

ANALISIS *APPLICATION PROGRAMMING INTERFACE* PADA *MOBILE E-VOTING* MENGGUNAKAN METODE *TEST-DRIVEN DEVELOPMENT*

Sunardi¹, Imam Riadi², Pradana Ananda Raharja³

¹Program Studi Teknik Elektro, Universitas Ahmad Dahlan Yogyakarta

²Program Studi Sistem Informasi, Universitas Ahmad Dahlan Yogyakarta

³Program Studi Magister Teknik Informatika, Universitas Ahmad Dahlan Yogyakarta

Informasi Makalah

Dikirim, 8 April 2019
Direvisi, 20 September 2019
Diterima, Oktober 2019

Kata Kunci:

E-voting
Application Programming Interface
Basis data
Test-Driven Development
Error

INTISARI

Pengujian pada perangkat lunak dibutuhkan sebagai jaminan mutu terhadap kemampuan perangkat lunak. Perangkat lunak diterapkan pada pekerjaan yang nyata tanpa memiliki kendala terhadap kinerja dalam memberikan perintah, memproses perintah hingga memperoleh hasil dari pemrosesan sesuai dengan tujuan pengembangan. Pengujian unit *Application Programming Interface (API)* pada *web service* fokus sebagai upaya untuk menguji kemampuan pada tiap unit. Unit yang diuji secara spesifik akan digabungkan dengan unit lain hingga menjadi suatu sistem terintegrasi. Aplikasi *mobile* membutuhkan dukungan *API* sebagai penghubung dengan basis data, dimana aplikasi *mobile* dengan basis data tersebut tidak saling berhubungan secara langsung. Tampilan antarmuka pada aplikasi *mobile* hanya difokuskan sebagai interaksi visual antara pengguna dengan sistem. Sistem yang diberi perintah akan merespon dengan umpan balik yang dihubungkan melalui perantara dan merespon kembali melalui perantara dan dikembalikan dalam bentuk visual melalui antarmuka. Pengujian pada tiap unit sistem menggunakan proses *Test-Driven Development (TDD)* untuk menguji tiap fungsi. Perancangan dan pembangunan *web service* pada aplikasi *e-voting* memerlukan pengujian sehingga dalam penerapannya dapat digunakan sesuai dengan spesifikasi yang telah direncanakan tanpa menimbulkan *bug* atau *error* yang dapat mengganggu kinerja sistem. *API* yang telah diuji dan berinteraksi dengan antarmuka dapat melakukan proses data pemilih, validasi data pemilih dan pemilihan. Sistem ini juga telah diuji dengan metode *System Usability Scale* dengan skor total 69,02.

ABSTRACT

Testing on the software is needed to guarantee its capabilities. So software in real condition wouldn't have any defect with the performance to providing with in line with the goal of development. Testing the Application Programming Interface (API) unit on the web service is an effort to testing the performance of each unit. Specifically tested unit will be combined with other units to become an integrated system. A Mobile application requires API support as the connector to a database, which not directly related to mobile application. The user-interface in mobile application only act as a visual interaction between the user and the system. The command system will respond through feedback which connected into a connector and gives the response back through the connector and returns in visual form through an interface. Test-Driven Development (TTD) was used in testing each system unit. Designing and building web service on e-voting application requires testing so that application can be used properly without causing bugs or errors that can interface system performance. Approved APIs are able to process voter data, validate voter data and election. The system also tested with "System Usability Scale" method which gains 69,02 total score.

Keyword:

E-voting
Application Programming Interface
Database
Test-Driven Development
Errors

Korespondensi Penulis:

Pradana Ananda Raharja

Program Studi Teknik Informatika
Pascasarjana Universitas Ahmad Dahlan, Yogyakarta
Jl. Prof. Dr. Soepomo, S.H., Janturan, Warungboto, Umbulharjo, Yogyakarta, 55164
Email: pradana1807048024@webmail.uad.ac.id / anandaprada@gmail.com

