

Analisis dan Perancangan Sistem Manajemen *Event* Berbasis *Mobile Push Notification*

Ardiansyah

Abstract - Konvergensi teknologi Internet dan piranti mobile/smartphone saat ini bisa dimanfaatkan secara optimal sebagai salah satu media promosi dan pengelolaan sebuah acara. Dengan keberadaan teknologi *push service* akan memungkinkan setiap informasi acara baru dapat diterima oleh calon peserta yang prospektif. Selain itu pula dengan mengusung teknologi *cloud computing*, setiap penyelenggara acara cukup menggunakan *service* yang disediakan khusus di server Internet untuk mengelola acara yang akan diselenggarakan tanpa harus menyediakan infrastruktur sendiri. Pada penelitian ini telah berhasil dibuat rancangan sistem manajemen acara (*event*) yang diselenggarakan oleh *event organizer* atau sering disebut *EO*. Rancangan yang dibuat meliputi struktur tabel, menu hingga antarmuka aplikasi meliputi antarmuka berbasis web untuk administrator *EO* dan antarmuka berbasis mobile iPhone untuk user.

Keywords: *event, push notification, mobile, antarmuka.*

1. PENDAHULUAN

Salah satu indikator kesuksesan penyelenggaraan sebuah *event* atau acara dapat dilihat dari banyaknya peserta yang mengikuti. Ajang pameran akan disebut sukses besar dan meriah bila pengunjungnya sampai penuh setiap hari. Konser musik tentu akan sangat sukses bila penggemarnya penuh sesak memenuhi arena konser. Begitu pula penyelenggaraan seminar atau konferensi akan sangat memuaskan penyelenggara (*event organizer*) bila pesertanya penuh dan antusias mengikuti acara hingga akhir acara.

Akan tetapi tidak mudah untuk bisa mengumpulkan sedemikian banyak peserta dalam setiap *event*. Penyelenggara akan menghadapi kendala utama yaitu bagaimana menerapkan teknik promosi acara yang tepat sehingga dapat menjangkau sasaran yang tepat pula.

Dengan begitu pengeluaran untuk promosi atau iklan tidak terbuang sia-sia. Terdapat berbagai macam bentuk publikasi dan promosi acara, dari yang berbiaya murah, menengah hingga mahal.

Namun tidak selamanya bisa efektif menarik minat calon peserta untuk mengikuti acara. Di sisi lain, bagi pihak penyelenggara dengan modal yang minim, tentu tidak bisa leluasa menggunakan seluruh media promosi tersebut.

Keberadaan konvergensi teknologi Internet dan piranti *mobile/smartphone* saat ini sebenarnya bisa dimanfaatkan secara optimal sebagai salah satu media promosi acara dan pengelolaan acara. Sebagai media/alat promosi, keberadaan teknologi *push service* akan memungkinkan setiap informasi acara baru dapat diterima oleh calon peserta yang prospektif. Selain itu pula dengan mengusung teknologi *cloud computing* setiap penyelenggara acara cukup menggunakan *service* yang disediakan khusus di Internet untuk mengelola acara yang akan diselenggarakan tanpa harus menyediakan infrastruktur sendiri.

2. RUMUSAN MASALAH

Dari latar belakang di atas dapat dikemukakan beberapa rumusan masalah adalah sebagai berikut:

1. Media publikasi dan promosi yang ada saat ini tidak selalu efektif menjangkau calon peserta acara, selain pula ada beberapa dari media tersebut yang berbiaya mahal.
2. Penyelenggara acara selalu dibebankan untuk menyediakan infrastruktur pengelolaan acara sendiri, baik dari sisi *hardware* maupun *software*. Tentu saja hal ini akan menambah beban secara finansial.

3. RUANG LINGKUP

Ruang lingkup penelitian ini adalah:

1. Dari sisi *stakeholder* :
 - Calon peserta acara
 - Penyelenggara acara (*event organizer*)
2. Dari sisi sistem:
 - Internet
 - Mobile

Ardiansyah, Mobile Technology Innovation Center (MoTIC) Program Studi Teknik Informatika, Fakultas Teknologi Industri Universitas Ahmad Dahlan, Jl. Prof. Dr. Soepomo, Janturan Yogyakarta 55164 Telp. 0274-563515 Fax. 0274-564604 email: ardi@uad.ac.id, ardian2007@gmail.com

4. BATASAN PENELITIAN

Penelitian ini dibatasi hanya pada perancangan aplikasi sistem manajemen *event* disertai rancangan antarmuka (*interface*) yang diharapkan pada penelitian selanjutnya bisa dikembangkan dalam bentuk aplikasi jadi. Penelitian ini tidak sampai pada tahap implementasi pengembangan aplikasi.

