

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Posisi pada kegiatan kerja magang adalah sebagai *Developer Intern* dalam divisi *Mobility* tim *Technical*. Supervisi pelaksanaan kegiatan magang ini adalah Bapak Perdana Bagas Tirta Kumbara, selaku *Team Leader* divisi *Mobility*. Koordinasi dilakukan secara *online* melalui *discord*, karena tim developer bekerja secara *WFH/hybrid*. *Microsoft Teams* juga dapat digunakan sebagai alternatif untuk melakukan meeting *online*. Tracking progres menggunakan tools *online* yaitu *Trello*. Tim developer melakukan *Weekly meeting* pada setiap hari Kamis untuk mengupdate pekerjaan selama seminggu dan diskusi lainnya. *Scrum meeting* tim developer dilakukan setiap pagi untuk meng-update progres pekerjaan setiap harinya.

3.2 Tugas yang Dilakukan

Tugas yang diberikan dalam kegiatan magang di PT Helios Informatika Nusantara adalah sebagai berikut.

1. Melakukan riset terhadap database *MongoDB* untuk mengetahui performa dan kapabilitas *MongoDB* dibanding database yang digunakan sebelumnya untuk dijadikan pertimbangan untuk migrasi.
2. Membuat *prototype* untuk aplikasi sesuai dengan keperluan client. Dalam pelaksanaan magang ini merupakan pembuatan aplikasi keyboard.
3. Mengintegrasikan *AI ChatGPT* sebagai chatbot, melalui *Whatsapp*.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan pada Tabel 3.1.

Tabel 3.1. Pekerjaan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Mengeksplorasi tentang database MongoDB
2	Mencoba interaksi dengan database MongoDB
3	Menguji <i>benchmark</i> database MongoDB dengan JMeter
4	Membuat flowchart untuk suatu sistem aplikasi
5	Mengeksplorasi untuk membuat keyboard Android
6	Mengubah <i>styling</i> keyboard Android
7	Membuat <i>request network</i> dalam keyboard Android
8	- Membuat list menu dalam keyboard Android - Men- <i>share</i> invoice dari dalam keyboard
9	- Mempelajari React Native - Membuat keyboard dari nol dengan React Native
10	Membuat <i>key</i> pada keyboard
11	Membuat <i>special key</i> (selain huruf/angka) pada keyboard
12	Membuat list menu dalam keyboard
13	Membuat konfigurasi untuk keyboard
14	Fix bug dalam keyboard
15-17	Membuat <i>popup</i> ketika key ditekan
18	Membuat <i>alternative key</i> dalam <i>popup</i>
19	Mengeksplorasi mengenai Azure OpenAI
20	Mengeksplorasi mengenai Azure Cognitive Search
21	Mengeksplorasi mengenai Azure Cognitive Search
22	Mengeksplorasi pengaturan sumber data untuk Azure Search
23	Mengeksplorasi layanan AI chatbot lainnya
24	Mengeksplorasi <i>deployment</i> atau CI/CD dengan Azure
25	- Mengeksplorasi mengenai API Microsoft Teams - Implementasi/demo penggunaan Azure OpenAI

3.3.1 Riset Database MongoDB

A. Requirements

Requirement dari riset *database* MongoDB adalah untuk menguji kecepatan/performa C-R-U-D pada *database* MongoDB. Performa yang diharapkan adalah *throughput* hingga 3000 *request/s* untuk menggantikan *database* saat ini.

B. Hosting

Terdapat tiga hosting MongoDB yang digunakan, yaitu MongoDB Atlas, Azure CosmosDB for MongoDB, dan hosting lokal menggunakan laptop pribadi.

B.1 MongoDB Atlas

MongoDB Atlas merupakan hosting *database* MongoDB yang disediakan oleh perusahaan pembuat MongoDB sendiri. MongoDB Atlas mempunyai 3 jenis *pricing*, yaitu *Dedicated*, *Shared*, dan *Serverless*. Jenis *Shared* terdapat pilihan/*tier* berbayar dan ada yang gratis. Riset ini menggunakan jenis *Shared* yang tidak berbayar (gratis), dengan hosting AWS di Singapura (SG).

B.2 Azure CosmosDB for MongoDB

Berhubung Helios merupakan partner Microsoft/Azure, maka diutamakan untuk penggunaan Azure jika memungkinkan. Pada Azure, tidak terdapat hosting MongoDB sendiri namun terdapat *service* yang bernama Azure CosmosDB for MongoDB. Azure CosmosDB merupakan *database* yang dikembangkan Microsoft/Azure untuk NoSQL. Azure CosmosDB for MongoDB tetap menggunakan Azure CosmosDB untuk penyimpanan namun digunakan seperti MongoDB, artinya kode/*library* untuk MongoDB dapat digunakan pada Azure CosmosDB for MongoDB. Akan tetapi, ada fitur pada MongoDB yang tidak dapat digunakan pada Azure CosmosDB for MongoDB, seperti *watch/listener* tidak dapat mendeteksi ketika data di-*delete*.

Harga pada Azure CosmosDB for MongoDB dihitung berdasarkan hitungan *Request Unit/s* (RU/s). Hitungan RU berdasarkan pada lama komputasi (CPU), penggunaan memori/RAM, dan IO yang digunakan, sehingga setiap *read/write* menggunakan lebih dari 1 RU. Riset ini juga menggunakan tipe *Free* dengan 1000 RU/s, di-hosting di Singapura.

B.3 Self-Hosting

Serupa dengan *database* MySQL, MongoDB terdapat server *community edition* yang dapat digunakan secara gratis (di luar penggunaan komersil). Riset ini menggunakan laptop Dell Latitude 7490 untuk hosting lokal, dengan spesifikasi berikut.

1. Processor: Intel Core i7-8650U
2. RAM: 16GB

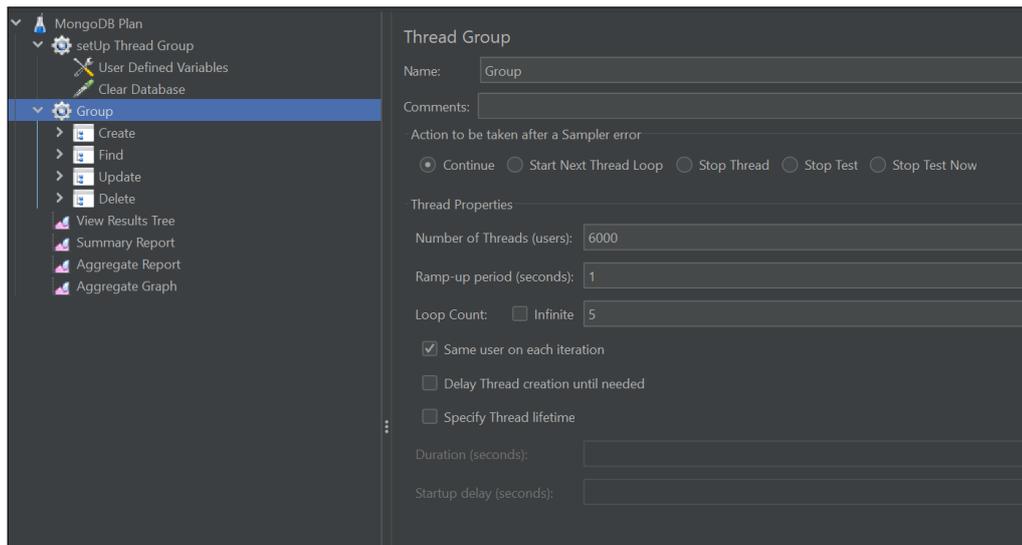
C. Benchmark

Untuk mengukur/benchmark ketiga hosting MongoDB, digunakan aplikasi Apache JMeter. Apache JMeter adalah aplikasi untuk mengukur performa aplikasi web. Untuk mengukur performa MongoDB dibuat aplikasi web server sederhana dengan menggunakan NodeJS Express. Pada server tersebut terdapat API Endpoint untuk *create*, *delete*, *find (read)*, dan *update database* MongoDB.

Setup JMeter dapat dilihat pada Gambar 3.1 dengan rincian sebagai berikut.

1. *setUp Group* untuk konfigurasi variabel url *database*, serta mengosongkan isi *database*.
2. *Thread Group* dengan 6000 *user (thread)*, diulang sebanyak 5 iterasi untuk mensimulasikan 5 sesi.
3. Pada setiap sesi untuk setiap *user (thread)*, dilakukan *Create*, *Find*, *Update*, dan *Delete* sebanyak 2 kali.
4. Maka, jumlah panggilan setiap C-R-U-D adalah $6000 \times 5 \times 2 = 60000$

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



Gambar 3.1. Setup JMeter untuk benchmark

D. Hasil Benchmark

Benchmark menggunakan Apache JMeter menghasilkan report seperti pada gambar-gambar berikut. Gambar 3.2 menunjukkan hasil benchmark untuk MongoDB yang di host di laptop lokal dengan rata-rata 951.56 *req/s*, Gambar 3.3 menunjukkan hasil benchmark untuk MongoDB dengan Azure CosmosDB for MongoDB dengan rata-rata 103.89 *req/s*, dan Gambar 3.4 menunjukkan hasil benchmark untuk MongoDB pada MongoDB Atlas dengan rata-rata 126.29 *req/s*. Nilai tersebut adalah nilai kolom *Throughput* pada report hasil benchmark. Benchmark untuk Azure CosmosDB for MongoDB tidak berhasil jika menggunakan 6000 *user*, mungkin dikarenakan limitasi *free tier*, yang menyebabkan benchmark Azure hanya dilakukan menggunakan 1000 *user*, sehingga jumlah pemanggilan untuk benchmark Azure adalah 10000.

Berdasarkan hasil benchmark tersebut, hosting *free* pada Azure CosmosDB for MongoDB dan MongoDB Atlas, maupun *self-hosted*, dinilai kurang memadai untuk keperluan di Helios, sehingga untuk saat ini *database* tidak akan dilakukan migrasi ke MongoDB.

Statistics - Local Laptop															
Requests		Executions			Response Times (ms)							Throughput		Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent		
Total	240001	0	0.00%	6156.18	22	8433	6509.00	7040.00	7103.00	7197.99	951.56	474.14	111.05		
Clear Database	1	0	0.00%	34.00	34	34	34.00	34.00	34.00	34.00	29.41	7.18	3.48		
Create Local	60000	0	0.00%	6000.66	22	8433	6013.00	7096.00	7147.00	7491.00	277.27	65.50	32.49		
Delete Local	60000	0	0.00%	6768.78	3273	8350	6565.00	7433.00	7479.00	7531.00	242.01	237.38	28.36		
Find Local	60000	0	0.00%	5632.37	47	8233	5777.00	6784.00	6996.00	7055.00	261.86	73.35	30.18		
Update Local	60000	0	0.00%	6223.00	504	8384	6801.00	7058.00	7094.00	7175.98	247.45	122.71	29.00		

Gambar 3.2. Hasil Benchmark JMeter MongoDB di Laptop Lokal

Statistics - Azure 1000 RU/s (free)															
Requests		Executions			Response Times (ms)							Throughput		Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent		
Total	40001	0	0.00%	7748.74	41	13550	9135.00	11512.00	11712.00	12137.99	126.29	44.34	14.74		
Clear Database	1	0	0.00%	377.00	377	377	377.00	377.00	377.00	377.00	2.65	0.65	0.31		
Create Azure	10000	0	0.00%	7469.18	41	11754	7712.50	9293.00	9529.00	10156.00	37.08	8.73	4.35		
Delete Azure	10000	0	0.00%	10156.55	3222	12756	10053.00	11429.00	11608.00	11915.97	33.45	18.24	3.92		
Find Azure	10000	0	0.00%	4318.14	741	9695	3808.50	6666.90	7268.95	7942.00	36.01	10.05	4.15		
Update Azure	10000	0	0.00%	9051.81	2461	13550	10458.00	11870.00	11990.95	12342.99	33.94	11.70	3.98		

Gambar 3.3. Hasil Benchmark JMeter MongoDB di Azure CosmosDB

Statistics - MongoDB Atlas Shared (Free) AWS-SG															
Requests		Executions			Response Times (ms)							Throughput		Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent		
Total	240001	0	0.00%	56889.13	31	62426	57094.00	57955.00	58009.00	58083.99	103.89	54.21	12.12		
Clear Database	1	0	0.00%	59.00	59	59	59.00	59.00	59.00	59.00	16.95	4.14	2.00		
Create Atlas	60000	0	0.00%	54538.27	31	60986	57764.00	58130.00	58389.95	59948.00	30.49	7.20	3.57		
Delete Atlas	60000	0	0.00%	57395.44	53674	61086	57059.00	57751.00	57811.00	57905.00	27.51	29.04	3.22		
Find Atlas	60000	0	0.00%	57830.67	1140	62426	57787.00	61529.90	61748.95	62207.00	28.79	8.06	3.32		
Update Atlas	60000	0	0.00%	57793.07	56834	62337	57333.00	57968.00	58021.00	58200.00	27.47	14.16	3.22		

Gambar 3.4. Hasil Benchmark JMeter MongoDB di MongoDB Atlas

3.3.2 Prototype Android Keyboard

A. Requirements

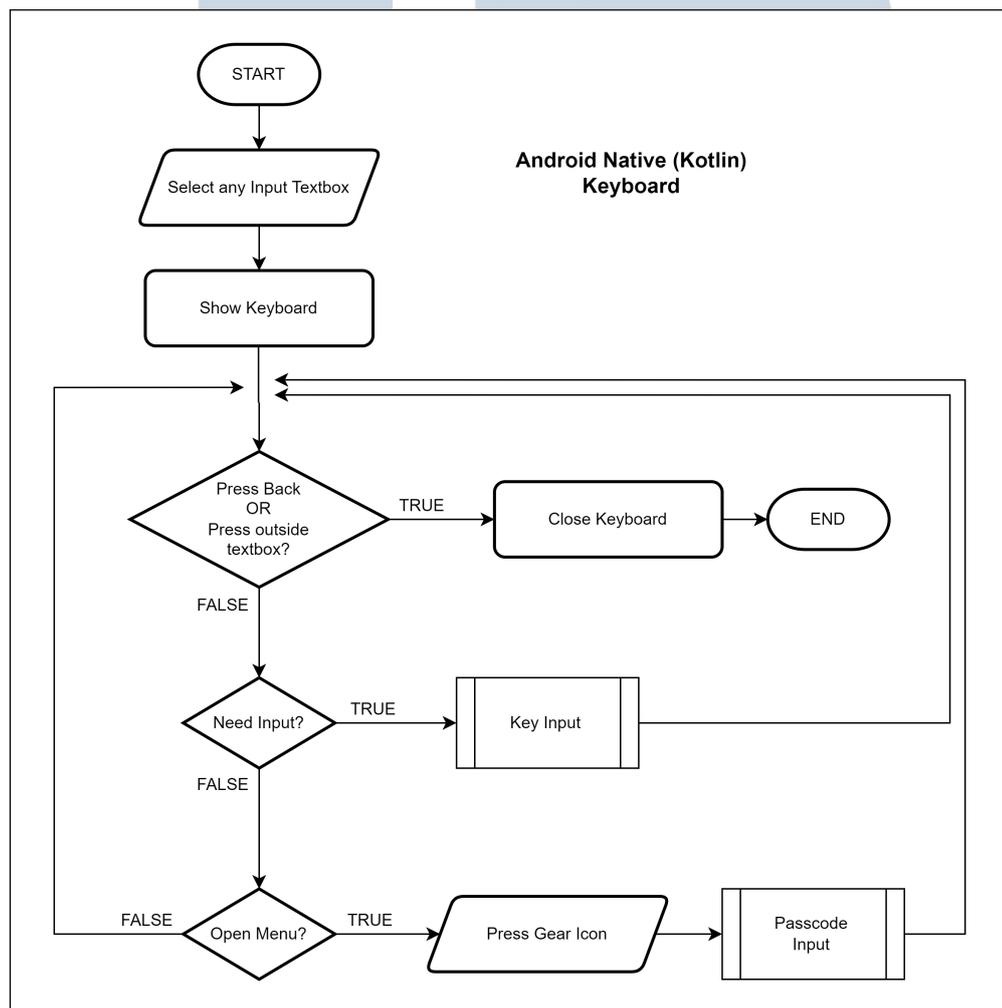
Requirement untuk *prototype* android keyboard adalah sebagai berikut.

