Implementasi Teknologi Cloud Messaging, Cloud Notification, Cloud Security Dan Kompresi Gambar Pada Aplikasi Solidare

Adam Mukharil Bachtiar[1], Handika[2]

[1] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
[2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : adammb@outlook.com[1], handika_@outlook.com[2]

ABSTRAK

Solidare merupakan sebuah aplikasi *mobile* yang memiliki beberapa fungsionalitas penting dalam penggunaannya. Beberapa fitur penting selain donasi juga mendukung penggunaan aplikasi *Solidare* dan kemudian didapatkan sebuah permasalahan penting bahwa terdapat kebutuhan dari pengguna yang belum direalisasikan oleh sistem sebelumnya. Salah satu permasalahan yang terjadi pada fitur aplikasi *Solidare* yang membutuhkan kemampuan untuk melakukan unggah file gambar ke sistem. Kendala terhadap besarnya ukuran file gambar yang akan dikirimkan menjadi salah satu tolak ukur bahwa dibutuhkan sebuah penerapan algoritma untuk melakukan kompresi terhadap file gambar tersebut. Permasalahan lain kemudian muncul pada kebutuhan-kebutuhan lainnya seperti dibutuhkannya sistem pemberitahuan kepada pengguna secara langsung serta sistem yang dapat melakukan komunikasi teks singkat menjadi masalah tambahan yang terdapat pada aplikasi *Solidare*. Melihat beberapa kebutuhan tersebut bahwa setiap teknologi yang akan diterapkan membutuhkan sebuah pengamanan terhadap hal-hal yang memiliki resiko tinggi seperti data serta keamanan lain maka diterapkan juga sebuah keamanan dari *Cloud Computing*.

Dengan penerapan beberapa teknologi seperti *cloud notification*, *cloud messaging*, *cloud security* serta algoritma untuk melakukan kompresi gambar pada aplikasi *Solidare* dapat memaksimalkan penggunaan aplikasi ini dengan lebih baik. Penerapan teknologi-teknologi tersebut berfokus pada aplikasi berbasis mobile Windows Phone yang saat ini mengalami pengembangan fitur.

Berdasarkan hasil pengujian yang dilakukan yakni pengujian fungsionalitas dengan menggunakan metode *black box* maka dapat disimpulkan bahwa implementasi teknologi cloud messaging, *cloud notification*, *cloud security* dan kompresi gambar pada aplikasi *solidare* mampu mencapai tujuan sebagai pemecahan masalah yang terjadi.

Kata Kunci: *Kata kunci: Solidare, Windows Phone, Teknologi Awan.*

1. PENDAHULUAN

Pada bagian ini akan dipaparkan mengenai latar belakang masalah, maksud dan tujuan.

1.1 Latar Belakang Masalah

Solidare merupakan sebuah aplikasi *mobile* yang memiliki beberapa fungsionalitas penting dalam penggunaannya. Salah satu fitur yang diberikan oleh aplikasi *solidare* adalah kemampuan bagi pengguna untuk melakukan donasi terhadap anak asuh yang membutuhkan donasi tersebut. Selain fitur donasi, aplikasi solidare juga memiliki banyak fitur yang menyesuaikan kebutuhan pengguna dalam memudahkan proses donasi seperti fitur konfirmasi pembayaran donasi, fitur konsultasi donatur serta beberapa fitur penting lainnya. Melihat kebutuhan yang sangat besar terhadap beberapa fitur yang terdapat pada aplikasi *solidare* maka dibutuhkan sebuah teknologi yang dapat lebih mempermudah segala aktifitas pengguna pada aplikasi *solidare*.

Pada beberapa fitur di aplikasi *Solidare* khususnya fitur yang memungkinkan pengguna untuk melakukan unggah file gambar ke sistem memiliki kendala dalam besarnya ukuran file gambar yang akan dikirimkan. Dibutuhkan sebuah teknologi yang dapat melakukan kompresi pada file gambar tersebut sehinga menutup kemungkinan kendala yang sama terjadi kembali.

