

LAMPIRAN

Lampiran 1. Manual Penggunaan Aplikasi

Seperti namanya system ini digunakan untuk manajemen komplain dan penyebaran informasi tersegmentasi. Latar belakang pembuatan aplikasi ini adalah untuk menyebarkan informasi secara massif kepada mahasiswa di kampus dengan cepat dan efisien melalui aplikasi Android oleh biro/dosen/departemen tertentu di kampus. Tepat untuk menyebarkan informasi akademik kepada mahasiswa baik oleh biro tertentu, dosen, fakultas maupun jurusan.

Sistem ini terdiri dari 2 bagian, yakni halaman administrasi berbasis web untuk biro, dosen atau department (fakultas/jurusan) tertentu di kampus. Kemudian aplikasi Android yang harus diinstall oleh para mahasiswa (pengguna). Secara umum fitur dari system dapat dirangkum sebagai berikut:

Aktor Administrator (Biro, Dosen, Jurusan dan Fakultas) via halaman administrasi berbasis web:

- Menambahkan informasi yang akan disebarkan pada halaman administrasi berbasis web.
- Menyebarkan informasi tersebut kepada mahasiswa
- Melihat komplain dari mahasiswa melalui halaman web (yang diajukan oleh mereka via Aplikasi Android).
- Memberikan *feedback* terhadap komplain yang diajukan mahasiswa.
- Melakukan manajemen data pengguna yang telah mendaftar sebelumnya
- Melakukan manajemen department dan administrator yang dapat menggunakan aplikasi berbasis web.
- Melakukan *backup/restore database* system aplikasi.

Aktor Mahasiswa via Aplikasi Android:

- Mendaftar dan *login* sebagai pengguna via aplikasi Android untuk dapat menerima notifikasi dan menggunakan fitur yang ada pada aplikasi.

- Menerima notifikasi dari langsung di HP Android mereka jika ada informasi yang disebar oleh administrator (Biro, Dosen, Jurusan dan Fakultas).
- Melihat informasi yang disebar oleh administrator melalui aplikasi Android.
- Mengirimkan komplain kepada biro / departemen tertentu di kampus melalui aplikasi Android.
- Menerima notifikasi dari langsung di HP Android mereka jika ada balasan dari komplain yang mereka ajukan.
- Melihat detail komplain dan balasan dari administrator atas komplain yang diajukan.
- Memberikan balasan kembali atas *feedback* yang telah diberikan oleh administrator

Dari sisi admin terdapat halaman admin khusus untuk menyebarkan informasi. Seperti yang terlihat pada gambar berikut.

Informations [Home](#) / [Informations](#)

Information List

Information Title: Spread Type: Date Spread:

Writer: User Group Target: Status:

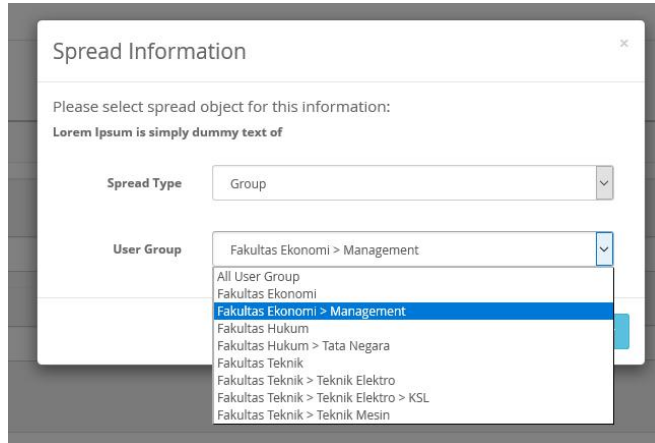
[Filter](#)

<input type="checkbox"/>	Information Title	Writer	User/User Group Target	Date Spread	Status	Action
<input type="checkbox"/>	Lorem Ipsum is simply dummy text of	Agus Aminanto	Azka Al khonsa	27/08/2017	Enabled Spread	Edit Spread
<input type="checkbox"/>	Lorem Ipsum is simply	User Satu	Fakultas Hukum > Tata Negara	17/08/2017	Enabled Spread	Edit Spread
<input type="checkbox"/>	It is a long established	Agus Aminanto	All User Group	21/08/2017	Enabled Spread	Edit Spread
<input type="checkbox"/>	UMY Membuka Pendaftaran Mahasiswa Baru Jalur CBT untuk Program Studi Tertentu	Ardi syauki	Azka Al khonsa	27/08/2017	Enabled Spread	Edit Spread

Showing 1 to 4 of 4 (1 Pages)

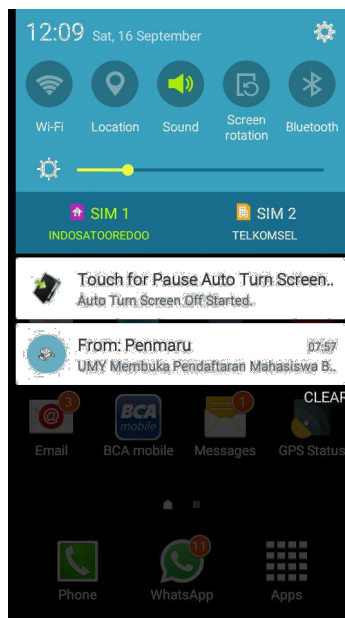
Daftar informasi pada halaman manajemen informasi

Klik pada tombol spread akan membuka popup dialog seperti Gambar dibawah ini.



Fitur penyebaran informasi pada halaman admin

Kemudian ada aplikasi Android khusus yang harus diinstall oleh mahasiswa (pengguna). Sehingga ketika admin (biro/dosen/departemen) menyebarkan informasi tertentu, mahasiswa dapat menerima notifikasi seperti pada Gambar berikut.



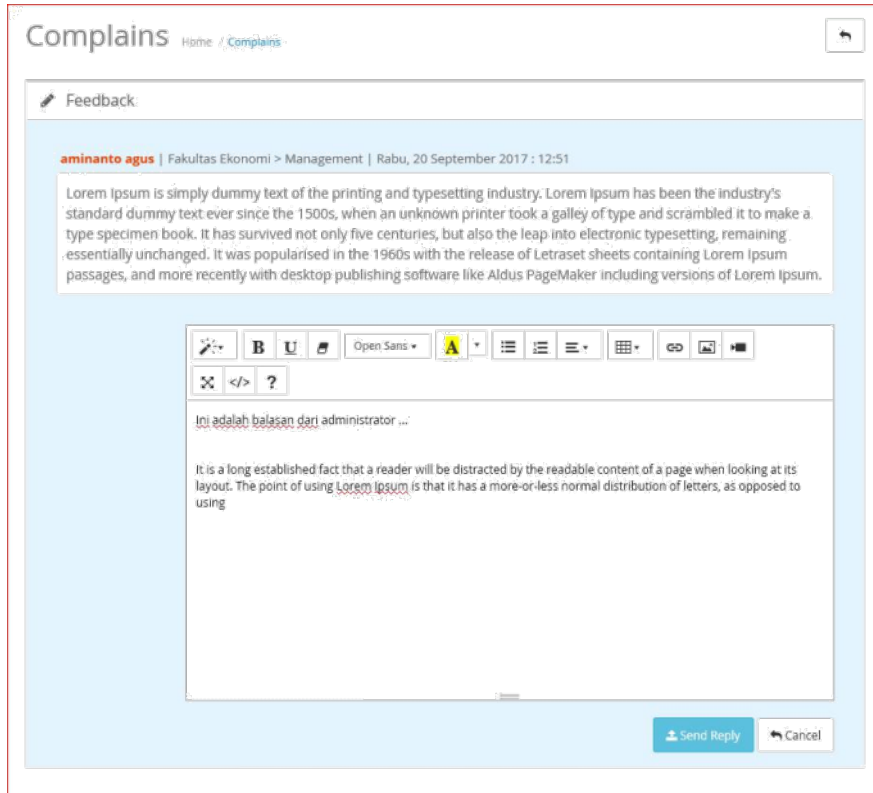
Notifikasi pada aplikasi Android saat admin menyebarkan informasi

Selain itu pada sistem ini mahasiswa juga dapat mengajukan pengaduan (complain) terkait layanan/peristiwa yang terjadi di kampus. Isi komplain ini dapat beragam, mulai dari kerusakan fasilitas kampus hingga pelanggaran tata tertib kampus oleh mahasiswa. Seorang mahasiswa dapat mengajukan komplain ini kepada biro tertentu di kampus untuk diperbaiki.