1. *User* dapat mengetik dengan aplikasi keyboard yang dibuat.
2. *User* dapat menggunakan fitur lain melalui menu di dalam keyboard.
3. *User* dapat membagikan gambar dari fitur menu dalam keyboard.

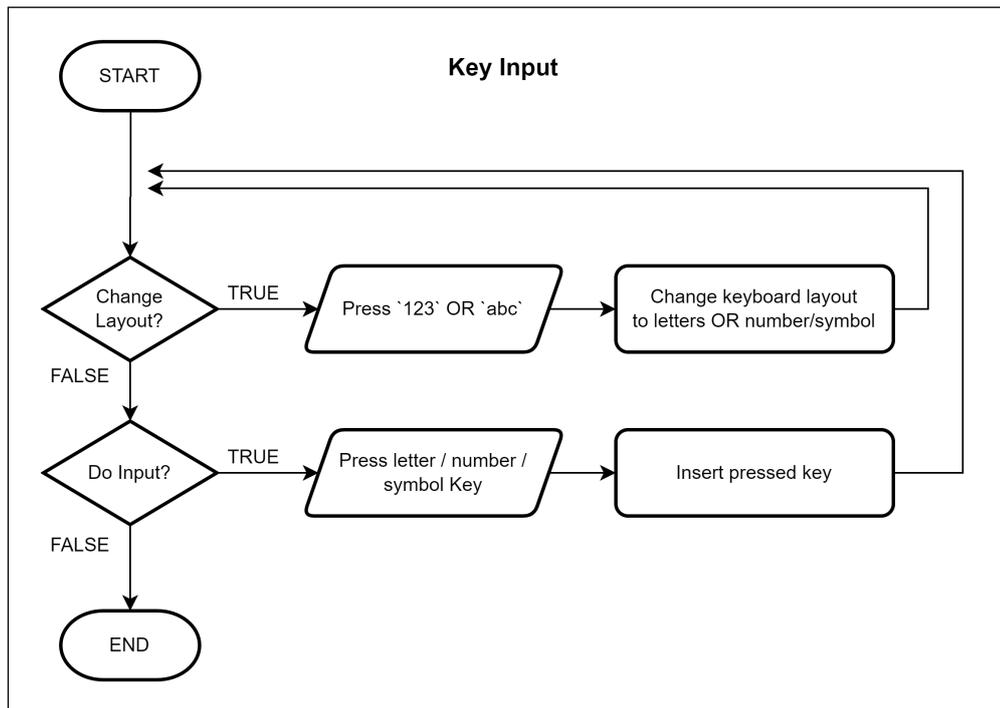
B. Android Native (Kotlin)

B.1 Flowchart

Alur utama dari keyboard dapat dilihat pada Gambar 3.5. Setelah membuka keyboard dengan menekan *textbox*, *user* dapat melakukan pengetikan ataupun membuka menu dengan tombol Gear. Jika *user* menekan *back* atau menekan di luar *textbox*, maka keyboard akan tertutup.



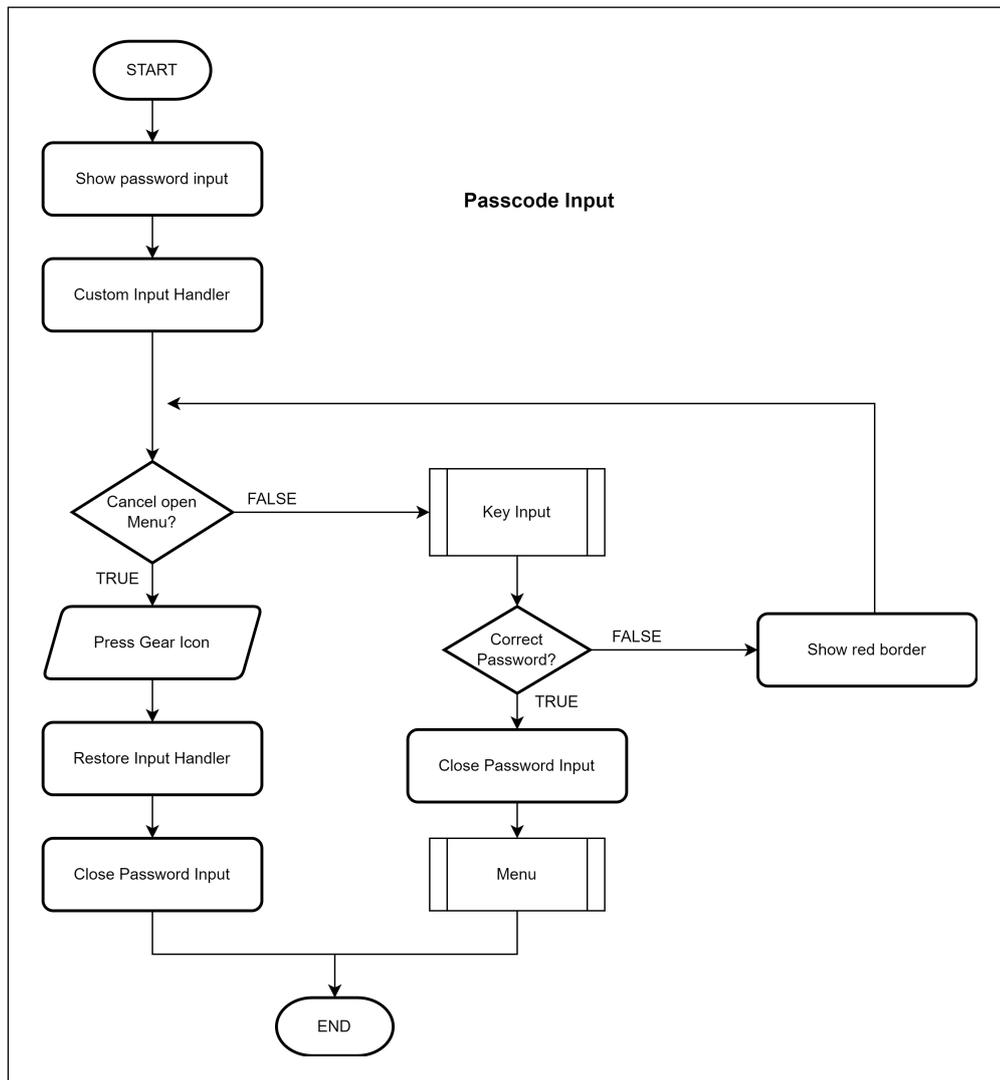
Gambar 3.5. Flowchart Keyboard Android (Kotlin)



Gambar 3.6. Flowchart Key Input (Keyboard Android/Kotlin)

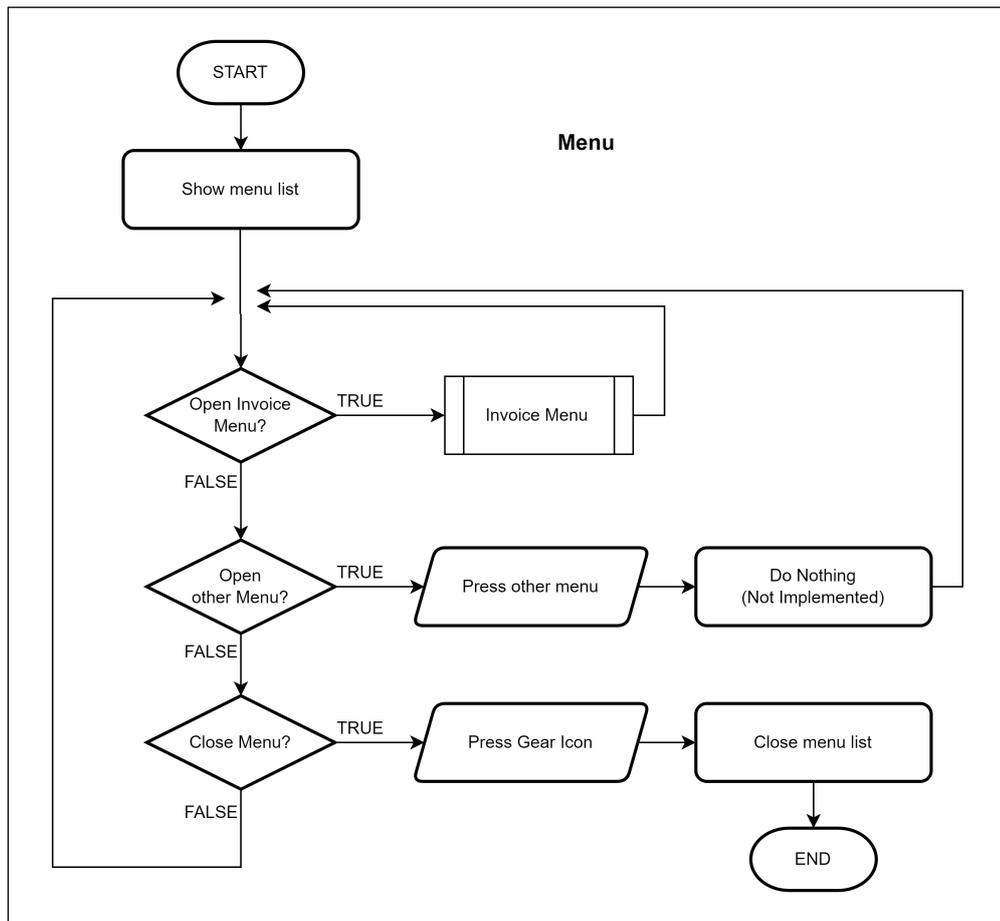
Gambar 3.6 menunjukkan proses untuk pengetikan. *User* dapat menekan tombol “123” untuk mengganti ke tampilan keyboard nomor dan simbol, atau tombol “abc” untuk kembali ke tampilan huruf. Jika *user* menekan huruf, nomor, atau simbol, maka akan dilakukan input sesuai dengan key yang ditekan, selanjutnya keyboard.





Gambar 3.7. Flowchart Passcode Input (Keyboard Android/Kotlin)

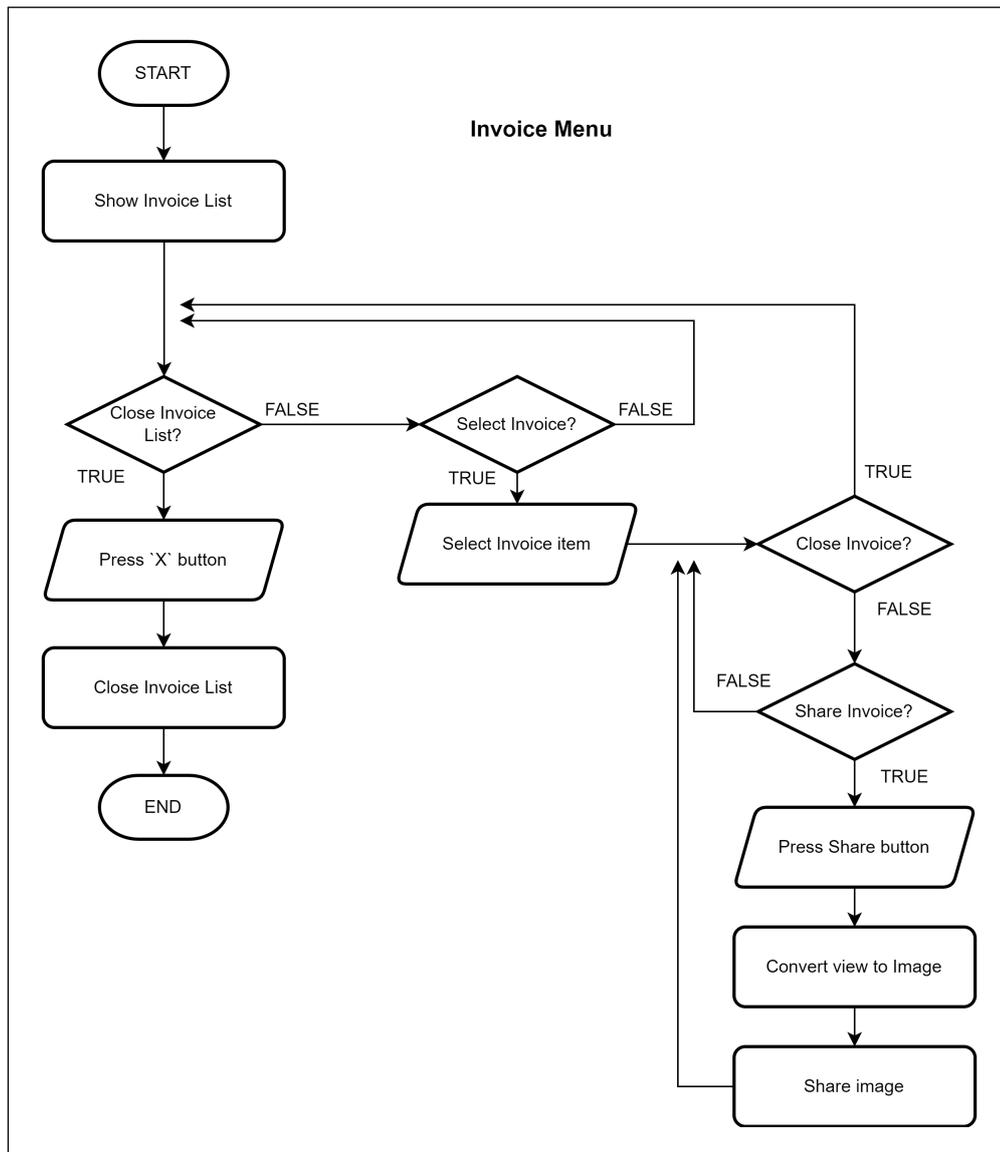
Jika *user* menekan tombol Gear, maka akan tampil kolom input untuk memasukkan password dahulu sebelum memasuki Menu. Ketika kolom input ditampilkan, *input handler* default pada keyboard diganti dengan *input handler custom* sehingga keyboard hanya akan melakukan input pada *textbox* password di dalam keyboard dan tidak akan mengisi *textbox* di luar keyboard. Flow untuk input password dapat dilihat pada Gambar 3.7.