Fitur konfirmasi adalah salah satu fitur yang sangat penting pada aplikasi *Solidare* karena pada fitur ini terdapat proses transaksi pembayaran yang merupakan proses yang sangat krusial bagi seorang donatur. Proses konfirmasi tersebut terdiri dari beberapa tahap mulai dari pengiriman data transaksi yang telah dilakukan donatur hingga kemudian data tersebut dilakukan validasi pembayaran oleh pihak administrasi sebuah lembaga amal. Informasi mengenai validasi pembayaran tersebut dapat langsung diterima oleh donatur maka tercipta kebutuhan baru guna memberitahukan

informasi secara langsung, cepat dan fleksibel kepada donatur. Dalam kasus ini dibutuhkan sebuah teknologi tambahan untuk menunjang kebutuhan akan pemberitahuan informasi kepada pengguna.

Tidak hanya fitur untuk melakukan konfirmasi, teknologi tambahan yang lebih mumpuni juga dibutuhkan pada fitur konsultasi yang terdapat pada aplikasi *Solidare*. Fitur tersebut memungkinkan donatur untuk dapat berkomunikasi secara langsung kepada pihak lembaga amal dengan memberikan kontak-kontak yang dibutuhkan oleh donatur. Kendala yang kemudian terjadi ketika seorang donatur ingin melakukan komunikasi teks secara *on-time* dengan pihak lembaga amal terkait. Pada versi sebelumnya aplikasi *Solidare* belum menyediakan fasilitas untuk pengguna melakukan komunikasi teks secara *online* sehingga donatur masih harus menghubungi pihak lembaga amal dengan menggunakan kontak lain yang dapat dihubungi.

Pada kedua masalah yang telah dipaparkan sebelumnya dibutuhkan sebuah teknologi yang dapat menangani beberapa aspek pada kedua fitur tersebut. *Cloud Computing* merupakan sebutan dari teknologi yang dapat mencapai hal-hal yang dibutuhkan pada teknologi yang akan diterapkan. *Cloud Computing* atau biasa disebut dengan komputasi awan sendiri merupakan gabungan pemanfaatan teknologi komputer dalam suatu jaringan dengan pengembangan berbasis internet yang berfungsi untuk menjalankan program melalui komputer-komputer yang terkoneksi pada waktu yang sama. Dibalik itu semua, *Cloud Computing* juga merupakan hal yang sangat rentan terjadinya *disrupt services*, pencurian informasi, kurangnya privasi dan rusaknya informasi. Melihat beberapa aspek yang dapat terjadi maka sangat dibutuhkan sebuah keamanan dari *Cloud Computing* yang nantinya digunakan dalam penerapan teknologi untuk fitur-fitur lainnya.

1.2 Maksud dan Tujuan

Maksud dari penelitian ini adalah untuk mengimplementasikan teknologi *cloud messaging*, *cloud notification*, *cloud security* dan kompresi gambar pada aplikasi *Solidare*. Tujuan dari penelitian ini adalah:

1. Membantu donatur agar mendapatkan informasi mengenai validasi dari konfirmasi pembayaran yang dilakukan dengan lebih fleksibel dan cepat.
2. Memberikan kemudahan kepada donatur untuk dapat terhubung secara langsung dengan pihak lembaga amal terkait.
3. Menjaga keamanan dan kerentanan dari setiap informasi yang dimiliki oleh aplikasi Solidare.
4. Mengurangi penggunaan ukuran data yang besar dalam proses pengiriman file gambar ke dalam sistem.

1.3 Cloud Computing

Cloud computing mengacu pada aplikasi dan layanan yang berjalan pada jaringan terdistribusi menggunakan sumber daya virtual dan diakses oleh protokol internet umum dan standar jaringan. Hal ini dibedakan dengan gagasan bahwa *resource* adalah virtual dan tak terbatas dan itu merupakan rincian dari *physical system* dimana *software* berjalan diabstraksikan dari pengguna.

Dalam upaya untuk lebih menggambarkan *cloud computing*, sejumlah jenis *cloud* telah didefinisikan. Terdapat dua kelas yang berbeda dari *cloud* diantaranya adalah yang didasarkan pada *deployment model*, dan berdasarkan *service model*. *Deployment model* memberitahu dimana *cloud* berada dan untuk tujuan apa. *Public*, *private*, *community*, dan *hybrid clouds* merupakan *deployment models*.

