "updated_by":null,
}
```



Gambar 7. Contoh Response JSON.

Pertama adalah pengkodean serta tampilan pada Admin, tampilan yang pertama muncul adalah halaman *login admin*, tampilan *page login admin* bisa dilihat pada gambar 8.



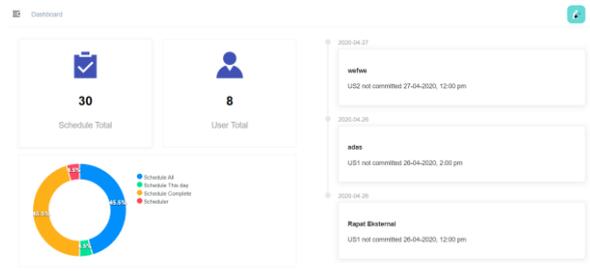
Gambar 8. Login Form Admin

Form login admin terdiri dari *Username* dan *Password*, username, dan sistem login menggunakan *JWT (Java Web Token)*, untuk menginisiasi browser menyimpan token supaya tidak login kembali ketika membuka aplikasi.

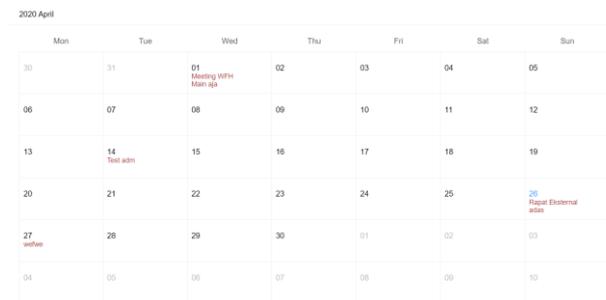
Selanjutnya adalah *Dashboard* admin, tampilan dashboard bisa dilihat pada gambar 9 dan 10.

Dashboard ini dibuat untuk memenuhi *User Experience* [12] dari sisi pengguna yaitu admin, dibuat sedetail

mungkin tetapi mencakup kebutuhan informasi jadwal yang telah dibuat.



Gambar 9. Dashboard Admin



Gambar 10. Dashboard Kalender Admin

Gambar 9 adalah bagian pada *dashboard admin*, yang menampilkan detail jadwal secara singkat. Dan pada Gambar 10 adalah kalender untuk melihat jadwal sesuai bulan yang sedang berjalan.

Pada gambar 10, terdapat kalender yang berisi data pertemuan yang telah dijadwalkan, hal itu membuat admin dengan mudah mengecek pertemuan yang akan dilaksanakan oleh karyawan/pengguna.

Source Code Controller Dashboard

```
function get_data_dashboard(){
  $all = $model->count();
  $day =
  $model->count(array('DATE(schedule_end_date) =
  ?', date("Y-m-d")));
  $complete =
  $model->count(array('DATE(schedule_end_date)
  <= ?', date("Y-m-d")));
  $scheduler =
  $model->count(array('DATE(schedule_end_date)
  >= ?', date("Y-m-d")));
}
```

Source Code berikut digunakan untuk mendapatkan data yang ditampilkan pada *dashboard*.

Selanjutnya adalah Data *User/Pengguna*, dalam halaman ini bisa membuat/menghapus/mengupdate data untuk pengguna yang nantinya digunakan untuk *login* pada aplikasi pengguna. Daftar dari pengguna ditampilkan pada *table* yang berisi *name, email, phone, position, created at, func edit dan func delete* ditampilkan pada gambar 11.

Fungsi untuk menambahkan pengguna terdiri dari *username, user email, user phone, user position, user description, user address, user password dan user password confirmation*, bisa dilihat pada gambar 12.

Fungsi untuk merubah detail, fungsi ini dibuat untuk merubah detail user jika dilakukan perubahan data pada user. Terdiri dari *username*, *user email*, *user phone*, *user position (owner/staff)*, *user description* dan *user address*. Bisa dilihat pada gambar 13.

