BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Menduduki dan menjabat sebagai *Software Quality Assurance Intern* di perusahaan tiket.com yang ditempatkan didalam divisi SQA Core Apps dimana divisi tersebut lebih fokus terhadap hal-hal teknikal. Tugas-tugasyang diberikan sendiri lebih banyak menggunakan *software* seperti Charles Proxy untuk mereproduksi *error* dengan sengaja didalam aplikasi menggunakan API yang dipanggil oleh *software* tersebut. Tugas lainnya adalah untuk memastikan dan menjadi jembatan komunikasi antara *developer* dan *designer* sebuah fitur baru atau *button* baru untuk memastikan apakah yang dibuat oleh *developer* sudah sesuai dengan harapan dan desain darisisi *designer* dari segi bentuk, warna, lebar, dan spesifikasi lainnya. Sebagai SQA, bekerja didalam divisi SQA Core Apps dengan 9 rekan lainnya yang terdiri dari 3 *intern*, 5 pegawai tetap, dan 1 pengawas atau yang disebut *Technology Quality Lead* yang akan memandu kinerja divisi. Berikut adalah gambaran anggota divisi SQA Core Apps.



Gambar 3.1 Anggota Divisi SQA Core Apps

Gambar 3.1 merupakan divisi SQA *Core* yang dipimpin oleh Rohitash Mathur yang menjabat sebagai *Technology Quality Lead*. Setiap dua minggu sekali, beliau akan membagi tugas untuk setiap anggotanya yang disebut sebagai *sprint plan*. Pembagian tugas akan dibagikan melalui Microsoft Excel Sheet yang berisi data-data mengenai pembagian tugas, setelah pembagian tugas, setiap anggota diharuskan untuk membuat *subtask* didalam JIRA ID tersebut yang didalamnya berisi proses

pengerjaan *testing* dari QA. *Subtask* tersebut diharuskan berisi *steps to* 15 Peran Software Quality Assurance Intern ..., Patrick Kwak, Universitas Multimedia Nusantara reproduce, envinronment, testcases yang dibuat dengan TestRail, QA end date, QA effort, QA start date. Data-data tersebut digunakan untuk memastikan bahwa testcases yang dilakukan oleh QA sudah sesuai dengan harapan developer untuk meminimalisir adanya bug serta QA end date, effort, dan start date digunakan untuk memberitahu developer berapa banyak hari yang akan digunakan dan dipakai untuk melakukan testing pada JIRA ID tersebut. Komunikasi antara developer dengan QA dilakukan sesuai kesepakatan berdua seperti menggunakan Google Meet atau Slack karena banyak developer yang melakukan work from home serta banyak pegawai yang bukan merupakan warga negara Indonesia. Berikut adalah gambaran workflow dari QA divisi SQA Core Apps.

3.2 Tugas dan Uraian Kerja Magang

Dilakukan pekerjaan tugas bersifat P0 yaitu tugas dengan prioritas tertinggi yaitu Network Error Handling dan Common Error Handling. Network Error Handling ditujukan untuk membuat error screen yang lebih user friendly serta untuk memudahkan developer mengetahui apa penyebab terjadinya error ini karena memiliki kodenya masing-masing dari kode 400 hingga 500 dan exceptions. Commong error handling ditujukan untuk mengganti komponen yang digunakan untuk menampilkan error screen tersebut kedalam komponen yang lebih baru namun harus dipastikan bahwa perilaku setiap error screen sama meskipun menggunakan komponen baru sehingga harus dilakukan pengecekan pada setiap api yang dapat menghasilkan error screen di setiap vertical yaitu hotel, flights, TTD, ground transport, dan payment.

3.2.1 Tugas yang Dilakukan

Tabel dibawah merupakan informasi mengenai tugas apa saja yang dilakukan selama masa pelaksanaan kerja magang di perusahaan tiket.com.

No	Minggu	Proyek	Keterangan
1	27 Juni 2022 –	User Acceptance Test	Melakukan UAT terhadap
	15 Juli 2022	dan <i>Training</i>	aplikasi untuk mendapatkan
			wawasan mengenai aplikasi
			serta untuk mencari bug yang
			mungkin terdapat pada
			aplikasi <i>live</i> .
2	13 Juli 2022 – 15	Training	Menonton video dan slide
	Juli 2022		yang telah diberikan untuk
			menambah wawasan untuk
			apa saja yang akan dilakukan
3	18 Juli 2022 –	Network error handling	Mengganti error screen yang
	17 Oktober 2022		lama dengan yang baru yang
			lebih user friendly dari kode
			400 hingga 500.
			Memudahkan developer
			untuk mengetahui apa
			penyebabnya error karena
			diberikan kode dan memiliki
			informasi yang lebih jelas
			untuk user mengenai error
			yang terjadi.
	UNI	VERS	ITAS

Tabel 2. Tugas Kerja Magang

M U L T I M E D I A N U S A N T A R A

No	Minggu	Proyek	Keterangan
4	18 Oktober 2022 – 2 November 2022	Common error handling	Mengganti komponen <i>error</i> <i>screen</i> dengan komponen yang baru namun harus dipastikan bahwa <i>behaviour</i> setiap <i>error</i> harus sama dengan yang versi sebelumnya.
5	3 November2022 – 4 November 2022	Design task	Memverifikasi bug yang telah dibenarkan oleh developer, mengenai container dari sebuah label yang tidak berubah sesuai bentuk dan jumlah tulisan didalamnya sehingga tulisan tidak terlihat.

