

BAB 3

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Organisasi

PT Indobest Artha Kreasi memiliki divisi API yang beranggotakan sebanyak 4 orang. Divisi API bertanggung jawab untuk membantu dalam proses bisnis B2B (*Business to business*), merawat dan mengembangkan *Platform Developer*, dan membuat beberapa *website* sesuai dengan permintaan klien. Pada IAK, divisi API di supervisor langsung oleh Gandhi Pranata selaku *Lead Software Engineer*.

Pada saat berada di divisi API, koordinasi dilakukan melalui grup *telegram* yang terdiri dari CEO, CTO, tim *Marketing*, dan seluruh tim API. Kemudian, terdapat grup internal untuk keperluan koordinasi internal yang terdiri dari semua anggota tim API. Rapat bersama untuk tim API dilakukan setiap minggu pada hari senin jam 11.00 secara langsung (tatap muka) atau melalui *google meeting*. Rapat mingguan dilakukan untuk membahas perkembangan atau *task* yang sudah dilakukan pada minggu sebelumnya dan sekaligus menentukan fokus *task* untuk minggu ini.

3.2. Tugas yang Dilakukan

Proyek yang dilakukan selama masa kerja magang di IAK adalah untuk membuat sebuah *website* untuk klien dari IAK (Gamefinity) yang memungkinkan pengguna untuk dapat bertransaksi berbagai macam produk prabayar (voucher game, pulsa, dan paket internet) yang tersedia pada IAK melalui *website*. Agar dapat berfungsi dengan baik, *website* Gamefinity memiliki beberapa fitur utama seperti fitur transaksi(beli), fitur *auto refund*, dan fitur pengiriman *receipt*.

Bahasa-bahasa pemrograman yang digunakan untuk membangun *website* Gamefinity adalah PHP dan *JavaScript*. PHP digunakan untuk membangun bagian *server-side scripting* atau *Back-end* dari *website* Gamefinity, sedangkan *JavaScript* digunakan untuk membangun bagian *client-side scripting* atau *Front-end* dari *website* Gamefinity. *Framework* yang digunakan untuk membangun dan mengembangkan *website* Gamefinity adalah Laravel sebagai *framework* dasar dari *website* Gamefinity, Vuejs digunakan untuk memberikan tampilan *Progressive Web App* (PWA) pada *website*, dan *tailwind-css* untuk memberikan *styling* pada *website*.

Tabel 3.1. Realisasi Proyek *Website* Gamefinity

Minggu Ke -	Pekerjaan yang dilakukan
1	Mengikuti <i>meeting</i> untuk membahas mengenai <i>requirement</i> yang dibutuhkan dari klien Gamefinity
2	Mencari tahu dan mempelajari mengenai bahasa-bahasa pemrograman yang akan digunakan untuk mengembangkan proyek
3	Inisialisasi proyek Gamefinity dan pengembangan tampilan awal menggunakan <i>dummy data</i>
4	Pengembangan tampilan untuk masing-masing produk menggunakan <i>dummy data</i>
5	Pengembangan sisi <i>Back-end</i> untuk mengambil data <i>pricelist</i>
6	Pengembangan sisi <i>Back-end</i> untuk memproses transaksi (<i>Payment</i>) menggunakan Midtrans
7	Pengembangan sisi <i>Back-end</i> untuk memproses API Prepaid IAK
8	Melakukan <i>testing</i> dan perbaikan pada <i>website</i> Gamefinity

Minggu pertama dari proyek Gamefinity diisi dengan *requirement gathering* bersama dengan klien Gamefinity untuk membahas apa saja fitur-fitur yang diperlukan dan diinginkan pada *website* Gamefinity. Setelah *requirement gathering* selesai dilakukan, kegiatan berikutnya adalah membuat rancangan kasar atau *mockup* dari *website*. *Mockup* dibuat untuk memberikan gambaran kepada klien mengenai garis besar tampilan dari *website*. Kemudian, *Mockup* yang sudah selesai dibuat akan ditunjukkan kepada klien Gamefinity untuk dinilai dan disesuaikan. Setelah *mockup* disetujui, CTO dan *Lead Software Engineer* untuk tim API memberikan arahan mengenai *Tech Stack* yang disarankan untuk digunakan dalam proyek Gamefinity.

Pada minggu kedua dari proyek Gamefinity, aktivitas yang dilakukan didominasi dengan pembelajaran dan riset mengenai *Tech Stack* yang disarankan pada minggu sebelumnya. Sumber utama pembelajaran berasal dari dokumentasi untuk masing-masing *Tech Stack*. *Tech Stack* yang akan digunakan untuk membangun proyek Gamefinity adalah Laravel, Vuejs, Tailwind-css, dan Midtrans sebagai *Payment Gateway*.