Name	Email	Phone	Position	Created At	Search
ACM	admin@jabon.com	81111030		Feb 5, 2020 12:28 PM	✖
US1	user1@jabon.com			Feb 7, 2020 6:38 PM	✖
US2	user2@jabon.com			Feb 7, 2020 6:38 PM	✖
US3	user3@jabon.com			Feb 7, 2020 6:40 PM	✖
test	test@jabon.com	9991928	staff	Apr 12, 2020 5:02 PM	✖
Mama	mama@jabon.com	81111111	owner	Apr 12, 2020 5:40 PM	✖
ACM	admin@jabon.com	2302000	staff	Apr 12, 2020 5:40 PM	✖
test	test@jabon.com	93020027	staff	Apr 25, 2020 8:34 PM	✖

Gambar 11. Tabel Data User

Dashboard / Users

* User name

* User email

* User phone

* User position

User description

User address

User password

User password confirmation

Gambar 12. Form Create User

* User name

* User email

* User phone

* User position

User description

User address

Gambar 13. Form Edit User

Selanjutnya adalah fungsi dari *Schedule*, dalam halaman ini admin bisa melihat keseluruhan riwayat dari jadwal yang sudah dibuat, tabel terdiri dari *accordion* untuk melihat sebagian detail jadwal, *code* pertemuan, nama pertemuan, *start*, *end* dan lokasi pertemuan, *func* detail jadwal, *func* merubah jadwal dan *func* hapus.

Dari halaman ini, admin dapat melihat keseluruhan jadwal yang telah dilaksanakan dan terjadwal. Halaman ini sangat penting untuk admin dalam melakukan pendataan, ditampilkan pada gambar 14.

Terdapat juga fungsi membuat jadwal, terdiri dari *Schedule Name*, *Schedule Description*, *Schedule Type*, *Schedule Start Date & Time*, *Schedule Start Date & Time*, *Location*, *Choose User*, *Upload Document*. Ditampilkan pada gambar 15.

Code	Name	Start	End	Location
2002170003	Makan Bareng	2020-03-18 12:00:00	Mar 20, 2020 12:00 PM	Jalan Nginden Baru VIII, Kota Surabaya, Indonesia
2002200004	Menanam Jagung	2020-02-27 12:00:00	Feb 27, 2020 2:00 PM	Jalan Mega Kuningan Barat VIII, Jakarta Pusat, Indonesia

Gambar 14. Tampilan Riwayat Admin

Schedule name

Schedule description

Schedule Type Internal Eksternal

Schedule Start Date & Time

Schedule End Date & Time

User address

Location

Schedule Document

Gambar 15. Tampilan Membuat Jadwal

Selanjutnya adalah halaman detail jadwal, terdiri dari detail dari rincian data pertemuan yang telah dibuat seperti kode pertemuan, nama, tanggal dan waktu, deskripsi, alamat, dibuat pada dan pembuat jadwal.

Selain itu, informasi mengenai tracking melalui *upload* foto dapat dilihat disini juga. Informasi tersebut ditampilkan dengan validasi simbol centang jika data di dalam *exif data photo* yang berupa (*long-lat*) benar sesuai dengan lokasi pertemuan yang ditentukan pada saat membuat jadwal, dan jika silang maka ada 2 kemungkinan yaitu belum mengupload gambar dan lokasi pada *exif data* tidak sesuai dengan lokasi pertemuan. Akan dibahas pada halaman user, karena validasi menggunakan sistem otomatis pada fungsi *upload* gambar. Tampilan bisa dilihat pada gambar 16.