1. PENDAHULUAN

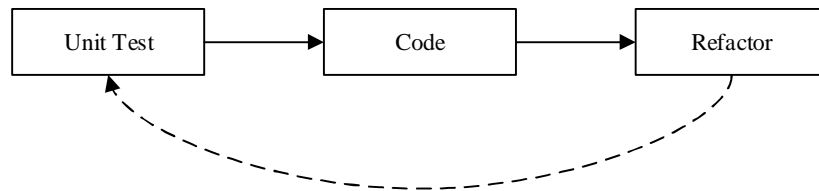
Voting adalah Suatu cara untuk memutuskan sesuatu dengan bantuan media, keputusan yang digunakan adalah keputusan dengan suara terbanyak. Media yang digunakan untuk mengumpulkan aspirasi dapat berupa media kertas yang saat ini digunakan. Hingga kini media *voting* yang masih digunakan memiliki kelemahan baik dalam prasyarat dan sarana. Data pemilih yang tidak sah, suara yang tidak sah serta proses *voting* itu masih belum efektif [1] [2]. Sistem *e-voting* yang telah dilindungi undang-undang dalam pemilihan umum dimasa mendatang terkait dengan keputusan Mahkamah Konstitusi yang menetapkan pasal 88 UU No. 32/2004 tentang Pemerintahan Daerah adalah Konstitusional selama *e-voting* menerapkan syarat Kumulatif sebagai asas pemilu yang LUBERJURDIL dapat dilaksanakan [3].

Application Programming Interface (API) memiliki manfaat yang dimungkinkan pengembang dapat mengintegrasikan antara 2 bagian aplikasi atau aplikasi yang berbeda. Pengembangan aplikasi yang membutuhkan API yang terdiri dari beberapa element seperti *function*, *protocols*, dan *tools*. API digunakan untuk mempersingkat proses pengembangan sehingga pengembang tidak perlu membuat fitur yang sama. API adalah sebuah *class* yang dirancang untuk menghubungkan antara aplikasi *mobile* dengan basis data. Melalui *class* ini dapat berkeja untuk mengakses dan mengeksekusi beragam perintah dari aplikasi *mobile* [4] [5]. *Web service* sebagai komponen penghubung antara lapis antar muka dengan *server* yang mendukung prosedur dan fungsi-fungsi yang akan memproses perintah. *Class* yang ada pada *web service* memiliki peranan tersendiri sehingga tiap *class* tersebut diujikan kemampuannya dalam memproses perintah [6]. Aplikasi *mobile* merupakan sebuah teknologi yang memungkinkan dalam memenuhi kebutuhan informasi melalui perangkat dengan ukuran tertentu. Pada umumnya sistem operasi *mobile* yang digunakan saat ini adalah android dan iOS, android yaitu sistem operasi yang berbasis Linux sedangkan iOS adalah sistem operasi yang didistribusikan oleh Apple Inc. [7] [8] [9] [10].

Pengujian perangkat lunak bertujuan untuk menilai atau meningkatkan keandalan dalam menjamin mutu kerja perangkat sehingga terhindar dari kegagalan atau *error* yang tidak diinginkan. Tanpa pengujian dan jaminan mutu pada perangkat lunak yang dirancang dan dibangun akan berakibat fatal dengan contoh kasus perangkat tersebut memiliki *logic error*, *runtime error*, dan *syntax error*. Pada *error* tersebut berakibat sistem yang dirancang tidak berjalan dengan semestinya. Sistem yang memiliki *error* atau *bug* berpengaruh terhadap kinerja seperti kesalahan pengisian data, kesalahan dalam pemrosesan, kesalahan pada hasil yang diharapkan. Maka perangkat lunak memerlukan pengujian yang mendetail pada unit yang tersedia sebelum unit-unit tersebut dihubungkan hingga menjadi sebuah sistem untuk dioperasikan [11] [12] [13]. Berdasarkan rancangan dan uji coba ditemukan kesalahan dalam *request* dan *response* pada aplikasi *mobile* ke *database* melalui API yang menyebabkan gangguan fatal, sehingga diperlukan uji coba pada fungsi yang tersedia pada API menggunakan metode TDD.

