

## TEKNIK PENGUJIAN PERANGKAT LUNAK DALAM EVALUASI SISTEM LAYANAN MANDIRI PEMANTAUAN HAJI PADA KEMENTERIAN AGAMA PROVINSI JAWA TENGAH

**Danang Wahyu Utomo**

Fakultas Ilmu Komputer, Program Studi Teknik Informatika  
Universitas Dian Nuswantoro Semarang  
Email: danang.wu@dsn.dinus.ac.id

**Defri Kurniawan**

Fakultas Ilmu Komputer, Program Studi Teknik Informatika  
Universitas Dian Nuswantoro Semarang  
Email: defri.kurniawan@dsn.dinus.ac.id

**Yani Parti Astuti**

Fakultas Ilmu Komputer, Program Studi Teknik Informatika  
Universitas Dian Nuswantoro Semarang  
Email: yani.parti.astuti@dsn.dinus.ac.id

### ABSTRAK

Permasalahan sistem layanan haji tidak hanya fokus pada antrian panjang pendaftar haji, namun juga pada layanan pantauan keberangkatan dan pemulangan, rencana perjalanan haji, layanan kesehatan, dan penyediaan peta shalawat bagi jamaah. Kementerian Agama Provinsi Jawa Tengah telah mengembangkan prototipe sistem mandiri layanan haji yang berfokus pada pantauan keberangkatan dan pemulangan, rencana perjalanan haji, jamaah sakit dan wafat, dan peta shalawat. Pada pengembangan sebelumnya evaluasi dilakukan hanya menggunakan *test case* kebutuhan fungsional. Evaluasi hanya mengecek apakah kebutuhan fungsional sudah di tambahkan pada prototipe sistem; bagaimana progres pengembangannya. Pada penelitian ini, kami mengusulkan teknik pengujian perangkat lunak untuk mengevaluasi keseluruhan prototipe sistem yang terdiri dari *white box testing*, *black box testing*, dan *user acceptance testing*. Teknik pengujian diusulkan untuk mengetahui *bug* dan kesalahan – kesalahan yang terjadi pada internal dan eksternal sistem. Selain itu, kami melibatkan para petugas TPHD untuk mendapatkan hasil evaluasi berdasarkan kebutuhan pengguna. Pada pengujian *white box* diperoleh bahwa *source code* memiliki derajat kompleksitas yang rendah, artinya sistem tidak terlalu banyak memberikan alternatif halaman. Pada pengujian *black box* diperoleh bahwa secara umum masukan dan luaran sistem telah sesuai dengan hasil yang diharapkan. Fitur dan informasi yang disediakan tidak memberikan makna ganda dalam penggunaannya. Pada pengujian UAT, kami menggunakan kuesioner dengan *likert scale* skala 5. Hasilnya, para responden setuju (diatas 56%) bahwa secara keseluruhan sistem layanan haji dapat membantu petugas TPHD. Meskipun, masih terdapat beberapa kelemahan yaitu, sistem kurang *responsive* dan fungsionalitas sistem perlu diperbaiki.

**Kata kunci:** haji; sistem pemantauan; keberangkatan dan pemulangan haji; kementerian agama; *testing*; *software engineering*.

### ABSTRACT

*The problem of hajj system not only focuses on long queues of pilgrims but also on departure and arrival monitoring service; pilgrimage plans; healthcare; and provision of maps of shalawat. Ministry of religious affairs central java province has developed prototype hajj service systems that focuses on monitor departure and arrival; pilgrimage plans; pilgrim sick and dead; map of shalawat. In the last development, evaluation just performed using test case-functional requirement. The evaluation only check whether functional requirement have been added on prototype system; how the development progress. In this work, we proposed software testing technique to evaluate overall system consists of white box testing; black box testing; and user acceptance testing. The testing technique is proposed to determine bug and error that occur on the internal and external system. In addition to, we involve TPHD staff to get the evaluation results based on user needs (pilgrim). In white box testing obtained that source code has low complexity, it means that the system does not provide too many alternative pages. In black box testing obtained that generally, input and output in accordance with expected result. Feature and information does not have ambiguity. In UAT, we use questionnaire with likert scale-5 scale. The result show that the respondent agree above 42.6% the overall hajj service system can help TPHD staff. Although, the systems are still less responsive and functionality of system need to maintain.*

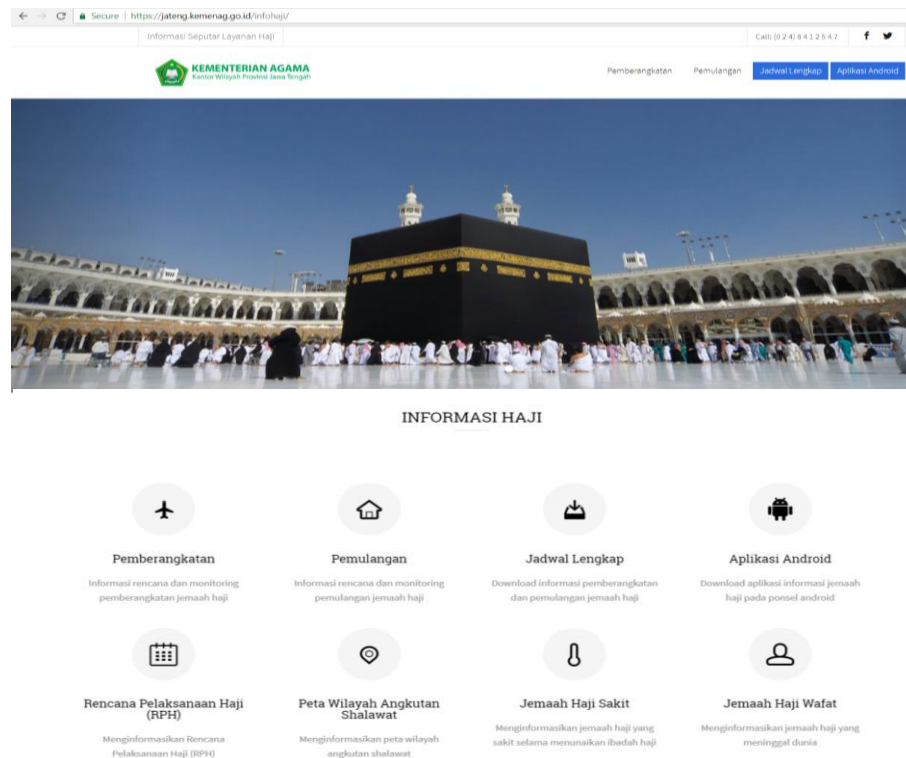
**Keywords:** haji; monitoring system; departure and arrival; ministry of religious affairs; *testing*; *software engineering*.

## 1 PENDAHULUAN

Haji di Indonesia selalu mengalami peningkatan jumlah pendaftar di tiap tahunnya. Salah satunya adalah Provinsi Jawa Tengah yang mendapatkan jumlah pendaftar terbanyak di Indonesia. Berdasarkan berita online "<https://www.jawapos.com/radarsemarang/read/2018/05/01/69416/jamaah-haji-jateng-30471-kloter-terbanyak>". Provinsi Jawa Tengah mendapatkan jumlah kloter terbanyak di Indonesia yaitu 95 kloter dengan kuota jamaah haji sebanyak 30.471 jamaah. Adanya peningkatan jumlah pendaftar ini tentunya menambah daftar antrian panjang (*waiting list*) pendaftar jamaah haji yang dapat berdampak pada lamanya waktu antrian (umumnya 5 sampai 6 tahun). Selain itu, penyelesaian masalah di masing – masing provinsi berbeda, hal ini yang menyebabkan tidak sinkronisasinya informasi antar petugas haji ke Kementerian Agama pusat. Permasalahan seperti pantauan haji, informasi yang tidak *up to date* dan ketidaksiapan petugas TPHD dalam membuat laporan operasional masih menjadi masalah utama bagi Kementerian Agama. [1] menyatakan bahwa masalah tentang pantauan jamaah haji masih menjadi topik utama bagi pemerintah. Perlu adanya perbaikan pelayanan dalam meningkatkan indeks kepuasan terhadap layanan jamaah haji [2]. Kecil kemungkinan untuk memberikan layanan dengan proses manual yaitu tanpa melibatkan sistem (proses dari komputer) [3]. adanya alat bantu seperti sistem layanan informasi khusus haji sangat membantu petugas haji (TPHD) dalam memandu jamaah haji.

