

# SKRIPSI

## KOLEKTOR PENGUMUMAN INFORMATIKA



Ellena Angelica

NPM: 2015730029

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2019



**UNDERGRADUATE THESIS**

**INFORMATICS ANNOUNCEMENT COLLECTOR**



**Ellena Angelica**

**NPM: 2015730029**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2019**



## **ABSTRAK**

Penelitian pada skripsi ini adalah membuat fitur kolektor pengumuman di BlueTape sebagai solusi dari kekurangan kotak masuk *email* yang kurang terorganisir. Fitur kolektor pengumuman akan melakukan sinkronisasi *email* untuk mengumpulkan *email* yang berisi pengumuman di ruang lingkup jurusan Informatika menggunakan PHP IMAP. Setelah memastikan pengumuman dilakukan oleh pihak yang berwenang, pengumuman akan ditampilkan di BlueTape. Dengan memanfaatkan layanan LINE®, mahasiswa jurusan Informatika akan diberitahu apabila ada pengumuman baru di BlueTape.

Pengujian pada skripsi ini dilakukan dua kali. Pengujian pertama dilakukan dengan mengirimkan berbagai macam *email* ke alamat email milik BlueTape. Pengujian kedua dilakukan dengan melibatkan mahasiswa dan dosen secara langsung. Hasil pengujian menunjukkan fitur ini telah berfungsi sesuai harapan, yaitu dapat melakukan sinkronisasi email, dapat menampilkan email yang berisi pengumuman di BlueTape, dan dapat memunculkan notifikasi pada akun LINE®.

**Kata-kata kunci:** BlueTape, IMAP, LINE



## **ABSTRACT**

The research in this undergraduate thesis is to make announcement collector feature on BlueTape as a solution to the lack of an organized email inbox. The announcement collector feature will synchronize email to collect email which contains an announcement in the scope of Informatics department using PHP IMAP. After making sure the announcement is made by the authorities, the announcement will be displayed on BlueTape. By utilizing the LINE@ service, Informatics students will be notified when there is a new announcement at BlueTape.

The test in this undergraduate thesis is done twice. The first test was done by sending various kinds of email to the email address of BlueTape. The second test was carried out by involving students and lecturers directly. The test results show that this feature has functioned as expected, that is, it can synchronize email, can display emails containing announcements at BlueTape, and can display notifications on LINE@ accounts.

**Keywords:** BlueTape, IMAP, LINE



# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>ix</b>
<b>DAFTAR GAMBAR</b>	<b>xi</b>
<b>DAFTAR TABEL</b>	<b>xiii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	1
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 BlueTape . . . . .	5
2.2 Heroku [1] . . . . .	5
2.2.1 Arsitektur Heroku . . . . .	5
2.2.2 Deploy Perangkat Lunak . . . . .	20
2.2.3 Ephemeral Filesystem . . . . .	24
2.2.4 Basis Data dan Manajemen Data . . . . .	24
2.2.5 Verifikasi Akun . . . . .	29
2.3 Cron [2] . . . . .	29
2.4 Gmail API [3] . . . . .	31
2.4.1 Resource . . . . .	31
2.4.2 Scope . . . . .	32
2.4.3 Penggunaan pada umumnya . . . . .	32
2.4.4 Implementasi Otorisasi dari Sisi <i>Server</i> . . . . .	33
2.5 PHP IMAP [4] . . . . .	33
2.6 LINE [5] . . . . .	41
2.6.1 LINE Login . . . . .	41
2.6.2 LINE Bot Designer . . . . .	42
2.6.3 Clova . . . . .	42
2.6.4 LINE Pay . . . . .	43
2.6.5 Messaging API . . . . .	44
2.6.6 LINE@ Manager . . . . .	51
2.7 System Usability Scale [6] . . . . .	51
<b>3 ANALISIS</b>	<b>53</b>
3.1 Analisis Sistem yang Sudah Ada . . . . .	53
3.1.1 Aturan Konstribusi BlueTape . . . . .	54
3.1.2 Autentikasi . . . . .	55
3.1.3 Fitur - Fitur BlueTape . . . . .	56

3.1.4	Hak Akses . . . . .	61
3.2	Analisis Fitur yang Dibangun . . . . .	62
3.2.1	Diagram Use Case . . . . .	62
3.2.2	Modifikasi BlueTape agar Dapat Berjalan di Heroku . . . . .	63
3.2.3	Sinkronisasi Email . . . . .	66
3.2.4	Menghubungkan BlueTape dengan LINE . . . . .	67
<b>4</b>	<b>PERANCANGAN</b>	<b>69</b>
4.1	Perancangan Kelas . . . . .	69
4.1.1	Model . . . . .	70
4.1.2	View . . . . .	73
4.1.3	Controller . . . . .	73
4.2	Perancangan Basis Data . . . . .	77
4.3	Perancangan Antarmuka . . . . .	78
4.3.1	Perancangan Antarmuka pada BlueTape . . . . .	78
4.3.2	Perancangan Antarmuka pada Bot BlueTape . . . . .	81
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>83</b>
5.1	Implementasi . . . . .	83
5.1.1	Lingkungan Pengembangan . . . . .	83
5.1.2	Implementasi Basis Data . . . . .	84
5.1.3	Implementasi Kelas . . . . .	84
5.2	Pengujian . . . . .	85
5.2.1	Lingkungan Pengujian . . . . .	85
5.2.2	Pengujian Fungsional . . . . .	85
5.2.3	Pengujian Eksperimental . . . . .	89
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>107</b>
6.1	Kesimpulan . . . . .	107
6.2	Saran . . . . .	107
<b>DAFTAR REFERENSI</b>		<b>109</b>
<b>A KODE PROGRAM</b>		<b>111</b>
<b>B LAMPIRAN PENGUJIAN EKSPERIMENTAL</b>		<b>123</b>
B.1	Kuesioner . . . . .	123
B.1.1	Dosen . . . . .	124
B.1.2	Mahasiswa . . . . .	125
B.1.3	<i>System Usability Scale</i> dan Saran . . . . .	127
B.2	Hasil Mentah Kuesioner . . . . .	130

## DAFTAR GAMBAR

2.1	Diagram hubungan antara <i>process type</i> dan <i>dyno</i> . . . . .	6
2.2	Config vars pada dashboard Heroku . . . . .	10
2.3	Tabel buildpack heroku . . . . .	13
2.4	<i>Deploy</i> menggunakan Github <i>Dashboard</i> . . . . .	22
2.5	<i>Deploy</i> menggunakan Github secara manual . . . . .	22
2.6	<i>Deploy</i> menggunakan Github secara otomatis . . . . .	23
2.7	Tabel plan Heroku Postgres . . . . .	25
2.8	LINE Login . . . . .	41
2.9	LINE Bot Designer . . . . .	42
2.10	Situs web Clova . . . . .	42
2.11	Situs web LINE Pay . . . . .	43
2.12	Push message dan reply message pada Messaging API . . . . .	44
2.13	Arsitektur Messaging API . . . . .	45
2.14	Tampilan LINE Developers <i>console</i> saat login . . . . .	46
2.15	Tampilan LINE Developers <i>console</i> saat <i>register developer</i> . . . . .	46
2.16	Tampilan LINE Developers <i>console</i> saat membuat <i>channel</i> . . . . .	47
2.17	Tampilan LINE Developers <i>console</i> saat konfirmasi pembuatan <i>channel</i> . . . . .	48
3.1	Tampilan utama BlueTape . . . . .	53
3.2	Tampilan Cetak Transkrip . . . . .	57
3.3	Tampilan hasil Request Cetak Transkrip . . . . .	57
3.4	Tampilan Manajemen Transkrip BlueTape . . . . .	58
3.5	Tampilan request perubahan kuliah . . . . .	58
3.6	Tampilan manage perubahan kuliah . . . . .	59
3.7	Tampilan tambah jadwal dosen . . . . .	59
3.8	Tampilan jadwal dosen . . . . .	60
3.9	Tampilan edit jadwal dosen . . . . .	60
3.10	Tampilan lihat jadwal dosen . . . . .	61
3.11	Use case diagram fitur kolektor pengumuman . . . . .	62
4.1	Class diagram fitur kolektor pengumuman . . . . .	69
4.2	Mockup antarmuka main . . . . .	79
4.3	Mockup antarmuka read . . . . .	80
4.4	Mockup antarmuka bot BlueTape . . . . .	81
5.1	Pengumuman Ujian Eksperimental . . . . .	89
5.2	Kode QR Bot Shadowtape . . . . .	89
5.3	<i>Email</i> pada tanggal 24 April 2019 oleh Bapak Pascal . . . . .	90
5.4	<i>Email</i> pada tanggal 25 April 2019 oleh peneliti . . . . .	91
5.5	Tampilan pesan LINE setelah pengumuman disebar . . . . .	92
5.6	Tampilan BlueTape setelah URL dibuka dan pengguna belum <i>login</i> . . . . .	93
5.7	Tampilan URL pengumuman yang diumumkan oleh Pak Pascal Alfadian . . . . .	94
5.8	Tampilan URL pengumuman yang diumumkan oleh peneliti . . . . .	95

5.9	Diagram profil responden . . . . .	95
5.10	Diagram penggunaan <i>email</i> di kalangan mahasiswa . . . . .	96
5.11	Diagram penggunaan LINE di kalangan mahasiswa . . . . .	96
5.12	Diagram mahasiswa yang menerima notifikasi LINE . . . . .	97
5.13	Diagram mahasiswa yang dapat membuka URL pengumuman . . . . .	97
5.14	Diagram mahasiswa yang diarahkan kembali ke URL pengumuman setelah <i>login</i> . . . . .	98
5.15	Diagram dosen yang dapat melihat pengumuman yang diumumkannya . . . . .	98
5.16	Diagram dosen yang judul pengumumannya sesuai dengan subjek <i>email</i> yang ia kirim . . . . .	99
5.17	Diagram dosen yang isi pengumumannya sesuai dengan isi <i>email</i> yang ia kirim . . . . .	99
5.18	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang pertama . . . . .	100
5.19	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang kedua . . . . .	100
5.20	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang ketiga . . . . .	101
5.21	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang keempat . . . . .	101
5.22	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang kelima . . . . .	102
5.23	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang keenam . . . . .	102
5.24	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang ketujuh . . . . .	103
5.25	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang kedelapan . . . . .	103
5.26	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang kesembilan . . . . .	104
5.27	Diagram jawaban pertanyaan <i>System Usability Scale</i> yang kesepuluh . . . . .	104
5.28	Diagram Skor System Usability Scale . . . . .	105
B.1	Kuesioner bagian pertama . . . . .	123
B.2	Kuesioner bagian Dosen pertanyaan pertama . . . . .	124
B.3	Kuesioner bagian Dosen pertanyaan kedua dan ketiga . . . . .	124
B.4	Kuesioner bagian Dosen pertanyaan keempat . . . . .	125
B.5	Kuesioner bagian Mahasiswa pertanyaan pertama dan kedua . . . . .	125
B.6	Kuesioner bagian Mahasiswa pertanyaan ketiga . . . . .	126
B.7	Kuesioner bagian Mahasiswa pertanyaan keempat . . . . .	126
B.8	Kuesioner bagian Mahasiswa pertanyaan kelima . . . . .	126
B.9	Kuesioner bagian <i>System Usability Scale</i> dan Saran bagian 1 . . . . .	127
B.10	Kuesioner bagian <i>System Usability Scale</i> dan Saran bagian 2 . . . . .	128
B.11	Kuesioner bagian <i>System Usability Scale</i> dan Saran bagian 3 . . . . .	129

## DAFTAR TABEL

2.1	Tabel perbandingan antar tipe <i>dyno</i> . . . . .	8
2.2	Tabel perbandingan fitur antar tipe <i>dyno</i> . . . . .	9
2.3	Tabel perbandingan antar <i>stack</i> . . . . .	15
4.1	Rincian method <code>__construct</code> . . . . .	70
4.2	Rincian method <code>checkEmail</code> . . . . .	70
4.3	Rincian method <code>proceedEmail</code> . . . . .	71
4.4	Rincian method <code>__construct</code> . . . . .	72
4.5	Rincian method <code>proceedWebhook</code> . . . . .	72
4.6	Rincian method <code>pushMessageToAllFollowers</code> . . . . .	73
4.7	Rincian method <code>fifteenminutely</code> . . . . .	74
4.8	Rincian method <code>__construct</code> . . . . .	74
4.9	Rincian method <code>index</code> . . . . .	75
4.10	Rincian method <code>read</code> . . . . .	75
4.11	Rincian method <code>page</code> . . . . .	75
4.12	Rincian method <code>pagination</code> . . . . .	76
4.13	Rincian method <code>webhook</code> . . . . .	77
4.14	Perancangan Tabel Pengumuman . . . . .	77
4.15	Perancangan Tabel PengumumanLineFollowers . . . . .	78
5.1	Pengujian Filter <i>Email</i> Pengumuman . . . . .	86
5.2	Pengujian Mengirim <i>Email</i> dengan Isi <i>Email</i> yang Variatif . . . . .	86
5.3	Pengujian Notifikasi LINE . . . . .	88
5.4	Tabel saran . . . . .	105



1

## BAB 1

2

### PENDAHULUAN

#### 3 1.1 Latar Belakang

4 Pengumuman di jurusan Teknik Informatika UNPAR pada umumnya dilakukan lewat *email*.  
5 Pengumuman lewat *email* lebih praktis daripada pengumuman di papan pengumuman karena  
6 *email* dijamin sampai ke pihak yang dituju setelah dikirim. Namun, layanan *email* memiliki kotak  
7 masuk yang kurang terorganisir. Berbagai macam *email* yang masuk tercampur di kotak masuk  
8 sehingga dapat menyulitkan pemilik *email* untuk mencari *email* yang penting. Hal ini juga dapat  
9 mengakibatkan pengumuman-pengumuman penting tidak dibaca secara tidak sengaja.

10 Penelitian pada skripsi ini hendak membuat solusi untuk kekurangan *email* tersebut dengan  
11 cara membuat fitur kolektor pengumuman di BlueTape. Fitur ini akan mengumpulkan *email* yang  
12 berisi pengumuman dan menampilkannya di BlueTape. Agar penerima pengumuman tahu ada  
13 pengumuman baru di BlueTape, fitur ini juga akan memanfaatkan layanan LINE@. LINE@ adalah  
14 layanan dari *Line Corporation* yang memungkinkan pemilik bisnis atau organisasi membuat akun  
15 khusus (disebut akun LINE@) yang dapat mengirim pesan ke banyak pengikut secara bersamaan.  
16 Penerima pengumuman akan diminta untuk mengikuti akun LINE@ BlueTape sehingga dapat  
17 menerima notifikasi LINE apabila ada pengumuman baru. Agar BlueTape versi skripsi ini dapat  
18 diakses melalui internet, fitur ini juga akan memanfaatkan layanan Heroku.

#### 19 1.2 Rumusan Masalah

20 Rumusan masalah dari skripsi ini adalah:

- 21 • Bagaimana cara memodifikasi BlueTape agar dapat berjalan di Heroku?
- 22 • Bagaimana cara mengimplementasikan fitur kolektor pengumuman pada BlueTape?

#### 23 1.3 Tujuan

24 Tujuan dari skripsi ini adalah:

- 25 • Memodifikasi BlueTape agar dapat berjalan di Heroku.
- 26 • Mengimplementasikan fitur kolektor pengumuman pada BlueTape.

## 1 1.4 Batasan Masalah

2 Pada skripsi ini masalah dibatasi dengan batasan-batasan sebagai berikut:

- 3 • Fitur ini tidak akan menampilkan lampiran karena dapat membuat masalah lebih kompleks.
- 4 • Pengumuman di BlueTape dapat dilihat oleh semua mahasiswa dan dosen yang memiliki hak  
5 akses ke BlueTape.
- 6 • Semua akun yang mengikuti akun LINE@ BlueTape dapat menerima notifikasi apabila ada  
7 pengumuman baru.
- 8 • Sinkronisasi *email* dan pengiriman notifikasi LINE@ tidak dilakukan setiap saat, melainkan  
9 pada periode tertentu.

## 10 1.5 Metodologi

11 Metode penelitian pada skripsi ini sebagai berikut:

- 12 1. Melakukan studi literatur tentang PHP IMAP, LINE@, dan Heroku.
- 13 2. Memodifikasi BlueTape sehingga dapat mengumpulkan *email* yang berisi pengumuman.
- 14 3. Memodifikasi BlueTape sehingga dapat mengirim notifikasi ke akun Line@ BlueTape.
- 15 4. Memodifikasi BlueTape sehingga dapat berjalan di Heroku.
- 16 5. Melakukan pengujian.
- 17 6. Menulis dokumen skripsi.

## 18 1.6 Sistematika Pembahasan

19 Berikut adalah sistematika pembahasan skripsi ini :

- 20 1. Bab 1: Pendahuluan  
Bab ini membahas gambaran umum dari skripsi.
- 21 2. Bab 2: Dasar Teori  
Bab ini membahas dasar teori yang mendukung pembuatan skripsi ini.
- 22 3. Bab 3: Analisis  
Bab ini membahas analisis yang dilakukan terhadap masalah yang diusung dalam skripsi ini.
- 23 4. Bab 4: Perancangan  
Bab ini membahas perancangan sistem yang dibangun pada skripsi ini.
- 24 5. Bab 5: Implementasi dan Pengujian  
Bab ini membahas hasil implementasi yang dilakukan beserta pengujian sistem.

## 1 6. Bab 6: Kesimpulan dan saran

2 Bab ini berisi kesimpulan yang didapat dari penelitian dan saran oleh peneliti kepada pembaca  
3 yang hendak melanjutkan penelitian ini.



<sup>1</sup>

## BAB 2

<sup>2</sup>

### LANDASAN TEORI

<sup>3</sup> Bab ini berisi landasan teori yang dipakai pada skripsi ini.

#### <sup>4</sup> 2.1 BlueTape

<sup>5</sup> BlueTape adalah aplikasi yang berfungsi untuk membantu urusan-urusan *paper-based* di FTIS  
<sup>6</sup> UNPAR menjadi *paperless*. Aplikasi ini berbasis web dengan memanfaatkan *framework* CodeIgniter  
<sup>7</sup> dan ZURB Foundation. Selain itu, BlueTape menggunakan layanan OAuth dari Google untuk  
<sup>8</sup> autentikasi pengguna. Saat skripsi ini ditulis, aplikasi BlueTape memiliki tiga layanan, yaitu  
<sup>9</sup> Transkrip *Request / Manage*, Perubahan Kuliah *Request / Manage*, dan perekam jadwal dosen.  
<sup>10</sup> Layanan Transkrip *Request / Manage* memberikan layanan untuk melakukan permohonan serta  
<sup>11</sup> pencetakan transkrip mahasiswa. Layanan Perubahan Kuliah *Request / Manage* memberikan  
<sup>12</sup> layanan untuk permohonan dan pencetakan perubahan jadwal kuliah oleh dosen. Layanan perekam  
<sup>13</sup> jadwal dosen memberikan layanan untuk menyimpan dan menampilkan jadwal dosen. <sup>1</sup>

#### <sup>14</sup> 2.2 Heroku [1]

<sup>15</sup> Heroku adalah *cloud platform* yang memungkinkan penggunanya untuk membangun, menjalankan,  
<sup>16</sup> dan mengoperasikan aplikasi pada *cloud*. Heroku mendukung beberapa bahasa pemrograman,  
<sup>17</sup> meliputi: Ruby, Node.js, Java, Python, Clojure, Scala, Go, dan PHP.

##### <sup>18</sup> 2.2.1 Arsitektur Heroku

<sup>19</sup> Heroku memungkinkan penggunanya untuk menyebarkan, menjalankan, dan mengelola aplikasi  
<sup>20</sup> yang ditulis di dalam bahasa yang didukung oleh Heroku. Heroku mendefinisikan aplikasi sebagai  
<sup>21</sup> gabungan dari *source code* (kode program) yang ditulis di dalam salah satu bahasa yang didukung  
<sup>22</sup> Heroku, deskripsi *dependency* yang dipakai, dan *file Procfile*.

###### <sup>23</sup> 2.2.1.1 *Dependency*

<sup>24</sup> Agar aplikasi dapat dibangun dan dijalankan, aplikasi mungkin membutuhkan kode program dari  
<sup>25</sup> pihak ketiga. Kode program tersebut disebut *dependency*. Aturan peletakkan deskripsi *dependency*  
<sup>26</sup> berbeda-beda untuk tiap bahasa. Contoh: Aplikasi dengan bahasa Node.js menuliskan deskripsi  
<sup>27</sup> *dependency* pada *file package.json*.

---

<sup>1</sup><https://github.com/ftisunpar/BlueTape>

### 2.2.1.2 Procfile

Heroku dapat mengetahui bagian aplikasi mana yang dapat dijalankan untuk sebagian besar bahasa pemrograman. Contoh: Heroku langsung mengetahui bagian "main" pada `package.json` adalah bagian aplikasi yang dapat dijalankan pada aplikasi dengan bahasa Node.js. Namun, Heroku terkadang membutuhkan petunjuk secara eksplisit untuk mengetahui bagian yang dapat dijalankan. Petunjuk tersebut ditulis di sebuah file teks bernama Procfile.

Procfile adalah file tanpa ekstensi file (Contoh ekstensi file: ".txt") yang diletakkan di direktori root. Isi dari Procfile adalah teks yang tiap barisnya memiliki format sebagai berikut:

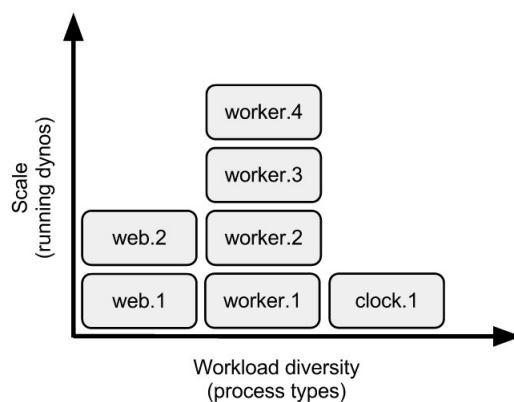
```
<process type>: <command>
```

Keterangan:

- `<process type>`: nama *process type*, contoh: `urgentworker`. Aplikasi yang sederhana hanya membutuhkan *process type web* saja.
- `<command>`: perintah yang harus dijalankan oleh setiap *dyno* dari *process type* tersebut pada saat startup.

### 2.2.1.3 Process Type

*Process type* adalah prototipe yang menjadi tempat dimana dyno dibentuk. *Process type* yang tersedia dapat berbeda untuk tiap bahasa pemrograman. Namun, ada dua *process type* spesial pada Heroku, yaitu *process type web* dan *process type release*. *Process type web* adalah satu-satunya *process type* yang dapat menerima arus HTTP eksternal dari *router* Heroku. *Process type release* adalah *process type* yang digunakan untuk menyebutkan perintah yang dijalankan selama fase *release*.



Gambar 2.1: Diagram hubungan antara *process type* dan *dyno*

*Process type* dan *dyno* saling berhubungan. Hubungan tersebut dapat dilihat pada Gambar 2.1. Sumbu x menyatakan *process type* yang dipakai, sementara sumbu y menyatakan jumlah *dyno* yang berjalan pada *process type* tersebut. Semakin banyak *dyno* pada suatu *process type* maka konkurensi untuk pekerjaan yang ditangani *process type* tersebut akan meningkat. Semakin banyak *process type* maka semakin beragam beban kerja.

1 Jumlah *dyno* yang berjalan pada *process type* dapat diatur dengan mengetikkan perintah berikut  
2 pada *command shell*:

```
3 $ heroku ps:scale <process=dyno list>
```

6 Keterangan:

- 7 • <process=dyno list>: daftar pasangan *process type* dengan jumlah *dyno* yang ditugaskan  
8 untuk proses tersebut.

9 Contoh:

```
10 $ heroku ps:scale web=2 worker=4 clock=1
```

#### 13 2.2.1.4 Dyno

14 *Dyno* adalah wadah aplikasi berbasis Unix yang menjalankan kode program di Heroku. Pada  
15 umumnya, Heroku akan menjalankan satu *web dyno* secara otomatis saat aplikasi disebar ke Heroku  
16 untuk pertama kali.

17 *Dyno* dikelompokkan ke dalam tiga kelompok, yaitu:

- 18 • *Web dyno*

19 *Web dyno* adalah *dyno* yang berjalan pada *process type web*. *Web dyno* adalah satu-satunya  
20 *dyno* yang dapat menerima arus HTTP dari *router* Heroku.

- 21 • *Worker dyno*

22 *Worker dyno* merujuk kepada *dyno* yang berjalan pada *process type* yang disebutkan di dalam  
23 *Procfile* selain *process type web*. *Worker dyno* biasanya digunakan untuk pekerjaan pada latar  
24 belakang, sistem antrian, dan pekerjaan berjangka waktu tertentu.

- 25 • *One-off dyno*

26 *One-off dyno* adalah *dyno* yang bersifat sementara. *One-off dyno* tidak dituliskan pada *Procfile*  
27 untuk menjalankannya, melainkan menggunakan *command shell*. Contoh penggunaan *one-off*  
28 *dyno* adalah pada saat migrasi basis data.

29 Heroku menyediakan berbagai macam tipe *dyno*, yaitu: tipe *free*, tipe *hobby*, tipe *standard*,  
30 dan tipe *performance*. Tipe *dyno* mempengaruhi karakteristik *dyno* dan kinerja aplikasi. Pada  
31 aplikasi dengan tipe *dyno free* dan tipe *dyno hobby*, semua *process type* pada aplikasi tersebut  
32 hanya dapat memakai tipe *dyno* tersebut. Namun, tipe *standard* dan tipe *dyno performance* dapat  
33 digunakan di aplikasi yang sama. Pada *dyno* tipe *free* dan tipe *hobby*, jumlah *dyno* yang dapat  
34 dipakai untuk tiap *process type* adalah satu. Pada tipe *dyno performance*, jumlah *dyno* yang dapat  
35 dipakai untuk tiap *process type* adalah sepuluh. Pada tipe *dyno free*, jumlah *dyno* yang dapat  
36 dijalankan secara bersamaan pada satu aplikasi adalah dua. Pada tipe *dyno* lainnya, jumlah *dyno*  
37 yang dapat dijalankan secara bersamaan pada satu aplikasi adalah seratus.

38 Tipe *dyno free* memiliki beberapa batasan, yaitu dapat masuk ke kondisi *sleep* dan penggunaan  
39 per bulannya dibatasi dengan *free dyno hours*. Kondisi *sleep* adalah kondisi saat *web dyno* tidak

- 1 aktif. Kondisi *sleep* terjadi apabila tidak ada arus HTTP masuk setelah 30 menit aktif. Apabila  
 2 ada arus HTTP masuk pada saat aplikasi tertidur, maka aplikasi akan aktif kembali. Namun, ada  
 3 keterlambatan saat memproses arus HTTP tersebut. *Free dyno hours* adalah jumlah jam saat *web*  
 4 *dyno* tidak dalam kondisi *sleep*, saat *worker dyno* berjalan, dan saat *one-off dyno* berjalan. Akun  
 5 yang belum diverifikasi dapat menggunakan 550 *free dyno hours* tiap bulannya, sedangkan akun  
 6 yang telah diverifikasi dapat menggunakan 1000 *free dyno hours* tiap bulannya. Setiap aplikasi  
 7 dengan tipe *dyno free* yang dimiliki pengguna memakai kuota yang sama. Apabila penggunaan *dyno*  
 8 telah melebihi 80% kuota, maka pengguna akan dikirimi *email* peringatan. Apabila penggunaan  
 9 telah melebihi 100% kuota, maka pengguna akan dikirimi *email* notifikasi kedua dan aplikasinya  
 10 akan masuk ke kondisi *sleep* selama sisa hari pada bulan tersebut. Jumlah *free dyno hours*  
 11 yang tersisa dapat dilihat dengan mengetikkan "`heroku ps -a`" diikuti dengan nama aplikasi  
 12 di *command shell* atau dengan melihat bagian *free dyno usage* pada halaman billing (<https://dashboard.heroku.com/account/billing>).  
 13
- 14 Perbandingan antar tipe *dyno* dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1: Tabel perbandingan antar tipe *dyno*

Tipe <i>dyno</i>	Kondisi <i>sleep</i>	Memory (RAM)	Harga <i>dyno</i> per bulan
<i>free</i>	Ada	512 MB	\$0
<i>hobby</i>	Tidak	512 MB	\$7
<i>standard-1x</i>	Tidak	512 MB	\$25
<i>standard-2x</i>	Tidak	1024 MB	\$50
<i>performance-m</i>	Tidak	2.5 GB	\$250
<i>performance-l</i>	Tidak	14 GB	\$500

Tabel 2.2: Tabel perbandingan fitur antar tipe *dyno*

Tipe <i>dyno</i>	Fitur
<i>free</i>	<ul style="list-style-type: none"> <li>Mendukung penyebaran aplikasi menggunakan Git dan Docker.</li> <li>Mengganti <i>domain</i>.</li> <li>Dapat menggunakan <i>pipeline</i><sup>2</sup></li> <li>Pembaruan sistem operasi secara berkala.</li> <li>Pembaruan versi bahasa pemrograman secara berkala.</li> </ul>
<i>hobby</i>	<ul style="list-style-type: none"> <li>Semua fitur diatas.</li> <li>Dapat menggunakan SSL dan sertifikat TLS.</li> <li>Dapat menggunakan <i>Application metrics</i></li> </ul>
<i>standard-1x</i>	<ul style="list-style-type: none"> <li>Semua fitur diatas.</li> <li>Dapat menambah jumlah <i>dyno</i>.</li> <li>Dapat mengubah ukuran <i>dyno</i>.</li> <li>Dapat menggunakan fitur <i>preboot</i>.</li> <li>Dapat menggunakan fitur <i>language runtime metrics</i>.</li> </ul>
<i>standard-2x</i>	<ul style="list-style-type: none"> <li>Semua fitur diatas.</li> </ul>
<i>performance-m</i>	<ul style="list-style-type: none"> <li>Semua fitur diatas.</li> <li>Dapat menggunakan fitur <i>autoscaling</i> untuk <i>web dynos</i></li> <li>Sumber daya komputasi terdedikasi.</li> </ul>
<i>performance-l</i>	<ul style="list-style-type: none"> <li>Semua fitur diatas.</li> </ul>

### <sup>1</sup> 2.2.1.5 *Config vars*

<sup>2</sup> Pada Heroku, *environment variable* disimpan di dalam *config vars*. *Environment variable* adalah konfigurasi yang dapat berubah-ubah pada lingkungan yang berbeda. Konfigurasi tersebut meliputi

<sup>2</sup> *Pipeline* adalah sekelompok aplikasi Heroku yang menggunakan kode program yang sama.

1 informasi *database*, informasi kredensial, atau informasi lain yang bersifat spesifik pada aplikasi.  
 2 Penggunaan *environment variable* dapat menghindari tersimpannya informasi penting di pengontrol  
 3 versi.

4 *Config vars* dapat diatur dengan tiga cara :

5 • Menggunakan Heroku CLI

6 Heroku CLI menggunakan *command shell* untuk menerima perintah yang harus dijalankan.

7 Berikut daftar perintah yang berkaitan dengan *config vars* beserta fungsinya:

8 – Menampilkan seluruh *config vars* beserta nilainya :

```
9 $ heroku config
```

12 – Menampilkan nilai dari *config vars* tertentu

```
13 $ heroku config : get <config vars>
```

16 Keterangan :

17 \* <config vars>: nama *config vars*

18 – Menambah *config vars*

```
19 $ heroku config : set <config vars> = <config values>
```

22 Keterangan :

23 \* <config vars>: nama *config vars*

24 \* <config values>: nilai dari *config vars* tersebut

25 – Menghapus *config vars*

```
26 $ heroku config : unset <config vars>
```

29 Keterangan :

30 \* <config vars> : nama *config vars*

31 • Menggunakan Heroku *Dashboard*



Gambar 2.2: Config vars pada dashboard Heroku

32 Heroku *dashboard* berada di <https://dashboard.heroku.com/>. Setelah memilih menu  
 33 "Settings" di aplikasi yang dipilih, pilih menu "Config vars" untuk melihat, menambah, dan  
 34 menghapus *config vars*. Tampilan menu "Config vars" dapat dilihat pada Gambar 2.2.

- 1     • Menggunakan Heroku Platform API

2     *Config vars* dapat diatur dengan mengirimkan PATCH *request* ke Heroku Platform API dengan  
3     format berikut:

4           PATCH /apps/{app\_id\_or\_name}/config-vars

5     Contoh *request* menggunakan curl:

6       \$ curl -n -X PATCH https://api.heroku.com/apps/\$APP\_ID\_OR\_NAME/config-vars \  
7           -d '{  
8            "FOO": "bar",  
9            "BAZ": "qux"  
10          }' \  
11           -H "Content-Type: application/json" \  
12           -H "Accept: application/vnd.heroku+json; version=3"

13     Contoh *response* dari Heroku Platform API:

14       HTTP/1.1 200 OK  
15       ETag: "0123456789abcdef0123456789abcdef"  
16       Last-Modified: Sun, 01 Jan 2012 12:00:00 GMT  
17       RateLimit-Remaining: 4500  
18       {  
19        "FOO": "bar",  
20        "BAZ": "qux"  
21      }

22     Apabila pengguna ingin melihat *config vars*, maka pengguna dapat mengirimkan GET *request*  
23     dengan format berikut:

24           GET /apps/{app\_id\_or\_name}/config-vars

25     Contoh *request* menggunakan curl:

26       \$ curl -n https://api.heroku.com/apps/\$APP\_ID\_OR\_NAME/config-vars \  
27           -H "Accept: application/vnd.heroku+json; version=3"

28     Contoh *response* dari Heroku Platform API:

29       HTTP/1.1 200 OK  
30       ETag: "0123456789abcdef0123456789abcdef"  
31       Last-Modified: Sun, 01 Jan 2012 12:00:00 GMT  
32       RateLimit-Remaining: 4500  
33       {  
34        "FOO": "bar",  
35        "BAZ": "qux"  
36      }

- 1 Dalam mengatur *config vars*, ada beberapa hal yang harus diperhatikan:
- 2 • Setiap *config vars* ditambah atau dihapus, aplikasi akan dimulai ulang dan *release* baru akan  
3 dibuat.
- 4 • Jika aplikasi menggunakan *add-ons*, biasanya *add-ons* tersebut akan menambahkan satu atau  
5 lebih *config vars* ke aplikasi. Nilai dari *config vars* tersebut mungkin diperbarui oleh penyedia  
6 *add-ons* kapan saja.
- 7 • Data *config vars* (kombinasi dari semua kunci dan nilainya) tidak dapat melebihi 32 kb per  
8 aplikasi
- 9 • Nama *config vars* tidak boleh diawali dengan tanda hubung bawah dua kali (\_).
- 10 • Nama *config vars* tidak bisa diawali dengan HEROKU\_, kecuali ditambahkan oleh platform  
11 Heroku sendiri.

12 **2.2.1.6 Add-ons**

13 Pengguna Heroku dapat memanfaatkan *add-ons* untuk memakai layanan penyokong, misalnya  
14 basis data, sistem antrean, layanan *email*, dan lain-lain. Pengguna Heroku dapat mencari *add-ons* di  
15 situs web *Elements Marketplace* (<https://elements.heroku.com/addons>). Beberapa *add-ons* di  
16 dalam situs web tersebut disediakan oleh Heroku dan beberapa lainnya disediakan oleh pihak ketiga.  
17 Pengguna Heroku dapat memasang *add-ons* pada aplikasi dengan menekan tombol "Install" di  
18 situs web *Elements Marketplace* atau dengan mengetikkan perintah berikut pada *command shell* :

19  
20 \$ heroku addons:create <nama add-ons>:<tipe add-ons>  
21

22 Keterangan:

- 23 • <nama add-ons> : nama *add-ons*  
24 • <tipe add-ons> : tipe *add-ons*

25 Contoh:

26  
27 \$ heroku addons:create heroku-redis:hobby-dev  
28

29 Menambah *add-ons* selain *add-ons* Heroku Postgres dan Heroku Connect membutuhkan verifikasi  
30 akun.

31 **2.2.1.7 Slug**

32 *Slug* adalah gabungan dari *source code*, *dependency*, *language runtime*, dan hasil *build* yang siap  
33 untuk dijalankan. *Slug* dibuat oleh *buildpack* setiap pengguna melakukan *deploy*. Ukuran maksimal  
34 *slug* adalah 500 MB. Waktu pembuatan *slug* dibatasi sampai 15 menit. Agar waktu pembuatan *slug*  
35 lebih singkat, pengguna dapat mengecualikan *file* yang tidak diperlukan untuk menjalankan aplikasi  
36 dengan menuliskannya di dalam *file .slugignore*. *File .slugignore* adalah *file* yang menyebutkan  
37 *file* yang harus diabaikan pada saat pembuatan *slug*. *File* ini diletakkan di direktori *root*. Cara  
38 penulisan *.slugignore* kurang lebih sama dengan *.gitignore*. Contoh isi *.slugignore*:

```

1 # Heres a comment
2 *.psd
3 *.pdf
4 /test
5 /spec
6
7

```

### 8 2.2.1.8 Buildpack

Buildpack	Shorthand
<a href="#">Ruby</a>	heroku/ruby
<a href="#">Node.js</a>	heroku/nodejs
<a href="#">Clojure</a>	heroku/clojure
<a href="#">Python</a>	heroku/python
<a href="#">Java</a>	heroku/java
<a href="#">Gradle</a>	heroku/gradle
<a href="#">Grails 3.x</a>	heroku/gradle
<a href="#">Scala</a>	heroku/scala
<a href="#">Play 2.x</a>	heroku/scala
<a href="#">PHP</a>	heroku/php
<a href="#">Go</a>	heroku/go

Gambar 2.3: Tabel buildpack heroku

9     *Buildpack* adalah kumpulan kode program yang bertugas untuk mengubah *source code* menjadi  
10    *slug*, sehingga *dyno* dapat mengeksekusinya. Heroku biasanya akan secara otomatis menugaskan  
11    *buildpack* yang akan dipakai setelah *build* pertama berhasil. Daftar *buildpack* yang tersedia di  
12    Heroku dapat dilihat pada Gambar 2.3. Kolom *Buildpack* menyatakan nama *buildpack* dan kolom  
13    *shorthand* menyatakan nama panggil *buildpack* di Heroku CLI.

14     Jumlah *buildpack* yang dipakai oleh satu aplikasi biasanya hanya satu, tapi ada beberapa kasus  
15    *buildpack* yang dipakai lebih dari satu. Contoh kasus *buildpack* tambahan dibutuhkan:

- 16
  - Satu aplikasi memiliki lebih dari satu bahasa pemrograman.
  - Aplikasi menjalankan *daemon*.
  - Aplikasi membutuhkan *dependency* yang diambil dengan *APT (Advanced Package Tool)*.

19     *Buildpack* dapat diatur dengan mengetikkan perintah di *command shell* menggunakan Heroku  
20    CLI. Berikut daftar perintah yang berkaitan dengan *buildpack* beserta fungsinya:

- 21
  - Mengatur *buildpack* yang dipakai saat aplikasi pertama kali dibuat:

```
$ heroku create myapp --buildpack <nama buildpack>
```

25     Keterangan:

1       – <nama buildpack>: nama panggil *buildpack*, contoh: heroku/php.

- 2     ● Mengubah dengan mengatur nilai *buildpack*:

3           \$ heroku buildpacks:set <nama buildpack>

6     Keterangan:

7       – <nama buildpack>: nama panggil *buildpack*, contoh: heroku/php.

- 8     ● Menghilangkan *buildpack* dari aplikasi:

9           \$ heroku buildpacks:remove <nama buildpack>

12    Keterangan:

13       – <nama buildpack>: nama panggil *buildpack*, contoh: heroku/php.

- 14    ● Mencari buildpack:

15           \$ heroku buildpacks:search <kata kunci>

18    Keterangan:

19       – <kata kunci>: kata kunci pencarian, misalnya nama bahasa pemrograman. Contoh  
20       kata kunci: elixir.