Minggu ketiga dari proyek Gamefinity diawali dengan menginisialisasi *repository* dan *Dummy Data* untuk proyek Gamefinity. Setelah inisialisasi selesai dilakukan, pengembangan dari *website* dimulai dari sisi *Front-end* terlebih dahulu

menggunakan *Dummy Data* yang sudah dibuat. Pengembangan *Front-end* pada minggu ketiga berfokus pada halaman awal dari *website* untuk menampilkan seluruh produk yang ada.

Minggu keempat dari proyek Gamefinity diisi dengan pengembangan sisi *Front-end* untuk tampilan detail pada setiap produk yang tampil pada halaman awal. Pada proses pengembangan minggu keempat, terdapat kendala yang terjadi dikarenakan tidak terbiasa menggunakan fitur *Data Binding* yang ada pada Vuejs. Kemudian, dilakukan beberapa perbaikan dan penyesuaian untuk transisi halaman awal dengan halaman detail produk.

Pada minggu kelima dari proyek Gamefinity, pengembangan sisi *Back-end* dimulai. Pengembangan dimulai dengan melakukan *fetch* untuk mengambil data *pricelist* melalui API *Prepaid* IAK. Kendala-kendala yang terjadi pada minggu kelima adalah terdapat beberapa keterlambatan informasi mengenai produk-produk yang akan dijual dan terdapat beberapa miskomunikasi untuk pembuatan kredensial untuk mengakses API *Prepaid* IAK. Selain pengembangan sisi *Back-end*, pada sisi *Front-end* ditambahkan fitur tampilan *skeleton* untuk memberikan *User Experience* yang lebih baik.

Minggu keenam dari proyek Gamefinity melanjutkan pengembangan sisi *Back-end* untuk melakukan transaksi dengan menggunakan Midtrans. Kendala yang terjadi pada minggu keenam adalah kesulitan pada saat integrasi Midtrans, dikarenakan dokumentasi yang ada pada *website* Midtrans kurang lengkap dan terstruktur. Kemudian, kendala yang lain adalah belum memahami model bisnis dari transaksi *prepaid* yang berjalan secara asinkronus.

Minggu ketujuh dari proyek Gamefinity melanjutkan pengembangan sisi *Back-end* untuk melakukan transaksi melalui API *prepaid* IAK setelah sukses melakukan pembayaran melalui Midtrans. Selain itu, pada minggu ketujuh pada sisi *Back-end* ditambahkan fitur pengiriman *receipt* atau *invoice*. Kendala yang terjadi pada saat pengembangan fitur *receipt* adalah kesulitan untuk melakukan percobaan atau *testing* secara lokal. Selanjutnya minggu kedelapan dari proyek Gamefinity diisi dengan kegiatan *testing* secara internal dan eksternal. Kemudian, melakukan perbaikan atau penyesuaian pada hasil temuan pada saat *testing*.

3.3. Uraian Pelaksanaan Magang

Proses kerja magang selama berada di PT Indobest Artha Kreasi telah dilakukan kurang lebih selama 5 bulan. Uraian pengerjaan terdiri dengan perangkat yang digunakan, *flowchart*, dan hasil implementasi.

3.3.1. Website Gamefinity

Beberapa *software* dan *hardware* yang digunakan dalam mendukung proses kerja magang ini. *Software* yang digunakan adalah sebagai berikut.

- *Framework* laravel v7.0, vuejs v2.6, dan tailwindcss v.2.0
- PHP v7.3.33
- OS Linux Ubuntu v20.04
- Google Chrome v101.0.4951.54
- PhpStorm v2021.3