5. TUJUAN DAN MANFAAT PENELITIAN

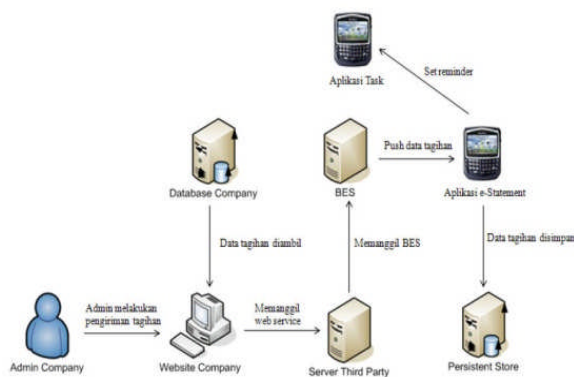
Terwujudnya rancangan sistem aplikasi iEvent yang dapat digunakan oleh para pengguna mulai dari penyelenggara acara hingga calon peserta dengan memanfaatkan teknologi Internet dan piranti *mobile*.

Aplikasi iEvent memiliki prospek yang cukup besar untuk menjadi salah satu produk atau komoditi bisnis konten Internet dan *mobile*. Sehingga bila nanti sudah dikembangkan dalam bentuk aplikasi jadi, potensi penggunaannya akan sangat besar. Sehingga nilai bisnis yang ada di balik produk ini akan sangat menjanjikan.

6. KAJIAN TEORI

6.1 Penelitian Terdahulu

Penelitian sebelumnya [5] berhasil mengembangkan aplikasi e-Statement berbasis *push mail* pada piranti *smartphone* BlackBerry.



Gambar 1 Proses Pengiriman Tagihan [5]

Gambar 1 merupakan proses pengiriman tagihan. Setiap data tagihan disimpan pada *database* perusahaan yang akan dikirim per bulan sesuai dengan periode tagihan. Data tagihan akan dikirimkan berdasarkan PIN (*Personal Identification Number*) BlackBerry yang unik untuk setiap perangkatnya. Pihak admin perusahaan mengirimkan data tagihan, tetapi tidak langsung ke perangkat BlackBerry, melainkan melalui BES (*BlackBerry Enterprise Server*). BES yang berupa BlackBerry Push Data Server disediakan oleh pihak *server third party*

yang memberikan akses *web service* kepada pihak admin perusahaan untuk melakukan *push*. BES akan mengirimkan data tagihan tersebut menggunakan BlackBerry Push-API sehingga aplikasi pada perangkat *smartphone* BlackBerry tidak perlu melakukan *poll server* untuk mendapatkan data terbaru. Teknologi BlackBerry Push-API ini akan memberikan data tanpa keterlibatan pengguna, sehingga ketika data baru diterima oleh perangkat, maka data tersebut akan segera disinkronisasikan dan tersedia pada *smartphone* BlackBerry pengguna saat mereka membuka aplikasi. Penyimpanan data tagihan pada perangkat *smartphone* BlackBerry menggunakan *Persistent Store*. *Persistent Store* pada *smartphone* BlackBerry dirancang untuk menyediakan tempat penyimpanan yang fleksibel. Dengan menggunakan *Persistent Store*, BlackBerry dapat menyimpan objek Java ke memori tanpa harus *serialize* data tersebut, dan begitu aplikasi dibuka, data-data yang disimpan pada *Persistent Store* tersebut dapat diambil dan ditampilkan kembali. Aplikasi juga memanggil aplikasi *task* untuk membuat *reminder* pembayaran tagihan [5].

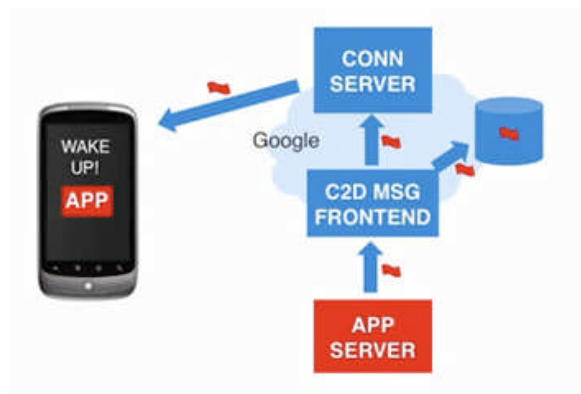
6.2 Landasan Teori

Beberapa penyedia layanan infrastruktur *push service*

6.2.1 Android Cloud to Device Messaging (AC2DM)

Android Cloud to Device Messaging (AC2DM) merupakan sebuah layanan (*service*) yang membantu para pengembang *software* untuk dapat mengirimkan data dari *server* ke aplikasi yang sudah terpasang di piranti *mobile* berbasis Android. Layanan ini menyediakan sebuah mekanisme yang sederhana, ringan dan cepat untuk membantu *server* agar dapat digunakan untuk mengirimkan pesan atau data ke aplikasi yang telah terpasang di piranti *mobile* secara langsung, sehingga *user* dapat menerima informasi terbaru dengan cepat saat itu juga [2].