Service model menggambarkan jenis layanan yang penyedia layanan tawarkan. *Service model* yang paling dikenal diantaranya adalah *software as a service*, *platform as a service*, dan *infrastructure as a service* atau dikenal sebagai SPI model. *Service model* saling membangun satu sama lain dan menentukan apa yang *vendor* harus kelola, dan apa yang merupakan tanggung jawab klien.

Cloud computing merupakan pergeseran paradigma nyata dalam cara dimana sistem dikerahkan. Skala besar dari sistem *cloud computing* telah diaktifkan seiring popularitas internet, dan pertumbuhan beberapa perusahaan jasa besar. *Cloud computing* membuat impian akan *utility computing* menjadi lebih mungkin, jauh lebih *scalable*, dan sistem yang tersedia secara universal.

Tidak semua aplikasi diuntungkan dengan *deployment* di *cloud*. Masalah dengan *latency*, kontrol transaksi, dan dalam keamanan tertentu, serta kepatuhan terhadap peraturan patut menjadi perhatian khusus.[1]

1.4 Microsoft Azure

Microsoft Azure adalah platform *cloud computing Microsoft*, yang menyediakan berbagai macam layanan yang dapat digunakan tanpa membeli. *Azure* memungkinkan pesatnya pengembangan yan memberikan solusi, dan menyediakan sumber daya untuk menyelesaikan tugas-tugas yang mungkin tidak sesuai dalam lingkungan lokal. Komputasi, *storage*, jaringan, dan layanan aplikasi *Azure* berfokus pada membangun solusi besar tanpa perlu khawatir tentang bagaimana pemasangan *physical infrastructure*.

Microsoft menyediakan support untuk *public*, *private*, dan *hybrid cloud*. *Windows Azure Pack* adalah *add-on* gratis untuk *Microsoft System Center* yang memungkinkan kita untuk meng-*host* banyak layanan inti *Azure* di data *center* kita sendiri, dan memberikan pengalaman *self-service* portal ke pengguna. Kita dapat mengintegrasikan itu semua ke

dalam *hybrid cloud* melalui penggunaan *virtual private network*.[2]

1.5 Windows Phone

Windows phone diperkenalkan pada tanggal 15 Februari 2010 pada pameran MWC, Barcelona. *Microsoft* memperkenalkan sebuah sistem operasi yang dianggap sebagai *a revolutionary new platfrom*.

Windows phone tercipta hasil kerja sama antara CEO *Microsoft* Steve Ballmer dengan CEO *Nokia* Stephen Elop yang dimana sistem operasi ini diterapkan pada ponsel genggam dengan merk *Nokia*. Windows phone dihadirkan dengan antarmuka pengguna yang tampak lebih bersih dan segar. Filosofi desain dari sistem informasi ini sendiri dinamakan *metro style* sehingga menunjukan ciri yang jelas, informasi yang mudah diperoleh dan menggunakan simbol-simbol yang mudah dipahami.[3]

1.6 Algoritma Joint Photograpic Experts

Joint Photograpic Experts (JPEG, dibaca jay-peg) di rancang untuk kompresi beberapa *full-color* atau *gray-scale* dari suatu gambar yang asli, seperti pemandangan asli di dunia ini. JPEGs bekerja dengan baik pada continous tone images seperi *photographs* atau semua pekerjaan seni yang menginginkan yang nyata; tetapi tidak terlalu bagus pada ketajaman gambar dan seni pewarnaan seperti penulisan, kartun yang sederhana atau gambar yang mengunakan banyak garis. JPEG sudah mendukung untuk 24-bit color *depth* atau sama dengan 16,7 juta warna (224 = 16.777.216 warna). *Progressive* JPEGs (p-JPEGs) adalah tipe dari beberapa persen lebih kecil dibandingkan baseline JPEGs: tetapi keuntungan dari JPEG dan tipetipenya telihat pada langkah-langkahnya sama seperti *iinterlaced GIFs*.