- 21    ● Menampilkan informasi *buildpack*:

22           \$ heroku buildpacks:info <nama buildpack>

25    Keterangan:

26       – <nama buildpack>: nama panggil *buildpack*, contoh: heroku/php.

- 27    ● Mengembalikan aplikasi ke *buildpack* awalnya:

28           \$ heroku buildpacks:clear

- 31    ● Mengatur urutan eksekusi *buildpack*:

32           \$ heroku buildpacks:set --index <index> <nama buildpack>

35    Keterangan:

36       – <index>: urutan eksekusi *buildpack*

37       – <nama buildpack>: nama panggil *buildpack*, contoh : heroku/php.

- 38    ● Melihat daftar *buildpack* yang dipakai pada satu aplikasi:

39           \$ heroku buildpacks

<sup>1</sup> **2.2.1.9 Stack**

<sup>2</sup> *Stack* merujuk pada sistem operasi yang dikelola dan dipelihara oleh Heroku. *Stack* biasanya  
<sup>3</sup> memakai sistem operasi Linux, misalnya Ubuntu. Saat skripsi ini dibuat, Heroku menyediakan tiga  
<sup>4</sup> pilihan *stack*: Cedar-14, Heroku-16, dan Heroku-18. Perbandingan ketiga *stack* tersebut dapat  
<sup>5</sup> dilihat di Tabel 2.3.

Tabel 2.3: Tabel perbandingan antar *stack*

Nama <i>stack</i>	Sistem Operasi yang dipakai	Periode dukungan
Cedar-14	Ubuntu 14.04	Sampai bulan April 2019
Heroku-16	Ubuntu 16.04	Sampai bulan April 2021
Heroku-18	Ubuntu 18.04	Sampai bulan April 2023

<sup>6</sup>

<sup>7</sup> Pengguna Heroku dapat melihat *stack* yang dipakai pada suatu aplikasi dengan mengetikkan  
<sup>8</sup> perintah berikut pada *command shell*:

<sup>9</sup>

```
$ heroku stack
```

<sup>10</sup>

<sup>12</sup> Pada saat aplikasi baru dibentuk, Heroku akan secara otomatis memakai *stack* yang memiliki  
<sup>13</sup> periode dukungan paling panjang. Namun, pengguna dapat mengganti *stack* yang dipakai dengan  
<sup>14</sup> mengetikkan perintah berikut pada *command shell*:

<sup>15</sup>

```
$ heroku stack:set <stack>
```

<sup>16</sup>

<sup>18</sup> Keterangan:

- <sup>19</sup> • <*stack*>: nama *stack*, contoh: cedar-14.

<sup>20</sup> **2.2.1.10 Region**

<sup>21</sup> Aplikasi yang ada di Heroku dapat disebarluaskan ke lokasi geografis yang berbeda. Pada heroku,  
<sup>22</sup> lokasi penyebaran aplikasi disebut *region*. *Region* yang tersedia untuk pengguna Heroku yang  
<sup>23</sup> merupakan perorangan adalah lokasi dengan *runtime Common Runtime*, sedangkan *region* yang  
<sup>24</sup> tersedia untuk pengguna Heroku yang merupakan badan bisnis (Heroku Enterprise) adalah lokasi  
<sup>25</sup> dengan *runtime Private Space*. Pada saat skripsi ini dibuat, pengguna Heroku yang merupakan  
<sup>26</sup> perorangan hanya dapat memilih antara *Europe* (Eropa) dan *United States* (Amerika Serikat).  
<sup>27</sup> Apabila pengguna tidak memilih *region* saat membuat aplikasi, maka Heroku akan secara otomatis  
<sup>28</sup> mengatur *region* menjadi *United States*. Beberapa layanan yang disediakan untuk membangun  
<sup>29</sup> aplikasi di Heroku hanya dapat digunakan di *region* tertentu, sehingga pengguna lebih baik  
<sup>30</sup> merencanakannya terlebih dahulu sebelum memilih *region*.

<sup>31</sup> Berikut perintah-perintah penting yang berkaitan dengan *region* yang dapat diketikkan pada  
<sup>32</sup> *command shell* :

- <sup>33</sup> • Memeriksa *region* yang tersedia di Heroku:

<sup>34</sup>

```
$ heroku regions
```

<sup>35</sup>

- 1     • Mengatur *region* aplikasi:

2       \$ heroku create --region <id region>

5     Keterangan:

6       – <id region>: *id region* yang ingin dipakai, contoh: eu. *Id region* bisa dilihat dengan  
7       memeriksa daftar region yang tersedia.

- 8     • Memeriksa *region* yang dipakai oleh aplikasi:

9       \$ heroku info

### 12 2.2.1.11 Releases

13     Setiap pengguna melakukan *deploy*, merubah *config vars*, dan merubah daftar *add-ons* yang  
14    dipakai, Heroku akan mencatat aktivitas tersebut di *releases*. *Releases* adalah buku besar yang  
15    mencatat setiap ada perubahan baru pada aplikasi. *Releases* berbentuk tabel yang memiliki empat  
16    kolom, yaitu "Rel", "Change", "By", dan "When". Kolom "rel" berisi versi aplikasi, kolom "Change"  
17    berisi keterangan singkat perubahan yang terjadi, kolom "By" berisi alamat *email* pengguna yang  
18    melakukan perubahan, dan kolom "When" berisi waktu perubahan tersebut terjadi. Pengguna dapat  
19    melihat *releases* dengan mengetikkan perintah berikut di *command shell*:

20       \$ heroku releases

23     Berikut contoh isi *releases*:

Rel	Change	By	When
v52	Config add AWS_S3_KEY	jim@example.com	5 minutes ago
v51	Deploy de63889	stephan@example.com	7 minutes ago
v50	Deploy 7c35f77	stephan@example.com	3 hours ago
v49	Rollback to v46	joe@example.com	2010-09-12 15:32:17 -0700

32     Releases berguna ketika pengguna ingin mengembalikan aplikasi ke versi tertentu. Cara  
33    mengembalikan aplikasi ke versi tertentu adalah dengan mengetikkan perintah berikut pada *command*  
34    *shell*:

36       \$ heroku releases:rollback <version>

38     Keterangan:

- 39       • <version>: Versi aplikasi

### 40 2.2.1.12 Log

41     Log adalah catatan setiap proses yang terjadi di aplikasi. Heroku menggunakan Logplex untuk  
42    menyampaikan log ini. Logplex akan secara otomatis menambahkan entri log baru dari semua  
43    dyno yang berjalan di aplikasi, dan juga komponen lain seperti router. Pengguna dapat melihat log  
44    dengan cara mengetikkan perintah berikut pada *command shell*:

```
1 $ heroku logs
```

4 Isi *log* memiliki format sebagai berikut:

```
5 <timestamp> <source>[<dyno>]: <message>
```

8 Keterangan:

- 9 • <timestamp>: Waktu *log* dicatat dengan tingkat ketelitian hingga mikrodetik.
- 10 • <source>: Jika *log* berasal dari *dyno*, maka isinya adalah "app". Jika *log* berasal dari komponen  
11 sistem Heroku (HTTP *router*, *dyno manager*) maka isinya adalah "heroku".
- 12 • dyno: Nama *dyno* atau komponen yang dicatat di *log*.
- 13 • message: Keterangan *log*.

14 Berikut contoh isi *log*:

```
15
16 2010-09-16T15:13:46.677020+00:00 app[web.1]: Processing PostController#list (for
17   ↪ 208.39.138.12 at 2010-09-16 15:13:46) [GET]
18 2010-09-16T15:13:46.677023+00:00 app[web.1]: Rendering template within layouts/
19   ↪ application
20 2010-09-16T15:13:46.677902+00:00 app[web.1]: Rendering post/list
21 2010-09-16T15:13:46.678990+00:00 app[web.1]: Rendered includes/_header (0.1ms)
22 2010-09-16T15:13:46.698234+00:00 app[web.1]: Completed in 74ms (View: 31, DB: 40)
23   ↪ | 200 OK [http://myapp.herokuapp.com/]
24 2010-09-16T15:13:46.723498+00:00 heroku[router]: at=info method=GET path="/posts"
25   ↪ host=myapp.herokuapp.com" fwd="204.204.204.204" dyno=web.1 connect=1ms
26   ↪ service=18ms status=200 bytes=975
27 2010-09-16T15:13:47.893472+00:00 app[worker.1]: 2 jobs processed at 16.6761 j/s,
28   ↪ 0 failed ...
```

30 Heroku mengelompokkan *log* ke dalam empat kelompok:

- 31 • *App logs*

32 *Log* yang berasal dari aplikasi, termasuk *log* yang dihasilkan oleh kode program dan *dependency*.  
33 Pengguna dapat melihat *app logs* dengan cara mengetikkan perintah berikut pada *command shell*:

```
36 $ heroku logs --source app
```

- 38 • *System Logs*

39 *Log* yang berisi pesan tentang tindakan yang diambil oleh infrastruktur Heroku mewakili  
40 aplikasi, misalnya: saat terjadi *error*. Pengguna dapat melihat *system logs* dengan cara  
41 mengetikkan perintah berikut pada *command shell*:

```
42 $ heroku logs --source heroku
```

1     ● *API logs*

2       *Log* yang berisi pesan tentang tindakan administratif yang diambil oleh pembuat aplikasi,  
3       misalnya: saat melakukan *deploy*. Pengguna dapat melihat *API logs* dengan cara mengetikkan  
4       perintah berikut pada *command shell*:

5           \$ heroku logs --source app --dyno api  
6  
7

8     ● *Add-on logs*

9       *Log* yang berisi pesan dari layanan *add-ons*.

10      Pengguna Heroku dapat menuliskan *log* dengan cara mengirimkan pesan ke *standard out (stdout)*  
11     atau *standard error (stderr)*. Cara mengirimkannya berbeda-beda tergantung bahasa pemrograman  
12     yang dipakai. Pada bahasa PHP, *log* dapat dikirim dengan menggunakan fungsi "*error\_log*" atau  
13     menulis ke "php://stderr".

14      Contoh penggunaan fungsi "*error\_log*":

15           error\_log("hello, this is a test!");  
16  
17

18      Contoh menulis ke "php://stderr":

19           file\_put\_contents("php://stderr", "hello, this is a test!\n");  
20  
21

22      Apabila aplikasi menggunakan *framework* Codeigniter, maka pembuat aplikasi perlu menambahkan  
23     file *application/core/MY\_Log.php* dan mengatur "sub-class prefix" menjadi "MY\_". Isi file  
24     tersebut adalah sebagai berikut:

25  
26           <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');  
27  
28  
29           // this class is adapted from system/libraries/Log.php  
30           /\*\*  
31            \* CodeIgniter  
32            \*  
33            \* An open source application development framework for PHP 5.1.6 or newer  
34            \*  
35            \* @package CodeIgniter  
36            \* @author EllisLab Dev Team  
37            \* @copyright Copyright (c) 2008 - 2014, EllisLab, Inc.  
38            \* @copyright Copyright (c) 2014 - 2015, British Columbia Institute of Technology  
39            \*        → (<http://bcit.ca/>)  
40            \* @license [http://codeigniter.com/user\\_guide/license.html](http://codeigniter.com/user_guide/license.html)  
41            \* @link <http://codeigniter.com>  
42            \* @since Version 1.0  
43            \* @filesource  
44            \*/  
45

```
1 // -----
2
3 /**
4  * Logging Class
5 *
6  * @package CodeIgniter
7  * @subpackage Libraries
8  * @category Logging
9  * @author EllisLab Dev Team
10 * @link http://codeigniter.com/user_guide/general/errors.html
11 */
12 class MY_Log {
13
14     protected $_threshold = 1;
15     protected $_date_fmt = 'Y-m-d H:i:s';
16     protected $_levels = array('ERROR' => '1', 'DEBUG' => '2', 'INFO' => '3', 'ALL
17         => '4');
18
19 /**
20  * Constructor
21 */
22 public function __construct()
23 {
24     $config =& get_config();
25
26     if (is_numeric($config['log_threshold']))
27     {
28         $this->_threshold = $config['log_threshold'];
29     }
30
31     if ($config['log_date_format'] != '')
32     {
33         $this->_date_fmt = $config['log_date_format'];
34     }
35 }
36
37 // -----
38
39 /**
40  * Write Log to php://stderr
41 *
42  * Generally this function will be called using the global log_message()
```

```

1      ↢ function
2
3      *
4      * @param string the error level
5      * @param string the error message
6      * @param bool whether the error is a native PHP error
7      * @return bool
8
9      */
10
11     public function write_log($level = 'error', $msg, $php_error = FALSE)
12     {
13
14         $level = strtoupper($level);
15
16         if ( ! isset($this->_levels[$level]) OR ($this->_levels[$level] > $this->
17             ↢ _threshold))
18         {
19             return FALSE;
20         }
21
22         file_put_contents('php://stderr', $level.' '.(($level == 'INFO') ? ' - ' :
23             ↢ '-').' '.date($this->_date_fmt). ' --> '.$msg."\n");
24
25         return TRUE;
26     }
27
28 }
29 // END Log Class
30
31 /* End of file MY_Log.php */
32 /* Location: ./application/core/MY_Log.php */

```

### 30 2.2.2 Deploy Perangkat Lunak

31 Heroku menggunakan Git sebagai sarana utama untuk melakukan *deploy* ke Heroku. Deploy  
 32 adalah proses penyebaran aplikasi dari satu lingkungan ke lingkungan lain, misalnya dari lingkungan  
 33 lokal aplikasi ke lingkungan Heroku. Namun, Heroku juga menyediakan cara lain untuk melakukan  
 34 *deploy*:

- 35 • Docker
- 36 • GitHub
- 37 • Tombol Deploy di dashboard Heroku
- 38 • WAR deployment

### 1    2.2.2.1 Deploy Menggunakan Git

2       Deploy dengan Git dapat dilakukan apabila Git telah terpasang. Git dapat dipasang dengan  
3       mengikuti petunjuk unduhan pada <https://git-scm.com>. Sebelum *deploy* dengan Git dapat  
4       dilakukan, Git perlu diinisialisasi terlebih dahulu dengan mengetikkan perintah berikut pada  
5       *command shell* :

```
6       $ git init  
7  
8
```

9       Setelah melakukan inisialisasi Git, aplikasi Heroku dapat dibuat. Setiap aplikasi Heroku dibuat,  
10      maka `git remote` secara otomatis juga dibuat. Nama *remote* dapat diperiksa dengan mengetikkan  
11      perintah berikut pada *command shell* :

```
12      $ git remote  
13  
14
```

15      Nama *remote* dapat diubah dengan mengetikkan perintah berikut pada *command shell* :

```
16      $ git remote rename <nama lama> <nama baru>  
17  
18
```

19      Keterangan:

- 20       • `<nama lama>`: nama *remote* yang ingin diganti.  
21       • `<nama baru>`: nama baru untuk *remote* tersebut.

22      Deploy dapat dimulai dengan mengetikkan perintah berikut pada *command shell* :

```
23  
24      $ git add <nama file>  
25      $ git commit -m "<message>"  
26      $ git push <nama remote> <nama branch>  
27
```

28      Keterangan :

- 29       • `<nama file>`: nama *file* yang telah diubah. Menulis tanda titik untuk nama *file* mewakili  
30       semua *file* yang telah berubah.
- 31       • `<message>`: pesan yang menerangkan hal yang ingin disebar.
- 32       • `<nama remote>` : nama *remote* dari tujuan deploy. Bila *developer* tidak mengubah nama  
33       *remote*, nama remotenya adalah `heroku`.
- 34       • `<nama branch>` : nama cabang dari tujuan deploy. Heroku secara otomatis membuat satu  
35       cabang bernama `master`.

### 36    2.2.2.2 Deploy Menggunakan Docker

37       Deploy dengan Docker dapat dilakukan apabila Docker telah terpasang dan pengguna Heroku  
38       telah masuk ke akun Heroku. Langkah-langkah untuk melakukan *deploy* menggunakan Docker  
39       adalah sebagai berikut:

- 40       • Masuk ke *Container Registry* dengan mengetikkan perintah berikut pada *command shell*:

```
1 $ heroku container:login
2
3
```

- 4 • Clone source code contoh dari Alpine dengan mengetikkan perintah berikut pada *command shell*:

```
5 $ git clone https://github.com/heroku/alpinehelloworld.git
6
7
```

- 9 • Membuat aplikasi Heroku baru dengan mengetikkan perintah berikut pada *command shell*:

```
10 $ heroku create
11
12
```

- 13 • Membangun *image* dan melakukan *deploy* ke *Container Registry* dengan mengetikkan perintah berikut pada *command shell*:

```
14 $ heroku container:push web
15
16
```

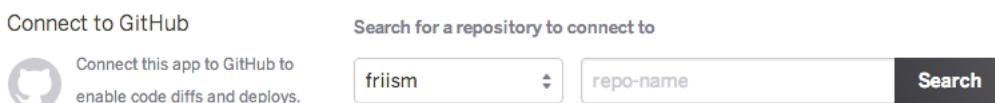
- 18 • Melepaskan *image* ke aplikasi dengan mengetikkan perintah berikut pada *command shell*:

```
19 $ heroku container:release web
20
21
```

- 22 • Membuka aplikasi dengan mengetikkan perintah berikut pada *command shell*:

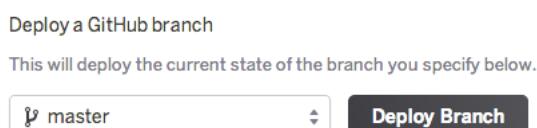
```
23 $ heroku open
24
25
```

### 26 2.2.2.3 Deploy Menggunakan GitHub

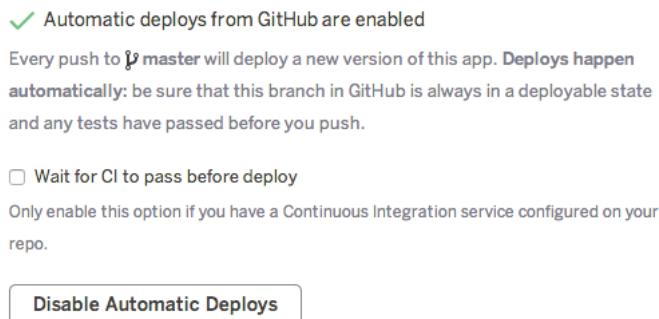


Gambar 2.4: Deploy menggunakan Github Dashboard

27 Deploy dengan cara ini membuat Heroku melakukan *deploy* secara otomatis ke GitHub apabila  
28 *build* berhasil. Deploy ini dapat dilakukan apabila GitHub *integration* telah aktif dan akun GitHub  
29 telah terautentikasi pada Heroku. Autentikasi ini hanya perlu dilakukan satu kali per satu akun  
30 Heroku. Setelah kedua hal tersebut telah siap, sambungkan *repository* GitHub dengan aplikasi  
31 Heroku. (Gambar 2.4).



Gambar 2.5: Deploy menggunakan Github secara manual



Gambar 2.6: *Deploy* menggunakan Github secara otomatis

1     Ada dua cara untuk melakukan deploy, yaitu secara manual dan secara otomatis. Untuk cara  
 2    manual, *deploy* dilakukan dari GitHub (Gambar 2.5). Untuk cara otomatis, *deploy* dilakukan  
 3    dengan mengaktifkan "*Automatic deploys from GitHub*" (Gambar 2.6).

4    **2.2.2.4 Deploy Langsung di situs web Heroku**

5     Pada situs web Heroku, tombol "*Deploy to Heroku*" memungkinkan pengguna untuk melakukan  
 6    deploy aplikasi tanpa meninggalkan situs web Heroku. Cara ini hampir tidak memerlukan konfigurasi.  
 7    Sebelum dapat melakukan *deploy* dengan cara ini, aplikasi harus memiliki *file app.json* yang sah  
 8    pada direktori *root*, dan *source code* aplikasi harus berada di *repository* GitHub.

9     *app.json* adalah *file* berisi deskripsi aplikasi *web*. Isinya dapat berupa *environment variable*,  
 10   *add-ons*, dan informasi lain yang diperlukan untuk menjalankan aplikasi pada Heroku. Heroku  
  11   tidak mewajibkan penulisan informasi tertentu, tapi Heroku merekomendasikan untuk setidaknya  
  12   menuliskan nama aplikasi(*name*), deskripsi aplikasi (*description*), dan logo aplikasi (*logo*). Berikut  
  13   contoh isi dari *app.json* :

```
14
15 {
16   "name": "Node.js Sample",
17   "description": "A barebones Node.js app using Express 4",
18   "repository": "https://github.com/heroku/node-js-sample",
19   "logo": "https://node-js-sample.herokuapp.com/node.png",
20   "keywords": ["node", "express", "static"]
21 }
22
```

23    **2.2.2.5 WAR Deployment**

24     *WAR* (*Web Application Archive*) adalah jenis *file* arsip yang digunakan untuk membungkus  
 25   aplikasi *web*. Dokumen ini dapat berisi halaman *web* statis, *file* XML, dan lain-lain. [7]

26     *WAR deployment* dapat dilakukan menggunakan *plugin* Heroku Java CLI Plugin. Pemasangan  
 27   dan pembaruan *plugin* ini dapat dilakukan dengan mengetikkan perintah berikut pada *command*  
 28   *shell*:

```
29
30 heroku plugins:install java
31
```

32     *Deploy* dapat dilakukan dengan mengetikkan perintah berikut pada *command shell*:

1                   heroku war:deploy <path\_to\_war\_file> --app <app\_name>  
2  
3

4 Keterangan:

- 5         • <path\_to\_war\_file>: letak *file* WAR.  
6         • <app\_name>: nama aplikasi.

7 **2.2.3 Ephemeral Filesystem**

8 Heroku memiliki *ephemeral filesystem*, yang berarti perubahan apapun pada *file* di lingkungan  
9 Heroku tidak akan disimpan secara permanen. Kondisi *file* di Heroku akan selalu dikembalikan  
10 ke *deploy baru*. Perubahan *file* hanya akan bertahan paling lama satu hari. Kondisi ini membuat  
11 aplikasi di Heroku membutuhkan layanan dari pihak ketiga untuk menyimpan hasil unggahan  
12 pengguna aplikasi.

13 **2.2.4 Basis Data dan Manajemen Data**

14 Heroku menyediakan tiga layanan data untuk semua pelanggan :

- 15         • Heroku Postgres  
16         • Heroku Redis  
17         • Apache Kafka

18 Heroku juga menyediakan pilihan lain untuk pengguna Heroku Enterprise, yaitu Heroku Connect.  
19 Selain itu, Heroku juga memungkinkan penggunaan layanan data dari pihak ketiga. Layanan data  
20 dari pihak ketiga ini tersedia sebagai add-ons.

21 **2.2.4.1 Heroku Postgres**

22 Heroku Postgres adalah basis data SQL yang disediakan secara langsung oleh Heroku. Heroku  
23 Postgres dapat diakses oleh bahasa apapun dengan PostgreSQL *driver*. Heroku secara otomatis  
24 membuat satu basis data menggunakan *add-ons* Heroku Postgres setiap aplikasi dibuat. Perintah  
25 berikut dapat diketikkan pada *command shell* apabila ingin menambah basis data baru:  
26

27                   \$ heroku addons:create heroku-postgresql:<PLAN\_NAME>  
28

29 Keterangan :

- 30         • <PLAN\_NAME> : nama *plan* Heroku Postgres yang ingin dipakai. Heroku secara otomatis  
31         menggunakan Heroku Postgres tipe **hobby-dev**.

32 Heroku menawarkan lima *plan* untuk Heroku Postgres, yaitu:

- 33         • Hobby Tier: *plan* dengan toleransi gagal bekerja sampai 4 jam per bulan.  
34         • Standard Tier: *plan* dengan toleransi gagal bekerja sampai 1 jam per bulan.  
35         • Premium Tier: *plan* dengan toleransi gagal bekerja sampai 15 menit per bulan.

- Private Tier: *plan* yang dapat dipilih oleh pengguna Heroku *Enterprise*, memiliki toleransi gagal bekerja sampai 15 menit per bulan.
- Shield Tier : *plan* yang dapat dipilih oleh pengguna Heroku *Enterprise*, memiliki toleransi gagal bekerja sampai 15 menit per bulan. Kelebihannya daripada *Private Tier* adalah basis data ini memiliki keamanan yang lebih ketat.

Gambar 2.7 menunjukkan tabel perbandingan antar *plan*. Di antara lima *plan* Heroku Postgres, hanya *plan Hobby* yang gratis. *Plan* lain memiliki harga yang bervariasi berdasarkan ukuran RAM, batas penyimpanan, dan batas koneksi yang bisa dibuat.

Heroku Postgres tier	Downtime Tolerance	Fork	Follow	Rollback	HA	Disk Encryption
Hobby	< 4 hr downtime per mo.	No	No	No	No	No
Standard	< 1 hr downtime per mo.	Yes	Yes	4 days	No	Yes
Premium	< 15 min downtime per mo.	Yes	Yes	1 week	Yes	Yes
Private	< 15 min downtime per mo.	Yes	Yes	1 week	Yes	Yes
Shield	< 15 min downtime per mo.	Yes	Yes	1 week	Yes	Yes

Gambar 2.7: Tabel plan Heroku Postgres

Semua *plan* memiliki fitur yang sama:

- Dikelola secara otomatis oleh Heroku.
  - Data dijamin tidak akan hilang.
  - Dapat melakukan *backup* basis data secara harian menggunakan Heroku PGBackups.
  - Dapat menggunakan layanan Dataclips (<https://data.heroku.com/dataclips>) untuk berbagi data dan *query*.
  - Akses basis data diproteksi dengan SSL.
  - Menjalankan Postgres versi 9.4, 9.5, 9.6, atau 10 tanpa modifikasi
  - Tersedia ekstensi Postgres
  - Memiliki tampilan *web* yang dapat diakses pada situs web <https://data.heroku.com/>
- Developer juga dapat menambahkan versi yang ingin dipakai dengan cara menambahkan `-version` di belakang perintah tersebut, contoh :

```
$ heroku addons:create heroku-postgresql:<PLAN_NAME>--version=9.5
```

Secara otomatis, Heroku menggunakan versi paling baru dari Heroku Postgres. Saat skripsi ini ditulis, versi terbaru adalah versi 10.

Setelah dipasang, Heroku akan secara otomatis menambahkan *config vars* DATABASE\_URL ke aplikasi. Apabila Heroku Postgres yang dipakai ada lebih dari satu, nama *config vars* akan menjadi HEROKU\_POSTGRESQL\_<COLOR>\_URL dengan <COLOR> adalah nama warna yang dihasilkan secara acak oleh Heroku. Contoh: HEROKU\_POSTGRESQL\_<BLUE>\_URL.

1 Apabila basis data yang digunakan lebih dari satu, basis data utama dapat diatur dengan  
2 mengetikkan perintah berikut pada *command shell*:

```
3  
4 $ heroku pg:promote <database_url>  
5
```

6 Keterangan:

- 7 • <database\_url>: URL dari basis data.

8 Apabila pengguna Heroku ingin berbagi Heroku Postgres kepada banyak aplikasi, pengguna  
9 Heroku dapat mengetikkan perintah berikut pada *command shell* :

```
10  
11 $ heroku addons:attach <originating_app>::DATABASE --app <receiver-app>  
12
```

13 Keterangan:

- 14 • <originating\_app>: nama aplikasi yang memiliki basis data yang ingin dibagi ke aplikasi  
15 lain.

- 16 • <receiver-app>: nama aplikasi yang akan menerima basis data dari aplikasi lain.

17 Pengguna Heroku dapat berhenti berbagi basis data dengan mengetikkan perintah berikut pada  
18 *command shell* :

```
19  
20 $ heroku addons:detach <database_url> --app <application_name>  
21
```

22 Keterangan:

- 23 • <database\_url>: URL dari basis data

- 24 • <application\_name>: nama aplikasinya.

25 Berikut adalah perintah-perintah dasar dari Heroku Postgres yang dapat diketikkan pada  
26 *command shell*:

- 27 • Melihat semua basis data milik aplikasi dan karakteristiknya:

```
28  
29 $ heroku pg:info  
30
```

- 31 • Mengawasi status basis data secara terus menerus:

```
32  
33 $ watch heroku pg:info  
34
```

- 35 • Mengadakan sesi psql dengan basis data:

```
36  
37 $ heroku pg:psql  
38
```

39 atau

```
40  
41 $ heroku pg:psql <database_name>  
42
```

43 Keterangan:

1       – <database\_name> diisi dengan nama basis data atau cukup warna basis data, misalnya  
2       gray).

- 3       • Menarik data dari basis data Heroku Postgres ke basis data di mesin lokal:

4  
5  
6       \$ heroku pg:pull  
7

- 8       • Memasukkan data dari basis data di mesin lokal ke basis data di Heroku Postgres:

9  
10  
11       \$ heroku pg:push <nama\_db\_lokal> <nama\_db\_heroku> --app <nama\_aplikasi>  
12

13       Keterangan:

14       – <nama\_db\_lokal>: nama basis data di lingkungan lokal.  
15       – <nama\_db\_heroku>: nama basis data di lingkungan Heroku.

- 16       • Melihat daftar *query* yang berjalan:

17  
18  
19       \$ heroku pg:ps  
20

- 21       • Menghentikan *query* yang berjalan:

22  
23  
24       \$ heroku pg:kill <procpid>  
25

26       Keterangan:

27       – <procpid>: id *query*, didapat dari melihat daftar *query* yang berjalan.

- 28       • Menghentikan query yang berjalan secara paksa:

29  
30  
31       \$ heroku pg:kill --force <procpid>  
32

33       Keterangan:

34       – <procpid>: id *query*, didapat dari melihat daftar *query* yang berjalan.

- 35       • Menghentikan semua query yang berjalan:

36  
37  
38       \$ heroku pg:killall  
39

- 40       • Menghapus semua data di dalam basis data:

```

1   $ heroku pg:reset <database\_name>
2

```

4 Keterangan:

- 5 • <database\_name> diisi dengan nama basis data atau cukup warna basis data, misalnya gray).

6 Basis data Heroku Postgres dapat diakses secara langsung menggunakan aplikasi pihak ketiga,  
7 misalnya PGAdmin. Informasi yang dibutuhkan untuk dapat terkoneksi basis data menggunakan  
8 aplikasi tersebut didapatkan dengan mengetikkan perintah :

```

9
10  $ heroku pg:credentials DATABASE
11

```

12 atau

```

13
14  $ heroku config | grep HEROKU_POSTGRESQL
15

```

16 Pada saat akan melakukan koneksi ke basis data, pengaturan SSL harus diatur ke `sslmode=require`.

#### 17 2.2.4.2 Heroku Redis

18 Heroku Redis adalah basis data berbasis *key-value store* yang bersifat *in-memory*. Heroku Redis  
19 dapat diakses oleh bahasa apapun dengan *Redis driver*. Cara memasang *add-ons* Heroku Redis  
20 pada *command shell*:

```

21
22  $ heroku addons:create heroku-redis: <PLAN_NAME>
23

```

24 Keterangan :

- 25 • <PLAN\_NAME>: nama *plan* Heroku Redis yang ingin dipakai. Heroku Redis memiliki dua  
26 pilihan *plan*: *Hobby Dev* dan *Premium*. *Hobby Dev* gratis, sedangkan *Premium* berbayar.  
27 Perbedaan kedua *plan* terletak pada jumlah memori dan batas koneksi yang dapat dibuat.

28 Heroku Redis memiliki kelebihan sebagai berikut:

- 29 • Memiliki analisa performa yang dapat membantu menemukan masalah basis data dengan  
30 mudah.
- 31 • Heroku Redis dapat diatur jumlah dan ukurannya sesuai kebutuhan memori dan koneksi.

#### 32 2.2.4.3 Apache Kafka

33 Apache Kafka adalah salah satu *add-ons* di Heroku yang disediakan oleh Kafka yang berintegrasi  
34 penuh dengan Heroku. Apache Kafka dideskripsikan Heroku sebagai *add-ons* yang memungkinkan  
35 *developer* mendistribusikan aplikasi yang dapat menangani jutaan *event* dan miliaran transaksi.  
36 Kafka didesain untuk memindahkan data yang sangat besar dengan reliabilitas yang tinggi dan  
37 toleran akan kerusakan.

38 Pemasangan Python 2.7, node 8.x, .NET Framework, dan Visual C++ Build Tools diperlukan  
39 sebelum memasang Apache Kafka. Pemasangan Apache Kafka dapat dilakukan dengan mengetikkan  
40 perintah berikut pada *command shell*:

```

41
42  $ heroku plugins:install heroku-kafka
43

```

### 2.2.5 Verifikasi Akun

Untuk mengakses beberapa layanan di Heroku, pengguna Heroku harus melakukan verifikasi akun terlebih dahulu. Verifikasi akun dilakukan dengan memasukkan informasi kartu kredit atau kartu debit pengguna. Kartu kredit yang diterima oleh Heroku adalah kartu Visa, MasterCard, American Express, Discover, dan JCB. Kartu debit yang diterima adalah kartu Visa, MasterCard, dan JCB. Kartu lain tidak diterima. Beberapa bank mungkin mensyaratkan penahanan satu dollar oleh pelaku verifikasi sebelum kartu dapat dikonfirmasi.

Berikut beberapa layanan di Heroku yang membutuhkan verifikasi akun:

- Menggunakan lebih dari satu *dyno* di dalam aplikasi.
- Menambah *add-ons* selain Heroku Postgres dan Heroku Connect.
- Mengubah *domain* aplikasi.
- Menerima transfer dari aplikasi yang memiliki sumber daya berbayar.
- Menambah batas standar penggunaan *one-off dyno*.
- Memiliki lebih dari 5 aplikasi dalam satu waktu. Akun yang terverifikasi dapat memiliki sampai 100 aplikasi.

Cara melakukan verifikasi akun Heroku:

- Masuk ke halaman *Account Settings* (<https://dashboard.heroku.com/account>)
- Pilih menu *Billing*
- Tekan tombol *Add Credit Card*

## 2.3 Cron [2]

*Cron* adalah program yang menjalankan *command* (perintah) yang tertera di file *crontab* pada jadwal tertentu di latar belakang secara otomatis. File *crontab* dapat ditemukan di direktori */var/spool/cron/crontabs*. *Crontab* di direktori ini sebaiknya tidak diakses secara langsung, melainkan menggunakan *command crontab*. File *crontab* juga dapat ditemukan di direktori */etc* atau subdirektori */etc*. File *crontab* di direktori tersebut adalah *file system crontab*.

Setiap *command* di *crontab* diawali dengan lima penanda waktu dan diikuti dengan nama *user* (jika berada di *file system crontab*). Lima penanda waktu tersebut secara berurutan adalah:

- *minute*: menandakan tiap menit berapa *command* dijalankan. *Value* yang sah adalah angka 0-59 atau tanda *asterisk* (\*).
- *hour*: menandakan tiap jam berapa *command* dijalankan. *Value* yang sah adalah angka 0-23 atau tanda *asterisk* (\*).
- *day of the month*: menandakan tiap tanggal berapa *command* dijalankan. *Value* yang sah adalah angka 1-31 atau tanda *asterisk* (\*).

- 1     • *month*: menandakan tiap bulan berapa *command* dijalankan. *Value* yang sah adalah angka  
2       1-12 atau tanda *asterisk* (\*).

- 3     • *day of the week*: menandakan tiap hari apa *command* dijalankan. *Value* yang sah adalah  
4       angka 0-7 atau tanda *asterisk* (\*) atau "Sun". Hari minggu dapat ditulis dengan angka 0 atau  
5       7 atau "Sun".

6     Contoh penulisan *command* di *crontab*:

7     

```
0 12 * * * /home/myscripts/lunch.sh
```

10   Pada contoh di atas, *cron* akan menjalankan *script* *lunch.sh* pada direktori */home/myscripts*  
11   setiap hari pada pukul 12.00. Tanda *asterisk* menandakan eksekusi dilakukan dari *range value*  
12   terendah sampai *value* tertinggi.

13   Penanda waktu boleh memiliki *value* lebih dari satu. Penanda waktu yang memiliki *value* lebih  
14   dari satu ditulis dalam bentuk *range*, *list*, atau *step*. *Range* ditulis dalam format dua *value* yang  
15   dipisahkan dengan tanda hubung (-). Range bersifat inklusif. Contoh: 8-11 untuk *hour* berarti  
16   *command* akan dieksekusi pada pukul 8, 9, 10, dan 11. *List* ditulis dalam format dua atau lebih  
17   *value* yang dipisahkan dengan koma. Contoh: 1,5,7. *Range* dan *list* dapat disatukan. Contoh:  
18   1,2,8-12. *Step* ditulis dalam format: *range* / <number>. Contoh : '0-23/2' untuk *hour*. Tanda  
19   tersebut berarti eksekusi dijalankan setiap dua jam sekali. Tanda *asterisk* juga dapat digunakan  
20   untuk menggantikan *range*. Contoh: \*/4 untuk *hour*. Tanda tersebut berarti eksekusi dijalankan  
21   setiap empat jam sekali.

22   Nama juga dapat digunakan untuk penanda waktu *month* atau *day of week*. Cukup dengan  
23   menggunakan tiga huruf pertama dari kata bahasa Inggrisnya tanpa menghiraukan huruf besar  
24   atau huruf kecil. Namun, nama tidak bisa dipakai untuk *range* atau *list*.

25   Selain itu, kelima penanda waktu dapat diganti dengan salah satu "*nickname*" berikut :

- 26     • @reboot: Eksekusi sekali saat *startup*.
- 27     • @yearly: Eksekusi setiap tahun (setiap tanggal 1 Januari), "0 0 1 1 \*"
- 28     • @anually: Sama seperti @yearly.
- 29     • @monthly: Eksekusi setiap bulan, "0 0 1 \* \*".
- 30     • @weekly: Eksekusi setiap minggu, "0 0 \* \* 0".
- 31     • @daily: Eksekusi setiap hari (pukul 00.00), "0 0 \* \* \*"
- 32     • @midnight: Sama seperti @daily.
- 33     • @hourly: Eksekusi setiap jam, "0 \* \* \* \*"

34   Baris kosong, spasi di awal baris, dan *tab* akan diabaikan oleh *cron*. Baris yang karakter  
35   bukan spasi pertamanya adalah # akan dianggap sebagai *comment* (komentar) dan akan diabaikan.  
36   *Comment* tidak boleh diletakkan di baris yang sama dengan *command* karena *comment* tersebut  
37   akan dianggap sebagai bagian dari *command*.

## 1    2.4 Gmail API [3]

2        Gmail adalah layanan *email* yang disediakan oleh perusahaan Google LLC. Gmail API dapat  
3        digunakan untuk mengakses *email* Gmail.

4    2.4.1 Resource

5        Gmail API menyediakan beberapa jenis *resource* :

- 6        • *Message*

7        *Message* merepresentasikan pesan dalam email. *Message* hanya bisa dibuat atau dihapus.  
8        Tidak ada properti dari *message* yang bisa diubah selain label yang diberikan ke message.

- 9        • *Label*

10       *Label* berfungsi sebagai sarana utama untuk mengelompokkan dan mengatur *message* dan  
11       *thread*. *Label* mempunyai hubungan banyak ke banyak dengan *message*. Artinya, satu *message*  
12       dapat memiliki beberapa *label* dan satu *label* dapat diberikan ke beberapa *message*.

13       *Label* dikelompokkan ke dalam dua kelompok: *label* sistem dan *label* pengguna. Contoh *label*  
14       sistem adalah *label* INBOX, TRASH, dan SPAM. *Label* sistem dibuat secara internal dan tidak  
15       dapat dibuat, dihapus, dan dimodifikasi. Namun, beberapa *label* sistem dapat diberikan  
16       ke *message* atau dilepaskan dari *message*. *Label* pengguna dapat ditambah, dihapus, dan  
17       dimodifikasi oleh pengguna atau aplikasi.

- 18        • *Draft*

19       *Draft* merepresentasikan *message* yang belum dikirim. *Message* tidak bisa dimodifikasi setelah  
20       dibuat, tapi *message* yang terdapat di dalam *draft* dapat dimodifikasi. Mengirimkan *draft*  
21       secara otomatis akan menghapus *draft* tersebut dan membuatnya menjadi *message* dengan  
22       *label* sistem SENT.

