# Pembangunan *Frontend* Dan Penerapan Algoritma Enkripsi AES Pada Aplikasi Mobile Solidare Apps

Muhammad Iqbal Tawaqal[1]

[1] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : iqbal.tawakal@*outlook*.com[1]

**ABSTRAK**

Artikel tahunan 2013 UNICEF menyatakan bahwa sekitar 2,3 juta anak usia 7-15 tahun terdapat anak Indonesia putus sekolah. Yayasan Rumah Zakat adalah lembaga amal yang menjadi fasilitator penyambung donasi para donatur kepada anak-anak putus sekolah. Namun dalam penerapannya terdapat permasalahan yang terjadi yaitu dari sisi donatur perihal kesulitan melakukan rekomendasi anak yang putus sekolah, pemilihan anak yang akan didonasi, dan kesulitan mengetahui perkembangan anak yang telah terdonasi. Dari sisi masyarakat perihal kesulitan mendapatkan informasi tentang anak putus sekolah dan bagaimana cara membantunya, dan juga perihal kerahasiaan data dari lembaga amal/komunitas sosial. Solusi yang dapat diterapkan untuk permasalahan di atas adalah dengan membangun *frontend* aplikasi Solidare.

Metodologi penelitian yang digunakan dalam pembangunan *frontend* aplikasi Solidare yaitu metode penelitian terapan. Tahap pertama pembangunan *frontend* aplikasi Solidare yaitu menganalisis masalah, kemudian menganalisis arsitektur sistem, selanjutnya menganalisis algoritma enkripsi AES-128 yang digunakan untuk keamanan pada data, dan mengumpulkan kebutuhan sistem yang akan dijadikan bahan sebagai fungsionalitas aplikasi. Setelah itu melakukan perancangan sistem *frontend*.

*Frontend* aplikasi Solidare yang telah dirancang kemudian diimplementasikan dalam bahasa pemrogramman C# pada *platform windows phone*. Dan dilanjutkan pengujian sistem dengan menggunakan metode *black box* dan pengujian beta. Hasil dari pengujian menunjukan bahwa sistem *front-end* telah memudahkan donatur/masyarakat dalam mendapatkan informasi mengenai anak putus sekolah, melakukan donasi, melakukan rekomendasi anak putus sekolah/anak kurang mampu, mendapatkan laporan perkembangan anak yang telah terdonasi, dan menerapkan algoritma enkripsi AES-128 guna menjaga kerahasiaan data yang ada pada sistem *front-end* aplikasi.

**Kata Kunci*: class library, personal finance.***

## 1. PENDAHULUAN

Pendidikan dan kemiskinan merupakan dua hal yang berkaitan erat pengaruhnya di tengah-tengah kehidupan bermasyarakat. Hal ini dapat dilihat pada artikel tahunan 2013 dari UNICEF untuk Indonesia yang berjudul "Cerita Dari Indonesia" didapatkan fakta bahwa masih ada sekitar 2,3 juta anak usia 7-15 tahun atau 42% dari sebagian besar penduduk Indonesia terdapat anak putus sekolah [1]. Faktor ekonomi menjadi penyebab utama anak putus sekolah yakni mencapai 36%. Mengatasi hal tersebut, lembaga amal yang telah mempunyai legalitas secara hukum hadir dan menjawab permasalahan tersebut dengan membantu anak-anak putus sekolah. Yayasan Rumah Zakat Indonesia ikut memberi solusi terhadap anak-anak putus sekolah dengan memberikan bantuan berupa beasiswa pendidikan dan ada pula beasiswa untuk kebutuhan perlengkapan sekolah dan program beasiswa lainnya. Tujuannya yaitu sebagai fasilitator penyambung donasi para donatur kepada anak-anak yang putus sekolah.

Berdasarkan hasil observasi dengan bagian beasiswa pendidikan di kantor pusat Yayasan RZ didapatkan beberapa fakta diantaranya adalah para donatur sulit mendapatkan informasi kegiatan/*news* terkini dari komunitas sosial ataupun dari lembaga amal. Sulitnya donatur melakukan donasi pada anak asuh lainnya karena belum tersedianya penyajian data anak asuh yang lebih memberikan pilihan sesuai keinginan donatur. Selain itu, donatur kesulitan jika ingin merekomendasikan anak putus sekolah yang berada di sekitar lingkungannya dan masyarakat umum masih kesulitan mendapatkan informasi mengenai anak kurang mampu/anak putus sekolah dan bagaimana cara membantu mereka.