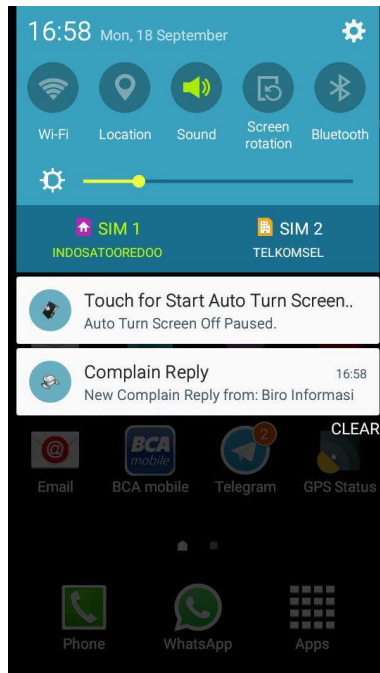
Form pengajuan komplain pada aplikasi Android

<input type="checkbox"/>	Complain Title	Reporter	Target Department	Date Spread	Status	Action
<input type="checkbox"/>	Website KRS tidak dapat diakses	aminanto agus	Biro Sistem Informasi	20/09/2017	Enabled	Edit Send Feedback

Daftar Komplain yang masuk di halaman manajemen komplain



Memberikan feedback dari komplain mahasiswa



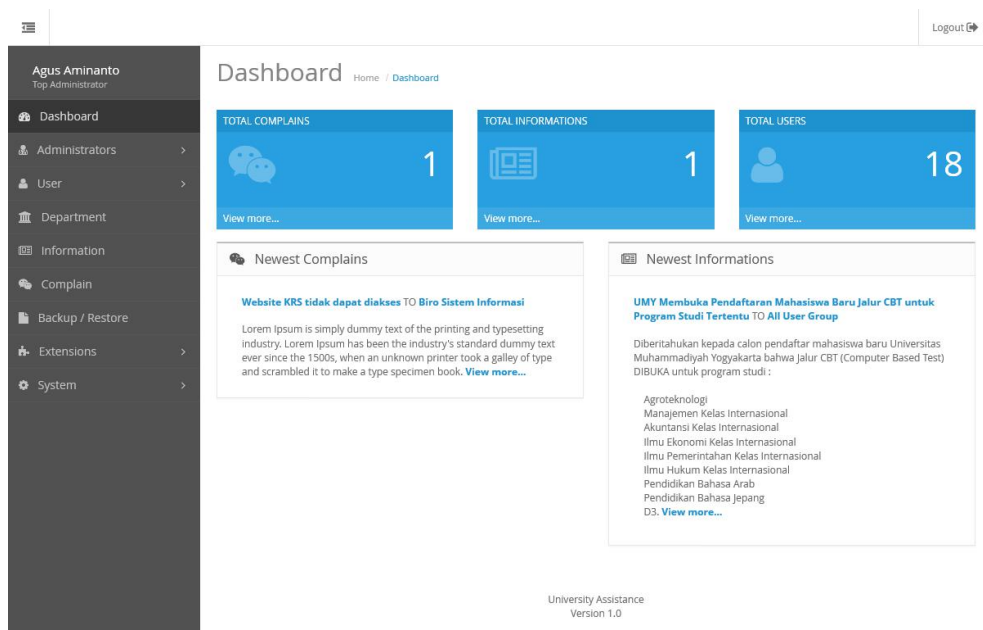
Notifikasi pada aplikasi Android saat admin memberikan feedback dari komplain

Sistem Aplikasi Web Administrasi

Aplikasi berbasis web ini digunakan untuk melakukan manajemen informasi pada aplikasi secara umum. Juga digunakan untuk menjalankan fitur utama seperti menyebarkan informasi. Dan berikut ini adalah gambaran singkat mengenai fitur di dalam aplikasi.

Halaman Dashboard

Halaman awal setelah *login* administrator, maka akan dibawa pada halaman dashboard seperti pada Gambar di bawah ini. Pada halaman ini administrator dapat melihat komplain terbaru yang masuk dan informasi terbaru yang telah ditambahkan. Juga informasi mengenai jumlah pengguna yang telah terdaftar.





Halaman dashboard system aplikasi web




Halaman Administrator

Halaman ini digunakan untuk menambahkan, mengurangi dan menyunting hak akses seorang administrator. Administrator disini dapat berupa seorang karyawan universitas yang ada pada biro, jurusan atau fakultas tertentu. Dapat pula dosen dari jurusan terkait. Halaman ini terdiri dari 2 halaman, yakni

halaman administrator dan halaman grup administrator yang masing-masing dapat dilihat pada Gambar dibawah ini.



Administrators [Home](#) / [Administrators](#)  

Administrator List






<input type="checkbox"/>	Name	Department	Status	Date Added	Action
<input type="checkbox"/>	Agus Aminanto	Biro Sistem Informasi	Enabled	13/04/2017	
<input type="checkbox"/>	Complain Admin	Biro Admisi	Enabled	30/04/2017	
<input type="checkbox"/>	Ardi syauki	Penmaru	Enabled	17/08/2017	

Showing 1 to 3 of 3 (1 Pages)

Halaman administrator

Administrator Groups [Home](#) / [Administrator Groups](#)  

Administrator Group

<input type="checkbox"/>	Administrator Group Name 	Description	Action
<input type="checkbox"/>	Complain Administrator	Complain Administrator	
<input type="checkbox"/>	Informasi Administrator	Informasi Administrator	
<input type="checkbox"/>	Penmaru	Penmaru	
<input type="checkbox"/>	Top Administrator	Top Administrator	

Showing 1 to 4 of 4 (1 Pages)

Halaman grup administrator

Halaman Pengguna (Users)

Halaman ini digunakan untuk melakukan manajemen pengguna baik menambah pengguna baru, mengedit informasi profil pengguna bahkan menghapusnya. Halaman ini terdiri dari 2 halaman, yakni halaman pengguna (*user*) dan halaman grup pengguna (*user group*).

Users [Home](#) / [Users](#)

User List

Name: User Group: Status:

NIM: E-Mail: [Filter](#)

<input type="checkbox"/>	Name	NIM	User Group	Email	Status	Date Added	Action
<input type="checkbox"/>	agus aminanto	20140120127	Fakultas Teknik	aminanto.agus@gmail.com	Enabled	13/06/2017	Edit
<input type="checkbox"/>	Rudi hartono	20140120128	Fakultas Teknik	lavabeta@gmail.com	Enabled	06/06/2017	Edit
<input type="checkbox"/>	Ardi Syauki	20140120129	Fakultas Teknik	ardi.syauki@gmail.com	Enabled	03/08/2017	Edit

Halaman manajemen pengguna

User Groups [Home](#) / [User Groups](#)

User Group

<input type="checkbox"/>	User Group Name	Description	Action
<input type="checkbox"/>	All User Group	All User Group	Edit
<input type="checkbox"/>	Fakultas Ekonomi	Fakultas Ekonomi	Edit
<input type="checkbox"/>	Fakultas Ekonomi > Management	Management	Edit
<input type="checkbox"/>	Fakultas Hukum	Fakultas Hukum	Edit
<input type="checkbox"/>	Fakultas Hukum > Kenegaraan	Kenegaraan	Edit
<input type="checkbox"/>	Fakultas Teknik	Fakultas Teknik	Edit
<input type="checkbox"/>	Fakultas Teknik > Teknik Elektro	Teknik Elektro	Edit

Halaman manajemen grup pengguna

Halaman Departemen

Halaman ini digunakan untuk manajemen departemen apa saja yang ada di institusi kampus. Departemen disini dapat berupa biro, jurusan atau fakultas. Halaman departemen ditunjukkan oleh gambar berikut.



Department			
<input type="checkbox"/>	Department Name ▼	Description	Action
<input type="checkbox"/>	Biro Admisi	Biro Admisi	
<input type="checkbox"/>	Biro Informasi	Biro Informasi	
<input type="checkbox"/>	Biro Sistem Informasi	Biro Sistem Informasi	
<input type="checkbox"/>	Penmaru	Biro Penmaru	

Showing 1 to 4 of 4 (1 Pages)

Halaman manajemen departemen

Halaman Informasi

Halaman ini merupakan salah satu fitur utama dari aplikasi untuk melakukan manajemen informasi sekaligus melakukan penyebaran informasi ke aplikasi Android pengguna (mahasiswa). Seperti terlihat pada gambar di bawah ini.