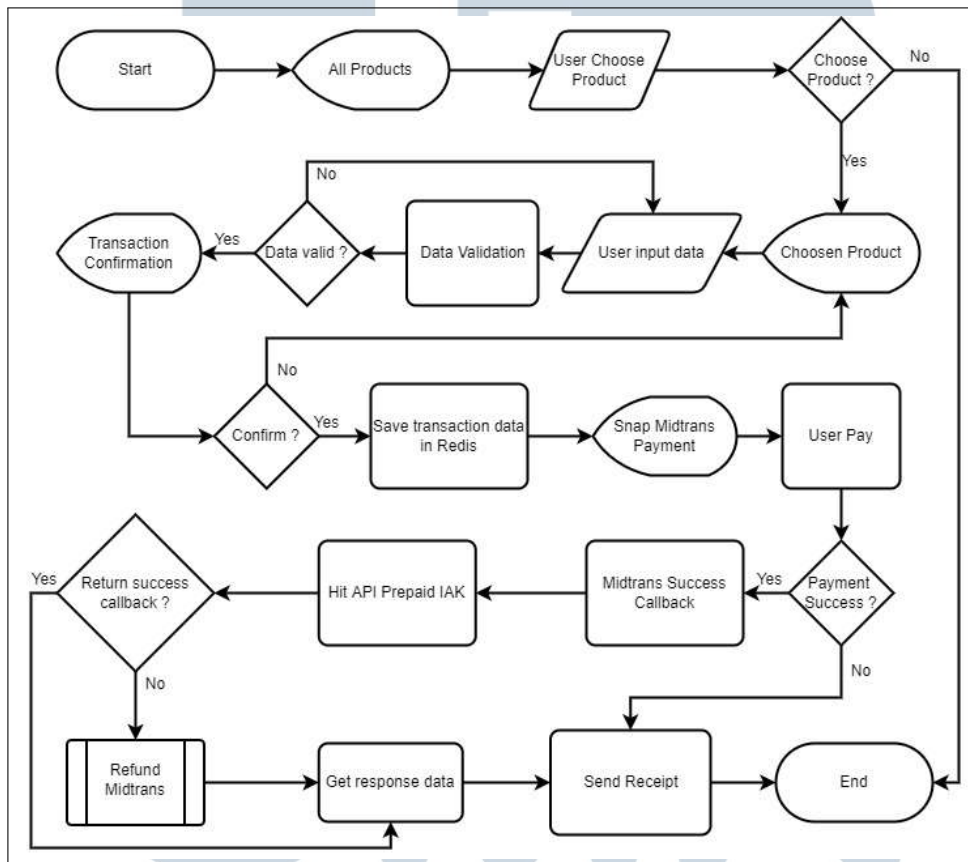
Spesifikasi dari perangkat keras (HP ProBook 430 G8) yang digunakan selama membuat *website* Gamefinity adalah sebagai berikut.

- CPU Intel(R) Core i5-1135G7
- RAM 16 GB DDR4
- Intel Iris Xe Graphics

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

A. Flowchart

Secara garis besar *flowchart* dari *website* Gamefinity terbagi menjadi 2 *flowchart*. *Flowchart* pertama (Gambar 3.1) berfungsi untuk menggambarkan alur utama transaksi pada *website* Gamefinity. Kemudian, *Flowchart* kedua (Gambar 3.2) berfungsi untuk menggambarkan alur dari fungsi *direct refund* yang ada pada *website* Gamefinity.



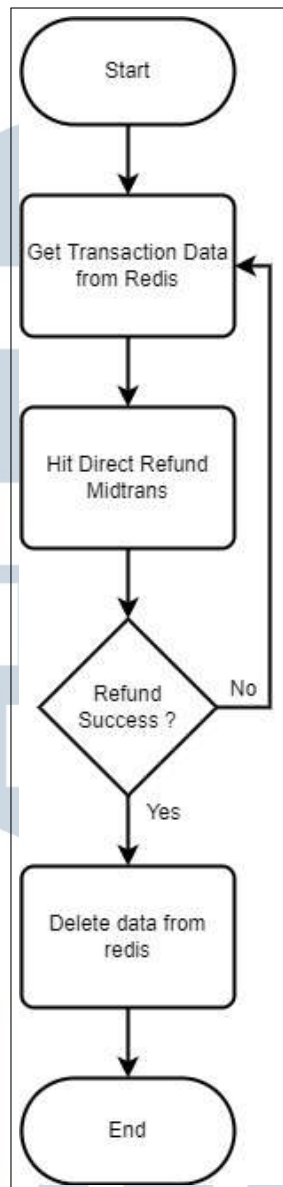
Gambar 3.1. *Flowchart* *website* Gamefinity

Gambar 3.1 merupakan *flowchart* utama dari *website* Gamefinity. Pertama, pengguna dari *website* Gamefinity yang ingin membeli produk yang tersedia akan diperlihatkan seluruh produk yang tersedia pada *website* melalui tampilan halaman awal. Kemudian setelah pengguna memilih salah satu produk yang ditampilkan, tampilan halaman akan berubah menjadi detail(nominal dan metode pembayaran) dari produk tersebut. Setelah pengguna mengisi data-data yang diperlukan seperti email, id permainan (jika ada), nomi-

nal, dan metode pembayaran, akan dilakukan validasi terhadap data-data tersebut. Apabila terdapat kesalahan pada data-data yang dimasukkan maka akan ditampilkan peringatan berwarna merah untuk bagian yang masih salah atau belum diisi. Bila data-data yang dimasukkan pengguna sudah lengkap dan benar maka, akan muncul tampilan pop-up untuk memberikan konfirmasi mengenai data-data yang sudah diinput dan nominal transaksi yang akan dilakukan. Apabila pengguna melakukan konfirmasi maka akan data transaksi akan disimpan pada Redis dan *website* akan memunculkan *snap* dari Midtrans untuk melakukan pembayaran.