Gambar 3.8. Flowchart Menu (Keyboard Android/Kotlin)

Setelah password berhasil, maka akan ditampilkan Menu pada keyboard. Untuk prototype ini, hanya ada menu Invoice, dan menu lainnya hanya sebagai *placeholder* yang tidak mempunyai fungsi. Flow untuk menu dapat dilihat pada Gambar 3.8.

U M M N
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA

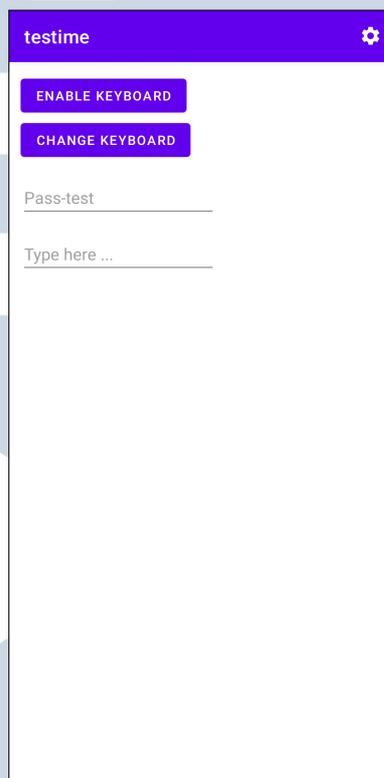


Gambar 3.9. Flowchart Menu Invoice (Keyboard Android/Kotlin)

Ketika menekan menu Invoice, akan ditampilkan daftar invoice yang ada. *User* dapat memilih invoice yang ingin dilihat. Setelah menampilkan invoice yang dipilih, *user* dapat membagikan invoice sebagai gambar melalui tombol Share. Ketika tombol Share ditekan, tampilan View untuk invoice akan dikonversi menjadi gambar, lalu dilakukan Share yang akan di-handle oleh Android. Flow untuk menu Invoice dapat dilihat pada Gambar 3.9

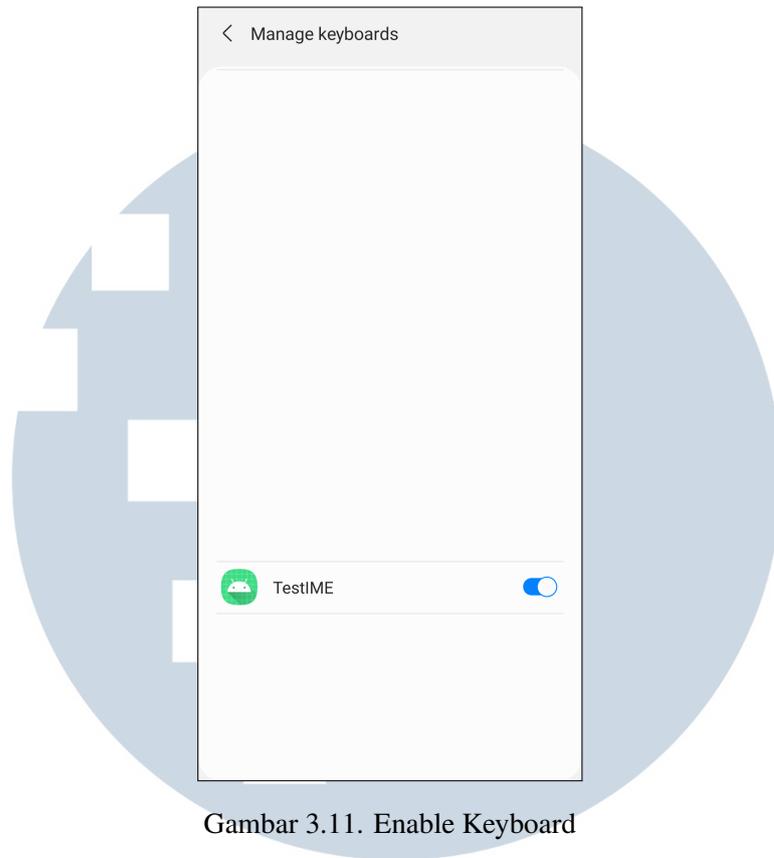
B.2 Hasil Prototype

Tampilan utama prototype dapat dilihat pada Gambar 3.10. Pada tampilan utama, terdapat 2 tombol untuk mengaktifkan keyboard serta mengganti keyboard. Keyboard diaktifkan dengan menekan tombol Enable Keyboard, lalu diaktifkan seperti pada Gambar 3.11. Kemudian untuk menggunakan keyboard, ditekan tombol Change Keyboard dan diaktifkan seperti pada Gambar 3.12.



Gambar 3.10. Tampilan Utama Prototype

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

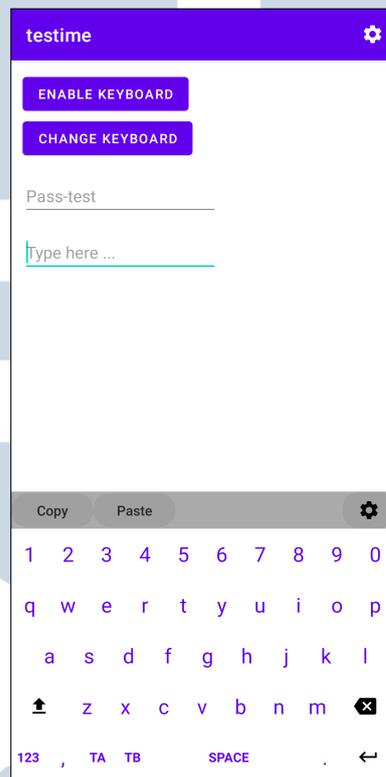


Gambar 3.11. Enable Keyboard



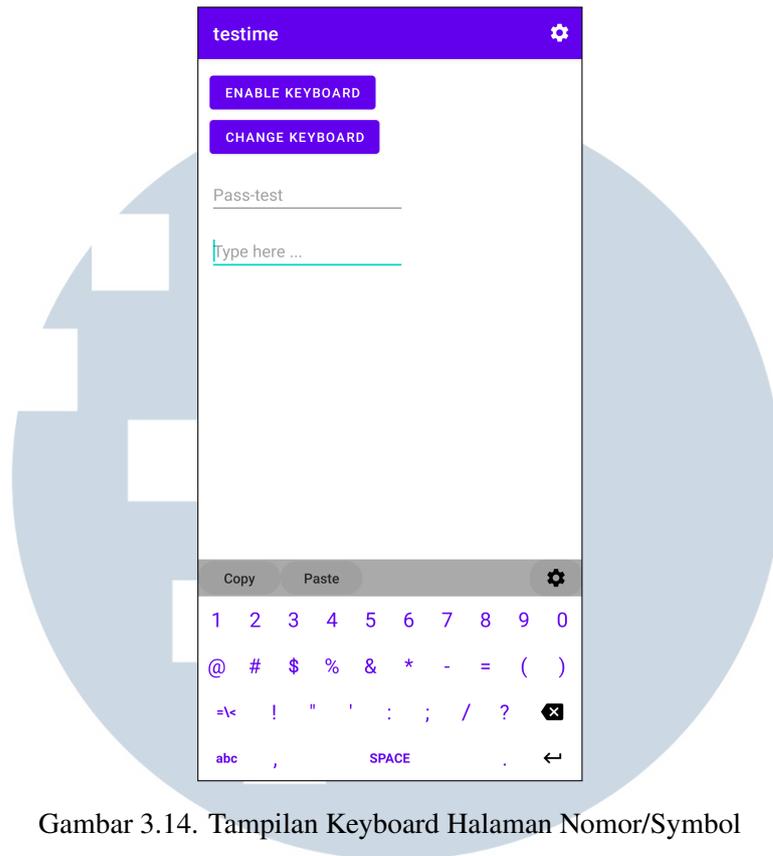
Gambar 3.12. Change Keyboard

Jika keyboard sudah aktif, maka ketika menekan kolom input teks, keyboard akan tampil seperti pada Gambar 3.13. Ketika tombol “123” ditekan, maka keyboard akan menampilkan halaman nomor dan simbol seperti pada Gambar 3.14. Tombol “abc” pada tampilan halaman nomor digunakan untuk kembali ke tampilan huruf.



Gambar 3.13. Tampilan Keyboard

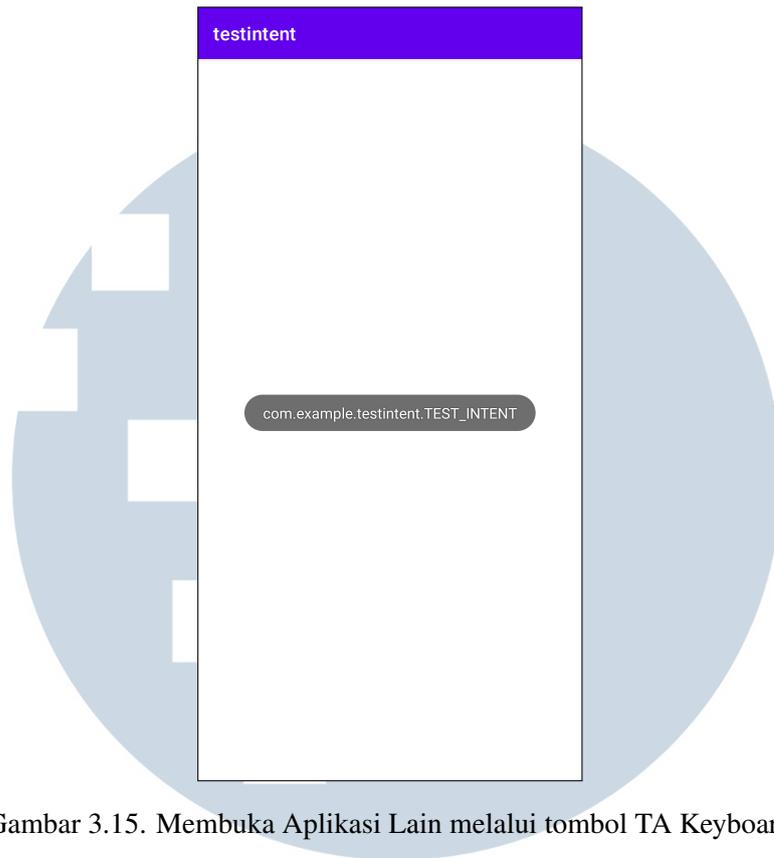
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



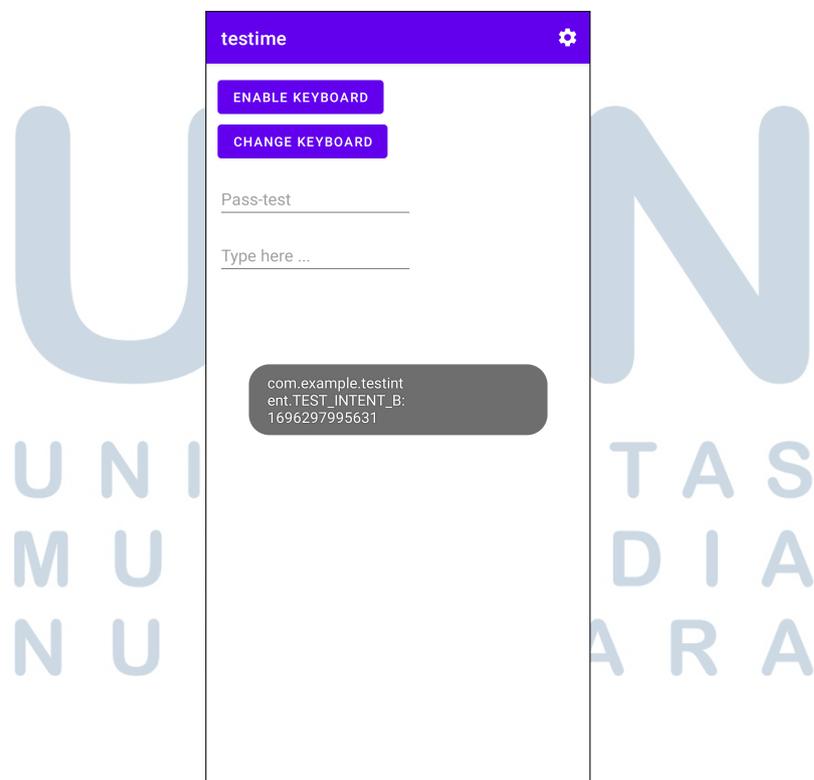
Gambar 3.14. Tampilan Keyboard Halaman Nomor/Symbol

Untuk menguji pembuatan tombol custom di keyboard serta membuka aplikasi lain, dibuat tombol TA dan TB. Tombol TA membuka aplikasi lain, terlihat pada Gambar 3.15. Tombol TB membuka aplikasi main/utama dari keyboard, terlihat pada Gambar 3.16.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

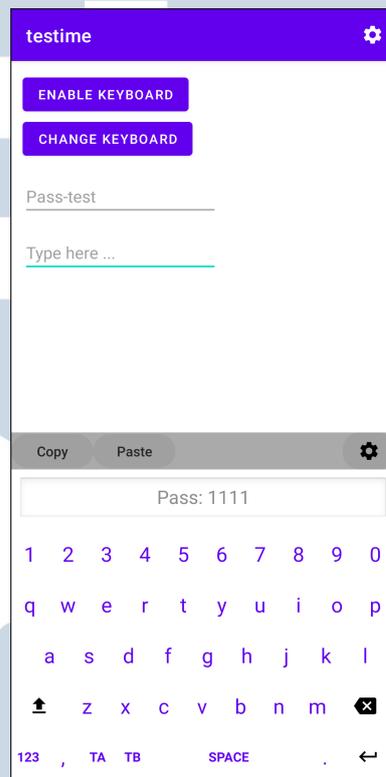


Gambar 3.15. Membuka Aplikasi Lain melalui tombol TA Keyboard



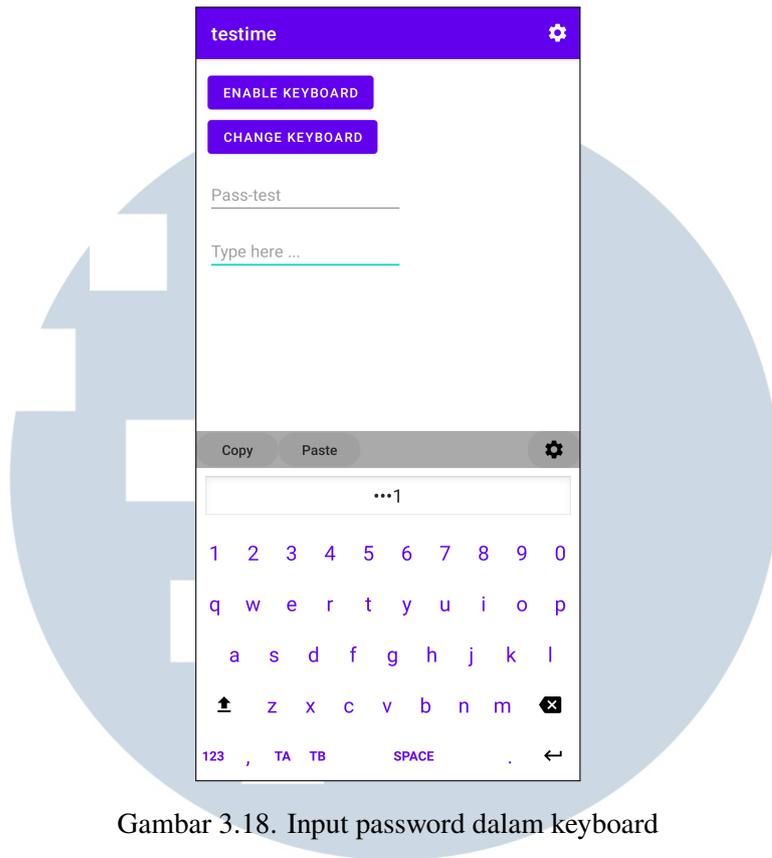
Gambar 3.16. Membuka Aplikasi Utama melalui tombol TB Keyboard

Pada bagian atas dari keyboard, terdapat tombol Copy, Paste, dan tombol berbentuk Gear. Tombol Copy dan Paste hanya untuk menguji fungsional tombol pada *top bar* keyboard. Tombol Gear digunakan untuk membuka menu pada keyboard. Namun sebelum masuk ke menu, diperlukan untuk memasukkan password, berhubung client bergerak di bidang perbankan, fitur ini mungkin memerlukan password seperti demikian. Tampilan input ketika tombol Gear ditekan dapat dilihat pada Gambar 3.17.