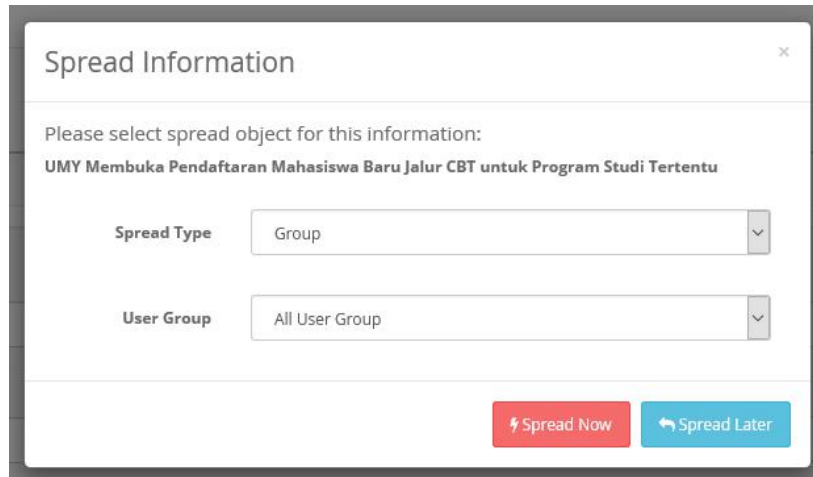


Information List						
<div style="display: flex; justify-content: space-between;"> <div> <p>Information Title</p> <input type="text" value="Information Title"/> </div> <div> <p>Spread Type</p> <input type="text" value="Group"/> </div> <div> <p>Date Spread</p> <input type="text" value="Date Spread"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div> <p>Writer</p> <input type="text" value="Writer"/> </div> <div> <p>User/Group Target</p> <input type="text" value="All User Group"/> </div> <div> <p>Status</p> <input type="text" value=""/> </div> </div> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Filter"/> </div>						
<input type="checkbox"/>	Information Title	Writer	User/User Group Target	Date Spread	Status	Action
<input type="checkbox"/>	UMY Membuka Pendaftaran Mahasiswa Baru Jalur CBT untuk Program Studi Tertentu	Ardi syauki	Azka Al khonsa	18/09/2017	Enabled Spread	<input type="button" value="Edit"/> <input type="button" value="Spread"/>

Showing 1 to 1 of 1 (1 Pages)

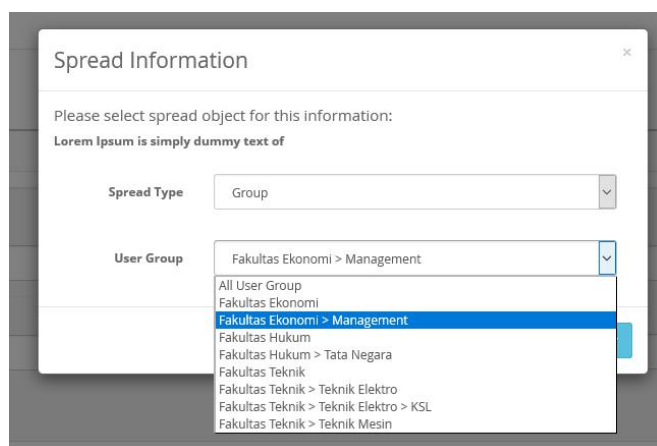
Halaman manajemen informasi

Penyebaran informasi dapat dilakukan via klik pada tombol *Spread Now*. Maka akan muncul kotak dialog seperti gambar dibawah ini.



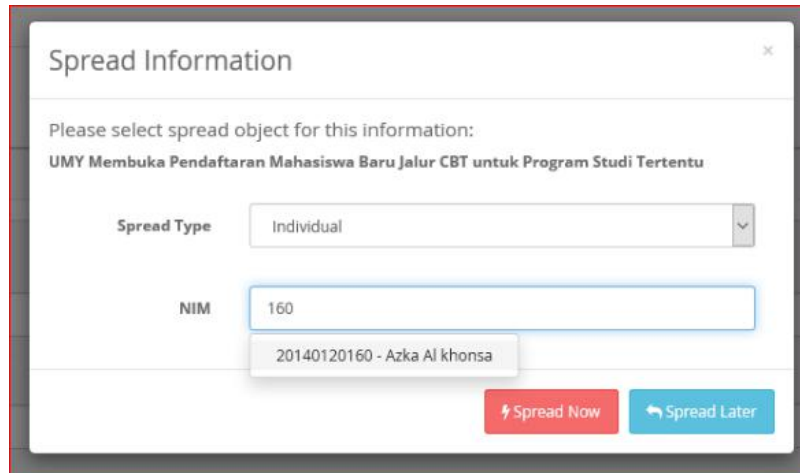
Jendela Spread Information untuk menyebarkan informasi

Selanjutnya administrator dapat memilih kelompok sebarang yang menjadi target informasi sebaran. Pengguna yang ada pada *user group* tertentu akan mendapatkan notifikasi pada HP Android mereka saat administrator menyebarkan informasi. Tombol *Spread Now* akan menyebarkan informasi ke pengguna sekaligus memberikan notifikasi pada HP Android mereka. Informasi yang telah disebar, dapat dilihat melalui aplikasi Android yang telah disediakan.



Pemilihan user group sebagai target sebaran informasi

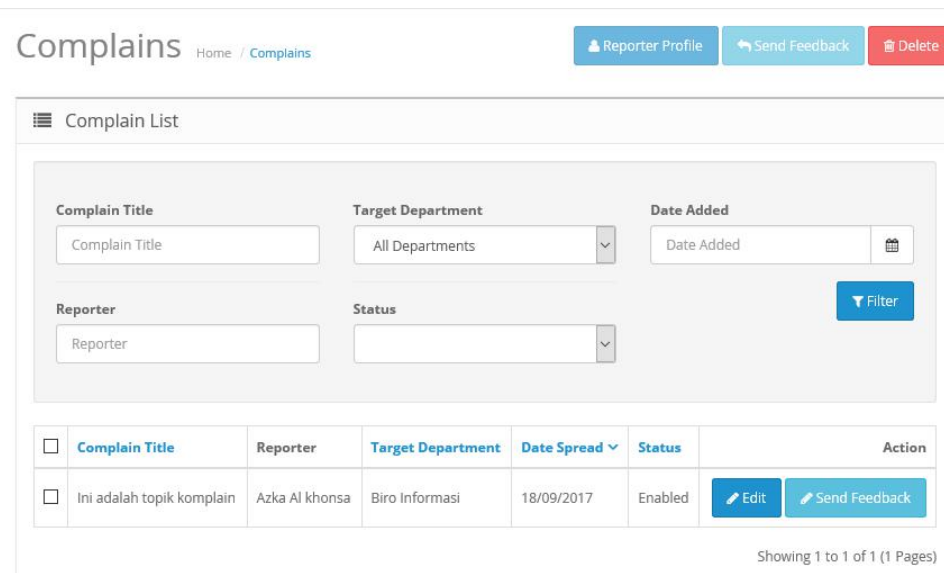
Penyebaran informasi juga dapat dilakukan kepada mahasiswa tertentu saja dengan acuan NIM seperti dapat dilihat pada gambar berikut.



Penyebaran informasi secara individual dengan acuan NIM

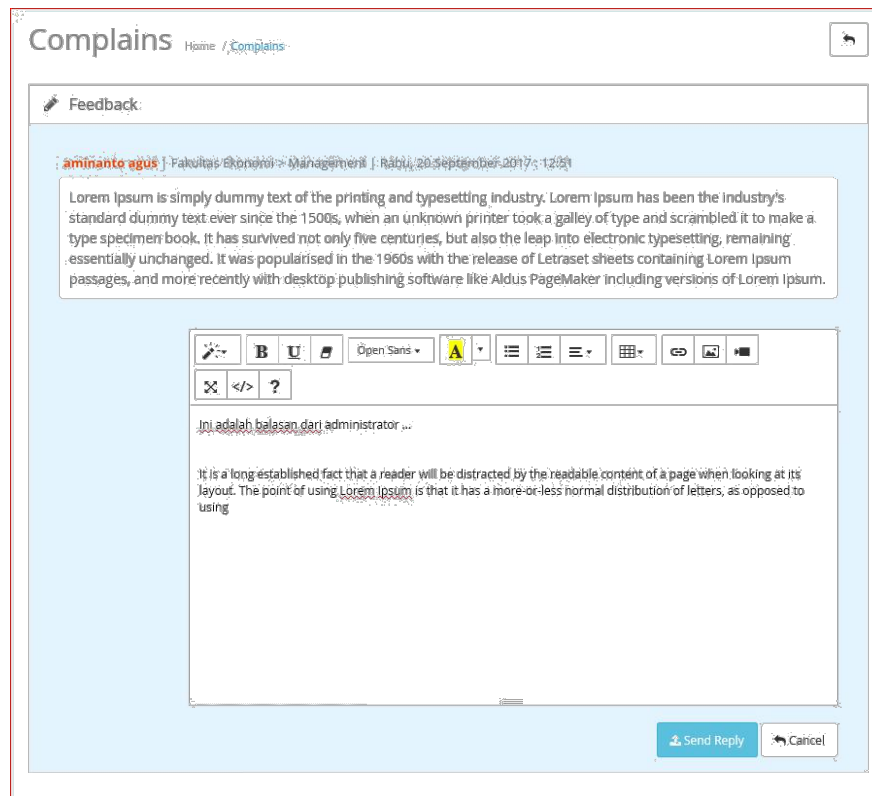
Halaman Complain

Halaman ini digunakan untuk manajemen complain yang telah diajukan oleh mahasiswa. Halaman ini ditunjukkan oleh gambar berikut.