3.2.2 Uraian Kerja Magang

1. User Acceptance Test dan Onboarding Session

Pada hari pertama praktik kerja magang, diberikan tugas yang akan dilakukan selama kurang lebih 2 minggu, yaitu untuk melakukan *user acceptance test* pada aplikasi yang telah dirilis di Google Play Store. Tugas ini ditujukan untuk memperoleh informasi-informasi yang terdapat didalam aplikasi, memperoleh gambaran besar seperti apa alur kerja aplikasi tiket. Karena sebagai QA salah satu aspek terpenting adalah untuk mengetahui alur kerja yang benar, maka tugas ini diberikan pada tahapan pertama sebagai salah satu bentuk *training*. Jika *bug* ditemukan, maka dapat mengumpulkan informasi-informasi tersebut didalam sebuah *spreadsheet* yang kemudian akan dipresentasikan kepada *supervisor* setiap harinya melalui *stand up daily meeting* untuk melaporkan progres UAT.

2. Training

Setelah selesai dilakukan tugas pertama yaitu UAT, maka *supervisor* akan mengirimkan Google Drive yang berisi mengenai Powerpoint dan video mengenai ilmu-ilmu *testing* yang akan digunakan didalam masa praktik kerja magang, diberi waktu 2 hari untuk mempelajari dari video-video tersebut. Setelah *training* dilakukan maka akan melanjutkan ke tugas selanjutnya.

3. Network Error Handling

Tugas pertama adalah untuk melakukan dan memegang proyek yang bernama Network Error handling dimana tugas QA di proyek ini adalah untuk memastikan bahwa *error screen* sudah sesuai dengan *expected result* di aplikasi yang telah disediakan *developer* dalam setiap *vertical*. Di samping tugas untuk memastikan layar, diharuskan juga mengecek apakah kode *error* yang di *log* didalam Charles Proxy dan Multihead search sudah sesuai dengan kode *error* yang ditampilkan di layar menggunakan bantuan beberapa *software*. Salah satu *software* yang digunakan pada tahapan awal adalah JIRA.

ICMT [Minimond] McH - reit over gade of geterlot() -> getivetworkerror() - Part 1					
@ Attach @ Link issue ~ ☑ Add Checklist @Zeplin for Jira ····					QA Assignee
					Sartik Dodt Satrick Kwi
Description					QAIBurt
Envirronment: GK					3
Show Correct Network error message (NEH)					Do the Date
Steps to reproduce:					None
1. Hit a specific and in charles					
2. Put a breakpoint in that specific api					Der End Date
3. Retrigger the api that has been breakpoint					None
4. Change the status error code					QA Rant0ate
5. Observe the error screen					Sep 27, 2022
festase http://wettestrail.com/index.php?/suites/eev/2347.Botoup.pv-casessection.id/Botoup.prder-ascRidisplay.deleted.case=0					OA End Date
					Sep 29, 2022
Comments					
None					Story point edimate
					None
				one and	PRLink
Alsochments (4)		Part literation			None
	3528	Date acced -		100	Contra Grant Dates
Active or derits tcharles. PNG	5 KB	04 Oct 2022 9/38pm	0	Φ	None
Activerefundlistelastic.PNG	40 KB	04 Oct 2022 9:38pm	0	φ	
		01 0 4 100 1 0 10 4	0	0	Gantt End Date
Activererundlistonanes.envo	3 KB	ne oct 1022 alashu	0	Ψ	None
Active orderlistelastic PNG	49 KB	04 Oct 2022 9:38pm	0	ф.	Baseline start date
					None
					Pamiles excidents
davis.					None
Nove All Comments History Working TIS-Status Transitions TIS-Assignee Transitions Checklist history			Newst	first 42	
					Sprint
Add a comment					None +2
					Priority
					= Medium
Pro tage press and a continuent					
Polyap (Rosa an D. Caminan).					