- 23        • *History*

24       *History* adalah riwayat modifikasi *message* yang diurutkan secara kronologis. *History* hanya  
25       menyimpan perubahan dalam jangka waktu 30 hari.

- 26        • *Thread*

27       *Thread* adalah kumpulan *message* yang merepresentasikan percakapan. *Thread* dapat memiliki  
28       *label*. *Thread* tidak dapat dibuat, tapi dapat dihapus. *Message* dapat dimasukkan ke *Thread*.

- 29        • *Setting*

30       *Setting* mengontrol perilaku fitur pada Gmail kepada penggunanya. *Setting* tersedia untuk  
31       akses POP dan IMAP, *forward email*, *filter*, *vacation auto-response*, *send-as aliases*, *signatures*,  
32       dan *delegates*.

### 1    2.4.2 Scope

2    Gmail API menggunakan OAuth 2.0 untuk menangani autentikasi dan *authorization*. Untuk  
3    menggunakan Gmail API, aplikasi harus menyebutkan *scope* yang dipakai di aplikasi. *Scope* adalah  
4    *string* yang mengidentifikasi *resource* yang ingin di akses. *Scope* ini digunakan bersama dengan  
5    token untuk mengamankan akses ke *resource* pengguna. *Token* tersebut memiliki masa kadaluarsa.  
6    Contoh *scope*:

- 7       • <https://www.googleapis.com/auth/gmail.readonly>: *scope* untuk membaca *message* dari  
8       Gmail.
- 9       • <https://www.googleapis.com/auth/gmail.modify>: *scope* untuk mengubah *label* pada *thread*  
10      atau *message*.
- 11      • <https://www.googleapis.com/auth/gmail.compose>: *scope* untuk mengirim *message* mewakili  
12      pengguna.

### 13    2.4.3 Penggunaan pada umumnya

#### 14    2.4.3.1 Mengirim *message*

- 15      1. Membuat konten *email*
- 16      2. Membuat *string* yang dikodekan berdasarkan *base64url* dari konten
- 17      3. Membuat *resource message* dan memasukkan *string* tersebut ke properti *raw*
- 18      4. Memanggil *message.send* untuk mengirim *message*

#### 19    2.4.3.2 Mengambil *email* yang diterima

20      Pada Gmail API, mengambil *email* yang diterima dapat dilakukan dengan melakukan GET  
21      *request* ke "https://www.googleapis.com/gmail/v1/users/*userId*/messages/*id*" dengan *userId*  
22      adalah alamat *email* pengguna dan *id* adalah ID *email*. Saat mengambil message, format dari  
23      respon dapat diatur. Format FULL mengembalikan seluruh informasi dari message. Format MINIMAL  
24      hanya mengembalikan metadata seperti label. Format RAW mengembalikan properti *raw* saja. Secara  
25      otomatis, format dari respon memakai format FULL.

#### 26    2.4.3.3 Perubahan di *history*

27      Perubahan *message* direpresentasikan oleh History objects. Properti *start\_history\_id*  
28      memperbolehkan aplikasi mengatur dari titik mana perubahan ingin dikembalikan. Beberapa  
29      perubahan dapat mempengaruhi lebih dari satu *message*, sehingga *history* yang merepresentasikan  
30      perubahan tersebut akan berisi beberapa *message*.

#### 31    2.4.3.4 Manajemen *Label*

32      *Label* yang diberikan ke sebuah *thread* juga diberikan ke semua *message* di dalam *thread*. Jika  
33      sebuah *label* dihapus, *label* tersebut akan dihapus dari semua *thread* dan *message* yang memiliki *label*

1 tersebut. Properti `messageListVisibility` digunakan untuk menentukan apakah `message` dengan  
2 `label` tersebut ada di `message list`. Properti `labelListVisibility` digunakan untuk menentukan  
3 apakah ada `label` tersebut di daftar `label`. Untuk mengubah `label`, gunakan `messages.modify`  
4 dan `threads.modify`.

#### 5 2.4.4 Implementasi Otorisasi dari Sisi Server

6 Setiap `request` ke Gmail API harus menggunakan OAuth 2.0. *Developer* perlu menggunakan alur  
7 dari sisi *server* ketika aplikasinya membutuhkan akses Google API mewakili *user*. Pendekatan ini  
8 membutuhkan *access token* dan *refresh token* untuk *server*. Untuk mulai menggunakan Gmail API,  
9 *developer* harus mendapatkan *client id* dan *client secret* terlebih dahulu. *Client id* dan *client secret*  
10 dapat dimiliki apabila *developer* telah membuat *project* di Google API *Console* dan menyalakan  
11 Gmail API.

12 Ketika *user* membuka aplikasi untuk pertama kalinya, sebuah *dialog* akan muncul dan menanyakan  
13 izin dari *user* agar aplikasi boleh mengakses akun Gmail miliknya. *Dialog* tersebut akan menyatakan  
14 *scope* yang dipakai aplikasi. Setelah *user* mengizinkan, *dialog* tersebut tidak akan muncul lagi,  
15 kecuali *scope* aplikasi diubah.

16 Setelah *user sign-in* untuk pertama kalinya, *authorization result object* akan dikembalikan ke  
17 *server*. *Object* tersebut berisi *authorization code*. *Authorization code* adalah *code* sekali pakai yang  
18 dapat ditukar dengan *access token*. *Access token* ini akan diberikan ke Gmail API agar aplikasi  
19 diberi izin untuk mengakses ke data user pada waktu yang terbatas. Selain *access token*, *server*  
20 juga mendapatkan *refresh token*. *Refresh token* ini dapat digunakan untuk menerima *access token*  
21 baru setelah *token* yang lama kadaluarsa. *Refresh token* ini harus disimpan di suatu tempat agar  
22 bisa dipakai. Jika *refresh token* ini tidak disimpan, maka aplikasi harus mengirim *request* dengan  
23 *query approval\_prompt* yang diset ke *force*. Ini dapat mengakibatkan *user* mendapat dialog untuk  
24 meminta izin lagi.

25 *Refresh token* dapat kadaluarsa. *Refresh token* kadaluarsa jika :

- 26 • User mencabut izinnya.
- 27 • *Refresh token* sudah tidak digunakan selama enam bulan.
- 28 • User mengganti *password* dan *refresh token* berisi Gmail *scopes*.
- 29 • Akun *user* telah melebihi batas maksimal dari *refresh token* yang diizinkan. Batas maksimalnya  
30 adalah 50 *refresh token* per akun per klien. Jika batas ini dilampaui, membuat *refresh token*  
31 baru akan menggugurkan *refresh token* yang lama tanpa peringatan. Batas ini tidak berlaku  
32 untuk *service account*.

### 33 2.5 PHP IMAP [4]

34 *IMAP* (*Internet Message Access Protocol*) adalah metode untuk mengakses pesan elektronik  
35 yang disimpan di sebuah *mail server*.

36 Ekstensi ini dapat digunakan apabila *c-client library* sudah terpasang. Library ini dapat  
37 ditemukan di <https://www.washington.edu/imap/>. Dokumen IMAP tidak boleh diletakkan

1 langsung ke dalam direktori *system*, karena dapat memicu konflik. Sebaiknya membuat direktori baru  
 2 di dalam direktori *system*, lalu masukkan file IMAP ke dalamnya. Contoh : /usr/local/imap-2000b.  
 3 Di dalam direktori baru tambahkan direktori lagi bernama **lib/** dan **include/**. Semua file dengan  
 4 ekstensi .c dimasukkan ke direktori **lib/**. Saat IMAP dikompilasi, file bernama **c-client.a** akan  
 5 terbentuk. Dokumen tersebut juga diletakkan di direktori **lib/**.

6 Setelah itu, kompilasi PHP dengan **-with-imap[=DIR]** dengan DIR adalah tempat c-client.  
 7 Contoh: **with-imap=/usr/local/imap-2000b**. Pengguna sistem operasi Windows mungkin harus  
 8 mengaktifkan **php\_imap.dll**.

9 IMAP tidak didukung pada sistem operasi Windows yang versinya lebih lama dari Windows  
 10 2000. Hal ini karena IMAP menggunakan fungsi enkripsi agar koneksi lewat SSL ke *mail server*  
 11 aktif.

12 Di dalam sistem operasi Ubuntu, pemasangan PHP IMAP bisa dilakukan dengan mudah.

```
14
15 // Pasang libc-client-dev
16 $ sudo apt-get install libc-client-dev
17
18 // Pasang PHP<versi> imap:
19 // sudo apt-get install php<versi>-imap
20 // Contoh :
21 sudo apt-get install php5-imap
22
```

24 Berikut adalah *function* dasar dari imap:

25 • **imap\_alerts**

26 – Deskripsi: Fungsi ini mengembalikan semua *alert message* yang telah terjadi. Ketika  
 27 fungsi ini dipanggil, semua *alert message* yang ada di *stack* dihapus.  
 28 – *Parameter*: Tidak ada.  
 29 – *Return values*: Mengembalikan *array* yang berisi semua *alert message* yang dihasilkan  
 30 atau FALSE jika tidak ada satupun *alert message*.

31 • **imap\_close**

32 – Deskripsi: Fungsi ini berfungsi untuk menutup *IMAP stream*.

33 – *Parameter*:

34 \* **imap\_stream**: *IMAP stream* yang dikembalikan oleh **imap\_open**.  
 35 \* **flag**: Jika diatur ke CL\_EXPUNGE, fungsi akan secara diam-diam menghapus  
 36 semua pesan yang ditandai untuk dihapus sebelum menutup *IMAP stream*.  
 37 – *Return values*: Mengembalikan TRUE jika sukses atau FALSE jika gagal.

38 • **imap\_errors**

39 – Deskripsi: Fungsi ini mengembalikan semua *error* yang telah terjadi. Ketika fungsi ini  
 40 dipanggil, semua *error* yang ada di *stack* dihapus.

- 1       – *Parameter*: Tidak ada.
- 2       – *Return values*: Mengembalikan *array* yang berisi semua *error* yang dihasilkan atau  
3        FALSE jika tidak ada satupun *error*.
- 4     ● `imap_fetch_overview`
- 5       – Deskripsi: Fungsi ini mengambil *mail header* berdasarkan urutan yang diberikan dan  
6        mengembalikan ikhtisar kontennya.
- 7       – *Parameter* :
- 8           \* `imap_stream`: *IMAP stream* yang dikembalikan oleh `imap_open`.
- 9           \* `sequence`: Deskripsi cara pengurutan message. Cara menyebutkan urutan bisa  
10        menggunakan sintaks X,Y atau mengambil semua dalam interval dengan sintaks  
11        X:Y.
- 12       \* `options`: *sequence* akan berisi *message index* atau UID, jika parameter ini diatur ke  
13        `FT_UID`.
- 14       – *Return values*: Mengembalikan *array of objects*. Tiap *object* mendeskripsikan satu  
15        *message header*. *Object* berisi macam-macam *property*. *Object* hanya akan menyebutkan  
16        sebuah *property* jika *property* tersebut memang ada. *Property* yang mungkin adalah :
- 17           \* `subject`: Subjek pesan
- 18           \* `from`: Pengirim
- 19           \* `to`: Penerima
- 20           \* `date`: Tanggal pengiriman
- 21           \* `message_id`: *Message id*
- 22           \* `references`: *Message id* yang behubungan
- 23           \* `in_reply_to`: *Message id* untuk membalas
- 24           \* `size`: ukuran dalam *bytes*
- 25           \* `uid`: UID yang dimiliki di dalam *mailbox*
- 26           \* `msgno`: urutan message di dalam *mailbox*
- 27           \* `recent`: menandakan bahwa *message* ini adalah *message* yang baru-baru ini diterima
- 28           \* `flagged`: menandakan bahwa *message* ini adalah *message* yang ditandai
- 29           \* `answered`: menandakan bahwa *message* ini adalah *message* yang ditandai sebagai  
30        telah dijawab
- 31           \* `deleted`: menandakan bahwa *message* ini adalah *message* yang ditandai untuk dihapus
- 32           \* `seen`: menandakan bahwa *message* ini adalah *message* yang sudah dibaca
- 33           \* `draft`: menandakan bahwa *message* ini adalah *message* yang ditandai sebagai *draft*
- 34           \* `update`: UNIX *timestamp* dari tanggal kedatangan pesan
- 35     ● `imap_fetchbody`
- 36       – Deskripsi: Mengambil bagian tertentu dari *body* dari *message* yang disebutkan. Bagian  
37        dari *body* tidak di-*decode* oleh fungsi ini.

1       – *Parameter:*

- 2           \* imap\_stream: *IMAP stream* yang dikembalikan oleh imap\_open.
- 3           \* msg\_number: nomor *message*
- 4           \* section: Nomor bagian. Ini adalah serangkaian bilangan bulat yang dibatasi oleh
- 5           periode yang diindeks ke daftar bagian *body* sesuai spesifikasi IMAP4.
- 6           \* options: *bitmask* dengan satu atau lebih dari :
- 7              · FT\_UID: msg\_number adalah UID
- 8              · FT\_PEEK: Jangan memberikan *seen flag* jika belum diberikan.
- 9              · FT\_INTERNAL: Mengembalikan *string* di dalam format internal, tidak akan
- 10             dikanonikkan ke CRLF.

11       – *Return values:* Mengembalikan bagian tertentu dari *body message* yang disebutkan

12       sebagai *text string*.

13       • imap\_fetchheader

14       – Deskripsi: Fungsi ini mengambil *header* yang lengkap dan tidak terfilter (RFC2822

15       format) dari *message*.

16       – *Parameter :*

- 17           \* imap\_stream: *IMAP stream* yang dikembalikan oleh imap\_open.
- 18           \* msg\_number: nomor *message*
- 19           \* options: pilihan yang mungkin adalah :
- 20              · FT\_UID: msgno argument adalah UID
- 21              · FT\_INTERNAL: Mengembalikan *string* di dalam format internal, tidak akan
- 22             dikanonikkan ke CRLF.
- 23              · FT\_PREFETCHTEXT: RFC822.TEXT harus diambil sebelumnya pada saat
- 24             yang sama. Ini menghindari RTT tambahan pada koneksi IMAP jika teks pesan
- 25             lengkap diinginkan.

26       – *Return values:* Mengembalikan header dari message yang disebutkan sebagai text string.

27       • imap\_fetchstructure

28       – Deskripsi: Mengambil semua informasi terstruktur untuk message yang diberikan.

29       – *Parameter :*

- 30           \* imap\_stream: *IMAP stream* yang dikembalikan oleh imap\_open.
- 31           \* msg\_number: nomor *message*
- 32           \* options: parameter opsional ini hanya memiliki satu opsi, FT\_UID, yang memberitahu
- 33             fungsi untuk memperlakukan msg\_number argument sebagai UID.

34       – *Return values:* Mengembalikan sebuah *object* termasuk *envelope*, *internal date*, *size*,

35       *flags* dan *body structure* serta *object* serupa untuk tiap *mime attachment*. Struktur dari

36       *object* adalah sebagai berikut :

- 37           \* type: *Primary body type*

```
1      * encoding: Body transfer encoding
2      * ifsubtype: TRUE jika ada subtype string
3      * subtype: MIME subtype
4      * ifdescription: TRUE jika ada description string
5      * description: Content description string
6      * ifid: TRUE jika ada identification string
7      * id: Identification string
8      * lines: Jumlah lines
9      * bytes: Jumlah bytes
10     * ifdisposition: TRUE jika ada disposition string
11     * disposition: Disposition string
12     * ifdparameters: TRUE jika dparameters array tersedia
13     * dparameters: Array of objects dimana tiap object memiliki "attribute" dan "value"
14       property berdasarkan parameter pada Content-disposition MIME header.
15     * ifparameters: TRUE jika parameters array tersedia
16     * parameters: Array of objects dimana tiap object memiliki "attribute" dan "value"
17       property.
18     * parts: Array of objects identik dalam structure dengan top-level object, masing-masing
19       berdasarkan pada MIME body part.
```

- 20 • imap\_headerinfo
  - 21 – Deskripsi: Fungsi ini berfungsi untuk mendapatkan informasi dari *message number* yang  
22 diberikan dengan membaca *header*.
    - 23 – *Parameter* :
      - 24 \* imap\_stream: *IMAP stream* yang dikembalikan oleh imap\_open.
      - 25 \* msg\_number: nomor *message*
      - 26 \* fromlength: jumlah karakter untuk *fetchfrom property*. Harus lebih besar atau sama  
27 dengan nol.
      - 28 \* subjectlength: jumlah karakter untuk *fetchsubject property*. Harus lebih besar atau  
29 sama dengan nol.
      - 30 \* defaulthost
    - 31 – *Return values*: Mengembalikan FALSE jika terjadi *error*. Jika sukses, mengembalikan  
32 informasi di dalam *object* dengan *property* berikut :
      - 33 \* toaddress: *full "to" : line*, sampai dengan 1024 karakter.
      - 34 \* to: *array of objects* dari *To: line*, dengan *property* berikut: *personal, adl, mailbox,*  
35 dan *host*
      - 36 \* fromaddress: *full "from" : line*, sampai dengan 1024 karakter.
      - 37 \* from: *array of objects* dari *From: line*, dengan *property* berikut: *personal, adl,*  
38 *mailbox*, dan *host*.

```
1      * ccaddress: full "cc": line, sampai dengan 1024 karakter.  
2      * cc: array of objects dari Cc: line, dengan property berikut: personal, adl, mailbox,  
3          dan host.  
4      * bccaddress: full "bcc" : line, sampai dengan 1024 karakter.  
5      * bcc: array of objects dari Bcc: line, dengan property berikut: personal, adl, mailbox,  
6          dan host.  
7      * reply_toaddress: full "Reply-To" : line, sampai dengan 1024 karakter.  
8      * reply_to: array of objects dari Reply-To: line, dengan property berikut: personal,  
9          adl, mailbox, dan host.  
10     * senderaddress: full "sender" : line, sampai dengan 1024 karakter.  
11     * sender: array of objects dari Sender: line, dengan property berikut: personal, adl,  
12          mailbox, dan host.  
13     * return_pathaddress: full "Return-Path": line, sampai dengan 1024 karakter.  
14     * return_path: array of objects dari Return-Path: line, dengan property berikut:  
15          personal, adl, mailbox, dan host.  
16     * remail  
17         * date: tanggal message yang ditemukan di header  
18         * Date: sama dengan date  
19         * subject: subyek pesan  
20         * Subject: sama dengan subject  
21         * in_reply_to  
22         * message_id  
23         * newsgroups  
24         * followup_to  
25         * references  
26         * Recent: R jika diterima baru-baru ini dan telah dilihat, N jika diterima baru-baru  
27             ini dan belum pernah dilihat, dan ' ' jika tidak diterima baru-baru ini.  
28         * Unseen: U jika belum pernah dilihat dan tidak diterima baru-baru ini, ' ' jika telah  
29             dilihat atau belum pernah dilihat dan diterima baru-baru ini.  
30         * Flagged: F jika telah ditandai, ' ' jika tidak ditandai.  
31         * Answered: A jika telah dijawab, ' ' jika belum dijawab.  
32         * Deleted: D jika telah dihapus, ' ' jika belum dihapus.  
33         * Draft: X jika message adalah sebuah draft, ' ' jika bukan draft.  
34         * Msgno: nomor message  
35         * MailDate  
36         * Size: ukuran message  
37         * update: tanggal message diterima dalam format waktu Unix.  
38         * fetchfrom: from line diformat untuk memenuhi fromlength characters.  
39         * fetchsubject: subject line diformat untuk memenuhi subjectlength characters.
```

- 1     ● `imap_last_error`
  - 2       – Deskripsi: Fungsi ini berfungsi untuk mengembalikan *full text* dari IMAP *error message*  
3       terakhir yang terjadi pada *page* sekarang. *Error stack* tidak diganggu-gugat.
  - 4       – *Parameter*: Tidak ada
  - 5       – *Return values*: Mengembalikan *full text* dari IMAP *error message* terakhir yang terjadi  
6       pada *page* sekarang. Mengembalikan FALSE jika tidak ada *error message*.
- 7     ● `imap_open`
  - 8       – Deskripsi: Fungsi ini berfungsi untuk membuka *IMAP stream* ke sebuah *mailbox*. Fungsi  
9       ini dapat juga digunakan untuk membuka *stream* ke POP3 dan NNTP *server*, tapi  
10      beberapa fungsi dan fitur hanya tersedia pada *IMAP server*.
  - 11       – *Parameter* :
    - 12           \* *mailbox*: nama *mailbox* yang terdiri dari *server* dan *mailbox path* pada server ini.
    - 13           \* *username*
    - 14           \* *password*
    - 15           \* *options*: *bit mask* dengan satu atau lebih dari berikut:
      - 16             · `OP_READONLY`: Membuka *mailbox*, *read-only*
      - 17             · `OP_ANONYMOUS`: Tidak menggunakan atau memperbarui `.newsrc` untuk  
18             *news* (hanya NNTP)
      - 19             · `OP_HALFOPEN`: Untuk nama IMAP dan NNTP, membuka koneksi tapi tidak  
20             membuka *mailbox*
      - 21             · `OP_EXPUNGE`: Menghapus pesan yang ditandai untuk dihapus sebelum menutup  
22             *mailbox*
      - 23             · `OP_DEBUG`: *Debug protocol negotiations*
      - 24             · `OP_SHORTCACHE`: *Short (elt-only) caching*
      - 25             · `OP_SILENT`: Jangan mengoper *events* (penggunaan internal)
      - 26             · `OP_PROTOTYPE`: Mengembalikan *driver prototype*
      - 27             · `OP_SECURE`: Jangan melakukan autentikasi yang tidak aman
    - 28           \* *n\_retries*: Jumlah maksimum percobaan untuk terkoneksi
    - 29           \* *params*: parameter koneksi, *string/key* berikut mungkin dapat digunakan untuk  
30             mengatur satu atau lebih parameter koneksi: `DISABLE_AUTHENTICATOR`  
31             (Menonaktifkan *property* autentikasi).
  - 32       – *Return values*: Mengembalikan *IMAP stream* jika berhasil dan FALSE jika terjadi *error*.
- 33     ● `imap_qprint`
  - 34       – Deskripsi: Menkonversi *quoted-printable string* ke dalam *8 bit string* berdasarkan  
35       RFC2045, section 6.7.
  - 36       – *Parameter*: *quoted-printable string*
  - 37       – *Return values*: *8 bit string*

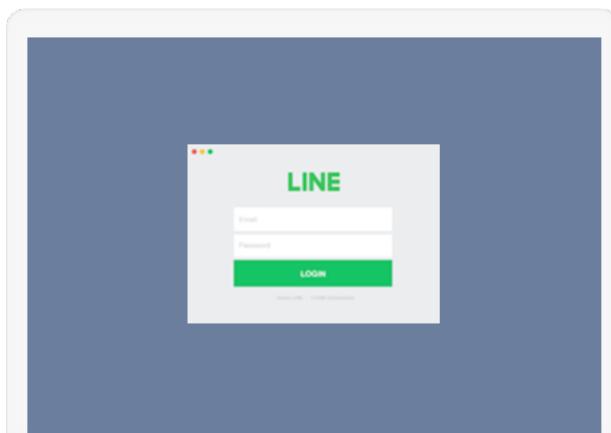
- 1     • imap\_search
  - 2       – Deskripsi: Fungsi ini berfungsi untuk melakukan pencarian pada *mailbox* yang sedang  
3       terbuka pada *IMAP stream* yang diberikan.
  - 4       – *Parameter:*
    - 5           \* imap\_stream: *IMAP stream* yang dikembalikan oleh imap\_open.
    - 6           \* criteria: *string* yang dibatasi dengan spasi, di dalamnya berisi satu atau lebih kata  
7           kunci berikut:
      - 8              · ALL: mengembalikan semua *message* yang sesuai dengan kriteria lainnya.
      - 9              · ANSWERED: mengembalikan *message* yang bertanda ANSWERED
      - 10             · BCC "string": mengembalikan *message* dengan "string" di dalam *Bcc: field*
      - 11             · BEFORE "date": mengembalikan *message* dengan *Date:* sebelum "date"
      - 12             · BODY "string": mengembalikan *message* dengan "string" di dalam *body message*
      - 13             · CC "string": mengembalikan *message* dengan "string" di dalam *Cc: field*
      - 14             · DELETED: mengembalikan *message* yang dihapus
      - 15             · FLAGGED: mengembalikan *message* yang bertanda FLAGGED (terkadang  
16             merujuk ke *message* bertanda *Important* atau *Urgent*)
      - 17             · FROM "string": mengembalikan *message* dengan "string" di dalam *From: field*
      - 18             · KEYWORD "string": mengembalikan *message* dengan "string" sebagai *keyword*
      - 19             · NEW: mengembalikan *message* baru
      - 20             · OLD: mengembalikan *message* lama
      - 21             · ON "date": mengembalikan *message* dengan *Date:* cocok dengan "date"
      - 22             · RECENT: mengembalikan *message* yang baru-baru ini diterima atau yang  
23             bertanda RECENT
      - 24             · SEEN: mengembalikan *message* yang telah dilihat atau yang bertanda SEEN
      - 25             · SINCE "date": mengembalikan *message* dengan *Date:* sejak "date"
      - 26             · SUBJECT "string": mengembalikan *message* dengan "string" di dalam *Subject*
      - 27             · TEXT "string": mengembalikan *message* dengan text "string"
      - 28             · TO "string": mengembalikan *message* dengan "string" di dalam *To:*
      - 29             · UNANSWERED: mengembalikan *message* yang belum dijawab
      - 30             · UNDELETED: mengembalikan *message* yang belum dihapus
      - 31             · UNFLAGGED: mengembalikan *message* yang belum ada tandanya
      - 32             · UNKEYWORD "string": mengembalikan *message* yang tidak memiliki *keyword*  
33             "string"
      - 34             · UNSEEN: mengembalikan *message* yang belum pernah dilihat
    - 35           \* options: *Values* sah untuk *options* adalah SE\_UID, yang menyebabkan *array* yang  
36             dikembalikan berisi UID, bukan *message sequence number*.
    - 37           \* charset: *MIME character set* untuk digunakan saat mencari *strings*.
  - 38       – *Return values:* Mengembalikan *array* dari *message number* atau UID.

## 1 **2.6 LINE [5]**

2 LINE adalah aplikasi pengirim pesan yang tersedia dalam *platform android, ios, dan desktop*.  
3 LINE memiliki beberapa produk yang dapat digunakan *developer* aplikasi. Produk-produk tersebut  
4 adalah:

- 5 1. LINE Login  
6 2. LINE Bot Designer  
7 3. Clova  
8 4. LINE Pay  
9 5. Messaging API

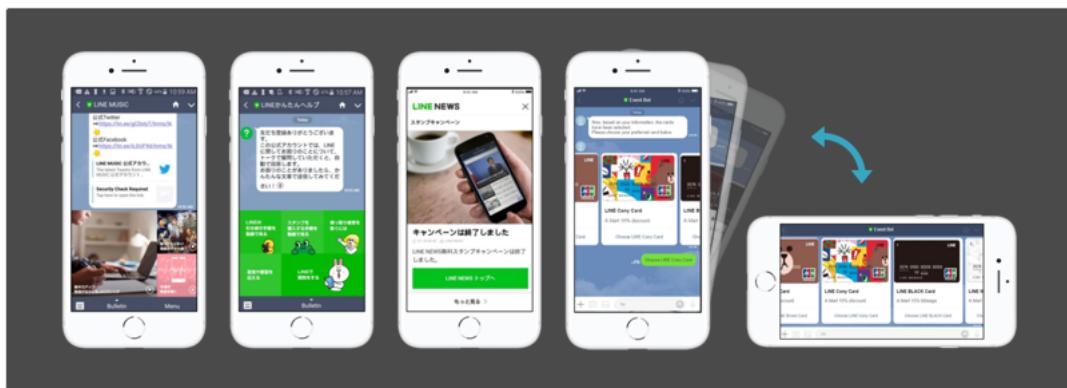
### 10 **2.6.1 LINE Login**



Gambar 2.8: LINE Login

11 LINE Login adalah produk dari LINE yang memungkinkan *developer* membuat aplikasinya  
12 menyediakan pilihan *login* melalui akun LINE. Pengguna aplikasi yang dibuat *developer* tidak  
13 perlu mendaftar menggunakan *email* dan *password*. *Login* menjadi lebih mudah dan cepat. LINE  
14 menyediakan LINE SDK untuk mengintegrasikan LINE Login dengan *native apps*.

### 1 2.6.2 LINE Bot Designer



Gambar 2.9: LINE Bot Designer

2 LINE Bot Designer (Gambar 2.9) adalah produk LINE yang memungkinkan *developer* membuat  
 3 prototipe LINE bot lebih cepat dan lebih mudah tanpa mengetahui pemrograman. Dengan produk  
 4 ini, *developer* dapat mendesain *chatbots* sesuai skenario yang diinginkan.

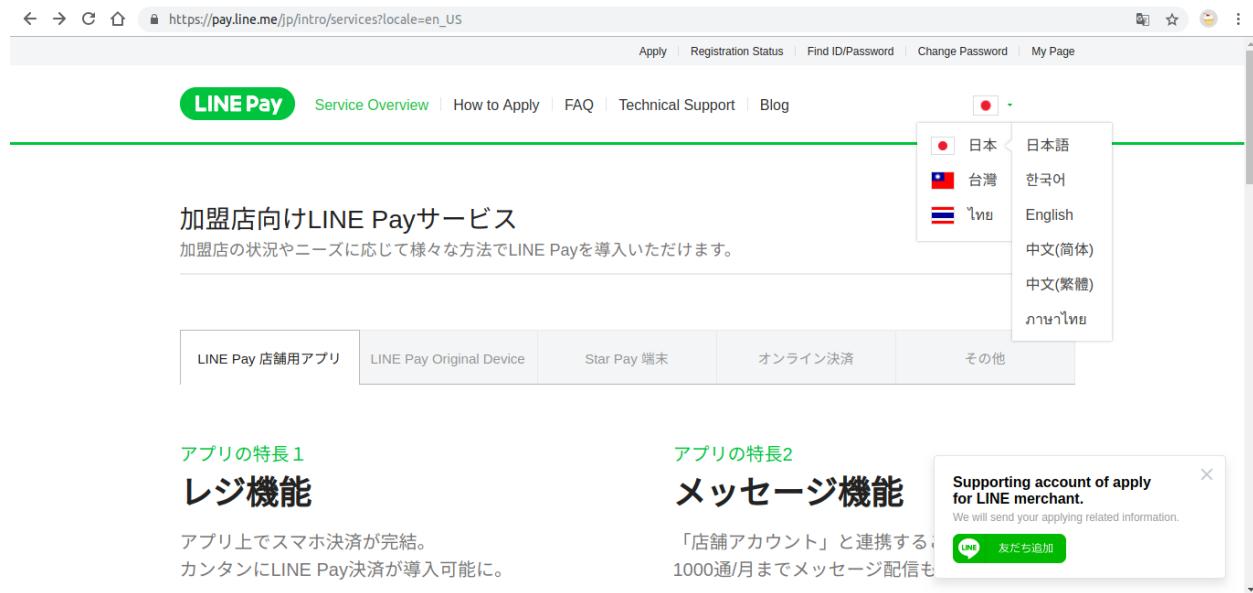
### 5 2.6.3 Clova



Gambar 2.10: Situs web Clova (<https://clova-developers.line.me>)

6 Clova adalah sebuah *AI Assistant* (aplikasi dengan kecerdasan buatan yang berfungsi sebagai  
 7 asisten) yang dipasang di dalam *Clova Wave* dan *Clova Friends*. Pada saat skripsi ini dibuat,  
 8 Clova masih dalam tahap pengembangan dan tersedia dalam versi beta. Tidak ada dokumentasi  
 9 resmi untuk produk ini, tapi ada situs web resminya: <https://clova-developers.line.me>  
 10 (Gambar 2.10). Pada saat skripsi ini ditulis, situs web ini hanya tersedia dalam bahasa Jepang  
 11 sehingga membutuhkan penerjemah untuk memahami isinya.

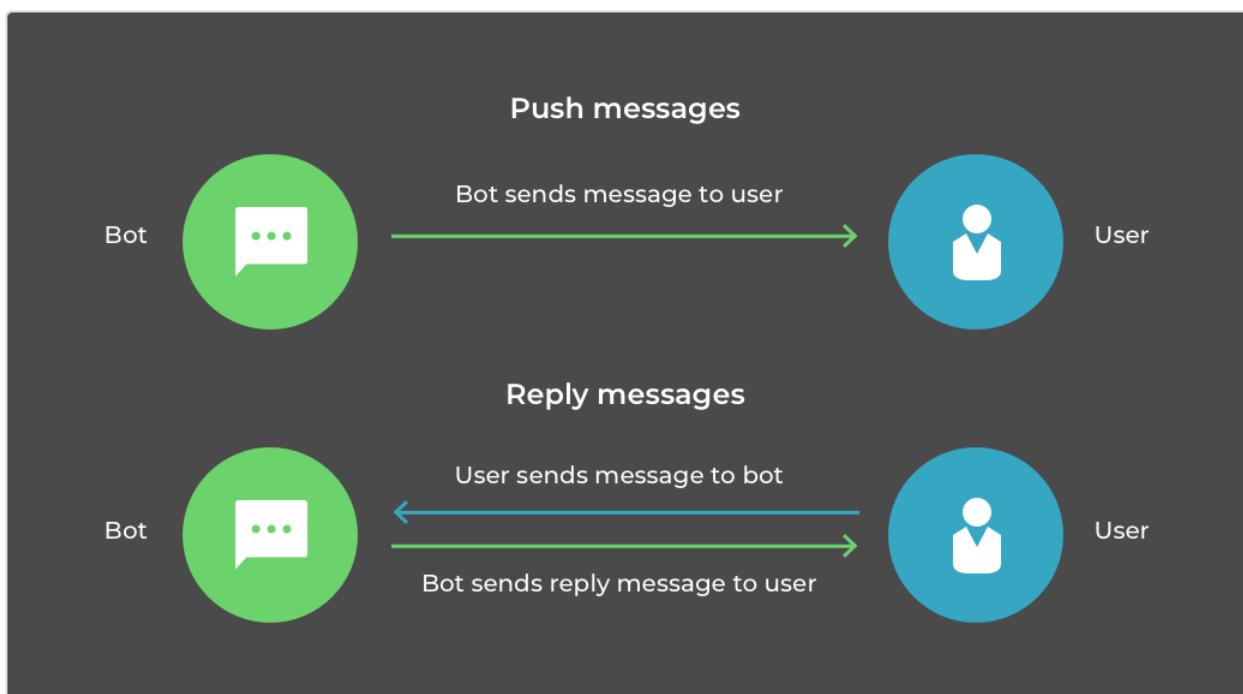
### 1 2.6.4 LINE Pay



Gambar 2.11: Situs web LINE Pay (<https://pay.line.me>)

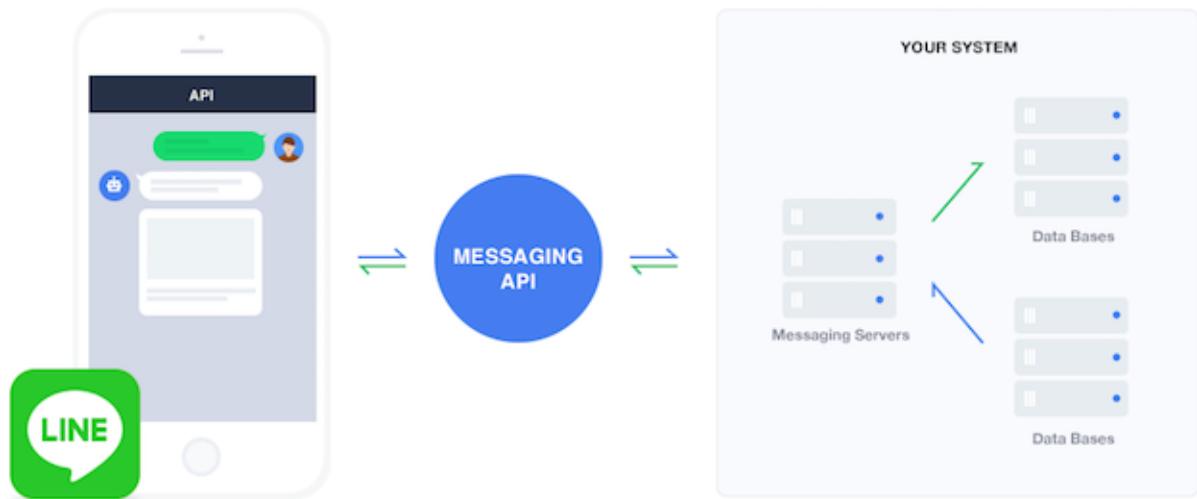
2 LINE Pay adalah produk LINE yang memungkinkan *developer* mengintegrasikan aplikasi yang  
3 dibuat *developer* dengan fitur pembayaran melalui LINE Pay. Tidak ada dokumentasi resmi untuk  
4 produk ini, tapi ada situs web resminya: <https://pay.line.me> (Gambar 2.11). Situs web ini  
5 menyediakan informasi LINE Pay di negara Jepang, Republik Tiongkok / Taiwan, dan Thailand.  
6 Situs ini tersedia dalam bahasa Jepang, Korea, Inggris, China dengan aksara sederhana, China  
7 dengan aksara tradisional, dan Thailand.

### 1 2.6.5 Messaging API



Gambar 2.12: Messaging API memungkinkan *developer* mengirim push message dan reply message

- 2 Messaging API adalah produk LINE yang memungkinkan *developer* untuk membangun *bot*  
3 sebagai sarana komunikasi dua arah antara layanan yang dibangun *developer* dengan pengguna  
4 LINE. Dengan Messaging API, *developer* dapat mengirimkan *push message* dan *reply message*  
5 (Gambar 2.12) ke akun LINE@. *Push message* adalah pesan yang *bot* kirimkan ke pengguna LINE.  
6 *Reply message* adalah pesan yang *bot* kirimkan untuk membalas pesan dari pengguna LINE.  
7 LINE Menyediakan Messaging API untuk membangun messaging bot. Messaging API memungkinkan  
8 data dioper antara server dari aplikasi bot dengan LINE Platform. Ketika pengguna LINE  
9 mengirimkan pesan ke bot, sebuah *webhook* akan terpicu dan LINE Platform akan mengirimkan  
10 permintaan ke URL *webhook bot*. *Server* akan mengirim permintaan ke LINE Platform untuk  
11 merespon pengguna. Permintaan akan dikirimkan dalam format JSON. Arsitektur dari Messaging  
12 API dapat dilihat pada Gambar 2.13.



Gambar 2.13: Arsitektur Messaging API

1      Developer dapat melakukan hal-hal berikut dengan Messaging API :

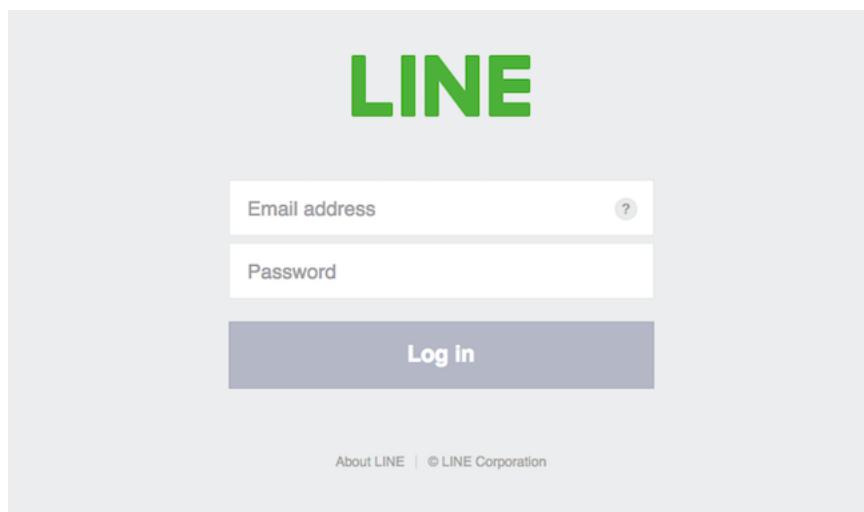
- 2      • Mengirimkan *reply message*  
3      • Mengirimkan *push message*  
4      • Mengirimkan berbagai jenis pesan  
5      • Mendapatkan profil pengguna yang berinteraksi dengan *bot*  
6      • Bergabung dengan percakapan grup (*group chats*)

7      Untuk menggunakan Messaging API, *developer* memerlukan akun LINE@. Messaging API  
8      juga dapat digunakan menggunakan akun resmi/*official accounts*. Akun resmi mendapatkan fitur  
9      tambahan untuk pengguna *enterprise*.

