Pembangunan Class Library Untuk Domain Aplikasi Event Ticketing Pada Platform Windows Phone

Hermina Nur Karimah[1]

[1] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : herminanurkarimah@yahoo.co.id[1]

ABSTRAK

Berdasarkan survei yang dilakukan oleh Arts Organizations and Digital Tachnologies terhadap 1000 responden yang telah disurvei secara global menyatakan bahwa penjualan tiket secara online menggunakan teknologi mobile mencapai 72%. Hal ini memicu berkembangnya pembangunan aplikasi event ticketing. Pembangunan perangkat lunak dengan waktu yang terbatas membuat banyak pengembang hanya dapat mengimplementasikan daripada menganalisis suatu perangkat lunak dengan domain aplikasi tertentu. Permasalahan yang timbul adalah perangkat lunak yang dibangun belum optimal karena pengembang kesulitan untuk menggunakan kembali fungsi yang pernah dibuat sebelumnya. Solusi yang ditawarkan pada penelitian ini adalah dengan membangun sebuah class library yang menyediakan fungsi umum pada domain aplikasi event ticketing.
Class library merupakan kumpulan kelas dan method yang menyediakan fungsionalitas sebagai dasar pembangunan sebuah sistem. Pada tahap awal pada penelitian ini dilakukan analisis domain terhadap tiga aplikasi untuk memperoleh frozenspot dan hotspot. Kelas-kelas pada class library dirancang dan kemudian diimplementasikan berdasarkan hotspot yang telah didefinisikan. Hasil implementasi dari class library diuji untuk memastikan class library dapat berjalan dengan baik dan sesuai kebutuhan pengguna.
Pengujian class library dilakukan dengan metode unit testing, pengujian integrasi, dan pengujian acceptance. Dari hasi pengujian maka dapat disimpulkan bahwa class library event ticketing dapat memudahkan pembangunan perangkat lunak event ticketing dengan menyediakan fungsi-fungsi umum yang terdapat pada domain event ticketing yang dapat digunakan kembali untuk proses pengembangannya. Sehingga pengembang dapat mengoptimalkan pengembangan aplikasi. Saran yang diberikan adalah menambahkan exception dan fungsi tambahan lainnya.

Kata Kunci: *class library*, *event ticketing*.

1 PENDAHULUAN

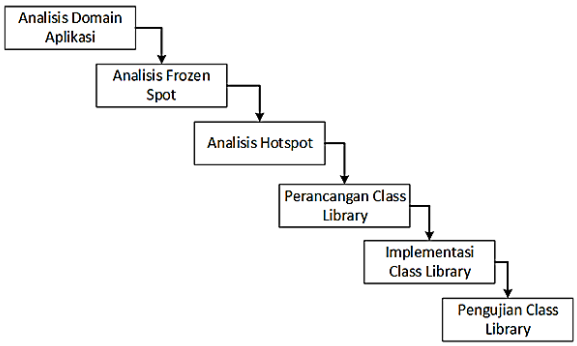
Pembangunan aplikasi mobile dengan domain *event ticketing* seharusnya dapat lebih optimal karena berdasarkan observasi dari tiga buah aplikasi sejenis diketahui bahwa terdapat beberapa kesamaan fungsional dasar. Seharusnya fungsional dasar tersebut dapat digunakan kembali pada pembangunan aplikasi sejenis. Keterbatasan waktu dalam pembangunan sebuah aplikasi dan pembuatan fungsi yang sama secara berulang-ulang pada suatu domain aplikasi *event ticketing* membuat aplikasi yang dibangun oleh pengembang tidak optimal. Hal ini lah yang memicu pembangunan *class library* sebagai sebuah solusi dalam pembangunan perangkat lunak *event ticketing*.
Class library merupakan sekumpulan fungsi yang dapat mempermudah programmer dalam membangun perangkat lunak. Class library yang dibangun haruslah konsisten dan dapat terdokumentasi dengan baik sehingga mudah untuk digunkan.