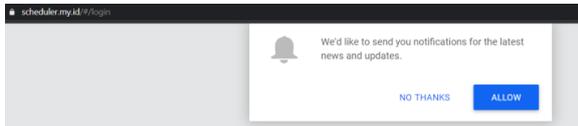
Code: 2002170003
Name: Makan Bareng
Date & Time: Mar 18, 2020 12:00 PM To Mar 18, 2020 12:00 PM
Description: Makan Makan
Address: Jalan Nginden Baru VIII, Kota Surabaya, Indonesia
Created At: 2020-03-17 13:21:32
Created By: ACM

Member					
Name	Email	Attend	Position	Image	Created At
US1	user1@jabon.com	Yes			Feb 17, 2020 1:21 PM
US2	user2@jabon.com	No	730200000,113,7832711278		Feb 17, 2020 1:21 PM

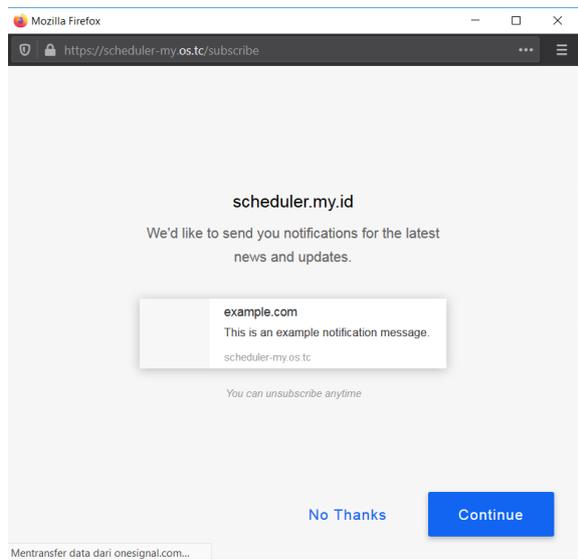
Gambar 16. Tampilan detail jadwal

Pembahasan selanjutnya adalah untuk bagian User/Pengguna. Karena ada fitur yaitu notifikasi, maka

pengguna akan diminta mengizinkan aplikasi untuk mengirimkan notifikasi berupa jadwal pertemuan. Contoh pemberian izin notifikasi aplikasi disertakan pada gambar 17. Karena aplikasi ini mengirimkan banyak notifikasi ke berbagai perangkat, maka akan menggunakan layanan *one signal*, sebagai pengirim notifikasi yang diharuskan untuk *subscribe* sebagai akses dari pemberian id *device* untuk layanan tersebut, dapat dilihat pada gambar 18.

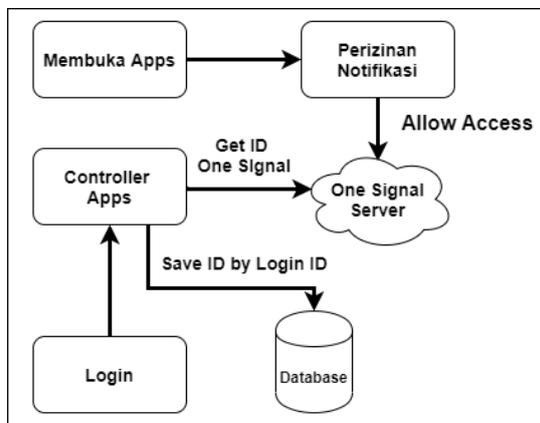


Gambar 17. Allow Permission Notification



Gambar 18. Subscribe One Signal

Setelah melakukan *subscribe* pada *One Signal*, perangkat yang terdaftar akan mendapatkan ID perangkat, *user* harus melakukan *login* untuk mencatat ke dalam *database* dari ID yang di dapatkan dari layanan tersebut. Akan digambarkan pada *working flow* diagram pada gambar 19.