Halaman manajemen complain

Setiap komplain yang diajukan mahasiswa via aplikasi Android akan tampil pada halaman ini. Kemudian administrator dapat meresponnya via halaman *Feedback* yang dapat diakses via klik pada tombol *Send Feedback*.

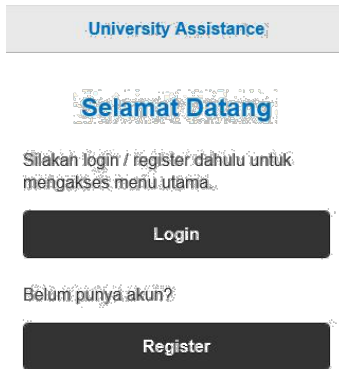


Halaman untuk memberikan feedback pada komplain yang diajukan pengguna

Setiap *feedback* yang diberikan administrator, maka pengguna akan mendapatkan notifikasi pada HP Android mereka. Selanjutnya mereka dapat melihat *feedback* tersebut pada halaman notifikasi melalui aplikasi Android yang disediakan.

Aplikasi Android Pengguna

Aplikasi ini dapat didownload gratis di: <http://s.id/umyapp>. Buka URL tersebut pada *browser* di HP Android Anda dan download file tersebut. Setelah selesai, install file *.apk* yang baru saja didownload. Kemudian buka aplikasi, maka Anda akan menemui gambar seperti berikut.



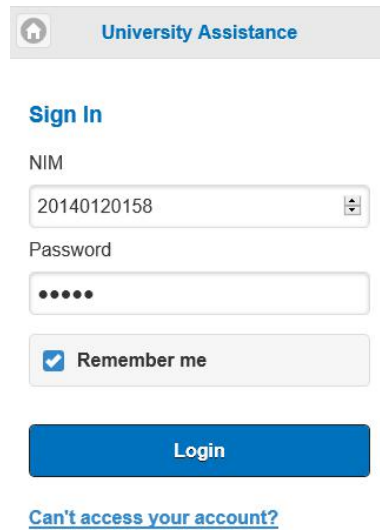
Halaman awal aplikasi

Sekarang Anda harus Register untuk dapat menggunakan fitur dari aplikasi.

A screenshot of a mobile application interface showing a registration form. At the top, there is a grey header bar with a home icon and the text "University Assistance". Below the header, the text "Register" is displayed in blue. Underneath, a message reads "Silakan isi form pendaftaran berikut dengan baik dan benar." Below this message are five input fields: "Nomer Induk Mahasiswa (NIM)", "Nama Depan", "Nama Belakang", "Email", and "No. HP". The "NIM" and "No. HP" fields have dropdown arrows on their right sides.

Halaman registrasi

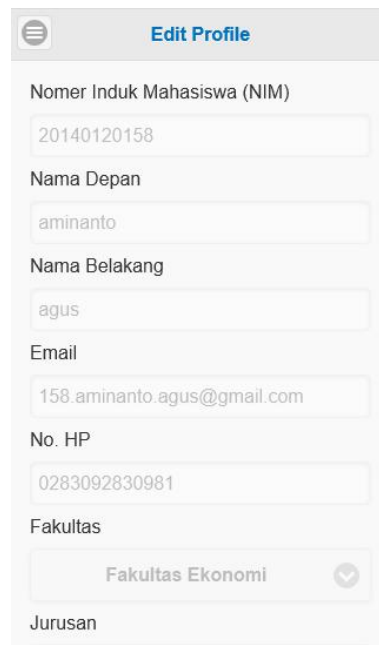
Login dengan *credential* yang telah dimasukkan sebelumnya meliputi NIM dan Password.



The screenshot shows the 'University Assistance' login interface. At the top, there is a header with a home icon and the text 'University Assistance'. Below this, the section is titled 'Sign In'. There are two input fields: 'NIM' with the value '20140120158' and 'Password' with masked characters '•••••'. A 'Remember me' checkbox is checked. A blue 'Login' button is positioned below the fields. At the bottom, there is a link that says 'Can't access your account?'.

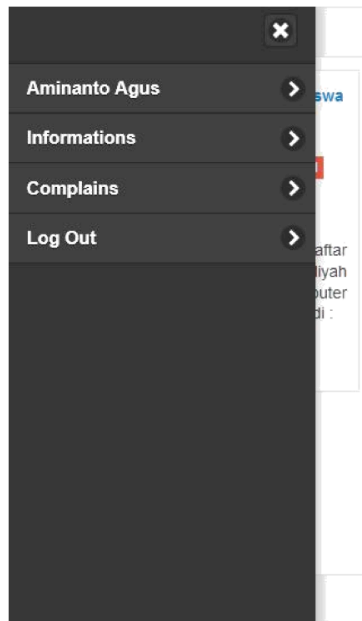
TE UMY 2017

Halaman Login



The screenshot shows the 'Edit Profile' page. It features a header with a menu icon and the text 'Edit Profile'. Below the header, there are several input fields for user information: 'Nomer Induk Mahasiswa (NIM)' with '20140120158', 'Nama Depan' with 'aminanto', 'Nama Belakang' with 'agus', 'Email' with '158.aminanto.agus@gmail.com', 'No. HP' with '0283092830981', 'Fakultas' with a dropdown menu showing 'Fakultas Ekonomi', and 'Jurusan' which is currently empty.

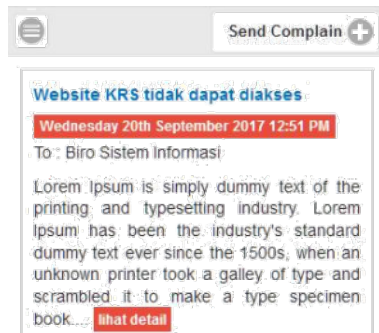
Halaman profile



Main menu yang dapat diakses via button di kiri atas

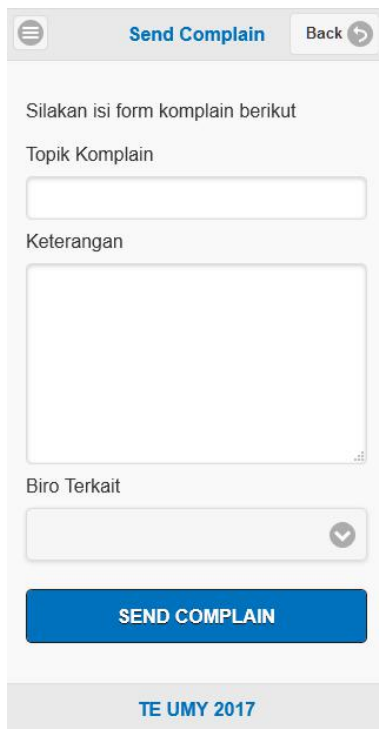


Halaman informasi



TE UMY 2017

Halaman complain



Halaman pengiriman komplain

Complain Reply Back

aminanto agus | Fakultas Ekonomi & Management | Wednesday 20th September 2017 12:51 PM