Adanya permasalahan yang berbeda – beda di masing – masing provinsi, Menteri Agama mengeluarkan kebijakan bahwa setiap Kementerian Agama harus mengembangkan sistem mandiri dengan konten berdasarkan permasalahan yang ada di masing – masing provinsi. Kementerian Agama Provinsi Jawa Tengah telah melakukan perbaikan guna meningkatkan indeks kepuasan jamaah haji. Pada awal 2018, kami telah mengembangkan prototipe sistem mandiri untuk Kementerian Agama Provinsi Jawa Tengah dengan fokus layanan mandiri haji [4]. Prototipe tersebut fokus pada layanan pemantauan haji dimana dikhususkan pada pemantauan data jamaah haji, sistem keberangkatan dan pemulangan, serta penyesuaian jadwal petugas haji (TPHD). Permasalahan seperti tidak sinkronnya informasi antar petugas haji, laporan operasional yang *out of date*, jadwal petugas yang tidak diperbarui, dan kurang akuratnya data jamaah di provinsi dengan di Kementerian Agama pusat dapat diminimalisir. Sistem yang diusulkan mampu melengkapi dan memperbaiki sistem layanan haji sebelumnya.

Secara teknis, sistem yang kami usulkan (Gambar 1) mampu memberikan alternatif solusi bagi Kementerian Agama Provinsi Jawa Tengah terkait identifikasi, pemantauan dan keakuratan data. Keuntungannya, para petugas dapat saling memantau kejadian – kejadian selama ibadah haji berlangsung dan mampu memberikan keakuratan jadwal petugas haji terkait keberangkatan dan pemulangan haji. Pada sistem sebelumnya, jadwal petugas hanya di *sharing* melalui *pdf* via *whatsapp*, *BBM*. Kelemahannya, tidak ada *update* jadwal jika ada salah satu petugas yang ijin atau mengajukan tukar jadwal. Keuntungan lainnya, sistem memudahkan petugas haji dalam memberikan laporan operasional langsung kepada Kepala Kementerian Agama. Hasil laporannya juga sesuai dengan kejadian – kejadian di waktu yang sama. Sistem mampu mengurangi keterlambatan petugas dalam memberikan laporan kepada Kementerian Agama tiap harinya. Sistem yang telah diusulkan masih memiliki kelemahan karena masih dikembangkan menggunakan metode pengembangan sistem prototipe. Masih terdapat modul – modul yang belum diterapkan pada sistem. Tentunya, fungsi yang dijalankan juga belum berjalan maksimal. Sebagai contoh, modul layanan rencana pelaksanaan haji, peta wilayah angkutan, dan berbagai fitur pendukung belum diterapkan pada sistem. Untuk mengetahui kelemahan dan kekurangan pada sebuah sistem, perlu dilakukan pengujian sistem. Pengujian perangkat lunak bertujuan untuk mengetahui komponen apa saja yang sesuai dengan kebutuhan fungsional sistem, kekurangan apa saja yang harus diperbaiki, dan mengetahui apakah pengguna khususnya petugas haji (TPHD) mudah dalam menggunakan sistem. Strategi dan teknik pengujian perangkat lunak sangat diperlukan untuk menentukan kualitas dari perangkat lunak [5]. Pada artikel (penelitian [4]) sebelumnya, pengujian masih menggunakan *beta testing* dan *unit testing* yaitu pengujian untuk mengetahui apakah komponen kebutuhan fungsional telah diterapkan pada sistem, berapa persen keberhasilan komponen yang telah diterapkan pada sistem.



Gambar 1. Sistem Layanan Mandiri Haji Kemenag Prov. Jawa Tengah [4]

Pada penelitian ini, kami mengusulkan penggunaan teknik pengujian perangkat lunak sebagai evaluasi sistem mandiri Kementerian Agama Provinsi Jawa Tengah pada layanan pemantauan haji. Teknik pengujian meliputi pengujian ulang *unit testing*, *white box*, *black box*, dan *User Acceptance Testing*. Penggunaan teknik pengujian perangkat lunak bertujuan meningkatkan indeks kepuasan jamaah haji dalam menggunakan sistem layanan mandiri. Selain itu, sistem dapat digunakan sebagai alat bantu petugas haji yang masih baru dalam TPHD tahun ini. Adanya teknik pengujian perangkat lunak yang diusulkan, diharapkan mampu meningkatkan kualitas layanan sistem mandiri sehingga indeks kepuasan jamaah haji naik.

## 2 METODOLOGI PENELITIAN

Pengujian perangkat lunak merupakan bagian dari *Software Development Life Cycle* (SDLC) yang wajib dilakukan untuk mengidentifikasi semua kesalahan dan kelengkapan kebutuhan fungsional / non-fungsional. Meskipun tidak semua kesalahan (*error*) tidak dapat diidentifikasi, setidaknya dapat mengurangi kesalahan dari semua fungsi dalam sistem. Teknik pengujian terdiri dari pengujian statis dan dinamis [6]. Pengujian statis digunakan sebagai verifikasi sebelum *code deployment* seperti spesifikasi kebutuhan perangkat lunak, desain dokumen, *source code*, dan konten *web page*. Pengujian dinamis digunakan sebagai validasi pengujian fungsional dan struktural.

Pada penelitian ini kami menggunakan teknik pengujian dinamis sebagai pengujian sistem layanan mandiri haji. Pada penelitian sebelumnya, kami hanya menggunakan *test case* sebagai dasar pengujian untuk mengetahui bahwa komponen yang digunakan sudah sesuai dengan kebutuhan fungsional sistem dan mengetahui tingkat keberhasilan dari modul atau fungsi (*source code*). Pada penelitian ini, kami melanjutkan pengujian perangkat lunak dengan menggunakan teknik pengujian *white box*, *black box*, dan metode survey (*user acceptance testing*).

*White box testing* merupakan pengujian yang difokuskan pada internal sistem yaitu *source code program*. Tujuan dari pengujian *white box* digunakan sebagai alat uji kompleksitas dari code program. Bagi programmer, *white box* sangat penting untuk menentukan kompleksitas dari suatu code. Pengujian *white box* juga dapat digunakan sebagai validasi apakah *source code* mengikuti desain; apakah *source code* sesuai dengan kebutuhan fungsional; apakah *source code* memiliki kerentanan [7]. Pada sistem layanan haji, *white box* diterapkan di semua class seperti class keberangkatan, pemulangan, rencana perjalanan haji, jamaah sakit dan wafat, dan class peta shalawat. Pengujiannya menggunakan *basis path* yaitu penomoran pada *source code* yang fungsinya untuk mengetahui kompleksitas code. Setelah *basis path* diperoleh, penomoran tersebut digunakan dalam penghitungan *cyclometric complexity*. Jika hasil penghitungan dibawah 10, maka kompleksitas program masih bagus.

Pengujian *black box* biasanya digunakan untuk menguji pekerjaan internal aplikasi tanpa pengetahuan pemrograman. Pengujian *black box* digunakan untuk menguji fungsional maupun input output pada aplikasi. Teknik pengujian ini ditujukan kepada para penguji yang tidak memiliki pemahaman dalam pemrograman. Biasanya, pengujian *black box* diterapkan di semua level pengujian perangkat lunak seperti: *unit testing*, *integration testing*, *functional testing* dan *acceptance testing*. Fokus utama dalam pengujian *black box* adalah mengetahui input dalam sistem; luaran yang diharapkan; dan hasil nyata berdasarkan input dari program [7]. Pada sistem layanan haji, pengujian *black box* ditujukan kepada fungsional program seperti menguji rencana keberangkatan jika diberi masukan kosong (*blank*); data valid; atau data di luar *range*.