Gambar 3.2 Informasi dari QA Subtask

Proyek dimulai dengan membuat *spreadsheet* sebagai sarana penyimpanan data-data yang akan dikumpulkan serta membuat *subtask* seperti gambar 3.2 didalam JIRA *parent task* dari Network error handling. Kemudian akan mengisi informasi didalam *subtask* tersebut berupa informasi cara melakukan *testing*, apa saja yang akan dilakukan *testing*, QA *effort*, *QA start date*, *QA end date* dimana informasi tersebut dapat dilihat oleh semua karyawan perusahaan. Informasi tersebut ditujukan sehingga *timeline* dan pekerjaan perusahaan tetap dapat dimonitor serta keefektifan setiap karyawan dapat dimonitor.



			-	
NCT-2629	Show correct Network Error message for getDownloadFileAsync API	=	AK	DON
NCT-2622	Show correct Network Error message for cance/Order API	=	AK	DON
NCT-2623	Show correct Network Error message for deleteOrder API	=	AK	DON
NCT-2624	Show correct Network Error message for getDownkadFileAsync(DownkadApiService) API	=	AK	DON
NCT-2625	Show correct Network Error message for getExtraBenefitImfo API	=	AK	DON
NCT-2627	Show correct Network Error m essage for getMyOrderDetailAirportTrain API	=	AK	DON
NCT-2628	Show correct Network Error message for gatOrderDetailAirportTransfer API	=	AK	DON
NCT-2629	Show correct Network Error message for getMyOrderDetailCariAsync API	=	AK	DON
NCT-2620	Show correct Network Error message for getMyOrderDetailTrainAsync API	=	AK	009
NCT-2631	Show correct Network Error message for getOrderDetailFlight API	=	AK	009
NCT-2622	Show correct Network Error message for getFlightGroup API	=	AK	DOR
NCT-2623	Show correct Network Error message for getMyOrderGroupAirportTrain API	=	AK	DOR
NCT-2635	Show correct Network Error m essage for getMyOrderGroupTrainAsyn: API	=	AK	DO
NCT-2626	Show correct Network Error message for getMyOrderEventDetailAsync API	=	AK	DO
NCT-2637	Show correct Network Error m essage for getMyOrderListNonLoginAPI	=	AK	DOR
NCT-2628	Show correct Network Error message for getPaymentDetailAsync API	=	AK	DO
NCT-2640	Show correct Network Error message for DownloadApiService[getReceipt] API	=	AK	DO
NCT-2641	Show correct Network Error message for getTravelDocuments API	=	AK	DO
NCT-2642	Show correct Network Error m essage for getMyHistoryOrderList API	=	AK	DOP
NCT-2643	Show correct Network Error message for deleteOrder API	=	AK	DOR
NCT-2644	Show correct Network Error message for getPastOrderList API	=	AK	DO
NCT-2645	Show correct Network Error message for getActiveOrderList API	=	AK	DO
NCT-2647	Show correct Network Error message for getPendingOrderList API	=	AK	DOR
NCT-2648	Show correct Network Error messages for getRefundList API	=	AK	DO
NCT 2650	Show correct Network Error message for getWaitingPaymentOrderList API	=	AK	DOP
NCT-2707	Show correct Network Error message for getOngoingRefund API	=	AK	DO
NCT-2772	Change gatError to getNetworkError in Remaining Apis where error is Not handled	-	AK	DO
	In the second se	_	-	_

Gambar 3.3 Informasi di dalam Parent Task

Setelah membuat *subtask*, tugasnya dimulai yaitu untuk melihat JIRA *parent task* seperti pada gambar 3.3 yang telah disediakan oleh *developer*, didalam JIRA *parent task* tersebut terdapat beberapa *subtask* lainnya sesuai dengan API yang akan dilakukan *testing*.

```
Show correct Network Error message for cancelOrder API

Attach P Link issue Add Checklist 2 Zeplin for Jia ...

Description

Please fix Network Error Handling for the following API to show Network Error message in case of Network/Internet errors

API Service: hyorderApiService

API: ms-gateway/tix-my-order-core/manage-order/cancel-order

Error Component: ErrorBottomSheetDialogNonDragableFragment.newInstance

API trigger point: Your order -- select order for which payment is not done then click ellipsize and cancel order then apiwill hit

PR Link: https://github.com/tiket/tiketandroid/pull/7620 - Connect to preview

Kome
```

Gambar 3.4 Informasi di dalam Subtask

Gambar 3.4 memiliki informasi mengenai masing-masing *subtask* yang akan membahas cara mereproduksi *error* yang akan dites serta *expected result* yang telah disediakan oleh *developer* untuk sehingga dapat mereproduksi *error screen* yang akan dilakukan *testing*.