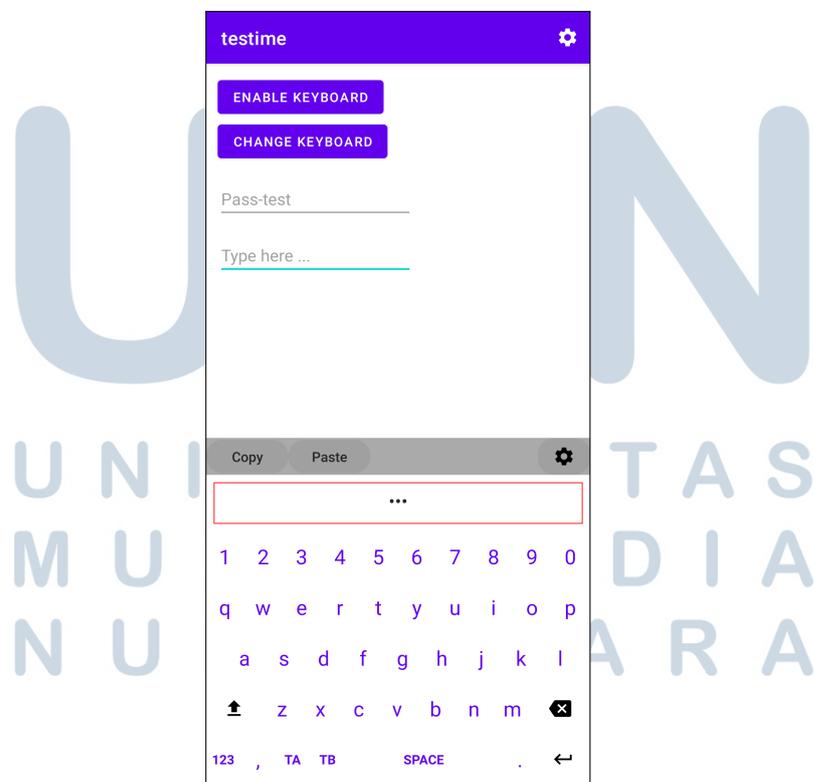


Gambar 3.17. Tampilan Keyboard ketika tombol Gear ditekan

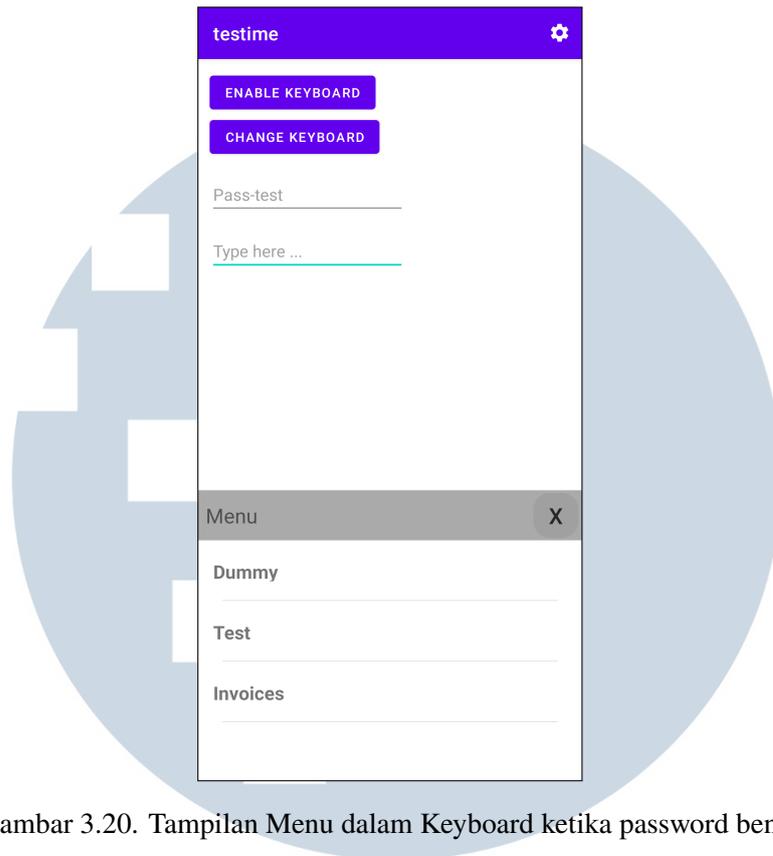
Tampilan ketika menginput password dalam keyboard dapat dilihat pada Gambar 3.18. Ketika password salah, *textbox* pada keyboard akan menampilkan garis merah, seperti pada Gambar 3.19. Jika password benar, maka akan ditampilkan menu seperti pada Gambar 3.20.



Gambar 3.18. Input password dalam keyboard



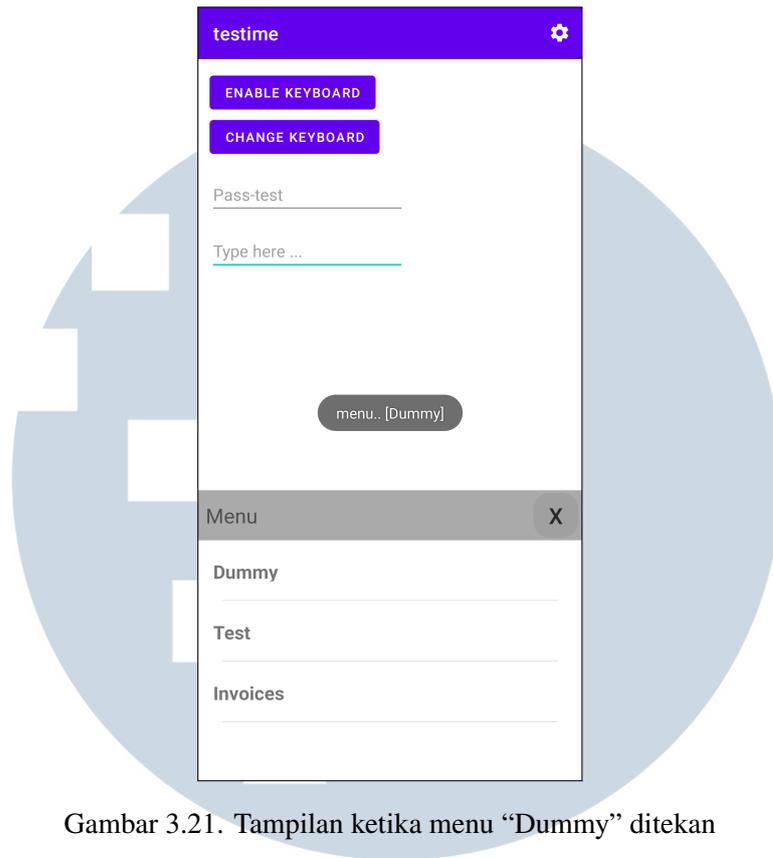
Gambar 3.19. Tampilan ketika password salah



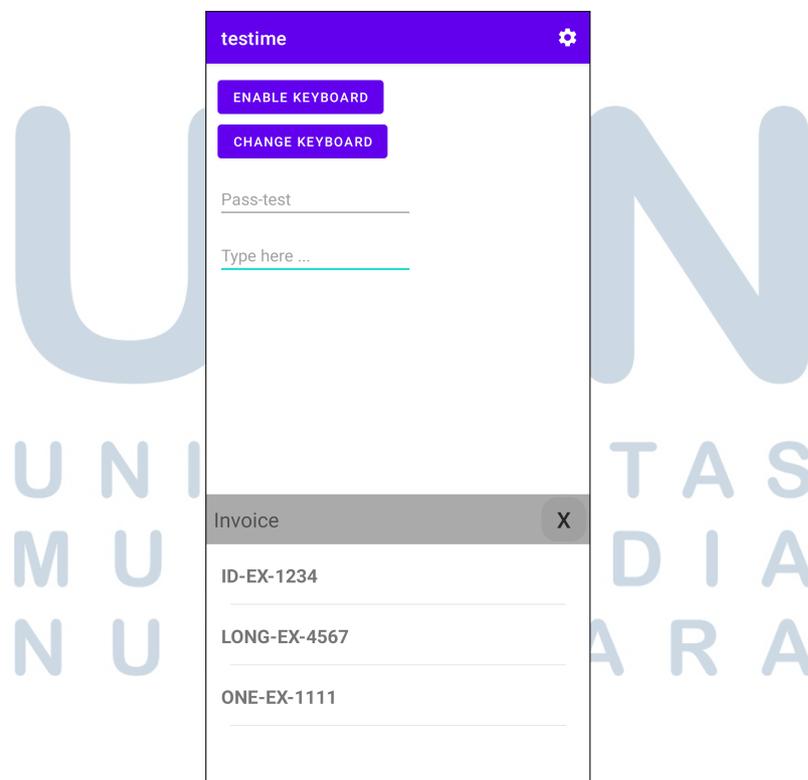
Gambar 3.20. Tampilan Menu dalam Keyboard ketika password benar

Tampilan menu terdapat 3 item, Dummy, Test, dan Invoice, serta tombol X untuk tutup menu atau kembali ke halaman menu sebelumnya. Menu Dummy dan Test hanya akan menampilkan *toast* ketika ditekan seperti pada Gambar 3.21. Menu Invoice akan membuka menu baru yang menampilkan list invoice, terlihat pada Gambar 3.22.

U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA

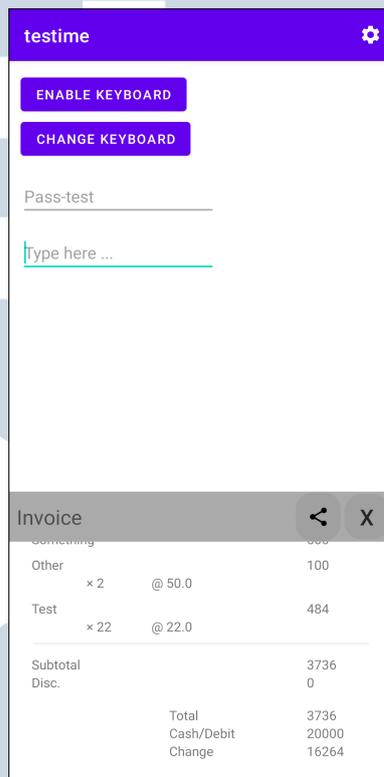


Gambar 3.21. Tampilan ketika menu “Dummy” ditekan



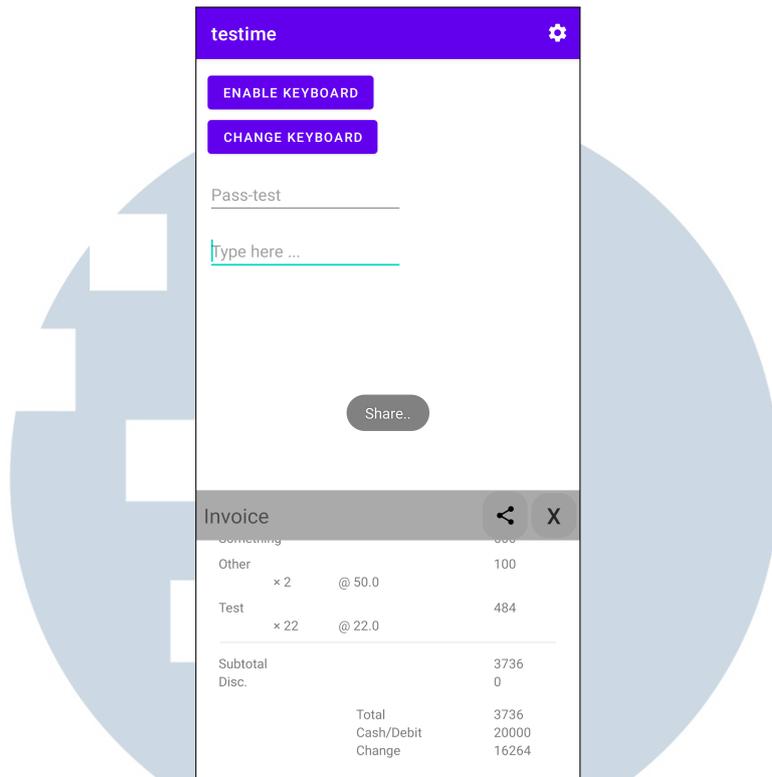
Gambar 3.22. Tampilan ketika menu Invoice ditekan

Ketika salah satu invoice ditekan, maka akan menampilkan contoh invoice seperti pada Gambar 3.23. Tampilan invoice bukan merupakan gambar, tetapi dibuat berdasarkan data menggunakan View. Dalam tampilan invoice, terdapat tombol Share dan exit untuk kembali ke menu. Tombol Share dapat digunakan untuk membagikan gambar dari view invoice yang terlihat. Gambar 3.24 menampilkan tampilan ketika tombol Share ditekan dan Gambar 3.25 menampilkan tampilan jika memilih untuk membagikan melalui Whatsapp.