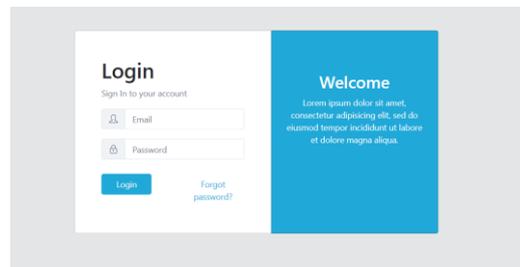
Gambar 19. Working Flow Allow Acces Notification dan memperoleh ID One Signal

Model Push Notifikasi Jadwal

```

public function sendNotification()
{
    $user = new Users();
    $userId = $this->Scheduler_detail
    >getAll('schedule_detail_User_id');
    $users = $user->find(array('id IN ?',
    $userId));
    $id = $users->getAll('User_onesignal_id');
    $heading = array("en" => $this-
    >schedule_name);
    $subtitle = array("en" => date('d-m-Y H:i',
    strtotime($this->schedule_start_date)) . ' - '
    . date('d-m-Y H:i', strtotime($this-
    >schedule_end_date)));
    $content = array("en" => $this-
    >schedule_description
    );
}
  
```

Selanjutnya adalah *Form Login* untuk user/pengguna berisi email dan *password* yang telah di daftarkan oleh *admin*, jika *email/password* yang dimasukkan salah maka akan muncul peringatan dan tidak bisa *login*. Bisa dilihat pada gambar 20.

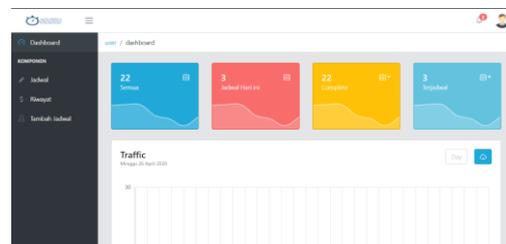


Gambar 20. Halaman Login

Login menggunakan *JWT (Java Web Token)* sama seperti *admin*, tujuannya supaya pengguna tidak harus selalu *login* ketika menutup *browser* karena kehilangan sesi koneksi dengan server.

Login juga berguna untuk mengetahui sesi *user*, hal ini untuk mempilkan data sesuai pengguna yang *login*. Seperti jadwal dan riwayat.

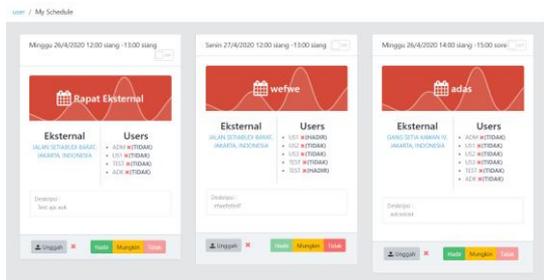
Setelah berhasil *login*, maka akan menuju *dashboard user*, tampilan *dashboard* user bisa dilihat pada gambar 21.



Gambar 21. Dashboard User/Pengguna

Dalam *dashboard* tersedia *board* untuk mengetahui detail dari keseluruhan jadwal, jadwal hari ini, yang telah selesai dan jadwal yang sudah dijadwalkan. Dan *chart* untuk mengetahui seberapa sering mereka melakukan pertemuan, hal ini berguna untuk mengetahui produktivitas karyawan.

Selanjutnya adalah tampilan pada jadwal/schedule. Tampilan ini telah dimuat berupa *card box*, yang berguna untuk mendapatkan detail setiap pertemuan. Bisa dilihat pada Gambar 22.



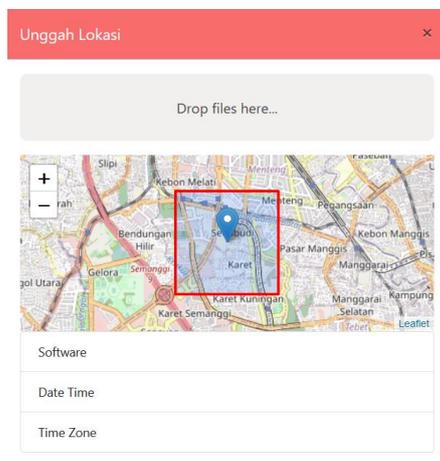
Gambar 22. Tampilan jadwal.