Selain beberapa fakta dan permasalahan tersebut, pada sebuah pembangunan aplikasi *mobile* baiknya dimulai dari mempertimbangkan keamanan data jenis apa yang digunakan. Pada penelitian ini, kemanan data pada aplikasi *mobile* Solidare menggunakan algoritma *Advanced Encryption System* (AES) [2].

Kemudian pemilihan perangkat *mobile* dalam penelitian ini adalah karena perangkat mobile diprediksi akan mendominasi pasar hingga 2,16 miliar ditahun 2016 dengan tingkat pertumbuhan 12,6% dari tahun 2015 [3].

### 1.1 Metode Pembangunan Perangkat Lunak

Metode pembangunan perangkat lunak yang digunakan pada pembangunan frontend Solidare yaitu waterfall model. Berikut proses model waterfall [4]:

1. *Requirement analysis and definition*

   Pengumpulan kebutuhan telah terdefinisi secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun.
2. *System and software design*

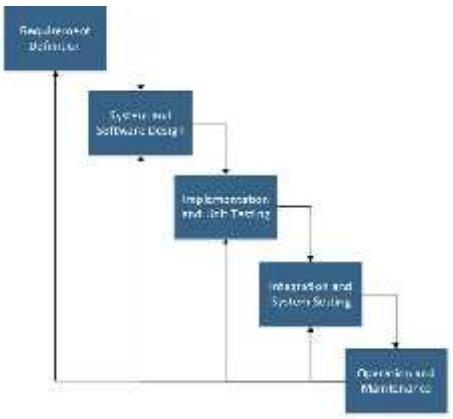
   Melakukan desain pada perangkat lunak. Dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
3. *Implementation and unit testing*

   Hasil desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrogramman yang telah ditentukan. Program yang dibangun langsung diuji berdasarkan unit-unitnya.
4. *Integration and system testing*

   Penyatuan unit-unit program kemudian sistem diuji secara keseluruhan.
5. *Operation and maintenance*

   Mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi yang sebenarnya.

Berikut tahapan-tahapan pengembangan perangkat lunak dengan model waterfall dapat dilihat pada Gambar I 1 berikut:



*Gambar I 1 Model Waterfall*

### 1.2 Citizen Journalism

Konsep dasar dalam *citizen journalism* yaitu memposisikan audiens sebagai produsen berita juga, bukan hanya konsumen pasif seperti selama ini berjalan dalam logika kerja jurnalisme tradisional berbasis media massa. Artinya posisi antara jurnalis sebagai pencari dan penulis berita, narasumber sebagai sumber berita, dan audiens sebagai konsumen berita sudah tidak berlaku. Antara produsen dan konsumen berita tidak bisa lagi diidentifikasi secara rinci karena setiap orang dapat memerankan keduanya. Intinya, dalam c*itizen journalism* yang diutamakan adalah peran aktif yang dilakukan oleh masyarakat dalam kegiatan pengumpulan, pelaporan, analisis serta penyampaian informasi dan berita [5].

### II.1 *Advanced Encryption Standard*

*Advanced Encryption Standard* adalah sebuah algoritma kriptografi simetris yang dapat digunakan untuk mengamankan data. Algoritma ini merupakan standar enkripsi dengan kunci-simetris. Algoritma AES dengan blok *ciphertext* simetris dapat mengenkripsi *(encipher)* dan dekripsi *(decipher)* pada sebuah informasi. Jenis Algoritma ini terbagi menjadi 3 yaitu AES-128, AES-192 dan AES-256. Masing-masing jenis algoritma AES tersebut dapat mengenkrip dan dekrip data pada blok 128 bit, blok 128 bit adalah ukuran tetap blok chiper yang digunakan pada algoritma AES. Blok cipher memiliki karakteristik berikut:

a. *Plaintext* dibagi menjadi blok-blok bit dengan panjang sama, misalnya 128 bit.
b. Panjang kunci enkripsi = panjang blok
c. Enkripsi dilakukan terhadap blok bit *plaintext* menggunakan bit-bit kunci.
d. Algoritma enkripsi menghasilkan blok *ciphertext* yang panjangnya = blok *plaintext*.