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Agus Aminanto | Biro Penmaru | Wednesday 20th September 2017 22:58 PM

remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Send a Feedback

CANCEL

SEND FEEDBACK

TE UMY 2017

Halaman complain reply untuk membalas feedback dari administrator

Lampiran 2. Create table syntax

```
1 -- Create syntax for TABLE 'ua_administrator'
2 CREATE TABLE `ua_administrator` (
3   `administrator_id` int(11) NOT NULL AUTO_INCREMENT,
4   `administrator_group_id` int(11) NOT NULL,
5   `department_id` int(11) NOT NULL,
6   `username` varchar(20) NOT NULL,
7   `password` varchar(40) NOT NULL,
8   `salt` varchar(9) NOT NULL,
9   `firstname` varchar(32) NOT NULL,
10  `lastname` varchar(32) NOT NULL,
11  `email` varchar(96) NOT NULL,
12  `image` varchar(255) NOT NULL,
13  `code` varchar(40) NOT NULL,
14  `ip` varchar(40) NOT NULL,
15  `status` tinyint(1) NOT NULL,
16  `date_added` datetime NOT NULL,
17  PRIMARY KEY (`administrator_id`)
18 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;
19 -- Create syntax for TABLE 'ua_administrator_group'
20 CREATE TABLE `ua_administrator_group` (
21   `administrator_group_id` int(11) NOT NULL
22  AUTO_INCREMENT,
23   `name` varchar(64) NOT NULL,
24   `description` teks NOT NULL,
25   `permission` teks NOT NULL,
26   PRIMARY KEY (`administrator_group_id`)
27 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;
28 -- Create syntax for TABLE 'ua_complain'
29 CREATE TABLE `ua_complain` (
30   `complain_id` int(11) NOT NULL AUTO_INCREMENT,
31   `department_id` tinyint(11) NOT NULL,
32   `title` varchar(75) NOT NULL,
33   `user_id` int(11) NOT NULL,
34   `content` teks NOT NULL,
35   `date_added` datetime NOT NULL,
36   `date_modified` datetime NOT NULL,
37   `status` tinyint(1) NOT NULL DEFAULT '1',
38   PRIMARY KEY (`complain_id`)
39 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
40
41 -- Create syntax for TABLE 'ua_department'
42 CREATE TABLE `ua_department` (
43   `department_id` int(11) NOT NULL AUTO_INCREMENT,
44   `name` varchar(45) NOT NULL,
45   `description` teks,
46   PRIMARY KEY (`department_id`)
47
```

```

48 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
49
50 -- Create syntax for TABLE 'ua_feedback'
51 CREATE TABLE `ua_feedback` (
52   `feedback_id` int(11) NOT NULL AUTO_INCREMENT,
53   `administrator_id` tinyint(1) NOT NULL,
54   `feedback` teks NOT NULL,
55   `complain_id` int(11) NOT NULL,
56   `date_added` datetime NOT NULL,
57   `date_modified` datetime NOT NULL,
58   PRIMARY KEY (`feedback_id`)
59 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
60
61 -- Create syntax for TABLE 'ua_information'
62 CREATE TABLE `ua_information` (
63   `information_id` int(11) NOT NULL AUTO_INCREMENT,
64   `title` varchar(255) NOT NULL DEFAULT "",
65   `description` teks NOT NULL,
66   `information` teks NOT NULL,
67   `status` enum('1','0') NOT NULL,
68   `administrator_id` int(11) NOT NULL,
69   `department_id` int(11) NOT NULL,
70   `writer` varchar(75) NOT NULL DEFAULT "",
71   `date_modified` datetime NOT NULL,
72   `date_added` datetime NOT NULL,
73   `date_spread` datetime NOT NULL,
74   PRIMARY KEY (`information_id`)
75 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
76
77 -- Create syntax for TABLE 'ua_information_spread'
78 CREATE TABLE `ua_information_spread` (
79   `information_spread_id` int(11) NOT NULL
80   AUTO_INCREMENT,
81   `information_id` int(11) NOT NULL,
82   `spread_type` enum('INDIVIDUAL','GROUP') NOT NULL
83   DEFAULT 'GROUP',
84   `user_group_id` int(11) NOT NULL,
85   `user_id` teks NOT NULL,
86   PRIMARY KEY (`information_spread_id`)
87 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
88
89 -- Create syntax for TABLE 'ua_token'
90 CREATE TABLE `ua_token` (
91   `token_id` int(11) NOT NULL AUTO_INCREMENT,
92   `token` teks NOT NULL,
93   `user_id` int(11) NOT NULL,
94   PRIMARY KEY (`token_id`)
95 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
96

```

```

97 CREATE TABLE `ua_user` (
98   `user_id` int(11) NOT NULL AUTO_INCREMENT,
99   `nim` varchar(11) NOT NULL,
100  `salt` varchar(9) NOT NULL DEFAULT "",
101  `password` varchar(32) NOT NULL,
102  `firstname` varchar(50) NOT NULL,
103  `lastname` varchar(50) NOT NULL DEFAULT "",
104  `no_hp` varchar(15) NOT NULL,
105  `email` varchar(45) NOT NULL,
106  `image` varchar(255) NOT NULL DEFAULT "",
107  `device_id` varchar(255) NOT NULL DEFAULT "",
108  `status` tinyint(1) NOT NULL,
109  `date_added` datetime NOT NULL,
110  PRIMARY KEY (`user_id`)
111 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
112 -- Create syntax for TABLE `ua_user_group`
113 CREATE TABLE `ua_user_group` (
114   `user_group_id` int(11) unsigned NOT NULL
115   AUTO_INCREMENT,
116   `parent_id` int(3) NOT NULL,
117   `name` varchar(45) NOT NULL,
118   `path` varchar(64) NOT NULL,
119   `description` teks NOT NULL,
120   `status` tinyint(1) NOT NULL DEFAULT '1',
121   `date_added` datetime NOT NULL,
122   `date_modified` datetime NOT NULL,
123   PRIMARY KEY (`user_group_id`)
124 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
125 -- Create syntax for TABLE `ua_user_login`
126 CREATE TABLE `ua_user_login` (
127   `user_login_id` int(11) NOT NULL AUTO_INCREMENT,
128   `nim` varchar(11) NOT NULL,
129   `ip` varchar(40) NOT NULL,
130   `total` int(4) NOT NULL,
131   `date_added` datetime NOT NULL,
132   `date_modified` datetime NOT NULL,
133   PRIMARY KEY (`user_login_id`),
134   KEY `nim` (`nim`),
135   KEY `ip` (`ip`)
136 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;
137 -- Create syntax for TABLE `ua_user_to_user_group`
138 CREATE TABLE `ua_user_to_user_group` (
139   `user_id` int(11) NOT NULL,
140   `user_group_id` int(11) NOT NULL,
141   PRIMARY KEY (`user_id`,`user_group_id`),
142   KEY `user_group_id` (`user_group_id`)
143 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;

```

Lampiran 3. Skrip untuk validasi *login* administrasi

```
1  if (($this->request->server['REQUEST_METHOD'] == 'POST') &&
2  $this->validate()) {
3      $this->session->data['token'] = token(32);
4
5      if (isset($this->request->post['redirect']) &&
6  (strpos($this->request->post['redirect'], HTTP_SERVER) === 0
7  || strpos($this->request->post['redirect'], HTTPS_SERVER) ===
8  0)) {
9          $this->response->redirect($this->request-
10 >post['redirect'] . '&token=' . $this->session-
11 >data['token']);
12     } else {
13         $this->response->redirect($this->url-
14 >link('common/dashboard', 'token=' . $this->session-
15 >data['token'], true));
16     }
17 }
```

Lampiran 4. Fungsi validasi untuk mengecek *username* dan *password*

```
1 protected function validate() {
2     if (!isset($this->request->post['username']) ||
3         !isset($this->request->post['password']) || !$this->user-
4         >login($this->request->post['username'],
5             html_entity_decode($this->request->post['password'],
6                 ENT_QUOTES, 'UTF-8')) {
7         $this->error['warning'] = $this->language-
8         >get('error_login');
9     }
10    return !$this->error;
11 }
```

Lampiran 5. SQL statement untuk menyeleksi data sesuai *username* dan *password*

```
1 public function login($username, $password) {
2     $administrator_query = $this->db->query("SELECT * FROM
3     ua_administrator WHERE username = '" . $this->db-
4     >escape($username) . "' AND (password = SHA1(CONCAT(salt,
5     SHA1(CONCAT(salt, SHA1('" . $this->db-
6     >escape(htmlspecialchars($password, ENT_QUOTES)) . "')))) OR
7     password = '" . $this->db->escape(md5($password)) . "' ) AND
8     status = '1'");
9     ...
```