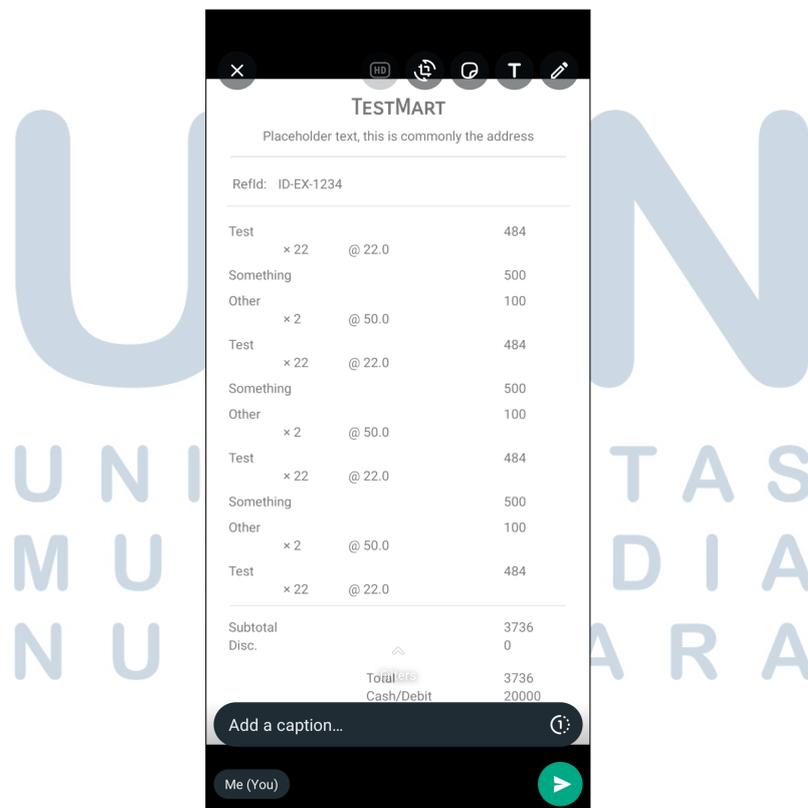


Gambar 3.23. Tampilan Invoice ketika ID-EX-1234 ditekan

UNIVERSITAS
MULTIMEDIA
NUSANTARA

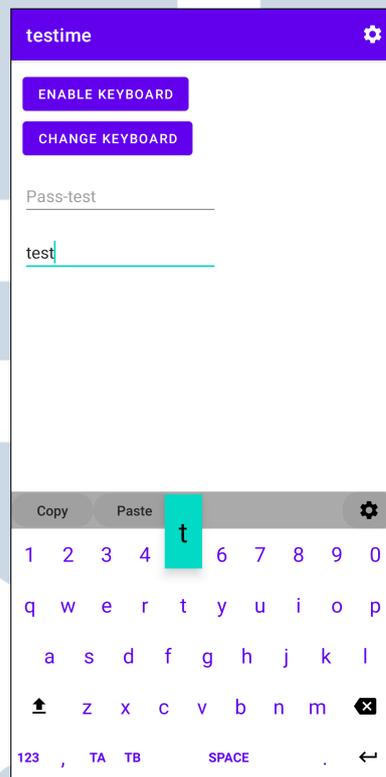


Gambar 3.24. Tampilan ketika tombol Share ditekan



Gambar 3.25. Tampilan ketika share ke Whatsapp

Tombol/key pada keyboard dapat menampilkan *popup* ketika ditekan seperti pada Gambar 3.26. Namun class `Keyboard` pada Android sudah *deprecated* dan tidak diperbaharui, sehingga tidak mudah untuk mengubah atau memperbaiki *style* dari keyboard, seperti popup ini yang tidak terpadu dengan keyboard.



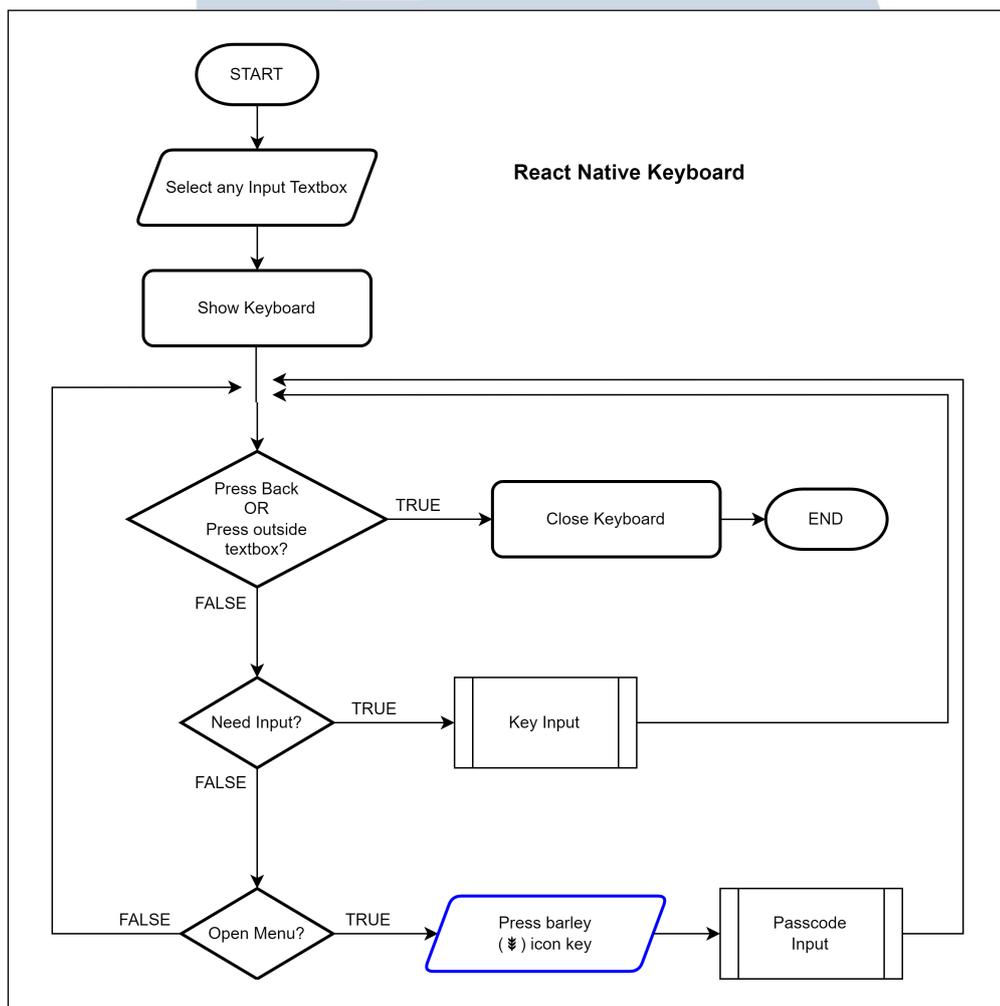
Gambar 3.26. Tampilan Popup pada Keyboard

C. Prototype Android Keyboard dengan React Native

Pembuatan keyboard Android dimulai dengan menggunakan Android Native/Kotlin. Kemudian, terdapat perubahan atau penambahan informasi dari client adalah bahwa aplikasi akan menggunakan React Native, sehingga prototype keyboard perlu dibuat ulang menggunakan React Native. Prototype keyboard dengan React Native dibuat dengan *requirement* yang sama. Pembuatan prototype keyboard dengan React Native memerlukan waktu yang lebih lama karena pada React Native tidak terdapat fitur/class resmi ataupun *library* untuk keyboard, sehingga diperlukan untuk membuat *management* untuk seluruh *state* keyboard, serta dilakukan *handling* untuk setiap sentuhan *user* secara manual.

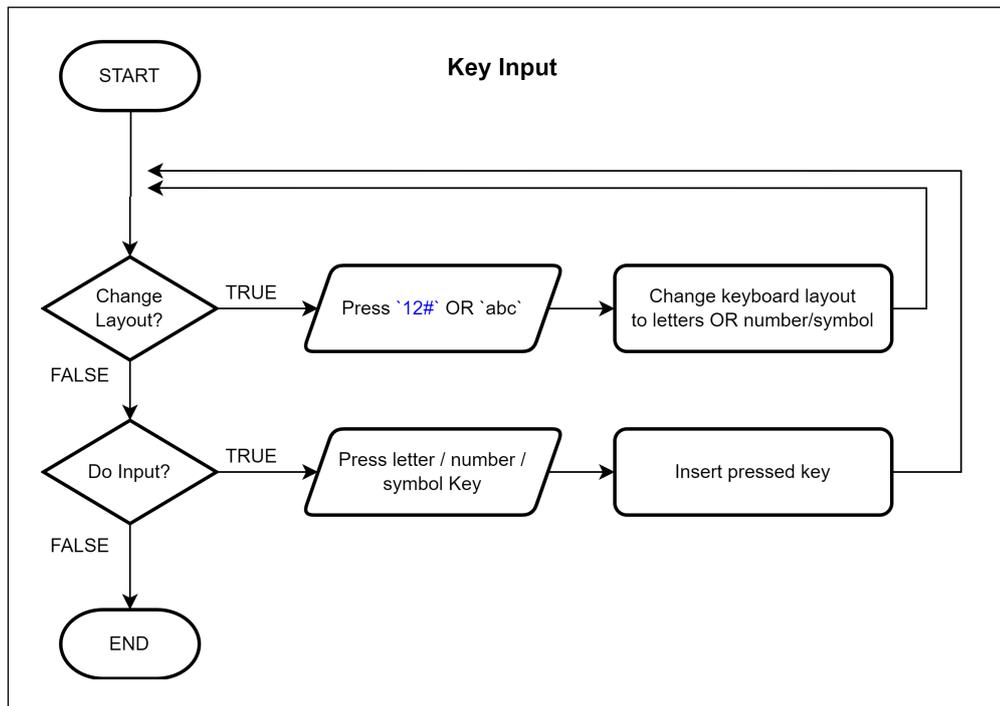
C.1 Flowchart

Flow dari keyboard React Native menyerupai flow dari keyboard Android (Kotlin), karena merupakan pembuatan ulang. Perubahan pada flow utama adalah proses untuk membuka Menu. Pada keyboard React Native ini, pembukaan Menu menggunakan key di dalam keyboard yang berbentuk padi di sebelah kanan tombol “12#”. Flow utama ini dapat dilihat pada Gambar 3.27.



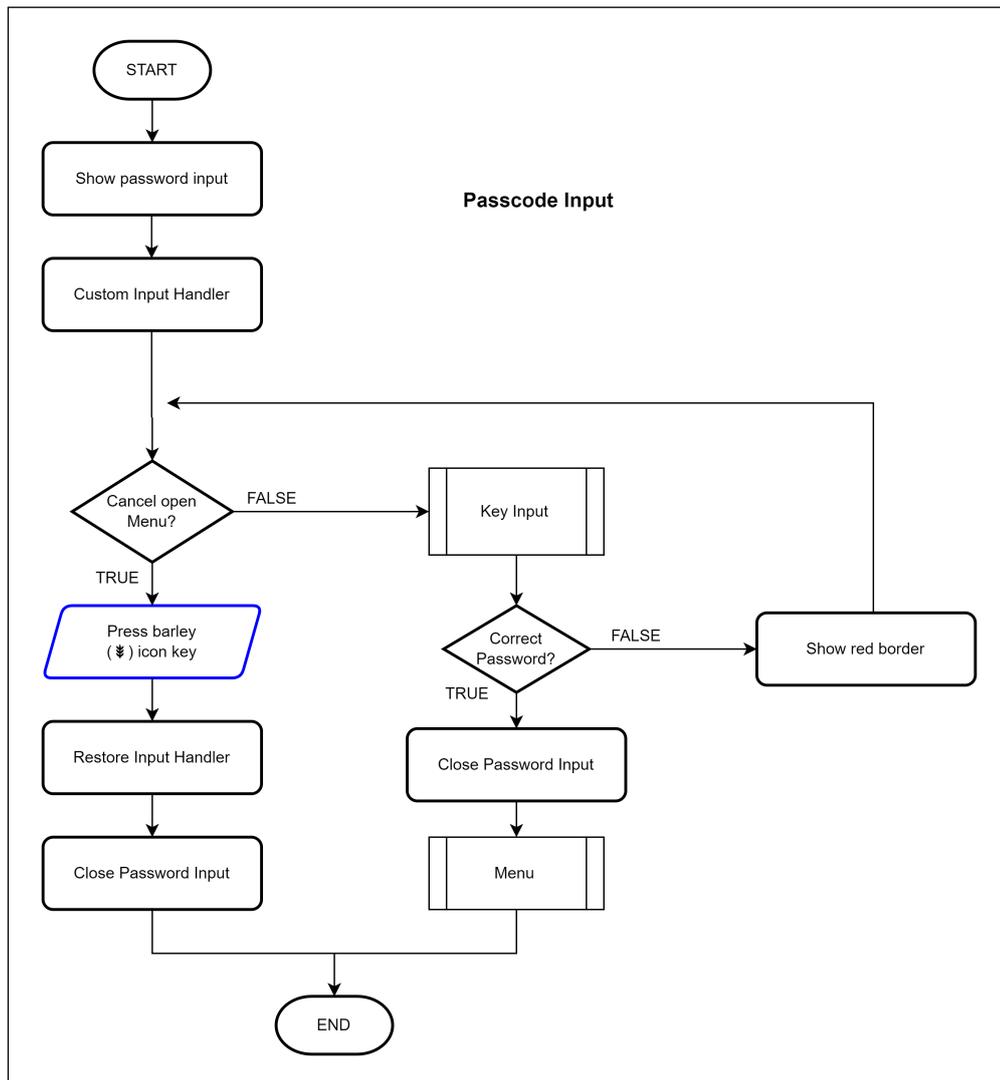
Gambar 3.27. Flowchart Keyboard React Native

Flow untuk proses pengetikan juga hanya terdapat perubahan tulisan dari “123” menjadi “12#” untuk mengubah tampilan keyboard. Flow untuk proses pengetikan dapat dilihat pada Gambar 3.28



Gambar 3.28. Flowchart Key Input (Keyboard React Native)