2. METODE

Metode *TDD* pada penelitian berikut digunakan sebagai pendekatan dalam menguji *API*. Metode *TDD* yaitu suatu proses pengembangan perangkat lunak untuk menguji tiap unit pada sistem dengan siklus singkat [14]. Dalam menerapkan *TDD* terdapat hal yang harus dicermati yaitu mereplika suatu kondisi atau *event* disebut dengan *Mocking*. Proses yang terdapat pada pendekatan *TDD* antara lain, pertama menyiapkan beberapa rekayasa pengujian pada suatu komponen, membuat komponen tersebut hingga lulus uji, komponen



yang lulus uji disimpan dan disusun dengan komponen lainnya, dan Semua fitur, komponen serta *source code* yang telah lolos uji akan digabungkan secara lengkap dapat dilihat pada Gambar 1 [15]. *TDD* memiliki manfaat yang didapatkan yaitu apabila terdapat *bug* dan *error* dapat diketahui dengan tepat dan perbandingan dalam menguji seluruh sistem, pendekatan *TDD* memberikan waktu yang cepat dalam penggabungan sistem [16].

Gambar 1. Alur Kerja TDD [15]

3. HASIL DAN PEMBAHASAN

3.1. Pengujian Unit

Dalam pembuatan pengujian unit yang menggunakan *tools PHPUnit* sebagai aplikasi pengujian pada *class*, *class* berikut menjadi objek dalam pengujian pada *class* diterangkan pada Gambar 2.

```
MINGW64:/c/bin
ASUS@LAPTOP-OTT2H0KN MINGW64 /c/bin
$ phpunit TestClass.php
PHPUnit 3.7.21 by Sebastian Bergmann.

Time: 0 seconds, Memory: 1.75Mb
OK (1 test, 1 assertion)
ASUS@LAPTOP-OTT2H0KN MINGW64 /c/bin
$
```

Gambar 2. Pengujian *Class*

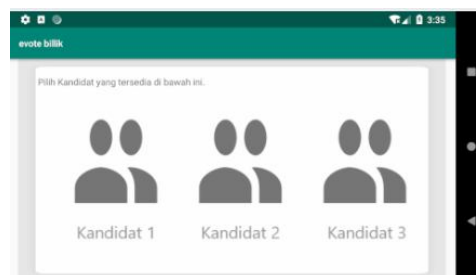
Pada Gambar 2 merupakan unit yang diuji melalui *link address* menggunakan Postman, *link address* pada Gambar 2 digunakan untuk menguji proses yang diambil dari *API* untuk dibandingkan dengan data yang ada pada *database*. *API* mengambil data seperti data kandidat, data pemilih dan data administrasi untuk diproses.

3.2. Implementasi Program

Unit pengujian yang telah dibuat kemudian diimplementasikan menjadi sebuah sistem, sistem yang diimplementasikan terdiri dari 2 buah modul, modul pada Gambar 3 merupakan modul administrasi dan Gambar 4 adalah modul bilik suara. Pada modul ke 2 merupakan implementasi sistem administrasi *e-voting* dengan fitur utama yaitu pemrosesan data pemilih, validasi data pemilih, data kandidat dan hasil pemilihan. Menu pemrosesan data pemilih untuk melakukan input data pemilih yang sah sebagai pemilih, menu validasi pemilihan digunakan sebagai validasi pemilih sah yang hadir atau sebagai presensi pemilih yang akan melakukan pemilihan, data kandidat untuk melakukan proses data kandidat sebagai kandidat yang sah dalam proses pemilihan dan data tersebut terkoneksi ke modul bilik suara, kemudian menu hasil pemilihan adalah menu yang menampilkan persentase pemilihan yang hadir selama pemilihan berlangsung dan hingga pemilihan telah ditutup maka akan secara otomatis menampilkan data pemilihan pada tiap kandidat. Pada menu ini pemilih yang telah melakukan pemilihan tidak dapat melakukan pemilihan lebih dari 1 kali dan data pemilih atas kandidat yang dipilih tidak terekam dalam *database*, akan tetapi hanya nilai kandidat yang dipilih akan bertambah. Pada Gambar 4 adalah implementasi dari modul bilik suara, pada menu tersebut peserta pemilihan yang sah yang hadir dan telah divalidasi maka pemilih dapat melakukan kegiatan pemilihan, pada modul bilik suara pemilih akan disajikan foto kandidat pemilihan. Kandidat yang dipilih oleh peserta pemilihan akan dikonfirmasi kebenaran data yang dipilih peserta pemilihan. Pada fitur konfirmasi tersebut dimaksudkan untuk meyakinkan pemilih apakah pemilih telah melakukan pemilihan sesuai prosedur.