### 1.3 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON sendiri memiliki struktur data yang terdiri dari *json array* dan *json object*. *Json array* adalah kumpulan nilai yang terurutkan dan memiliki bentuk kurung kotak ( [ ] ), sedangkan *json object* adalah kumpulan pasangan *string* dan *value* dan memiliki bentuk kurung kurawal ( { } ) [9].

### 1.4 Pemrogramman Berorientasi Objek

Pemrograman berorientasi objek (PBO) merupakan kelanjutan dari proses analisis dan perancangan berorientasi objek. Pada pemrograman berorientasi objek, komponen yang didisain dalam proses desain diimplementasikan dengan menggunakan bahasa pemrograman berorientasi

objek. Berikut adalah beberapa konsep pemrograman berorientasi objek, yaitu:

1. Objek

Di dalam PBO, objek merupakan susunan/kerangka berupa data serta prosedur untuk mengoperasikan data tersebut.

2. Abstraksi

Abstraksi merupakan perwakilan data ataupun prosedur yang sama dengan menyembunyikan suatu objek jika tidak berhubungan dan dapat digunakan di kasus lain jika berhubungan.

3. Enkapsulasi

Enkapsulasi merupakan suatu proses yang tidak mengizinkan akses secara langsung ke data (data disembunyikan). Agar dapat mengakses data, terlebih dahulu terhubung dengan objek yang bertanggung jawab terhadap data tersebut.

4. *Polymorphism*

*Polymorphism* adalah kemampuan dari dua objek berbeda dalam merespon permintaan yang sama dengan caranya sendiri.

5. Pewarisan

Pewarisan *(inheritance)* digunakan untuk mengklasifikasikan objek di dalam program berdasarkan karakteristik umum.

6. Agregasi

Agregasi adalah kondisi ketika sebuah objek digabungkan dengan bagian dari objek lain dan kemudian bekerja sama.

### 1.5 *Unified Modelling Language* (UML)

*Unified Modelling Language* (UML) adalah bahasa standar pemodelan yang digunakan pada pembangunan perangkat lunak dengan pendekatan berorientasi objek. Dalam perancangan sistem, model memiliki peranan penting untuk mengelola kompleksitas. Pemodelan dapat menghubungkan setiap aspek penting dari perancangan sistem. Sebuah model adalah representasi dari benda nyata. Model yang dibangun adalah penyederhanaan dari sistem yang akan dibangun. Model memberikan maksud dan kelangsungan hidup sistem untuk dipahami serta dievaluasi. Dan memberikan pemahaman lebih cepat dibandingkan harus langsung ke sistem sebenarnya.

UML versi 2.0 memiliki 13 diagram yang dapat digunakan. Namun, dalam penelitian ini hanya 4 diagram UML yang akan digunakan. Tabel 0-1 menjelaskan mengenai diagram yang akan digunakan.

Tabel 0-1 Diagram-diagram UML 2.0 yang Digunakan

| Nama Diagram | Deskripsi |
|---|---|
| *Use Case Diagram* | Menggambarkan interaksi antara pengguna dengan sistem yang akan dibangun. Membantu dalam pemetaan kebutuhan dalam sistem. |
| *Activity Diagram* | Menggambarkan aktivitas secara sekuensial dan parallel dalam sistem. |
| *Class Diagram* | Menggambarkan hubungan *class*, *interface* dalam sistem. |
| *Sequence Diagram* | Menggambarkan interaksi antar objek dimana urutan setiap objek dan interaksinya sangat penting. |