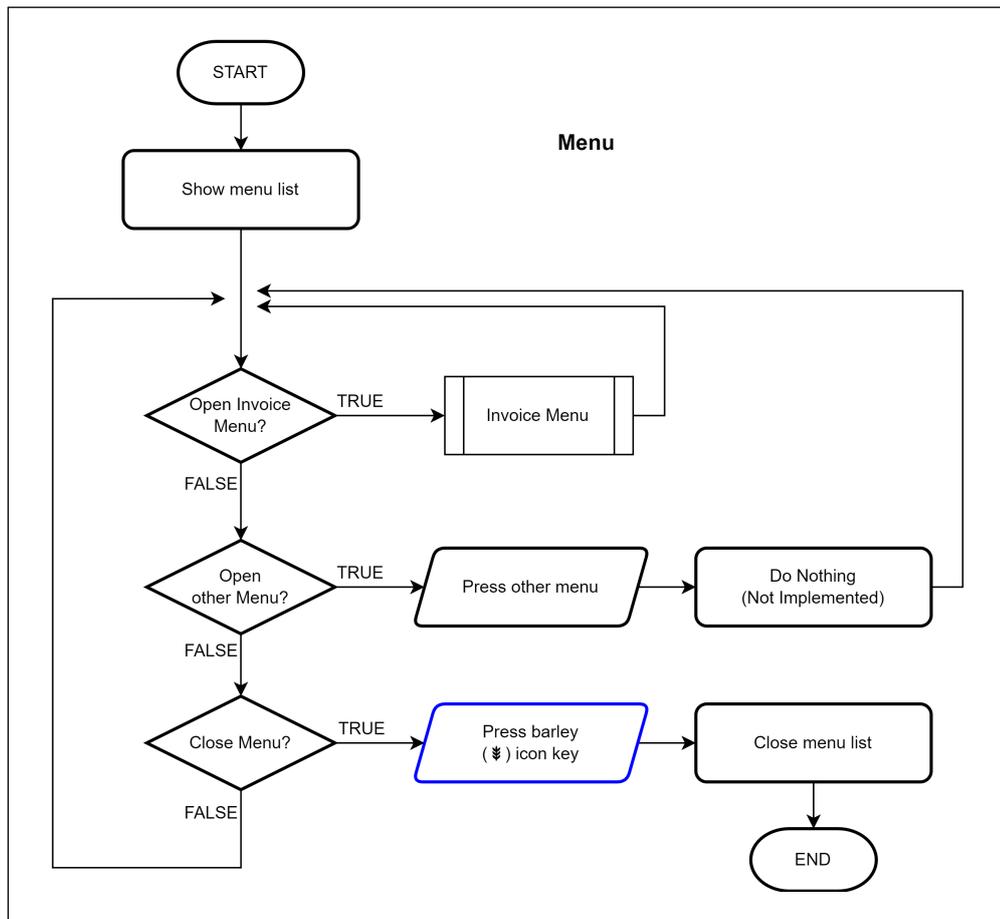
UMMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA



Gambar 3.29. Flowchart Passcode Input (Keyboard React Native)

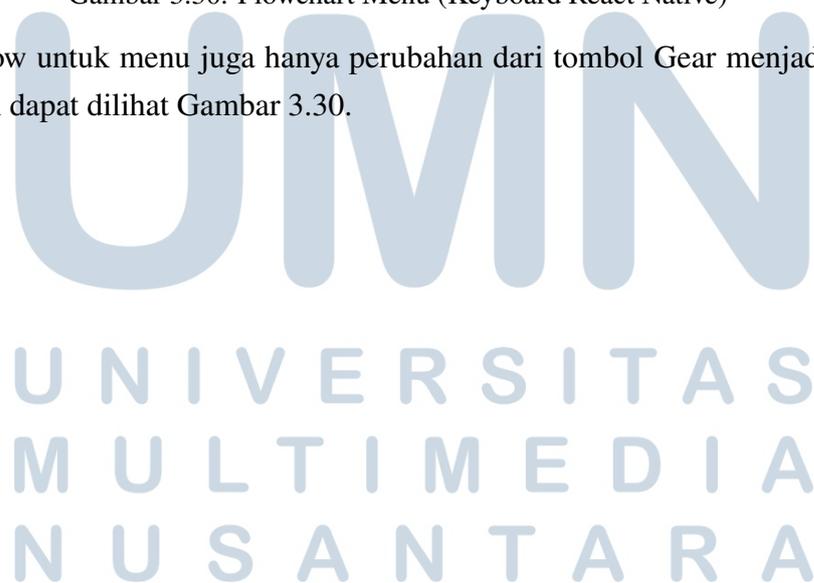
Flow untuk input password juga hanya perubahan tombol, yaitu dari tombol Gear menjadi tombol *barley* atau serupa dengan gambar padi. Flow input password dapat dilihat pada Gambar 3.29 berikut.

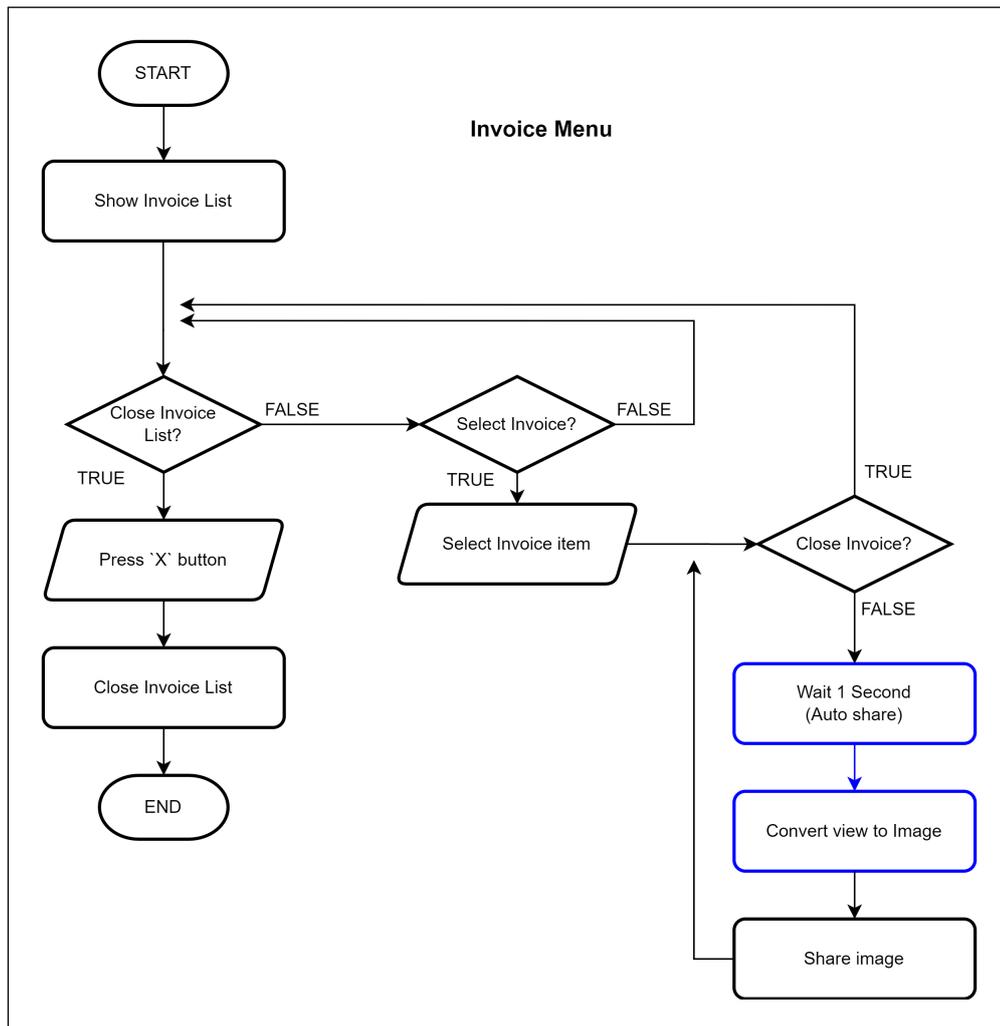
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.30. Flowchart Menu (Keyboard React Native)

Flow untuk menu juga hanya perubahan dari tombol Gear menjadi tombol *barley* dan dapat dilihat Gambar 3.30.





Gambar 3.31. Flowchart Menu Invoice (Keyboard React Native)

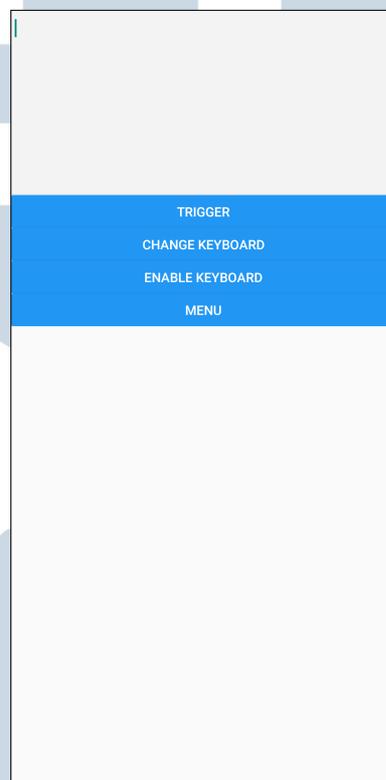
Untuk melakukan share Invoice, terdapat sedikit perubahan untuk share yang tidak lagi memerlukan tekan tombol Share, tetapi otomatis akan share setelah 1 detik. Flow untuk menu Invoice menjadi seperti pada Gambar 3.31.

C.2 Hasil Prototype

Tampilan utama prototype dengan React Native dapat dilihat pada Gambar 3.32. Tampilan yang serupa dengan prototype Kotlin, yaitu terdapat *textbox* untuk mengetik dan 2 tombol untuk mengaktifkan dan mengganti keyboard. Tombol Enable Keyboard akan membuka setting Android, untuk mengaktifkan keyboard seperti pada Gambar 3.36. Lalu, tombol Change Keyboard digunakan untuk menggunakan keyboard, seperti pada Gambar 3.34

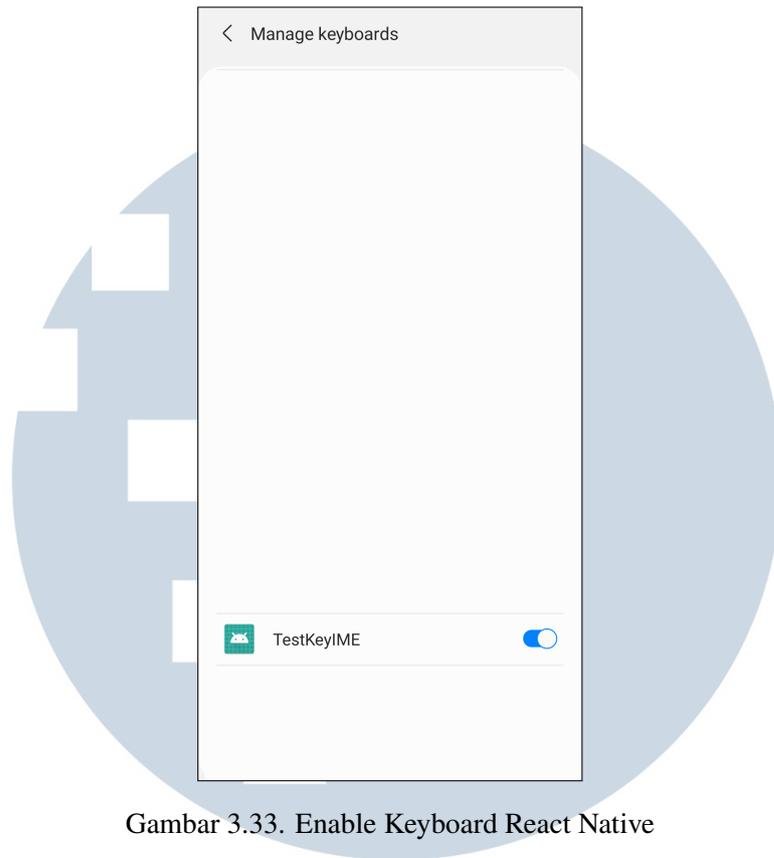
Pada prototype keyboard React Native, terdapat tambahan 2 tombol,

yaitu Trigger dan Menu. Tombol Trigger digunakan untuk melakukan *rerender* secara manual pada React Native, dikarenakan fitur *hot reload* terkadang tidak menyebabkan *rerender*. Tombol Menu digunakan untuk percobaan awal pembuatan menu, serta untuk mempercepat percobaan fitur Menu. Hal ini dikarenakan keyboard di dalam Android berupa Service, bukan Activity, sehingga fitur *hot reload* tidak dapat digunakan. Untuk *reload* keyboard, diperlukan untuk mengganti ke keyboard lain, kemudian kembali ke keyboard yang dibuat (TestKeyIme), sehingga memerlukan waktu lebih lama. Tampilan ketika tombol Menu ditekan dapat dilihat pada Gambar 3.35, menampilkan menu yang sama pada Gambar 3.42.



Gambar 3.32. Tampilan Utama Prototype dengan React Native

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.33. Enable Keyboard React Native



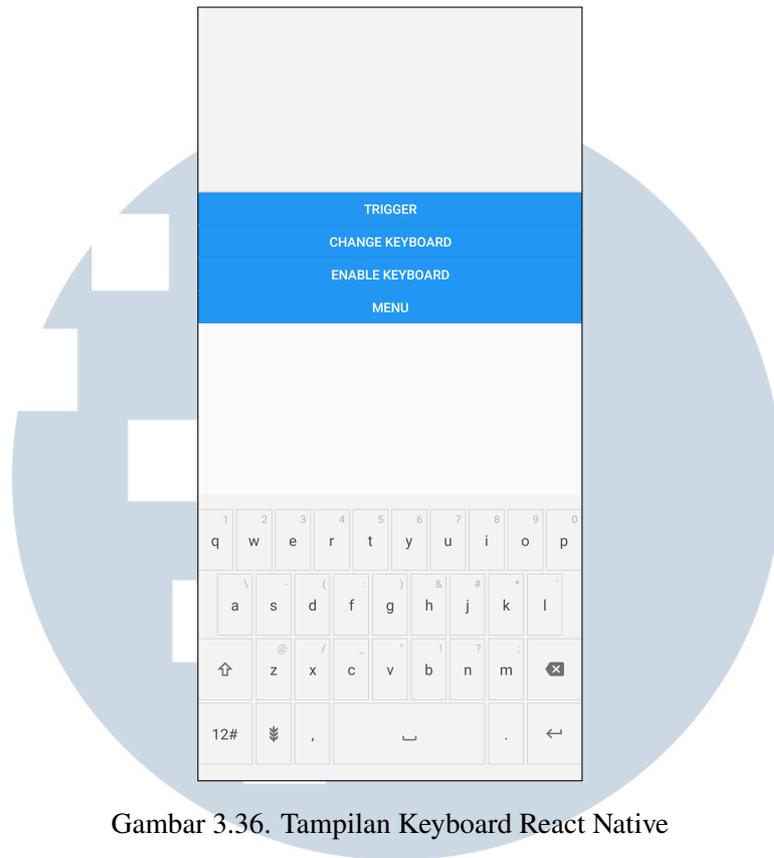
Gambar 3.34. Change Keyboard React Native



Gambar 3.35. Tampilan ketika tombol Menu ditekan

Ketika menekan kolom *textbox*, maka keyboard akan tampil seperti pada Gambar 3.36. Tombol “12#” digunakan untuk mengganti tampilan keyboard menjadi tampilan nomor dan simbol seperti pada Gambar 3.37. Untuk kembali ke tampilan huruf digunakan tombol “abc”. Tampilan simbol terdapat dua halaman sehingga terdapat tombol “1/2”, untuk mengganti halaman simbol, dapat dilihat halaman ke-dua pada Gambar 3.38. Pada halaman kedua terdapat tombol *emoji* tersenyum, untuk membuka halaman *emoji*. Namun fitur tersebut belum diimplementasikan karena tidak termasuk dalam *requirement*, dan diperlukan untuk mencari *library* untuk mengelola daftar emoji, sehingga tombol tersebut hanya akan menampilkan tulisan “Not Implemented” seperti pada Gambar 3.39