Gambar 3.5 Software Charles Proxy

Gambar 3.5 menunjukkan tampilan awal dari *software* Charles Proxy. Setelah melihat *subtask* untuk setiap *vertical*, dapat mengunduh *software* Charles proxy untuk kebutuhan *testing*. Charles Proxy dapat diunduh di Google Chrome, namun setelah mengunduh, diperlukan konfigurasi didalam Charles Proxy agar dapat disambungkan dengan *emulator* atau *real device* sesuai yang dibutuhkan menggunakan Proxy. Dalam kegiatan saat ini, digunakan *real device* sehingga Charles Proxy dihubungkan dengan *real device* menggunakan IP Adress yang sama.

xcluded from this list are any loopback or link	local addresses.	
Network Interface	IP Address	
Realtek PCIe GbE Family Controller	192.168.1.2	
/irtualBox Host-Only Ethernet Adapter	192.168.56.1	

Gambar 3.6 IP Address untuk Menghubungkan Proxy

Gambar 3.6 menunjukkan IP address yang akan digunakan untuk menghubungkan Charles Proxy dengan *real device* yang dimiliki, setiap jaringan memiliki IP address yang berbeda serta IP address secara berkala berganti-ganti sehingga diperlukan pengecekan secara rutin agar koneksi dapat terhubung. IP address perlu diingat dan disalin ke proxy *real device* agar dapat terhubung.



10:04 🖸 🛋 🔌 🛱 🖓 .il 77% 🗎
Advanced
IP settings DHCP ▼
Proxy Manual 🔻
HTTP proxy used by browser but may not be used by other applications. Proxy host name
192.108.1.2
Bypass proxy for example.com,mycomp.test.com,localhost
User name
Password
Metered network Detect automatically
MAC address type
Cancel Save

Gambar 3.7 Konfigurasi Real Device Proxy

Gambar 3.7 menunjukkan tampilan konfigurasi untuk menghubungkan Charles Proxy dengan *real device*, memasukan IP address dan host dengan angka 8888 sebagai *default*. Setelah sudah menekan tombol save maka aplikasi akan otomatis terdeteksi di Charles Proxy dan dapat melanjutkan dengan kegiatan *testing* Network Error Handling (NEH) dengan cara mengecek dan mengubah *status code* pada Charles Proxy untuk dengan sengaja merubah *output output* yang dihasilkan berupa *error screens*.



	۲	₽	10	Ø	1	C	1	#	*	۲			
Structure Sequence	Ove	erview		nary	Chart								
> 😁 https://s-light.tiket.photos	Name			Value									
✓	Ho	st		https://s	-light.tike	t.photos							
> 💼 tix-eagleeye	Pat												
🗸 💼 ms-gateway	Not	tes		SSL Pro	xying not	enabled	for this ho	st. Enable	in the P	roxy Menu,	SSL Proxy	ing Setting	s
> intx-flight-master-discovery	Pro	tocols		HTTP/1									
> 🚞 tix-train-search-v2	✓ Red	quests											
> 🛅 tix-train-trx		Complet	ed										
> 🚞 tix-profile-core		Incomple	ete										
✓ == tix-payment-core		Failed											
> 🚞 payment		Blocked											
> 🚞 bankBin		DNS											
🗸 🚞 bca		Connect											
🗸 🛅 oneklik		TLS Han	idshakes										
🔀 credential?customerIdMerchar		Kept Aliv											
🕕 credential?customerIdMerchar	∨ Tin	ning											
🔀 credential?customerIdMerchar		Start		11/9/22	10:51:19								
🔀 credential?customerIdMerchar		End		11/9/22	10:51:42								
🔀 credential?customerIdMerchar		Timespa		22.68 s									
🔀 credential?customerIdMerchar		Request	s/sec	0.18									
🔀 credential?customerIdMerchar		Duration		26.20 s									
🔀 credential?customerldMerchar		DNS		461 ms									
🙆 credential?customerIdMerchar		Connect		170 ms									
闷 credential?customerIdMerchar		TLS Han	dshake	3.22 s									
> 🛅 tix-home		Latency		0 ms									
> 🛅 tix-push-notif		Speed		5.03 KB	/s								
> 🛅 tix-member-external		Request	Speed										
> 🚞 tix-member-core		Respons	e Speed										
> 🛅 tix-members-core	✓ Siz	e											
> 🛅 apiv1		Request	s	6.19 KB									
> 📀 https://mpayment.tiket.com		Respons	es	125.57	KB								
> 🍛 https://gql.tiket.com		Combine	d	131.76	КB								
		Compre	ssion										
Filter: tiket													
CONNECT https://android-context-data.googleapis.com											ap Local	Autophic	

Gambar 3.8 Tampilan Charles Proxy

Gambar 3.8 merupakan tampilan Charles Proxy ketika sebuah API sudah di *trigger* untuk diubah *status code* nya. *Status code* tersebut akan dilakukan *testing* dari 400-500 dan *exceptions*. Setiap *status code* sendiri memiliki hasil akhir yang berbeda-beda sesuai *expected result* yang sudah disetujukan oleh tim desainer. *Status code* yang akan dilakukan pengecekan adalah *status code* yang sudah disetujukan oleh tim *developer*, tidak semua *status code* akan dilakukan *testing*, tabel 2 akan menjelaskan *status code* apa saja yang akan dilakukan *testing* pada Network Error Handling.