*User acceptance testing (UAT)* merupakan pengujian yang ditujukan di luar sistem yaitu *user*. Pengujian ditujukan pengguna yang berkaitan dengan sistem layanan haji yaitu petugas TPHD. Tujuan dari *user acceptance testing* adalah untuk mengetahui kelayakan dari perangkat lunak [8]. Secara teknis, pengujian *white box* dan pengujian *black box* cukup untuk menentukan apakah perangkat lunak layak di rilis kepada pengguna. Namun, adanya UAT dapat mengetahui kesalahan – kesalahan yang tidak diketahui pada pengujian *white box* dan *black box*. Pada penelitian ini, UAT dilakukan dengan metode survey yaitu dengan menyebarkan kuesioner kepada pengguna (petugas TPHD) yang sebelumnya sudah diberikan tutorial penggunaan sistem layanan haji. Model kuesioner menggunakan *likert scale* dengan skala 5 yaitu *strongly agree*; *agree*; *neutral/undecided*; *disagree*; *strongly disagree*. UAT digunakan untuk menjawab permasalahan perangkat lunak seputar *system metric*; *usability*; *satisfaction* dan beberapa *setting* pada masing – masing fungsi/fitur.

### 3 HASIL PENGUJIAN DAN PEMBAHASAN

#### 3.1 White Box testing

Pengujian *white box* hanya dilakukan pada *class – class* utama yaitu *class* yang sesuai dengan kebutuhan fungsional perangkat lunak. Kebutuhan fungsional seperti fitur keberangkatan, pemulangan, rencana perjalanan haji, jamaah sakit dan wafat, dan peta shalawat. Berikut adalah **contoh** pengujian *white box* pada *source code* ‘*class keberangkatan*’ dengan *basis path*:

```
1. <?php
   defined('BASEPATH') OR exit('No Direct Script Allowed');
   class Keberangkatan extends CI_Controller{
       public function __construct(){
           parent::__construct();
       }

2.   public function index()
       {
           $data['data']=$this->db->query("SELECT * FROM info_kloter a WHERE
           a.`idEvent` IN (SELECT id FROM info_event WHERE info_event.`enabled` =
           1)")->result();
           $data['title'] = 'SETUP KEBERANGKATAN';
           $this->load->view('templatesadmin/top1', $data);
           $this->load->view('templatesadmin/header');
           $this->load->view('templatesadmin/sidemenu');
           $this->load->view('admin/keberangkatan/keberangkatan', $data);
           //$this->load->view('templatesadmin/footer');
           //$this->load->view('templatesadmin/bottom-keberangkatan');
       }

3.   public function detailkeberangkatan($id)
       {
           $data['data'] = $this->db->query('Select * From info_kloter where
           id=$id')->result();
           $data['title']="Detail Keberangkatan Kloter";
           $this->load->view('templatesadmin/top', $data);
           $this->load->view('templatesadmin/header');
           $this->load->view('templatesadmin/sidemenu');
           $this->load->view('admin/vdetailkeberangkatan', $data);
           $this->load->view('templatesadmin/footer');
           $this->load->view('templatesadmin/bottom-vdetailkeberangkatan');
       }

4.   public function fkeberangkatan($id)
       {
           $this->db->where('id', $id);
```

```
$data['data']=$this->db->get('info_kloter')->result();
$data['title']="Rubah Kloter";
$this->load->view('templatesadmin/top1', $data);
$this->load->view('templatesadmin/header');
$this->load->view('templatesadmin/sidemenu');
$this->load->view('admin/keberangkatan/kloterkeberangkatan', $data);
}

5. public function editkeberangkatan()
{
    $id=$this->input->post('id');
    $this->form_validation->set_rules('tglmasukasrama', 'Tanggal Masuk
    Asrama', 'required');
    $this->form_validation->set_rules('jammasukasrama', 'Jam Masuk Asrama',
    'required');
    $this->form_validation->set_rules('tglberangkatdariasrama', 'Tanggal
    Berangkat Dari Asrama', 'required');
    $this->form_validation->set_rules('jamberangkatdariasrama', 'Jam
    Berangkat Dari Asrama', 'required');
    $this->form_validation->set_rules('bandarakeberangkatan', 'Bandara
    Keberangkatan', 'required');
    $this->form_validation->set_rules('noflight', 'No Flight', 'required');

6. if ($this->form_validation->run() == false) {
    $this->db->where('id', $id);
    $data['data']=$this->db->get('info_kloter')->result();
    $data['title']="Rubah Kloter";
    $this->load->view('templatesadmin/top1', $data);
    $this->load->view('templatesadmin/header');
    $this->load->view('templatesadmin/sidemenu');
    $this->load->view('admin/keberangkatan/kloterkeberangkatan', $data);
}

7. else{
    $tglMasukAsrama = $this->input->post('tglmasukasrama');
    $jamMasukAsrama = $this->input->post('jammasukasrama');
    $tglBerangkatdrAsrama = $this->input->post('tglberangkatdariasrama');
    $jamBerangkatdrAsrama = $this->input->post('jamberangkatdariasrama');
    $tglTibadiMakkah = $this->input->post('tgltibadimakkah');
    $jamTibadiMakkah = $this->input->post('jamtibadimakkah');
    $bandaraBerangkat = $this->input->post('bandarakeberangkatan');
    $noFlight_Berangkat= $this->input->post('noflight');
    $noMKT_Makkah = $this->input->post('nomktmakkah');
    $noRMH_Makkah = $this->input->post('normhmakkah');
    $KPST_Makkah = $this->input->post('kpstmakkah');
    $wilayah_Makkah = $this->input->post('wilayahmakkah');
    $tgljamMasukAsrama = new DateTime($tglMasukAsrama.' '.$jamMasukAsrama);
    $tgljamBerangkatdrasrama = new DateTime($tglBerangkatdrAsrama.'
    '.$jamBerangkatdrAsrama);
    $tgljamtibadimakkah= new DateTime($tglTibadiMakkah.' '.$jamTibadiMakkah);

    $this->db->set('tglMasukAsrama', $tgljamMasukAsrama->format('Y-m-d
    H:i:s'));
    $this->db->set('tglBerangkatdrAsrama', $tgljamBerangkatdrasrama-
    >format('Y-m-d H:i:s'));
    $this->db->set('tglTibaBandaraBerangkat', $tgljamtibadimakkah->format('Y-
    m-d H:i:s'));
        $this->db->set('bandaraBerangkat', $bandaraBerangkat);
        $this->db->set('noFlight_Berangkat', $noFlight_Berangkat);
        $this->db->set('noMKT_Makkah', $noMKT_Makkah);
        $this->db->set('noRMH_Makkah', $noRMH_Makkah);
        $this->db->set('KPST_Makkah', $KPST_Makkah);
        $this->db->set('wilayah_Makkah', $wilayah_Makkah);
        $this->db->where('id', $id);
        if ($this->db->update('info_kloter'))
        {
            redirect ('keberangkatan');
        }
    }
}
```

```
    }
8.   public function renkeberangkatan()
    {
9.   $id=$this->input->post('id');
    $this->form_validation->set_rules('tgledt', 'Tanggal EDT', 'required');
    $this->form_validation->set_rules('jamedt', 'Jam EDT', 'required');

10.  if ($this->form_validation->run() == false) {
    $this->db->where('id', $id);
    $query=$this->db->get('info_kloter')->row();
    $data['namakloter'] = $query->namaKloter;

    $this->db->where('id', $id);
    $data['data']=$this->db->get('info_kloter')->result();
        $data['title']="Rencana";
        $this->load->view('templatesadmin/top1', $data);
        $this->load->view('templatesadmin/header');
        $this->load->view('templatesadmin/sidemenu');
        $this->load->view('admin/keberangkatan/rencana', $data);
11.  }else{
        $tglEDT = $this->input->post('tgledt');
        $jamEDT = $this->input->post('jamedt');
        $tgljamedt = new DateTime($tglEDT.'.'.$jamEDT);
        $this->db->set('EDT', $tgljamedt->format('Y-m-d H:i:s'));
        $this->db->where('id', $id);

        if($this->db->update('info_kloter'))
            {
                redirect ('keberangkatan');
            }
        }
    }

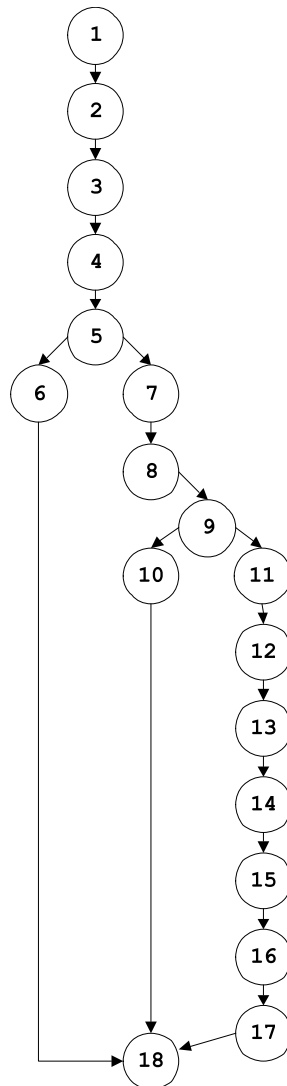
12.  public function aktkeberangkatan()
    {
13.  $this->form_validation->set_rules('tgladt', 'Tanggal ADT', 'required');
    $this->form_validation->set_rules('jamadt', 'Jam ADT', 'required');

14.  if ($this->form_validation->run() == false) {
    $id=$this->input->post('id');
    $this->db->where('id', $id);
    $query=$this->db->get('info_kloter')->row();
    $data['namakloter'] = $query->namaKloter;
    $this->db->where('id', $id);
    $data['data']=$this->db->get('info_kloter')->result();
    $data['title']="Aktualisasi";
    $this->load->view('templatesadmin/top1', $data);
    $this->load->view('templatesadmin/header');
    $this->load->view('templatesadmin/sidemenu');
    $this->load->view('admin/keberangkatan/aktualisasi', $data);

15.  }else{
        $id=$this->input->post('id');
        $tglADT = $this->input->post('tgladt');
        $jamADT = $this->input->post('jamadt');
        $tgljamadt = new DateTime($tglADT.'.'.$jamADT);
        $this->db->set('ADT', $tgljamadt->format('Y-m-d H:i:s'));
        $this->db->where('id', $id);
        if ($this->db->update('info_kloter'))
            {
                redirect ('keberangkatan');
            }
        }
    }

16.  public function faktualisasi($id)
    {
        $this->db->where('id', $id);
        $query=$this->db->get('info_kloter')->row();
        $data['namakloter'] = $query->namaKloter;
```

```
$this->db->where('id', $id);  
$data['data']=$this->db->get('info_kloter')->result();  
$data['title']="Aktualisasi";  
$this->load->view('templatesadmin/top1', $data);  
$this->load->view('templatesadmin/header');  
$this->load->view('templatesadmin/sidemenu');  
$this->load->view('admin/keberangkatan/aktualisasi', $data);  
}  
public function frenkeberangkatan($id)  
17. {  
    $this->db->where('id', $id);  
    $query=$this->db->get('info_kloter')->row();  
    $data['namakloter'] = $query->namaKloter;  
  
    $this->db->where('id', $id);  
    $data['data']=$this->db->get('info_kloter')->result();  
    $data['title']="Rencana";  
    $this->load->view('templatesadmin/top1', $data);  
    $this->load->view('templatesadmin/header');  
    $this->load->view('templatesadmin/sidemenu');  
    $this->load->view('admin/keberangkatan/rencana', $data);  
18. }
```