Gambar 3. Implementasi program modul administrasi



Gambar 4. Implementasi program modul bilik suara

3.3. Hasil Pengujian

Tahap yang dilakukan setelah melakukan tahap implementasi kode adalah tahapan pengujian menggunakan metode *TTD*, hasil pengujian yang telah dilakukan dapat dilihat pada Gambar 6 sampai dengan Gambar 7. Pada Gambar 5 dalam *class* *TestClass* dalam baris kode 5 sampai dengan baris kode 12 berfungsi sebagai autentikasi ke fitur administrasi yang mengacu pada file *ceklogin.php*.

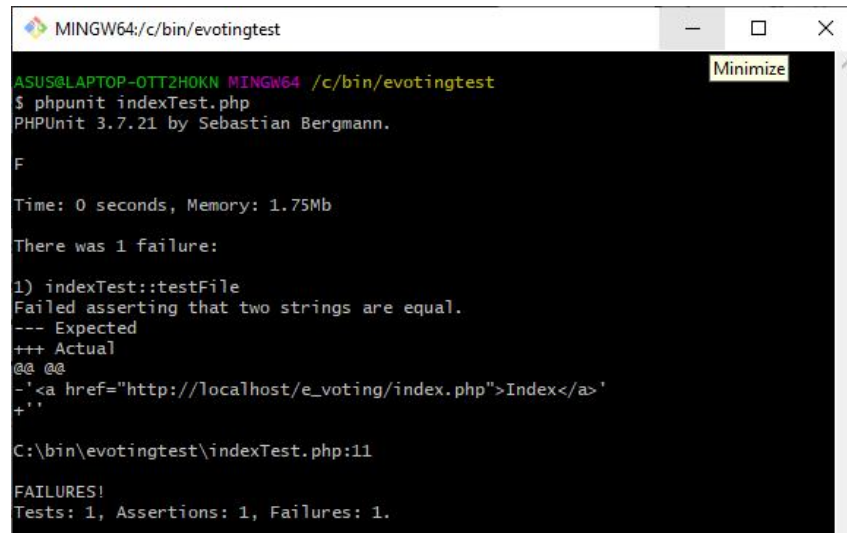
```

1. <?php
2. //Pengujian Autentifikasi
3. class indexTest extends PHPUnit_Framework_TestCase
4. {
5.     public function testFile()
6.     {
7.         ob_start();
8.         include 'ceklogin.php';
9.         $content = ob_get_contents();
10.        ob_end_clean();
11.        $this->assertEquals('<a
href="http://localhost/e_voting/index.php">Index</a>', $content);
12.    }
13. }

```

Gambar 5. Implementasi kode yang akan diuji pada autentifikasi

Pada Gambar 6 merupakan hasil pengujian yang dijalankan, hasil pengujian gagal karena *class* yang dihubungkan sebagai autentifikasi tidak ditemukan ketika dilakukan pengujian. Sedangkan Gambar 7 hasil pengujian yang berhasil dilakukan sebagai autentifikasi pada fitur administrasi. Berdasarkan akun yang digunakan dapat melakukan login untuk mengakses fitur yang disediakan. Kemudian Tabel 1 menunjukkan bahwa pengujian menggunakan metode *TDD* pada tiap unit telah berhasil dilakukan.



```

MINGW64/c/bin/evotingtest
ASUS@LAPTOP-OTT2H0KN MINGW64 /c/bin/evotingtest
$ phpunit indexTest.php
PHPUnit 3.7.21 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 1.75Mb

There was 1 failure:

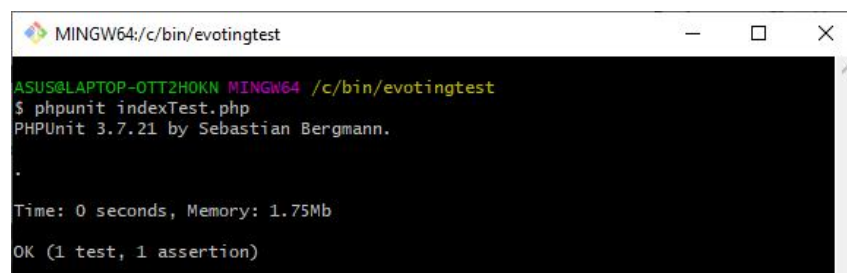
1) indexTest::testFile
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'<a href="http://localhost/e_voting/index.php">Index</a>'
+'

C:\bin\evotingtest\indexTest.php:11

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

Gambar 6. Hasil pengujian gagal pada fitur administrasi