JPEG adalah algoritma kompresi secara *lossy*. JPEG bekerja dengan merubah gambar spasial dan merepresentasikan ke dalam pemetaan frekuensi. *Discrete Cosine Transform* (DCT) dengan memisahkan antara informasi frekuensi yang rendah dan tinggi dari sebuah gambar. Informasi frekuensi yang tinggi akan diseleksi untuk dihilangkan yang terikat pada penganturan kualitas yang digunakan. Kompresi dengan tingkatan yang lebih baik, tingkatan yang lebih baik dari informasi yang dihilangkan. Waktu Kompresi dan dekompresi dilaksanakan dengan simetris.[4]

1.7 Cloud Computing Security

API (Application Programming Interface) adalah kumpulan fungsi-fungsi atau perintah-perintah untuk menggantikan bahasa yang digunakan dalam system call dengan bahasa yang lebih terstruktur dan mudah dimengerti.

Keuntungan memprogram dengan menggunakan API adalah:

1. Portabilitas. Programmer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah terinstall API tersebut. Sedangkan system call berbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda.
2. Lebih Mudah Dimengerti. API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan.

System call interface ini berfungsi sebagai penghubung antara API dan system call yang dimengerti oleh sistem operasi. System call interface ini akan menerjemahkan perintah dalam API dan kemudian akan memanggil system calls yang diperlukan.

Untuk membuka suatu file tersebut user menggunakan program yang telah dibuat dengan menggunakan bantuan API, maka perintah dari user tersebut diterjemahkan dulu oleh program menjadi perintah open(). Perintah open() ini merupakan perintah dari API dan bukan perintah yang langsung dimengerti oleh kernel sistem operasi. Oleh karena itu, agar keinginan user dapat dimengerti oleh sistem operasi, maka perintah open() tadi diterjemahkan ke dalam bentuk system call oleh system call interface. Implementasi perintah open() tadi bisa bermacam-macam tergantung dari sistem operasi yang kita gunakan.[2]

2. ISI PENELITIAN

2.1 Analisis dan Perancangan Sistem

Berikut adalah analisis perancangan dari sistem yang dibangun.

2.1.1 Analisis Masalah

Analisis masalah merupakan penjabaran masalah yang ada sebelum diimplementasikannya teknologi yang diteliti pada perangkat lunak *Solidare* ini. Analisis masalah meliputi hal-hal sebagai berikut:

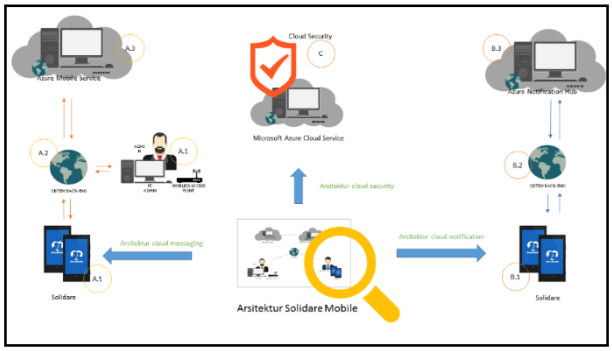
a. Donatur sebagai pengguna mengalami kesulitan ketika tidak mendapatkan informasi mengenai validasi yang telah dilakukan oleh pihak lembaga amal terhadap pembayaran donasi yang telah dilakukan.
b. Donatur mengalami kesulitan untuk terhubung dengan pihak lembaga amal secara cepat dan langsung.
c. Keamanan sistem aplikasi *Solidare* masih rentan terhadap ancaman-ancaman yang dapat terjadi pada *cloud computing*.

Sistem kesulitan untuk melakukan pengiriman data berupa file gambar dikarenakan besarnya ukuran file gambar tersebut.

2.1.2 Analisis Arsitektur Sistem

Pada analisis arsitektur sistem diberikan gambaran umum mengenai sistem yang dibangun

serta teknologi yang diimplementasikan. Gambaran yang diberikan terdiri dari proses pertukaran data sampai dengan bagaimana pengguna, *backend*, dan teknologi *cloud* saling berkomunikasi. Gambaran mengenai arsitektur pengimplementasian teknologi pada sistem dari perangkat lunak *Solidare* dapat dilihat pada gambar berikut.