Gambar 2. Basis Path – Keberangkatan

Berdasarkan Gambar.2, basis path digunakan sebagian acuan *cyclometric complexity*. Kompleksitas bertujuan mengukur jalur pada *source code* keberangkatan. Pengukuran kompleksitas didasarkan pada jumlah *edge (E)* dan *node (N)* dengan menggunakan rumus:  $V(G) = E - N + 2$ . Jumlah *edge* = 19; jumlah *node* = 18. Penghitungan *cyclometric complexity*:

$$V(G) = E - N + 2$$

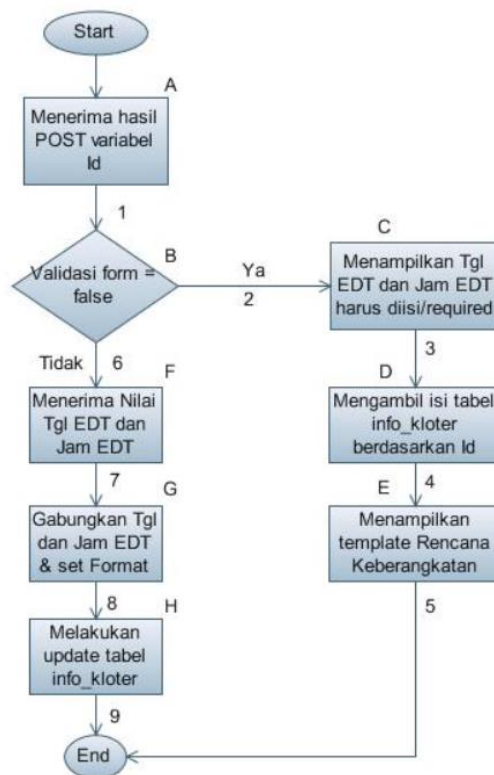
$$V(G) = 19 - 18 + 2$$

$$V(G) = 3$$

Dari hasil penghitungan diperoleh 3 jalur:

- 1) Path 1: 1-2-3-4-5-6-18
- 2) Path 2: 1-2-3-4-5-6-7-8-9-10-18
- 3) Path 3: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18

Masing – masing *path* minimal eksekusi satu kali. Jika dilihat pada stuktur kode, class Keberangkatan memiliki beberapa fungsi yang saling berkaitan. Masing – masing fungsi berpengaruh pada proses pendataan keberangkatan jamaah haji. Output yang diharapkan adalah menampilkan kebutuhan fungsional pada class keberangkatan, namun berdasarkan kondisi path yang diuji, sistem tidak selalu menampilkan hasil yang diharapkan. Ada kondisi yang menggambarkan bahwa sistem tidak dapat menampilkan data jika tidak ada inputan pada form tanggal EDT dan jam EDT.



Gambar 3. Structured Based Testing - Function renkeberangkatan()

Berdasarkan contoh pada Gambar 3, fungsi *renkeberangkatan()* merupakan fungsi yang menampilkan kebutuhan fungsional Rencana Keberangkatan Jamaah Haji. *Flowchart* tersebut menggambarkan bahwa Tgl EDT dan Jam EDT harus diisi untuk mendapatkan output yang maksimal. Fungsi ini dapat meng-*cover* semua jalur yang ada dan melakukan *update data* dengan kondisi jika input form telah valid sesuai aturan.

### 3.2 Black Box Testing

Pengujian *black box* menguji eksternal dari pekerjaan sistem seperti input/output dari sistem, hasil yang diharapkan, dan *user interface* apakah sudah sesuai dengan desain (perancangan). Pada sistem layanan mandiri haji, kami menguji berdasarkan class yang telah di uji pada *white box*. Tujuannya untuk mengetahui tingkat kesalahan dan ketidaksesuaian program. Kasus uji yang dipakai berdasarkan parameter berikut: masukan jika nilai kosong (blank); masukan jika nilai valid; dan masukan jika nilai tidak valid. Pada Tabel 1, pengujian kami arahkan pada



fungsi Keberangkatan. Materi kasus uji yang dipakai adalah menguji input pada fungsi keberangkatan dan monitoring keberangkatan. Berikut adalah contoh pengujian *black box* pada fungsi Keberangkatan:

**Tabel 1. Pengujian fungsi keberangkatan**

<i>Kasus Uji</i>	<i>Tahap Pengujian</i>	<i>Data Uji</i>	<i>Yang Diharapkan</i>	<i>Pengamatan</i>	<i>Status</i>
Menguji input rencana keberangkatan dengan data kosong (blank)	1. Memilih tombol 'Renc' 2. Menekan tombol Simpan	-	Memberi peringatan kolom kosong harus terisi	Menampilkan peringatan kolom kosong harus terisi	Pass
Menguji input rencana keberangkatan dengan data tanggal EDT dan jam valid	1. Memilih tombol 'Renc' 2. Input tanggal EDT 3. Input Jam 4. Menekan tombol Simpan	EDT: 31 Juli 2017 Jam: 07:00:00	Tanggal EDT dan Jam tersimpan sesuai input	Tanggal EDT dan Jam tersimpan sesuai input	Pass
Menguji input rencana keberangkatan dengan data tanggal EDT dan Jam tidak valid	1. Memilih tombol 'Renc' 2. Input tanggal EDT 3. Input Jam 4. Menekan tombol Simpan	EDT: 28 Feb 2018 Jam: 26:00:00	Konfirmasi tahun melebihi tahun event dan jam melebihi 24	Data tanggal EDT tetap tersimpan, jam tidak dapat diinput	Pass
Menguji input monitoring keberangkatan dengan data kosong	1. Memilih tombol 'Akt' 2. Menekan tombol Simpan	-	Memberi peringatan kolom kosong harus terisi	Menampilkan peringatan kolom kosong harus terisi	Pass
Menguji input monitoring keberangkatan dengan data tanggal ADT dan jam valid	1. Memilih tombol 'Akt' 2. Input tanggal ADT 3. Input Jam 4. Menekan tombol Simpan	ADT: 30 Juli 2017 Jam: 05:20:00	Tanggal ADT dan jam tersimpan sesuai input	Tanggal ADT dan jam tersimpan sesuai input	Pass
Menguji input monitoring keberangkatan dengan data tanggal ADT dan jam tidak valid	1. Memilih tombol 'Akt' 2. Input tanggal ADT 3. Input Jam 4. Menekan tombol Simpan	ADT: 28 Feb 2018 Jam: 26:00:00	Konfirmasi tahun melebihi tahun event dan jam melebihi 24	Data tanggal ADT tetap tersimpan, jam tidak dapat diinput	Pass

Pada fungsi kepulangan, monitoring kepulangan, rencana perjalanan haji, jamaah sakit dan wafat, dan peta shalawat *test case* yang diuji sama seperti fungsi keberangkatan. Keseluruhan fungsi pada sistem layanan haji memenuhi hasil yang diharapkan pada pengujian *black box*. Jika dilihat secara teknis, pengujian *black box* memang lebih mudah di banding *white box*. Karena para penguji hanya fokus pada masukan, hasil yang diharapkan dan luaran nyata dari input. Namun, jika pengujian dilakukan secara bertahap, misal masih ada beberapa fungsi yang tidak berhasil dijalankan, pengujian *black box* membutuhkan eksekusi berkali-kali agar fungsi – fungsi tersebut berjalan sesuai dengan hasil yang diharapkan. Keuntungannya, pengujian dapat dilakukan ketika *environment* sedang dijalankan. Selain itu, perangkat lunak yang dihasilkan lebih efisien karena diuji berdasarkan sudut pandang pengguna yang berbeda.

### 3.3 User Acceptance Testing

*User Acceptance Testing* merupakan pengujian yang melibatkan *end user*. Tujuannya untuk mengetahui apa yang sistem lakukan dan keuntungan apa yang diperoleh dari sistem berdasarkan sudut pandang pengguna akhir (*end user*). Pengujian UAT melibatkan 75 responden petugas TPHD. Pengujian

dilakukan di Kemenag Prov. Jateng dengan sebelumnya diberikan workshop tutorial penggunaan sistem layanan haji. Para responden menjawab kuesioner setelah menggunakan tutorial sistem layanan haji. Pada tabel 2 mempresentasikan materi survei kuesioner yang terdiri dari 4 bagian: *setting* fungsi; *system metric*; *usability*; dan *user satisfaction*.

**Tabel 2. Survey kuesioner**

No	Materi
1	Setting keberangkatan dan pemulangan
2	Setting Rencana Perjalanan Haji
3	Setting Jamaah Sakit
4	Setting Jamaah Wafat
5	Display Peta Shalawat Haji
6	Mudah digunakan dan dipahami
7	Interface menarik
8	Sistem Responsive
9	Fungsionalitas
10	Performa Baik

Materi 1 sampai dengan 5 merupakan fokus *setting* yang meliputi pertanyaan apakah sistem memudahkan petugas haji memasukkan data keberangkatan haji; apakah sistem mampu memantau data keberangkatan haji; apakah sistem memudahkan petugas haji memantau perkembangan kepulangan jamaah haji; apakah sistem memudahkan petugas haji mengatur Rencana Perjalanan Haji; apakah sistem mampu menginformasikan jumlah jamaah sakit dan jamaah wafat; apakah sistem memudahkan petugas haji dalam membuat laporan operasional. Pada materi 5, merupakan fungsi tambahan untuk melengkapi fitur layanan haji dalam mengatur dan menginformasikan peta sholawat pada jamaah haji.

Materi 6 dan 7 merupakan fokus *usability* yang meliputi pertanyaan apakah sistem mudah digunakan; apakah fitur – fitur sistem mudah dipahami; apakah *interface* sistem menarik dalam penggunaan warna, teks, dan *button* yang digunakan dapat dipahami pengguna.

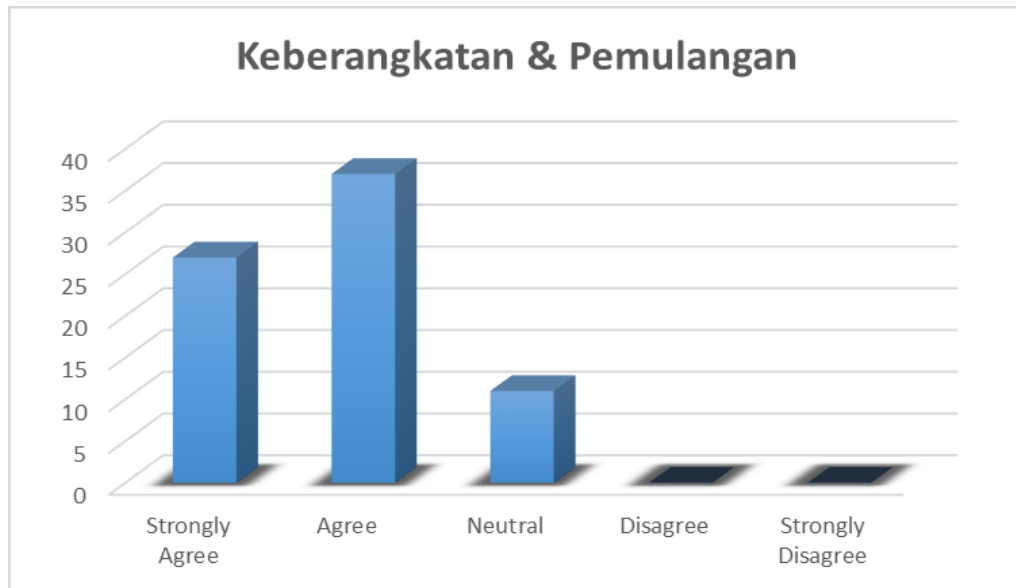
Materi 8 dan 10 merupakan fokus *system metric* yang meliputi pertanyaan apakah sistem responsive yaitu fungsi – fungsi dapat dijalankan secara optimal meliputi pencarian data, update informasi berita, dan tata letak (*layout*) yang mampu menyesuaikan resolusi layar; apakah keseluruhan fungsi dapat dijalankan dengan baik yaitu *button* yang digunakan tidak mengalami *error*.

Materi 9 merupakan fokus *user satisfaction* yang meliputi *user interface* dan *functionality*. Fokus *user satisfaction* meliputi pertanyaan apakah tampilan pengguna (jamaah haji) sesuai kebutuhan pengguna seperti halaman *login*, dashboard petugas haji, dan penyusunan tata letak menu; apakah semua menu dapat dijalankan; apakah semua menu menampilkan informasi yang sesuai; apakah informasi dapat di perbarui.