Layanan AC2DM menangani seluruh aspek mulai dari pengiriman pesan dari *server* aplikasi hingga masuk ke *frontend* C2D (*cloud to device*), *connection server*, hingga pengiriman notifikasi ke aplikasi sasaran di piranti *mobile* Android. Gambaran layanan AC2DM ini diilustrasikan seperti pada gambar 2.

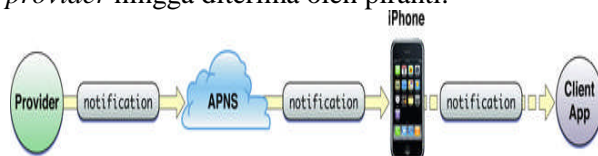


Gambar 2 Layanan AC2DM [4]

6.2.2 Apple Push Notification Service

Apple Push Notification Service (disingkat APNs) merupakan pusat dari fitur *push notification* yang dikembangkan Apple. APNs merupakan salah satu fitur layanan yang handal dan sangat efisien dalam hal penyebaran informasi ke berbagai macam piranti *mobile* seperti iPhone, iPad dan iPod *touch*. Setiap piranti diberikan sambungan IP (Internet Protocol) yang absah dan terenkripsi dengan layanan serta dapat menerima notifikasi melalui sambungan tersebut. Apabila ada notifikasi baru untuk sebuah aplikasi, sedangkan aplikasi tersebut dalam keadaan tidak aktif, maka piranti *mobile* akan memberikan *alert* kepada *user* bahwa ada notifikasi data baru untuk aplikasi tersebut [1].

Pengembang *software* (*provider*) menginisiasi pengiriman dari *server*. *Provider* tersambung dengan APNs melalui kanal yang *persistent* dan aman sambil memantau data yang masuk ke aplikasi *client*. Ketika ada data baru yang masuk, *provider* menyiapkan dan mengirimkan notifikasi melalui kanal tersebut ke APNs, yang selanjutnya mem-*push* notifikasi ke piranti tujuan. Gambar 3 memberikan gambaran sederhana mengenai pengiriman notifikasi dari *provider* hingga diterima oleh piranti.



Gambar 3 Push notification dari provider ke aplikasi client [1]

6.2.3 Event Organizer

Event Organizer atau sering disebut EO tidak jauh beda pengertiannya dengan sebuah kepanitiaan. Mulai dari level 'Perpisahan Sekolah' sampai 'Pindah Jabatan', EO pasti

dibutuhkan agar acara tersebut bisa berjalan sukses. Organizer mempunyai ruang lingkup kerja yang luas, sesuai jenis *event* yang ada dan perkembangannya. Kebanyakan masyarakat masih menganggap bahwa EO hanya untuk pentas musik saja. Padahal EO adalah sekelompok orang, yang terdiri dari tim pelaksana, tim pekerja, tim produksi dan tim manajemen yang melaksanakan tugas operasional suatu program acara atau melakukan pengorganisasian untuk mewujudkan suatu program acara. Macam-macam jenis acara antara lain dapat dikelompokkan sebagai berikut [8]:

1. Olahraga
 - a. Pertandingan Profesional
 - b. Kompetisi Peringkat
 - c. Pertandingan Persahabatan/Eksebisi
 - d. Lomba-lomba
2. Seni
 - a. Pementasan/Pagelaran Profit Oriented
 - b. Pementasan/Pagelaran Program Acara
 - c. Non Profit
 - d. Institusional/Privat
 - e. Lomba/Festival
 - f. Pentas Eksebisi/Apresiasi
2. Topik Wicara
 - a. Diskusi: seminar, sarasehan, *talk show*, dialog, *variety show*, presentasi.
3. Pameran: komoditas perdagangan dan pameran seni.
4. Pribadi: pesta pernikahan, pesta ulang tahun, syukuran, jabatan baru, pisah sambut, pesta peringatan pribadi.

6.2.4 Target Peserta

Target Market peserta acara adalah sekelompok orang yang memiliki kesamaan profil demografis dan psikografis. Gabungan kedua profil tersebut disebut audiens.

1. Demografis adalah data profil mengenai seseorang atau peta pasar, seperti umur, jenis kelamin, status perkawinan, punya anak atau tidak, di mana mereka tinggal, status pekerjaan dan jumlah penghasilan, juga informasi yang lebih spesifik seperti apakah memiliki komputer sendiri atau tidak, usia kendaraan, seberapa sering melakukan perjalanan bisnis dan sebagainya.
2. Psikografis adalah data profil mengenai pasar yang berhubungan dengan mengara orang melakukan sesuatu, apa yang memotivasi mereka lebih memilih suatu produk dibanding produk lainnya, serta berhubungan dengan pertanyaan-

pertanyaan seputar gaya hidup, dan sebagainya [3].