Pengguna dapat melakukan pembayaran pada Midtrans melalui metode pembayaran yang sudah dipilih sebelumnya. Apabila pengguna gagal melakukan pembayaran atau tidak melakukan pembayaran, maka Midtrans akan memberikan respon *callback* gagal. Setelah Midtrans memberikan respon *callback* gagal, maka secara otomatis *receipt* akan terkirim kepada *email* pengguna untuk memberitahukan bahwa transaksi yang dilakukan gagal. Sedangkan, apabila pengguna menyelesaikan pembayaran dan Midtrans memberikan respon *callback* transaksi sukses maka langkah berikutnya yang terjadi adalah melakukan hit untuk top-up pada API prabayar IAK. Apabila API IAK memberikan respon *callback* transaksi gagal maka akan dilakukan *refund* untuk transaksi tersebut dan pengguna akan menerima *receipt* transaksi gagal. Apabila API IAK memberikan respon *callback* sukses maka data-data seperti kode voucher akan diambil dari respon API IAK dan diletakan pada *receipt* untuk dikirimkan pada pengguna melalui *email*.

Gambar 3.2 merupakan *flowchart* untuk proses *refund* pada *website* Gamefinity. Pertama, sistem akan mengambil data transaksi yang sudah disimpan pada Redis sebelumnya. Selanjutnya, data transaksi yang sudah didapatkan akan dikirimkan kepada Midtrans untuk dilakukan *Direct Refund*. Apabila proses *Direct Refund* Midtrans berhasil maka dana yang sudah dibayarkan pengguna sebelumnya akan secara otomatis dikembalikan dan data akan dihapus dari Redis. Apabila proses *Direct Refund* gagal maka data akan tetap tersimpan di Redis selama 24 jam untuk dicoba kembali.



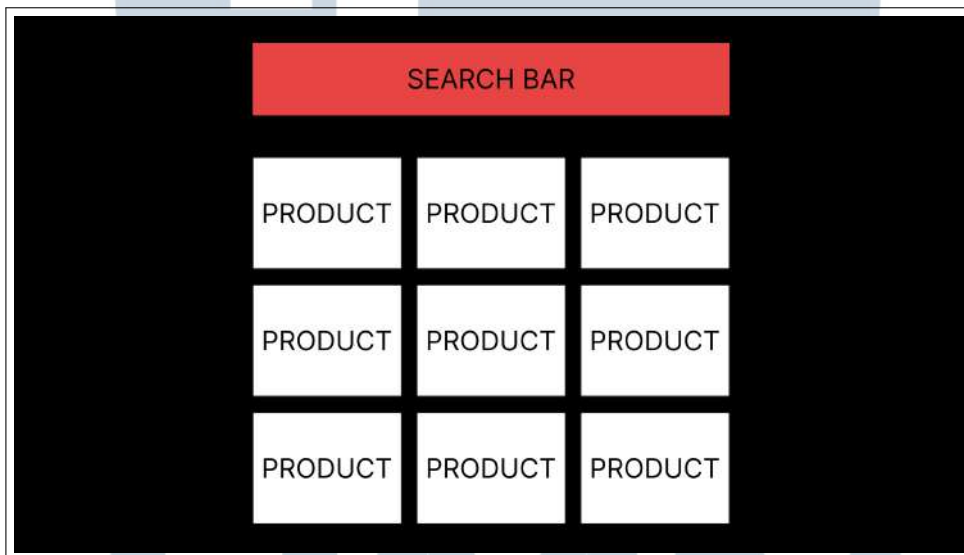
Gambar 3.2. Flowchart Refund

B. Rancangan Tampilan

Sebelum memulai proses pengembangan *website* Gamefinity, beberapa rancangan tampilan dibuat untuk memberikan garis besar tampilan *website* Gamefinity yang akan dibuat kepada klien. Rancangan tampilan yang dibuat meliputi tampilan halaman awal, halaman *checkout*, halaman nominal produk, dan *modal box* untuk konfirmasi.

B.1. Halaman Awal

Gambar 3.3 merupakan rancangan halaman awal *website* Gamefinity terdapat sebuah *search bar* yang terletak pada bagian atas halaman untuk melakukan pencarian berdasarkan nama produk yang tersedia. Kemudian dibawah *search bar* akan ditampilkan seluruh produk yang akan dijual pada *website* Gamefinity dalam bentuk *card container*. Apabila salah satu produk dipilih atau diklik maka tampilan akan berubah menjadi seperti gambar 3.4.