10     **2.6.5.1 Membuat *Channel***

11     Untuk memulai membangun bot dengan Messaging API, *developer* perlu membuat *channel*  
12     terlebih dahulu. *Channel* adalah penyambung antara LINE platform dan aplikasi yang dibuat  
13     *developer*. Berikut langkah-langkah untuk membuat *channel*:

- 14     1. Langkah ke-1 : Masuk ke LINE Developers *console*



Gambar 2.14: Tampilan LINE Developers *console* saat login

1      *Developer* perlu masuk ke LINE Developers *console* (<https://developers.line.me/en/>)  
 2      dengan alamat *email* dan *password* dari akun LINE *developer* (Gambar 2.14). Jika *developer*  
 3      belum memiliki akun LINE, *developer* perlu mengunduh aplikasi LINE untuk mendaftar akun  
 4      LINE.

5      2. Langkah ke-2 : Mendaftar sebagai *developer*

Gambar 2.15: Tampilan LINE Developers *console* saat *register developer*

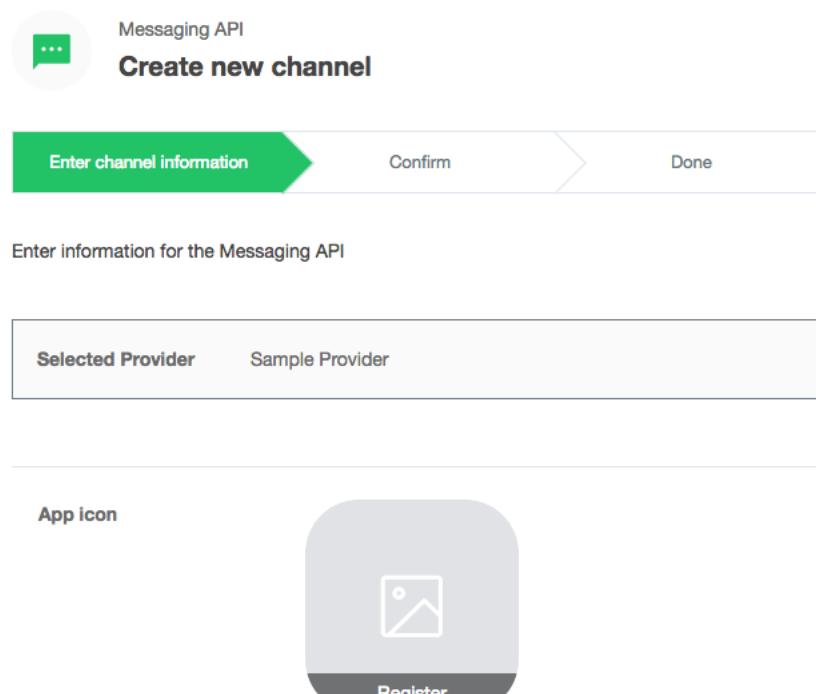
6      Apabila *developer* baru pertama kali masuk ke LINE Developers *console*, *developer* perlu

1 membuat akun *developer* (Gambar 2.15). *Developer* hanya perlu mencantumkan nama dan  
2 alamat *email* untuk mendaftar.

3 3. Langkah ke-3 : Membuat *provider* baru

4 *Provider* adalah individu atau perusahaan yang menyediakan aplikasi yang akan dibuat.  
5 *Developer* perlu mencantumkan nama *provider* untuk membuat *provider* baru. *Developer*  
6 dapat menuliskan nama *developer* sendiri atau nama perusahaan *developer*.

7 4. Langkah ke-4 : Membuat *channel*



Gambar 2.16: Tampilan LINE Developers *console* saat membuat *channel*

8 *Developer* perlu memasukkan informasi yang dibutuhkan untuk membuat *channel*:

9 • Ikon aplikasi

10 Dokumen gambar untuk ikon aplikasi harus dibawah 3 MB dengan ekstensi JPEG/PNG/GIF/BMP.

11 • Nama aplikasi

12 Nama aplikasi tidak boleh lebih dari 20 karakter. Kata "LINE" tidak dapat digunakan  
13 sebagai nama aplikasi, walaupun kapitalisasinya tidak sama. Setelah dikonfirmasi, nama  
14 aplikasi tidak dapat diubah untuk tujuh hari ke depan.

15 • Deskripsi aplikasi

16 Deskripsi aplikasi tidak boleh lebih dari 500 karakter.

17 • Plan

18 Pada saat mendaftar *channel*, pilihan *plan* yang tersedia hanya dua: *Developer Trial* dan  
19 *Free*. *Plan Developer Trial* memungkinkan *developer* untuk membuat bot yang dapat  
20 mengirimkan *push message* dan memiliki 50 teman. Apabila *developer* memilih plan ini,

1 maka *developer* tidak dapat melakukan *upgrade* atau membeli ID *premium*. *Plan Free*  
 2 memungkinkan *developer* untuk membuat bot dengan jumlah teman tak terbatas, tapi  
 3 *developer* tidak dapat mengirimkan *push message*. *Developer* dapat melakukan *upgrade*  
 4 kapan saja dengan *plan* ini.

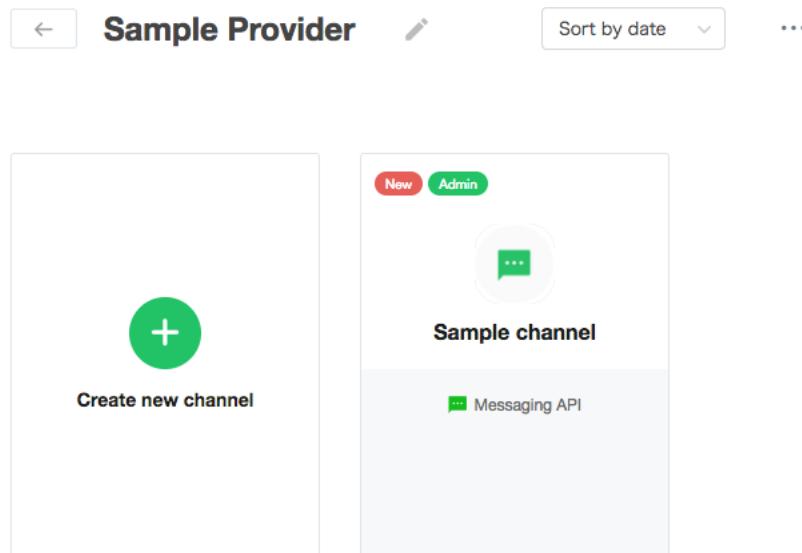
5 • Kategori dan Subkategori

6 *Developer* dapat memilih kategori dan subkategori yang cocok dengan aplikasi yang  
 7 sedang dikembangkan.

8 • Alamat *email*

9 Alamat *email* yang dicantumkan adalah alamat *email* yang akan menerima notifikasi  
 10 dan pengumuman penting dari LINE. Maksimal karakter pada alamat *email* adalah 100  
 11 karakter.

12 5. Konfirmasi



Gambar 2.17: Tampilan LINE Developers *console* saat konfirmasi pembuatan *channel*

13 Konfirmasi *channel* yang baru saja dibuat. Tampilan setelah *channel* dikonfirmasi dapat  
 14 dilihat di Gambar 2.17.

15 2.6.5.2 Membuat *bot*

16 Setelah membangun *channel*, *developer* perlu menyiapkan *server* untuk menjadi *host* dari bot.  
 17 *Developer* dapat menggunakan layanan *cloud platform*, seperti Heroku. Setelah itu, *developer* dapat  
 18 mulai mengatur *bot* pada *console*.

19 Aplikasi *bot* membutuhkan *channel access token* untuk membuat *API call* dan *webhook URL*  
 20 untuk menerima *webhook payload* dari LINE Platform. *Channel access token* adalah *long-lived token*  
 21 (token yang tidak memiliki kadaluarsa) yang harus diatur di dalam *authorization header* ketika  
 22 membuat *API call*. *Developer* dapat menerbitkan lagi *channel access token* kapanpun melalui

1    *console*. Untuk menerbitkan *channel access token*, klik Issue pada "Channel settings" di halaman  
2    *console*. Sedangkan *webhook URL* adalah titik akhir dari *server* aplikasi *bot* dimana *webhook payload*  
3    dikirimkan.

4        Untuk mengatur *webhook URL*, *developer* dapat memasukkannya ke halaman *Channel settings*  
5    pada *console*. *Webhooks* harus diaktifkan terlebih dahulu dengan menekan tombol *enable webhooks*.  
6    Untuk memeriksa apakah *webhook URL* dapat menerima *event webhook*, tekan tombol *Verify* dan  
7    pastikan hasilnya "*Success*". *Webhook URL* harus menggunakan HTTPS dan memiliki sertifikat  
8    SSL yang diterbitkan oleh *certificate authority* (CA) yang terotorisasi.

9        Setelah *token* dan *webhook URL* berhasil diset, tambahkan *bot* sebagai teman melalui akun  
10   LINE. *Developer* dapat melakukannya dengan *scan* kode QR pada *Channel Settings*.

#### 11   **2.6.5.3 Menkonfigurasi Keamanan**

12      *Developer* dapat mengkonfigurasi keamanan tapi tidak wajib dilakukan. Untuk meningkatkan  
13   keamanan, *developer* dapat mengatur *server* yang dapat memanggil API pada LINE Platform pada  
14   *Security settings*. *Developer* dapat mendaftarkan alamat IP secara individual atau jika *developer*  
15   memiliki *server* yang banyak *developer* dapat menggunakan notasi *Classless Inter-Domain Routing*  
16   (CIDR) untuk mendaftarkan alamat jaringan.

#### 17   **2.6.5.4 Alur kerja Messaging API**

18      Ketika *user* berinteraksi dengan *bot* seperti mengirimkan pesan atau menambah *bot* sebagai  
19   teman, LINE Platform mengirimkan HTTP POST *request* yang berisi *webhook event object* ke *bot*  
20   *server* yang disebutkan di kolom "*Webhook URL*" pada *console*. *Request header* berisi *signature*.

21      Untuk mengecek apakah *server* dapat menerima *webhook event*, blokir *bot* pada LINE dan cek  
22   *server logs* untuk menkonfirmasi bahwa *server* dapat menerima *unfollow event* dari LINE Platform.

23      Untuk memastikan *request* yang dikirim berasal dari LINE Platform, *bot server* harus memvalidasi  
24   *X-Line-Signature* pada *request header*. Caranya dengan : 1. Menggunakan *channel secret* sebagai  
25   *secret key*, menghasilkan *Base64-encoded digest* dari *request body* menggunakan algoritma HMAC-  
26   SHA256 2. Menkonfirmasi *signature X-Line-Signature* dalam *request header* cocok dengan *digest*.

#### 27   **2.6.5.5 Webhook Event Object**

28      1. Berikut *webhook event object* khusus untuk *one-on-one chat*:

- 29            • *Message Event*

30            Menunjukkan bahwa ada *user* yang mengirim pesan. *Event* ini dapat dibalas.

- 31            • *Follow Event*

32            Menunjukkan bahwa akun *bot* ditambahkan sebagai teman (atau dibuka blokirnya).  
33            *Event* ini dapat dibalas.

- 34            • *Unfollow Event*

35            Menunjukkan bahwa akun *bot* diblokir

- 36            • *Postback event*

37            Menunjukkan *user* melakukan aksi *postback*. *Event* ini dapat dibalas.

1     • *Beacon event*

2         Menunjukkan bahwa *user* telah masuk atau keluar dari jangkauan *LINE Beacon*. *Event*  
3         ini dapat dibalas.

4     • *Account link event*

5         Menunjukkan bahwa *user* telah menghubungkan akun LINE dengan akun layanan  
6         *developer*.

7     2. Berikut *webhook event object* khusus untuk *group chats*:

8     • *Message event*

9         Menunjukkan bahwa ada *user* yang mengirim pesan. *Event* ini dapat dibalas.

10    • *Join event*

11         Menunjukkan *bot* telah bergabung ke sebuah *group chat*

12    • *Leave event*

13         Menunjukkan *bot* telah keluar dari sebuah *group chat*

14    • *Postback event*

15         Menunjukkan *user* melakukan aksi *postback*. *Event* ini dapat dibalas.

16 **2.6.5.6 Operasi pada bot**

17 *Developer* dapat melakukan operasi berikut lewat *bot* :

18 1. Mengirim *reply message*

19 *Reply message* adalah pesan yang dikirim sebagai respons dari *user-generated event*. *User-*  
20 *generated event* adalah *event* yang muncul karena *user* berinteraksi dengan *bot*, misalnya  
21 mengirim pesan. *Developer* hanya dapat membalas *webhook events* yang memiliki *reply*  
22 *token*. Untuk membalas pesan, kirim HTTP POST *request* ke /bot/message/reply. Sertakan  
23 *channel access token* di dalam *authorization header* dan *reply token* di *request body*. *Developer*  
24 dapat mengirimkan sampai 5 *message object* per *request*.

25 2. Mengirim *push message*

26 Untuk mengirim *push message*, *developer* harus memerhatikan *plan* yang dipakai. Apabila  
27 *developer* memakai *plan Free* maka *developer* tidak dapat melakukan operasi ini. *Push message*  
28 adalah pesan yang dapat *bot* kirimkan ke *user* kapan saja. *Push message* tidak membutuhkan  
29 *reply token* seperti saat mengirim *reply message*. Ketika mengirim *push message*, sebutkan *user*  
30 *ID* di dalam *property to*. *ID* penerima dapat ditemukan dari *webhook event object*. Apabila  
31 penerima hanya satu, kirimkan *request* ke /bot/message/push. Sedangkan apabila penerima  
32 ada beberapa, kirimkan ke /bot/message/multicast. *Developer* dapat mengirimkan sampai  
33 5 *message object* per *request*.

34 3. Mendapatkan konten yang dikirim oleh *user*

35 Untuk mengambil gambar, video, atau audio yang dikirim *user*, kirimkan HTTP GET *request*  
36 ke /bot/message/*messageId*/content. Konten yang dikirim oleh *user* otomatis dihapus  
37 dalam jangka waktu tertentu.

1     4. Mendapatkan informasi *user profile*

2         Untuk mendapatkan informasi *user profile* dari *user* yang menambahkan bot atau mengirim  
3         pesan ke bot, kirimkan HTTP GET *request* ke /bot/profile/*userId*. *Request* ini akan  
4         mengembalikan *display name*, *user ID*, *profile image URL*, dan *status message* (jika tersedia)  
5         dari *user*.

6     **2.6.6 LINE@ Manager**

7         LINE@ Manager adalah alat untuk mengatur akun LINE@ (LINE bot). *Developer* dapat  
8         meningkatkan *user experience* dengan mengatur halaman akun, membuat *Timeline post*, dan  
9         menggunakan fitur lain yang disediakan LINE@ Manager. Berikut adalah hal-hal yang bisa  
10      dilakukan:

11     1. Mengubah tampilan halaman akun

12         *Developer* dapat mengubah gambar cover, logo, tombol, dan informasi yang disediakan

13     2. Mengatur *greeting message*

14         Jika *developer* mengaktifkan *greeting message* pada *Channel settings*, maka *developer* dapat  
15         mengatur *greeting message* yang akan dikirim ke *user* saat pertama kali menambahkan bot  
16         sebagai teman. *Developer* dapat melakukannya juga dengan program melalui *follow webhook*  
17         event.

18     3. Mengatur *auto reply message*

19         Jika *developer* mengaktifkan "Auto reply message" pada *Channel settings*, maka *developer*  
20         dapat mengatur pesan balasan otomatis setiap *user* mengirimkan pesan ke bot.

21     **2.7 System Usability Scale [6]**

22         System Usability Scale adalah salah satu metode untuk menguji usabilitas dari sebuah sistem  
23         atau produk. Metode ini menggunakan kuesioner yang terdiri dari 10 pernyataan berikut:

24     1. I think that I would like to use this system frequently.

25     2. I found the system unnecessarily complex.

26     3. I thought the system was easy to use.

27     4. I think that I would need the support of a technical person to be able to use this system.

28     5. I found the various functions in this system were well integrated.

29     6. I thought there was too much inconsistency in this system.

30     7. I would imagine that most people would learn to use this system very quickly.

31     8. I found the system very cumbersome to use.

32     9. I felt very confident using the system.

- 1      10. I needed to learn a lot of things before I could get going with this system.
- 2      Setiap pernyataan dinilai oleh responden dengan angka di antara 1 sampai 5. Angka 1 menunjukkan  
3      responden tidak setuju dengan pernyataan tersebut dan angka 5 menunjukkan bahwa responden  
4      sangat setuju dengan pernyataan tersebut. Setelah seluruh responden mengisi kuesioner tersebut,  
5      skor *System Usability Scale* dari tiap responden dihitung. Cara menghitungnya adalah sebagai  
6      berikut:
- 7      1. Kurangi nilai pertanyaan bermomor ganjil dengan angka 1.  
8      2. Kurangi 5 dengan nilai pertanyaan bermomor genap.  
9      3. Jumlahkan semua hasilnya lalu kalikan dengan 2.5.
- 10     Menurut analisis Tullis(2008) dan Bangor, Kortum, dan Miller(2009), apabila rata-rata skor *System  
11    Usability Scale* kurang dari 50, maka usabilitas sistem atau produk termasuk jelek. Apabila rata-rata  
12    skor *System Usability Scale* di antara 50-70, maka usabilitas sistem atau produk termasuk kelompok  
13    rata-rata. Apabila rata-rata skor *System Usability Scale* lebih dari 70, maka usabilitas sistem atau  
14    produk termasuk bagus.

1

## BAB 3

2

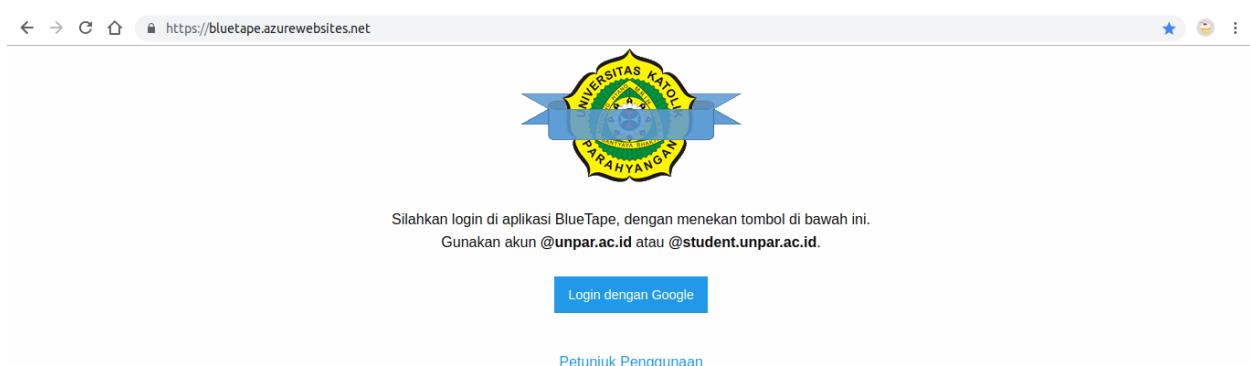
### ANALISIS

3

Pengumpulan data dalam skripsi ini dilakukan dengan cara studi pustaka.

4

#### 3.1 Analisis Sistem yang Sudah Ada



Gambar 3.1: Tampilan utama BlueTape

5

BlueTape adalah perangkat lunak yang berfungsi untuk membantu urusan-urusan *paper-based* di

6

FTIS UNPAR menjadi *paperless*. Pada saat skripsi ini dibuat, BlueTape dapat diakses melalui situs

7

web <https://bluetape.azurewebsites.net/> (Gambar 3.1). Perangkat lunak ini bersifat open

8

source, sehingga kode program BlueTape bisa dipelajari, diubah, dan distribusi oleh siapapun untuk

9

tujuan apapun. Kode program ini dapat diakses di <https://github.com/ftisunpar/BlueTape>.

10

BlueTape memanfaatkan CodeIgniter (versi 3.1.4) dan ZURB Foundation.

11

Pola pengembangan yang dipakai BlueTape adalah MVC (Model-View-Controller). MVC

12

(Model-View-Controller) adalah sebuah metode untuk membuat perangkat lunak menjadi tiga

13

bagian : Model, View, dan Controller. Model adalah kelas yang merepresentasikan struktur data.

14

View adalah informasi yang disajikan ke pengguna. Controller adalah penghubung antara Model,

15

View, dan sumber daya lain yang dibutuhkan untuk mengolah HTTP request dan menghasilkan

16

situs web.

### 1 3.1.1 Aturan Konstribusi BlueTape

2 Terdapat beberapa aturan apabila ingin berkonstribusi pada pengembangan BlueTape. Aturan-  
 3 aturan tersebut tertera pada dokumen CONTRIBUTING.md ([https://github.com/ftisunpar/BlueTape/  
 blob/master/CONTRIBUTING.md](https://github.com/ftisunpar/BlueTape/blob/master/CONTRIBUTING.md)).

#### 5 3.1.1.1 Pengelompokan Module

6 Perangkat lunak BlueTape dikelompokkan dalam module. Setiap module memiliki nama yang  
 7 mengikuti aturan CamelCase. Jika beberapa module tergabung pada satu topik yang sama, topik  
 8 tersebut harus digunakan sebagai kata pertama dalam penamaan module. Contohnya : apabila  
 9 nama topik adalah Transkrip, maka contoh nama modulennya adalah TranskripRequest dan  
 10 TranskripManage.

11 Penamaan dokumen pada controller, view, model, config file, nama tabel, dan migration script  
 12 menggunakan nama module atau topik. Contoh penamaan :

- 13 • Controller: controllers/TranskripRequest.php, controllers/TranskripManage.php
- 14 • View: views/TranskripRequest/\*.php, views/TranskripManage/\*.php
- 15 • Model (opsional): models/Transkrip/\*\_model.php
- 16 • Config file (opsional): config/Transkrip.php
- 17 • Nama tabel (opsional): Transkrip
- 18 • Migration script (opsional): migrations/20160222120000\_Transkrip\_initial.php

#### 19 3.1.1.2 Model

20 Model dibuat hanya jika fungsi-fungsi di dalamnya digunakan lebih dari sekali. Apabila hanya  
 21 digunakan sekali, letakkan fungsi pada controller.

#### 22 3.1.1.3 Library bluetape

23 Library bluetape berisi fungsi-fungsi yang umum digunakan di BlueTape. Contoh : fungsi  
 24 untuk konversi email ke NPM.

#### 25 3.1.1.4 Hak Akses

26 Hak akses dan nama module diatur pada dokumen config/modules.php. Contoh :

```
28 $config['module-names'] = array(
29     'TranskripRequest' => 'Permohonan Cetak Transkrip',
30     'TranskripManage' => 'Manajemen Cetak Transkrip'
31 );
32
33 $config['modules'] = array(
34     'TranskripRequest' => array('root', 'mahasiswa.ftis'),
```

```

1   'TranskripManage' => array('root', 'tu.ftis')
2 );
3
4 $config['roles'] = array(
5   'root' => 'pascal@unpar\\.ac\\.id',
6   'tu.ftis' => '(shao\\.wei)@unpar\\.ac\\.id',
7   'mahasiswa.ftis' => '7[123]\\d{5}@student\\.unpar\\.ac\\.id'
8 );

```

10 Apabila diperlukan, kontributor boleh menambahkan role baru pada array config "roles". Setiap  
11 elemen array memetakan role dengan alamat email yang tergabung dalam role tersebut, dengan  
12 notasi regular expression.

### 13 3.1.2 Autentikasi

14 Setiap module wajib memeriksa hak akses sebelum ditampilkan. Hal tersebut dilakukan dengan  
15 cara memanfaatkan template berikut pada controller:

```

16 <?php
17 defined('BASEPATH') OR exit('No direct script access allowed');
18 class NamaPage extends CI_Controller {
19
20
21   public function __construct() {
22     parent::__construct();
23     try {
24       $this->Auth_model->checkModuleAllowed(get_class());
25     } catch (Exception $ex) {
26       $this->session->set_flashdata('error', $ex->getMessage());
27       header('Location: /');
28     }
29   }
30
31   // ... implementasikan method-method Anda yang lain di sini...
32 }
33

```

#### 34 3.1.2.1 View

35 Setiap view menggunakan template yang menampilkan nama module, menu navigasi, dan *flash*  
36 *message* (jika diperlukan). Setiap view membutuhkan parameter `currentModule`, selain parameter-  
37 parameter lainnya. Jika ingin memanggil view dari controller, fungsi `get_class()` dapat digunakan.  
38 Berikut adalah cara sederhana memanggil view :

```

39
40 $this->load->view('NamaPage/main', array('currentModule' => get_class()));
41

```

42 View memanfaatkan framework Zurb Foundation, dan berisi template menu utama serta flash  
43 message. Oleh karena itu, kode berikut digunakan untuk memulai membuat view :

```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?><!doctype html>
4 <html class="no-js" lang="en">
5     <?php $this->load->view('templates/head_loggedin'); ?>
6     <body>
7         <?php $this->load->view('templates/topbar_loggedin'); ?>
8         <?php $this->load->view('templates/flashmessage'); ?>
9
10        <!-- Tulislah isi view Anda di sini. -->
11
12
13        <script src="/public/foundation-6/js/vendor/jquery.min.js"></script>
14        <script src="/public/foundation-6/js/vendor/what-input.min.js"></script>
15        <script src="/public/foundation-6/js/foundation.min.js"></script>
16        <script src="/public/foundation-6/js/app.js"></script>
17    </body>
18 </html>
19
```

### 20 3.1.3 Fitur - Fitur BlueTape

21 Saat skripsi ini ditulis, BlueTape memiliki perangkat lunak BlueTape memiliki tiga layanan,  
22 yaitu Transkrip *Request / Manage*, Perubahan Kuliah *Request / Manage*, dan perekam jadwal  
23 dosen. Layanan Transkrip *Request / Manage* memberikan layanan untuk melakukan permohonan  
24 serta pencetakan transkrip mahasiswa. Layanan Perubahan Kuliah *Request / Manage* memberikan  
25 layanan untuk permohonan dan pencetakan perubahan jadwal kuliah oleh dosen. Layanan perekam  
26 jadwal dosen untuk merekam dan menampilkan jadwal dosen.

### 1 3.1.3.1 Transkrip *Request*

Gambar 3.2: Tampilan Cetak Transkrip

2 Gambar 3.2 menampilkan halaman utama saat cetak transkrip. Halaman ini hanya bisa diakses  
 3 oleh user yang termasuk roles 'root' dan 'mahasiswa.ftis'. Terdapat form yang meminta input alamat  
 4 email pemohon, npm, nama, tipe transkrip, dan keperluan. Input alamat email pemohon, npm, dan  
 5 nama sudah terisi otomatis dan tidak bisa diubah lagi sehingga user hanya perlu mengganti tipe  
 6 transkrip dan mengisi keperluan. Tipe transkrip memiliki tiga pilihan, yaitu : DPS Bahasa Indonesia  
 7 (Seluruh Semester), DPS Bahasa Inggris (Seluruh Semester), dan LHS (Semester Terakhir). Selain  
 8 form tersebut, terdapat tabel histori permohonan yang akan menampilkan riwayat permohonan  
 9 cetak transkrip jika sudah pernah memohon. Tabel histori permohonan memiliki tujuh kolom, yaitu  
 10 : ID, Status, Tanggal Permohonan, Tipe Transkrip, Tanggal Jawab/Cetak, Keterangan, dan Aksi  
 11 (tindakan yang bisa dilakukan dengan record). Gambar 3.3 menampilkan tampilan setelah form  
 12 permohonan transkrip baru dikirimkan. Pada Gambar 3.3 aksi yang tersedia hanya melihat detail  
 13 permohonan.

Gambar 3.3: Tampilan hasil Request Cetak Transkrip

### 1 3.1.3.2 Transkrip Manage

Permintaan Transkrip

Cari NPM: 2013730013 Cari

ID	Status	Tanggal Permohonan	Tipe Transkrip	NPM	Aksi
#1	MENUNGGU	Thursday, 15 November 2018	DPS_ID	2015730029	

Gambar 3.4: Tampilan Manajemen Transkrip BlueTape

2 Gambar 3.4 menampilkan tampilan halaman Manajemen Cetak Transkrip. Halaman ini  
 3 hanya bisa diakses oleh user yang termasuk roles 'root' dan 'tu.ftis'. Halaman ini memiliki kolom  
 4 pencarian yang dapat diisi dengan npm mahasiswa. Angka "2013730013" pada Gambar 3.4  
 5 merupakan placeholder saja, bukan input user. Apabila input kosong, maka semua permohonan  
 6 akan ditampilkan pada tabel di bawah kolom pencarian. Pada Gambar 3.4, permohonan yang  
 7 masuk baru satu saja. Apabila input diisi dan tombol "Cari ditekan", maka permohonan yang  
 8 ditampilkan hanya permohonan milik npm yang diinput. User dapat melakukan empat aksi untuk  
 9 tiap record yang ditampilkan : melihat detail permohonan (simbol mata), menolak permohonan  
 10 (simbol jempol ke bawah), menyetujui permohonan (simbol print), dan menghapus permohonan  
 11 (simbol tempat sampah).

### 12 3.1.3.3 Perubahan Kuliah Request

Permohonan Baru

Pemohon: 7315029@student.unpar.ac.id Nama: ELLENA ANGELICA

Kode MK: \_\_\_\_\_ Nama Mata Kuliah: \_\_\_\_\_ Kelas: \_\_\_\_\_ Jenis Perubahan: Diganti

Dari Hari & Jam: \_\_\_\_\_ Dari Ruang: \_\_\_\_\_ Keterangan Tambahan: \_\_\_\_\_

Menjadi Hari & Jam: \_\_\_\_\_ Menjadi Ruang: \_\_\_\_\_

**Kirim Permohonan** **Tambah Pertemuan Ekstra**

Histori Permohonan

ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Tanggal Jawab	Keterangan	Aksi
----	--------	--------------------	---------	-----------	---------------	------------	------

Gambar 3.5: Tampilan request perubahan kuliah

1        Gambar 3.5 menampilkan halaman Perubahan Kuliah. Halaman ini hanya bisa diakses oleh  
 2        user yang termasuk roles 'root' dan 'staf.unpar'. Terdapat form yang meminta input alamat email  
 3        pemohon, nama, kode mk, nama mata kuliah, kelas, jenis perubahan, hari dan jam sebelum dan  
 4        sesudah diubah, serta ruang sebelum dan sesudah diubah. Input alamat email pemohon dan nama  
 5        sudah terisi otomatis dan tidak bisa diubah lagi. Jenis perubahan memiliki tiga pilihan, yaitu  
 6        : diganti, tambahan, dan ditadakan. Selain form tersebut, terdapat tabel histori permohonan  
 7        yang akan menampilkan riwayat permohonan perubahan jadwal kuliah. Tabel histori permohonan  
 8        memiliki delapan kolom, yaitu : ID, Status, Tanggal Permohonan, Kode MK, Perubahan, Tanggal  
 9        Jawab, Keterangan, dan Aksi (tindakan yang bisa dilakukan dengan record).

10 **3.1.3.4 Perubahan Kuliah Manage**

Permohonan Perubahan Kuliah					
ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Aksi
#1	MENUNGGU	Thursday, 15 November 2018	AIF401	Ditiadakan	

Gambar 3.6: Tampilan manage perubahan kuliah

11        Gambar 3.6 menampilkan halaman Manajemen Perubahan Kuliah. Halaman ini hanya bisa  
 12        diakses oleh user yang termasuk roles 'root' dan 'tu.ftis'. Halaman ini berisi riwayat permohonan  
 13        perubahan kuliah. User dapat melakukan empat aksi untuk tiap record yang ditampilkan : melihat  
 14        detail permohonan (simbol mata), menolak permohonan (simbol jempol ke bawah), menyetujui  
 15        permohonan (simbol print), dan menghapus permohonan (simbol tempat sampah).

16 **3.1.3.5 Entri Jadwal Dosen**

Tambah Jadwal					
Hari	Senin	Durasi	1 jam	Label	
Jam Mulai	7:00	Jenis	Konsultasi	Tambah	

Gambar 3.7: Tampilan tambah jadwal dosen

17        Gambar 3.7

Daftar Jadwal					
	Senin	Selasa	Rabu	Kamis	Jumat
7-8	Bimbingan				
8-9					
9-10					
10-11					
11-12					
12-13					
13-14					
14-15					
15-16					
16-17					

Terakhir diupdate pada : 15 November 2018

[Delete All](#) [Ekspor ke XLS](#)

Gambar 3.8: Tampilan jadwal dosen

- 1 Halaman Entri Jadwal Dosen hanya bisa diakses oleh user yang termasuk roles 'root' dan 'dosen.informatika'. Halaman ini terdiri dari dua bagian : form tambah jadwal dan daftar jadwal.
- 2 Gambar 3.7 menunjukkan form tambah jadwal. Form ini meminta input hari, durasi, label, jam mulai dan jenis. Gambar 3.8 menunjukkan daftar jadwal milik user. Kolom yang memiliki isi bisa diklik untuk diedit. Gambar 3.9 menunjukkan tampilan edit jadwal.

Durasi
Label
X

### Edit Jadwal

Hari

Jam Mulai

Durasi

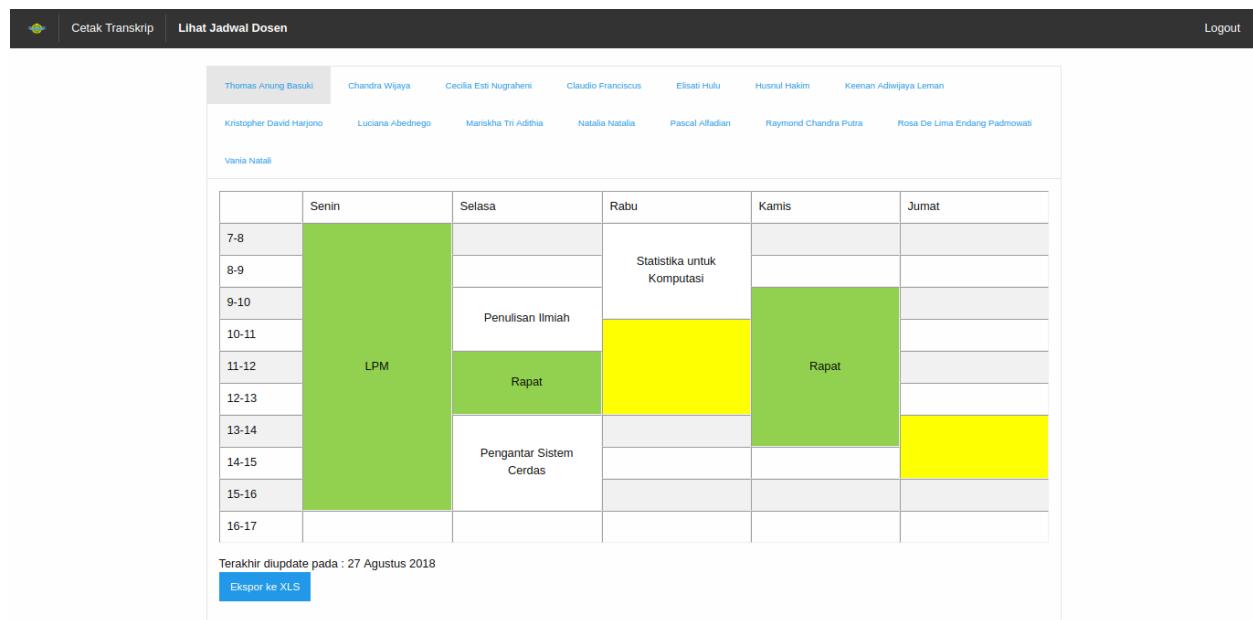
Jenis

Label

[Save](#)
[Delete](#)

Gambar 3.9: Tampilan edit jadwal dosen

### 1 3.1.3.6 Lihat Jadwal Dosen



Gambar 3.10: Tampilan lihat jadwal dosen

2 Gambar 3.10 menampilkan halaman Lihat Jadwal Dosen. Halaman ini bisa diakses oleh user  
 3 yang termasuk roles 'root', 'mahasiswa.informatika', dan 'dosen.informatika'. Halaman ini memiliki  
 4 tab-tab yang masing-masing diberi label nama dosen. Di bawah tab terdapat tabel jadwal dari  
 5 dosen yang tabnya aktif.

### 6 3.1.4 Hak Akses

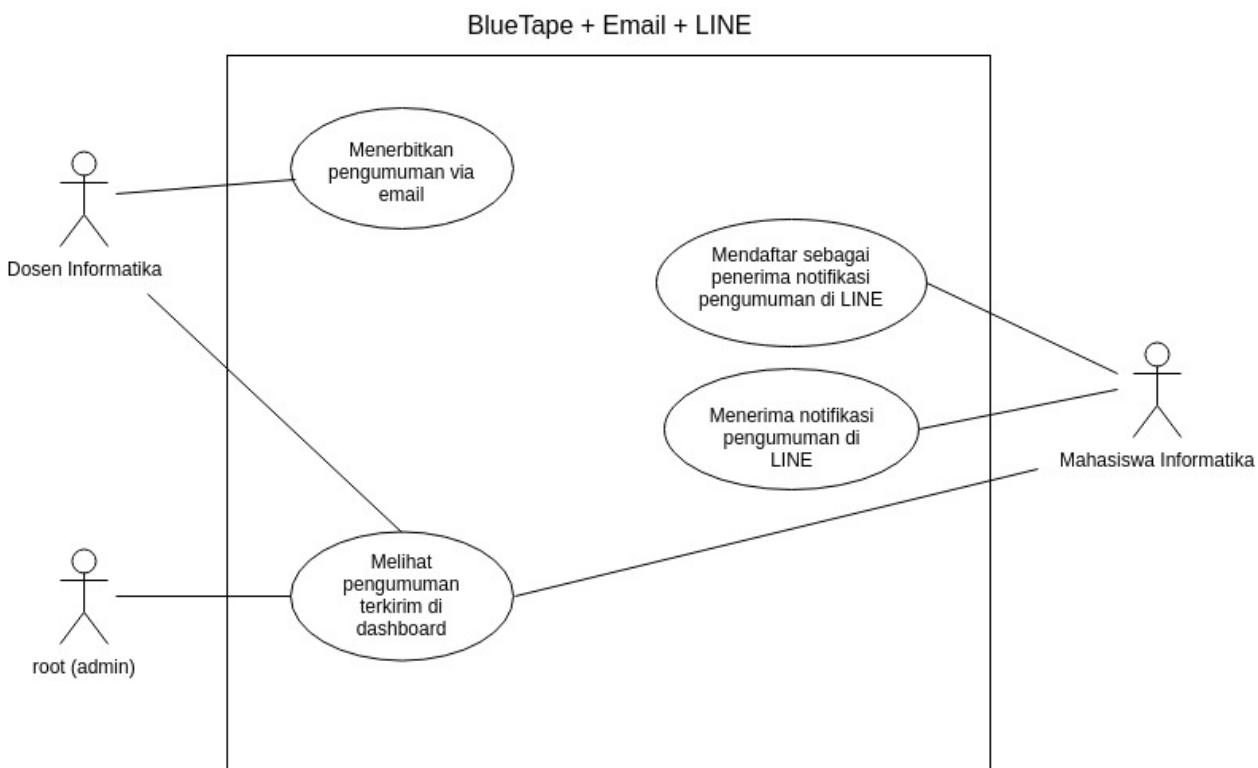
7 Hak akses dan nama module diatur pada dokumen `config/modules.php` yang terletak di  
 8 dalam direktori `config`. Hak akses dikelompokkan di dalam kelompok yang disebut *role*. Saat  
 9 skripsi ini dibuat, hak akses dibagi ke dalam lima *role* : root, mahasiswa.ftis, tu.ftis, staf.unpar,  
 10 dosen.informatika, dan mahasiswa.informatika. *Role* root berisi daftar alamat email dari pengembang  
 11 bluetape. *Role* tu.ftis berisi daftar alamat email dari tata usaha ftis. *Role* mahasiswa.ftis berisi  
 12 daftar alamat email dari mahasiswa ftis. *Role* staf.unpar berisi daftar alamat email dari staf unpar.  
 13 *Role* dosen.informatika berisi daftar alamat email dari dosen informatika.