7. METODOLOGI PERANCANGAN SISTEM

7.1 Tahap Perancangan Sistem

1. Spesifikasi analisis kebutuhan sistem
2. Perancangan Sistem
 - i) Perancangan Diagram Konteks
 - ii) Perancangan DFD (Data Flow Diagram)

7.2 Analisis dan pengolahan data

Tahapan ini dilakukan untuk mengolah hasil data-data *event organizer* yang telah diperoleh pada tahap pengumpulan data sebelumnya.

7.3 Analisis kebutuhan sistem.

Hasil dari pengolahan data akan digunakan untuk menganalisis kebutuhan sistem yang akan dirancang. Kebutuhan sistem ini antara lain bisa melakukan *event entry*, pemantauan jumlah peserta, pendaftaran peserta, hingga konfirmasi pembayaran peserta.

7.4 Analisis Model Konseptual Sistem

Membuat model rancangan sistem dalam bentuk diagram yang mewakili abstraksi sistem yang akan dirancang.

7.5 Membuat Data Flow Diagram (DFD)

Langkah ini dilakukan untuk membuat sebuah skema aliran data dalam rancangan sistem. Terdapat dua entitas yang akan terlibat dalam diagram konteksnya, yaitu entitas EO dan User.

7.6 Merancang Entity Relational Diagram (ERD)

Merancang relasi antarentitas yang dibutuhkan dalam perancangan sistem. Terdapat beberapa entitas yang terlihat seperti EO, User, Bank dan Speaker.

7.7 Merancang Tabel

Hasil dari rancangan ERD akan digunakan untuk membuat rancangan tabel yang dibutuhkan dalam sistem. Rancangan tabel dibuat dengan menggunakan fitur *designer* dari *tools PHPMyAdmin*.

7.8 Merancang Antarmuka

Hasil dari rancangan ini adalah berupa gambar *user interface* dalam bentuk form-form. Beberapa antarmuka yang dikembangkan antara lain *push*

notification, Detail Event, *attendees progress*, dan lain sebagainya.

8. HASIL DAN PEMBAHASAN

8.1 Analisis Kebutuhan User

- Pengelola dapat mem-*publish* acara mereka
- User dapat mengetahui ada event terbaru
- User dapat melihat deskripsi acara
- User dapat memesan tiket acara
- User dapat membayar tiket acara
- Pengelola dapat mengetahui daftar peserta yang melakukan pemesanan tiket
- Pengelola dapat mengetahui daftar peserta yang telah melakukan pembayaran
- Pengelola dapat mengirimkan pesan broadcast
- User dapat mengetahui jumlah peserta

8.2 Analisis Aturan Bisnis

- Hanya pengelola yang terdaftar yang bisa mem-*publish* acara.
- Pengelola & user harus melakukan aktivasi
- User harus memasang aplikasi di HP untuk bisa melakukan pemesanan tiket/melihat informasi
- Hanya *user* yang terdaftar yang bisa melakukan pemesanan & pembayaran tiket
- Pembayaran hanya dapat dilakukan dengan transfer dan deposit.
- Bila saldo deposit pembayaran tidak mencukupi, maka transaksi *user* dibatalkan.

8.2 Perancangan Diagram Konteks

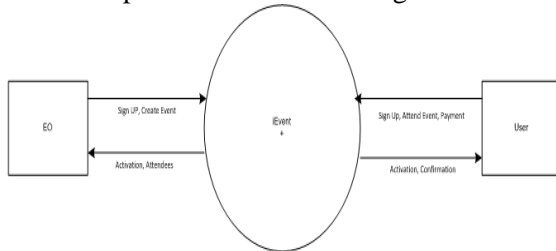
Diagram konteks merupakan penggambaran dari seluruh system yang akan dibuat. Tahap ini akan memperjelas hubungan antara masukan yang diberikan oleh pengguna system dan hasil keluaran dari system kepada pengguna.

Seperti pada gambar 4, sistem yang dirancang diberi nama *iEvent* dengan dua pengguna sistem yaitu EO yang merupakan singkatan dari Event Organizer dan juga User.

Diagram konteks ini menggambarkan tentang pengguna system EO mengalirkan data keluar berupa Sign Up EO dan Create Event yang kemudian oleh system *iEvent* akan mengirimkan data berupa Activation dan Attendees. Sign Up EO berisi data-data Event Organizer sebagai

pengelola *event* dan Create Event berisi data-data *event* yang diselenggarakan oleh EO.