```

MINGW64/c/bin/evotingtest
ASUS@LAPTOP-OTT2H0KN MINGW64 /c/bin/evotingtest
$ phpunit indexTest.php
PHPUnit 3.7.21 by Sebastian Bergmann.

.

Time: 0 seconds, Memory: 1.75Mb

OK (1 test, 1 assertion)

```

Gambar 7. Hasil pengujian berhasil pada fitur administrasi

Tabel 1. Hasil pengujian pada unit yang tersedia pada *API*

Pengujian Ke-	Unit	Hasil Pengujian
1	Login	Berhasil
2	Data Pemilih	Berhasil
3	Daftar Hadir	Berhasil
4	Daftar Antrian	Berhasil
5	Kandidat	Berhasil
6	Proses Pemilihan	Berhasil
7	Status Bilik	Berhasil
8	Pemilihan	Berhasil

3.4. Pengujian *System Usability Scale*

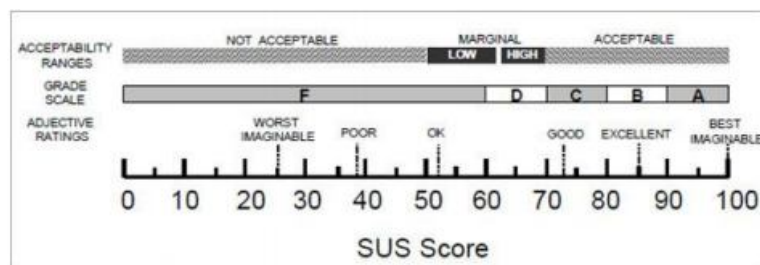
Setelah melakukan implementasi dan pengujian maka selanjutnya dilakukan tahap pengujian *System Usability Scale (SUS)* dengan maksud untuk mengetahui pengalaman pengguna dalam menggunakan aplikasi

Analisis *Application Programming Interface* pada *Mobile E-voting*
menggunakan Metode *Test-Driven Development* (Sunardi)

yang telah dikembangkan [17] [18] [19] [20]. Terdapat 10 pernyataan dalam *SUS* yang diajukan ke pengguna yang telah melakukan uji coba mengoperasikan aplikasi, diantaranya sebagai berikut:

1. Saya merasa akan menggunakan sistem ini kembali.
2. Saya merasa sistem ini tidak rumit untuk digunakan.
3. Saya berfikir sistem ini mudah digunakan.
4. Saya merasa akan membutuhkan bantuan orang lain terkait penggunaan teknis sistem ini.
5. Saya melihat beragam fungsi pada sistem ini terintegrasi dengan baik.
6. Saya melihat terdapat banyak ketidak konsistenan dalam sistem ini.
7. Saya berfikir bahwa orang lain akan cepat mempelajari penggunaan sistem ini.
8. Saya melihat bahwa sistem ini membingungkan.
9. Saya merasa bahwa tidak ada hambatan dalam menggunakan sistem ini.
10. Saya perlu mempelajari banyak hal dalam penggunaan sistem ini.

Berikut hasil dari pengambilan data yang diperoleh dari 36 responden yang sudah menyatakan pengalaman penggunaan sistem. Pernyataan dari Q1 - Q10 disajikan dalam rekapitulasi secara keseluruhan pada Tabel 2. Skala *SUS* dijelaskan pada Gambar 8 sebagai panduan skor *SUS* yang digunakan untuk kalibrasi skor. Didalam *SUS Score* terdapat 3 kategori yang berkaitan anatara lain *Acceptability Ranges*, *Grade Scale*, dan *Adjective Ratings*.



Gambar 8. Skor *SUS*

Tabel 2. Rekapitulasi Pernyataan Responden

Responden	Total Skor Pernyataan Responden										Total
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
1	3	3	4	2	3	2	4	1	3	3	70,0
2	3	4	4	4	3	1	3	1	3	1	67,5
3	3	4	2	2	4	3	3	3	3	2	72,5
4	3	3	3	1	4	1	3	1	4	2	62,5
5	2	3	3	3	4	3	2	2	3	2	67,5
6	3	4	4	3	3	2	3	2	3	2	72,5
7	3	4	4	3	4	2	3	2	2	2	72,5
8	4	4	4	4	4	1	1	4	1	4	77,5
9	2	4	3	2	4	2	2	2	3	2	65,0
10	2	3	3	3	3	2	2	3	2	4	67,5
11	3	3	4	3	3	2	2	2	3	4	72,5