14 Setiap *role* memiliki batasan dalam mengakses module di BlueTape. *Role* root tidak memiliki  
 15 batasan dan dapat mengakses setiap module yang ada. *Role* tu.ftis hanya dapat mengakses  
 16 module `TranskripManage`, dan module `PerubahanKuliahManage`. *Role* mahasiswa.ftis hanya dapat  
 17 mengakses module `TranskripRequest` dan module `LihatJadwalDosen`. *Role* staf.unpar hanya dapat  
 18 mengakses module `PerubahanKuliahRequest`. *Role* dosen.informatika hanya dapat mengakses  
 19 module `EntriJadwalDosen`.

## 3.2 Analisis Fitur yang Dibangun

2 Bagian ini membahas analisis fitur kolektor pengumuman.

### 3.2.1 Diagram Use Case



Gambar 3.11: Use case diagram fitur kolektor pengumuman

4 Gambar 3.11 merupakan gambar diagram use case fitur kolektor pengumuman. Pada diagram  
 5 use case fitur kolektor pengumuman terdapat tiga aktor : dosen informatika, mahasiswa informatika,  
 6 dan root (admin). Berikut ini adalah penjelasan dari skenario pada diagram use case tersebut :

7 1. Menerbitkan pengumuman via email

8 • Aktor : Dosen Informatika.

9 • Skenario Normal

10 (a) Dosen mengirimkan email ke alamat email yang dikhususkan untuk menampung  
 11 pengumuman di jurusan Teknik Informatika.

12 (b) BlueTape mengecek email tersebut pada periode tertentu.

13 (c) Jika alamat email yang dosen pakai terdaftar di BlueTape, maka BlueTape akan  
 14 menampilkannya dan mengirim notifikasi ke LINE.

15 • Skenario Exception

16 (a) Dosen mengirimkan email ke alamat email yang dikhususkan untuk menampung  
 17 pengumuman di jurusan Teknik Informatika.

- 1       (b) BlueTape mengecek email tersebut pada periode tertentu.  
2       (c) Jika alamat email yang dosen pakai tidak terdaftar di BlueTape, maka BlueTape  
3       akan mengabaikan email tersebut.
- 4     2. Mendaftar sebagai penerima notifikasi pengumuman di LINE
- 5       • Aktor : Mahasiswa Informatika.  
6       • Skenario Normal  
7           (a) Mahasiswa mengikuti bot BlueTape dengan menambahkannya sebagai teman.  
8           (b) Bot BlueTape akan mengirim notifikasi kepada mahasiswa jika ada pengumuman  
9       baru di BlueTape.
- 10    3. Menerima notifikasi pengumuman di LINE
- 11       • Aktor : Mahasiswa Informatika.  
12       • Skenario Normal  
13           (a) Mahasiswa menerima notifikasi dari bot BlueTape saat ada pengumuman baru di  
14       BlueTape.  
15           (b) Mahasiswa mengunjungi URL yang dicantumkan di pesan dari notifikasi tersebut.  
16           (c) Mahasiswa perlu login menggunakan email student miliknya terlebih dahulu sebelum  
17       dapat mengunjungi URL tersebut.
- 18    4. Melihat pengumuman terkirim di dashboard
- 19       • Aktor : Dosen Informatika, Mahasiswa Informatika, dan root (admin).  
20       • Skenario Normal  
21           (a) Dosen Informatika, Mahasiswa Informatika, atau root (admin) mengunjungi menu  
22       Pengumuman. Menu Pengumuman berisi daftar pengumuman yang masuk ke  
23       BlueTape. Daftar tersebut disortir dari yang paling baru.  
24           (b) Saat salah satu pengumuman diklik, maka isi dan detail pengumuman tersebut akan  
25       ditampilkan.

26 **3.2.2 Modifikasi BlueTape agar Dapat Berjalan di Heroku**

27 **3.2.2.1 Dependensi**

28     BlueTape membutuhkan dependensi tambahan agar perangkat lunak dapat dijalankan. Dependensi  
29     tambahan perangkat lunak BlueTape tertera pada dokumen `composer.json`. Berikut isinya :

```
31       {  
32         "require": {  
33            "google/apiclient": "^1.0",  
34            "ext-imap": "*",  
35            "phpoffice/phpexcel": "^1.8",
```

```

1         "linecorp/line-bot-sdk": "^3.6"
2     }
3 }
4

```

5 Pada saat skripsi ini ditulis, BlueTape telah memakai dua package : package `google/apiclient`  
 6 dan package `phpoffice/phpexcel`. Package `google/apiclient` adalah package yang diperlukan  
 7 untuk autentikasi akun saat masuk ke BlueTape. Sedangkan package `phpoffice/phpexcel` adalah  
 8 package yang digunakan untuk menghasilkan dokumen excel. Package yang ditambahkan untuk  
 9 skripsi ini adalah package `ext-imap` dan `line-bot-sdk`. Package `ext-imap` digunakan untuk  
 10 mengakses email. Package `line-bot-sdk` digunakan untuk menghubungkan perangkat lunak  
 11 dengan layanan yang disediakan oleh LINE.

### 12 3.2.2.2 Tipe Proses

13 BlueTape memiliki satu tipe proses, yaitu tipe proses `web` dan tipe proses `release`. Tipe proses  
 14 `web` adalah tipe proses yang digunakan untuk menerima arus HTTP eksternal dari router Heroku.  
 15 Peneliti tidak dapat menambahkan tipe proses lain karena itu berarti peneliti harus menambah  
 16 dyno. Penambahan dyno perlu informasi kartu kredit untuk verifikasi akun.

### 17 3.2.2.3 Procfile

18 Procfile adalah dokumen yang menjelaskan Heroku bagian-bagian perangkat lunak yang dapat  
 19 dieksekusi. Procfile berisi daftar tipe proses beserta cara menjalankannya. BlueTape hanya memiliki  
 20 satu tipe proses, yaitu tipe proses `web`. Isi Procfile adalah :

```

21
22 web: vendor/bin/heroku-php-apache2 www/
23

```

24 Maksud dari satu baris Procfile tersebut adalah : untuk menjalankan tipe proses web, heroku  
 25 harus menjalankan server apache di heroku dan kemudian server menjalankan perangkat lunak  
 26 web yang ada di direktori `www`. Perintah `vendor/bin/heroku-php-apache2` adalah perintah  
 27 untuk menjalankan server apache di heroku yang ada di package `heroku-php-apache2`. Package  
 28 `heroku-php-apache2` ini otomatis disediakan saat membuat perangkat lunak php di heroku sehingga  
 29 tidak perlu ditambahkan di composer.json. Perintah `www/` berguna untuk mengarahkan server  
 30 apache heroku ke direktori `www`.

### 31 3.2.2.4 Slug

32 Peneliti tidak menambahkan dokumen `.slugignore` karena tidak diperlukan.

### 33 3.2.2.5 Buildpack

34 Buildpack yang dipakai pada skripsi ini hanya `heroku/php`. Buildpack ini secara otomatis  
 35 dipakai oleh Heroku karena BlueTape memakai bahasa PHP.

### 36 3.2.2.6 Dyno

37 Jenis dyno yang dipakai pada skripsi ini adalah free dyno. Jumlah dyno hanya satu. Dyno  
 38 tersebut merupakan dyno untuk tipe proses `web`.

<sup>1</sup> **3.2.2.7 Config Vars**

<sup>2</sup> Berikut adalah config vars yang dipakai pada skripsi ini:

- <sup>3</sup> • CI\_DB\_DATABASE : nama database yang digunakan.
- <sup>4</sup> • CI\_DB\_HOSTNAME : nama host dari database yang disebutkan di config var CI\_DB\_DATABASE.
- <sup>5</sup> • CI\_DB\_USERNAME : username yang digunakan untuk terhubung ke database yang disebutkan  
<sup>6</sup> di config var CI\_DB\_DATABASE.
- <sup>7</sup> • CI\_DB\_PASSWORD : password dari username yang disebutkan di config var CI\_DB\_USERNAME.
- <sup>8</sup> • HEROKU\_POSTGRESQL\_BLUE\_URL : URL database. Dibuat secara otomatis saat  
<sup>9</sup> membuat database.
- <sup>10</sup> • GOOGLE\_CLIENTID : Google Client ID, digunakan untuk melakukan autentikasi saat user  
<sup>11</sup> login.
- <sup>12</sup> • GOOGLE\_CLIENTSECRET : Google Client Secret, digunakan untuk melakukan autentikasi  
<sup>13</sup> saat user login.
- <sup>14</sup> • ANNOUNCEMENT\_EMAIL : alamat email yang dipakai untuk menampung pengumuman.
- <sup>15</sup> • ANNOUNCEMENT\_PASSWORD : password untuk alamat email yang disebutkan di config  
<sup>16</sup> var ANNOUNCEMENT\_EMAIL.
- <sup>17</sup> • HOSTNAME\_INCOMING\_EMAIL : nama host dari alamat email yang disebutkan di config  
<sup>18</sup> var ANNOUNCEMENT\_EMAIL.
- <sup>19</sup> • CI\_BASE\_URL : base URL BlueTape.
- <sup>20</sup> • LINE\_BOT\_CHANNEL\_SECRET : LINE Bot Channel Secret digunakan untuk terhubung  
<sup>21</sup> ke channel bot untuk pengumuman.
- <sup>22</sup> • LINE\_BOT\_CHANNEL\_TOKEN : LINE Bot Channel Token digunakan untuk terhubung  
<sup>23</sup> ke channel bot untuk pengumuman.
- <sup>24</sup> • SMTP\_HOST : SMTP host, konfigurasi untuk mengirim email.
- <sup>25</sup> • SMTP\_PASS : SMTP pass, konfigurasi untuk mengirim email.
- <sup>26</sup> • SMTP\_PORT : SMTP port, konfigurasi untuk mengirim email.
- <sup>27</sup> • SMTP\_USER : SMTP user, konfigurasi untuk mengirim email.

<sup>28</sup> **3.2.2.8 Region**

<sup>29</sup> Region yang dipakai adalah region default, yaitu United States.

### 1   **3.2.2.9   Stack**

2       Stack yang dipakai adalah stack yang paling baru, yaitu heroku-18. Stack ini dipilih karena  
3       Heroku masa kadaluarsa dukungan untuk stack ini yang paling lama (didukung sampai bulan  
4       April 2023). Alasan lain adalah komputer lokal yang digunakan untuk mengerjakan skripsi ini  
5       menggunakan lingkungan yang mirip dengan Heroku, yaitu menggunakan Ubuntu 18.04.

### 6   **3.2.2.10   Basis Data**

7       Basis data yang digunakan untuk skripsi ini adalah basis data Heroku Postgres dengan plan  
8       hobby-dev. Alasan utama basis data ini digunakan adalah karena penggunaan basis data lain  
9       seperti Heroku Redis dan Apache Kafka tidak memungkinkan. Basis data tersebut membutuhkan  
10      informasi akun kredit untuk verifikasi akun. Alasan lain adalah basis data ini adalah basis data  
11      yang langsung disediakan oleh Heroku.

12      Sebelumnya BlueTape menggunakan MySQL untuk basis datanya. Proses migrasi dari MySQL  
13      ke Heroku Postgres tidak rumit, karena menggunakan fitur Migration dari CodeIgniter. Namun,  
14      ada beberapa perubahan pada dokumen Migration. Perubahan-perubahan tersebut adalah :

- 15      • Menyelaraskan nama tabel karena sifat case sensitive pada Postgres
- 16      • Mengubah Replace menjadi Insert dan Update karena Replace tidak didukung oleh Postgres
- 17      • Mengubah tipe data kolom yang sebelumnya menggunakan DATETIME menjadi timestamp

### 18   **3.2.3   Sinkronisasi Email**

19      Awalnya sinkronisasi email dilakukan dengan memanfaatkan Gmail API. Namun, Gmail API  
20      membutuhkan token yang harus direfresh tiap periode tertentu. Sehingga peneliti perlu mencari  
21      alternatif lain. Peneliti memutuskan menggunakan PHP IMAP untuk melakukan sinkronisasi email.

22      Sebelum melakukan sinkronisasi email, email khusus untuk menampung email pengumuman  
23      harus dibuat terlebih dahulu. Email dibuat melalui provider email Gmail. Setelah email selesai  
24      dibuat, fitur IMAP perlu dinyalakan terlebih dahulu.

25      Proses sinkronisasi email dimulai dengan membuat koneksi IMAP ke email pengumuman tersebut.  
26      Dengan menggunakan koneksi IMAP yang telah didapat, email difilter dengan mencari email yang  
27      belum dibaca saja. Apabila hasil pencarian tidak kosong, maka setiap email pada hasil pencarian  
28      akan diperiksa pengirimnya. Pengirim email akan dinyatakan sebagai pemberi pengumuman  
29      yang valid apabila ia terdaftar di dalam daftar pengirim yang terverifikasi. Apabila sebuah email  
30      dinyatakan memiliki pengirim yang valid, maka informasi dari email tersebut akan diproses dan  
31      dimasukkan ke basis data.

32      Informasi yang perlu disimpan dari email pengumuman adalah alamat email pengirim, nama  
33      pengirim, tanggal email tersebut terkirim, subjek email, isi email, dan ketersediaan lampiran.  
34      Sebuah tabel baru diperlukan untuk menampung informasi ini. Tabel ini akan diakses saat halaman  
35      pengumuman akan ditampilkan. Setiap informasi dari email tersimpan, maka satu push message  
36      akan dikirim ke LINE.

37      Sinkronisasi email perlu dilakukan secara berkala dan otomatis. Pada skripsi ini sinkronisasi  
38      email dilakukan per hari dengan menggunakan cron dan add-on Heroku Scheduler.

### **3.2.4 Menghubungkan BlueTape dengan LINE**

Produk LINE yang digunakan untuk menghubungkan BlueTape dengan LINE adalah LINE Messaging API. Produk ini paling memenuhi kriteria fitur kolektor pengumuman, yaitu dapat mengirimkan push message.

Sebelum menghubungkan BlueTape dengan LINE, ada beberapa hal yang harus dilakukan terlebih dahulu di LINE developer console. Pertama, membuat akun LINE dan mendaftar sebagai developer di LINE developer console. Kedua, membuat provider di LINE developer console. Ketiga, membuat channel pada provider tersebut. Channel yang dibuat harus menggunakan plan Developer Trial karena plan ini yang memiliki fitur push message. Setelah channel terbuat, sebuah akun bot otomatis dibuat.

Pada channel, terdapat tiga bagian penting : channel access token, channel secret, dan webhook URL. Channel access token dan channel secret digunakan untuk autentikasi saat aplikasi BlueTape dan LINE berinteraksi. Kedua kode ini dapat diubah-ubah menggunakan tombol issue pada masing-masing kolom. Sedangkan webhook URL adalah alamat URL untuk menerima POST request berisi event-event yang terjadi pada bot. Contoh : event following yang terjadi saat ada akun LINE yang follow akun bot. Webhook URL dapat diubah-ubah, namun harus menggunakan alamat https. Selain itu, fitur csrf protection milik Codeigniter harus dimatikan pada alamat URL tersebut.

Pada skripsi ini, tidak semua event perlu ditangani. Event yang harus ditangani cukup follow event dan unfollow event. Saat follow event terjadi, user id dari akun follower akan disimpan di sebuah tabel di basis data. Apabila user id mengeblok akun bot sehingga unfollow event terjadi, maka user id tersebut dihapus dari tabel. Penyimpanan ini diperlukan karena user id diperlukan saat mengirim push message dan tidak ada fungsi pada line-bot-sdk untuk mendapatkan user id follower akun bot.



1

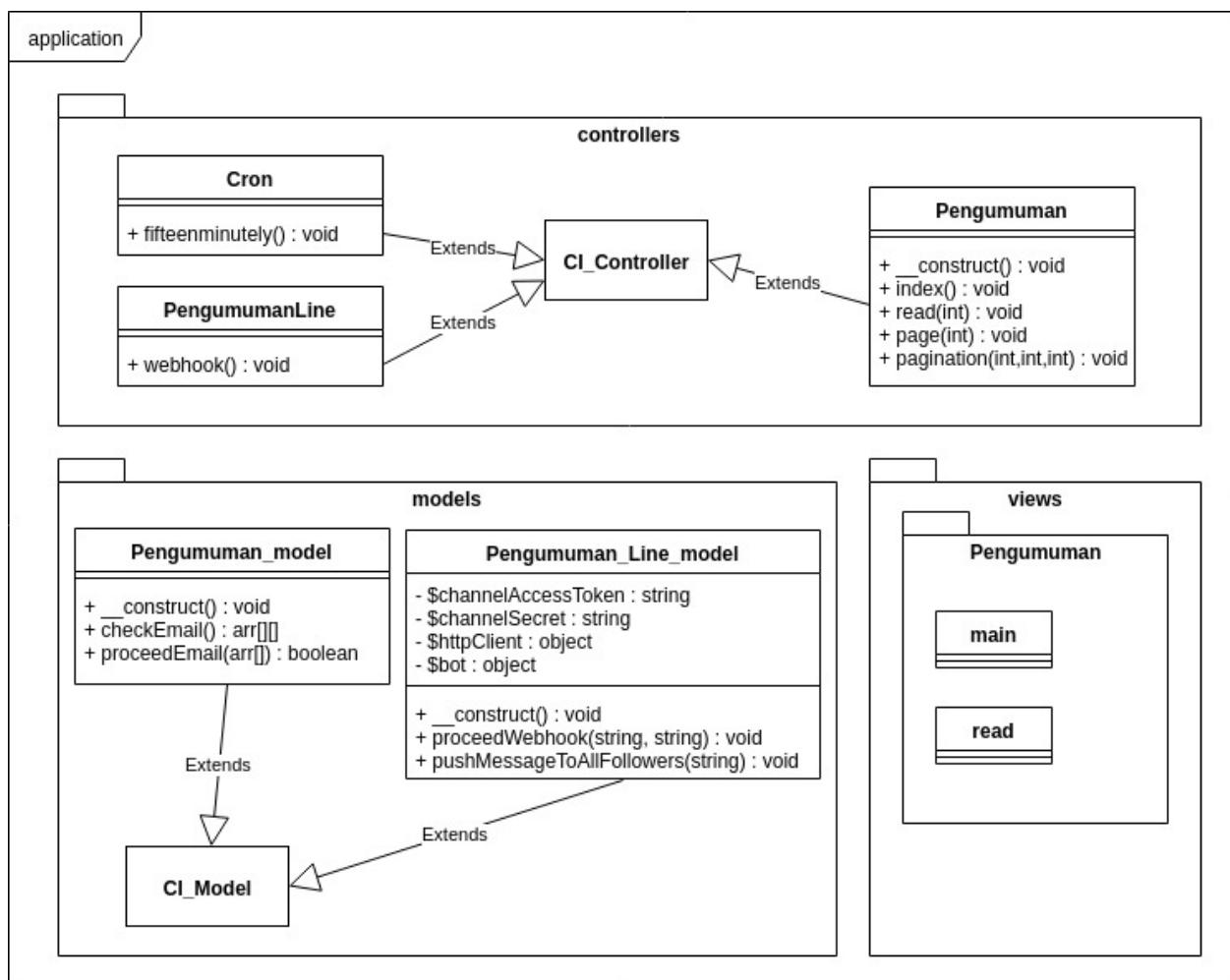
## BAB 4

2

### PERANCANGAN

3 Bab ini membahas perancangan fitur kolektor pengumuman. Pembahasan dibagi menjadi tiga  
 4 bagian, yaitu : perancangan kelas, perancangan basis data, dan perancangan antarmuka.

5 **4.1 Perancangan Kelas**



Gambar 4.1: Class diagram fitur kolektor pengumuman

6 Gambar 4.1 merupakan gambar diagram kelas yang dipakai untuk pembangunan fitur kolektor  
 7 pengumuman. Penjelasan untuk diagram kelas tersebut akan diberikan di subbagian. Penjelasan

1 dibagi menjadi tiga bagian : controller, model, dan view.

## 2 4.1.1 Model

3 Model yang digunakan ada dua, yaitu Pengumuman\_model dan Pengumuman\_Line\_model.

### 4 4.1.1.1 Pengumuman\_model

5 Pengumuman\_model berisi algoritma yang dibutuhkan oleh fitur kolektor pengumuman. Pengumuman\_model  
6 memiliki tiga method : \_\_construct, checkEmail, dan proceedEmail. Tabel 4.1, 4.2, dan 4.3  
7 menjelaskan secara rinci method-method tersebut.

Tabel 4.1: Rincian method \_\_construct

Nama Method	__construct
Parameter Input	-
Parameter Output	-
Tabel yang berhubungan	-
Kelas yang berhubungan	-
Deskripsi	Method ini digunakan untuk konstruksi
Algoritma	<ol style="list-style-type: none"> <li>1. Konstruksi dari parent.</li> <li>2. Load config auth dan modules.</li> </ol>

Tabel 4.2: Rincian method checkEmail

Nama Method	checkEmail
Parameter Input	-
Parameter Output	Array dua dimensi. Dimensi pertama mewakili satu email, sedangkan dimensi kedua mewakili informasi dari email tersebut.
Tabel yang berhubungan	-
Kelas yang berhubungan	-
Deskripsi	
Algoritma	<ol style="list-style-type: none"> <li>1. Membuka koneksi imap ke email pengumuman.</li> <li>2. Mencari email yang belum dibaca.</li> <li>3. Apabila ada email yang belum dibaca, maka email tersebut dan informasinya dimasukkan ke dalam array.</li> <li>4. Menutup koneksi imap dan mengembalikan array.</li> </ol>

Tabel 4.3: Rincian method proceedEmail

Nama Method	proceedEmail
Parameter Input	Array asosiatif berisi informasi dari sebuah email.
Parameter Output	Mengembalikan true apabila email termasuk email pengumuman dan mengembalikan false apabila email tidak termasuk email pengumuman.
Tabel yang berhubungan	Pengumuman
Kelas yang berhubungan	Pengumuman_Line_model
Deskripsi	Method ini berfungsi untuk memroses email pengumuman
Algoritma	<ol style="list-style-type: none"> <li>1. load config pengumuman</li> <li>2. Memeriksa pengirim setiap email. Apabila pengirim terdaftar di config pengirimTerverifikasi, maka email tersebut akan masuk ke tahap berikutnya.</li> <li>3. Tahap berikutnya adalah memasukkan informasi email ke dalam baris record di tabel Pengumuman dan mengirimkan pesan melalui Line. Pesan Line dikirimkan dengan bantuan method pushMessageToAllFollowers yang terdapat pada Pengumuman_Line_model.</li> <li>4. Mengembalikan true apabila sebelumnya email diidentifikasi sebagai email pengumuman dan mengembalikan false apabila sebelumnya email diidentifikasi tidak termasuk email pengumuman.</li> </ol>

#### 4.1.1.2 Pengumuman\_Line\_model

Pengumuman\_Line\_model berisi algoritma yang dibutuhkan untuk berkomunikasi dengan Line API. Algoritma tersebut sengaja tidak disatukan ke dalam kelas Pengumuman\_model karena menggunakan banyak package tambahan dan memiliki atribut-atribut yang tidak dibutuhkan semua method yang berada di Pengumuman\_model. Atribut-atribut tersebut adalah : \$channelAccessToken, \$channelSecret, \$httpClient, dan \$bot. Pengumuman\_Line\_model memiliki tiga method : \_\_construct, proceedWebhook, dan pushMessageToAllFollowers. Tabel 4.4, 4.5, dan 4.6 menjelaskan secara rinci method-method tersebut.

Tabel 4.4: Rincian method \_\_construct

Nama Method	__construct
Parameter Input	-
Parameter Output	-
Tabel yang berhubungan	-
Kelas yang berhubungan	-
Deskripsi	Method ini digunakan untuk konstruksi
Algoritma	<ol style="list-style-type: none"> <li>1. Konstruksi dari parent.</li> <li>2. Load config auth dan modules.</li> <li>3. Assign value untuk setiap atribut.</li> </ol>

Tabel 4.5: Rincian method proceedWebhook

Nama Method	proceedWebhook
Parameter Input	HTTP request body dan X Line Signature
Parameter Output	-
Tabel yang berhubungan	PengumumanLineFollowers
Kelas yang berhubungan	kelas-kelas di package LINE/LINEBot
Deskripsi	Method ini untuk memproses event yang masuk ke dalam method webhook yang terdapat di PengumumanLine
Algoritma	<ol style="list-style-type: none"> <li>1. Validasi signature dengan menggunakan method validateSignature yang membutuhkan parameter input HTTP request body dan X Line Signature.</li> <li>2. Jika signature valid, maka jenis event yang masuk akan dicek dan ditangani. Event yang wajib ditangani adalah FollowEvent dan UnfollowEvent. Apabila FollowEvent terjadi (ada user yang mengikuti bot atau membuka blokir bot), maka id user line tersebut akan dimasukkan ke tabel PengumumanLineFollowers. Apabila UnfollowEvent terjadi (ada user yang memblokir bot), maka id user line tersebut akan dihapus dari tabel PengumumanLineFollowers. Penyimpanan id user diperlukan untuk mengirim pesan ke pengikut bot.</li> </ol>

Tabel 4.6: Rincian method pushMessageToAllFollowers

Nama Method	pushMessageToAllFollowers
Parameter Input	Pesan yang ingin dikirimkan ke pengikut bot
Parameter Output	-
Tabel yang berhubungan	PengumumanLineFollowers
Kelas yang berhubungan	kelas-kelas di package LINE/LINEBot
Deskripsi	
Algoritma	<ol style="list-style-type: none"> <li>1. Membuat query untuk mendapatkan semua id user line pengikut bot dan memasukkan hasilnya ke suatu array.</li> <li>2. Membuat text message builder dengan parameter input pesan yang ingin dikirimkan ke pengikut bot.</li> <li>3. Melakukan multicast dengan parameter input array id user line yang akan dikirimkan pesan dan text message builder.</li> </ol>

#### **4.1.2 View**

View dibagi menjadi dua file php, yaitu main dan read. File main berfungsi untuk mengatur tampilan saat menampilkan daftar pengumuman. Sedangkan file read berfungsi untuk mengatur tampilan saat informasi dari satu email pengumuman ditampilkan.

#### **4.1.3 Controller**

Controller yang digunakan ada tiga, yaitu Cron, Pengumuman, dan PengumumanLine.

##### **4.1.3.1 Cron**

Controller Cron berfungsi untuk menjalankan perintah-perintah yang harus dijalankan pada jadwal tertentu. Method yang dimiliki hanya satu, yaitu : fifteenminutely(). Tabel 4.7 menjelaskan secara rinci method fifteenminutely().

Tabel 4.7: Rincian method fifteenminutely

Nama Method	fifteenminutely
Parameter Input	-
Parameter Output	-
Tabel yang berhubungan	-
Kelas yang berhubungan	Pengumuman_model
Deskripsi	Method yang perintah di dalamnya akan dijalankan setiap lima belas menit sekali. Untuk keperluan skripsi ini, method ini diisi dengan perintah untuk memeriksa email. Namun, apabila ada pengembangan lebih lanjut dan ada kebutuhan untuk menjalankan perintah setiap lima belas menit, maka isi method ini dapat ditambah.
Algoritma	<ol style="list-style-type: none"> <li>1. Menjalankan method checkEmail() yang terdapat pada Pengumuman_model.</li> <li>2. Memeriksa output dari Pengumuman_model yang berupa array. Apabila tidak null, maka setiap elemen pada array tersebut akan dijadikan input dari method proceedEmail() yang terdapat pada Pengumuman_model.</li> <li>3. Menjalankan method proceedEmail() yang terdapat pada Pengumuman_model.</li> </ol>

#### **4.1.3.2 Pengumuman**

Controller Pengumuman berfungsi untuk mengatur hubungan antara Pengumuman\_model dan view yang ada di package Pengumuman. Pengumuman\_model memiliki lima method, yaitu \_\_construct, index, read, page, dan pagination. Tabel 4.8, 4.9, 4.10, 4.11, dan 4.12 menjelaskan secara rinci method-method tersebut.

Tabel 4.8: Rincian method \_\_construct

Nama Method	__construct
Parameter Input	-
Parameter Output	-
Tabel yang berhubungan	-
Kelas yang berhubungan	-
Deskripsi	Method ini digunakan untuk konstruksi
Algoritma	<ol style="list-style-type: none"> <li>1. Konstruksi dari parent.</li> <li>2. Mengecek module yang diizinkan.</li> <li>3. Load library BlueTape</li> <li>4. Load model Pengumuman_model</li> <li>5. load database</li> </ol>

Tabel 4.9: Rincian method index

Nama Method	index
Parameter Input	-
Parameter Output	-
Tabel yang berhubungan	Pengumuman
Kelas yang berhubungan	-
Deskripsi	Mengatur halaman utama pengumuman
Algoritma	<ol style="list-style-type: none"> <li>1. Mengambil info user</li> <li>2. Memasang link untuk setiap pengumuman yang ditampilkan pada daftar pengumuman.</li> <li>3. Menampilkan halaman pertama.</li> </ol>

Tabel 4.10: Rincian method read

Nama Method	read
Parameter Input	Id pengumuman
Parameter Output	-
Tabel yang berhubungan	Pengumuman
Kelas yang berhubungan	-
Deskripsi	Mengatur halaman yang menampilkan detail pengumuman
Algoritma	<ol style="list-style-type: none"> <li>1. Membuat query untuk mendapatkan seluruh informasi yang dimiliki oleh id pengumuman yang diinput dan memasukkannya ke suatu array.</li> <li>2. Load view untuk read dan oper array tersebut ke view.</li> </ol>

Tabel 4.11: Rincian method page

Nama Method	page
Parameter Input	nomor halaman yang ingin ditampilkan
Parameter Output	-
Tabel yang berhubungan	-
Kelas yang berhubungan	-
Deskripsi	Menampilkan halaman pengumuman pada nomor halaman yang diinput.
Algoritma	<ol style="list-style-type: none"> <li>1. Mengatur batas maksimal jumlah pengumuman yang ditampilkan pada satu halaman.</li> <li>2. Memanggil method pagination dengan input nomor halaman, batas tersebut, dan id pengumuman yang akan ditampilkan di baris pertama.</li> </ol>

Tabel 4.12: Rincian method pagination

Nama Method	pagination
Parameter Input	Nomor halaman yang ingin ditampilkan, batas maksimal jumlah pengumuman yang ditampilkan pada satu halaman, dan id pengumuman yang akan ditampilkan di baris pertama.
Parameter Output	-
Tabel yang berhubungan	Pengumuman
Kelas yang berhubungan	-
Deskripsi	Menampilkan halaman pengumuman pada nomor halaman yang diinput.
Algoritma	<ol style="list-style-type: none"> <li>1. Membuat query untuk mengetahui jumlah pengumuman yang ada di tabel Pengumuman</li> <li>2. Membuat query untuk mendapatkan informasi dari pengumuman yang akan tampil di halaman tersebut dan memasukkannya ke suatu array.</li> <li>3. Load view untuk main dan oper jumlah pengumuman yang ada di tabel Pengumuman, array tersebut, nomor halaman yang sedang ditampilkan, dan batas maksimal jumlah pengumuman yang ditampilkan pada satu halaman ke view.</li> </ol>

#### <sup>1</sup> 4.1.3.3 PengumumanLine

<sup>2</sup> Controller PengumumanLine berfungsi untuk menerima webhook dari LineAPI. PengumumanLine  
<sup>3</sup> memiliki satu method yaitu webhook. Method ini sengaja dipisah dari controller Pengumuman  
<sup>4</sup> karena proteksi csrf perlu dibuka untuk menerima webhook. Tabel [4.13](#) menjelaskan method  
<sup>5</sup> tersebut secara rinci.

Tabel 4.13: Rincian method webhook

Nama Method	webhook
Parameter Input	Request berisi event dari Line API melalui POST
Parameter Output	-
Tabel yang berhubungan	-
Kelas yang berhubungan	Pengumuman_Line_model
Deskripsi	Method ini berfungsi untuk menerima request berisi event dari Line API melalui POST
Algoritma	<ol style="list-style-type: none"> <li>1. Memeriksa apakah request yang masuk melalui POST. Jika tidak, maka akan mengembalikan http response code 405.</li> <li>2. Mengambil request yang masuk dan menyimpannya ke suatu variabel.</li> <li>3. Memeriksa apakah ada HTTP_X_LINE_SIGNATURE pada POST yang masuk. Jika tidak ada, maka akan mengembalikan http response code 400.</li> <li>4. Jika ada, maka jalankan method proceedWebhook dengan parameter input variabel yang berisi request body dan X Line Signature.</li> </ol>

## 1 4.2 Perancangan Basis Data

2 Tabel-tabel yang sudah ada di BlueTape tidak diubah, namun ada dua tabel yang ditambahkan.  
 3 Kedua tabel tersebut adalah tabel Pengumuman dan tabel Line\_followers. Tabel pengumuman  
 4 berguna untuk menyimpan informasi dari email pengumuman. Sedangkan tabel Line\_followers  
 5 berguna untuk menyimpan user id dari follower akun bot BlueTape.

### 6 4.2.0.1 Tabel Pengumuman

Tabel 4.14: Perancangan Tabel Pengumuman

Atribut	Tipe Data	Constraint	PK*	FK*
id	int	-	Ya	Tidak
namaPengirim	VARCHAR	256	Tidak	Tidak
emailPengirim	VARCHAR	256	Tidak	Tidak
waktuTerkirim	timestamp	-	Tidak	Tidak
subjek	VARCHAR	256	Tidak	Tidak
isi	TEXT	256	Tidak	Tidak
ketersediaanLampiran	VARCHAR	1	Tidak	Tidak

7

8 \*PK = Primary Key

9 \*FK = Foreign Key

10

- 1 Keterangan atribut :
- 2 • **id** : Id pengumuman. Auto increment.
- 3 • **namaPengirim** : Nama pengirim email pengumuman.
- 4 • **emailPengirim** : Alamat email pengirim email pengumuman.
- 5 • **waktuTerkirim** : Waktu terkirim email pengumuman.
- 6 • **subjek** : Subjek email pengumuman.
- 7 • **isi** : Isi email pengumuman. Boleh kosong.
- 8 • **ketersediaanLampiran** : Jika email pengumuman memiliki lampiran maka atribut ini  
9 memiliki value 'Y'. Jika tidak, maka valuenya 'N'.

10 **4.2.0.2 Tabel PengumumanLineFollowers**

Tabel 4.15: Perancangan Tabel PengumumanLineFollowers

Atribut	Tipe Data	Constraint	PK*	FK*
userId	VARCHAR	256	Ya	Tidak

- 11
- 12 \*PK = Primary Key
- 13 \*FK = Foreign Key
- 14
- 15 Keterangan atribut :
- 16 • **userId** : User id dari akun LINE yang follow akun bot BlueTape.

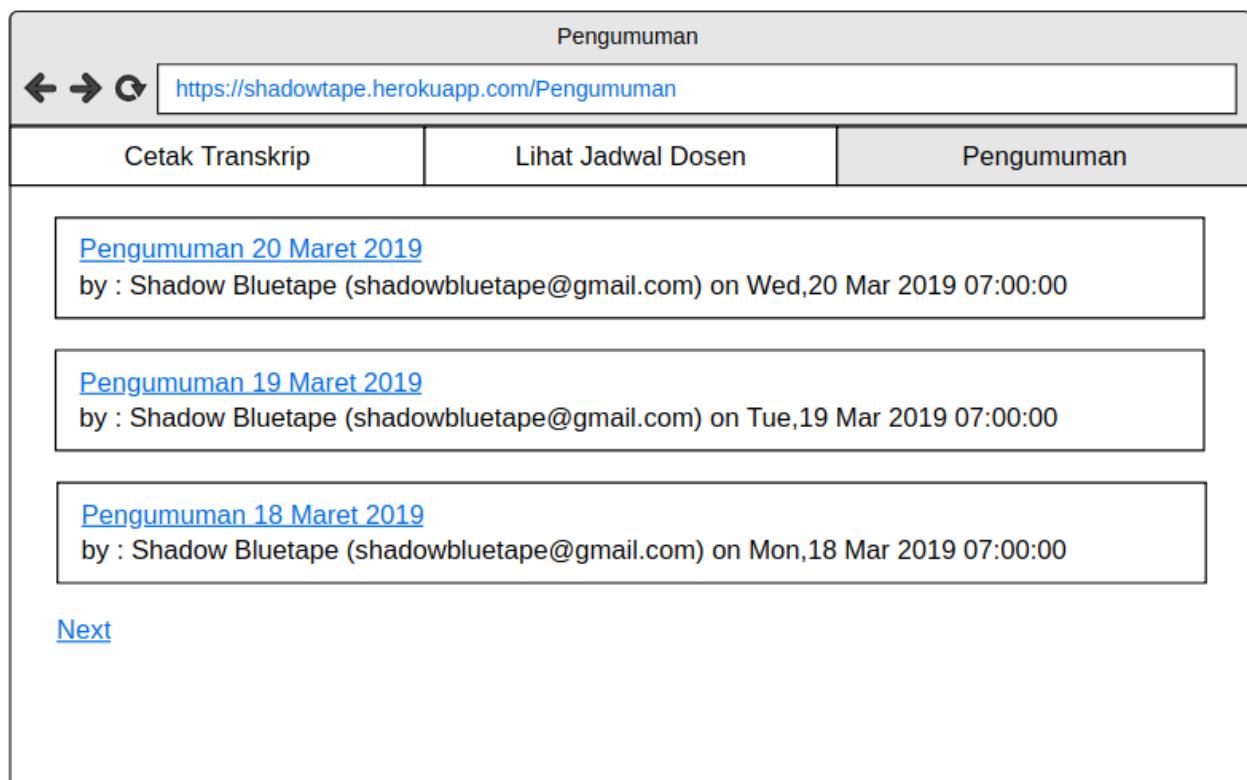
17 **4.3 Perancangan Antarmuka**

18 Bagian ini membahas perancangan antarmuka untuk fitur kolektor pengumuman pada BlueTape  
19 dan bot BlueTape.

20 **4.3.1 Perancangan Antarmuka pada BlueTape**

21 Antarmuka yang digunakan fitur kolektor pengumuman ada dua, yaitu main dan read. Penjelasan  
22 untuk desain kedua antarmuka tersebut terdapat di subbagian.

1 4.3.1.1 main



Gambar 4.2: Mockup antarmuka main

- 2 Antarmuka main menampilkan daftar pengumuman. Jumlah pengumuman yang ditampilkan di  
3 satu halaman dibatasi pada angka tertentu. Di halaman main terdapat navigasi next dan prev untuk  
4 menampilkan daftar selanjutnya dan sesudahnya. Gambar 4.2 menampilkan mockup antarmuka  
5 main.

### 1 4.3.1.2 read

The mockup shows a web-based application interface for viewing announcements. At the top, there's a header bar with the title "Pengumuman" and three navigation icons: back, forward, and refresh. Below the header is a URL bar containing the address <https://shadowtape.herokuapp.com/Pengumuman/read/4>. The main content area is divided into three horizontal sections: "Cetak Transkrip", "Lihat Jadwal Dosen", and "Pengumuman". The "Pengumuman" section contains the following text:

Pengumuman 20 Maret 2019  
by : Shadow Bluetape (shadowbluetape@gmail.com) on Wed,20 Mar 2019 07:00:00  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio euismod sit amet. Quisque laoreet sem sit amet orci ullamcorper at ultricies metus viverra. Pellentesque arcu mauris, malesuada quis ornare accumsan, blandit sed diam.  
\*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung email student Anda untuk mengunduhnya.

Gambar 4.3: Mockup antarmuka read

- 2 Antarmuka main menampilkan informasi detil dari pengumuman. Apabila pengumuman memiliki  
3 lampiran, maka tulisan "\*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung email  
4 student Anda untuk mengunduhnya." akan muncul. Jika tidak ada, maka tulisan tersebut tidak  
5 akan muncul. Gambar 4.3 menampilkan mockup antarmuka read.