1.1 Class Library

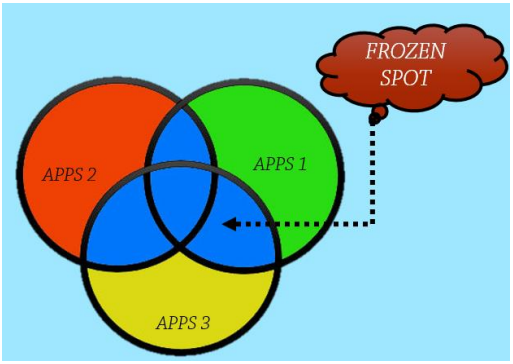
Class library merupakan kumpulan kelas-kelas, interface, dan tipe data yang membentuk sebuah library [1]. Class libarary berfungsi dalam menyediakan fungsionalitas untuk sebuah sistem yang dirancang untuk menjadi fondasi dari aplikasi, komponen-komponen yang dibangun, dan kontrol yang dibangun. *Class library* biasanya disimpan pada sebuah file berextensi .dll, barulah selanjutnya dapat digunakan. Pembuatan class library haruslah terdokumentasi agar dapat mempermudah penggunaannya serta penamaan yang konsisten dan sesuai dengan pedoman.

1.1.1. Metode Penyelesaian Masalah

Terdapat enam tahap dalam menyelesaikan masalah untuk pembangunan class library, yaitu analisis domain aplikasi, analisis frozen spot, analisis hotspot perancangan class library, Implementasi class library, dan pengujian class library.



Gambar 1 Metode penyelesaian masalah



Gambar 2 Ilustrasi *frozen spot*

1.2 Analisis Domain

Untuk mengetahui kesamaan fungsional yang terdapat dalam aplikasi maka dilakukan analisis terhadap tiga buah aplikasi sejenis. Tahap-tahap dalam analisis domain antara lain analisis *frozen spot*, analisis *hotspot*, dan pendefinisian kartu hotspot.

1.2.1 *Elektronik Tiketing*

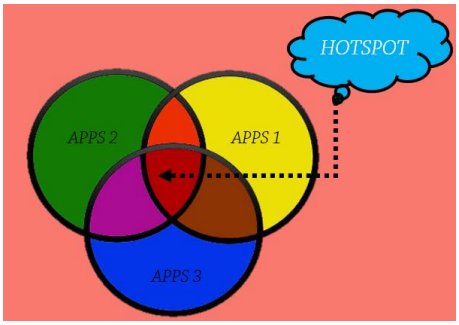
Tiket Elektronik (*Electronic Ticketing*) merupakan teknologi yang digunakan dalam pemesanan tiket secara online. Tiket elektronik telah dikembangkan untuk mengganti tiket kertas dan *voucher* kedalam bentuk elektronik. Keuntungan digunakannya tiket elektronik adalah pengguna mampu membeli tiket masuk, *voucher* makan dan minuman yang didownload pada suatu kartu. Semua orang dapat menggunakan smartphone untuk memesan tiket bukan menunggu dalam antrian untuk membeli kertas tiket. Tiket elektronik digunakan untuk mempercepat transaksi dan memudahkan penggunanya dalam pemesanan tiket.

1.2.2 *Frozen spot*

Frozen spot berisi fungsi-fungsi tetap yang terdapat pada aplikasi pada domain kasus tertentu [2]. Frozenspot adalah irisan dari fungsional dasar yang terdapat pada setiap aplikasi.

1.2.3 *Hotspot*

Hotspot merupakan titik fleksibilitas dari aplikasi yang artinya fungsional yang pasti terdapat pada setiap aplikasi pada domain kasus tertentu [2]. Penentuan *Hotspot* merupakan uraian dari *frozen spot*. *Hotspot* yang dirancang harus lebih spesifik dan dapat disesuaikan dengan kebutuhan domain. Metode untuk menentukan *hotspot* yaitu metode *blackbox* dan *whitebox*. Metode *blackbox* yaitu metode untuk menentukan hotspot bedasarkan hasil analisis dari frozen spot. Pada metode *blackbox* fungsional yang dapat dijadikan menjadi hotspot harus terdapat pada setiap aplikasi pada domain aplikasi tertentu. Sedangkan menentukan hotspot dengan menggunakan metode *whitebox* dilakukan berdasarkan kebutuhan dari hotspot lainnya.