**Tabel 3. Hasil evaluasi**

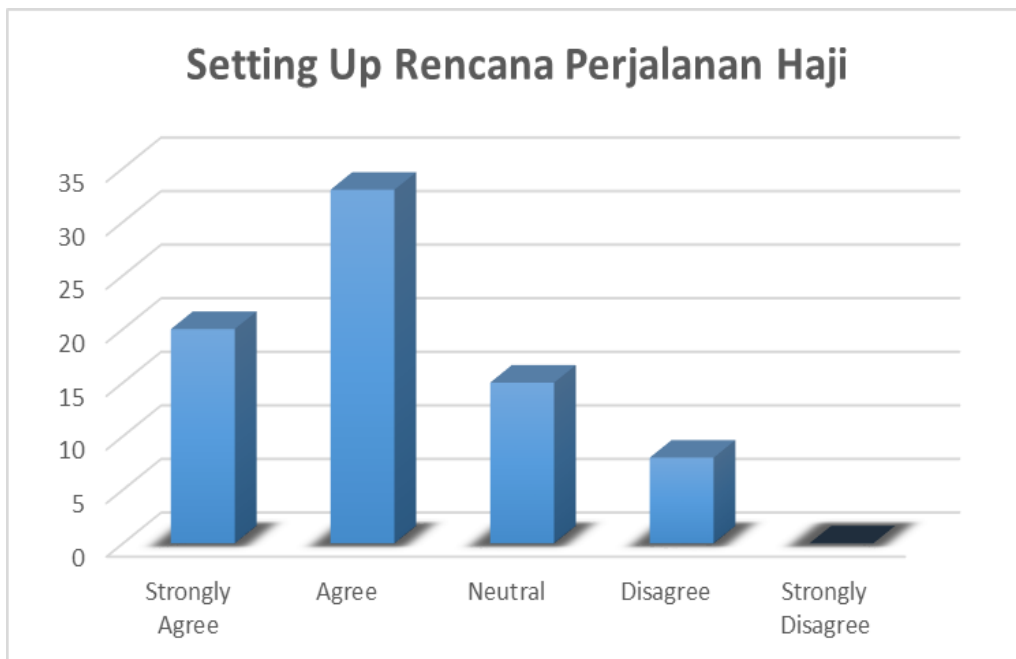
<i>Setting Up</i>	SA	A	N	D	SD
Setting keberangkatan & pemulangan	27	37	11	0	0
Setting Rencana Perjalanan Haji	20	33	15	8	0
Setting Jamaah Sakit	5	29	24	14	3
Setting Jamaah Wafat	4	25	21	18	7
Display Peta Shalawat Haji	25	34	16	0	0
<i>Usability</i>	SA	A	N	D	SD
Mudah digunakan dan dipahami	6	40	29	0	0
Interface menarik	31	25	9	0	0
<i>System Metric</i>	SA	A	N	D	SD
Sistem Responsive	3	28	30	15	0
Performa Baik	4	24	22	21	5
<i>User Satisfaction</i>	SA	A	N	D	SD
Fungsionalitas	0	17	27	21	11

Tabel 3 merupakan hasil UAT yang melibatkan petugas TPHD dalam mengevaluasi sistem layanan haji. Sebanyak 75 responden melakukan evaluasi dalam pengisian kuesioner. Diperoleh hasil kuesioner dalam bentuk *likert scale* yang akan dianalisis dalam sebuah grafik.



**Gambar 4. Grafik Setting Keberangkatan dan Pemulangan**

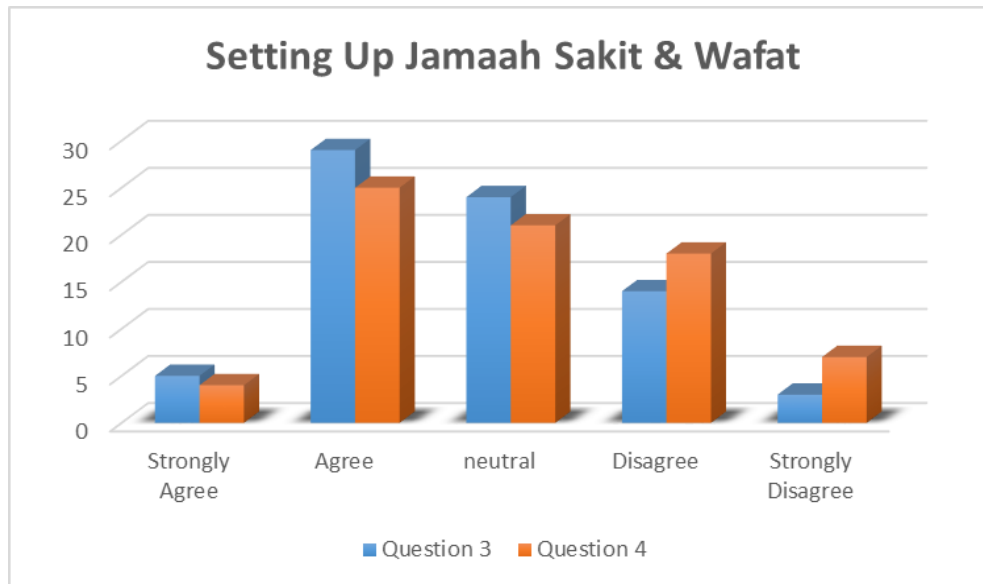
Pada gambar 4 diperoleh bahwa para responden setuju sistem layanan haji mampu digunakan sebagai alat untuk memantau jadwal keberangkatan dan pemulangan dan memudahkan petugas dalam memperbarui informasi terkait ibadah haji. Grafik menunjukkan bahwa 85.33% responden memberikan pernyataan setuju dan sisanya 14.67% menjawab netral (ragu - ragu). Kami menyimpulkan bahwa saat ini petugas TPHD membutuhkan alat untuk menunjang pekerjaannya dalam tugas haji. Terbukti, bahwa sistem layanan haji mampu membantu petugas dalam memantau perkembangan haji terkait keberangkatan dan pemulangan.



**Gambar 5. Grafik Setting Rencana Perjalanan Haji**

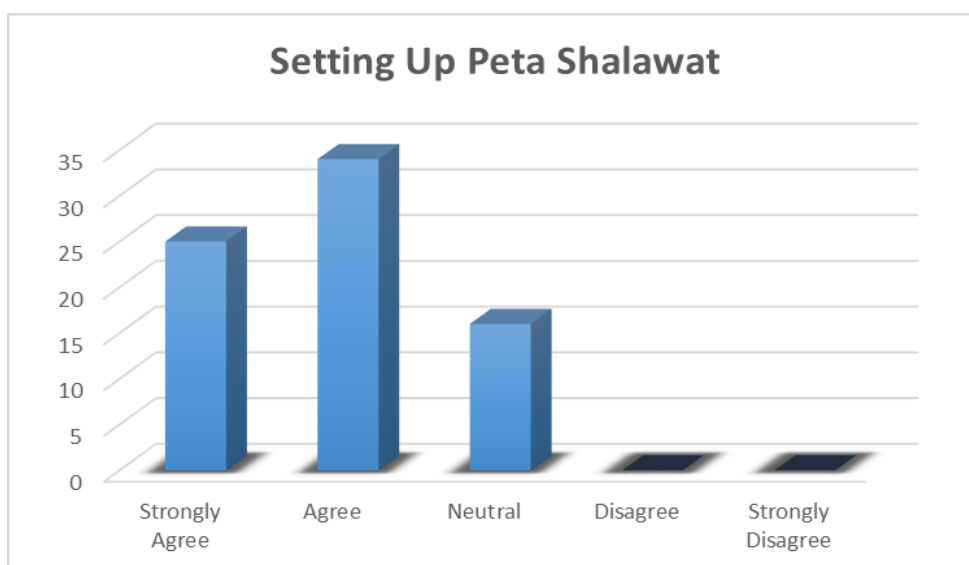
Pada Gambar 5, pertanyaan fokus pada setting rencana perjalanan haji. Grafik menunjukkan bahwa 70.67% para responden menjawab setuju; 20% ragu – ragu; dan 10,67% menjawab tidak setuju. Hal ini membuktikan bahwa saat ini sistem mampu membantu petugas haji dalam membuat Rencana Perjalanan Haji. Lebih dari 70% petugas haji dapat menggunakan dan memahami fitur Rencana Perjalanan Haji dalam sistem.