Jadwal dan detail pertemuan akan muncul jika pengguna ditambahkan pada saat pembuatan jadwal. Dan *user* yang ditambahkan bisa mengkonfirmasi keahadirannya pada pertemuan tersebut dengan pilihan hadir, mungkin, tidak. Sehingga jika ada yang tidak bisa mengikuti pertemuan, bisa melakukan perubahan jadwal pertemuan di kemudian hari.

Fitur lainnya adalah ketika klik/menekan lokasi pertemuan maka akan otomatis membuka Google Maps yang memberikan titik pertemuan. Ketika pengguna tidak mengetahui lokasi pertemuan, maka fitur ini sangat bermanfaat untuk memberikan arahan langsung menuju lokasi.

Sistem tracking menggunakan *Data Exif Photo* dipilih karena perusahaan tidak perlu melacak perjalanan dia dan menghindari sistem yang terus menerus melacak keberadaan orang tersebut pada saat jam di luar kerja.

Melalui perangkat *mobile*, bisa mengambil lokasi, jam dan detail perangkat yang digunakan untuk mengambil foto tersebut. Validasi menggunakan jarak *rectangle bound* menggunakan perhitungan *decimal degress* dengan toleransi kesalahan data *Latitude and Longitude* sebesar 0.3 atau sekitar 3 KM dari lokasi pertemuan yang telah ditentukan. Bisa dilihat pada Gambar 23 dan Tabel 5.



Gambar 24. Rectangle Bound

Validasi tersebut sudah otomatis dijalankan oleh sistem, maka tidak diperlukan validasi oleh admin atau orang lagi.

Tabel 5. Tabel Decimal Degrass

No	Long-Lat	Degrass
1	Bound-north	[0][0]
2	Bound-west	[0][0.03]
3	Bound-south	[0.03][0]
4	Bound-east	[0.03][0.03]

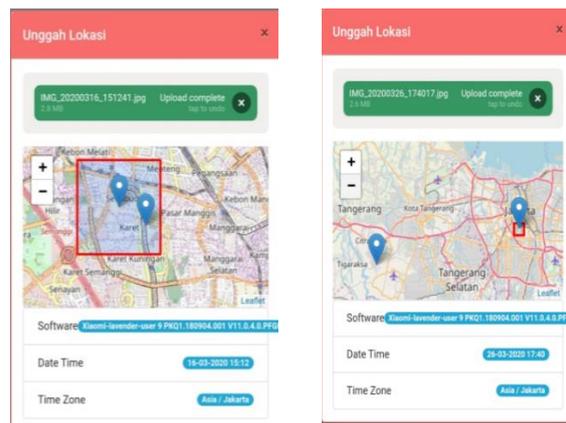
Get EXIF data Upload foto

```
request.onload = function()
{
that.softwareImage =
jsonResponse.EXIF.software
that.dateImage = jsonResponse.EXIF.datetime
that.timeZoneImage = 'Asia / Jakarta'
that.latitude = jsonResponse.EXIF.location[0]
that.longitude = jsonResponse.EXIF.location[1]
that.center = jsonResponse.EXIF.location
that.withPopup = jsonResponse.EXIF.location
that.currentCenter = jsonResponse.EXIF.
location
load(request.responseText);
}
```

Bound Rectangle Handle

```
handleUploadLocation(value, id)
{
const [lngLoc, latLoc] = (value.schedule_maps
|| '0,0').split(",");
this.latitude = 0
this.longitude = 0
this.Rectangle.bounds =
[[Number(latLoc) + 0.003,
Number(lngLoc) + 0.003],
[Number(latLoc) - 0.003,
Number(lngLoc) - 0.003]]
}
```

Hasil validasi sistem *Exif Data* gambar bisa dilihat pada saat setelah mengupload gambar tersebut ke dalam *form* yang telah di sediakan. Bisa dilihat pada gambar 24.