Sedangkan pengguna User mengalirkan data Sign Up User, Attend Event dan Payment, yang kemudian menerima data berupa Activation dan Confirmation. Adapun Sign Up User berisi data-data User sistem iEvent. Attend Event berisi data-data User dan kode *event* yang nantinya akan diikuti. Untuk Confirmation akan berisi data konfirmasi dari sistem bahwa User telah berhasil melakukan pendaftaran untuk mengikuti *event*.



Gambar 4 Diagram Konteks iEvent

8.3 Perancangan Data Flow Diagram (DFD)

Setelah merancang sebuah diagram konteks, kemudian akan dirancang DFD yang akan menjadi penggambaran dari seluruh system yang akan dibuat. DFD adalah sebuah cara untuk memodelkan alur data yang terjadi pada sebuah system.

8.3.1 DFD Level 0

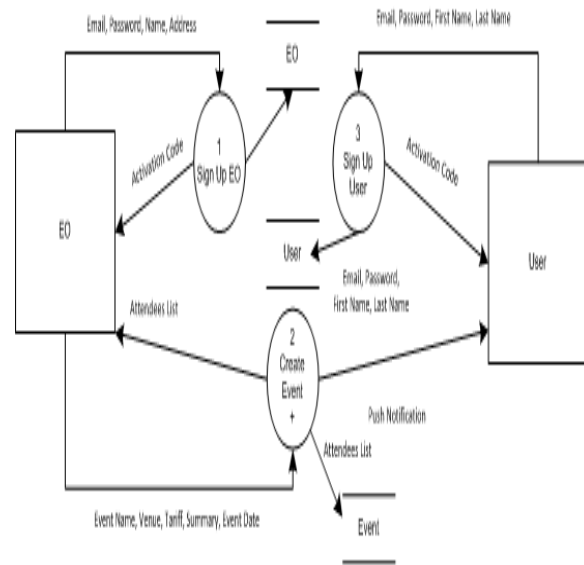
DFD Level 0 menggambarkan rancangan hasil dekomposisi dari konteks diagram. Pada DFD Level 0 ini akan terbentuk tiga proses yaitu proses Sign UP EO, Create Event dan Sign Up User.

Pada proses 1, yaitu Sign Up EO, datanya berasal dari entitas EO berupa Email, Password, Name dan Address. Seluruh data tersebut akan disimpan dalam *store* EO. Sedangkan data yang dihasilkan oleh proses 1 tersebut adalah berupa Activation Code yang dialirkan ke entitas EO.

Untuk proses 2, yaitu Create Event, datanya berasal dari entitas EO berupa Event Name, Venue, Tariff, Summary dan Event Date. Seluruh data tersebut akan disimpan dalam *store* Event. Sedangkan data yang dihasilkan dari proses 2 tersebut nantinya berupa Attendees List yang dialirkan ke entitas EO dan Push Notification yang dialirkan ke entitas User.

Pada proses 3, yaitu Sign Up User, datanya berasal dari entitas User berupa Email, Password, First Name dan Last Name. Seluruh data tersebut akan disimpan dalam *store* User. Sedangkan data yang dihasilkan dari proses 3 adalah berupa Activation Code yang dialirkan ke entitas User.

Untuk lebih lengkapnya penjelasan DFD Level 0 dapat dilihat pada gambar 5.



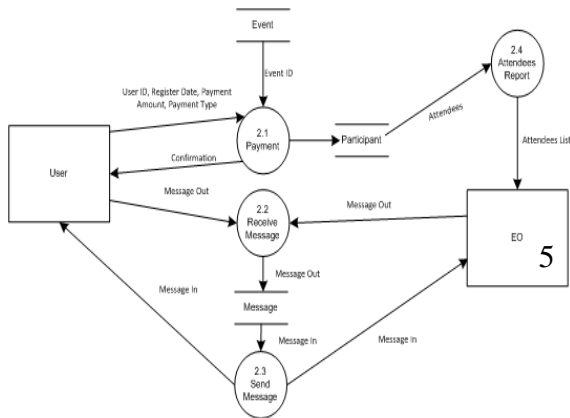
Gambar 5 DFD Level 0

8.3.2 DFD Level 1 Proses 2

DFD Level 1 menggambarkan rancangan hasil dekomposisi dari DFD Level 0 untuk proses 2 yaitu Create Event.

Pada proses 2.1 yaitu Payment, datanya berasal dari entitas User yaitu User ID, Register Date, Payment Amount, Payment Type dan dari *store* Event berupa data Event ID. Seluruh data yang masuk ke proses 2.1 akan disimpan ke dalam *store* Participant. Nantinya ketika entitas EO ingin melihat *report* berupa daftar Attendees, maka proses ini akan dilakukan oleh proses 2.4. Seluruh data dari *store* Participant akan dikirimkan ke proses 2.4, untuk selanjutnya dialirkan ke entitas EO dalam bentuk Attendees List.