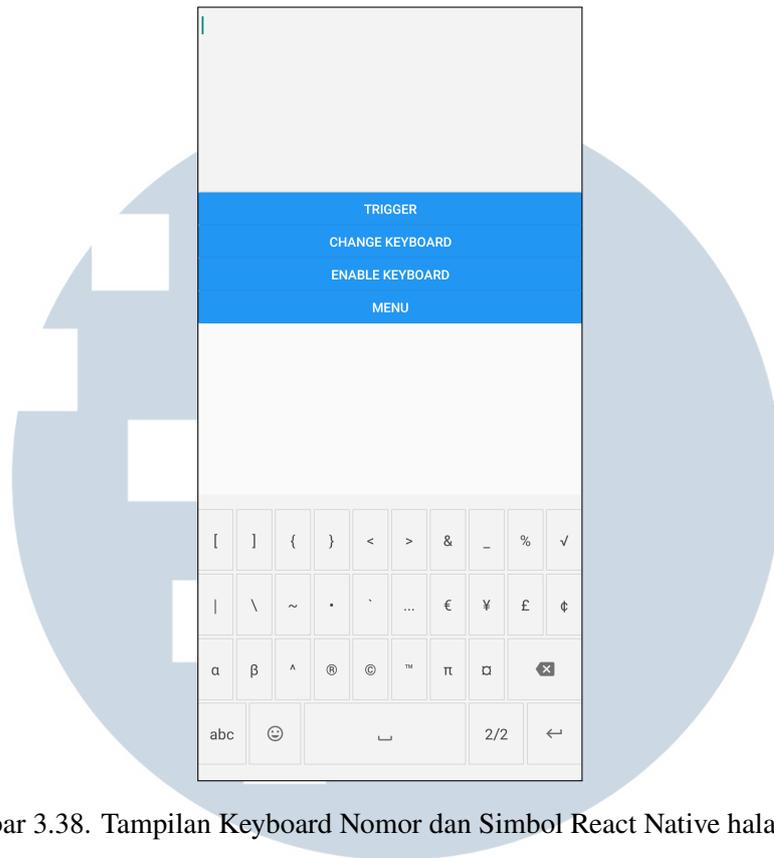
UNIVERSITAS
MULTIMEDIA
NUSANTARA



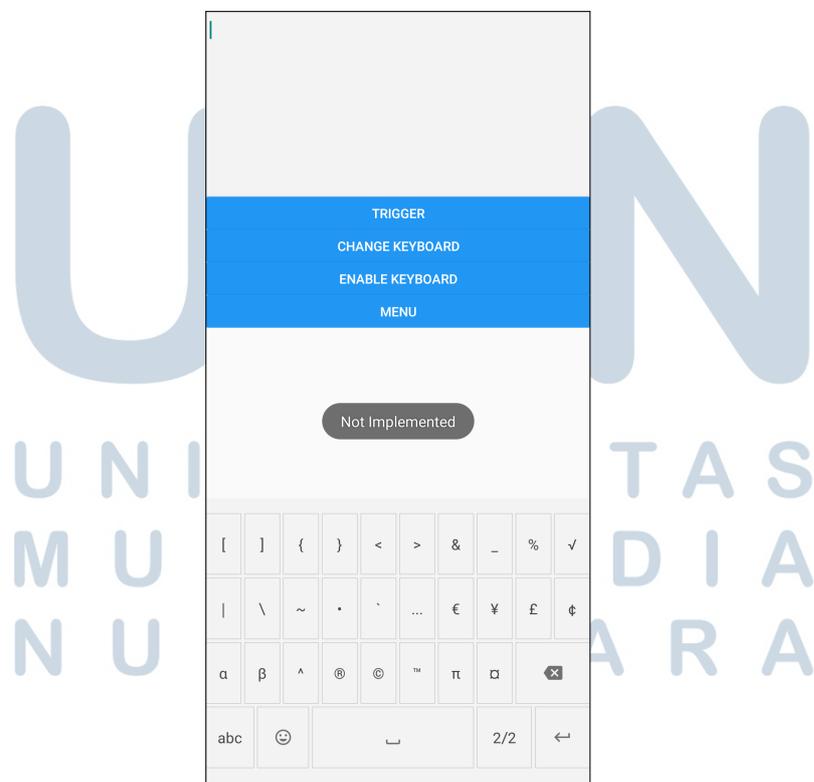
Gambar 3.36. Tampilan Keyboard React Native



Gambar 3.37. Tampilan Keyboard Nomor dan Simbol React Native halaman 1

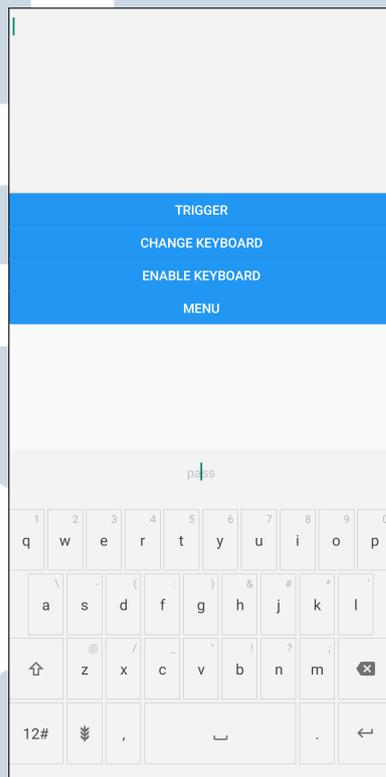


Gambar 3.38. Tampilan Keyboard Nomor dan Simbol React Native halaman 2



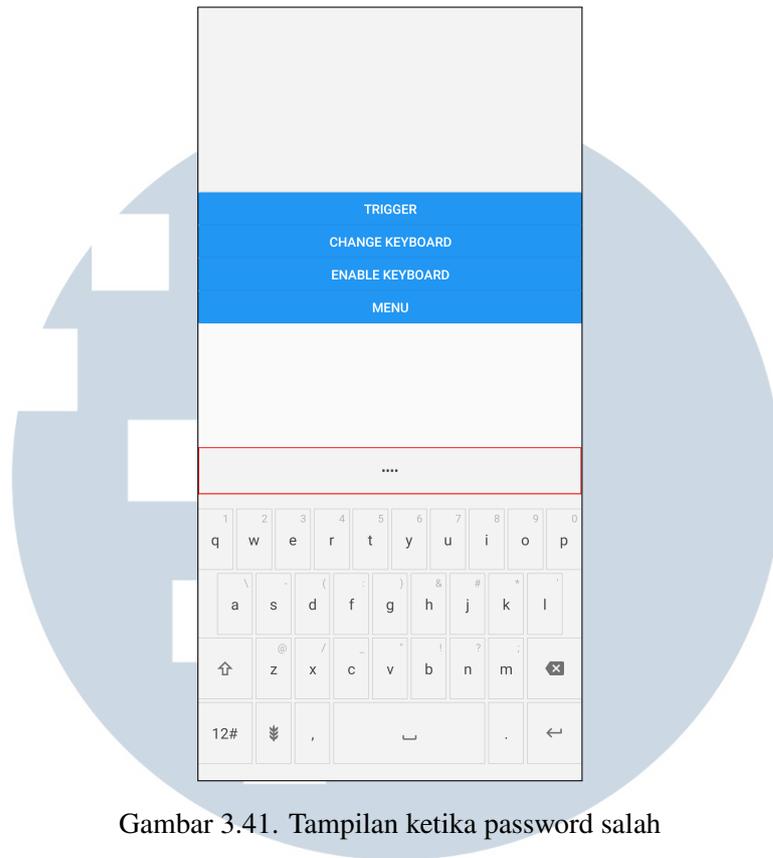
Gambar 3.39. Ketika tombol emoji ditekan

Tombol berbentuk padi di sebelah kanan tombol “12#” digunakan untuk membuka Menu pada keyboard. Namun, sebelum membuka menu harus memasukkan password, dengan tampilan seperti pada Gambar 3.40. Ketika password salah, *textbox* password akan menampilkan garis merah, terlihat pada Gambar 3.41. Jika password benar, maka akan ditampilkan menu yang dapat dilihat pada Gambar 3.42.

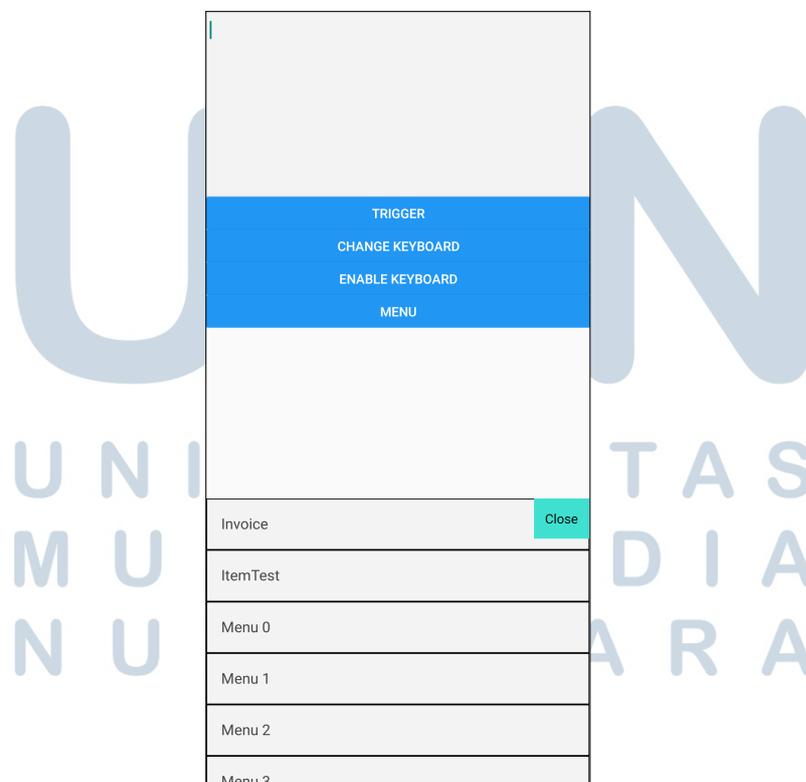


Gambar 3.40. Tampilan Input Password dalam Keyboard React Native

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

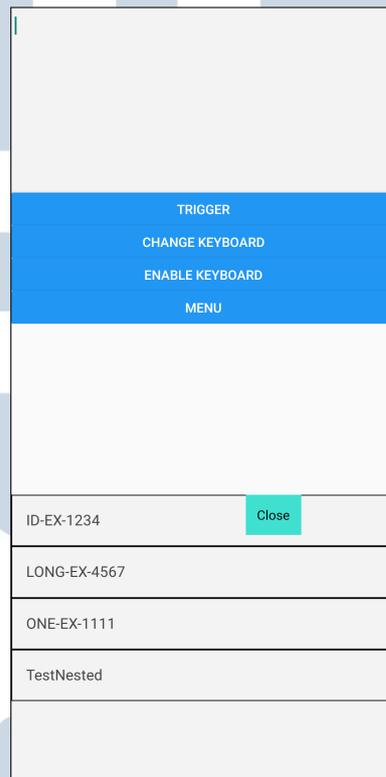


Gambar 3.41. Tampilan ketika password salah



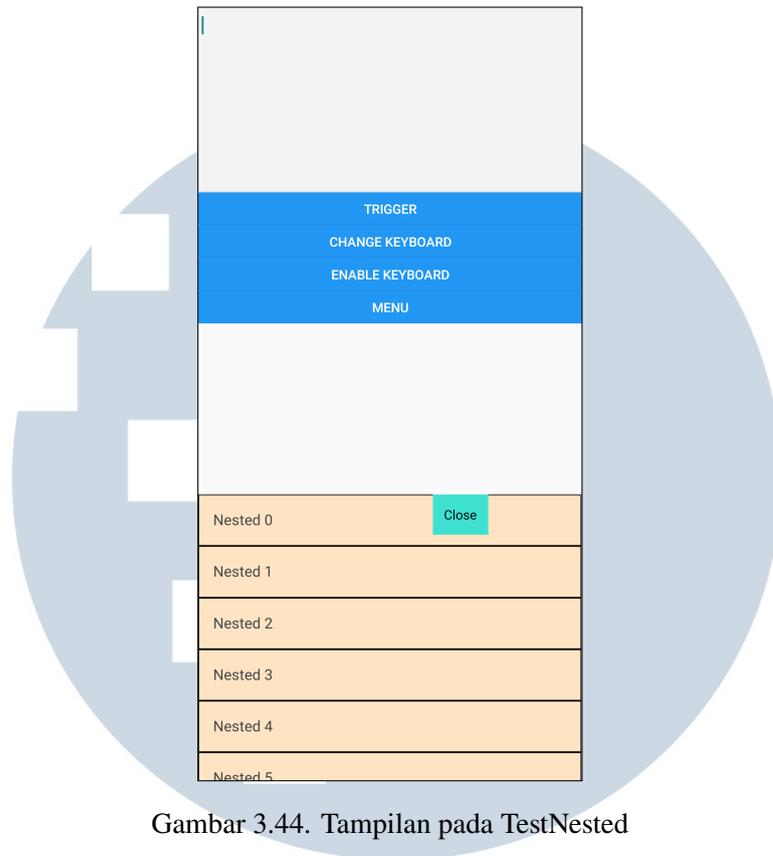
Gambar 3.42. Tampilan Menu dalam Keyboard React Native ketika password benar

Menu selain Invoice tidak mempunyai fungsi atau tampilan, karena hanya merupakan percobaan untuk *looping* pada React Native. Menu Invoice akan menampilkan menu seperti pada Gambar 3.43. Dalam menu Invoice, terdapat 3 invoice dan menu TestNested. Menu TestNested digunakan untuk percobaan kedalaman menu di dalam menu, dengan tampilan pada Gambar 3.44.



Gambar 3.43. Tampilan pada menu Invoice

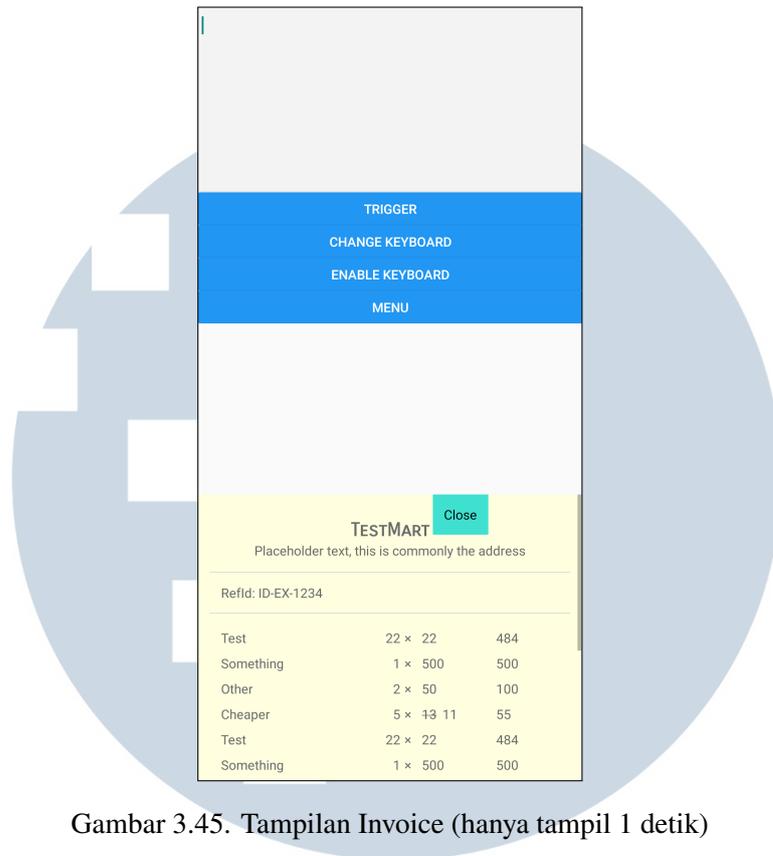
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.44. Tampilan pada TestNested

Ketika salah satu dari tiga invoice ditekan, maka akan ditampilkan invoice seperti pada gambar Gambar 3.45. Namun invoice hanya akan tampil 1 detik, sebelum secara otomatis akan melakukan share. Fitur share Android akan tampil, seperti pada Gambar 3.46. Jika melakukan share melalui Whatsapp, akan tampil gambar invoice seperti pada Gambar 3.47.