Tabel 2. Status Code yang akan Dilakukan Testing

Status code	Description
400	Bad Request
422	Unprocessable Entity (WebDAV)
429	Too Many Requests

25

Peran Software Quality Assurance Intern ..., Patrick Kwak, Universitas Multimedia Nusantara

500	Internal Server Error
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
520	Web Server Returned an Unknown Error
522	Connection Timed Out
-103	ConnectException
-106	SocketTimeoutException
-300	StreamResetException

Kemudian akan meng *trigger* API dan mengubah *status code* nya yang kemudian layar aplikasi di *real device* akan mengeluarkan*error screens* yang berbeda-beda sesuai *status code* yang digunakan. Tujuan dari tugas ini adalah untuk memunculkan *error screens* yang lebih *friendly* terhadap *user* serta membuat *user* tidak merasa *stuck* jika suatu *error* terjadi. Pembaruan *error screens* juga ditujukan agar *developer* akan mengalami kemudahan untuk melakukan *tracking* jika terjadi kegagalan dalam sebuah sistem di aplikasi sesuai dengan *code* yang diterima di setiap *error screens*.





Gambar 3.9 Error Screens Versi Sebelum

Gambar 3.9 merupakan *error screens* pada aplikasi versi sebelumnya, aplikasi sebelumnya hanya memiliki secara total 4 jumlah *error screens* yang tidak *friendly* pada *user. Error screens* tidak memberi *user* arahan dan penjelasan mengenai kenapa *error* terjadi sehingga *user* merasa *stuck* ketika mengalami *error* tersebut. Oleh karena itu, diperlukan *error screens* untuk setiap *error code* yang ada sehingga ditugaskan untuk mengecek apakah *error screens* yang baru sudah sesuai dengan *expected result*.





Gambar 3.10 Error Screens Versi Sesudah

Gambar 3.10 merupakan salah satu bentuk dari *error screens* yang baru dengan *code* 400 dan -300, *code* diakhir memudahkan *developer* untuk melakukan *tracking* kepada *error* yang terjadi dan *message error screens* yang baru pun lebih memiliki arahan kenapa *error* terjadi untuk *user*. Diperlukan juga untuk melakukan pengecekan untuk setiap *status code* di setiap API yang ada sesuai API yang digunakan oleh setiap *vertical* seperti *vertical payment*, *flights*, *TTD*, *hotel*, *platform* dan *ground transport*. Setelah *testing* dilakukan dari sisi visual, maka akan dilakukan *testing* pada network error handling dari sisi teknikalnya yaitu dengan mengecek *eagleeye* yang akan ter *log* didalam database perusahaan serta Charles Proxy.



Gambar 3.11 Testing Teknikal di Eagleeye Charles Proxy

Gambar 3.11 merupakan hasil *testing* dari sisi teknikal yang dilakukan pada Charles Proxy, yaitu untuk mengecek *eagleeye* apakah *error* yang terjadi di aplikasi sudah ter *log* didalam Charles Proxy dengan*code* dan *message* yang sama sesuai yang terjadi di aplikasi. Jika *code* dan *message* berbeda dengan yang terjadi di aplikasi, dapat membuat JIRA *card* berbentuk *bug* untuk memberitahu *developer* bahwa terjadi kesalahan dibagian tertentu dan kemudian akan menunggu *bug fix* untuk dilakukan *testing* kembali pada aplikasi. Selain mengecek *code* dan *message* yang di *log* didalam Charles Proxy, diharuskan juga untuk mengecek dari aplikasi Multihead search yang dapat digunakan di Google Chrome.