## 2. ISI PENLITIAN

### 2.1 Analisis Masalah

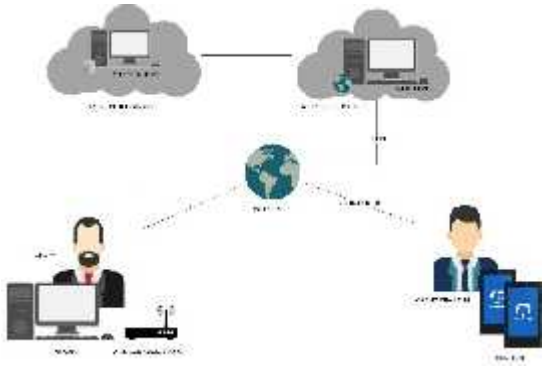
Berikut adalah beberapa permasalahan yang muncul pada penelitian ini yaitu:

1. Sulitnya donatur untuk merekomendasikan anak kurang mampu karena terbatasnya pengetahuan mengenai informasi/cara untuk merekomendasikannya ke komunitas sosial atau lembaga amal.
2. Sulitnya donatur melakukan donasi pada anak kurang mampu/anak putus sekolah lainnya karena belum tersedianya penyajian data anak kurang mampu/anak putus sekolah yang lebih memberikan pilihan sesuai dengan keinginan donatur.
3. Sulitnya donatur untuk mengetahui perkembangan anak karena belum adanya layanan yang memudahkan donatur mendapatkan laporan perkembangan anak yang terdonasi.
4. Masyarakat umum terkadang kesulitan mendapatkan informasi tentang anak kurang mampu/anak putus sekolah dan tidak tahu bagaimana cara menbantu mereka.

Data penting dari Yayasan dan data donatur yang bersifat internal dan rahasia, sehingga perlu untuk menerapkan keamanan pada data tersebut.

### 2.2 Analisis Arsitektur Sistem

Arsitektur sistem *frontend* Solidare dapat dilihat pada Gambar 0-1.



Gambar 0-1 Arsitektur Sistem *Frontend* Solidare

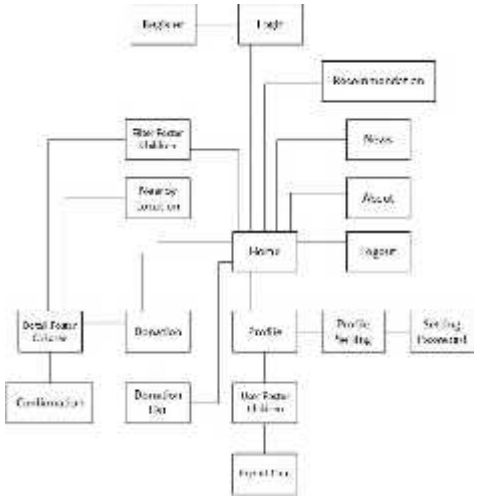
Gambaran arsitektur sistem secara umum terdiri dari dua buah subsistem. Pertama adalah web administrator dimana bagian ini administrator melakukan pengelolaan *website* yang telah di-*hosting* dan tersimpan dalam webserver dengan layanan *cloud computing* Microsoft Azure *Web Service*. Setelah tersimpan, di dalam web administrator ini disediakan pula *service* dalam bentuk API agar data lembaga amal dapat diakses oleh *platform* lain. Bentuk data hasil *encode* dari PHP-nya adalah JSON. Kedua adalah perangkat *mobile windows phone* dimana data berupa JSON tersebut akan di-*parsing* dengan JSON *Parser* agar data dari web administrator dapat diterima oleh para donatur.

### 2.3 Analisis Algoritma Enkripsi AES

Pada analisis ini akan dijelaskan bagaimana algoritma kriptografi diimplementasikan kedalam sistem *frontend* Solidare. Algoritma kriptografi yang digunakan dalam *frontend* Solidare adalah algoritma kriptografi *Advanced Encryption Standard* 128 Bit (AES-128). Di dalam algoritma AES-128 terdapat dua proses yaitu proses enkripsi dan proses dekripsi.