Lampiran 6. Prosedur dashboard untuk menampilkan informasi terbaru

```
1  ...
2  public function dashboard() {
3  $this->load->language('extension/dashboard/information');
4
5  $data['heading_title'] = $this->language-
6  >get('heading_title');
7
8  $data['teks_view'] = $this->language->get('teks_view');
9
10 $data['token'] = $this->session->data['token'];
11
12 // Total informations
13 $this->load->model('system/information');
14
15 $order_total = $this->model_system_information-
16 >getTotalInformations();
17
18 $data['total'] = $order_total;
19
20 $data['information'] = $this->url->link('system/information',
21 'token=' . $this->session->data['token'], true);
22 return $this->load-
23 >view('extension/dashboard/information_info', $data);
24 }
```


Lampiran 7. Seleksi data dari tabel grup administrator

```
1 public function getAdministratorGroups($data = array()) {
2     $sql = "SELECT * FROM ua_administrator_group";
3
4     $sql .= " ORDER BY name";
5
6     if (isset($data['order']) && ($data['order'] == 'DESC')) {
7         $sql .= " DESC";
8     } else {
9         $sql .= " ASC";
10    }
11
12    if (isset($data['start']) || isset($data['limit'])) {
13        if ($data['start'] < 0) {
14            $data['start'] = 0;
15        }
16
17        if ($data['limit'] < 1) {
18            $data['limit'] = 20;
19        }
20
21        $sql .= " LIMIT " . (int)$data['start'] . ", " .
22        (int)$data['limit'];
23    }
24
25    $query = $this->db->query($sql);
26
27    return $query->rows;
28 }
```

Lampiran 8. *SQL statement* untuk penambahan grup administrator

```
1 public function addAdministratorGroup($data) {
2     $this->db->query("INSERT INTO
3     ua_administrator_group SET name = '" . $this->db-
4     >escape($data['name']) . "', description = '" . $this->db-
5     >escape($data['description']) . "', permission = '" .
6     (isset($data['permission']) ? $this->db-
7     >escape(json_encode($data['permission'])) : '') . "'");
8
9     return $this->db->getLastId();
10 }
```

Lampiran 9. *SQL statement* untuk seleksi data pada manajemen administrator

```
1 public function getAdministrators($data = array()) {
2     $sql = "SELECT a.username, CONCAT(a.firstname, '
3     ',a.lastname) as name, a.administrator_id, a.status,
4     a.date_added, d.name as department FROM `ua_administrator` a
5     LEFT JOIN `ua_department` d ON (a.department_id =
6     d.department_id) ";
7
8     $sort_data = array(
9         'username',
10        'department_id',
11        'date_added',
12        'status'
13    );
14
15    $implode = array();
16
17    if (!empty($data['filter_name'])) {
18        $implode[] = "CONCAT(a.firstname, ' ', a.lastname)
19    LIKE '%" . $this->db->escape($data['filter_name']) . "%'";
20    }
21
22    if (!empty($data['filter_department_id'])) {
23        $implode[] = "d.deparment_id = '" .
24    (int)$data['filter_department_id'] . "'";
25    }
26
27    if (isset($data['filter_status']) &&
28    !is_null($data['filter_status'])) {
29        $implode[] = "u.status = '" .
30    (int)$data['filter_status'] . "'";
31    }
32
33    if ($implode) {
34        $sql .= " AND " . implode(" AND ", $implode);
35    }
36
37    if (isset($data['sort']) && in_array($data['sort'],
38    $sort_data)) {
39        $sql .= " ORDER BY " . $data['sort'];
40    } else {
41        $sql .= " ORDER BY a.date_added";
42    }
43
44    if (isset($data['order']) && ($data['order'] == 'DESC')) {
45        $sql .= " DESC";
46    } else {
47        $sql .= " ASC";
48    }
49
50    if (isset($data['start']) || isset($data['limit'])) {
51        if ($data['start'] < 0) {
52            $data['start'] = 0;
53        }
54
55        if ($data['limit'] < 1) {
56            $data['limit'] = 20;
57        }
58    }
```

```
59         $sql .= " LIMIT " . (int)$data['start'] . "," .  
60 (int)$data['limit'];  
61     }  
62  
63     $query = $this->db->query($sql);  
64  
65     return $query->rows;  
66 }
```

Lampiran 10. *SQL statement* untuk penambahan *user group*

```
1 public function addUserGroup($data) {
2     $this->db->query("INSERT INTO ua_user_group SET name = '" .
3     $this->db->escape($data['name']) . "', description = '" .
4     $this->db->escape($data['description']) . "', parent_id =
5     '".(int) $data['parent_id']."'");
6
7     $user_group_id = $this->db->getLastId();
8
9     $this->db->query("UPDATE ua_user_group SET path = '". $this->
10    >getPath($user_group_id)." ' WHERE user_group_id =
11    '".(int)$user_group_id."' ");
12
13         return $user_group_id;
15     }
```

Lampiran 11. Prosedur yang digunakan untuk mendapatkan *path* dari sebuah *user group*

```
1 public function getPath($user_group_id) {
2     $query = $this->db->query("SELECT ug.name, parent_id FROM
3     ua_user_group as ug WHERE ug.user_group_id='".(int)
4     $user_group_id."' ORDER BY ug.name ASC");
5
6     if ($query->row['parent_id']) {
7         return $this->getPath($query->row['parent_id']) .
8     $this->language->get('teks_separator') . $query->row['name'];
9     } else {
10        return $query->row['name'];
11    }
12 }
```

Lampiran 12. AJAX *request* untuk mendapatkan data *user group* dan membuat pilihan pada kolom *user group*

```
1  $('input[name=\'input_user_group\"]').autocomplete({
2  'source': function(request, response) {
3  $.ajax({
4  url: 'index.php?route=user/user_group/autocomplete&token=<?php
5  echo $token; ?>&filter_name=' + encodeURIComponent(request),
6  dataType: 'json',
7  success: function(json) {
8  response($.map(json, function(item) {
9  return {
10     label: item['path'],
11     value: item['user_group_id']
12     }
13     }
14     }
15     });
16     },
17     'select': function(item) {
18     $('input[name=\'input_user_group\"]').val('');
19
20     $('#user-group' + item['value']).remove();
21
22     $('#user-group').append('<div id="user-group' + item['value']
23     + '"><i class="fa fa-minus-circle"></i> ' + item['label'] +
24     '<input type="hidden" name="user_group[]" value="' +
25     item['value'] + '" /></div>');
26     }
27     });
```

Lampiran 13. *SQL statement* untuk seleksi data pada tabel informasi

```
1 public function getInformations($data = array()) {
2     $sql = "SELECT ui.*, ug.path as user_group, ug.user_group_id,
3     uis.spread_type, uis.user_id FROM `ua_information` as ui,
4     `ua_user_group` as ug, ua_information_spread as uis WHERE
5     ug.user_group_id = uis.user_group_id AND
6     ui.information_id=uis.information_id ";
7
8     $sort_data = array(
9         'date_spread',
10        'user_group_id'
11    );
12
13    $implode = array();
14
15    if (!empty($data['filter_date_spread'])) {
16        $implode[] = "DATE(ui.date_spread) = DATE('" . $this->
17        >db->escape($data['filter_date_spread']) . "')";
18    }
19
20    if (!empty($data['filter_user_group_id'])) {
21        $implode[] = "ug.user_group_id = '" .
22        (int)$data['filter_user_group_id'] . "'";
23    }
24
25    if (!empty($data['filter_spread_type'])) {
26        $implode[] = "uis.spread_type = UPPER('"
27        .$data['filter_spread_type'] . "')";
28    }
29
30    if (!empty($data['filter_user_id'])) {
31        $implode[] = "uis.user_id = '" . $this->db->
32        >escape(json_encode(array($data['filter_user_id']),true)).
33        "'";
34    }
35
36    if (!empty($data['filter_title'])) {
37        $implode[] = "ui.title = '" . $data['filter_title'] .
38        "'";
39    }
40
41    if (!empty($data['filter_administrator_id']) &&
42    !is_null($data['filter_administrator_id'])) {
43        $implode[] = "ui.administrator_id = '" .
44        (int)$data['filter_administrator_id'] . "'";
45    }
46
47    if (isset($data['filter_status']) &&
48    !is_null($data['filter_status'])) {
49        $implode[] = "ui.status = '" .
50        (int)$data['filter_status'] . "'";
51    }
52
53    if ($implode) {
54        $sql .= " AND " . implode(" AND ", $implode);
55    }
56
57    if (isset($data['sort']) && in_array($data['sort'],
58    $sort_data)) {
59        $sql .= " ORDER BY " . $data['sort'];
60    } else {
```



```
61     $sql .= " ORDER BY date_added";
62     }
63
64     if (isset($data['order']) && ($data['order'] == 'DESC'))
65     {
66         $sql .= " DESC";
67     } else {
68         $sql .= " ASC";
69     }
70
71     if (isset($data['start']) || isset($data['limit'])) {
72         if ($data['start'] < 0) {
73             $data['start'] = 0;
74         }
75
76         if ($data['limit'] < 1) {
77             $data['limit'] = 20;
78         }
79
80         $sql .= " LIMIT " . (int)$data['start'] . "," .
81 (int)$data['limit'];
82     }
83
84     $query = $this->db->query($sql);
85
86     return $query->rows;
87 }
```