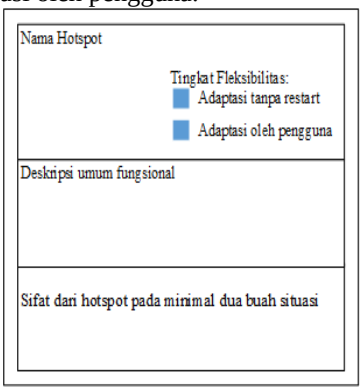


Gambar 3 Ilustrasi *Hotspot*

1.2.4 Kartu Hotspot

Kartu Hotspot digunakan untuk mendeskripsikan fungsionalitas dari setiap hotspot yang ada [3]. Katru

hotspot haruslah menggambarkan fungsionalitas pada hotspot yang telah didefinisikan. Bagian-bagian dari kartu hotspot, yaitu nama hotspot, derajat fleksibilitas, deskripsi fungsional, dan perilaku hotspot terhadap dua situasi yang berbeda. Derajat fleksibilitas ada dua jenis yaitu adaptasi tanpa restart dan adaptasi oleh pengguna.



Gambar 4 Ilustrasi Kartu Hotspot

1.3 Pengujian

Pengujian dilakukan untuk mengetahui apakah *class library* yang dibuat telah sesuai dengan kebutuhan atau tidak. Terdapat tiga pengujian yang dilakukan pada penelitian ini, yaitu *unit testing*, pengujian integrasi, dan pengujian *acceptance*.

1.3.1 *Unit Testing*

Unit testing merupakan sebuah program yang digunakan untuk menguji sebuah method atau kelas tertentu. Pengujian unit testing dilakukan dengan menerapkan beberapa skenario pengujian yang berbeda. Unit test mengeksekusi method yang diuji menggunakan data sampel sebagai parameter dan memeriksa apakah method yang dieksekusi tersebut telah benar dan berjalan dengan baik [4].

1.3.2 Pengujian Integrasi

Pengujian Integrasi dilakukan dengan mengintegrasikan suatu unit dengan unit yang lain yang saling berinteraksi sehingga menyusun suatu sistem. Pengujian integrasi digunakan untuk mengetahui kesalahan dan masalah yang terdapat pada perangkat lunak [5].

1.3.3 Pengujian Acceptance

Pengujian acceptance dilakukan untuk menentukan apakah perangkat lunak yang dibangun telah sesuai dengan tujuan. Pengujian *acceptance* memberikan kepercayaan kepada pengguna bahwa perangkat lunak yang dibuat telah sesuai dengan apa yang dibutuhkan oleh pengguna.

1.3.4 *Object Oriented Programming*

Object Oriented Programming merupakan sebuah pendekatan pengembangan perangkat lunak, dimana struktur perangkat lunak dianalisis berdasarkan objek yang saling berinteraksi untuk menyelesaikan tugasnya berdasarkan urutan logika yang telah ditentukan. Interaksi ini membutuhkan pertukaran pesan bolak-balik antara objek yang satu dengan objek yang lainnya. Setelah pesan diterima, objek dapat melakukan action atau method [6].

1.3.5 *Generic Programming*

Generic Programming bertujuan untuk meringankan beban programmer dengan cara memungkinkan fungsi umum untuk di definisikan sekali untuk banyak tipe data, bukan hanya untuk satu tipe data. Generic programming menggunakan menambahkan parameter program generik.

1.3.6 Pedoman pembangunan class library

Pedoman pembangunan Class Library ini merupakan pedoman yang digunakan pada .NET Framework. Tujuan dari pedoman ini adalah untuk membantu desainer class library untuk memastikan bahwa class library yang dibangunnya paham karena konsistensi API dan kemudahan penggunaan dengan menyediakan model pemrograman yang terpadu dan independen. Desain class library yang tidak konsisten akan merugikan produktifitas pengembang.

2 ISI PENELITIAN

Terdapat tiga tahap pada penelitian ini, yaitu tahap analisis *class library*, tahap perancangan *class libarary* dan tahap pengujian *class library*.

2.1 Analisis Class Library

Analisis class library memiliki tiga tahap yaitu analisis domain, analisis *frozen spot*, dan analisis hotspot.