Berikut adalah penjelasan dari arsitektur sistem perangkat lunak Solidare serta implementasi teknologi yang akan dilakukan:

1. Arsitektur *cloud messaging*

Arsitektur berikut menjelaskan alur pengiriman data serta proses yang terjadi pada cloud messaging yang diimplementasikan.

a. Pada poin A.1 menjelaskan bahwa sistem baik *mobile* maupun *website* mengirimkan sebuah identitas serta *channel* menuju ke *back-end* sistem untuk dikenali.
b. Poin A.2 memperlihatkan Hasil pengenalan dari sistem *back-end* kemudian dilanjutkan dengan melakukan validasi apakah identitas serta *channel* yang dikirimkan oleh kedua perangkat tersebut valid atau tidak. Jika hasil validasi mengatakan tidak maka dilakukan tahap selanjutnya yaitu A.3 sedangkan jika valid maka pesan langsung dikiriminkan menuju *channel* yang dituju.
c. Poin A.3 menjelaskan bahwa *Azure Mobile Service* melakukan pembangkitan identitas baru dan mengirimkan kembali kepada sistem *back-end* untuk dilakukan validasi kembali.

2. Arsitektur *cloud notification*

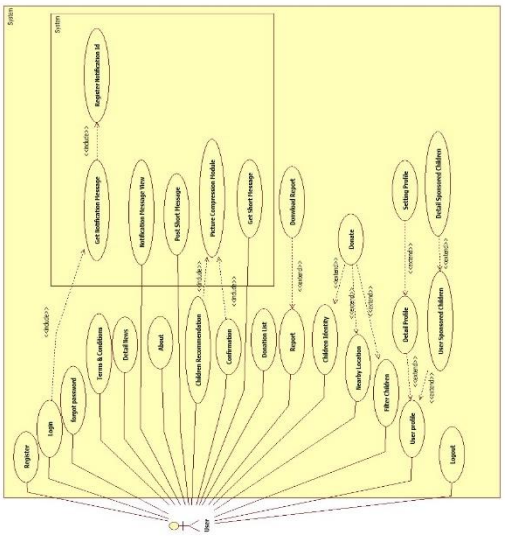
Arsitektur ini menjelaskan proses pengiriman data yang terjadi ketika sebuah notifikasi dikirimkan via *back-end* sistem.

a. Poin B.1 memperlihatkan bahwa sistem *mobile* mengirimkan sebuah indentitas serta *channel* yang terdaftar pada sebuah *back-end*.
b. Pada poin selanjutnya yaitu B.2 menjelaskan pengenalan terhadap identitas yang dikirimkan oleh sistem *mobile* kemudian divalidasi oleh sistem *back-end* untuk melakukan pencocokkan identitas yang telah dikirimkan.

Poin B.3 merupakan hasil pencocokan yang berasal dari *back-end* kemudian diterima oleh *Azure Notification Hub.*

2.1.3 Use Case Diagram

Berikut merupakan *use case diagram* pada pengembangan aplikasi Solidare.



2.2 Implementasi dan Pengujian Sistem

Berikut akan dipaparkan mengenai implementasi dan pengujian sistem.

2.2.1 Lingkungan Implementasi

Lingkungan implementasi mendeskripsikan mengenai implementasi sistem pada lingkungan perangkat lunak dan lingkungan perangkat keras. Berikut merupakan penjelasan penerapan yang dilakukan pada kedua lingkungan tersebut.

1. Lingkungan Perangkat Lunak

Spesifikasi perangkat lunak dalam pengembangan sistem adalah sebagai berikut:

a. Sistem Operasi Windows 8.
b. Bahasa pemrogramman C#.
c. Microsoft Visual Studio Community 2013 Update 3.

Penerapan sistem dilakukan dengan spesifikasi bahwa perangkat *mobile* menggunakan sistem operasi *Windows Phone* dengan versi 8.0.