### 2.4 Perancangan Arsitektural Menu

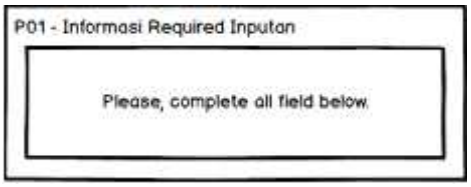
Berikut ini adalah perancangan arsitektural menu *frontend* Solidare:



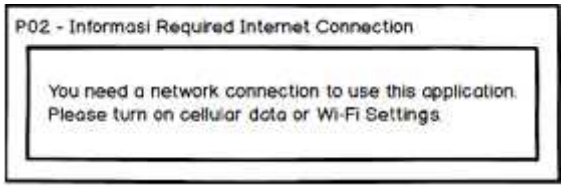
Gambar 0-2 Perancangan Arsitektural Menu

### 2.5 Perancangan *Pesan*

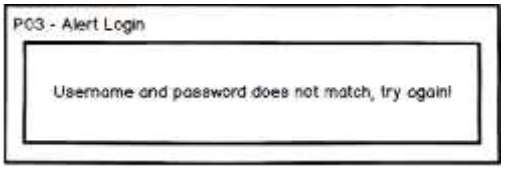
Perancangan pesan yang akan dibahas yaitu perancangan pesan yang ada pada *frontend* Solidare. Berikut beberapa perancangan pesan pada *frontend* Solidare.



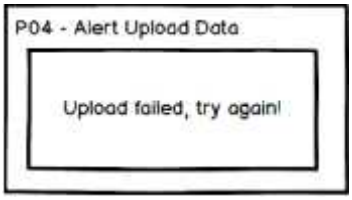

Gambar 0-3 *P01-Informasi Required Inputan*



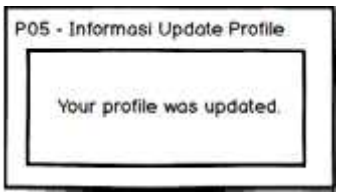

Gambar 0-4 P02-Informasi *Required Internet Connection*



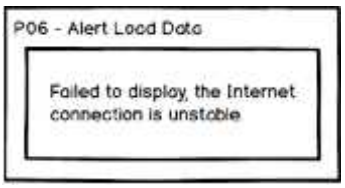

Gambar 0-5 *P03-Alert Login*



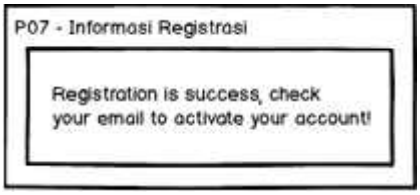

Gambar 0-6 *P04-Alert Upload Data*



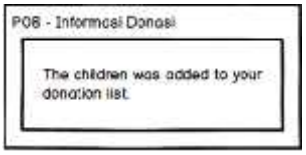

Gambar 0-7 *P05-Informasi Update Profile*



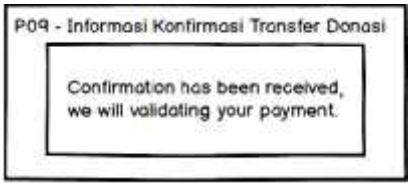

Gambar 0-8 *P06-Alert Load Data*




Gambar 0-9 *P07-Informasi Registrasi*



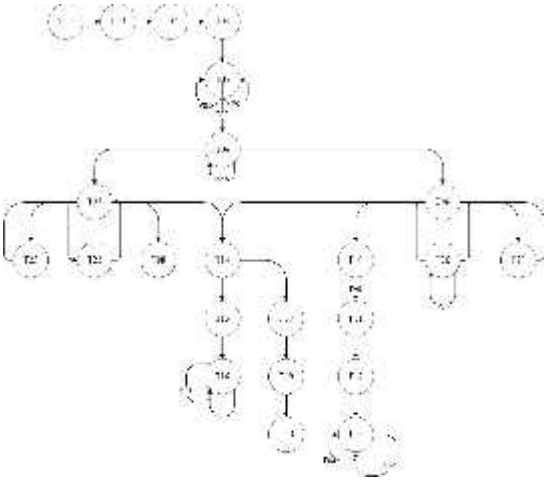

Gambar 0-10 *P08-Informasi Donasi*




Gambar 0-11 Informasi Konfirmasi Transfer Donasi

**2.6 Jaringan Semantik**

Berikut adalah jaringan semantik dari *frontend* Solidare.