2.1.1 Analisis *Frozen Spot*

Dari hasil analisis proses bisnis dan fungsionalitas pada ketiga aplikasi terdapat kesamaan fungsional. Terdapat sembilan fungsional yang ditetapkan menjadi frozen spot, yaitu authentifikasi pengguna, pemesanan tiket, pencarian event, penyajian detail event, penyajian data event, penyajian detail tiket, simpan event, penyajian kategori, dan integrasi dengan aplikasi lain.

Tabel 1 Hasil analisis *frozen spot*

Frozen spot	Keterangan
Authentifikasi pengguna (Login)	Authentifikasi pengguna terdapat pada ketiga aplikasi
Pemesanan tiket	Pemesanan tiket terdapat pada ketiga aplikasi
Pencarian event	Pencarian event hanya terdapat pada aplikasi Eventbrite dan Rukkus Event Tickets. Pencarian pada setiap aplikasi berdasarkan kata kunci, kategori, dan lokasi
Penyajian detail event	Penyajian detail event terdapat pada ketiga aplikasi.
Penyajian data event	Penyajian data event terdapat pada ketiga aplikasi
Penyajian detail tiket	Penyajian detail tiket terdapat pada ketiga aplikasi
Simpan event Favorit	Simpan Event hanya terdapat pada aplikasi Eventbrite dan Rukkus event ticket
Penyajian kategori	Penyajian kategori pada ketiga
Integrasi dengan aplikasi lain	Integrasi dengan aplikasi lain hanya terdapat pada aplikasi Eventbrite dan Rukkus Event Ticket

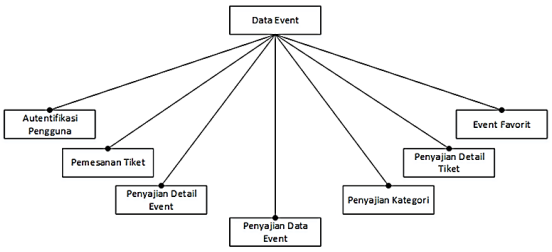
2.1.2 Analisis Domain

Domain kasus yang dipilih pada penelitian ini adalah event ticketing berdasarkan hasil wawancara terhadap salah satu programer. Domain aplikasi event ticketing merupakan aplikasi pencarian event dan pembelian tiket berbagai event secara online. Berdasarkan observasi yang dilakukan pada situs Windows Store terdapat beberapa aplikasi dengan domain kasus *event ticketing* masih belum optimal dalam dalam pembangunannya. Oleh karena itu pada pembangunan *class library event ticketing* pada penelitian ini dilakukan dengan menganalisis fungsi bisnis dan fungisonalitas dari tiga buah apilikasi yang menjadi tolak ukur dalam pembangunan *class library event ticketing*. Aplikasi yang menjadi tolak ukur pada penelitian ini antara lain adalah Eventbrite, Rukkus Event Tickets, dan Tiket.com.

Ketiga aplikasi tersebut memiliki domain aplikasi yang sama yaitu event ticketing. Ketiga aplikasi tersebut dipilih berdasarkan rating tertinggi dan fungsional yang terdapat dalam ketiga aplikasi tersebut. Aplikasi tersebut dianalisis persamaan, perbedaan, dan fitur-fitur yang terdapat dalam proses bisnis dari setiap aplikasi.

2.1.3 Analisis *Hotspot*

Hotspot ditentukan dengan terdapatnya kesamaan pada ketiga aplikasi dan kebutuhan dari hotspot lain. *Hotspot* ditentukan dengan metode *blackbox* dan *whitebox*. Terdapat enam fungsional yang didapat menggunakan metode *blackbox* dan sebuah fungsional yang didapat dengan metode *whitebox*. Setelah dianalisis terdapat enam fungsional yang muncul dari hasil analisis menggunakan metode blackbox yaitu Authentifikasi Pengguna, Pemesanan Tiket, Penyajian Detail Event, Penyajian Data Event, Penyajian Detail Tiket, dan Penyajian Kategori. Selain itu terdapat satu fungsional yang muncul dengan metode whitebox yaitu Simpan Event.*Hotspot* yang telah didefinisikan dapat dilihat pada.