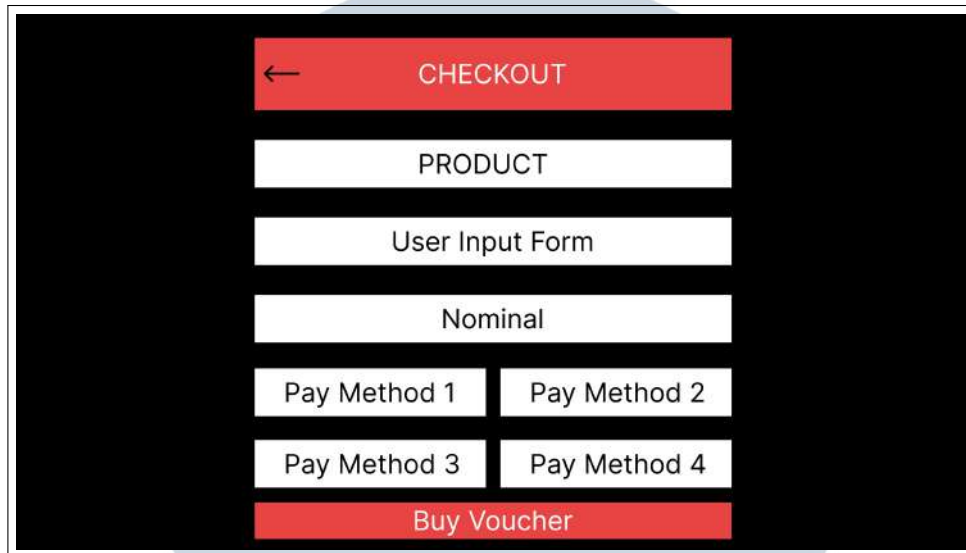


Gambar 3.3. Rancangan Tampilan Halaman Awal

B.2. Halaman Checkout

Gambar 3.4 merupakan rancangan halaman *checkout* setelah memilih sebuah produk pada *website* Gamefinity. Pada rancangan halaman *checkout* pada bagian atas terdapat judul halaman yaitu *checkout* dan tombol untuk kembali ke halaman awal. Kemudian pada isi dari rancangan halaman *checkout* terdapat 4 komponen penting, yaitu informasi produk yang dipilih, *form* yang akan diisi dengan informasi pengguna (email, nomor hp, ID permainan, dll), nominal produk yang dapat dipilih, dan pilihan metode pembayaran. Pada bagian bawah rancangan halaman *checkout* terdapat tombol

pembelian yang pada saat di klik akan menampilkan *modal box* untuk konfirmasi pengguna.



Gambar 3.4. Rancangan Tampilan Halaman *Checkout*

B.3. Halaman Nominal Produk

Gambar 3.5 merupakan rancangan halaman nominal produk setelah pengguna mengklik bagian nominal pada halaman *checkout*. Pada rancangan halaman nominal produk bagian atas terdapat judul halaman nominal produk dan tombol untuk kembali ke halaman *checkout*. Bagian utama dari halaman nominal produk berisi pilihan nominal yang tersedia untuk produk yang dipilih. Setelah memilih nominal produk, tampilan halaman akan kembali ke halaman *checkout*.

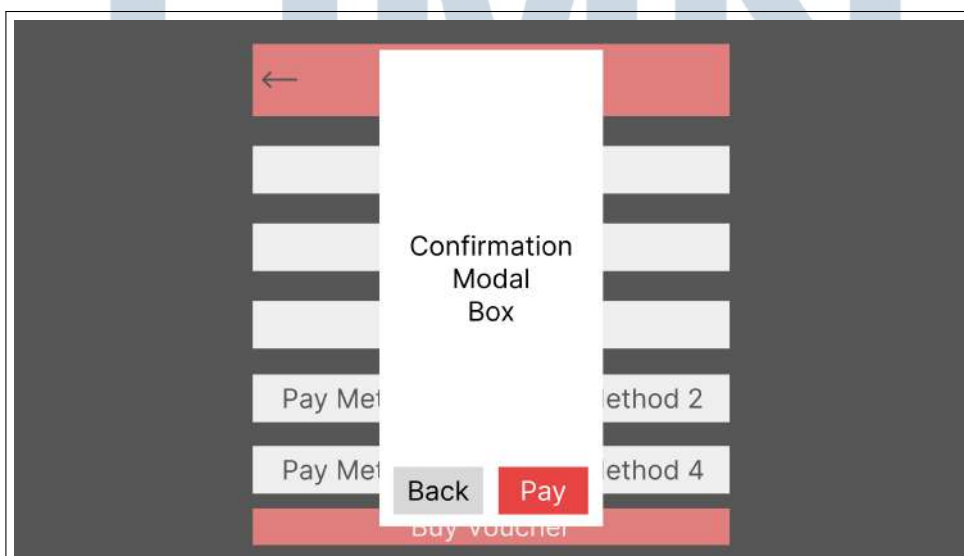
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.5. Rancangan Tampilan Halaman Nominal produk