2. Lingkungan Perangkat Keras

Spesifikasi perangkat keras dalam pengembangan sistem adalah sebagai berikut:

a. Processor Intel Core i7-4510U CPU @ 2.00 GHz 2.60 GHz
b. Harddisk 1 TB
c. Memory 4 GB
d. VGA 4 GB
e. Koneksi Internet tersedia
f. Windows Phone 8.0 Phablet Lumia 1520 6 inch

Dalam penerapannya sistem ini dapat dijalankan pada perangkat keras dengan spesifikasi sebagai berikut:

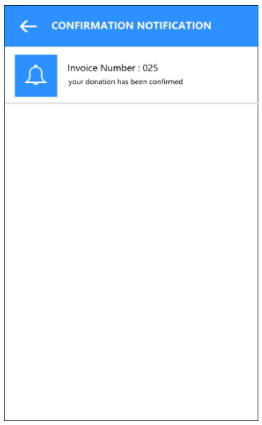
a. Mobile Windows Phone
b. Processor @ Dual-Core 1 GB
c. SD Card 8 GB
d. Memory 512 GB
e. Koneksi internet tersedia
f. Windows Phone version 8.0

2.2.2 Implementasi Antarmuka

Subbab implementasi antarmuka mendeskripsikan tentang implementasi antarmuka berdasarkan hasil perancangan yang dilakukan. Penjelasan mengenai pengimplementasian antarmuka dilakukan dengan mewakilkan setiap antarmuka berdasarkan nama, *class* atau *file* yang dapat mewakili. Berikut adalah implementasi antarmuka pada perangkat *mobile* terlihat pada tabel berikut.

No	Nama Antarmuka	Nama File
	Daftar Notifikasi Konfirmasi	ListConfirmationPage.xaml
	Halaman Pesan Singkat	LiveChat.xaml
	Bar Pesan Notifikasi	HomePage.xaml

Berikut implementasi dari sistem.







2.2.3 Pengujian

Proses pengujian sistem bertujuan untuk dapat mengidentifikasi kelengkapan serta mutu dari sistem yang dibangun maupun yang dikembangkan. Pengujian sistem juga dilakukan agar dapat diketahui apakah sebuah sistem telah dibangun sesuai dengan tujuan yang ingin dicapai.

Nama Proses	Point Pengujian	Jenis Pengujian	Detail Jenis Pengujian
Aktifasi Pesan Notifikasi	*Register Notification Id*	Black Box	Endurance Testing
	Get Notification Message	Black Box	Endurance Testing
Sistem *Cloud Messaging*	*Post Short Message*	Black Box	Endurance Testing
	Get Short Message	Black Box	Endurance Testing
Modul Kompresi Citra	*Picture Compression Module*	Black Box	Endurance Testing
Lihat Notifikasi Konfirmasi	*Notification Message View*	Black Box	Endurance Testing

Berdasarkan hasil pengujian yang telah dilakukan dan dipaparkan pada sub bab sebelumnya maka dilakukan evaluasi pengujian untuk memberikan kesimpulan terhadap hasil uji. Evaluasi pengujian yang dilakukan berdasar kepada pengujian fungsionalitas yang telah dilakukan dan dapat disimpulkan bahwa sistem yang dikembangkan telah sesuai dengan tujuan penelitian. Hal ini dapat disimpulkan karena terlihat dari hasil pengujian yang dilakukan dari segi validasi sistem dan penanganan kesalahan.

5. KESIMPULAN

Implementasi dari teknologi seperti *cloud messaging, cloud notification, cloud security* serta kompresi gambar yang dilakukan pada penelitian ini perlu dilakukan pengembangan-pengembangan ke

arah yang lebih baik sehingga dapat memenuhi kebutuhan pengguna di masa depan. Penggunaan algoritma kompresi gambar dapat lebih di teliti lagi mengenai algoritma yang sangat tepat untuk digunakan dalam mengurangi beban kapasistas pengiriman data. Selain itu dilakukan pengembangan pada fitur *messaging* serta fitur notifikasi.

6. DAFTAR PUSTAKA

[1] B. Sosinsky, "Cloud Computing Bible," Indianapolis, Wiley Publishing, 2011.

[2] M. Collier dan R. Shahan, "Fundamental Of Azure," Redmond, Microsoft Press, 2015.

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

[4] Y. Rafsyam, "Kompresi Citra menggunakan Teknik Lossy dengan Metode Algoritma," vol. 2, no. 1, 2010.