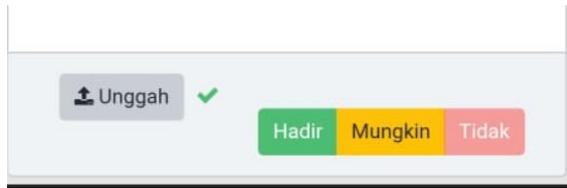


Gambar 24. Validasi berdasarkan Rectangle Bound

Gambar kiri adalah detail *long-lat* hasil *Exif Data* yang masih dalam lingkup *Rectangle Bound*, gambar kanan jika tidak sesuai dari lokasi pertemuan, hal ini menandakan detail *long-lat* salah.

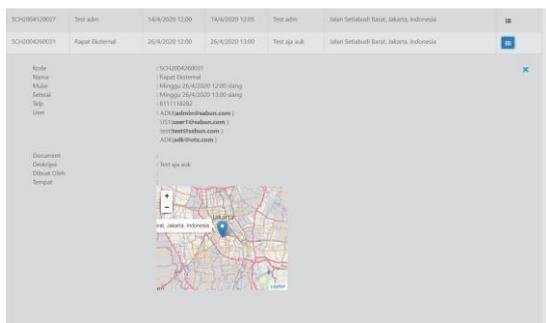
Jika validasi sistem berhasil, pemberituannya adalah *icon* pada sisi sebelah kanan tombol upload foto. Jika

benar maka icon akan berubah menjadi centang seperti gambar 25, dan jika salah maka akan tetap menjadi silang seperti Gambar 22.



Gambar 25. Gambar Yang Telah di Validasi Sistem

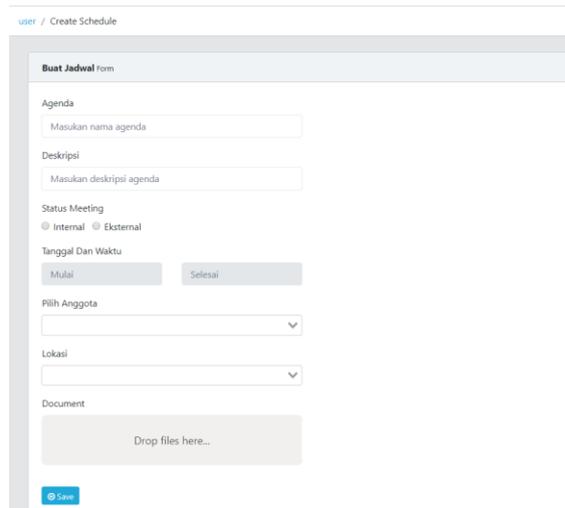
Selanjutnya adalah tampilan riwayat berdasarkan kegiatan pertemuan pengguna. Ditampilkan pada Gambar 26.



Gambar 26. Riwayat Pertemuan pengguna

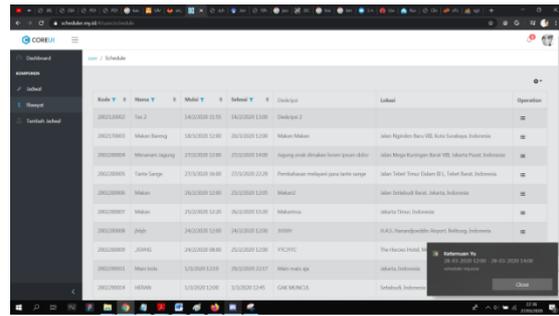
Tampilan menggunakan *table* dan *accordion* untuk menampilkan data pertemuan.

Selanjutnya adalah *Form* Pembuatan jadwal, terdiri dari Agenda, Deskripsi, Status Pertemuan (internal/eksternal), tanggal waktu mulai dan berakhirnya pertemuan, pilih anggota, pilih lokasi pertemuan dan unggah dokumen yang digunakan untuk pertemuan tersebut. Akan ditampilkan pada Gambar 27.