12	3	3	4	2	3	2	3	2	3	2	67,5
13	3	3	3	1	3	2	3	1	3	3	62,5
14	3	3	3	3	4	2	3	2	2	3	70,0
15	3	3	3	2	3	2	3	3	2	3	67,5
16	3	3	3	3	3	3	3	3	3	3	75,0
17	2	4	3	1	3	4	2	2	3	1	62,5
18	3	4	4	3	3	2	3	2	3	3	75,0
19	2	3	2	2	2	2	3	2	2	4	60,0
20	4	4	4	2	3	1	3	1	3	3	70,0
21	3	3	3	4	4	1	2	2	3	3	70,0
22	3	4	4	2	3	1	3	1	4	2	67,5
23	4	3	3	4	3	2	3	2	2	3	72,5
24	2	3	3	2	2	2	2	2	3	3	60,0

25	4	3	3	4	4	3	4	2	3	4	85,0
26	4	3	3	2	4	1	3	1	3	2	65,0
27	4	3	4	2	3	2	3	1	4	2	70,0
28	3	4	3	1	3	1	4	1	3	3	65,0
29	3	3	3	2	3	2	3	2	3	2	65,0
30	3	4	4	1	4	1	1	1	4	4	67,5
31	3	4	4	1	3	1	3	1	4	2	65,0
32	3	4	4	2	3	3	4	2	4	3	80,0
33	3	4	4	2	4	2	2	2	3	2	70,0
34	3	4	4	3	4	2	4	1	3	3	77,5
35	4	4	0	1	3	1	2	2	2	3	55,0
36	3	4	4	3	3	3	2	1	2	4	72,5

Berkaitan dengan hasil data dari responden yang kemudian diolah dalam Tabel 2 maka dapat disimpulkan bahwa skor *SUS* dapat dijumlahkan berdasarkan dari skor total 2485,0 dengan jumlah reponden sebanyak 36 orang kemudian dibagi jumlah reponden itu sendiri dengan rata-rata skor 69,02.

4. KESIMPULAN

Berkaitan dengan hasil pembahasan, telah diuji *class* pada *API* untuk aplikasi *e-voting* pemilihan kepala daerah dengan metode *Test-Driven Development* untuk melakukan proses data pemilih, validasi data pemilih dan pemilihan. Hasil pengujian tersebut menunjukkan dapat dintegrasikan antar *class* menjadi sebuah *API* dengan aplikasi *mobile*. Berdasarkan hasil yang diperoleh dari pengujian yang dilandasi dari pengalaman pengguna dalam menggunakan sistem menggunakan pengujian *System Usability Scale* dapat disimpulkan skor yang diperoleh adalah 69,02 sehingga dapat diketahui bahwa sistem dapat diterima oleh pengguna dengan tingkat kategori *Adjective Ratings* yang 'Ok' dan juga dapat diketahui bahwa *Acceptability Ranges* dalam rentang *Marginal 'High'*.

DAFTAR PUSTAKA

- [1] P. A. Raharja, Sunardi and I. Riadi, "Perancangan E-voting Berbasis Mobile Menggunakan Metode Test-Driven Development," in *Seminar Nasional PPM UNESA 2018*, Surabaya, 2018.
- [2] A. Yudhana, R. Umar and F. Alameka, "Perancangan Sistem Informasi Menggunakan Enterprise Architecture Planning (Studi Kasus Pada Kecamatan di Kota Samarinda)," *Jurnal Ilmu Komputer dan Informatika*, vol. 4, pp. 114-123, 2018.
- [3] E. Priyono and F. N. Dihan, "E-voting: Urgensi Transparansi dan Akuntabilitas," in *Seminar Nasional Informatika 2010*, Yogyakarta, 2010.
- [4] Z. Musliyana, "Sistem Pendaftaran Hotspot Berbasis Web Pada Hotspot Mikrotik STIMIK U'budiyah Menggunakan Mikrotik Application Programming Interface (API), PHP dan MySQL," in *Prosiding SNIKOM 2014*, Banda Aceh, 2014.
- [5] A. Yudhana, Sunardi and A. Ikrom, "APLIKASI ANDROID UNTUK MONITORING KUALITAS LAHAN PERTANIAN," in *Prosiding SNST*, Semarang, 2018.