B.4. *Modal Box* Konfirmasi

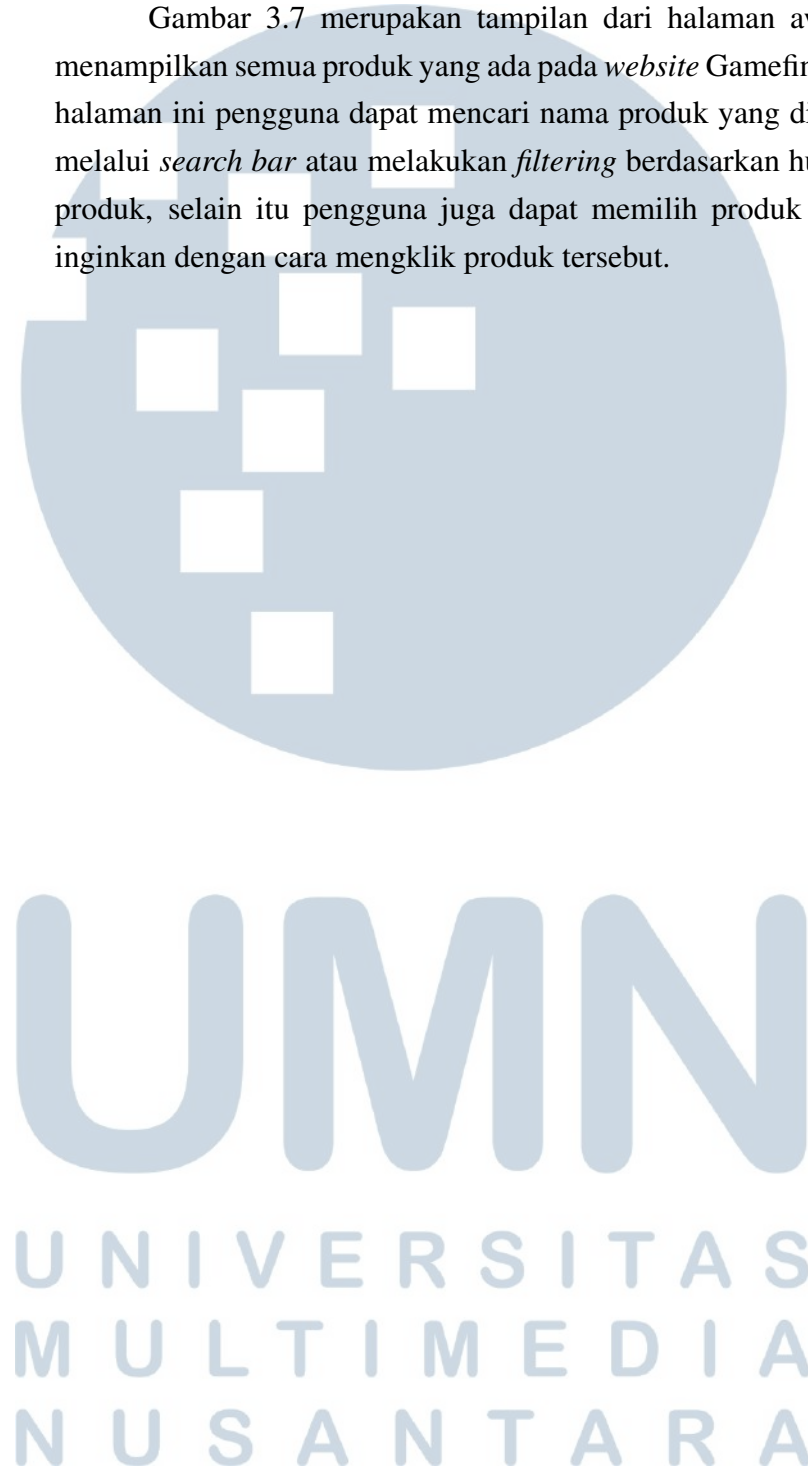
Gambar 3.6 merupakan rancangan *modal box* yang akan dimunculkan setelah pengguna mengklik tombol pembelian yang ada pada halaman *checkout*. Tujuan dari ditampilkannya *Modal box* adalah agar pengguna dapat mengecek atau mengonfirmasi kembali apakah produk, informasi pengguna, nominal produk, dan metode pembayaran yang dipilih sudah benar atau belum.