### 1 4.3.2 Perancangan Antarmuka pada Bot BlueTape



Gambar 4.4: Mockup antarmuka bot BlueTape

2 Gambar 4.4 menampilkan mockup antarmuka bot BlueTape. Chat pertama merupakan pesan  
3 yang akan ditampilkan saat user baru mengikuti bot atau membuka blokir bot. Chat kedua  
4 merupakan contoh pesan yang akan ditampilkan jika ada pengumuman baru yang masuk ke  
5 BlueTape. Chat terakhir merupakan pesan balasan jika user mengirimkan pesan ke bot dalam  
6 bentuk apapun (teks, sticker, gambar, video, dan suara).



1

## BAB 5

2

### IMPLEMENTASI DAN PENGUJIAN

3

Bab ini membahas proses implementasi dan proses pengujian fitur kolektor pengumuman.

4

#### 5.1 Implementasi

5

Bagian ini membahas implementasi dari perancangan yang telah dilakukan di Bab 4.

6

##### 5.1.1 Lingkungan Pengembangan

7

Berikut spesifikasi perangkat keras dan perangkat lunak yang dipakai untuk pengembangan pada skripsi ini:

8

###### Spesifikasi Perangkat keras

9

- Processor Intel® Celeron(R) CPU 1007U @ 1.50GHz x 2
- Graphics Intel® Ivybridge Mobile
- RAM 8 GB
- Harddisk 500GB SATA

10

###### Spesifikasi Perangkat lunak

11

- Sistem Operasi Ubuntu 18.04.1 LTS 64-bit
- Visual Studio Code version 1.31.0
- apache2 -v
- PHP 7.2.10 (cli)
- Composer version 1.7.2
- pgAdmin4 version 2.1 (Application Mode: Desktop)
- psql (PostgreSQL) 10.6
- heroku/7.21.0 linux-x64 node-v11.9.0

### 1 5.1.2 Implementasi Basis Data

2 Pada pembangunan fitur kolektor pengumuman, ada dua tabel yang ditambahkan. Kedua  
3 tabel tersebut adalah tabel Pengumuman dan tabel PengumumanLineFollowers. Pembuatan tabel  
4 menggunakan dua *file migration* terpisah: *file* 20181011103200\_Pengumuman\_Initial.php dan  
5 20190210224400\_PengumumanLineFollowers\_initial.php.

### 6 5.1.3 Implementasi Kelas

7 Pada pembangunan fitur kolektor pengumuman, dibuat kelas-kelas berikut:

- 8 • Kelas *model* Pengumuman\_model

9 Pengumuman\_model merupakan kelas yang berisi algoritma yang dibutuhkan oleh fitur  
10 kolektor pengumuman.

- 11 • Kelas *model* Pengumuman\_Line\_model

12 Pengumuman\_model merupakan kelas yang dikhkususkan untuk algoritma untuk menghubungkan  
13 BlueTape dengan LINE API.

- 14 • Kelas *controller* Cron

15 Cron merupakan kelas yang berfungsi untuk menjalankan perintah-perintah yang harus  
16 dijalankan pada jadwal tertentu. Pada skripsi ini perintah yang dijadwalkan adalah memeriksa  
17 *email*. Pada tahap perancangan, perintah untuk memeriksa *email* dijadwalkan tiap lima  
18 belas menit. Pada tahap implementasi, perintah tersebut dijadwalkan lebih jarang karena  
19 pemakaian *dyno free* dibatasi. Jadwal baru untuk perintah terebut adalah setiap hari pada  
20 pukul 12.00 siang.

- 21 • Kelas *controller* Pengumuman

22 Pengumuman merupakan kelas yang berfungsi untuk mengatur hubungan antara Pengumuman\_model  
23 dan *view* yang ada di *package* Pengumuman.

- 24 • Kelas *controller* PengumumanLine

25 Pengumuman\_Line merupakan kelas yang berfungsi untuk menerima *webhook* dari Line API.

26 Untuk mendukung kinerja kelas-kelas tersebut, dibuat juga *file*:

- 27 • *File view* main.php.

28 *File* ini digunakan untuk mengatur tampilan halaman utama Pengumuman.

- 29 • *File view* read.php.

30 *File* ini digunakan untuk mengatur tampilan halaman saat detail pengumuman ditampilkan.

- 31 • *File config* pengumuman.php.

32 *File* ini digunakan untuk menyimpan daftar pengirim pengumuman yang terverifikasi.

- 33 • *File migration* 20181011103200\_Pengumuman\_Initial.php

34 *File* ini digunakan untuk membuat tabel Pengumuman.

- 1     • *File migration 20190210224400\_PengumumanLineFollowers\_initial.php.*

2       *File ini digunakan untuk membuat tabel PengumumanLineFollowers.*

3       Selain *file yang telah disebutkan*, ada beberapa *file* yang isinya harus diubah:

- 4     • *file config database.php*

5       Pada *file* ini, informasi yang berkaitan dengan basis data disesuaikan dengan informasi basis  
6       data yang dipakai di skripsi ini.

- 7     • *file config modules.php*

8       Pada *file* ini, ditambahkan *modules* Pengumuman pada *config 'modules'*.

- 9     • *file config routes.php*

10       Pada *file* ini ditambahkan *routing* berikut:

11       

`$route['pengumuman/page-(:num)'] = '/pengumuman/page/$1';`

## 14    **5.2 Pengujian**

### 15    **5.2.1 Lingkungan Pengujian**

16       Berikut spesifikasi yang dipakai untuk pengujian pada skripsi ini:

- 17     • Heroku dengan spesifikasi :

18       – *Region: United States*

19       – *Stack: heroku-18*

20       – *Framework: PHP*

21       – Ukuran maksimum *slug: 500 MiB*

22       – Heroku Git URL: [urlhttps://git.heroku.com/shadowtape.git](https://git.heroku.com/shadowtape.git)

23       – *Buildpack: heroku/php*

24       – *Domain: <https://shadowtape.herokuapp.com/>*

25       – *Dyno Type: Free Dynos*

26       – *Add-ons* Heroku Postgres dan Heroku Scheduler

- 27     • Akun *bot LINE@* dengan *plan Developer*

### 28    **5.2.2 Pengujian Fungsional**

29       Pengujian fungsional dilakukan dengan metode *black box testing*. Berikut adalah hasil pengujianya

30    :

- 31     • Pengujian *Filter Email* Pengumuman

32       Pengujian ini bertujuan untuk menguji apakah *filter email* pengumuman berfungsi dengan  
33       baik. *Email* yang dikirim di pengujian ini memiliki subjek. Hasil pengujian dapat dilihat di  
34       Tabel 5.1.

Tabel 5.1: Pengujian Filter *Email* Pengumuman

Aksi	Reaksi yang diharapkan	Reaksi Perangkat Lunak
Mengirimkan <i>email</i> dengan <i>email</i> yang terdaftar lalu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman.
Mengirimkan <i>email</i> dengan <i>email</i> yang tidak terdaftar lalu menjalankan <i>Cron</i> .	<i>Email</i> tidak masuk ke basis data.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> tidak masuk ke basis data.

1

2 • Pengujian Mengirim *Email* dengan Isi *Email* yang Variatif

3 Pengujian ini bertujuan untuk menguji apakah isi *email* yang ditampilkan sesuai dengan  
 4 yang diharapkan. Pengujian ini dilakukan dengan mengirimkan beberapa *email* dengan  
 5 isi yang berbeda melalui salah satu *email* yang terdaftar di BlueTape, yaitu alamat *email*  
 6 [shadowbluetape@gmail.com](mailto:shadowbluetape@gmail.com). Hasil pengujian dapat dilihat di Tabel 5.2.

Tabel 5.2: Pengujian Mengirim *Email* dengan Isi *Email* yang Variatif

Aksi	Reaksi yang diharapkan	Reaksi Perangkat Lunak
Mengirimkan <i>email</i> tanpa subjek lalu menjalankan <i>Cron</i> .	<i>Email</i> tidak masuk ke basis data.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> tidak masuk ke basis data.
Mengirimkan <i>email</i> tanpa isi lalu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman.
Mengirimkan <i>email</i> dengan subjek dan isi lalu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman.
Mengirimkan <i>email</i> balasan lalu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman.

Mengirimkan <i>email</i> terusan lalu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman.
Mengirimkan <i>email</i> dengan lampiran lalu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman. Di bawah isi pengumuman, ada keterangan "*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung <i>email</i> student Anda untuk mengunduhnya.".	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman. Di bawah isi pengumuman, keterangan "*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung <i>email</i> student Anda untuk mengunduhnya." berhasil ditampilkan.
Mengirimkan <i>email</i> yang terdapat sisipan lampiran berupa gambar di isi <i>email</i> lalu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman. Di bawah isi pengumuman, ada keterangan "*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung <i>email</i> student Anda untuk mengunduhnya.". Isi pesan harus masih lengkap walaupun gambar tidak akan berhasil ditampilkan (karena file gambar tidak bisa disimpan di server).	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman. Di bawah isi pengumuman, keterangan "*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung <i>email</i> student Anda untuk mengunduhnya." berhasil ditampilkan. Isi pesan lengkap. Sesuai ekspektasi, gambar tidak bisa ditampilkan. Namun, ada keterangan alt yang berisi nama file.

Mengirimkan <i>email</i> yang isinya memakai berbagai jenis pemformatan yang bisa dilakukan di gmail, emoji yang disediakan gmail, dan sisipan URL. Setelah itu menjalankan <i>Cron</i> .	<i>Email</i> masuk ke basis data dan pengumuman ditampilkan di menu pengumuman. Isi <i>email</i> lengkap. Pemformatan tetap sama. Emoji tidak diharapkan bisa ditampilkan. Sisipan URL dapat ditampilkan dan URL dapat dikunjungi.	Reaksi sesuai dengan yang diharapkan. <i>Email</i> masuk ke basis data dan pengumuman dapat dilihat di menu pengumuman. Isi <i>email</i> lengkap. Pemformatan tetap sama. Emoji dapat ditampilkan. Beberapa emoji berubah bentuk tapi tetap memiliki bentuk yang sama. Beberapa emoji persis sama dengan yang ada di isi <i>email</i> yang asli. Sisipan URL dapat ditampilkan dan URL dapat dikunjungi.
---	--	--

1 • Pengujian Notifikasi LINE

2 Pengujian ini bertujuan untuk menguji apakah notifikasi LINE dapat terkirim ke bot LINE.

3 Tabel 5.3.

Tabel 5.3: Pengujian Notifikasi LINE

Aksi	Reaksi yang diharapkan	Reaksi Perangkat Lunak
Follow akun bot.	User Id LINE tercatat di basis data.	Reaksi sesuai dengan yang diharapkan. User Id LINE berhasil tercatat di basis data.
Mengirimkan <i>email</i> dengan <i>email</i> yang terdaftar lalu menjalankan <i>Cron</i> . <i>Email</i> harus memiliki subjek.	Setelah <i>Cron</i> sukses dijalankan, notifikasi LINE dari akun bot muncul.	Reaksi sesuai dengan yang diharapkan. Notifikasi LINE dari akun bot muncul.
Membuka pesan LINE yang masuk. Setelah itu membuka URL yang tercantum di pesan.	URL dapat dibuka. Apabila belum <i>login</i> ke BlueTape, pengguna akan diarahkan ke menu <i>login</i> . Setelah <i>login</i> , pengguna diarahkan kembali ke URL tersebut.	Reaksi sesuai dengan yang diharapkan. URL dapat dibuka. Apabila belum <i>login</i> ke BlueTape, pengguna akan diarahkan ke menu <i>login</i> . Setelah <i>login</i> , pengguna diarahkan kembali ke URL tersebut.

### 1 5.2.3 Pengujian Eksperimental

Course: AIF182100.04 Analisis dan Desain Perangkat Lunak | Universitas Katolik Parahyangan [ID] | <https://ide.unpar.ac.id/course/view.php?id=18053#section-14>

Untuk Mahasiswa  
Rekan-rekan mahasiswa,  
Terkait skripsi salah satu bimbingan saya, **Ellena / Kolektor Pengumuman Informatika**, saya memohon bantuan untuk melakukan pengujian eksperimental. Caranya, dengan add friend @ibz3613t atau scan QR code di bawah ini menggunakan aplikasi LINE.

Saat ada pengumuman dari saya / dosen lain, maka Anda akan menerima notifikasi di aplikasi LINE Anda.  
Supaya menarik, kisi-kisi UAS akan saya kirimkan melalui aplikasi ini.  
Nantinya, Anda akan diminta untuk mengisi kuisioner di alamat: <https://forms.gle/e7awCboE94kLEVlZ9>

Untuk Dosen  
Kirimkan e-mail ke shadowbluetape@gmail.com untuk memberikan pengumuman kepada mahasiswa dari akun e-mail @unpar.ac.id. Beberapa catatan:  
1. Untuk saat ini, mohon gunakan web-based GMail. Jika menggunakan mail client sepertinya masih ada bug, konten kosong.  
2. Karena alasannya biaya, pengumuman akan dikirimkan ke LINE setiap jam 12 siang (tidak instan)  
3. Attachment tidak akan muncul, tidak disimpan di server  
Nantinya, Anda akan diminta untuk mengisi kuisioner di alamat <https://forms.gle/e7awCboE94kLEVlZ9>

Gambar 5.1: Pengumuman Ujian Eksperimental

2 Pengujian eksperimental untuk aplikasi ini dilakukan pada mata kuliah AIF182100 (Analisis  
3 dan Desain Perangkat Lunak) kelas B dimana dosen kelas tersebut membuat pengumuman kepada  
4 mahasiswa untuk melakukan *add friend* dan menunggu pengumuman dari dosen. Pengumuman  
5 dapat dilihat pada Gambar 5.1. Berikut tata cara pengujianya:

- 6 1. Mahasiswa menambahkan bot Shadowtape dengan mencari id LINE "@ibz3613t" atau memindai  
7 kode QR (Gambar 5.2).

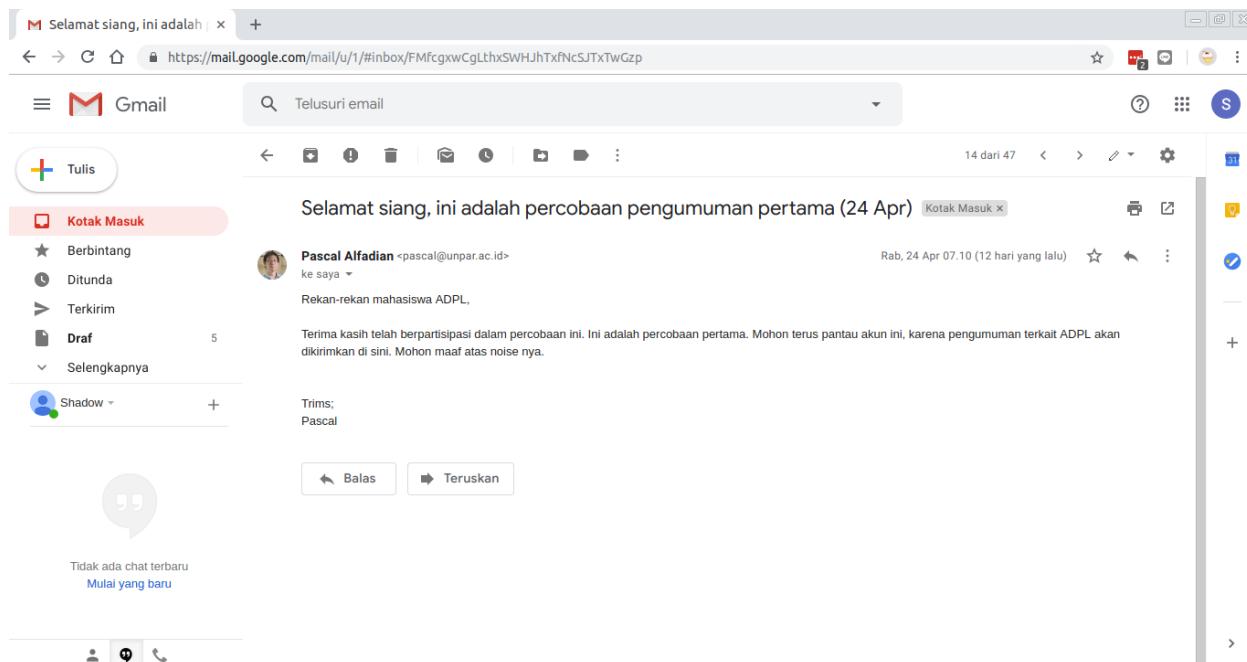


Gambar 5.2: Kode QR Bot Shadowtape

- 8 2. Dosen mengirimkan *email* berisi pengumuman ke shadowbluetape@gmail.com menggunakan  
9 alamat *email* yang terdaftar di BlueTape, yaitu alamat *email* dengan domain unpar.  
10 3. Apabila dosen mengirimkan *email*, maka notifikasi LINE dari akun Shadowtape akan diterima  
11 mahasiswa pada pukul 12.00 siang setelah *email* tersebut dikirim.

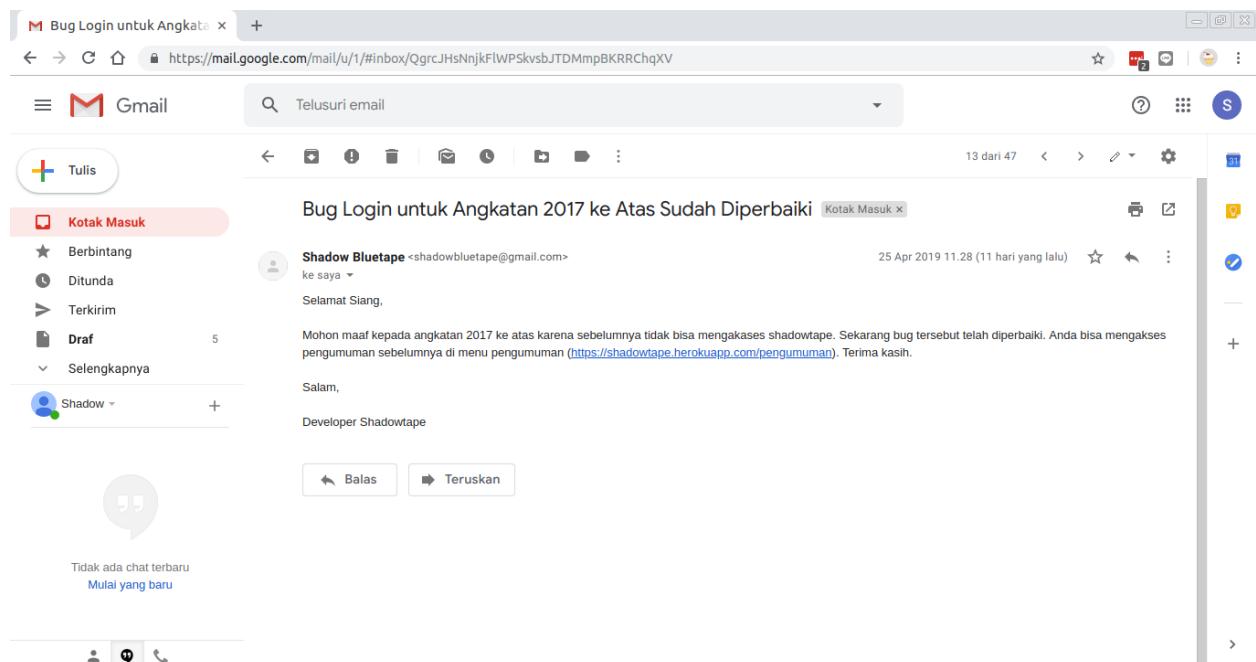
- 1     4. Apabila mahasiswa menerima notifikasi tersebut, mahasiswa akan menerima pesan yang terdapat subjek *email* dan URL di dalamnya. URL tersebut mengarah ke halaman detail pengumuman pada domain <http://shadowtape.herokuapp.com/>.
- 2
- 3
- 4     5. Apabila mahasiswa mengklik URL tersebut dan belum *login*, mahasiswa akan diarahkan ke halaman *login* sebelum diarahkan kembali ke URL tersebut.
- 5
- 6     6. Setelah masa pengujian berakhir, dosen dan mahasiswa mengisi kuesioner pada URL : <https://forms.gle/e7awCboE94kLEVLZ9>.
- 7

### 8     5.2.3.1 Pelaksanaan Pengujian



Gambar 5.3: *Email* pada tanggal 24 April 2019 oleh Bapak Pascal

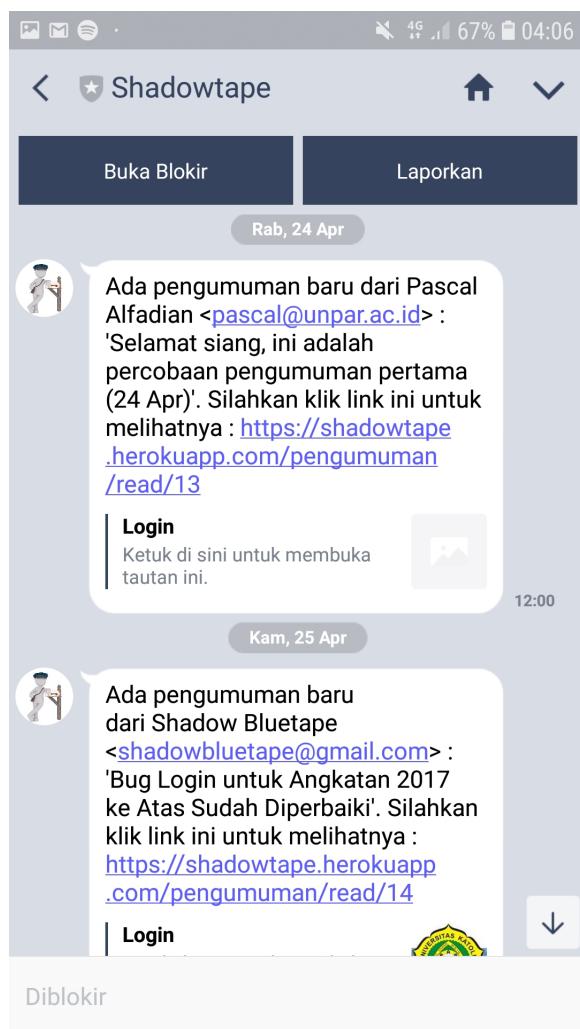
9     Pada tanggal 24 April 2019 pukul 07.10, Bapak Pascal Alfadian sebagai dosen mata kuliah  
10    AIF182100 (Analisis dan Desain Perangkat Lunak) kelas B mengirimkan *email* ke alamat *email*  
11    [shadowbluetape@gmail.com](mailto:shadowbluetape@gmail.com). Isi *email* dapat dilihat pada Gambar 5.3.



Gambar 5.4: *Email* pada tanggal 25 April 2019 oleh peneliti

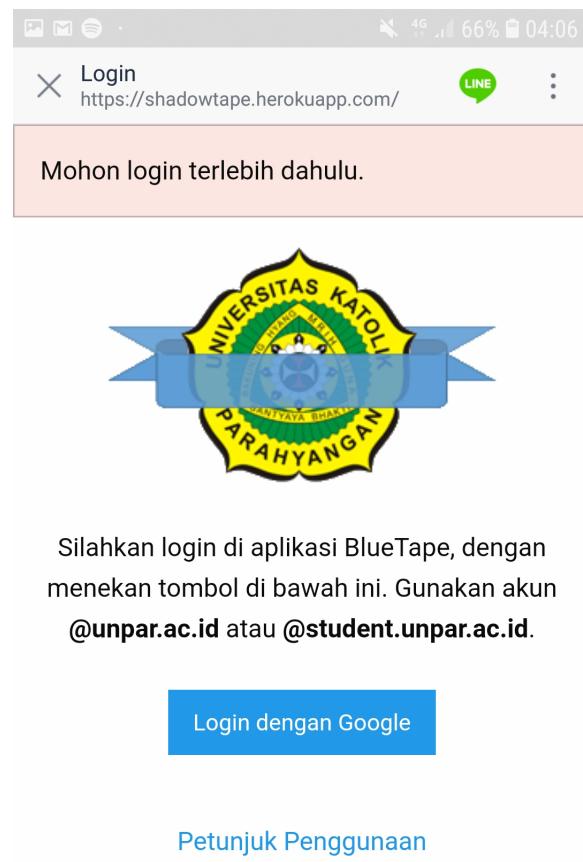
1      Pada tanggal 25 April 2019, peneliti menerima laporan bahwa ada masalah *login* untuk angkatan  
2      2017 ke atas. Peneliti menemukan masalah tersebut terjadi karena *repository* skripsi ini ((<https://github.com/EllenaAngelica/BlueTape>)) tertinggal 4 *commit* dari *repository* asal (<https://github.com/ftisunpar/BlueTape>). Peneliti memperbaiki masalah tersebut dengan melakukan  
4      *git pull*. Pada hari tersebut pukul 11.28, peneliti memakai *email* [shadowbluetape@gmail.com](mailto:shadowbluetape@gmail.com) ke  
5      alamat *email* ke [shadowbluetape@gmail.com](mailto:shadowbluetape@gmail.com) untuk memberitahu masalah tersebut telah diperbaiki  
6      sekaligus melakukan percobaan kedua. Isi *email* dapat dilihat pada Gambar 5.4.  
7

### 1 5.2.3.2 Hasil Pengujian



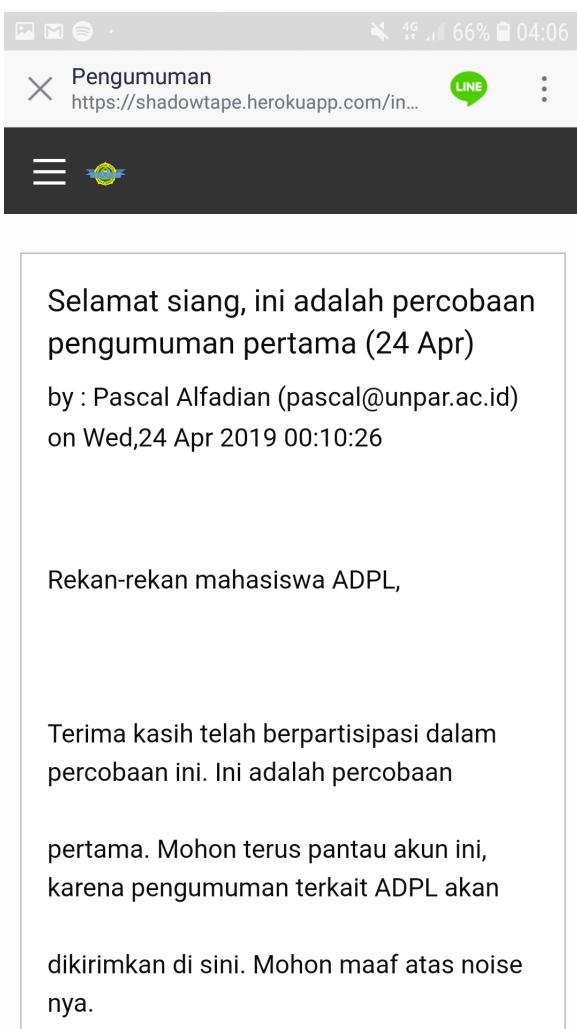
Gambar 5.5: Tampilan pesan LINE setelah pengumuman disebar

- 2 Gambar 5.5 menampilkan tampilan pesan LINE yang masuk setelah pengumuman disebar.
- 3 Karena peneliti tidak sengaja memblokir bot Shadowtape dan tidak dapat membuka blokirnya,
- 4 tampilan pada gambar tersebut memiliki tanda "Diblokir".



Gambar 5.6: Tampilan BlueTape setelah URL dibuka dan pengguna belum *login*

- <sup>1</sup> Gambar 5.6 menampilkan tampilan BlueTape setelah URL dibuka dan pengguna belum *login*.



Gambar 5.7: Tampilan URL pengumuman yang diumumkan oleh Pak Pascal Alfadian

- <sup>1</sup> Gambar 5.7 menampilkan tampilan URL pengumuman yang diumumkan oleh Pak Pascal
- <sup>2</sup> Alfadian.



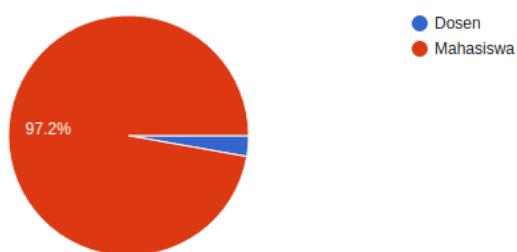
Gambar 5.8: Tampilan URL pengumuman yang diumumkan oleh peneliti

1 Gambar 5.8 menampilkan tampilan URL pengumuman yang diumumkan oleh peneliti.

2 **5.2.3.3 Hasil Kuesioner**

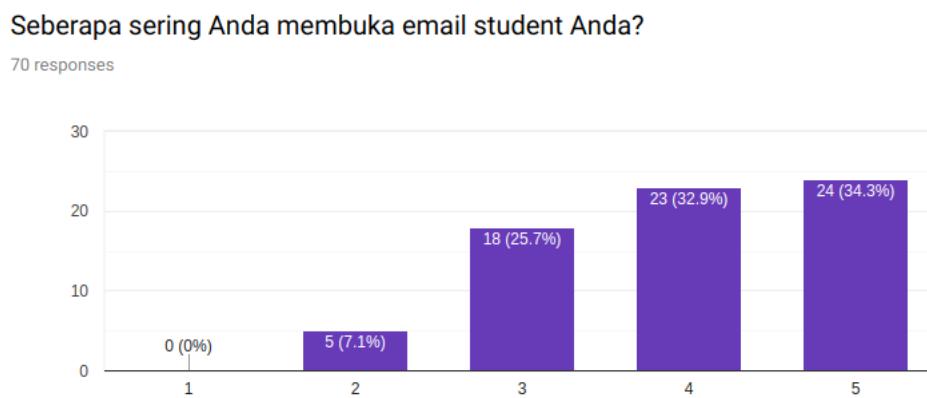
Apakah Anda dosen atau mahasiswa?

72 responses



Gambar 5.9: Diagram profil responden

- 1 Setelah masa pengujian berakhir, responden yang mengisi kuesioner adalah 72 responden yang  
 2 terdiri dari 70 mahasiswa dan 2 dosen. Diagram profil responden dapat dilihat pada Gambar 5.9.



Gambar 5.10: Diagram penggunaan *email* di kalangan mahasiswa

- 3 Pada pertanyaan "Seberapa sering Anda membuka *email student* Anda?", mayoritas mahasiswa  
 4 menjawab 5 (sering sekali). Jumlah mahasiswa yang menjawab 5 (sering sekali) adalah 24 orang.  
 5 Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.10.

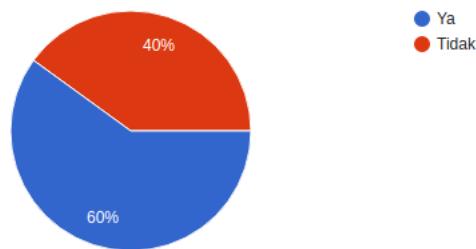


Gambar 5.11: Diagram penggunaan LINE di kalangan mahasiswa

- 6 Pada pertanyaan "Seberapa sering Anda membuka aplikasi LINE?", mayoritas mahasiswa  
 7 menjawab 5 (sering sekali). Jumlah mahasiswa yang menjawab 5 (sering sekali) adalah 53 orang.  
 8 Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.11.

Apakah Anda pernah menerima notifikasi pemberitahuan dari akun bot shadowtape pada jam 12 siang setelah mengikuti akun tersebut?

70 responses

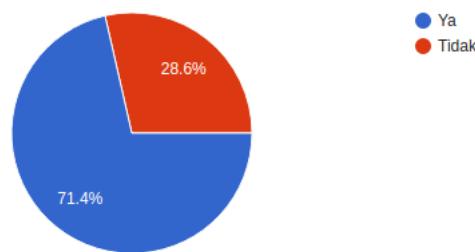


Gambar 5.12: Diagram mahasiswa yang menerima notifikasi LINE

- 1 Pada pertanyaan "Apakah Anda pernah menerima notifikasi pemberitahuan dari akun bot  
2 shadowtape pada jam 12 siang setelah mengikuti akun tersebut?", 42 mahasiswa menjawab "Ya"  
3 dan 28 mahasiswa menjawab "Tidak". Diagram jawaban untuk pertanyaan ini dapat dilihat  
4 pada Gambar 5.12. Mahasiswa yang menjawab "Tidak" belum tentu telah mengikuti *bot* karena  
5 jumlah maksimal teman akun LINE@ shadowtape adalah 50 akun. Peneliti memperkirakan jumlah  
6 mahasiswa yang tidak mendapatkan notifikasi LINE setelah mengikuti bot LINE tidak lebih dari 8  
7 mahasiswa.

Apakah Anda dapat membuka url yang dicantumkan di notifikasi pengumuman?

42 responses



Gambar 5.13: Diagram mahasiswa yang dapat membuka URL pengumuman

- 8 Pada pertanyaan "Apakah Anda dapat membuka url yang dicantumkan di notifikasi pengumuman?",  
9 30 mahasiswa menjawab "Ya" dan 12 mahasiswa menjawab "Tidak". Diagram jawaban untuk  
10 pertanyaan ini dapat dilihat pada Gambar 5.13. Peneliti menemukan beberapa mahasiswa yang  
11 menjawab "Tidak" adalah mahasiswa angkatan 2017 yang masih tidak dapat *login* setelah membaca  
12 kolom saran.

Jika Anda belum login saat membuka url yang dicantumkan di notifikasi pengumuman, apakah Anda diarahkan kembali ke url tersebut?

30 responses

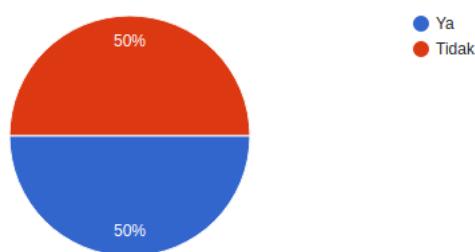


Gambar 5.14: Diagram mahasiswa yang diarahkan kembali ke URL pengumuman setelah *login*

- 1 Pada pertanyaan "Jika Anda belum login saat membuka url yang dicantumkan di notifikasi pengumuman, apakah Anda diarahkan kembali ke url tersebut?", 21 mahasiswa menjawab "Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan." dan 9 mahasiswa menjawab "Tidak, saya diarahkan ke halaman utama atau halaman lain.". Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.14.

Apakah Anda dapat melihat pengumuman yang Anda kirimkan di menu pengumuman shadowtape  
(<https://shadowtape.herokuapp.com/pengumuman>) pada jam 12 siang setelah Anda mengirimkan pengumuman?

2 responses

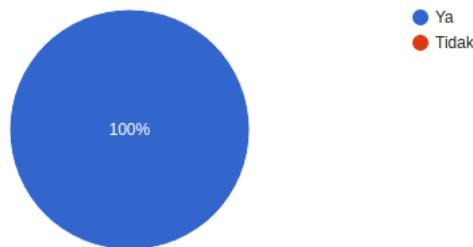


Gambar 5.15: Diagram dosen yang dapat melihat pengumuman yang diumumkannya

- 6 Pada pertanyaan "Apakah Anda dapat melihat pengumuman yang Anda kirimkan di menu pengumuman shadowtape (<https://shadowtape.herokuapp.com/pengumuman>) pada jam 12 siang setelah Anda mengirimkan pengumuman?", 1 dosen menjawab "Ya" dan 1 dosen menjawab "Tidak".
- 9 Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.15. Dosen yang menjawab "Ya" adalah Pak Pascal Alfadian, sedangkan dosen yang menjawab "Tidak" adalah Pak Keenan Adiwijaya Leman. Pak Pascal pernah mengirimkan *email* ke [shadowbluetape@gmail.com](mailto:shadowbluetape@gmail.com) pada tanggal 24 April 2019, sedangkan Pak Keenan tidak pernah.

Apakah judul pengumuman sesuai dengan subjek email yang Anda kirimkan?

1 response

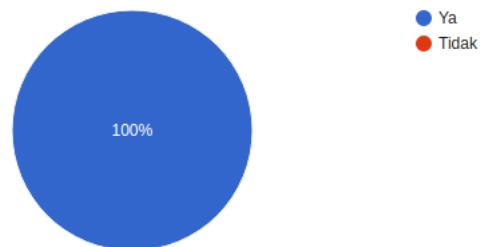


Gambar 5.16: Diagram dosen yang judul pengumumannya sesuai dengan subjek *email* yang ia kirim

- 1 Pada pertanyaan "Apakah judul pengumuman sesuai dengan subjek *email* yang Anda kirimkan?",
- 2 Pak Pascal Alfadian menjawab "Ya". Diagram jawaban untuk pertanyaan ini dapat dilihat pada
- 3 Gambar 5.16.

Apakah isi email sesuai dengan yang Anda kirimkan?

1 response

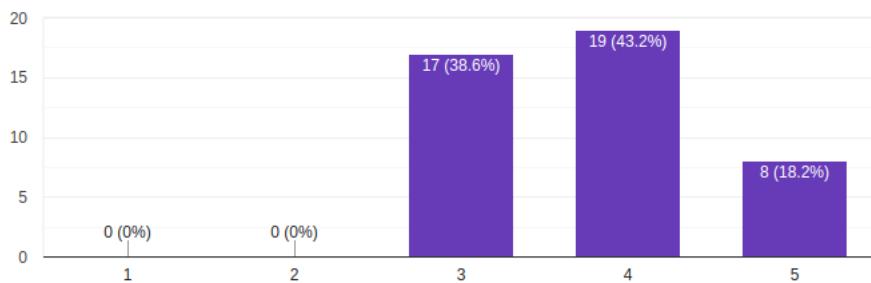


Gambar 5.17: Diagram dosen yang isi pengumumannya sesuai dengan isi *email* yang ia kirim

- 4 Pada pertanyaan "Apakah isi *email* sesuai dengan yang Anda kirimkan?", Pak Pascal Alfadian
- 5 menjawab "Ya". Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.17.

**Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya ingin menggunakan fitur ini" ?**

44 responses

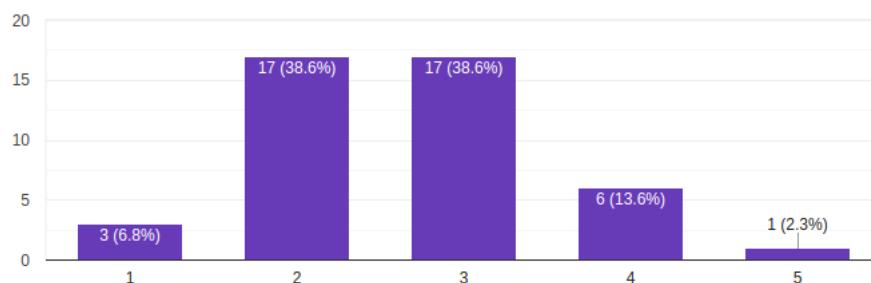


Gambar 5.18: Diagram jawaban pertanyaan *System Usability Scale* yang pertama

- 1 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya ingin menggunakan fitur ini" ?", mayoritas responden menjawab 4 (setuju).
- 2 Jumlah responden yang menjawab 4 (setuju)
- 3 adalah 19 orang. Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.18.

**Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu rumit" ?**

44 responses

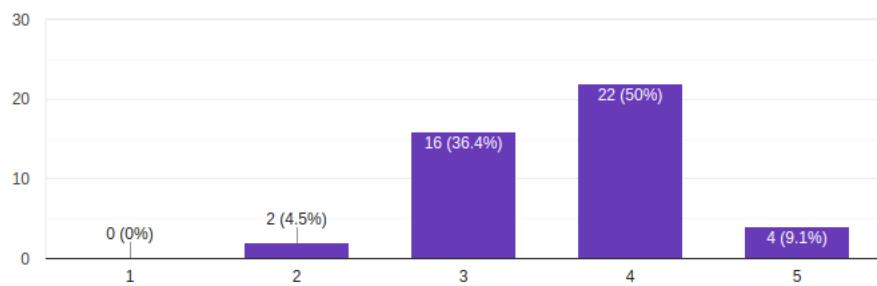


Gambar 5.19: Diagram jawaban pertanyaan *System Usability Scale* yang kedua

- 4 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu rumit" ?", mayoritas responden menjawab 2 (kurang setuju) dan 3 (netral).
- 5 Jumlah responden
- 6 yang menjawab 2 (kurang setuju) dan 3 (netral) masing-masing 17 orang. Diagram jawaban untuk
- 7 pertanyaan ini dapat dilihat pada Gambar 5.19.

**Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini mudah digunakan" ?**

44 responses

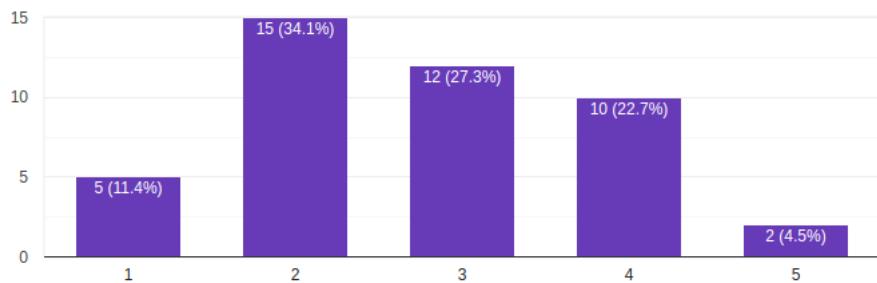


Gambar 5.20: Diagram jawaban pertanyaan *System Usability Scale* yang ketiga

- 1 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini mudah  
2 digunakan" ?", mayoritas responden menjawab 4 (setuju). Jumlah responden yang menjawab 4  
3 (setuju) adalah 22 orang. Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.20.

**Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya butuh bantuan teknisi untuk menggunakan fitur ini" ?**

44 responses

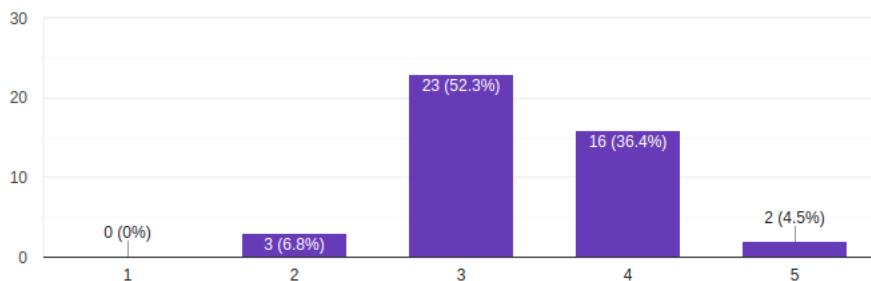


Gambar 5.21: Diagram jawaban pertanyaan *System Usability Scale* yang keempat

- 4 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya butuh bantuan  
5 teknisi untuk menggunakan fitur ini" ?", mayoritas responden menjawab 2 (kurang setuju). Jumlah  
6 responden yang menjawab 2 (kurang setuju) adalah 15 orang. Diagram jawaban untuk pertanyaan  
7 ini dapat dilihat pada Gambar 5.21.

Apakah Anda setuju dengan pernyataan ini : "Saya pikir berbagai fungsi yang ada di fitur ini terintegrasi dengan baik" ?

44 responses

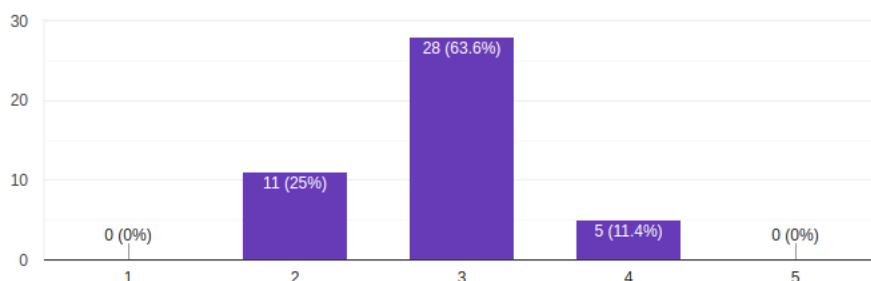


Gambar 5.22: Diagram jawaban pertanyaan *System Usability Scale* yang kelima

- 1 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir berbagai fungsi yang ada di fitur ini terintegrasi dengan baik" ?", mayoritas responden menjawab 3 (netral).
- 2 Jumlah responden yang menjawab 3 (netral) adalah 23 orang.
- 3 Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.22.

Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak hal yang TIDAK konsisten pada fitur ini" ?

44 responses

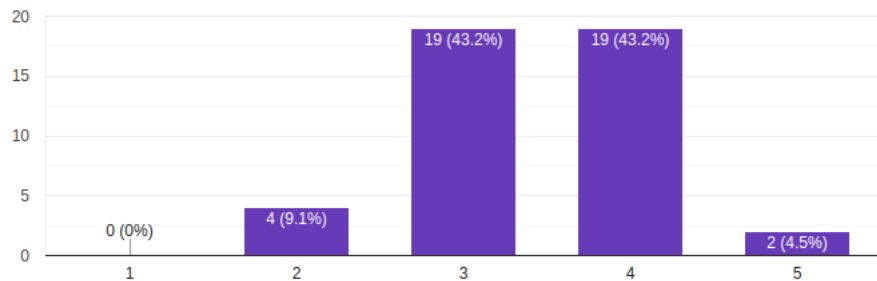


Gambar 5.23: Diagram jawaban pertanyaan *System Usability Scale* yang keenam

- 5 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak hal yang TIDAK konsisten pada fitur ini" ?", mayoritas responden menjawab 3 (netral).
- 6 Jumlah responden yang menjawab 3 (netral) adalah 28 orang.
- 7 Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.23.

Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak orang yang akan langsung mengerti cara memakai fitur ini" ?

44 responses

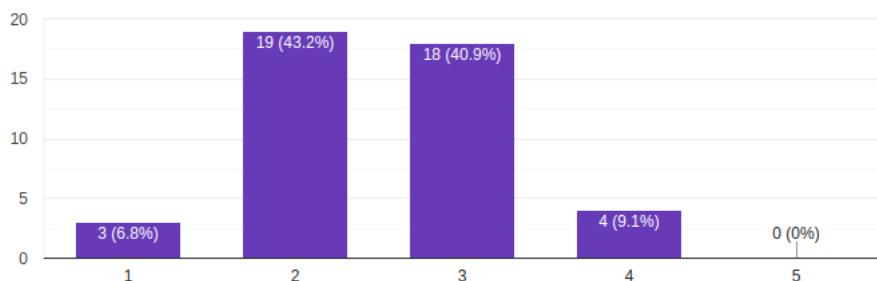


Gambar 5.24: Diagram jawaban pertanyaan *System Usability Scale* yang ketujuh

- 1 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak orang yang akan langsung mengerti cara memakai fitur ini" ?", mayoritas responden menjawab 3 (netral) dan 4 (setuju). Jumlah responden yang menjawab 3 (netral) dan 4 (setuju) masing-masing 19 orang.
- 2 Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.24.

Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu sulit untuk digunakan" ?

44 responses

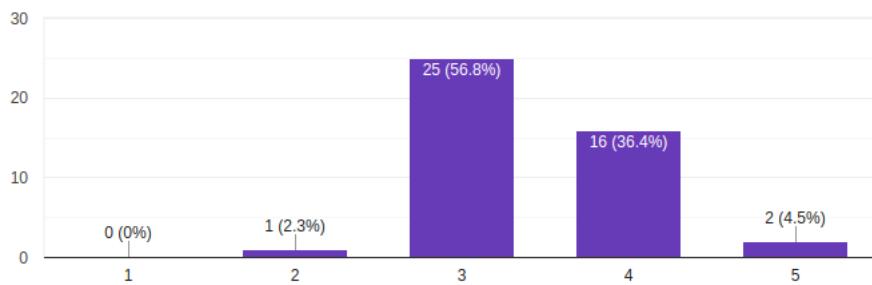


Gambar 5.25: Diagram jawaban pertanyaan *System Usability Scale* yang kedelapan

- 5 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu sulit untuk digunakan" ?", mayoritas responden menjawab 2 (kurang setuju). Jumlah responden yang menjawab 2 (kurang setuju) adalah 19 orang. Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.25.

**Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini sangat nyaman digunakan" ?**

44 responses

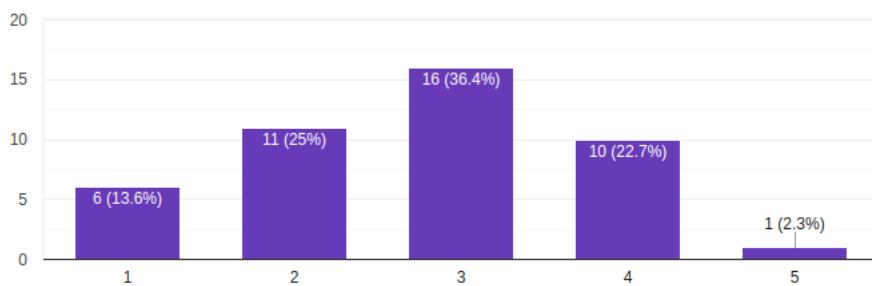


Gambar 5.26: Diagram jawaban pertanyaan *System Usability Scale* yang kesembilan

- 1 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini sangat nyaman digunakan" ?", mayoritas responden menjawab 3 (netral). Jumlah responden yang menjawab 3 (netral) adalah 25 orang. Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.26.
- 2
- 3

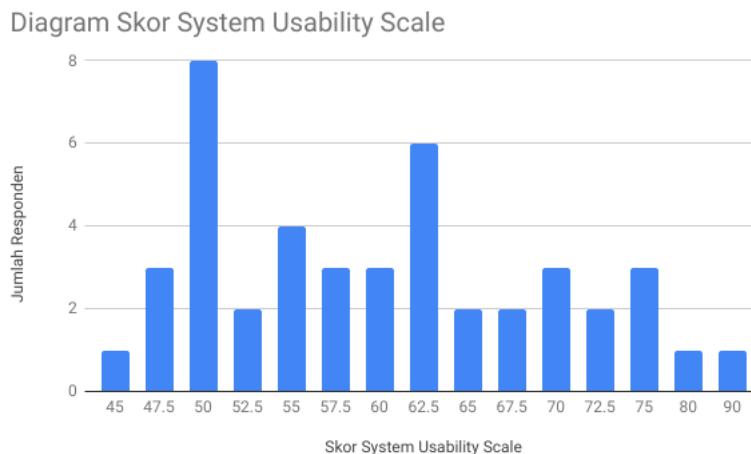
**Apakah Anda setuju dengan pernyataan ini : "Saya perlu belajar banyak hal terlebih dahulu sebelum dapat menggunakan fitur ini" ?**

44 responses



Gambar 5.27: Diagram jawaban pertanyaan *System Usability Scale* yang kesepuluh

- 4 Pada pertanyaan "Apakah Anda setuju dengan pernyataan ini : "Saya perlu belajar banyak hal terlebih dahulu sebelum dapat menggunakan fitur ini" ?", mayoritas responden menjawab 3 (netral).
- 5
- 6 Jumlah responden yang menjawab 3 (netral) adalah 16 orang. Diagram jawaban untuk pertanyaan ini dapat dilihat pada Gambar 5.27.
- 7



Gambar 5.28: Diagram Skor System Usability Scale

- 1 Setelah menghitung skor *System Usability Scale* seluruh responden yang menjawab pertanyaan  
 2 *System Usability Scale*, peneliti mendapatkan rata-ratanya adalah 60,34. Angka tersebut menunjukkan  
 3 bahwa usabilitas fitur kolektor pengumuman termasuk kelompok rata-rata. Diagram skor *System*  
 4 *Usability Scale* dapat dilihat pada Gambar 5.28.
- 5 Dari kolom saran, peneliti mendapatkan beberapa saran. Tabel 5.4 menampilkan saran yang  
 6 didapatkan peneliti.

Tabel 5.4: Tabel saran

Alamat email	Saran
keenan.leman@unpar.ac.id	Tidak bisa "add friend" karena akun telah mencapai batas banyak teman.
7316027@student.unpar.ac.id	Lebih diperbagus tampilannya sehingga menarik user untuk menggunakannya
7314028@student.unpar.ac.id	Good Job!
2017730022@student.unpar.ac.id	terdapat beberapa error dalam login, dimana email saya (@student.unpar) tidak memiliki akses
2017730047@student.unpar.ac.id	memperbaiki url login yang dikirimkan di shadowtape
2017730011@student.unpar.ac.id	saran saya, usahakan tidak menggunakan aplikasi luar lain seperti LINE. dikarenakan tidak semua orang menggunakan line
2017730037@student.unpar.ac.id	Link yang dibuka langsung dari line tidak memiliki hak akses
2017730008@student.unpar.ac.id	Fitur ini bagus, memudahkan mahasiswa dalam menerima notifikasi email. Namun, waktunya sebaiknya tidak hanya pada jam 12 siang karena waktu tersebut adalah saat orang-orang makan siang yang mungkin saja sedang tidak membuka gadget/aplikasi Line pada khususnya.
2017730044@student.unpar.ac.id	Log in masih tidak dapat dilakukan, terdapat pernyataan bahwa "alamat email" tidak memiliki hak akses ke pengumuman. Contohnya email yang saya gunakan "2017730044@student.unpar.ac.id" tidak memiliki hak akses ke pengumuman"
7316025@student.unpar.ac.id	"Lebih user friendly "

1 Setelah masa pengujian eksperimental selesai, peneliti menemukan bahwa layanan LINE@ akan  
2 diberhentikan. Saat pengujian eksperimental selesai dilaksanakan, pengguna LINE tidak bisa  
3 membuat akun LINE@ baru. Sebagai ganti LINE@, LINE membuat layanan baru tapi serupa  
4 bernama LINE Official Account. Akun LINE@ yang sudah ada akan diubah menjadi akun LINE  
5 Official Account pada bulan Juni 2019.

6 Perbedaan kedua layanan tersebut terletak pada metode penjualan mereka. LINE@ meminta  
7 bayaran apabila pemilik akun LINE@ ingin menambah banyak akun LINE yang dapat menambahkannya  
8 sebagai teman, tapi banyak pesan yang dapat dikirim tidak dibatasi. LINE Official Account meminta  
9 bayaran apabila pemilik akun LINE Official Account ingin menambah pesan yang dapat ia kirim,  
10 tapi jumlah akun yang dapat menambahkannya sebagai teman tidak dibatasi.

1

## BAB 6

2

### KESIMPULAN DAN SARAN

3

#### 6.1 Kesimpulan

4

Berikut adalah kesimpulan yang dapat diambil dari penelitian ini :

5

- BlueTape dapat dimodifikasi agar dapat berjalan di Heroku. BlueTape versi skripsi ini dapat diakses di <https://shadowtape.herokuapp.com/>.

7

- BlueTape telah memiliki fitur kolektor pengumuman. Fitur ini dapat melakukan sinkronisasi kotak masuk untuk alamat email [shadowbluetape@gmail.com](mailto:shadowbluetape@gmail.com) setiap jam. Email yang diidentifikasi sebagai pengumuman dapat ditampilkan di <https://shadowtape.herokuapp.com/pengumuman>. BlueTape dapat memunculkan notifikasi pada akun LINE@ Shadowtape.

11

#### 6.2 Saran

12

Berikut adalah saran untuk pengembangan lebih lanjut :

13

- Mencoba layanan baru LINE Official Account sebagai pengganti LINE@.

14

- Meneliti lebih lanjut masalah login yang ditemukan saat pengujian eksperimental.

15

- Meningkatkan usabilitas fitur ini agar calon pengguna nyaman menggunakannya.



## DAFTAR REFERENSI

- [1] Heroku (2018) Heroku dev center. <https://devcenter.heroku.com>. 15 November 2018.
- [2] Vixie, P. Cron manual page, 4th berkeley distribution. *The information from the crontab section (below and including the table) was taken (unedited, but with small additions) from the crontab manual pages. Type man, 1.*
- [3] Gmail (2018) Gmail api. <https://developers.google.com/gmail/api/>. 19 November 2018.
- [4] Group, P. (2008) Php:imap-manual. <http://php.net/manual/en/book imap.php>. 19 November 2018.
- [5] LINE (2018) Line developer. <https://developers.line.me>. 14 November 2018.
- [6] Albert, W. dan Tullis, T. (2013) *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics* Interactive Technologies. Elsevier Science.
- [7] Etzkorn, L. (2017) *Introduction to Middleware: Web Services, Object Components, and Cloud Computing*. CRC Press, Amerika Serikat.



## LAMPIRAN A

### KODE PROGRAM

Kode program untuk fitur kolektor pengumuman diketik di dalam *file* yang sudah ada dan *file* baru. Isi dari beberapa *file* yang sudah ada tidak ditampilkan seluruhnya pada lampiran ini, melainkan ditampilkan sebagian dalam format *diff*. *File* dalam format *diff* adalah hasil perbandingan antara isi *file* sebelum diubah dengan isi *file* pada perubahan terakhir memakai *git diff*. Isi *file* yang hampir seluruh barisnya berubah tidak ditampilkan dalam format *diff*, melainkan ditampilkan seluruhnya. Keterangan *listing* untuk *file* dengan format *diff* diakhiri dengan kata ".diff".

Berikut *file* yang kode programnya diubah atau ditambah:

Listing A.1: .gitignore.diff

```
1| diff --git a/.gitignore b/.gitignore
2| index 14bc68c..38c8d7d 100644
3| --- a/.gitignore
4| +++ b/.gitignore
5| @@ -1 +1,2 @@
6| ./nbproject/private/
7| \ No newline at end of file
8| ./nbproject/private/
9| ./vendor
```

Listing A.2: Procfile

```
1| web: vendor/bin/heroku-php-apache2 www/
```

Listing A.3: composer.json.diff

```
1| diff --git a/composer.json b/composer.json
2| index 8bcb1fd..b55200b 100644
3| --- a/composer.json
4| +++ b/composer.json
5| @@ -1,6 +1,8 @@
6| {
7|     "require": {
8|         "google/apiclient": "^1.0",
9|         "phpoffice/phpexcel": "^1.8"
10|     }
11|     "ext-imap": "*",
12|     "phpoffice/phpexcel": "^1.8",
13|     "linecorp/line-bot-sdk": "^3.6"
14| }
```

Listing A.4: /www/application/config/auth.php

```
1| <?php
2|
3| defined('BASEPATH') OR exit('No direct script access allowed');
4|
5| $config['domain'] = getenv('CI_BASE_URL');
6| $config['google-clientid'] = getenv('GOOGLE_CLIENTID');
7| $config['google-clientsecret'] = getenv('GOOGLE_CLIENTSECRET');
8| $config['google-redirecturi'] = $config['domain'] . '/auth/oauth2callback';
9|
10| $config['email-config'] = Array(
11|     'protocol' => 'smtp',
12|     'smtp_host' => getenv('SMTP_HOST'),
13|     'smtp_port' => intval(getenv('SMTP_PORT')),
14|     'smtp_user' => getenv('SMTP_USER'),
15|     'smtp_pass' => getenv('SMTP_PASS'),
16|     'mailtype' => 'html',
17|     'charset' => 'iso-8859-1'
18| );
```

Listing A.5: /www/application/config/autoload.php.diff

```

1 diff --git a/www/application/config/autoload.php b/www/application/config/autoload.php
2 index 66cb5fa..5d8982b 100644
3 --- a/www/application/config/autoload.php
4 +++ b/www/application/config/autoload.php
5 @@ -58,7 +58,7 @@ $autoload['packages'] = array();
6 |
7 | $autoload['libraries'] = array('user_agent' => 'ua');
8 */
9 -$autoload['libraries'] = array('session');
10 +$autoload['libraries'] = array('database', 'session');
11 /*
12 |
13 | -----

```

Listing A.6: /www/application/config/config.php.diff

```

1 diff --git a/www/application/config/config.php b/www/application/config/config.php
2 index 07d2eee..cd6385c 100644
3 --- a/www/application/config/config.php
4 +++ b/www/application/config/config.php
5 @@ -23,7 +23,7 @@ defined('BASEPATH') OR exit('No_direct_script_access_allowed');
6 | a PHP script and you can easily do that on your own.
7 |
8 */
9 -$config['base_url'] = $_ENV['CI_BASE_URL'];
10 +$config['base_url'] = getenv('CI_BASE_URL');
11 /*
12 |
13 | -----
14 @@ -213,7 +213,7 @@ $config['directory_trigger'] = 'd';
15 | your log files will fill up very fast.
16 |
17 */
18 -$config['log_threshold'] = 0;
19 +$config['log_threshold'] = array(1, 3);
20 /*
21 |
22 | -----
23 @@ -439,7 +439,11 @@ $config['global_xss_filtering'] = FALSE;
24 | 'csrf_regenerate' - Regenerate token on every submission
25 | 'csrf_exclude_uris' - Array of URIs which ignore CSRF checks
26 */
27 -$config['csrf_protection'] = TRUE;
28 +if (stripos($_SERVER['REQUEST_URI'], '/PengumumanLine/webhook') === FALSE) {
29 + $config['csrf_protection'] = TRUE;
30 +else{
31 + $config['csrf_protection'] = FALSE;
32 +}
33 $config['csrf_token_name'] = 'csrf_token';
34 $config['csrf_cookie_name'] = 'csrf_token';
35 $config['csrf_expire'] = 7200;
36 @@ -512,3 +516,5 @@ $config['rewrite_short_tags'] = FALSE;
37 | Array: array('10.0.1.200', '192.168.5.0/24')
38 */
39 $config['proxy_ips'] = '';
40 +
41 +$config['sess_save_path'] = sys_get_temp_dir();

```

Listing A.7: /www/application/config/database.php

```

1 <?php
2 defined('BASEPATH') OR exit('No_direct_script_access_allowed');
3 /*
4 |
5 | -----
6 | DATABASE CONNECTIVITY SETTINGS
7 |
8 | This file will contain the settings needed to access your database.
9 |
10 | For complete instructions please consult the 'Database Connection'
11 | page of the User Guide.
12 |
13 | -----
14 | EXPLANATION OF VARIABLES
15 |
16 |
17 | ['dsn'] The full DSN string describe a connection to the database.
18 | ['hostname'] The hostname of your database server.
19 | ['username'] The username used to connect to the database
20 | ['password'] The password used to connect to the database
21 | ['database'] The name of the database you want to connect to
22 | ['dbdriver'] The database driver. e.g.: mysqli.
23 |     Currently supported:
24 |         cubrid, ibase, mssql, mysql, mysqli, oci8,
25 |         odbc, pdo, postgre, sqlite, sqlite3, sqlsrv
26 | ['dbprefix'] You can add an optional prefix, which will be added
27 |             to the table name when using the Query Builder class
28 | ['pconnect'] TRUE/FALSE - Whether to use a persistent connection
29 | ['db_debug'] TRUE/FALSE - Whether database errors should be displayed.
30 | ['cache_on'] TRUE/FALSE - Enables/disables query caching
31 | ['cachedir'] The path to the folder where cache files should be stored
32 | ['char_set'] The character set used in communicating with the database
33 | ['dbcollat'] The character collation used in communicating with the database
34 |             NOTE: For MySQL and MySQLi databases, this setting is only used
35 |                 as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7

```

```

36 |     (and in table creation queries made with DB Forge).
37 |     There is an incompatibility in PHP with mysql_real_escape_string() which
38 |     can make your site vulnerable to SQL injection if you are using a
39 |     multi-byte character set and are running versions lower than these.
40 |     Sites using Latin-1 or UTF-8 database character set and collation are unaffected.
41 | ['swap_pre'] A default table prefix that should be swapped with the dbprefix
42 | ['encrypt'] Whether or not to use an encrypted connection.
43 |
44 |     'mysql' (deprecated), 'sqlsrv' and 'pdo/sqlsrv' drivers accept TRUE/FALSE
45 |     'mysqli' and 'pdo/mysql' drivers accept an array with the following options:
46 |
47 |         'ssl_key'      - Path to the private key file
48 |         'ssl_cert'     - Path to the public key certificate file
49 |         'ssl_ca'       - Path to the certificate authority file
50 |         'ssl_capath'   - Path to a directory containing trusted CA certificates in PEM format
51 |         'ssl_cipher'   - List of *allowed* ciphers to be used for the encryption, separated by colons (':')
52 |         'ssl_verify'   - TRUE/FALSE; Whether verify the server certificate or not ('mysqli' only)
53 |
54 | ['compress'] Whether or not to use client compression (MySQL only)
55 | ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
56 |                 - good for ensuring strict SQL while developing
57 | ['ssl_options'] Used to set various SSL options that can be used when making SSL connections.
58 | ['failover'] array - A array with 0 or more data for connections if the main should fail.
59 | ['save_queries'] TRUE/FALSE - Whether to "save" all executed queries.
60 |                 NOTE: Disabling this will also effectively disable both
61 | $this->db->last_query() and profiling of DB queries.
62 | When you run a query, with this setting set to TRUE (default),
63 | CodeIgniter will store the SQL statement for debugging purposes.
64 | However, this may cause high memory usage, especially if you run
65 | a lot of SQL queries ... disable this to avoid that problem.
66 |
67 | The $active_group variable lets you choose which connection group to
68 | make active. By default there is only one group (the 'default' group).
69 |
70 | The $query_builder variables lets you determine whether or not to load
71 | the query builder class.
72 */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn'      => '',
78     'hostname' => getenv('CI_DB_HOSTNAME'),
79     'username' => getenv('CI_DB_USERNAME'),
80     'password' => getenv('CI_DB_PASSWORD'),
81     'database' => getenv('CI_DB_DATABASE'),
82     'dbdriver' => 'postgre',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );

```

Listing A.8: /www/application/config/modules.php.diff

```

1| diff --git a/www/application/config/modules.php b/www/application/config/modules.php
2| index 5dfb8b1..db58232 100644
3| --- a/www/application/config/modules.php
4| +++ b/www/application/config/modules.php
5| @@ -8,8 +8,8 @@ $config['module-names'] = array(
6|     'PerubahanKuliahRequest' => 'Perubahan_Kuliah',
7|     'PerubahanKuliahManage' => 'Manajemen_Perubahan_Kuliah',
8|     'EntriJadwalDosen' => 'Entri_Jadwal_Dosen',
9|     'LihatJadwalDosen' => 'Lihat_Jadwal_Dosen'
10| -
11| +     'LihatJadwalDosen' => 'Lihat_Jadwal_Dosen',
12| +     'Pengumuman' => 'Pengumuman'
13| );
14|
15| $config['modules'] = array(
16| @@ -18,12 +18,13 @@ $config['modules'] = array(
17|     'PerubahanKuliahRequest' => array('root', 'staf.unpar'),
18|     'PerubahanKuliahManage' => array('root', 'tu.ftis'),
19|     'EntriJadwalDosen' => array('root', 'dosen.informatika' ),
20|     'LihatJadwalDosen' => array('root', 'mahasiswa.informatika', 'dosen.informatika'),
21|     'LihatJadwalDosen' => array('root', 'mahasiswa.informatika', 'dosen.informatika'),
22|     'Pengumuman' => array('root', 'mahasiswa.informatika', 'dosen.informatika')
23| );
24|
25| $config['roles'] = array(
26|     'root' => array('pascal@unpar.ac.id', 'shao.wei@unpar.ac.id'),
27|     'tu.ftis' => array('shao.wei@unpar.ac.id', 'pranyoto@unpar.ac.id', 'walip@unpar.ac.id', 'dwina@unpar.ac.id'),
28|     'root' => array('pascal@unpar.ac.id', 'shao.wei@unpar.ac.id', '7315029@student.unpar.ac.id'),
29|     'tu.ftis' => array('pascal@unpar.ac.id'), // array('shao.wei@unpar.ac.id', 'pranyoto@unpar.ac.id', 'walip@unpar.ac.id', 'dwina@unpar.ac.id'),
30|     'mahasiswa.ftis' => '[7[123]\\d{5}]|(20[1-9][0-9]7[123][0-9]{4})|(61[678][0-9]{7})@student\\.unpar\\.ac\\\\.id',
31|     'staf.unpar' => '.+@unpar\\.ac\\\\.id',
32|     'dosen.informatika' => array ('cheni@unpar.ac.id', 'mariskha@unpar.ac.id', 'anung@unpar.ac.id', 'moertini@unpar.ac.id', 'natalia@unpar.ac.id', 'chandraw@unpar.ac.id', 'elisatih@unpar.ac.id', 'gkarya@unpar.ac.id', 'husnulhakim@unpar.ac.id', '

```

```
joanna@unpar.ac.id', 'lionov@unpar.ac.id', 'luciana@unpar.ac.id', 'pascal@unpar.ac.id', 'rosad5@unpar.ac.id', 'vania.
natali@unpar.ac.id', 'kristopher.h@unpar.ac.id', 'raymond.chandra@unpar.ac.id', 'keenan.leman@unpar.ac.id'),
```

Listing A.9: /www/application/config/pengumuman.php

```
1 | <?php
2 |
3 | defined('BASEPATH') OR exit('No_direct_script_access_allowed');
4 |
5 | $config['pengirimTerverifikasi'] = array(
6 |     'shadowbluetape@gmail.com', 'cheni@unpar.ac.id', 'mariskha@unpar.ac.id', 'anung@unpar.ac.id', 'moertini@unpar.ac.id', '
7 |     natalia@unpar.ac.id', 'chandraw@unpar.ac.id', 'elisatih@unpar.ac.id', 'gkarya@unpar.ac.id', 'husnulhakim@unpar.ac.id', '
8 |     joanna@unpar.ac.id', 'lionov@unpar.ac.id', 'luciana@unpar.ac.id', 'claudio.franciscus@unpar.ac.id', 'pascal@unpar.ac.id'
9 |     , 'rosad5@unpar.ac.id', 'vania.natali@unpar.ac.id', 'kristopher.h@unpar.ac.id', 'raymond.chandra@unpar.ac.id', 'keenan.
10 |     leman@unpar.ac.id'
11 | );
12 | 
```

Listing A.10: /www/application/config/routes.php.diff

```
1 | diff --git a/www/application/config/routes.php b/www/application/config/routes.php
2 | index a19acda..8c388ea 100644
3 | --- a/www/application/config/routes.php
4 | +++ b/www/application/config/routes.php
5 | @@ -52,3 +52,5 @@ defined('BASEPATH') OR exit('No_direct_script_access_allowed');
6 |     $route['default_controller'] = 'auth';
7 |     $route['404_override'] = '';
8 |     $route['translate_uri_dashes'] = FALSE;
9 |
10 | +$route['pengumuman/page-(:num)'] = '/pengumuman/page/$1';
```

Listing A.11: /www/application/controllers/Auth.php.diff

```
1 | diff --git a/www/application/controllers/Auth.php b/www/application/controllers/Auth.php
2 | index da2c7b8..8c99c7d 100644
3 | --- a/www/application/controllers/Auth.php
4 | +++ b/www/application/controllers/Auth.php
5 | @@ -19,7 +19,12 @@ class Auth extends CI_Controller {
6 |     if ($code != NULL) {
7 |         $this->Auth_model->authenticateOAuthCode($code);
8 |         $userInfo = $this->Auth_model->getUserInfo();
9 |         header('Location:' . $userInfo['modules'][0]);
10 |         if ($this->session->has_userdata('redirect_url')){
11 |             redirect($this->session->userdata('redirect_url'));
12 |         }
13 |         else{
14 |             header('Location:' . $userInfo['modules'][0]);
15 |         }
16 |     } else {
17 |         throw new Exception("Mohon_login_terlebih_dahulu.");
18 |     }
```

Listing A.12: /www/application/controllers/Cron.php

```
1 | <?php
2 | defined('BASEPATH') OR exit('No_direct_script_access_allowed');
3 |
4 | class Cron extends CI_Controller {
5 |
6 |     public function daily() {
7 |         try {
8 |             $this->load->model('Pengumuman_model');
9 |             $newEmails = $this->Pengumuman_model->checkEmail();
10 |             $numberOfEmailsFound = 0;
11 |             $numberOfAnnouncementEmails = 0;
12 |             if ($newEmails != null){
13 |                 foreach ($newEmails as $newEmail){
14 |                     $numberOfEmailsFound = $numberOfEmailsFound + 1;
15 |                     $isPengumuman = $this->Pengumuman_model->proceedEmail($newEmail);
16 |                     if ($isPengumuman){
17 |                         $numberOfAnnouncementEmails = $numberOfAnnouncementEmails + 1;
18 |                     }
19 |                 }
20 |             }
21 |             log_message('info', "Successfully_performed_cron_jobs._The_number_of_new_emails_found_is_ " . $numberOfEmailsFound . "_
22 |             and_the_number_of_new_emails_that_are_announcements_is_ " . $numberOfAnnouncementEmails . ".");
23 |             http_response_code(200);
24 |             echo json_encode([
25 |                 'message' => "Successfully_performed_cron_jobs."
26 |             ]);
27 |         } catch (Exception $e) {
28 |             log_message('error', $e->getMessage());
29 |             http_response_code(500);
30 |             echo json_encode([
31 |                 'message' => $e->getMessage()
32 |             ]);
33 |         }
34 |     }
35 | }
```

Listing A.13: /www/application/controllers/Pengumuman.php

```
1 | <?php
```

```

2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Pengumuman extends CI_Controller {
5
6     public function __construct() {
7         parent::__construct();
8         try {
9             $this->load->helper('url');
10            $this->session->set_userdata('redirect_url', current_url());
11            $this->Auth_model->checkModuleAllowed(get_class());
12        } catch (Exception $ex) {
13            $this->session->set_flashdata('error', $ex->getMessage());
14            header('Location:' . '/');
15        }
16        $this->load->library('BlueTape');
17        $this->load->model('Pengumuman_model');
18        $this->load->database();
19    }
20
21    public function index() {
22        // Retrieve logged in user data
23        $userInfo = $this->Auth_model->getUserInfo();
24
25        $this->db->select();
26        $this->db->order_by('id', 'desc');
27        $query = $this->db->get('Pengumuman');
28        $announcements = $query->result_array();
29        foreach ($announcements as $announcement) {
30            $announcement['url'] = "/pengumuman/read/" . $announcement['id'];
31        }
32
33        $this->page(1);
34    }
35
36    public function read($id){
37        $this->db->where('id', $id);
38        $this->db->select('*');
39        $this->db->from('Pengumuman');
40        $query = $this->db->get();
41        $pengumuman= $query->row_array();
42        if ($pengumuman === NULL) {
43            show_404();
44            exit;
45        }
46        $this->load->view('Pengumuman/read', array(
47            'currentModule' => get_class(),
48            'pengumuman' => $pengumuman
49        ));
50    }
51
52    public function page($page){
53        $limit = 10;
54        $this->pagination($page,$limit,((($page-1)*$limit));
55    }
56
57    public function pagination($page,$limit,$i){
58        $jumlahPengumuman = $this->db->count_all('Pengumuman');
59        $this->db->select('*');
60        $this->db->order_by('waktuTerkirim', 'desc');
61        $this->db->from('Pengumuman');
62        $this->db->limit($limit,$i);
63        $query = $this->db->get();
64        $pengumumans = $query->result_array();
65        $currentPage = $page;
66        $pengumumanPerPage = $limit;
67        $this->load->view('Pengumuman/main', array(
68            'currentModule' => get_class(),
69            'jumlahPengumuman' => $jumlahPengumuman,
70            'pengumumans' => $pengumumans,
71            'currentPage' => $currentPage,
72            'pengumumanPerPage' => $pengumumanPerPage
73        ));
74    }
75}

```

Listing A.14: /www/application/controllers/PengumumanLine.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class PengumumanLine extends CI_Controller {
5
6     public function webhook(){
7         try{
8             if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
9                 http_response_code(405);
10                error_log('Method not allowed');
11                exit();
12            }
13
14            $httpPostRequestBody = file_get_contents('php://input');
15
16            if (strlen($httpPostRequestBody) === 0) {
17                http_response_code(400);
18                error_log('Missing request body');
19                exit();
20            }
21

```

```

22     $xLineSignature = $_SERVER['HTTP_X_LINE_SIGNATURE'];
23
24     if (empty($xLineSignature)) {
25         http_response_code(400); // Bad Request, Signature is Missing
26     }
27     else{
28         $this->load->model('Pengumuman_Line_model');
29         $this->Pengumuman_Line_model->proceedWebhook($httpPostRequestBody, $xLineSignature);
30     }
31     http_response_code(200);
32 }
33 catch(Exception $e){
34     http_response_code(500);
35 }
36 }
37 }
```

Listing A.15: /www/application/core/MY\_Log.php

```

1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2 // this class is adapted from system/libraries/Log.php
3 /**
4 * CodeIgniter
5 *
6 * An open source application development framework for PHP 5.1.6 or newer
7 *
8 * @package CodeIgniter
9 * @author EllisLab Dev Team
10 * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc.
11 * @copyright Copyright (c) 2014 - 2015, British Columbia Institute of Technology (http://bcit.ca/)
12 * @license http://codeigniter.com/user_guide/license.html
13 * @link http://codeigniter.com
14 * @since Version 1.0
15 * @filesource
16 */
17
18 /**
19 * -----
20 */
21 /**
22 * Logging Class
23 *
24 * @package CodeIgniter
25 * @subpackage Libraries
26 * @category Logging
27 * @author EllisLab Dev Team
28 * @link http://codeigniter.com/user_guide/general/errors.html
29 */
30 class MY_Log extends CI_Log {
31
32     protected $_threshold    = 1;
33     protected $_date_fmt    = 'Y-m-d H:i:s';
34     protected $_levels      = array('ERROR' => '1', 'DEBUG' => '2', 'INFO' => '3', 'ALL' => '4');
35
36     /**
37     * Constructor
38     */
39     public function __construct()
40     {
41         $config =& get_config();
42
43         if (is_numeric($config['log_threshold']))
44         {
45             $this->_threshold = $config['log_threshold'];
46         }
47
48         if ($config['log_date_format'] != '')
49         {
50             $this->_date_fmt = $config['log_date_format'];
51         }
52     }
53
54 /**
55 * Write Log to php://stderr
56 *
57 * Generally this function will be called using the global log_message() function
58 *
59 * @param string the error level
60 * @param string the error message
61 * @param bool whether the error is a native PHP error
62 * @return bool
63 */
64 public function write_log($level = 'error', $msg, $php_error = FALSE)
65 {
66     $level = strtoupper($level);
67
68     if ( ! isset($this->_levels[$level]) OR ($this->_levels[$level] > $this->_threshold))
69     {
70         return FALSE;
71     }
72
73     file_put_contents('php://stderr', $level.'.'.($level == 'INFO' ? '-' : '.').$this->_date_fmt.'-->'.$msg."\n");
74
75     return TRUE;
76 }
77 }
```

```

79 }
80 // END Log Class
81
82 /* End of file MY_Log.php */
83 /* Location: ./application/core/MY_Log.php */
84

```

Listing A.16: /www/application/migrations/20181011103200\_Pengumuman\_Initial.php

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Migration_Pengumuman_initial extends CI_Migration {
6
7     public function up() {
8         $fields = array(
9             'id' => array(
10                 'type' => 'int',
11                 'auto_increment' => TRUE
12             ),
13             'namaPengirim' => array(
14                 'type' => 'VARCHAR',
15                 'constraint' => '256'
16             ),
17             'emailPengirim' => array(
18                 'type' => 'VARCHAR',
19                 'constraint' => '256'
20             ),
21             'waktuTerkirim' => array(
22                 'type' => 'timestamp'
23             ),
24             'subjek' => array(
25                 'type' => 'VARCHAR',
26                 'constraint' => '256'
27             ),
28             'isi' => array(
29                 'type' => 'TEXT',
30                 'null' => TRUE
31             ),
32             'ketersediaanLampiran' => array(
33                 'type' => 'VARCHAR',
34                 'constraint' => '1'
35             )
36         );
37         $this->dbforge->add_field($fields);
38         $this->dbforge->add_key('id', TRUE);
39         $this->dbforge->create_table('Pengumuman');
40     }
41
42     public function down() {
43
44     }
45 }
46

```

Listing A.17: /www/application/migrations/20190210224400\_PengumumanLineFollowers\_initial.php