- [6] R. Hidayat and A. Ashari, "Penerapan Teknologi Web Service Untuk Integrasi Layanan Puskesmas dan Rumah Sakit," *Journal of Mathematics and Natural Sciences*, vol. 23, 2013.
- [7] J. Hendrawan, "Rancang Bangun Aplikasi Mobile Learning Untuk Tuntunan Shalat," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 1, pp. 44-59, 2018.
- [8] E. Prianto, A. Yudhana and A. Fadlil, "Analisis Perancangan Aplikasi Administrasi Kepegawaian dan Kegiatan Berbasis Android," in *Prosiding Seminar Nasional Multi Disiplin Ilmu & Call For Papers*, 2017.
- [9] F. Liu, K.-S. Liu, C. Chang and Y. Wang, "Research on the technology of iOS jailbreak," *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control*, pp. 644-647, 2016.
- [10] A. Yudhana, Sunardi and I. A. Mukaromah, "Implementation of Winnowing Algorithm with Dictionary

- English-Indonesia Technique to Detect Plagiarism," *International Journal of Advanced Computer Science and Applications*, vol. 9, pp. 183-189, 2018.
- [11] A. Yudhana, S. and A. J. S. Hartanta, "Perancangan Aplikasi Smartphone Android Untuk Penentuan Pola Satu Sisi Penggergajian Kayu Sengon," in *Prosiding SNST 2018*, 2018.
- [12] D. Contoneo, R. Pietrantiono and S. Russo, "Relai Testing: a technique to assess and improve software reliability," *IEEE Transactions on Software Engineering*, vol. 24, pp. 1-23, 2015.
- [13] A. Basir, A. Fadlil and I. Riadi, "Enterprise Architecture Planning Sistem Informasi Akademik Dengan TOGAF ADM," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 3, pp. 1-10, 2019.
- [14] J. H. Yuliansyah and I. Riadi, "Impemetasi Test Driven Development Pada Pengembangan Aplikasi Android Untuk Mahasiwa Universitas Ahmad Dahlan," *Jurnal Ilmu Teknik Elektro dan Informatika (JITEKI)*, vol. 4, pp. 43-50, Juni 2018.
- [15] A. N. A. Thohari and A. E. Amalia, "Jurnal Sistem Informasi dan Teknologi," *Implementasi Test Driven Development Dalam Pengembangan Aplikasi Berbasis Web*, vol. 1, pp. 1-10, 2018.
- [16] Dimas, "TDD (Test Driven Development)," 16 Maret 2018. [Online]. Available: <https://sis.binus.ac.id/2018/03/16/tdd-test-driven-development/>. [Accessed 7 Maret 2019].
- [17] Z. Sharfina and H. B. Santoso, "An Indonesian Adaptation of the System Usability," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2016.
- [18] J. Brooke, "SUS: A Retrospective," *Journal of Usability Studies*, vol. 8, pp. 29-40, 2013.
- [19] C. Katsanos, N. Tselios and M. Xenos, "Perceived Usability Evaluation of Learning Management Systems: A First Step," in *Panhellenic Conference on Informatics*, 2012.
- [20] Rosmiati, I. Riadi and Y. Prayudi, "A Maturity Level Framework for Measurement of Information Security Performance," *International Journal of Computer Applications*, vol. 141, pp. 1-6, 2016.