	eventType	platform	osVersion	Result Sc	ource	applanguage	appState	Gmeadtoune
	NETWORK "corr	elationId": "31b	a71c8-93ac-4c2b	-847717574d21bc	308 1664892:	318697",		0 🔺
	IMAGE_LOAD net	workProvider": "" workTupe": "wifi"	vifi",	4.44.0	27201817	id	Foreground	0
_	IMAGE_LOADuri"	: Ahtiroid / apiros	itekaca.tiket.con	Krist gateway/tix-	2720191700		Eorelgröttind	0
	IMAGE_LOAD	uestPayloadSize poosePayloadSiz	": 0, e": 6001.	4,44.0	27201817	id	Foreground	0
_	IMAGE_LOAD	oonseCode": "40	0 0 %	4.44.0	27201817	id	Foreground	0
	IMAGE_LOADreg	oonselime": 6/: uestMime": "-",	12	4.44.0	27201817	id	Foreground	0
	IMAGE_LOAD	oonseMime": "aj	oplication/json;ch	arset=UTF-8",	27201817	id	Foreground	0
	IMAGE_LOADdns	Nameő:d'-",	12	4.44.0	27201817	id	Foreground	0
	NETWORK "even	Time": 0, ention": "-".	10	4,43,1	30182381	en	Foreground	0
- 4	NETWORK "inst	allSize": 0,	10	4.43.1	30182381	en	Foreground	0
	"web IMAGE_LOAD'rest	ViewCacheSize": :CacheSize": 0,	10	4,43,1	30182381	en	Foreground	0
_	IMAGE LOAD	rDataSize": 0,	10	4,43,1	30182381	en	Foreground	0
	IMAGE LOAD	SlotCount": 0,	10	4,43,1	30182381	en	Foreground	
	IMAGE LOAPpet	Count": 0, work Count": 0.	10	4,43,1	30182381	en	Foreground	
	IMAGE LOAD	tionAccuracy": 0	' 10	4,43,1	30182381	en	Foreground	
	IMAGE LOAD	ocation": { at":1-6.2736682	28918457,	4,43,1	30182381	en	Foreground	0
	IMAGE LOAD	lon": 106.62133	7890625	4,43,1	30182381	en	Foreground	
	IMAGE LOAD	rnalStorage": 0,	10	4,43,1	30182381	en	Foreground	
	"ext IMAGE LOAP"tot:	ernalStorage": 0	,, 10	4.43.1	30182381	en	Foreground	
	IMAGE LOAD	iUsed": 0,	10	4,43,1	30182381	en	Foreground	
	NETWORK "cad	spot": false, heHit": false.	10	4 43 1	30182381		Foreground	
	NETWORK "aco	essEnabled": fal	se,	4 43 1	30182381	en	Foreground	
	TMAGE LOAD	cery : 44, Charging": false,	10	4 4 2 1	20192291		Foreground	
	IMAGE LOAPdat	Saver": true,	10	4 43 1	30182381	en	Foreground	
	TMAGE LOAD	ProtocolVer": "H	ПТТР_2",	4.42.1	20102201		Foreground	
		'er": "TLS_1_2", nTime": 0,	10	4 4 2 1	20102201	en en	Foreground	
	"que	ry": "param=e3	o",	4 42 1	20102201		Foreground	
	TMAGE LOAD	InsInfo": [],	10	4.43.1	20102201	en 	Foreground	
	IMAGE_LOAG	tainerName": "o	om.tiket.gits.v2sp	olash. SplashV2Acti	vity",	en 	Foreground	
	"com	ntiket.android.fe	eature.orderlist.pi	resentation.orderli:	stcontainer. Or	derListContainerFrag	iment",	
	The contract of the second sec	inessCode": "SL inessMsg": "SUC	ICCESS", DCESS",	4,43,1	00140104	en	roreground	
	IMAGE_LOADBUS	ical": "order",	10.0	4.43.0	30148136	en	active	
	"use	rErrorTitle": "Un: rErrorMsg": "The	able to process y: : page you're goil	our request", ng to doesn't exist	, or the serve	r is down. (400)"	acuve	
	IMAGE_LOAD	Androia	12	4.44.0	27201817		Foreground	
P	IMAGE_LOAD		15.2	4,43,0		id	active	
D.	IMAGE_LOAD		15.2	4.43.0		id	active	0 🔹

Gambar 3.12 Testing Teknikal di Multihead Search

Gambar 3.12 merupakan hasil *testing* dari Multihead search, Multihead search akan meng*log* semua aktivitas *eagleeye* dari semua aplikasi *debug* yang digunakan. Diharuskan untuk mencocokan *code* dan *message* yang ter *log* di Charles Proxy dan Multihead search sesuai dengan *error screens* yang terjadi di aplikasi. Diberikan domain perusahaan untuk mengecek aktivitas *eagleeye* yang terjadi pada setiap aplikasi yang akan bermunculan di Multihead search, untuk mengecek *log* aktivitas *testing* dari sisi teknikalnya. Ketika sudah mengecek dari sisi visual dan teknikal maka dapat memberi QA *sign off* yang memiliki arti bahwa tugas ini sudah dicek dan dapat dilanjutkan untuk *merge* yang akan ditambahkan kepada versi terbaru dari aplikasi.

4. Common Error Handling

Tugas selanjutnya yaitu common error handling dimana tugas QA adalah untuk mengecek apakah error screens yang terjadi sudah menggunakan komponen baru dan tetap memiliki *behaviour* yang sama dengan versi aplikasi sebelumnya meskipun menggunakan komponen yang baru sehingga tugas QA disini adalah untuk mengecek dua versi aplikasi yaitu sebelum dilakukan update dan sesudah dilakukan *update* pada sisi komponennya untukmemastikan bahwa behaviour yang dimiliki sama. Komponen yang baru juga mengalami perubahan kecil yaitu error screens tidak dapatditutup dengan menggunakan *sliding* sedangkan di *error screens* yang lama dapat ditutup dengan menggunakan *sliding*. Tugas common error handling dilakukan seperti biasa yaitu untuk memulai membuat spreadsheet yang akan digunakan sebagai sarana penyimpan informasi mengenai behaviour yang terjadi di setiap APIyang akan dilakukan *testing*. Setelah membuat *spreadsheet* maka tugas dapat dilanjutkan dengan tahapan selanjutnya yaitu untuk membuat subtask baru.