Pada Gambar 6, pertanyaan fokus pada pantauan jamaah sakit dan wafat. Pertanyaan – pertanyaan yang diberikan seputar proses pantauan jamaah sakit dan wafat yang dialami oleh petugas haji; apakah sistem mampu bersinkronisasi dengan halaman petugas medis (dokter); apakah sistem memberikan layanan interaktif terhadap semua petugas haji dan Kepala Kementerian Agama. Grafik menunjukkan bahwa responden masih ragu – ragu bahwa sistem layanan mandiri membantu petugas TPHD dalam memantau jamaah sakit dan wafat. Rata – rata responden menjawab ragu – ragu diatas 28% dan menjawab tidak setuju diatas 22%. Kami menyimpulkan bahwa saat ini sistem layanan mandiri perlu dikembangkan lebih lanjut untuk mempermudah petugas TPHD dalam memantau jamaah haji serta sistem layanan lebih interaktif.



Gambar 6. Grafik Setting Jamaah Sakit dan Wafat

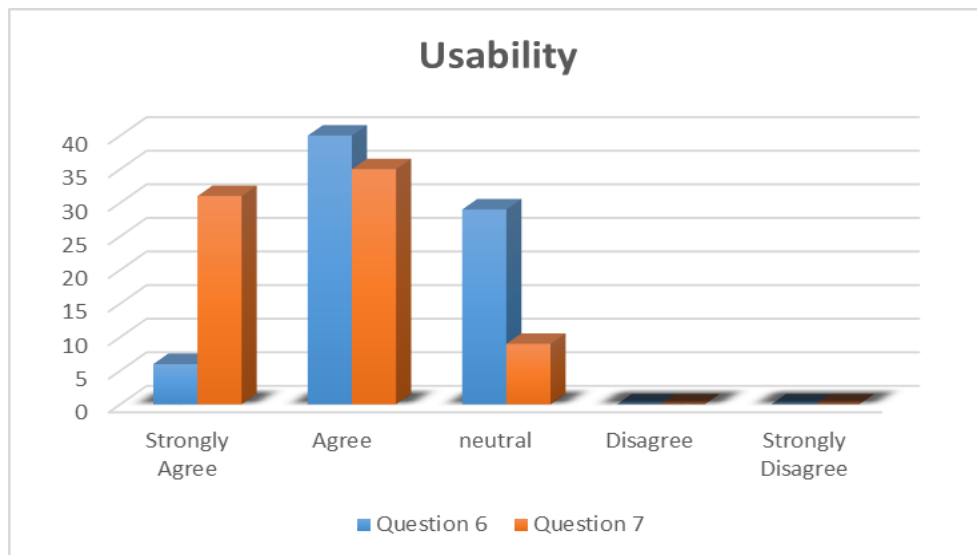
Grafik pada Gambar 7 merupakan fokus pada fitur pengadaan peta shalawat dalam sistem layanan mandiri. Pada sistem sebelumnya, peta shalawat hanya sebagai catatan bagi petugas haji jika para jamaah ada yang menanyakannya. Namun, pada tahun ini peta shalawat merupakan fitur tambahan yang memang harus ditampilkan pada sistem layanan haji. Grafik menunjukkan bahwa 78.66% para responden setuju fitur peta shalawat ditambahkan pada sistem layanan haji. Selain itu, para responden juga setuju bahwa sistem memudahkan para jamaah untuk mencari peta shalawat tanpa bertanya kepada petugas haji.



Gambar 7. Grafik Setting Peta Shalawat

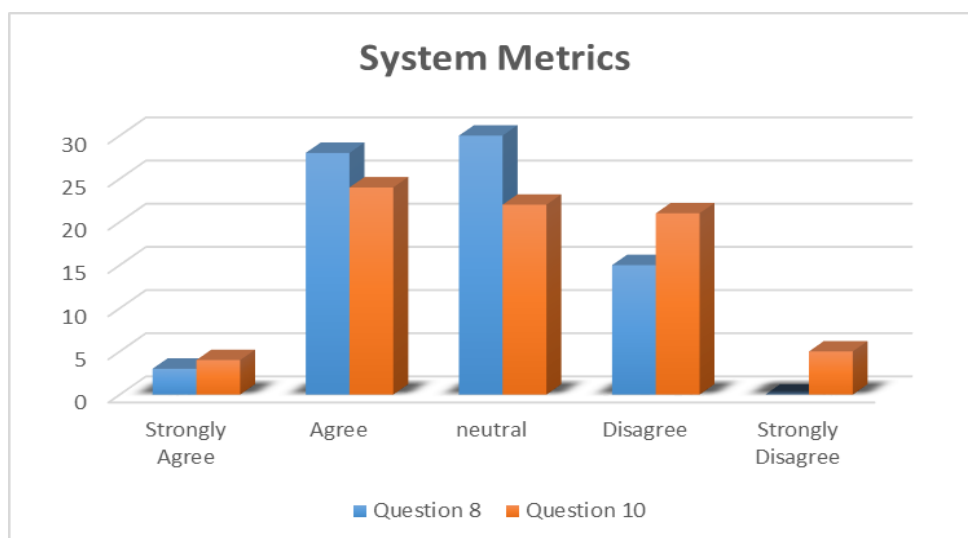
Pada halaman pengguna (*front end*), jamaah dapat melihat langsung peta shalawat tanpa harus bertanya pada petugas haji. Pada halaman pengguna, jamaah haji dapat melihat peta angkutan shalawat, rute perjalanan haji, dan informasi hotel di Arab Saudi. Berbagai rute perjalanan ditampilkan untuk memberikan kenyamanan beribadah, memberikan kebebasan bagi jamaah dalam perjalanan ibadah.

Grafik pada Gambar 8 fokus pada kemudahan dan kenyamanan pengguna dalam menggunakan sistem. Apakah petugas haji memahami dan mudah dalam menggunakan fitur, *button*, dan luaran dalam sistem. Grafik menunjukkan bahwa lebih dari 61.33% responden menjawab setuju. Kami menyimpulkan bahwa secara keseluruhan sistem layanan haji mudah digunakan dan kontennya mudah dipahami oleh petugas haji. Petugas haji menyatakan bahwa sistem layanan haji saat ini lebih mudah digunakan dari sistem sebelumnya. Tidak ada responden yang menyatakan kesulitan dalam menggunakan fitur – fitur sistem layanan haji.



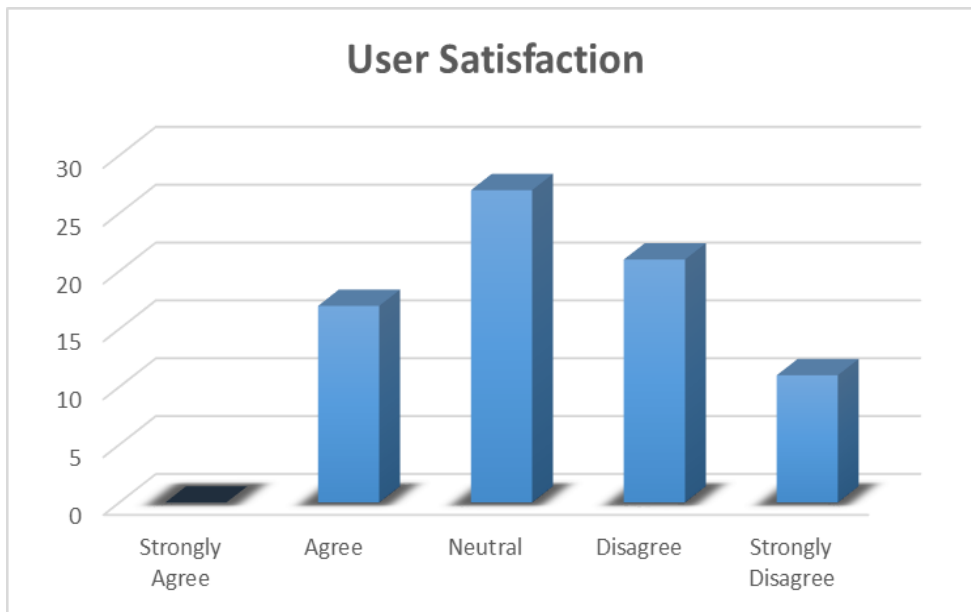
Gambar 8. Grafik Usability

Grafik pada Gambar 9 fokus pada sistem metric sistem layanan haji seperti apakah sistem layanan responsive terhadap masukan; apakah secara keseluruhan fitur dapat dijalankan dengan baik; apakah *button* selalu mengalami kesalahan. Grafik menunjukkan bahwa para responden menjawab setuju diatas 37% dan ragu – ragu diatas 57%. Kami menyimpulkan bahwa rata – rata para responden masih meragukan sistem layanan haji berjalan dengan baik. Meskipun fungsi – fungsi dapat dijalankan dengan baik, namun masih perlu ditingkatkan dalam performa dan responsive. Perlu adanya peningkatan dalam pencarian optimal mengenai informasi keberangkatannya, rencana perjalanan haji, dan peta shalawat.