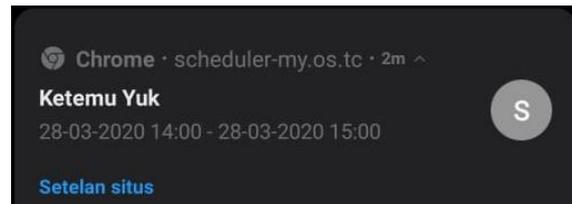


Gambar 27. Form Pembuatan Jadwal

Setelah membuat jadwal adalah mengirimkan notifikasi kepada pengguna yang ditambahkan pada saat membuat jadwal. Hasil pengiriman notifikasi bisa dilihat pada Gambar 28 dan 29.



Gambar 28. Notifikasi Pada Perangkat Desktop



Gambar 29. Notifikasi pada perangkat Mobile

Model Push Notifikasi Jadwal

```

$sendNotification = $model->sendNotification();
$this->data = ['success' => true, 'message' => 'Success', 'data' => $model->cast(), 'Notification' => $sendNotification];
}
catch(PDOException $e) {
    $err=$e->errorInfo;
    $this->data = ['success' => false, 'message' => 'Something went wrong', 'payload' => $err];
}
    
```

3.4 Hasil Uji Coba Black Box

Tahap akhir dari aplikasi ini adalah pengujian, dimana pengujian dibutuhkan untuk memvalidasi fungsional produk/aplikasi.

Tahap awal pada pengetesan pada Aplikasi Scheduler admin. Berikut hasil pengujian akan ditampilkan pada tabel 6.

Tabel 6. Hasil Pengujian Halaman Admin

No	Tampilan	Aksi	Hasil	Status
1	Login	Melakukan login dengan akun admin	Menuju Dashboard	Berhasil
2	Dashboard	Mengklik pada kalender detail pertemuan	Menuju detail jadwal pertemuan	Berhasil
3	Users	Merubah detail pengguna, menambah pengguna, menghapus pengguna	Berhasil merubah detail, menambah pengguna dan menghapus pengguna	Berhasil
4	Schedule	Merubah jadwal, menghapus jadwal, menambah jadwal,	Berhasil melakukan merubah jadwal, menghapus jadwal,	Berhasil

		melihat detail dari detail <i>Exif Data Photo</i>	menambah jadwal dan memantau hasil <i>Exif Data Photo</i>	
5	<i>Logout</i>	Mengklik tombol <i>logout</i> untuk keluar dari aplikasi.	Keluar dari aplikasi	Berhasil

Selanjutnya adalah pengujian *black box* untuk halaman pengguna. Bisa dilihat pada tabel 7.

Tabel 7. Hasil Pengujian Halaman User

No	Tampilan	Aksi	Hasil	Status
1	Login	Melakukan login dengan akun pengguna	Menuju Dashboard	Berhasil
2	Jadwal	Mengupdate kehadiran pada pertemuan, mengunggah foto dan klik pada lokasi pertemuan	Berhasil melakukan <i>update</i> kehadiran, mengunggah foto dan menuju google maps jika menekan lokasi pertemuan	Berhasil
3	Riwayat	Melihat riwayat pertemuan dan memunculkan detail pertemuan	Berhasil melihat riwayat pertemuan dan menampilkan detail pertemuan	Berhasil
4	Tambah Jadwal	Membuat jadwal pertemuan	Berhasil membuat jadwal pertemuan	Berhasil
5	Notifikasi	Mengirimkan notifikasi	Berhasil mengirim notifikasi	Berhasil

Pengujian aplikasi dengan metode *black box* dilakukan per halaman, dan juga melakukan pengujian fungsionalitas yang telah dibuat. Hasil dari pengujian dengan metode ini menunjukkan bahwa fungsi yang telah dibuat berhasil digunakan dan diimplementasikan.

Pengujian halaman admin menunjukkan hasil bahwa halaman admin telah berhasil mendapatkan informasi perincian jadwal, hasil dari *exif* foto dan mengontrol pengguna yang mengakses laman user.