Gambar 3.13 QA Subtask untuk CEH vertical GT

Kemudian dapat membuat QA *subtask* untuk mengisi informasi seperti pada gambar 3.13 yaitu untuk memberi informasi *scope* yang dilakukan didalam *testing*, apa saja langkah-langkah yang diambil untuk melakukan *testing* serta *testcases* yang telah dibuat untuk melakukan *testing*. Informasi lainnya berupa QA *end date*, QA *start date*, dan QA *effort* untuk memberitahu *developer* berapa lama *testing* akan dilakukan oleh QA.

ibtas ks				+ 100% Don
NCT-3472	Migrate Error Handling Components in AirportTransferCheckoutFragment	=	RAM	DONE~
NCT-9479	Migrate Error Handling Components in AirportTransferPrit AirportTransferCheckoulFagment	=	nuw.	DONE
NCT-3474	Migrate Error Handling Components in AirportTransferChooseMapLocationFragment	=	RAM	DONE
NCT+3475	Migrate Error Handling Components in AirportTransferSearchFormActivity	=	RAM	DONE
NCT-3476	Migrate Error Handling Components in UIExt	=	nuw	DONE
NCT-3477	Migrate Error Handling Components in CarRentalAdditionalZoneAutoCompleteBottomSheetDialog	=	RAN	DONE
NCT-9479	Migrate Error Handling Components in CarRentalBaseAutoCompleteBottomSheetDialog	=	RAM	DONE
NCT-3479	Migrate Error Handling Components in CarRentalBaseLoadableActivity	=	num	DONE
NCT-3403	Migrate Error Handling Components in CarRentalBookingFormActivity	=	RAM	DONE
NCT+3464	Migrate Error Handling Components in CarRentalFinalBookingFormFragment	=	RAM	DONE
NCT-3405	Migrate Error Handling Components in CarRentalFirstBookingFormFragment	=	RAN	DONE
NCT-3406	Migrate Error Handling Components in CarRentalBaseBookingFormFragment	=	RAM	DONE
NCT+9407	Migrate Error Handling Components in CarRentalBaseChooseMapLocationFragment	=	RAM	DONE
NCT-3400	Migrate Error Handling Components in CarRentaIRentAreaInfoBottomSheetDialog	=	RAN	DONE
NCT-3409	Migrate Error Handling Components in CarRentalChangeSearchFormBottomSheetDialog	=	RAM	DONE
NCT+3490	Migrate Error Handling Components in CarRentalSearchResultActivity	=	nu	DONE
NCT+3491	Migrate Error Handling Components in CarRentalVendorCatalogueBottomSheetDialog	=	RAN	DONE
NCT-3504	Migrate Error Handling Components in BottomSheetExt	=	RAM	DONE
NCT-3505	Migrate Error Handling Components in UlExt - CarRental	=	nuw	DONE
NCT-3506	Migrate Error Handling Components in LocationProvider/lewController	=	RAN	DONE
NCT+3507	Migrate Error Handling Components in BottomSheetExt - Train	=	RAM	DONE
NCT-3500	Migrate Error Handling Components in ContactProviderManager - Train	=	nuw	DONE
NCT-3509	Migrate Error Handling Components in AirportTransferSearchResultActivity	=	RAM	DONE

Gambar 3.14 Tampilan Parent Task CEH

Gambar 3.14 merupakan tampilan bentuk *parent task* yang memiliki beberapa *subtask* didalamnya yang menjelaskan informasi berbedabeda yang harus dilakukan *testing*. Setelah sudah membuat QA *subtask*, maka langkah selanjutnya dapat dilakukan yaitu untuk melihat *parent task* dari proyek yang akan dilakukan *testing*. *Parent task* tersebut berisi dengan *subtask subtask* lainnya yang masingmasing *subtask* akan menjelaskan cara melakukan *testing* secara rinci sehingga QA dapat mereproduksi *error screens* yang akan di *testing*.



Gambar 3.15 Tampilan Subtask berisi Informasi

Gambar 3.15 merupakan tampilan informasi yang terdapat pada masing-masing *subtask*, diharuskan untuk mengikuti setiap *flow* yang ada untuk melakukan *testing*. *Testing* dapat diikuti dengan langkah-langkah yang telah diberikan oleh *developer*, beberapa *flow* mengharuskan untuk mengubah JSON text dari sebuah API untuk menghasilkan *error screens* yang dicari. Setelah *error screens* ditampilkan maka dapat melanjuti langkah selanjutnya yaitu untuk melakukan komparasi.