Untuk proses 2.2 yaitu Receive Message, data Message Out dialirkan dari entitas User ke proses 2.2 untuk disimpan ke *store* Message. Dengan kata lain proses ini menggambarkan bagaimana User ataupun EO nanti bisa mengirimkan pesan kepada satu sama lain yang pertama-tama harus disimpan ke media penyimpanan terlebih dahulu. Selanjutnya dari *store* Message data dikeluarkan ke entitas User dan EO berupa Message In. Artinya proses 2.3 ini menggambarkan pesan yang dikirimkan tadi akan diteruskan kepada User atau EO. Untuk selengkapnya bisa dilihat pada gambar 6.



Gambar 6 DFD Level 1 Proses 2 Create Event

8.4 Perancangan Relasi Entitas

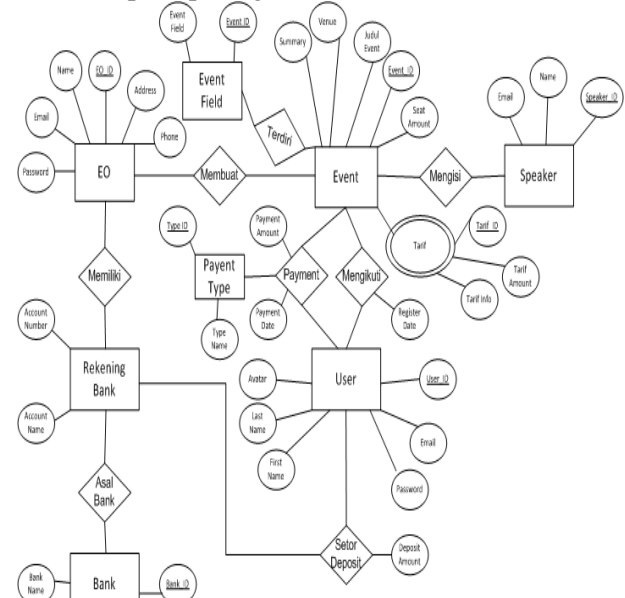
8.4.1 Model Data Konseptual ERD (Entity Relationship Diagram)

Dalam perancangan konsep ERD dihasilkan sebanyak delapan entitas yaitu EO, User, Event, Speaker, Bank, Event Field, Payment Type dan Rekening Bank. Dari kedelapan entitas tersebut memiliki delapan relasi yaitu relasi Membuat antara entitas EO dan Event, relasi Mengisi antara entitas Speaker dan Event, relasi Memiliki antara EO dan Rekening Bank, relasi Terdiri antara table Event dan Event Field, relasi Payment antara Event, Payment Type dan User, relasi Setor Deposit antara User dan Rekening Bank, relasi Asal Bank antara Bank dan Rekening Bank serta terakhir relasi Mengikuti antara User dan Event. Adapun detail dari atribut dari masing-masing entitas adalah sebagai berikut:

1. Entitas EO memiliki atribut antara lain EO_ID, Name, Email, Password, Address dan Phone.
2. Entitas Event memiliki atribut yaitu Event_ID, Judul Event, Venue, Summary, Seat Amount dan Tarif. Khusus atribut Tarif bersifat multi valued dengan atribut Tarif_ID, Tarif Amount dan Tarif Info.
3. Entitas Speaker terdiri dari atribut Speaker_ID, Name dan Email.
4. Entitas User memiliki atribut User_ID, First Name, Last Name, Email, Password dan Avatar.
5. Entitas Rekening Bank memiliki atribut Bank_ID, Bank Name, Account Name dan Account Number.
6. Entitas Bank memiliki atribut Bank_ID dan Bank_Name.

7. Entitas Payment Type memiliki atribut Type_ID dan Type Name.
8. Entitas Event Field memiliki atribut Event_ID dan Event_Field.

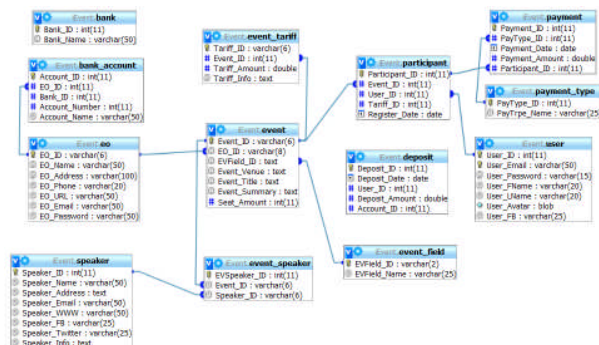
Rancangan konseptual ERD selengkapnya terlihat seperti pada gambar 7.