Gambar 5 Diagram fitur *hotspot*

Fungsional umum yang terdapat dalam ketiga aplikasi telah ditentukan sebagai hotspot. Setelah *hotspot* ditentukan maka tahap selanjutnya adalah perancangan *class library*

2.2 Perancangan Class Library

Kelas dan method yang terdapat pada *class library* event ticketing dirancang dilakukan dengan penyesuaian dengan hasil analisis *hotspot*. Dari setiap *hotspot* yang telah didefinisikan didapat enam buah kelas dari pengakategorian berdasarkan keterkaitan funsi dan data pada setiap *hotspot* .

Tabel 2 Pengkategorian kelas

Hotspot	Kelas
Autentifikasi pengguna	User

Hotspot	Kelas
Penyajian data event	Event
Penyajian detail event	
Event favorit	
Penyajian kategori	Category
Pemesanan tiket	Order
Penyajian detail tiket	Tickets
	Base

Hasil dari perancangan kelas beserta method-method yang akan dibangun dapat dilihat pada Gambar 6. Dapat dilihat bahwa kelas User, Event, Category, Ticket, dan Order memiliki relasi komposisi terhadap kelas Base.

Pada Kelas User terdapat atribut User. Selain terdapat atribut juga terdapat method PostUserEdit, GetUserDetail, dan PostUser. Atribut dan method tersebut memiliki fungsi masing-masing.



Gambar 6 Perancangan kelas user

Pada Kelas Event terdapat atribut Event dan EventList. Selain terdapat atribut juga terdapat method GetEventList, GetEventDetail, dan PostFavoriteEvent.



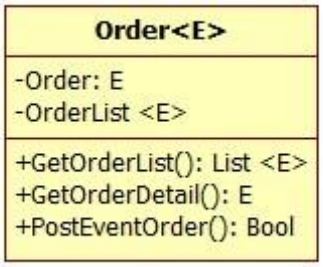
Gambar 7 Perancangan kelas event

Pada Kelas Category terdapat atribut CategoryList. Selain terdapat atribut juga terdapat method GetCategoryList.



Gambar 8 Perancangan kelas category

Pada Kelas Order terdapat atribut Order dan OrderList. Selain terdapat atribut juga terdapat method GetOrderList, GetOrderDetail dan PostEventOrder.



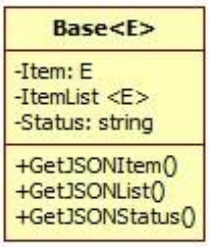
Gambar 9 Perancangan kelas order

Pada Kelas Tickets terdapat atribut Ticket dan TicketList. Selain terdapat atribut juga terdapat method GetTicketDetail dan GetTicketList.

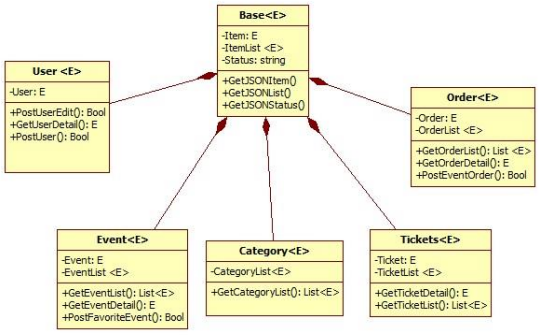


Gambar 10 Perancangan kelas tickets

Kelas Base memiliki atribut Item, ItemList, dan Status. Selain terdapat atribut juga terdapat method GetJSONItem, GetJSONList, GetJSONStatus.

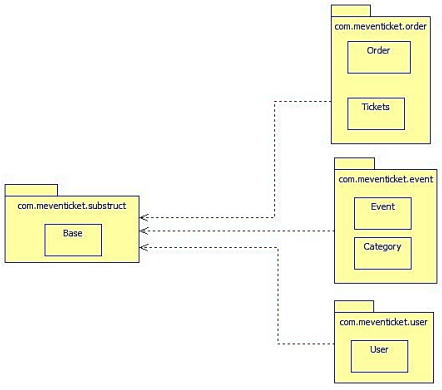


Gambar 11 Perancangan kelas base



Gambar 12 Perancangan Kelas

Kelas dan method-method yang telah dirancang kemudian dikelompokan menjadi beberapa *package*. *Package* digunakan untuk memudahkan untuk memahami struktur dan penggunaan dari class library. Penamaan *package* harus disesuaikan dengan pedoman penulisan. Terdapat empat buah *package* yang telah dirancang, yaitu com.meventicket.substruct, com.meventicket.user, com.meventicket.event, dan com.meventicket.order. Pada setiap package berisi kelas-kelas yang berelasi dengan kelas atau *package* lain. Perancangan *package* dapat dilihat pada Gambar 7.