33 Peran Software Quality Assurance Intern ..., Patrick Kwak, Universitas Multimedia Nusantara



Gambar 3.16 Error Screens Versi Sebelum

Gambar 3.16 merupakan contoh *error screens* yang dihasilkan di aplikasi versi sebelumnya. Aplikasi versi sebelumnya memiliki beberapa *behaviour* yang harus dicatat seperti ketika menekan tombol *setting* dan menekan tombol *back* tetap akan menampilkan *error screens* sebelumnya atau menghilang. Setelah sudah mencatat setiap *behaviour* dari setiap API maka tugas akan dilanjutkan ke langkah berikutnya yaitu untuk mengecek pada aplikasi versi yang baru.





Gambar 3.17 Error Screens Versi Baru

Gambar 3.17 merupakan contoh error screens yang dihasilkan di aplikasi versi baru. Memang secara visual, error screens tidak berubah namun secara komponen berubah sehingga diperlukan pengecekan kepada behaviour apakah behaviour komponen baru tetap memiliki behaviour yang sama dengan komponen yang lama. Ketika terjadi behaviour yang berbeda maka bug harus dilaporkan didalam JIRA untuk memberitahu developer bahwa terjadi kesalahan di suatu API sehingga dapat dibenarkan oleh developer. Setelah semua API sudah dilakukan testing, maka QA dapat

memberikan *sign off* yang memiliki arti bahwa tugas ini sudah dicek dan dapat dilanjutkan untuk *merge* yang akan ditambahkan kepada versi terbaru dari aplikasi.

5. Design Task

Tugas berikutnya adalah tugas *design* untuk mengecek apakah hasil *container* yang dibuat oleh *developer* sudah sesuai dengan *tech spec* milik divisi *designer*. *Tech spec* diberikan kepada menggunakan Figma dimana didalam Figma tersebut, terdapat informasi mengenai spesifikasi yang digunakan untuk setiap *container*, sehingga dapat melakukan *testing* dengan menggunakan Figma tersebut sebagai referensi. *Testing* diperlukan untuk mengecek di bagian IOS sehingga dibutuhkan BrowserStack karena tidak memiliki *real device* IOS. Dapat mengunduh *build* yang sudah diberikan *developer* dan melakukan *install* kepada Browserstack untuk dilakukan *testing*. Tugas utama QA di tugas ini adalah untuk memastikan bahwa *container* dari *label* berubah bentuknya sesuai isi tulisan didalam *container* tersebut.



🗸 Standar	Premium
Perlindungan M & Rumah	Menginap
 Perlindungan ruma bencana alam, dll 50.000.000. 	ah akibat perampokan, . sampai IDR
 Kompensasi pemberingan kan kan kan kan kan kan kan kan kan k	batalan reservasi sampa er kamar per malam.
Lihat 8 benefit lainnya	

Gambar 3.18 Hasil Akhir Design Task

Gambar 3.18 merupakan hasil akhir perubahan *design* pada *container*, dapat dilihat di gambar 28 bahwa ketika *text* melebihi maka *container* akan beradaptasi melebarkan untuk dapat menampung semua *text* yang ada didalam *container* tersebut.

3.3 Kendala yang Ditemukan

Dalam pelaksanaan praktik kerja magang yang dilakukan di perusahaan tiket.com, dialami beberapa kendala yang telah menghambat untuk menyelesaikan pekerjaan dan tugasnya, kendala tersebut adalah sebagai berikut:

• Pada masa pelaksanaan tugas NEH, dialami beberapa kesulitan dalam mengoperasi *software* karena belum pernah menggunakan *software* tersebut.

- Pada masa pelaksanaan tugas NEH, dialami kesulitan untuk mengkonfigurasi Charles Proxy.
- Pada masa pelaksanaan tugas NEH, dialami beberapa kesulitan untuk berkomunikasi dengan *developer*.

3.4 Solusi atas Kendala yang Ditemukan

Dalam pelaksanaan praktik kerja magang yang dilakukan di perusahaan tiket.com, dirumuskan solusi atas kendala kendala yang dihadapi di perusahaan sebagai berikut:

- Memberanikan diri untuk bertanya kepada rekan setim yanglebih berpengalaman untuk ilmu dan informasi cara mengoperasikan *software*.
- Memberanikan diri untuk bertanya kepada rekan setim yangpernah mengkonfigurasi *software* tersebut.
- Memberanikan diri untuk berkomunikasi dengan pihak *developer* agar tugas lebih jelas dan spesifik.



38 Peran Software Quality Assurance Intern ..., Patrick Kwak, Universitas Multimedia Nusantara