Pengujian halaman user menunjukkan hasil bahwa halaman user telah memenuhi fungsionalitas utama dalam aplikasi ini seperti mengunggah gambar, mengupdate kehadiran, mendapat notifikasi pada perangkat pengguna.

4. Kesimpulan

Kesimpulan dalam perancangan dan pembuatan aplikasi ini adalah perusahaan dapat mengembangkan lagi aplikasi ini dari rancangan awal agar dapat menjadi aplikasi yang lebih *Advanced* lagi dari sisi pengguna maupun sisi admin. Aplikasi ini sudah cukup menjawab permasalahan awal yaitu pemantauan, notifikasi dan *multi device*. Dengan adanya tampilan yang *responsive* ini juga semua *device* dapat mengaksesnya dan mendapatkan layanan yang sama. *Tracking location team* menggunakan *EXIF data photo* diharapkan mengurangi kekhawatiran pengguna terhadap pelacakan pribadi perangkat pengguna. Ada kemungkinan redundansi terhadap *ID One Signal* oleh karena itu harus di kelola lebih baik lagi.

Daftar Rujukan

- [1] L. E. McAndrews and J. E. Ha-Brookshire., 2018. *Working Together is Success: Examining Cross-Functional Team Performance in the Global Apparel Supply Chain*. In: Vancouver, *Itaa Annual Conference Proceedings*. Columbia, 8 Nov 2018. United States: Columbia. ISSN 2572-679X
- [2] M. K. A. Ashari and R. A. Firmansyah., 2017. Analisis Dan Perancangan Aplikasi Penjadwalan Pertemuan Mahasiswa Dan Dosen Berbasis Android (Studi Kasus: Ruang Pengajaran STMIK AMIKOM Yogyakarta). *SEMNAS TEKNOLOGI ONLINE*, vol. 5, no. 1, pp. 4–5.
- [3] J. Y. Lee and M.-H. Tsou., 2018. *Mapping spatiotemporal tourist behaviors and hotspots through location-based photo-sharing service (Flickr) data*. Switzerland: Springer Nature. doi:10.1007/978-3-319-71470-7_16
- [4] V. Mhatre and G. Gupta., 2018. *Adaptable push-notification state*. United States: Google Patents.
- [5] H. Alatas., 2013. *Responsive Web Design*. Yogyakarta: Lokomedia
- [6] R. S. Pressman., 2010. *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill Education.
- [7] M. Coccia., 2018. *The Fishbone diagram to identify, systematize and analyze the sources of general purpos Technologies*. *J. Soc. Adm. Sci.*, vol. 4, no. 4, pp. 291–303.
- [8] Binus University, 2016. Perbedaan White Box Testing dan Black Box Testing. [Online] (Updated 06 Oct 2016) Tersedia di: <http://student-activity.binus.ac.id/himsisfo/2016/10/perbedaan-white-box-testing-dan-black-box-testing/>. [Accessed 26-Apr-2020]
- [9] R. F. Olanrewaju, T. Islam, and N. Ali., 2015. *An empirical study of the evolution of PHP MVC framework, in Advanced Computer and Communication Engineering Technology*, pp. 399–410. doi: /10.1007/978-3-319-07674-4_40
- [10] Ciputra University, 2016. Komponen MVC – Model View Controller. [Online] (Updated Dec 2016) Tersedia di: <https://informatika.uc.ac.id/2016/12/sistem-pemrograman-model-view-controller-mvc>. [Accessed 27-Mar-2020]
- [11] Json Organisation, 2015. Pengenalan JSON. [Online] (Updated Agt 2015) Tersedia di: <https://www.json.org/json-id.html>. [Accessed 27-Mar-2020]
- [12] C. Mullins., 2015. *Responsive, mobile app, mobile first: untangling the UX design web in practical experience*. In: *Proceedings of the 33rd Annual International Conference on the Design of Communication*. Dallas, Texas 14-25 July 2015. Limerick, Ireland. doi: 10.1145/2775441.2775478