Gambar 7 Entity Relationship Diagram

8.5 Perancangan Tabel

Berdasarkan pada rancangan ERD sebelumnya maka selanjutnya dikembangkan rancangan struktur tabel beserta *mapping* relasi antartabel. Didapatkan sebanyak tiga belas tabel yang terdiri dari delapan tabel utama dan lima tabel hasil relasi seperti yang terlihat pada gambar 8. Kedelapan tabel utama tersebut antara lain Bank, Bank Account, EO, Speaker, Event, Deposit, User, Payment Type. Sedangkan kelima tabel hasil relasinya adalah tabel Participant, Payment, Tarif, Deposit dan Event Speaker.

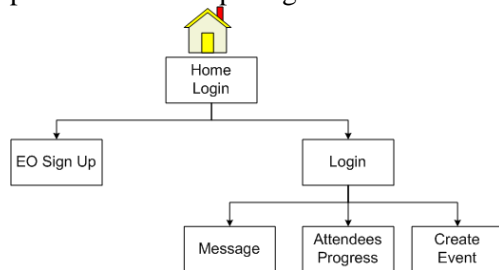


Gambar 8 Rancangan tabel dan *mapping* relasi antartabel

8.6 Perancangan Struktur Menu

Berdasarkan pada proses bisnis yang telah ditetapkan maka terdapat dua sistem menu yang digunakan yaitu sebagai berikut:

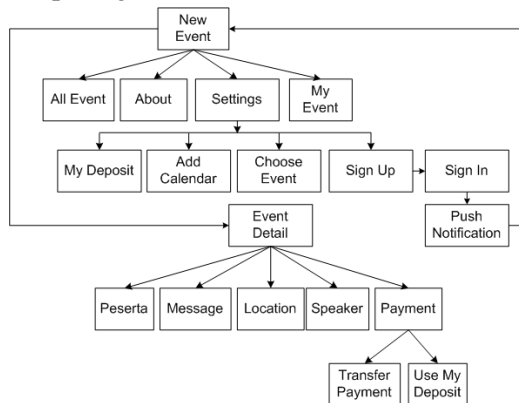
1. Struktur pohon (*tree structure*) untuk antarmuka aplikasi berbasis Web [7].
Struktur pohon digunakan karena dalam penerapannya nanti terdapat satu kelompok menu yang mesti dibagi yaitu ketika *user* berhasil *login* ke sistem Web, maka di dalamnya akan terdapat tiga pilihan dalam satu kelompok menu yang terdiri dari menu Message, Attendees Progress dan Create Event. Selengkapnya gambaran struktur pohon ini terlihat pada gambar 9.



Gambar 9 Struktur Menu EO Web Administrator iEvent

2. Struktur *cyclic network* untuk antarmuka aplikasi *user mobile*.

Struktur *cyclic network* [7] digunakan karena ada penggunaan kombinasi menu pohon di dalamnya. Terdapat pengelompokkan menu yang menggunakan menu pohon yaitu pada menu New Event, Settings, Event Detail dan Payment. Menggunakan *cyclic network* karena pada menu Push Notification yang merupakan menu anak dari menu Sign Up langsung mengarah ke menu New Event. Begitu pula dari menu New Event langsung mengarah ke menu Event Detail. Selengkapnya gambaran menu *cyclic network* ini terlihat pada gambar 10.



Gambar 10 Rancangan Struktur Menu User Mobile iEvent

8.7.1 Antarmuka Push Notification

Antarmuka Push Notification berfungsi sebagai informasi atau pemberitahuan adanya *event* baru yang dibuat oleh EO langsung ke piranti *mobile* pengguna. Informasi yang ditampilkan adalah nama, tempat dan tanggal *event*. Untuk selengkapnya bisa dilihat pada gambar 11.



Gambar 11 Antarmuka Push Notification

8.7.2. Antarmuka Detail Event

Antarmuka Detail Event ini akan digunakan untuk menampilkan seluruh informasi sebuah *event* yang akan diselenggarakan. Di dalamnya memuat informasi seperti EO, Title hingga Contact.

Apabila *user* menekan tombol Register Now di akhir detail *event* maka otomatis data *user* tersebut akan tersimpan ke sistem iEvent dengan status Register Unpaid seperti yang terlihat pada gambar 12.

Selain itu juga *event* yang dipilih tadi akan masuk ke daftar My Event. Jika *user* ingin melakukan pembayaran, maka dapat langsung menekan menu Payment di *bottom tab menu* yang selanjutnya akan membuka form antarmuka Payment.

Akan tetapi bila *user* ingin membayar *event* tersebut di lain waktu, maka ketika bermaksud untuk melakukan pembayaran, *user* cukup menekan menu My Events di *bottom tab menu*. Selanjutnya akan muncul daftar *event* yang telah diregistrasi oleh *user* atau Upcoming Events dan juga daftar *event* yang pernah diikuti oleh *user* atau Past Events. Bila *user* membuka detail *event* pada daftar My Event, maka selanjutnya tombol Register Now telah berubah menjadi tombol Pay Now. Jika tombol tersebut diklik maka akan muncul form Payment. Selengkapnya dapat dilihat pada gambar 13.