Gambar 0-12 Jaringan Semantik Frontend Solidare

**2.7 Implementasi Sistem**

Implementasi sistem yang dilakukan pada penelitian ini adalah dengan mendeskripsikan lingkungan dimana sistem akan diimplementasikan/diinstal. Kemudian implementasi dari setiap perancangan antarmuka dimana didaftar antarmuka mana saja yang diimplementasikan kemudian jika ingin melihat *screenshot*-nya dapat melihat pada bagian lampiran

Lingkungan implementasi merupakan spesifikasi *hardware, software* dimana sistem ini akan dipasang dan digunakan. Berikut adalah lingkungan implementasinya:

1. Lingkungan *Hardware*

Berikut adalah spesifikasi hardware dalam pembangunan perangkat lunak ini.

Tabel 0-2 Lingkungan Implementasi *Hardware* Pembangun *Frontend* Aplikasi

| No | Item | Spesifikasi |
|---|---|---|
| 1 | Processor | Intel Core i7-4510U CPU @ 2.00 GHz 2.60 GHz |
| 2 | Harddisk | 1 TB |
| 3 | Memory | 4 GB |
| 4 | VGA | 4 GB |
| 5 | Monitor | 14 inch |
| 6 | Mouse | Optical Microsoft Sculpt Touch |
| 7 | Keyboard | QWERTY Creative |
| 8 | USB port | 3 buah |
| 9 | Koneksi Internet | Ada |
| 10 | Windows Phone 8.0 | Phablet Lumia 1520 6 inch dan Lumia 520 4.5 inch |

Berikut adalah spesifikasi hardware untuk mengakses sistem:

Tabel 0-3 Lingkungan *Hardware*

| No | Item | Spesifikasi |
|---|---|---|
| 1 | Processor | @ Dual-Core 1 GB |
| 2 | SD Card | 8 GB |
| 3 | Memory | 512 GB |
| 4 | Koneksi Internet | Ada |
| 5 | Windows Phone version | 8.0 |

2. Lingkungan *Software*

Perangkat lunak yang digunakan dalam pembangunan *frontend* Solidare ini adalah:

1. Sistem Operasi Windows 8 Single Language 64 bit.
2. Bahasa pemrogramman C#.
3. Code editor menggunakan Microsoft Visual Studio Community 2013 Update 4.
4. Aplikasi windows phone dibangun menggunakan framework .NET 4.
5. Pertukaran data yang dilakukan menggunakan JSON.

Berikut adalah spesifikasi software untuk mengakses sistem:

Tabel 0-4 Lingkungan *Software*

| No | Item | Spesifikasi |
|---|---|---|
| 1. | Windows Phone version | 8.0 (Lumia Denim) |

**IV.1.1 Implementasi Antarmuka**

Bagian ini berisi implementasi antarmuka dari *windows phone* dimana implementasi

antarmukanya diwakili dengan nama antarmuka beserta nama file yang mewakilinya.

Tabel 0-5 Implementasi Antarmuka

| No | Nama Antarmuka | Nama File |
|---|---|---|
| 1 | Walkthrough Aplikasi | PageStart.xaml |
| 2 | Login | PageLogin.xaml |
| 3 | Register | PageRegister.xaml |
| 4 | Timeline News | NewHomepage.xaml |
| 5 | Detail News | PageDetailNews.xaml |
| 6 | Timeline Donation | NewHomepage.xaml |
| 7 | Profile | PageProfileDonatur.xaml |
| 8 | User Sponsored Children | PageProfileDonatur.xaml |
| 9 | Report Card | PageRapotAnak.xaml |
| 10 | Detail Profile | PageDetailDonatur.xaml |
| 11 | Setting Profile | PageSetting.xaml |
| 12 | User Change Password | PagePassword.xaml |
| 13 | Children Recommendation | PageRekomendasi.xaml |
| 14 | Detail Children | PageDetailAnak.xaml<br>PageProfileAnak.xaml<br>DetailAnakAsuhNearby.xaml |
| 15 | Donation List | PageKeranjangDonasi.xaml |
| 16 | Donation Detail | DetailDonation.xaml |
| 17 | Confirmation | PageKonfirmasi.xaml |
| 18 | Filter Children | PageFilter.xaml |
| 19 | Nearby Location | PageNearby.xaml |
| 20 | About | PageAbout.xaml |
| 21 | Terms & Conditions | PageTerms.xaml |

Dari daftar tabel di atas, dapat disimpulkan bahwa seluruh antarmuka sudah diimplementasikan didalam *class*. Untuk antarmuka *detail*-nya dapat dilihat pada lampiran A.