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Migration_PengumumanLineFollowers_initial extends CI_Migration {
6
7     public function up() {
8         $fields = array(
9             'userId' => array(
10                 'type' => 'VARCHAR',
11                 'constraint' => '256'
12             )
13         );
14         $this->dbforge->add_field($fields);
15         $this->dbforge->add_key('userId', TRUE);
16         $this->dbforge->create_table('PengumumanLineFollowers');
17     }
18
19     public function down() {
20
21     }
22 }
23

```

Listing A.18: /www/application/models/Pengumuman\_model.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Pengumuman_model extends CI_Model {
5     public function __construct() {
6         parent::__construct();
7
8         $this->load->config('auth');
9         $this->load->config('modules');
10    }
11

```

```

12| public function checkEmail(){
13|     $newEmails = null;
14|
15|     $hostname = getEnv('HOSTNAME_INCOMING_EMAIL');
16|     $username = getEnv('ANNOUNCEMENT_EMAIL');
17|     $password = getEnv('ANNOUNCEMENT_PASSWORD');
18|
19|     $inbox = imap_open($hostname,$username,$password) or die('Cannot connect to Gmail: ' . imap_last_error());
20|
21|     $emails = imap_search($inbox,'UNSEEN');
22|
23|     if($emails) {
24|         $i = 0;
25|         foreach($emails as $emailNumber) {
26|             $invalid = false;
27|             $header = imap_headerinfo($inbox,$emailNumber);
28|             $from = isset($header->from) ? $header->from : null;
29|             if($from != null){
30|                 $fromaddress = null;
31|                 foreach($from as $id => $object){
32|                     $fromaddress = isset($object->mailbox) && isset($object->host) ? $object->mailbox . "@" . $object->host :
33|                                         null;
34|                 }
35|                 $bodymsg = '';
36|                 $attachmentExist = 'N';
37|                 $structure = imap_fetchstructure($inbox, $emailNumber);
38|                 if(isset($structure->parts) && is_array($structure->parts)) {
39|                     if(isset($structure->parts[1])){
40|                         $parts1 = $structure->parts[1];
41|                         if(isset($parts1->disposition) && $parts1->disposition == "ATTACHMENT"){
42|                             $attachmentExist = 'Y';
43|                             $parts0 = $structure->parts[0];
44|                             if(isset($parts0->parts[1])){
45|                                 $bodymsg = imap_qprint(imap_fetchbody($inbox, $emailNumber, '1.2'));
46|                             }
47|                             else if(isset($parts0->parts[0])){
48|                                 $bodymsg = imap_qprint(imap_fetchbody($inbox, $emailNumber, '1.1'));
49|                             }
50|                             else{
51|                                 $bodymsg = imap_qprint(imap_fetchbody($inbox, $emailNumber, '1'));
52|                             }
53|                             else{
54|                                 $bodymsg = imap_qprint(imap_fetchbody($inbox, $emailNumber, '2'));
55|                             }
56|                         }
57|                         else{
58|                             $bodymsg = imap_qprint(imap_fetchbody($inbox, $emailNumber, '1'));
59|                         }
60|                     }
61|                     else{
62|                         $bodymsg = imap_qprint(imap_fetchbody($inbox, $emailNumber, '1'));
63|                     }
64|
65|                     if($fromaddress != null){
66|                         $newEmails[$i]['emailFrom'] = $fromaddress;
67|                         $newEmails[$i]['from'] = isset($header->fromaddress) ? $header->fromaddress : $fromaddress;
68|                         if(isset($header->update)){
69|                             $newEmails[$i]['date'] = date("Y-m-d H:i:s", $header->update);
70|                         }
71|                         else{
72|                             $invalid = true;
73|                         }
74|
75|                         if(isset($header->subject)){
76|                             $newEmails[$i]['subject'] = $header->subject;
77|                         }
78|                         else{
79|                             $invalid = true;
80|                         }
81|
82|                         $newEmails[$i]['body'] = $bodymsg;
83|                         $newEmails[$i]['attachmentExist'] = $attachmentExist;
84|                     }
85|                     if($invalid){
86|                         unset($newEmails[$i]);
87|                     }
88|                     else{
89|                         $i++;
90|                     }
91|                 }
92|             }
93|         }
94|
95|         $errors = imap_errors();
96|
97|         imap_close($inbox);
98|
99|         return $newEmails;
100}
101
102 public function proceedEmail($newEmail){
103     $isPengumuman = false;
104     $this->config->load('pengumuman');
105     $terverifikasi = 0;
106     $daftarEmailTerverifikasi = $this->config->item('pengirimTerverifikasi');
107     foreach($daftarEmailTerverifikasi as $emailTerverifikasi){
108         if($newEmail['emailFrom'] == $emailTerverifikasi){
109             $terverifikasi = 1;

```

```
110 }
111 }
112 }
113 if($terverifikasi == 1){
114     $isPengumuman = true;
115     $this->db->insert('Pengumuman', array(
116         'namaPengirim' => $newEmail['from'],
117         'emailPengirim' => $newEmail['emailFrom'],
118         'waktuTerkirim' => $newEmail['date'],
119         'subjek' => $newEmail['subject'],
120         'isi' => $newEmail['body'],
121         'ketersediaanLampiran' => $newEmail['attachmentExist']
122     ));
123     $justInserted = $this->db->select("*")->order_by('id', 'desc')->limit(1)->get('Pengumuman')->row();
124     $id = $justInserted->id;
125
126     $this->load->model('Pengumuman_Line_model');
127     $message = "Ada pengumuman baru dari " . $newEmail['from'] . ". " . $newEmail['subject'] . ". Silahkan klik link ini untuk melihatnya: " . base_url() . "pengumuman/read/" . $id;
128     $this->Pengumuman_Line_model->pushMessageToAllFollowers($message);
129 }
130 return $isPengumuman;
131 }
132 }
```

Listing A.19: /www/application/models/Pengumuman\_Line\_model.php

```

72         $this->db->insert('PengumumanLineFollowers', array(
73             'userId' => $event->getUserId()
74         ));
75         $this->bot->replyText($event->getReplyToken(), 'Terima_kasih_telah_mengikuti_akun_ini._Pengumuman_baru_di_
76         Bluetape_akan_diberitahukan_melalui_akun_ini.');
77     } elseif ($event instanceof JoinEvent) {
78         // Not handled
79     } elseif ($event instanceof LeaveEvent) {
80         // Not handled
81     } elseif ($event instanceof PostbackEvent) {
82         // Not handled
83     } elseif ($event instanceof BeaconDetectionEvent) {
84         // Not handled
85     } elseif ($event instanceof AccountLinkEvent) {
86         // Not handled
87     } elseif ($event instanceof UnknownEvent) {
88         http_response_code(400); // Invalid event type
89     } else {
90         http_response_code(400); // Invalid event type
91     }
92 } catch (InvalidSignatureException $e) {
93     http_response_code(400); // Invalid signature
94 } catch (InvalidEventRequestException $e) {
95     http_response_code(400); // Invalid event request
96 }
97 }
98
99 public function pushMessageToAllFollowers($text){
100     $tos = [];
101     $query = $this->db->get('PengumumanLineFollowers');
102     foreach ($query->result() as $row){
103         $tos[] = $row->userId;
104     }
105
106     $ref = new ReflectionClass('LINE\LINEBot\MessageBuilder\TextMessageBuilder');
107     $textMessageBuilder = $ref->newInstanceArgs(array_merge([$text]));
108     $this->bot->multicast($tos, $textMessageBuilder);
109 }
110 }
111 }
```

Listing A.20: /www/application/views/Pengumuman/main.php

```

1 <?php
2 defined('BASEPATH') OR exit('No_direct_script_access_allowed');
3 ?><!doctype html>
4 <html class="no-js" lang="en">
5     <?php $this->load->view('templates/head_loggedin'); ?>
6     <body>
7         <?php $this->load->view('templates/topbar_loggedin'); ?>
8         <?php $this->load->view('templates/flashmessage'); ?>
9         <?php $this->load->view('templates/script_foundation'); ?>
10        <div class="row">
11            <div class="medium-12_column">
12                <?php
13                    if($jumlahPengumuman == 0){
14                        echo "Tidak Ada Pengumuman";
15                    } else{
16                        foreach($pengumumans as $pengumuman):
17
18                            >>>
19                                <div class="callout">
20                                    <h3><a href=<?="/pengumuman/read/" . $pengumuman['id']?> <?=$pengumuman['subjek']?></a></h3>
21                                    <p>
22                                        by : <?=$pengumuman['namaPengirim']?> (<?=$pengumuman['emailPengirim']?>) on <?=$date("D, d M Y, H:i:s", strtotime($pengumuman['waktuTerkirim']))?>
23                                    </p>
24                                </div>
25
26                            <?php
27                                endforeach;
28                                if($currentPage>1){
29                                    $previousPage = $currentPage-1;
30                                    echo "<a href='/pengumuman/page-". $previousPage . "'>Previous</a>";
31                                }
32                                if($jumlahPengumuman > $currentPage*$pengumumanPerPage){
33                                    $nextPage = $currentPage+1;
34                                    echo "<a href='/pengumuman/page-". $nextPage . "'>Next</a>";
35                                }
36
37                            </div>
38            </div>
39        </body>
40    </html>
```

Listing A.21: /www/application/views/Pengumuman/read.php

```

1 <?php
2 defined('BASEPATH') OR exit('No_direct_script_access_allowed');
3 ?><!doctype html>
4 <html class="no-js" lang="en">
5     <?php $this->load->view('templates/head_loggedin'); ?>
6     <body>
7         <?php $this->load->view('templates/topbar_loggedin'); ?>
8         <?php $this->load->view('templates/flashmessage'); ?>
```

```
9  <?php $this->load->view('templates/script_foundation'); ?>
10 <div class="row">
11   <div class="medium-12_column">
12     <div class="callout">
13       <h3><?= $pengumuman['subjek']?></h3>
14       <p>
15         by : <?= $pengumuman['namaPengirim']?> (<?= $pengumuman['emailPengirim']?>) on <?=date("D,d_M_Y:H:i:s",
16           strtotime($pengumuman['waktuTerkirim']))?>
17         </p>
18         <br><br>
19         <?php
20           if(!strpos($pengumuman['isi'], "<div")){
21             <?= nl2br(nl2br($pengumuman['isi'])); ?>
22           }
23         <?php
24           else{
25             <?= $pengumuman['isi']; ?>
26           <?php
27             }
28           </p>
29           <br><br>
30           <?php if($pengumuman['ketersediaanLampiran'] == 'Y')?>
31             <p>
32               *) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung email student Anda untuk
33               mengunduhnya.
34             </p>
35           <?php
36             endif;
37           ?>
38         </div>
39       </div>
40     </div>
41   </body>
42 </html>
```



## LAMPIRAN B

### LAMPIRAN PENGUJIAN EKSPERIMENTAL

#### B.1 Kuesioner

Kuesioner diawali dengan pertanyaan :

#### Survey Skripsi Kolektor Pengumuman Informatika

Survei ini dibuat untuk melengkapi skripsi dengan topik "Kolektor Pengumuman Informatika". Topik ini bertujuan untuk membuat fitur yang dapat memberikan notifikasi melalui bot LINE apabila ada pengumuman baru di ruang lingkup jurusan Teknik Informatika. Topik ini dibangun sebagai fitur tambahan pada BlueTape (pada skripsi ini diuji dulu di shadowtape).

Your email address ([7315029@student.unpar.ac.id](mailto:7315029@student.unpar.ac.id)) will be recorded when you submit this form.  
Not you? [Switch account](#)

\* Required

Apakah Anda dosen atau mahasiswa? \*

- Dosen  
 Mahasiswa

**NEXT**

Never submit passwords through Google Forms.

Gambar B.1: Kuesioner bagian pertama

Apabila responden memilih "Dosen", maka responden akan dialihkan ke bagian Dosen (Lampiran B.1.1). Apabila responden memilih "Mahasiswa", maka responden akan dialihkan ke bagian Mahasiswa (Lampiran B.1.2). Responden wajib mengisi semua pertanyaan di kuesioner ini kecuali pertanyaan saran.

### B.1.1 Dosen

Apakah Anda dapat melihat pengumuman yang Anda kirimkan di menu pengumuman shadowtape (<https://shadowtape.herokuapp.com/pengumuman>) pada jam 12 siang setelah Anda mengirimkan pengumuman? \*

- Ya
- Tidak

**BACK**      **NEXT**

Never submit passwords through Google Forms.

Gambar B.2: Kuesioner bagian Dosen pertanyaan pertama

Apabila responden memilih jawaban "Ya" pada pertanyaan pertama (Gambar B.2), maka responden akan dialihkan ke pertanyaan kedua (Gambar B.3). Apabila responden memilih jawaban "Tidak", maka responden akan dialihkan ke bagian *System Usability Scale* dan Saran (Lampiran B.1.3) dan Saran (Gambar B.1.3).

Apakah judul pengumuman sesuai dengan subjek email yang Anda kirimkan? \*

- Ya
- Tidak

Apakah isi email sesuai dengan yang Anda kirimkan? \*

- Ya
- Tidak

**BACK**      **NEXT**

Never submit passwords through Google Forms.

Gambar B.3: Kuesioner bagian Dosen pertanyaan kedua dan ketiga

Apabila responden memilih jawaban "Ya" pada pertanyaan ketiga (pertanyaan kedua di Gambar B.2), maka responden akan dialihkan ke pertanyaan keempat (Gambar B.4). Apabila responden memilih jawaban "Tidak", maka responden akan dialihkan ke bagian *System Usability Scale* dan Saran (Gambar B.1.3).

Bagian manakah yang tidak sesuai dengan isi email yang Anda kirimkan? \*

- Tidak ada in line attachment (Contoh : gambar yang disisipkan di antara paragraf)
- Tidak ada attachment, namun terdapat keterangan "\*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung email student Anda untuk mengunduhnya." di bawah isi email.
- Tidak ada attachment dan tidak ada keterangan "\*) Pengumuman ini memiliki lampiran, silahkan memeriksa langsung email student Anda untuk mengunduhnya." di bawah isi email.
- Formatnya berbeda
- Isi email (dalam bentuk teks) tidak lengkap.
- Other: \_\_\_\_\_

[BACK](#)[NEXT](#)

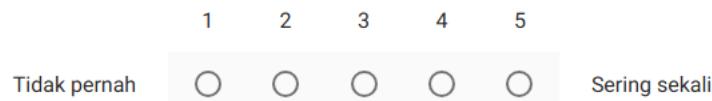
Never submit passwords through Google Forms.

Gambar B.4: Kuesioner bagian Dosen pertanyaan keempat

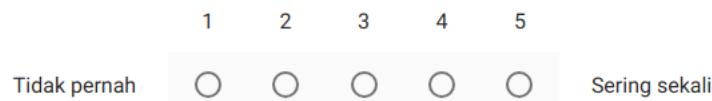
Apapun jawaban responden pada pertanyaan keempat (Gambar B.4), responden akan dialihkan ke bagian *System Usability Scale* dan Saran (Gambar B.1.3).

### B.1.2 Mahasiswa

Seberapa sering Anda membuka email student Anda? \*



Seberapa sering Anda membuka aplikasi LINE? \*

[BACK](#)[NEXT](#)

Never submit passwords through Google Forms.

Gambar B.5: Kuesioner bagian Mahasiswa pertanyaan pertama dan kedua

Apapun jawaban responden pada pertanyaan pertama dan kedua(Gambar B.5), responden akan dialihkan ke pertanyaan kedua (Gambar B.6).

Apakah Anda pernah menerima notifikasi pemberitahuan dari akun bot shadowtape pada jam 12 siang setelah mengikuti akun tersebut? \*

- Ya
- Tidak

[BACK](#) [NEXT](#)

Never submit passwords through Google Forms.

Gambar B.6: Kuesioner bagian Mahasiswa pertanyaan ketiga

Apabila responden memilih jawaban "Ya" pada pertanyaan ketiga (Gambar B.6), maka responden akan dialihkan ke pertanyaan keempat (Gambar B.7). Apabila responden memilih jawaban "Tidak", maka jawaban responden akan dikirim kepada peneliti.

Apakah Anda dapat membuka url yang dicantumkan di notifikasi pengumuman?

- Ya
- Tidak

[BACK](#) [NEXT](#)

Never submit passwords through Google Forms.

Gambar B.7: Kuesioner bagian Mahasiswa pertanyaan keempat

Apabila responden memilih jawaban "Ya" pada pertanyaan keempat (Gambar B.7), maka responden akan dialihkan ke pertanyaan kelima (Gambar B.8). Apabila responden memilih jawaban "Tidak", maka responden akan dialihkan ke bagian *System Usability Scale* dan Saran (Gambar B.1.3).

Jika Anda belum login saat membuka url yang dicantumkan di notifikasi pengumuman, apakah Anda diarahkan kembali ke url tersebut? \*

- Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
- Tidak, saya diarahkan ke halaman utama atau halaman lain.

[BACK](#) [NEXT](#)

Never submit passwords through Google Forms.

Gambar B.8: Kuesioner bagian Mahasiswa pertanyaan kelima

Apapun jawaban responden pada pertanyaan kelima (Gambar B.8), responden akan dialihkan

ke bagian *System Usability Scale* dan Saran (Gambar B.1.3).

### B.1.3 *System Usability Scale* dan Saran

Bagian ini adalah bagian terakhir dari kuesioner. Bagian ini memiliki 11 pertanyaan yang terdiri dari 10 pertanyaan hasil adaptasi pertanyaan yang ditanyakan saat uji usabilitas dengan metode *System Usability Scale* dan 1 pertanyaan saran. Setelah responden menjawab bagian ini, jawaban responden akan dikirim kepada peneliti. Pertanyaan pada bagian ini ditampilkan pada Gambar B.9, Gambar B.10, dan Gambar B.11.

**Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya ingin menggunakan fitur ini" ? \***

1	2	3	4	5		
Sangat tidak setuju	<input type="radio"/>	Sangat setuju				

**Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu rumit" ? \***

1	2	3	4	5		
Sangat tidak setuju	<input type="radio"/>	Sangat setuju				

Gambar B.9: Kuesioner bagian *System Usability Scale* dan Saran bagian 1

Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini mudah digunakan" ? \*

1      2      3      4      5

Sangat tidak setuju                               Sangat setuju

Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya butuh bantuan teknisi untuk menggunakan fitur ini" ? \*

1      2      3      4      5

Sangat tidak setuju                               Sangat setuju

Apakah Anda setuju dengan pernyataan ini : "Saya pikir berbagai fungsi yang ada di fitur ini terintegrasi dengan baik" ? \*

1      2      3      4      5

Sangat tidak setuju                               Sangat setuju

Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak hal yang TIDAK konsisten pada fitur ini" ? \*

1      2      3      4      5

Sangat tidak setuju                               Sangat setuju

Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak orang yang akan langsung mengerti cara memakai fitur ini" ? \*

1      2      3      4      5

Sangat tidak setuju                               Sangat setuju

Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu sulit untuk digunakan" ? \*

1      2      3      4      5

Sangat tidak setuju

Sangat setuju

Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini sangat nyaman digunakan" ? \*

1      2      3      4      5

Sangat tidak setuju

Sangat setuju

Apakah Anda setuju dengan pernyataan ini : "Saya perlu belajar banyak hal terlebih dahulu sebelum dapat menggunakan fitur ini" ? \*

1      2      3      4      5

Sangat tidak setuju

Sangat setuju

Saran Anda untuk fitur ini

Your answer

BACK

NEXT

Never submit passwords through Google Forms.

Gambar B.11: Kuesioner bagian *System Usability Scale* dan Saran bagian 3

## B.2 Hasil Mentah Kuesioner

Waktu Pengisian Survei	Alamat Email	Apakah Anda dosen atau mahasiswa?
4/26/2019 15:28:55	2017730023@student.unpar.ac.id	Mahasiswa
4/26/2019 15:49:26	keenan.leman@unpar.ac.id	Dosen
4/26/2019 15:54:15	7316071@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:22	7314035@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:28	2017730017@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:29	2017730026@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:31	2017730002@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:40	7316003@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:42	2017730030@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:50	2017730021@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:53	2017730046@student.unpar.ac.id	Mahasiswa
4/26/2019 15:54:54	7316021@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:02	2017730024@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:03	7316016@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:11	7313010@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:14	2017730050@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:23	7316027@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:28	2017730060@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:29	2017730073@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:30	2017730045@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:31	7316088@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:32	2017730052@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:42	7316049@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:43	7316028@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:45	2017730027@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:46	2017730056@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:46	7316032@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:49	7316089@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:50	2017730006@student.unpar.ac.id	Mahasiswa
4/26/2019 15:55:59	7314075@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:05	2017730003@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:12	7314028@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:15	2017730068@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:15	7316006@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:16	2017730009@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:19	2017730018@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:22	2017730035@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:24	2017730019@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:27	7316042@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:39	2017730016@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:39	2017730034@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:44	2017730042@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:44	2017730005@student.unpar.ac.id	Mahasiswa
4/26/2019 15:56:51	2017730015@student.unpar.ac.id	Mahasiswa

4/26/2019 15:57:04	7315064@student.unpar.ac.id	Mahasiswa
4/26/2019 15:57:12	2017730082@student.unpar.ac.id	Mahasiswa
4/26/2019 15:57:17	2017730088@student.unpar.ac.id	Mahasiswa
4/26/2019 15:57:26	2017730038@student.unpar.ac.id	Mahasiswa
4/26/2019 15:57:29	2017730077@student.unpar.ac.id	Mahasiswa
4/26/2019 15:57:29	2017730063@student.unpar.ac.id	Mahasiswa
4/26/2019 15:57:49	7316018@student.unpar.ac.id	Mahasiswa
4/26/2019 15:57:55	2017730022@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:02	2017730028@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:04	7316073@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:06	2017730047@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:07	2017730067@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:16	7316045@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:22	7316024@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:30	7316008@student.unpar.ac.id	Mahasiswa
4/26/2019 15:58:37	2017730070@student.unpar.ac.id	Mahasiswa
4/26/2019 15:59:03	2017730011@student.unpar.ac.id	Mahasiswa
4/26/2019 15:59:22	7314053@student.unpar.ac.id	Mahasiswa
4/26/2019 16:00:21	2017730037@student.unpar.ac.id	Mahasiswa
4/26/2019 16:01:05	2017730008@student.unpar.ac.id	Mahasiswa
4/26/2019 16:01:25	2017730054@student.unpar.ac.id	Mahasiswa
4/26/2019 16:02:07	2017730044@student.unpar.ac.id	Mahasiswa
4/26/2019 16:02:13	7316025@student.unpar.ac.id	Mahasiswa
4/26/2019 16:02:20	2017730069@student.unpar.ac.id	Mahasiswa
4/26/2019 16:02:48	2017730010@student.unpar.ac.id	Mahasiswa
4/26/2019 16:52:21	2017730033@student.unpar.ac.id	Mahasiswa
4/26/2019 17:10:31	2017730043@student.unpar.ac.id	Mahasiswa
4/29/2019 10:43:09	pascal@unpar.ac.id	Dosen

<b>Alamat email</b>	<b>Apakah Anda dapat melihat pengumuman yang Anda kirimkan di menu pengumuman shadowtape (<a href="https://shadowtape.herokuapp.com/pengumuman">https://shadowtape.herokuapp.com/pengumuman</a>) pada jam 12 siang setelah Anda mengirimkan pengumuman?</b>
keenan.leman@unpar.ac.id	Tidak
pascal@unpar.ac.id	Ya

<b>Alamat email</b>	<b>Apakah judul pengumuman sesuai dengan subjek email yang Anda kirimkan?</b>
pascal@unpar.ac.id	Ya

<b>Alamat email</b>	<b>Apakah isi email sesuai dengan yang Anda kirimkan?</b>
pascal@unpar.ac.id	Ya

<b>Alamat email</b>	<b>Bagian manakah yang tidak sesuai dengan isi email yang Anda kirimkan?</b>
	<i>Tidak ada respon</i>

Alamat email	Seberapa sering Anda membuka email student Anda?	Seberapa sering Anda membuka aplikasi LINE?
2017730023@student.unpar.ac.id	3	5
7316071@student.unpar.ac.id	4	5
7314035@student.unpar.ac.id	4	4
2017730017@student.unpar.ac.id	3	5
2017730026@student.unpar.ac.id	3	5
2017730002@student.unpar.ac.id	3	5
7316003@student.unpar.ac.id	5	5
2017730030@student.unpar.ac.id	4	5
2017730021@student.unpar.ac.id	5	5
2017730046@student.unpar.ac.id	2	4
7316021@student.unpar.ac.id	2	4
2017730024@student.unpar.ac.id	4	5
7316016@student.unpar.ac.id	5	5
7313010@student.unpar.ac.id	5	3
2017730050@student.unpar.ac.id	5	5
7316027@student.unpar.ac.id	5	5
2017730060@student.unpar.ac.id	5	5
2017730073@student.unpar.ac.id	5	5
2017730045@student.unpar.ac.id	5	5
7316088@student.unpar.ac.id	3	5
2017730052@student.unpar.ac.id	5	5
7316049@student.unpar.ac.id	4	5
7316028@student.unpar.ac.id	3	4
2017730027@student.unpar.ac.id	4	5
2017730056@student.unpar.ac.id	5	5
7316032@student.unpar.ac.id	2	5
7316089@student.unpar.ac.id	3	4
2017730006@student.unpar.ac.id	4	5
7314075@student.unpar.ac.id	3	2
2017730003@student.unpar.ac.id	3	5
7314028@student.unpar.ac.id	4	4
2017730068@student.unpar.ac.id	4	4
7316006@student.unpar.ac.id	3	4
2017730009@student.unpar.ac.id	5	5
2017730018@student.unpar.ac.id	3	4
2017730035@student.unpar.ac.id	4	4
2017730019@student.unpar.ac.id	3	4
7316042@student.unpar.ac.id	5	5
2017730016@student.unpar.ac.id	2	5
2017730034@student.unpar.ac.id	4	5
2017730042@student.unpar.ac.id	4	5
2017730005@student.unpar.ac.id	3	4
2017730015@student.unpar.ac.id	4	5

7315064@student.unpar.ac.id	5	5
2017730082@student.unpar.ac.id	3	5
2017730088@student.unpar.ac.id	4	5
2017730038@student.unpar.ac.id	5	5
2017730077@student.unpar.ac.id	2	5
2017730063@student.unpar.ac.id	3	5
7316018@student.unpar.ac.id	4	5
2017730022@student.unpar.ac.id	5	5
2017730028@student.unpar.ac.id	5	5
7316073@student.unpar.ac.id	5	5
2017730047@student.unpar.ac.id	5	5
2017730067@student.unpar.ac.id	4	5
7316045@student.unpar.ac.id	5	5
7316024@student.unpar.ac.id	3	5
7316008@student.unpar.ac.id	5	5
2017730070@student.unpar.ac.id	4	5
2017730011@student.unpar.ac.id	5	5
7314053@student.unpar.ac.id	3	3
2017730037@student.unpar.ac.id	5	5
2017730008@student.unpar.ac.id	5	5
2017730054@student.unpar.ac.id	3	5
2017730044@student.unpar.ac.id	4	4
7316025@student.unpar.ac.id	4	5
2017730069@student.unpar.ac.id	4	5
2017730010@student.unpar.ac.id	4	5
2017730033@student.unpar.ac.id	4	4
2017730043@student.unpar.ac.id	4	5

Alamat email	Apakah Anda pernah menerima notifikasi pemberitahuan dari akun bot shadowtape pada jam 12 siang setelah mengikuti akun tersebut?
2017730023@student.unpar.ac.id	Ya
7316071@student.unpar.ac.id	Tidak
7314035@student.unpar.ac.id	Tidak
2017730017@student.unpar.ac.id	Tidak
2017730026@student.unpar.ac.id	Tidak
2017730002@student.unpar.ac.id	Tidak
7316003@student.unpar.ac.id	Ya
2017730030@student.unpar.ac.id	Tidak
2017730021@student.unpar.ac.id	Tidak
2017730046@student.unpar.ac.id	Tidak
7316021@student.unpar.ac.id	Tidak
2017730024@student.unpar.ac.id	Ya
7316016@student.unpar.ac.id	Tidak
7313010@student.unpar.ac.id	Tidak
2017730050@student.unpar.ac.id	Tidak
7316027@student.unpar.ac.id	Ya
2017730060@student.unpar.ac.id	Ya
2017730073@student.unpar.ac.id	Tidak
2017730045@student.unpar.ac.id	Tidak
7316088@student.unpar.ac.id	Tidak
2017730052@student.unpar.ac.id	Ya
7316049@student.unpar.ac.id	Tidak
7316028@student.unpar.ac.id	Ya
2017730027@student.unpar.ac.id	Ya
2017730056@student.unpar.ac.id	Tidak
7316032@student.unpar.ac.id	Ya
7316089@student.unpar.ac.id	Ya
2017730006@student.unpar.ac.id	Ya
7314075@student.unpar.ac.id	Ya
2017730003@student.unpar.ac.id	Ya
7314028@student.unpar.ac.id	Ya
2017730068@student.unpar.ac.id	Ya
7316006@student.unpar.ac.id	Ya
2017730009@student.unpar.ac.id	Ya
2017730018@student.unpar.ac.id	Tidak
2017730035@student.unpar.ac.id	Ya
2017730019@student.unpar.ac.id	Tidak
7316042@student.unpar.ac.id	Tidak
2017730016@student.unpar.ac.id	Tidak
2017730034@student.unpar.ac.id	Tidak
2017730042@student.unpar.ac.id	Tidak
2017730005@student.unpar.ac.id	Tidak
2017730015@student.unpar.ac.id	Ya

7315064@student.unpar.ac.id	Ya
2017730082@student.unpar.ac.id	Tidak
2017730088@student.unpar.ac.id	Ya
2017730038@student.unpar.ac.id	Ya
2017730077@student.unpar.ac.id	Ya
2017730063@student.unpar.ac.id	Tidak
7316018@student.unpar.ac.id	Ya
2017730022@student.unpar.ac.id	Ya
2017730028@student.unpar.ac.id	Ya
7316073@student.unpar.ac.id	Tidak
2017730047@student.unpar.ac.id	Ya
2017730067@student.unpar.ac.id	Ya
7316045@student.unpar.ac.id	Ya
7316024@student.unpar.ac.id	Ya
7316008@student.unpar.ac.id	Ya
2017730070@student.unpar.ac.id	Ya
2017730011@student.unpar.ac.id	Ya
7314053@student.unpar.ac.id	Ya
2017730037@student.unpar.ac.id	Ya
2017730008@student.unpar.ac.id	Ya
2017730054@student.unpar.ac.id	Ya
2017730044@student.unpar.ac.id	Ya
7316025@student.unpar.ac.id	Ya
2017730069@student.unpar.ac.id	Ya
2017730010@student.unpar.ac.id	Ya
2017730033@student.unpar.ac.id	Tidak
2017730043@student.unpar.ac.id	Ya

Alamat email	Apakah Anda dapat membuka url yang dicantumkan di notifikasi pengumuman?
2017730023@student.unpar.ac.id	Tidak
7316003@student.unpar.ac.id	Ya
2017730024@student.unpar.ac.id	Tidak
7316027@student.unpar.ac.id	Ya
2017730060@student.unpar.ac.id	Tidak
2017730052@student.unpar.ac.id	Ya
7316028@student.unpar.ac.id	Ya
2017730027@student.unpar.ac.id	Ya
7316032@student.unpar.ac.id	Ya
7316089@student.unpar.ac.id	Ya
2017730006@student.unpar.ac.id	Ya
7314075@student.unpar.ac.id	Ya
2017730003@student.unpar.ac.id	Tidak
7314028@student.unpar.ac.id	Ya
2017730068@student.unpar.ac.id	Ya
7316006@student.unpar.ac.id	Ya
2017730009@student.unpar.ac.id	Ya
2017730035@student.unpar.ac.id	Tidak
2017730015@student.unpar.ac.id	Tidak
7315064@student.unpar.ac.id	Ya
2017730088@student.unpar.ac.id	Ya
2017730038@student.unpar.ac.id	Ya
2017730077@student.unpar.ac.id	Ya
7316018@student.unpar.ac.id	Ya
2017730022@student.unpar.ac.id	Tidak
2017730028@student.unpar.ac.id	Ya
2017730047@student.unpar.ac.id	Ya
2017730067@student.unpar.ac.id	Tidak
7316045@student.unpar.ac.id	Tidak
7316024@student.unpar.ac.id	Ya
7316008@student.unpar.ac.id	Ya
2017730070@student.unpar.ac.id	Tidak
2017730011@student.unpar.ac.id	Tidak
7314053@student.unpar.ac.id	Ya
2017730037@student.unpar.ac.id	Ya
2017730008@student.unpar.ac.id	Tidak
2017730054@student.unpar.ac.id	Ya
2017730044@student.unpar.ac.id	Ya
7316025@student.unpar.ac.id	Ya
2017730069@student.unpar.ac.id	Ya
2017730010@student.unpar.ac.id	Ya
2017730043@student.unpar.ac.id	Ya

Alamat email	Jika Anda belum login saat membuka url yang dicantumkan di notifikasi pengumuman, apakah Anda diarahkan kembali ke url tersebut?
7316003@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
7316027@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730052@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
7316028@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730027@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
7316032@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
7316089@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
2017730006@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
7314075@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
7314028@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730068@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
7316006@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730009@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
7315064@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730088@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730038@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
2017730077@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
7316018@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730028@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730047@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
7316024@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
7316008@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
7314053@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730037@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730054@student.unpar.ac.id	Tidak, saya diarahkan ke halaman utama atau halaman lain.
2017730044@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
7316025@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730069@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730010@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.
2017730043@student.unpar.ac.id	Ya, saya diarahkan kembali ke url pengumuman yang bersangkutan.

Alamat email	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	Skor SUS
2017730023@student.unpar.ac.id	4	2	4	4	3	2	4	1	4	2	70
keenan.leman@student.unpar.ac.id	5	1	5	1	3	3	5	1	5	1	90
7316003@student.unpar.ac.id	4	3	4	2	4	3	4	2	4	2	70
2017730024@student.unpar.ac.id	3	3	3	4	3	3	3	3	3	3	47.5
7316027@student.unpar.ac.id	4	2	4	4	4	3	3	2	4	4	60
2017730060@student.unpar.ac.id	4	2	4	3	3	3	3	3	3	3	57.5
2017730052@student.unpar.ac.id	4	4	4	3	4	3	5	2	3	2	65
7316028@student.unpar.ac.id	3	2	2	1	4	3	3	3	3	3	57.5
2017730027@student.unpar.ac.id	4	3	4	4	4	3	4	4	3	4	52.5
7316032@student.unpar.ac.id	3	4	3	1	3	3	4	2	3	3	57.5
7316089@student.unpar.ac.id	3	4	3	3	2	3	3	3	3	3	45
2017730006@student.unpar.ac.id	4	4	4	4	4	4	4	4	4	4	50
7314075@student.unpar.ac.id	3	3	3	3	3	3	3	3	3	4	47.5
2017730003@student.unpar.ac.id	4	2	4	2	4	3	4	2	4	1	75
7314028@student.unpar.ac.id	4	2	4	2	4	2	4	2	4	2	75
2017730068@student.unpar.ac.id	3	4	4	4	4	4	4	2	4	4	52.5
7316006@student.unpar.ac.id	3	2	4	3	4	3	3	2	3	2	62.5
2017730009@student.unpar.ac.id	4	1	3	3	3	2	4	3	3	3	62.5
2017730035@student.unpar.ac.id	3	2	4	3	3	3	2	3	2	3	50
2017730015@student.unpar.ac.id	3	3	4	2	3	3	3	3	3	3	55
7315064@student.unpar.ac.id	5	5	5	5	5	4	4	4	4	5	50
2017730088@student.unpar.ac.id	5	1	5	3	3	3	4	2	4	2	75
2017730038@student.unpar.ac.id	5	2	2	5	5	3	3	3	3	3	55
2017730077@student.unpar.ac.id	4	2	4	2	3	3	3	3	4	3	62.5
7316018@student.unpar.ac.id	3	3	3	3	3	3	3	3	3	3	50
2017730022@student.unpar.ac.id	4	4	3	4	2	3	3	3	3	2	47.5
2017730028@student.unpar.ac.id	4	2	3	3	3	2	4	2	4	4	62.5
2017730047@student.unpar.ac.id	3	3	3	3	3	3	3	3	3	3	50
2017730067@student.unpar.ac.id	4	2	4	2	3	3	3	3	3	2	62.5
7316045@student.unpar.ac.id	4	3	3	2	3	3	4	2	3	3	60
7316024@student.unpar.ac.id	3	3	3	2	3	4	4	4	4	4	50
7316008@student.unpar.ac.id	3	2	4	1	4	2	3	2	3	1	72.5
2017730070@student.unpar.ac.id	5	2	4	4	4	2	4	2	4	2	72.5
2017730011@student.unpar.ac.id	3	3	4	2	3	4	2	2	3	2	55
7314053@student.unpar.ac.id	5	3	4	1	4	3	4	1	4	1	80
2017730037@student.unpar.ac.id	3	3	3	3	3	3	3	3	3	3	50
2017730008@student.unpar.ac.id	5	2	4	4	4	3	4	2	3	4	62.5
2017730054@student.unpar.ac.id	4	3	3	2	2	2	2	2	3	1	60
2017730044@student.unpar.ac.id	4	2	5	2	3	2	2	2	4	3	67.5
7316025@student.unpar.ac.id	3	2	3	2	3	2	3	2	3	1	65
2017730069@student.unpar.ac.id	4	3	3	4	3	3	3	3	3	3	50
2017730010@student.unpar.ac.id	3	3	3	2	4	2	3	3	3	4	55
2017730043@student.unpar.ac.id	5	3	4	2	4	3	4	3	5	4	67.5
pascal@unpar.ac.id	4	3	4	2	3	2	4	2	4	2	70

Keterangan :

- (1) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya ingin menggunakan fitur ini" ?
- (2) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu rumit" ?
- (3) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini mudah digunakan" ?
- (4) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir saya butuh bantuan teknisi untuk menggunakan fitur ini" ?
- (5) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir berbagai fungsi yang ada di fitur ini terintegrasi dengan baik" ?
- (6) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak hal yang TIDAK konsisten pada fitur ini" ?
- (7) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir banyak orang yang akan langsung mengerti cara memakai fitur ini" ?
- (8) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini terlalu sulit untuk digunakan" ?
- (9) = Apakah Anda setuju dengan pernyataan ini : "Saya pikir fitur ini sangat nyaman digunakan" ?
- (10) = Apakah Anda setuju dengan pernyataan ini : "Saya perlu belajar banyak hal terlebih dahulu sebelum dapat menggunakan fitur ini" ?

Alamat email	Saran Anda untuk fitur ini
keenan.leman@unpar.ac.id	Tidak bisa "add friend" karena akun telah mencapai batas banyak teman.
7316027@student.unpar.ac.id	Lebih diperbagus tampilannya sehingga menarik user untuk menggunakannya
7314028@student.unpar.ac.id	Good Job!
2017730022@student.unpar.ac.id	terdapat beberapa error dalam login, dimana email saya (@student.unpar) tidak memiliki akses
2017730047@student.unpar.ac.id	memperbaiki url login yang dikirimkan di shadowtape
2017730011@student.unpar.ac.id	saran saya, usahakan tidak menggunakan aplikasi luar lain seperti LINE. dikarenakan tidak semua orang menggunakan line
2017730037@student.unpar.ac.id	Link yang dibuka langsung dari line tidak memiliki hak akses
2017730008@student.unpar.ac.id	Fitur ini bagus, memudahkan mahasiswa dalam menerima notifikasi email. Namun, waktunya sebaiknya tidak hanya pada jam 12 siang karena waktu tersebut adalah saat orang-orang makan siang yang mungkin saja sedang tidak membuka gadget/aplikasi Line pada khususnya.
2017730044@student.unpar.ac.id	Log in masih tidak dapat dilakukan, terdapat pernyataan bahwa "alamat email" tidak memiliki hak akses ke pengumuman. Contohnya email yang saya gunakan "2017730044@student.unpar.ac.id tidak memiliki hak akses ke pengumuman"
7316025@student.unpar.ac.id	Lebih user friendly