8.7 Perancangan Antarmuka



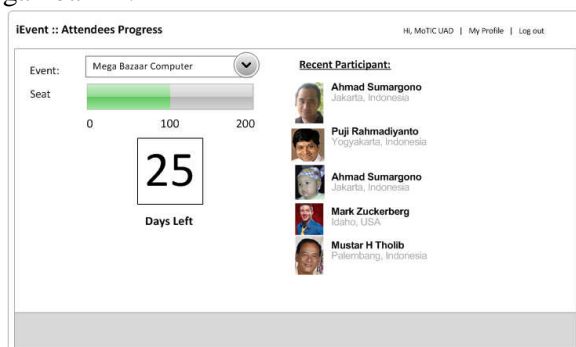
Gambar 12 Anatarmuka Detail Event Register Now



Gambar 13 Antarmuka Detail Event Pay Now

8.7.3 Antarmuka Attendees Progress

Setiap kali *user* berhasil melakukan pembayaran maka otomatis terjadi perubahan status yang sebelumnya Register Unpaid menjadi Register Paid. Untuk itu pihak Administrator EO dapat melihat perkembangan peserta yang telah terdaftar lewat antarmuka *web* yaitu Attendees Progress. Selengkapnya dapat dilihat pada gambar 14.



Gambar 14 Antarmuka Attendees Progress

9. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan berhasil disimpulkan beberapa hal sebagai berikut:

- Telah berhasil dirancang sebuah sistem manajemen *event* yang bisa mengelola sebuah acara mulai dari sisi administrator pihak EO hingga sisi *user* yang menggunakan antarmuka berbasis piranti *mobile* khususnya.
- Rancangan yang dibuat memungkinkan para *user* bisa melakukan registrasi hingga melakukan pembayaran sekaligus.

10. SARAN

Untuk pengembangan lebih lanjut maka beberapa saran (*future works*) yang dapat dilakukan adalah:

- Perlu dilakukan perancangan lebih lanjut mengenai klasifikasi kelas tiket seperti kelas Reguler, Premium, Platinum yang biasanya berlaku untuk beberapa *event* tertentu.
- Perlu dilakukan perancangan dan mekanisme penerapan *mobile ticket* dan *mobile check-in* yang memungkinkan untuk dijadikan sebagai sarana pengganti tiket manual.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Ahmad Dahlan (UAD) yang telah memberi dukungan finansial terhadap penelitian ini melalui program penelitian reguler yang diselenggarakan oleh Lembaga Penelitian dan Pengembangan UAD dengan nomor kontrak M.85/LPP-UAD/I/2011.

DAFTAR PUSTAKA

- [1] Apple, 2010, Apple Push Notification Service, <http://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html>, diakses 12 Januari 2011.
- [2] Google, 2011, Android Cloud to Device Messaging Framework, <http://code.google.com/intl/id/android/c2dm/>, diakses 12 Januari 2011.
- [3] Grey, A.M. dan Reid, K.S., 2006, *Event Sponsorship*, Edisi Bahasa Indonesia, Penerbit PPM, Jakarta.
- [4] Michelle, 2010, Android Cloud to Device Messaging to Mobile Phones, <http://mobile.downloadatoz.com/tutorial/440,android-cloud-to-device-messaging-to-mobile-phones-how-to.html>, diakses 13 Januari 2011.
- [5] Putra, S.M., Deliana, H., Rosmiyani, dan Halim, F., 2010, *E-Statement Pada Smartphone BlackBerry*, Prosiding SNATI UII 2010, 19 Juni 2010, halaman B-78-B-83, Yogyakarta.

- [6] Sanjaya, I., 2008, Kompetisi Layanan Push Mail di Kalangan Operator Seluler, <http://balitbang.depkominfo.go.id/addfile/jurnal/buletin%20postal/buletin%20vol.6%20no.2%20juni%202008/iman.doc>, diakses 30 Oktober 2010.
- [7] Shneiderman, B. 1998. *Designing The User Interface: Strategies for Effective Human Computer Interaction*. Massachusetts: Addison Wesley Longman.
- [8] Suseno, I, 2005, *Cara Pinter Jadi Event Organizer*, CV. Galang Press, Yogyakarta.
- [9] Wikipedia, 2010, Push Technology, http://en.wikipedia.org/wiki/Push_technology, diakses 30 Oktober 2010.
- [10] Wikipedia, 2010, Komputasi Awan, http://id.wikipedia.org/wiki/Komputasi_awan, diakses 30 Oktober 2010.