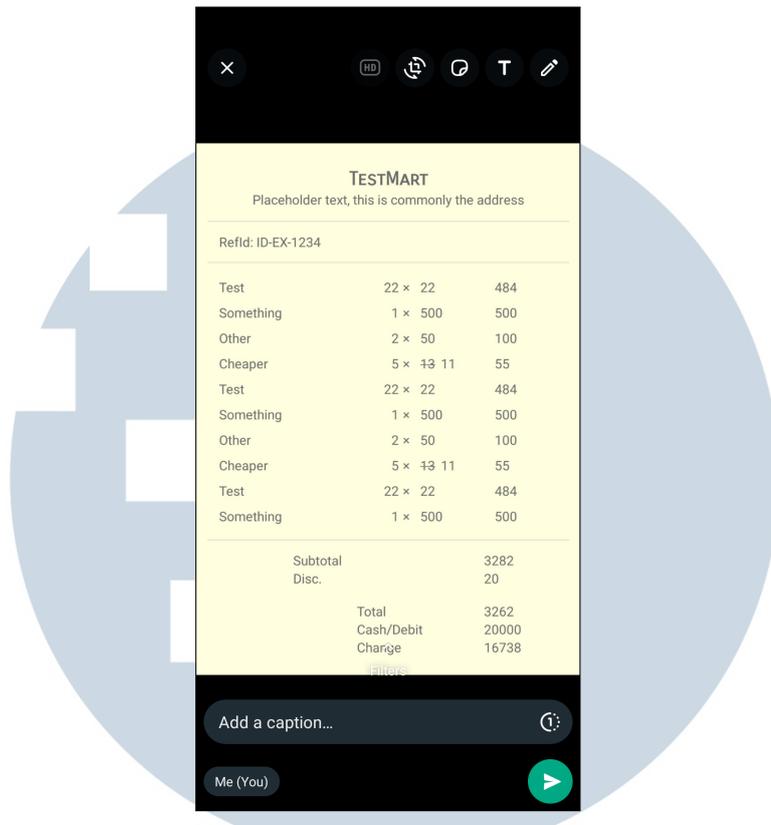
UMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA



Gambar 3.45. Tampilan Invoice (hanya tampil 1 detik)



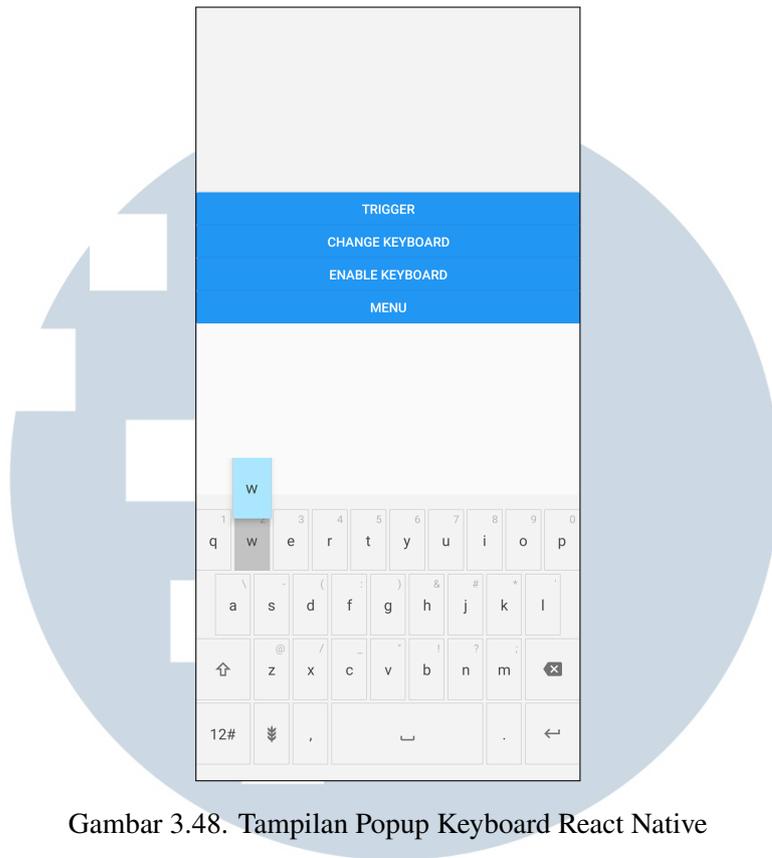
Gambar 3.46. Tampilan share gambar invoice



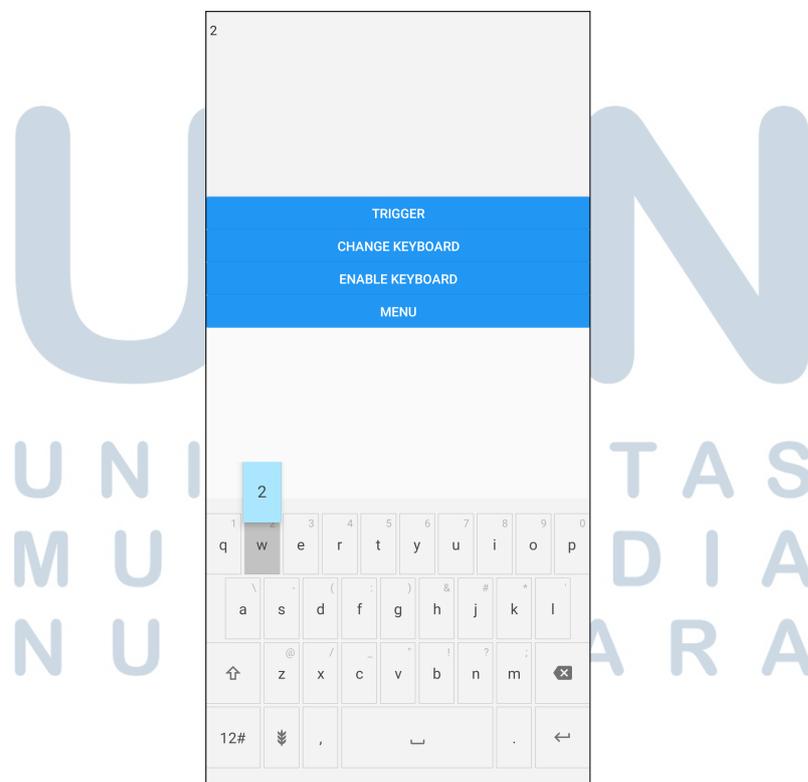
Gambar 3.47. Tampilan share gambar Invoice melalui Whatsapp

Tombol pada keyboard juga dapat menampilkan *popup* ketika ditekan yang dapat dilihat pada Gambar 3.48. Ketika tombol pada keyboard di-*hold*, maka *popup* akan berubah menjadi tombol alternatif, seperti pada Gambar 3.49

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.48. Tampilan Popup Keyboard React Native



Gambar 3.49. Tampilan Popup Alternative Keyboard React Native

3.3.3 Integrasi OpenAI ChatGPT

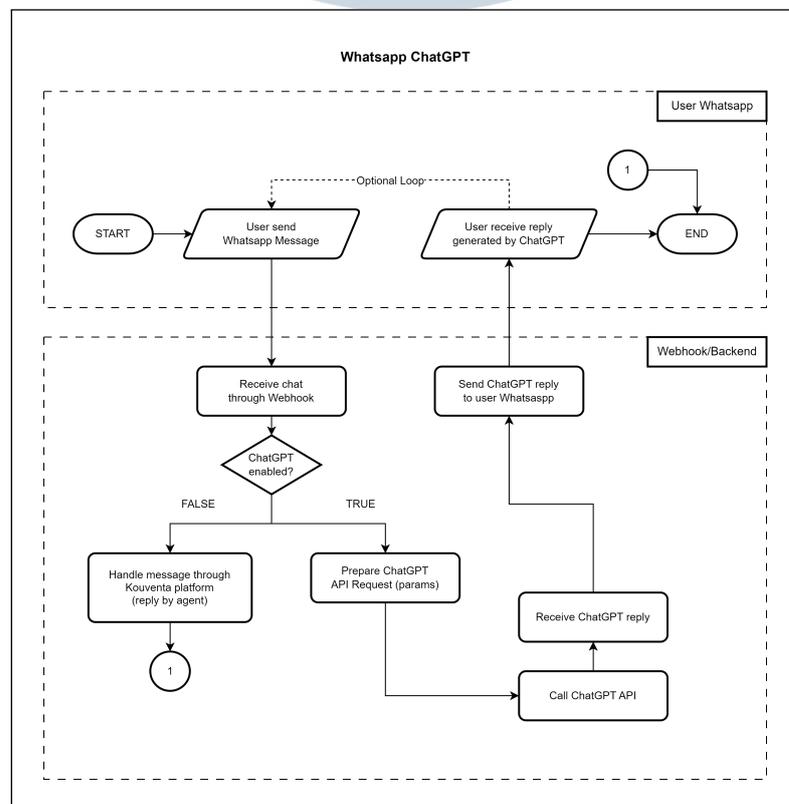
A. Requirements

Requirement untuk mengintegrasikan OpenAI ChatGPT adalah sebagai berikut.

1. *User* dapat melakukan *chatting* melalui Whatsapp yang akan dibalas oleh ChatGPT
2. ChatGPT dapat mengambil data dari *database* secara otomatis

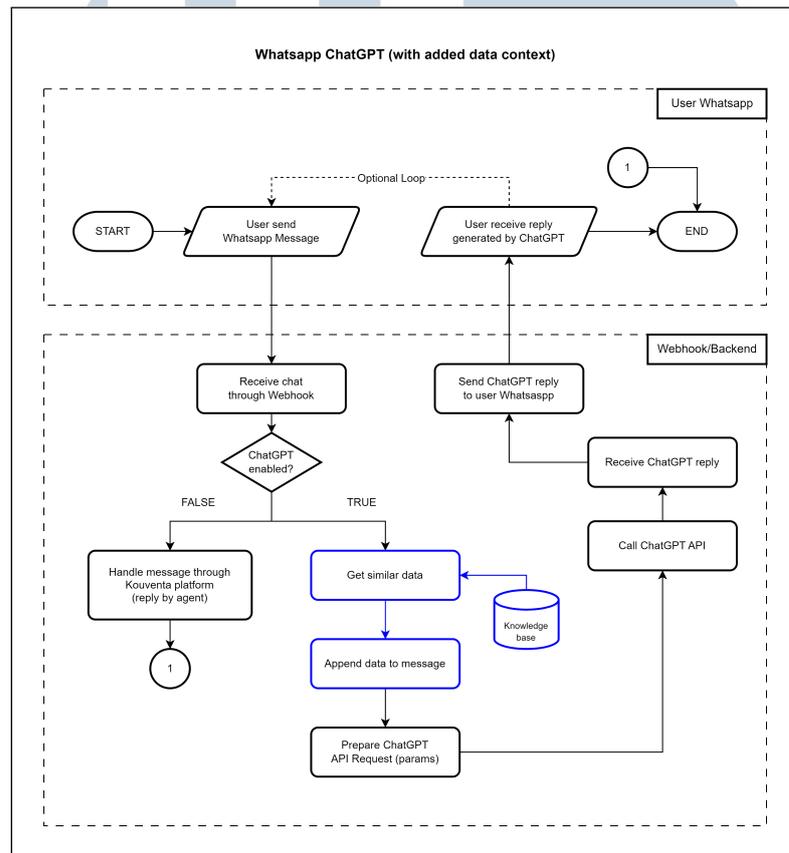
B. Process Flow

OpenAI ChatGPT diintegrasikan dengan Kouventa, sehingga sudah terkonfigurasi untuk *webhook* Whatsapp. Maka, untuk mengintegrasikan ChatGPT hanya perlu mengarahkan teks dari Whatsapp ke API ChatGPT. Alur dari mengirim pesan melalui Whatsapp hingga mendapat balasan ChatGPT dapat dilihat pada Gambar 3.50



Gambar 3.50. Flowchart Integrasi ChatGPT dengan Whatsapp

Selain hanya menjawab pesan dari *user*, ChatGPT memerlukan sumber data/informasi untuk menjawab pertanyaan *user* dengan data yang sesuai dengan keperluan bisnis. Oleh karena itu, diperlukan untuk mengambil data dari *database* untuk diberikan ke ChatGPT untuk menjawab. Maka, flow untuk integrasi ChatGPT terdapat penambahan menjadi seperti Gambar 3.51 berikut.

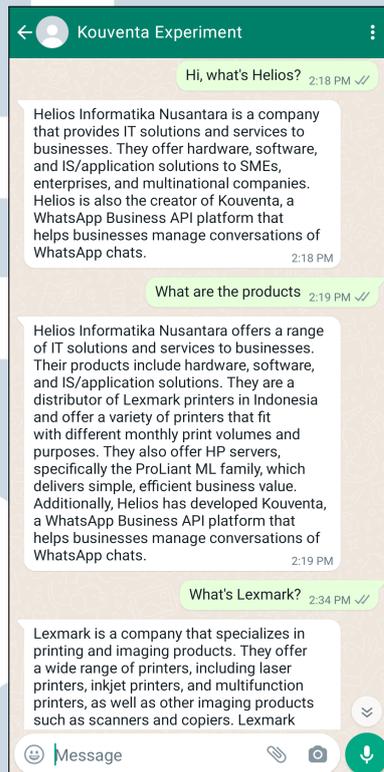


Gambar 3.51. Flowchart Integrasi ChatGPT dengan Whatsapp + Data

UNIVERSITAS
MULTIMEDIA
NUSANTARA

C. Hasil

Hasil dari integrasi OpenAI ChatGPT dengan Whatsapp dapat dilihat pada Gambar 3.52. Informasi mengenai produk Helios didapatkan langsung dari *database*, sehingga jawaban yang terlihat pada Gambar 3.52 merupakan informasi yang terdapat pada *database*.



Gambar 3.52. Contoh *chat* dengan ChatGPT melalui Whatsapp

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