## 2.8 Pengujian Sistem

Pengujian sistem *frontend* ini terdiri dari dua tahap yakni pengujian alpha dan pengujian beta. Metode yang digunakan dalam pengujian sistem *frontend* ini adalah dengan menggunakan metode pengujian *black box* dimana pengujian ini berfokus pada persyaratan fungsional perangkat lunak.

Pada bagian skenario pengujian ini terbagi menjadi dua jenis, yaitu skenario pengujian fungsional dan skenario pengujian *acceptance.*

1. Skenario pengujian fungsional

Skenario pengujian fungsional yang dilakukan adalah dengan menguji 1 sistem *frontend* yaitu *windows phone*. Skenario pengujian sistem *frontend windows phone* dapat dilihat pada Tabel 0-6.

Tabel 0-6 Skenario Pengujian Fungsional

| No | Kelas Uji | Poin Pengujian | Jenis Pengujian |
|---|---|---|---|
| 1 | *Login* | Isi data *login* | *Equivalence Partitioning* |
| | | Validasi *field login* | *Equivalence Partitioning* |
| 2 | *Register* | Isi data register | *Equivalence Partitioning* |
| | | Validasi *field register* | *Equivalence Partitioning* |
| 3 | *Forgot password* | Memasukan *email* dan nomor telepon | *Equivalence Partitioning* |
| | | Validasi *field email* dan *nomor telepon* | *Equivalence Partitioning* |
| 4 | *Setting profile* | Isi data *profile* | *Equivalence Partitioning* |
| 5 | *Change password* | Memasukan *current-password* dan *new-password* | *Equivalence Partitioning* |
| | | Validasi *field current-password* dan *new-password* | *Equivalence Partitioning* |
| 6 | *Donation* | Memililh anak yang akan didonasi | *Equivalence Partitioning* |
| 7 | *Confirmation* transfer donasi | Isi data *confirmation* | *Equivalence Partitioning* |
| | | Validasi *field confirmation* | *Equivalence Partitioning* |
| 9 | Rekomendasi anak | Isi data rekomendasi anak | *Equivalence Partitioning* |
| 11 | Filtering anak berdasarkan status anak | Memilih status *orphan, orphans,* dan *poor children*. | *Equivalence Partitioning* |
| 12 | Filtering anak berdasarkan jenjang pendidikan anak | Memilih jenjang pendidikan *elementary school, junior school,* dan *high school*. | *Equivalence Partitioning* |
| 13 | Filtering anak berdasarkan jenis kelamin anak | Memilih jenis kelamin anak laki-laki dan perempuan. | *Equivalence Partitioning* |

2. Skenario pengujian beta

Berikut adalah daftar pertanyaan kuesioner pengujian beta yang disebar kepada 30 masyarakat sabagai calon pengguna *frontend* aplikasi yang dibangun yang dapat dilihat pada Tabel 0-7.

Tabel 0-7 Daftar Pertanyaan Kuesioner

| NO | PERTANYAAN | SS | S | N | T | ST | Metode Hasil Pengujian |
|---|---|---|---|---|---|---|---|
| 1 | Apakah anda setuju jika daftar anak kurang mampu/anak putus sekolah ditampilkan pada aplikasi ini? | | | | | | Skala *Likert* |
| 2 | Apakah anda setuju aplikasi ini memberikan kemudahan untuk merekomendasikan anak kurang mampu/anak putus sekolah? | | | | | | |
| 3 | Apakah anda setuju aplikasi ini membantu dalam berdonasi? | | | | | | |