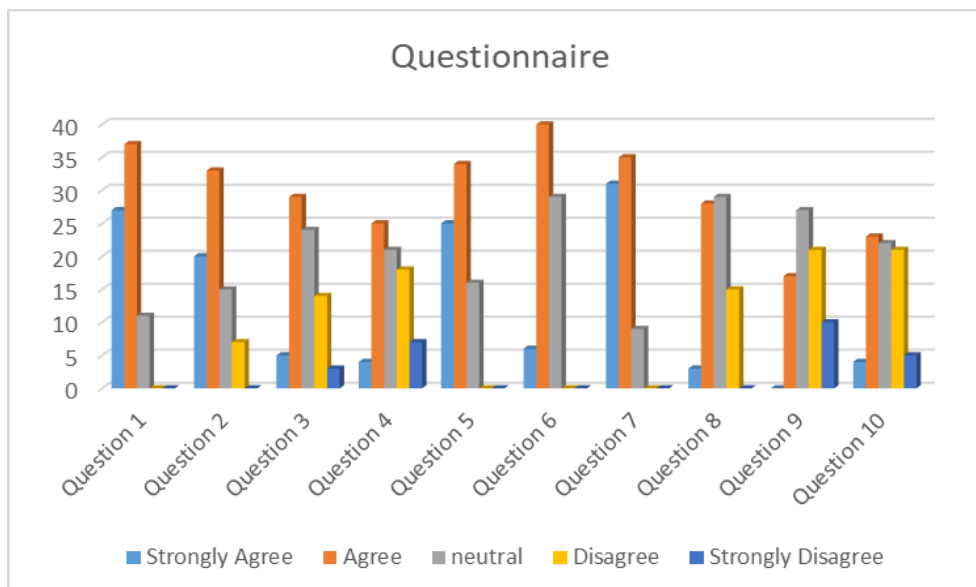
Gambar 9. System Metrics

Grafik pada Gambar 10 fokus pada pertanyaan 9 yaitu kepuasan pengguna terhadap fungsional sistem yang dijalankan; kepuasan terhadap fungsi dari fitur – fitur yang diberikan; serta kepuasan informasi yang diberikan. Grafik menunjukkan para responden menjawab setuju 22.67% dan 36% ragu – ragu. Hal ini membuktikan bahwa para petugas haji masih ragu – ragu bahwa secara keseluruhan informasi yang diberikan dapat digunakan seutuhnya dalam pemantauan haji. Masih terdapat beberapa *bug* pada fitur; *button*; dan *update* informasi berita.



**Gambar 10. User Satisfaction**

Berdasarkan pengujian UAT, secara keseluruhan kami menyimpulkan bahwa prototype sistem layanan mandiri haji masih layak digunakan dalam membantu petugas haji memantau kejadian – kejadian dalam ibadah haji tahun ini. Permasalahan mengenai keterlambatan laporan operasional, informasi keberangkatan dan pemulangan, dan peta shalawat mampu diselesaikan menggunakan sistem layanan mandiri. Meskipun, ada beberapa fitur – fitur yang harus ditingkatkan untuk meningkatkan indeks kepuasan jamaah haji.



**Gambar 11. Rata – Rata Responden**

Pada gambar 11 menunjukkan bahwa secara keseluruhan, responden menjawab setuju (bar orange) lebih tinggi di banding bar untuk *neutral*, *disagree*, dan *strongly agree*. Hal ini membuktikan bahwa para responden percaya sistem layanan mandiri dapat digunakan secara maksimal dalam memantau perkembangan ibadah haji tahun ini.

#### 4 KESIMPULAN

Teknik pengujian yang diusulkan digunakan sebagai evaluasi lanjutan pada pengembangan prototype sistem mandiri layanan haji Kementerian Jawa Tengah Provinsi Jawa Tengah. Sistem layanan haji menyediakan fungsi – fungsi dalam membantu petugas TPHD menjalankan tugasnya mendampingi dan memantau jamaah haji. Selain itu, sistem juga ditujukan kepada jamaah haji dalam update berita tentang haji, jadwal keberangkatan dan pemulangan haji, rencana perjalanan haji, serta adanya fungsi baru mengenai peta shalawat membantu jamaah dalam menentukan perjalanan. Teknik pengujian yang kami usulkan adalah *white box*, *black box*, dan *user acceptance testing*. Pengujian *white box* merupakan pengujian terhadap *source code* program, melibatkan para developer untuk mengetahui kompleksitas program. Pengujian *black box* merupakan pengujian eksternal sistem dimana pengujian melibatkan *stakeholder* untuk mengetahui apakah masukan dan luaran sesuai dengan hasil yang diharapkan. Pengujian *user acceptance testing* melibatkan *end user* yaitu petugas TPHD. Pengujian menggunakan metode survey dengan penyebaran kuesioner kepada petugas TPHD. Sebagai analisis hasil kuesioner, kami menggunakan *likert scale* untuk mengambil hasil uji. Berdasarkan hasil pengujian, diperoleh bahwa secara garis besar responden setuju diatas 56% (didasar 42 responden) bahwa sistem layanan haji yang diusulkan mampu mengatasi permasalahan layanan haji seputar laporan operasional, tidak adanya pembaruan berita haji, dan terlambatnya informasi mengenai kejadian yang terjadi. Namun, beberapa kelemahan perlu diperbaiki lebih lanjut untuk mendapatkan sistem layanan haji lebih responsive. Fungsionalitas fitur, *button*, dan luaran perlu diperbaiki untuk menjadikan sistem lebih interaktif. Prototype sistem layanan haji dapat dijadikan acuan pada pengembangan selanjutnya yaitu *mobile application*. Targetnya, pada penerapan kedepan kami fokus pada *notification system*, *controlling*, dan *responsive system*.

#### DAFTAR PUSTAKA

- [1] S. M. R. Al Masud, A. A. Bakar, and S. Yussof, "A systematic review of technological issues in monitoring pilgrims' health during hajj: Current state, challenges and future directions," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 7, pp. 1935–1950, 2018.
- [2] M. Darul and F. Pati, "EFEKTIVITAS SISTEM INFORMASI DAN KOMPUTERISASI HAJI TERPADU ( SISKOHAT ) DALAM PENYELENGGARAAN IBADAH HAJI," *J. Ilmu Dakwah*, vol. 3, no. 2, pp. 225–248, 2015.
- [3] S. Kulkarni and S. K. Shah, "Monitoring and Safety of Pilgrims Using Stampede Detection and Pilgrim Tracking," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 4, no. 7, pp. 6642–6648, 2015.
- [4] D. Kurniawan and D. W. Utomo, "PERANCANGAN LAYANAN PEMANTAUAN HAJI PADA SISTEM MANDIRI KEMENTERIAN AGAMA PROVINSI JAWA TENGAH BERBASIS WEB," vol. 2018, no. Sentika, pp. 268–277, 2018.
- [5] E. R. Subhiyakti and D. W. Utomo, "Software Testing Techniques and Strategies Use In Novice Software Teams," *J. Sist. Inf.*, vol. 5, no. 5, pp. 556–562, 2016.
- [6] M. A. Sethi, "A Review Paper On Levels, Types & Techniques in Software Testing," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 7, pp. 269–271, 2017.
- [7] M. Kumar, S. K. Singh, and R. . Dwivedi, "A Comparative Study of Black Box Testing and White Box Testing Techniques," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 3, no. 10, pp. 32–44, 2015.

- [8] C. S. Theng, "Leisure Technology for the Elderly: A Survey , User Acceptance Testing and Conceptual Design," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 12, pp. 100–115, 2017.