Lampiran 14. Kode JQUERY untuk *request* prosedur via AJAX

```
1 $('#button-spread-now').on('click', function() {
2   $.ajax({
3     url:
4     'index.php?route=system/information/spread&token=<?php echo
5     $token; ?>&information_id=' + $('#button-spread-
6     now').attr("information_id"),
7     type: 'post',
8     data: $('.modal-body input[type=\'teks\'], .modal-body
9     select, .modal-body input[type=\'hidden\']'),
10    dataType: 'json',
11    beforeSend: function() {
12      $('#button-spread-now').button('loading');
13    },
14    complete: function() {
15      $('#button-spread-now').button('reset');
16    },
17    success: function(json) {
18      $('#modal-spread').modal('hide');
19      $('.alert').remove();
20
21      if (json['success']) {
22        $('#content > .container-fluid').prepend('<div
23        class="alert alert-success"><i class="fa fa-check-circle"></i>
24        ' + json['success'] + ' <button type="button" class="close"
25        data-dismiss="alert">&times;</button></div>');
26      }
27
28      if(json['push_result']) {
29        console.log(json['push_result']);
30      }
31
32      $("html, body").animate({ scrollTop: 0 }, "slow");
33
34      setTimeout(function() {
35        location.reload();
36      }, 3000);
37    }
38  });
39 });
```

Lampiran 15. Prosedur *spread()* yang menangani penyebaran informasi ke pengguna

```
1 public function spread() {
2     $json = array();
3
4     $this->load->language('system/information');
5
6     $this->load->model('system/information');
7
8     if($this->request->get['information_id']) {
9         $information_id = $this->request->get['information_id'];
10    } else {
11        $information_id = 0;
12    }
13
14    $json['data'] = $this->request->post;
15
16    // spread out
17    // log spread user ids
18    // Firebase push notification
19
20    $spread_info = $this->model_system_information-
21    >spreadInformation($information_id,$this->request->post);
22
23    if($spread_info) {
24        if($this->request->post['spread_type'] == "individual") {
25            $json['success'] = sprintf($this->language-
26            >get('success_spread_information2'),
27            $spread_info['total'],$spread_info['target']);
28            $json['push_result'] = $spread_info['push_result'];
29        } else {
30            $json['success'] = sprintf($this->language-
31            >get('success_spread_information'),
32            $spread_info['total'],$spread_info['target']);
33        }
34    }
35
36    } else {
37        $json['error'] = $this->language-
38        >get('error_spread_information');
39    }
40
41    $this->response->addHeader('Content-Type: application/json');
42    $this->response->setOutput(json_encode($json));
43 }
```

Lampiran 16. Prosedur *send()* yang digunakan untuk menangani fungsi notifikasi ke pengguna

```
1 public function send($information_id,$user_ids) {
2     $tokens = array();
3     $send = array();
4
5     $query = $this->db->query("SELECT ui.title, d.name as
6     department FROM ua_information ui, ua_department d WHERE
7     ui.department_id=d.department_id AND
8     ui.information_id='".(int)$information_id.'");
9
10    if($query->num_rows) {
11        $title = "From: ".$query->row['department'];
12        $message = $query->row['title'];
13    }
14
15    if(count($user_ids) > 1) {
16        $query = $this->db->query("SELECT token FROM ua_token WHERE
17        user_id IN (".implode(',',$user_ids).") ");
18    } else {
19        $query = $this->db->query("SELECT token FROM ua_token WHERE
20        user_id = '".(int)$user_ids[0].'");
21    }
22
23    $results = $query->rows;
24
25    if($results) {
26        foreach($results as $result) {
27            $this->sendNotification($result['token'],$title,$message);
28        }
29    }
30 }
```

Lampiran 17. Request notifikasi pada prosedur *sendNotification()*

```
1 public function sendNotification($token,$title,$message) {
2     ignore_user_abort();
3     ob_start();
4
5     $url = 'https://fcm.googleapis.com/fcm/send';
6
7     $fields = array(
8         'to' => $token,
9         'notification' => array(
10            'body' => $message,
11            'title' => $title,
12            "icon" => "myicon",
13            "sound" => "default");
14
15     $headers = array(
16         'Authorization:key='.GOOGLE_API_KEY,
17         'Content-Type: application/json'
18     );
19
20     $ch = curl_init();
21     curl_setopt($ch, CURLOPT_URL, $url);
22     curl_setopt($ch, CURLOPT_POST, true);
23     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
24     curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
25     curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
26     curl_setopt($ch, CURLOPT_POSTFIELDS,
27         json_encode($fields));
28
29     $result = curl_exec($ch);
30     if($result === false)
31         die('Curl failed ' . curl_error());
32     curl_close($ch);
33     return $result;
34 }
```

Lampiran 18. Seleksi data untuk ditampilkan pada halaman *complains*

```
1 public function getComplains($data = array()) {
2
3     $sql = "SELECT c.*, CONCAT(u.firstname, ' ', u.lastname) as
4     reporter, u.user_id, d.name as department FROM ua_complain as
5     c, ua_user as u, ua_department as d WHERE u.user_id =
6     c.user_id AND c.department_id = d.department_id ";
7
8     ...
```

Lampiran 19. Berkas *settings.js* yang digunakan untuk menyimpan variabel objek *request* aplikasi web

```
1 var Ua = Ua || {};  
2 Ua.Settings = Ua.Settings || {};  
3 Ua.Settings.mainUrl = "https://skripsi.agus-  
4 aminanto.id/index.php?route=";  
5 Ua.Settings.registerUrl = Ua.Settings.mainUrl +  
6 "Android/app/adduser";  
7 Ua.Settings.loginUrl = Ua.Settings.mainUrl +  
8 "Android/app/login";  
9 Ua.Settings.checkEmailUrl = Ua.Settings.mainUrl +  
10 "Android/app/checkemail";  
11 Ua.Settings.checkNimUrl = Ua.Settings.mainUrl +  
12 "Android/app/checknim";  
13 Ua.Settings.userGroupUrl = Ua.Settings.mainUrl +  
14 "Android/app/usergroup";  
15 Ua.Settings.getUserProfile = Ua.Settings.mainUrl +  
16 "Android/app/getUserProfile";  
17 Ua.Settings.getInformations = Ua.Settings.mainUrl +  
18 "Android/app/getInformations";  
19 Ua.Settings.getComplains = Ua.Settings.mainUrl +  
20 "Android/app/getComplains";  
21 Ua.Settings.getFeedbacks = Ua.Settings.mainUrl +  
22 "Android/app/getFeedbacks";  
23 Ua.Settings.saveToken = Ua.Settings.mainUrl +  
24 "Android/app/saveToken";  
25 Ua.Settings.departments = Ua.Settings.mainUrl +  
26 "Android/app/departments";  
27 Ua.Settings.addComplain = Ua.Settings.mainUrl +  
28 "Android/app/addComplain";  
29 Ua.Settings.addFeedback = Ua.Settings.mainUrl +  
30 "Android/app/addFeedback";
```

Lampiran 20. *Linking* berkas *css*, *javascript* dan inisialisasi *framework PhoneGap*

```
1   . . .
2   . . .
3   <link rel="stylesheet" href="css/jquery.mobile-1.4.5.min.css"
4   />
5   <script src="js/jquery-1.11.1.min.js"></script>
6   <script src="js/jquery.mobile-1.4.5.min.js"></script>
7   <script src="js/settings.js"></script>
8   <link rel="stylesheet" href="css/app.css" />
9
10  . . .
11  . . .
12
13  <script type="teks/javascript" src="cordova.js"></script>
14  <script type="teks/javascript" src="js/index.js"></script>
15  <script type="teks/javascript">
16  app.initialize();
17  </script>
```