| NO | PERTANYAAN | SS | S | N | T | ST | Metode Hasil Pengujian |
|---|---|---|---|---|---|---|---|
| 4 | Apakah anda setuju aplikasi ini memberikan kemudahan dalam melakukan konfirmasi transfer donasi? | | | | | | |
| 5 | Apakah anda setuju aplikasi ini memberikan *feedback* laporan perkembangan anak setiap semesternya? | | | | | | |
| 6 | Apakah anda setuju tampilan aplikasi ini mudah dipahami? | | | | | | |
| 7 | Apakah anda setuju jika pada sebuah aplikasi *mobile* hanya memiliki satu warna dominan? | | | | | | |
| 8 | Apakah anda setuju jenis *font* pada aplikasi ini mudah untuk dibaca? | | | | | | |
| 9 | Apakah anda setuju ukuran *font* pada aplikasi ini sesuai untuk dibaca? | | | | | | |
| 10 | Apakah anda setuju dengan adanya keamanan data pada aplikasi ini? | | | | | | |

Hasil pengujian merupakan hasil dari skenario pengujian yang telah disusun sebelumnya. Hasil pengujian yang dilakukan terdiri dari hasil pengujian fungsionalitas dan pengujian beta.

## 3. KESIMPULAN

Berdasarkan hasil pengujian *frontend* aplikasi Solidare pada platform windows phone maka dapat diperoleh kesimpulan berikut:

1. *Frontend* aplikasi Solidare pada platform windows phone sebagian besar telah dapat mempermudah donatur dalam melakukan donasi kepada anak kurang mampu/anak putus sekolah.
2. *Frontend* aplikasi Solidare pada *platform windows phone* sebagian besar telah dapat mempermudah donatur dalam melakukan rekomendasi anak kurang mampu/anak putus sekolah yang berada di sekitar lingkungan pengguna.
3. *Frontend* aplikasi Solidare pada *platform windows phone* sebagian besar telah dapat mempermudah donatur dalam mendapatkan laporan perkembangan anak yang telah terdonasi oleh pengguna.
4. *Frontend* aplikasi Solidare pada *platform windows phone* sebagian besar telah dapat membantu masyarakat sebagai media informasi mengenai berita dari komunitas sosial/lembaga amal dan sebagai media berdonasi kepada anak kurang mampu/anak putus sekolah.
5. *Frontend* aplikasi Solidare pada *platform windows phone* telah menggunakan algoritma enkripsi AES-128 pada proses pertukaran data dengan *web server*. Sehingga data terenkrip dengan baik dan mengurangi kemungkinan terjadinya intimidasi data dari pihak luar yang tidak bertanggung jawab.

## 4. DAFTAR PUSATAKA

### Bibliography

[1] Laporan Tahunan Indonesia, UNICEF, 2013.

[2] "Advanced Encryption Standard," 25 November [Online]. Av https://id.wikipedia.org/wiki/Advanced_Encryption_St [Diakses 5 July 2015].

[3] "Mobile App Developers: Start with Security," Februar [Online]. Available: https://www.ftc.gov/tips-advice/bu center/guidance/mobile-app-developers-start-security. [Diakses 5 July 2015].

[4] Z. Habibi, ""Citizen Journalism": Ketika Berita Tidak Memiliki Satu Muka," *Jurnal Komunikasi*, vol. 1, no. 2 18, 2007.

[5] j. org, "json org," [Online]. Available: http://json.org/ [Diakses 5 July 2015].

[6] I. Sommerville, Software Engineering, USA: Pearson Education, Inc., 2011.

[7] Ayushi, "Symmetric Key Cryptographic Algorithm," *International Journal of Computer Applications,* vol. 1, no. 1, pp. 1-4, 2010.

[8] C. Ireland, B. David, M. Newton dan K. Waugh, "Understanding Object-Relational Mapping: A Framework Based Approach," *International Journal on Advances in Software,*, vol. 2 no. 2&3, pp. 202-216, 2009.

[9] P. Pramudya, Membuat Aplikasi untuk Windows Phone, Bandung: ANDI, 2011.

[10] J-Lab, "J-LAB Knight Community News Network," Foundation, [Online]. Available: http://kcnn.org/le modules/principles-of-citizen-journalism/. [Diakses Agustus 2015].

[11] D. Selent, "Advanced Encryption Standard," *InSight: ACADEMIC JOURNAL,* vol. 6, no. 2, pp. 1-14, 2010.