Gambar 13 Perancangan *package*

2.3 Pengujian Class Library

Pengujian class library digunakan untuk mengetahui bahwa class library yang dibangun sesuai dengan apa yang telah direncanakan. Pengujian class library dilakukan dengan menggunakan *unit testing*, pengujian integrasi, dan pengujian acceptance. Setiap pengujian yang dilakukan dibuat skenario pengujian. Rincian dari pengujian unit testing dapat dilihat pada Tabel 3.

Tabel 3 Skenario *unit testing*

Item Uji	Poin Pengujian
Kelas User	Uji Method *GetUserDetail*
	Uji Method *PostUserEdit*
	Uji Method *PostUser*
Kelas Event	Uji Method *GetEventList*
	Uji Method *GetEventDetail*
	Uji Method *PostFavoriteEvent*
Kelas Category	Uji Method *GetCategoryList*
Kelas Order	Uji Method *GetOrderList*
	Uji Method *GetOrderDetail*
	Uji Method *PostEventOrder*
Kelas Tickets	Uji Method *GetTicketList*
	Uji Method *GetTicketDetail*

Pengujian integrasi dilakukan dengan mengintegrasikan class libaray *event ticketing* yang dibangun dengan sebuah aplikasi *event ticketing*. Rincian dari pengujian integrasi dapat dilihat pada Tabel 4.

Tabel 4 Skenario pengujian integrasi

Item Uji	Jenis Pengujian
GetUserDetail	*Integration Testing*
PostUserEdit	*Integration Testing*
PostUser	*Integration Testing*
GetEventList	*Integration Testing*
Detail Event	*Integration Testing*
PostFavoriteEvent	*Integration Testing*
GetCategoryList	*Integration Testing*
GetOrderList	*Integration Testing*
GetOrderDetail	*Integration Testing*
PostEventOrder	*Integration Testing*
GetTicketList	*Integration Testing*
GetTicketDetail	*Integration Testing*

Pengujian *Acceptance* dilakukan untuk mengetahui apakah *class library event ticketing* yang telah

dibangun telah sesuai dengan apa yang dibutuhkan *programmer*.

Pengujian integrasi *class library* dilakukan dengan mengimplementasikan class library event ticketing dalam membuat perangkat lunak *event ticketing* berbasis Windows Phone. Detail dari hasil implementasi fitur yang dikembangkan dari class library dapat di lihat pada tabel 5.

Tabel 5 Fitur hasil implementasi class library

Implementasi kelas	Fitur	Method yang digunakan
GetUserDetail	Menampilkan detail user	*GetUserDetail : UserItem(E)*
PostUser	Menambahakan user baru	PostUser(E)
GetEventList	Menampilkan list event	GetEventList : EventList(E)
GetEventDetail	Menampilkan detail dari event	GetEventDetail : EventItem(E)
PostFavoriteEvent	Menambahkan favorite event	PostFavoriteEvent(E)
GetCategoryList	Menampilkan list kategori	GetCategoryList : CategoryList(E)
GetOrderList	Menampilkan list pemesanan	GetOrderList : OrderList(E)
GetOrderDetail	Menampilkan detail pemesanan	GetOrderDetail : OrderItem(E)
PostEventOrder	Melakukan pemesanan tiket event	PostEventOrder (E)
GetTicketList	Menampilkan list ticket	GetTicketList : TicketList(E)
GetTicketDetail	Menampilkan detail ticket	GetTicketDetail : TicketItem

Pengujian acceptance dilakukan dengan tujuan untuk memberikan hasil pada programmer sebagai pengguna class library bahwa class library yang telah di bangun dapat digunakan dan memenuhi kebutuhan pengguna. Pengujian yang dilakukan pada penelitian ini adalah dengan mengajukan beberapa pertanyaan kepada beberapa programmer. Pertanyaan yang diajukan antara lain adalah sebagai berikut:

Pertanyaan yang diajukan	Kesimpulan hasil wawancara
Apakah *class library event ticketing* dapat membantu anda dalam membangun aplikasi *event ticketing?*	*Class library event ticketing* dapat membantu programmer dalam membangun aplikasi *event ticketing*.
Apakah *class library event ticketing* dapat membantu anda dalam membuat fungsionalitas pada pembangunan aplikasi *event ticketing*?	Class library event ticketing sangat membantu dalam membuat fungsionalitas dalam membangun aplikasi event ticketing.
Apakah dengan adanya dokumentasi *class libaray event ticketing*, penggunaan class library jadi lebih mudah?	Dokumentasi sangat membantu dan mempermudah programmer dalam penggunaan class library
Apakah fungsional-fungsional yang ada pada *class library event ticketing* sudah cukup mewakili fungsionalitas dasar pada aplikasi *event ticketing*?	Class library event ticketing sudah dapat mewakili fungsionalitas dasar pada aplikasi event ticketing
Pada *class library event ticketing* tidak dibangun *exception* atau penanganan *error*, apakah menurut anda hal tersebut diperlukan?	Penangnan error atau exception sangat diperlukan agar mempermudah programmer mengetahui letak kesalahan.

Pada bagian kesimpulan pengujian akan dipaparkan mengenai kesimpulan yang didapat setelah pengujian dilakukan. Kesimpulan yang diambil berdasarkan *unit testing*, pengujian integrasi dan pengujian *acceptance*.

1. Kesimpulan *unit testing class library*

Berdasarkan *unit testing* yang telah dilakukan maka dapat ditarik kesimpulan bahwa method yang yang terdapat dalam *class library* sudah dapat berjalan dengan baik namun terdapat

kekurangan karena tidak adanya penanganan eror.

2. Kesimpulan pengujian integrasi

Berdasarkan pengujian integrasi yang telah dilakukan maka dapat ditarik kesimpulan bahwa *class library* sudah dapat diintegrasikan kedalam aplikasi *event ticketing*.

3 KESIMPULAN

Dari proses penelitian Pembangunan *Class Library* dan pembangunan perangkat lunak uji, maka dapat disimpulkan bahwa:

1. Class library dapat memudahkan pengembang dalam pembangunan perangkat lunak *event ticketing* sehingga pengembang dapat mengoptimalkan pengembangan aplikasi yang dibuat.
2. Class library dapat memudahkan pengembang dengan menyediakan fungsi-fungsi umum yang terdapat pada domain *event ticketing* yang dapat digunakan kembali untuk proses pengembangannya.

Adapun saran dari hasil penelitian pembangunan Class Library pada domain aplikasi event ticketig antara lain:

1. Menambahkan fungsional pada class library event ticketing. Fungsional yang tedapat pada class library tidak hanya fungsional yang sederhana tetapi fungsional tambahan yang diperlukan dalam pembangunan aplikasi.

2. Menambahkan exception atau penanganan masalah pada setiap fungsi dalam class library sehingga programmer mengetahui letak kesalahan.

DAFTAR PUSTAKA

[1] "Class Library," Microsoft, [Online]. Available: https://msdn.microsoft.com/en-us/library/ca22fyhc%28v=vs.90%29.aspx. [Diakses 31 3 2015].

[2] N. Afif, "STUDI PEMBANGUNAN FRAMEWORK BERORIENTASI OBJEK STUDI KASUS PEMBANGUNAN FRAMEWORK PENJADWALAN," *Jurnal Teknosains*, vol. 7, no. 1, pp. 138-149, 2013.

[3] D. Yu, "Towards the Rapid Application Development Based on Predefined Frameworks," *JOURNAL OF SOFTWARE*, vol. 6, no. 9, pp. 1795-1804, 2011.

[4] S. Nakov, Fundamental of Computer Programming with C#, Svetlin Nakov, 2013.

[5] A. Orso, "Integration Testing of Object-Oriented Software," Dipartimento di Elettronica e Informazione, Milano, 1999.