Lampiran 21. Skrip yang menangani penyimpanan data registrasi dan registrasi *Firestore* token

```
1 $('#btn-register').on('click', function() {
2 // Setup form validation on the #register-form element
3 if(form.valid()) {
4 $.ajax({
5 url: Ua.Settings.registerUrl,
6 dataType: 'json',
7 type: 'post',
8 data: $('#form-register input[type=\'password\'], #form-
9 register input[type=\'number\'], #form-register
10 input[type=\'teks\'], #form-register input[type=\'hidden\'],
11 #form-register teksarea, #form-register select'),
12 success: function(json) {
13
14 if(json['success']) {
15 // get Firestore token and save it
16 if(json['user_id']) {
17 app.getToken(json['user_id']);
18 }
19
20 $('#p.success-register').html(json['success']);
21 $('#dlg-register-sent').popup("open");
22
23
24 } else if(json['error']) {
25 $('#p.error-register').html(json['success']);
26 $('#dlg-register-not-sent').popup("open");
27 }
28 . . .
29 . . .
```

Lampiran 22. Fungsi *getToken()* dan *saveToken()* untuk mendapatkan dan menyimpan token pada *database* aplikasi web

```
1 getToken: function(user_id) {
2   window.FirebasePlugin.getToken(function(token) {
3     console.log('FIREBASE TOKEN: ' + token);
4     // var token = 'token' + user_id;
5     if(token) {
6       app.saveToken(token, user_id);
7     } else {
8       . . .
9       . . .
10
11    saveToken: function(token, user_id) {
12      $.ajax({
13        url: Ua.Settings.saveToken,
14        dataType: 'json',
15        type: 'post',
16        data: 'token= ' + token + '&user_id=' + user_id,
17        success: function(json) {
18          if(json['success']) {
19            console.log('token saved');
20          } else {
21            console.log('token saving failure');
22          }
23        }
24      });
25    }
26  });
27 }
```

Lampiran 23. Permintaan data profil pengguna untuk ditampilkan pada halaman profil

```
1 . . .
2 . . .
3
4 $.ajax({
5   url: decodeURI(Ua.Settings.getUserProfile + '&user_id=' +
6     user_id),
7   dataType: 'json',
8   success: function(json) {
9     if(json['error']) {
10      $("#dlg-invalid-credentials").popup("open");
11      $('#invalid-credentials').on('click', function() {
12        window.location = 'index.html';
13      });
14    } else {
15      if(json['user_info']) {
16        $('#firstname').val(json['user_info']['firstname']);
17        $('#lastname').val(json['user_info']['lastname']);
18        $('#nim').val(json['user_info']['nim']);
19        $('#email').val(json['user_info']['email']);
20        $('#no_hp').val(json['user_info']['no_hp']);
21      }
22    }
23  }
24 });
```

Lampiran 24. Permintaan data informasi ke aplikasi web yang ditampilkan pada halaman informasi

```
1  ...
2  ...
3  var user_id =
4  JSON.parse(localStorage.getItem('user')).user_id;
5
6  $.ajax({
7  url: decodeURI(Ua.Settings.getInformations + '&user_id=' +
8  user_id),
9  dataType: 'json',
10 success: function(json) {
11 var html = '';
12 if(json['informations']) {
13 $.each(json['informations'], function(k, val) {
14 html += '<div class="information-item"><div class="title"><a
15 data-ajax="false"
16 ...
17 ...
18 }
19
20 $('#informations').html(html);
21 }
22 });
23 ...
24 ...
```

Lampiran 25. Konfigurasi *config.xml* pada *PhoneGap*

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <widget id="com.simplepush.app" version="1.0.0"
3 xmlns="http://www.w3.org/ns/widgets"
4 xmlns:gap="http://PhoneGap.com/ns/1.0">
5 <name>Simple Push</name>
6 <description>
7 Hello World sample application that responds to the
8 deviceready event.
9 </description>
10
11 <author email="support@PhoneGap.com"
12 href="http://PhoneGap.com">
13 PhoneGap Team
14 </author>
15
16 <content src="index.html" />
17 <preference name="DisallowOverscroll" value="true" />
18 <preference name="Android-minSdkVersion" value="14" />
19 <plugin name="cordova-plugin-battery-status" source="npm"
20 spec="~1.1.1" />
21 <plugin name="cordova-plugin-camera" source="npm"
22 spec="~2.1.1" />
23 <plugin name="cordova-plugin-media-capture" source="npm"
24 spec="~1.2.0" />
25 <plugin name="cordova-plugin-device" source="npm"
26 spec="~1.1.1" />
27 <plugin name="cordova-plugin-device-motion" source="npm"
28 spec="~1.2.0" />
29 <plugin name="cordova-plugin-device-orientation"
30 source="npm" spec="~1.0.2" />
31 <plugin name="cordova-plugin-dialogs" source="npm"
32 spec="~1.2.0" />
33 <plugin name="cordova-plugin-globalization" source="npm"
34 spec="~1.0.3" />
35 <plugin name="cordova-plugin-inappbrowser" source="npm"
36 spec="~1.3.0" />
37 <plugin name="cordova-plugin-media" source="npm"
38 spec="~2.2.0" />
39 <plugin name="cordova-plugin-network-information"
40 source="npm" spec="~1.2.0" />
41 <plugin name="cordova-plugin-statusbar" source="npm"
42 spec="~2.1.2" />
43 <plugin name="cordova-plugin-vibration" source="npm"
44 spec="~2.1.0" />
45 <plugin name="cordova-plugin-whitelist" source="npm"
46 spec="~1.2.1" />
47 <platform name="Android">
48 <icon density="ldpi" src="www/res/icon/Android/drawable-
49 ldpi-icon.png" />
50 <icon density="mdpi" src="www/res/icon/Android/drawable-
51 mdpi-icon.png" />
52 <icon density="hdpi" src="www/res/icon/Android/drawable-
53 hdpi-icon.png" />
54
```

```

55 <icon density="xhdpi" src="www/res/icon/Android/drawable-
56 xhdpi-icon.png" />
57 <icon density="xxhdpi" src="www/res/icon/Android/drawable-
58 xxhdpi-icon.png" />
59 <icon density="xxxhdpi" src="www/res/icon/Android/drawable-
60 xxxhdpi-icon.png" />
61 <splash density="land-ldpi"
62 src="www/res/screen/Android/drawable-land-ldpi-screen.png"
63 />
64 <splash density="land-mdpi"
65 src="www/res/screen/Android/drawable-land-mdpi-screen.png"
66 />
67 <splash density="land-hdpi"
68 src="www/res/screen/Android/drawable-land-hdpi-screen.png"
69 />
70 <splash density="land-xhdpi"
71 src="www/res/screen/Android/drawable-land-xhdpi-screen.png"
72 />
73 <splash density="land-xxhdpi"
74 src="www/res/screen/Android/drawable-land-xxhdpi-
75 screen.png" />
76 <splash density="land-xxxhdpi"
77 src="www/res/screen/Android/drawable-land-xxxhdpi-
78 screen.png" />
79 <splash density="port-ldpi"
80 src="www/res/screen/Android/drawable-port-ldpi-screen.png"
81 />
82 <splash density="port-mdpi"
83 src="www/res/screen/Android/drawable-port-mdpi-screen.png"
84 />
85 <splash density="port-hdpi"
86 src="www/res/screen/Android/drawable-port-hdpi-screen.png"
87 />
88 <splash density="port-xhdpi"
89 src="www/res/screen/Android/drawable-port-xhdpi-screen.png"
90 />
91 <splash density="port-xxhdpi"
92 src="www/res/screen/Android/drawable-port-xxhdpi-
93 screen.png" />
94 <splash density="port-xxxhdpi"
95 src="www/res/screen/Android/drawable-port-xxxhdpi-
96 screen.png" />
97 </platform>
98
99 <access origin="*" />
100 <allow-intent href="http://*/*" />
101 <allow-intent href="https://*/*" />
102 <allow-intent href="tel:*" />
103 <allow-intent href="sms:*" />
104 <allow-intent href="mailto:*" />
105 <allow-intent href="geo:*" />
106 <platform name="Android">
107 <allow-intent href="market:*" />
108 </platform>
109 <platform name="ios">
110 <allow-intent href="itms:*" />

```

```
111 <allow-intent href="itms-apps:*" />
112 </platform>
113 <plugin name="cordova-plugin-Firebase" spec="~0.1.23" />
114 <engine name="Android" spec="~6.1.2" />
115 </widget>
```