Gambar 3.6. Rancangan Tampilan *modal box* untuk Konfirmasi

C. Implementasi

Gambar 3.7 merupakan tampilan dari halaman awal yang menampilkan semua produk yang ada pada *website* Gamefinity. Pada halaman ini pengguna dapat mencari nama produk yang diinginkan melalui *search bar* atau melakukan *filtering* berdasarkan huruf awal produk, selain itu pengguna juga dapat memilih produk yang diinginkan dengan cara mengklik produk tersebut.

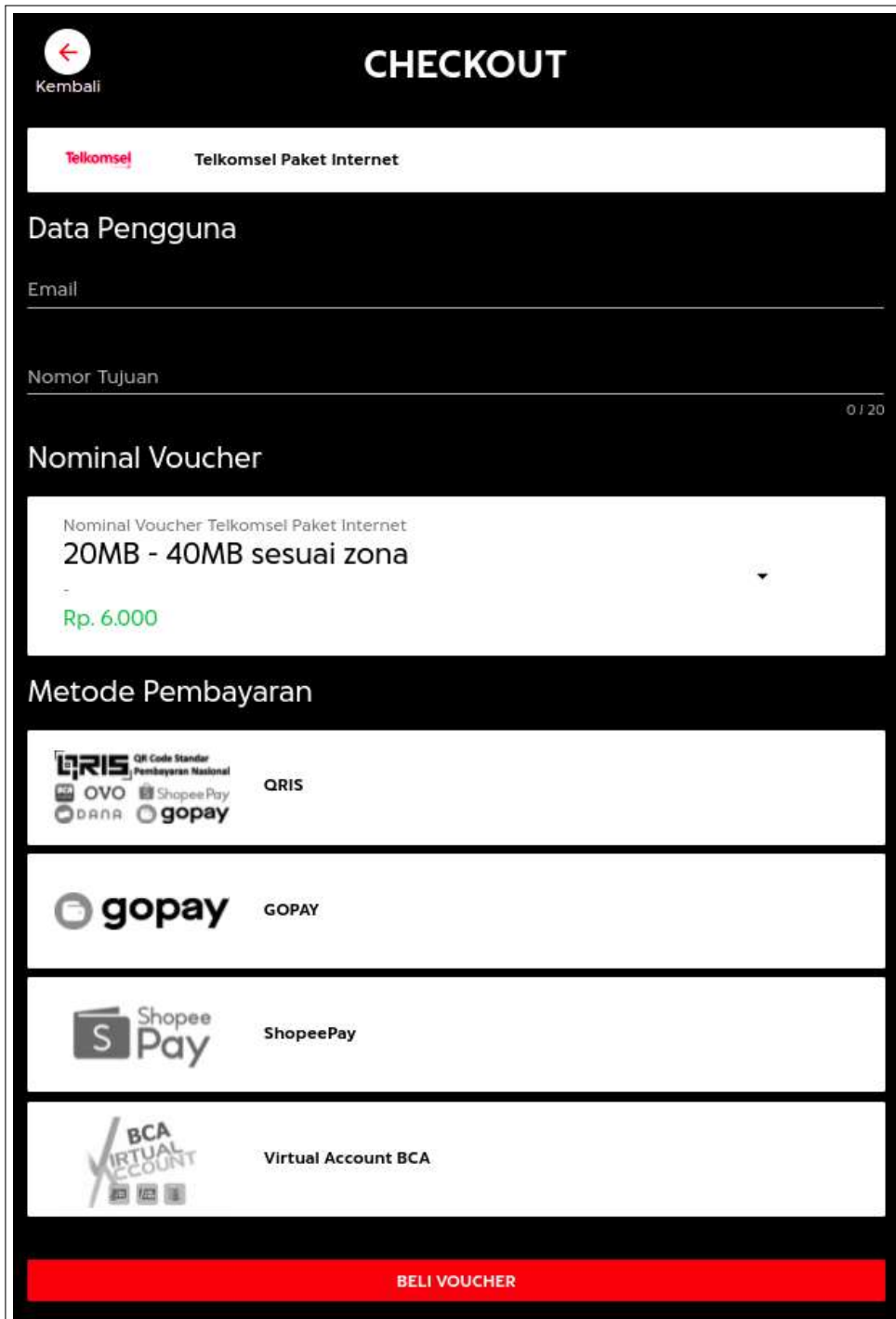




Gambar 3.7. Halaman awal *website* Gamefinity

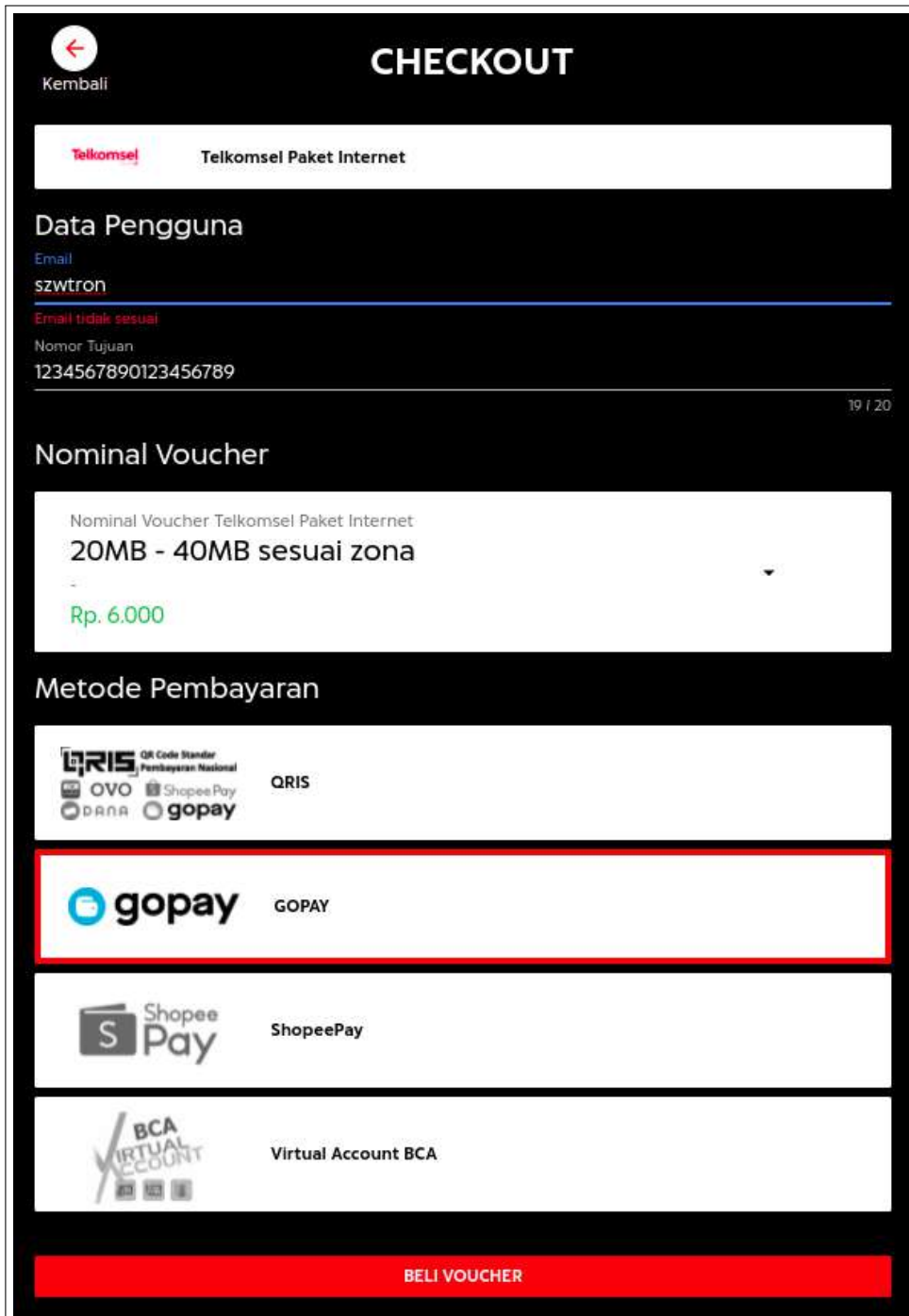
Gambar 3.8 merupakan tampilan dari halaman detail produk yang sudah dipilih oleh pengguna. Pada halaman ini pengguna perlu mengisi alamat email, nomor tujuan (bila diperlukan), nominal produk, dan metode pembayaran yang diinginkan. Untuk setiap detail yang perlu diisi pengguna terdapat validasi data untuk masing-masing *input*. Gambar 3.9 merupakan tampilan peringatan yang muncul apabila pengguna salah menginput alamat email. Kemudian, gambar 3.10 merupakan tampilan apabila pengguna salah menginput nomor tujuan (bila ada) akan dimunculkan *modal box* yang menginformasikan bahwa nomor yang dimasukkan pengguna salah. Kemudian, gambar 3.11 merupakan tampilan apabila pengguna ingin memilih nominal lain yang tersedia untuk produk yang dipilih. Gambar 3.12 merupakan tampilan peringatan yang muncul apabila pengguna tidak memilih metode pembayaran yang tersedia.





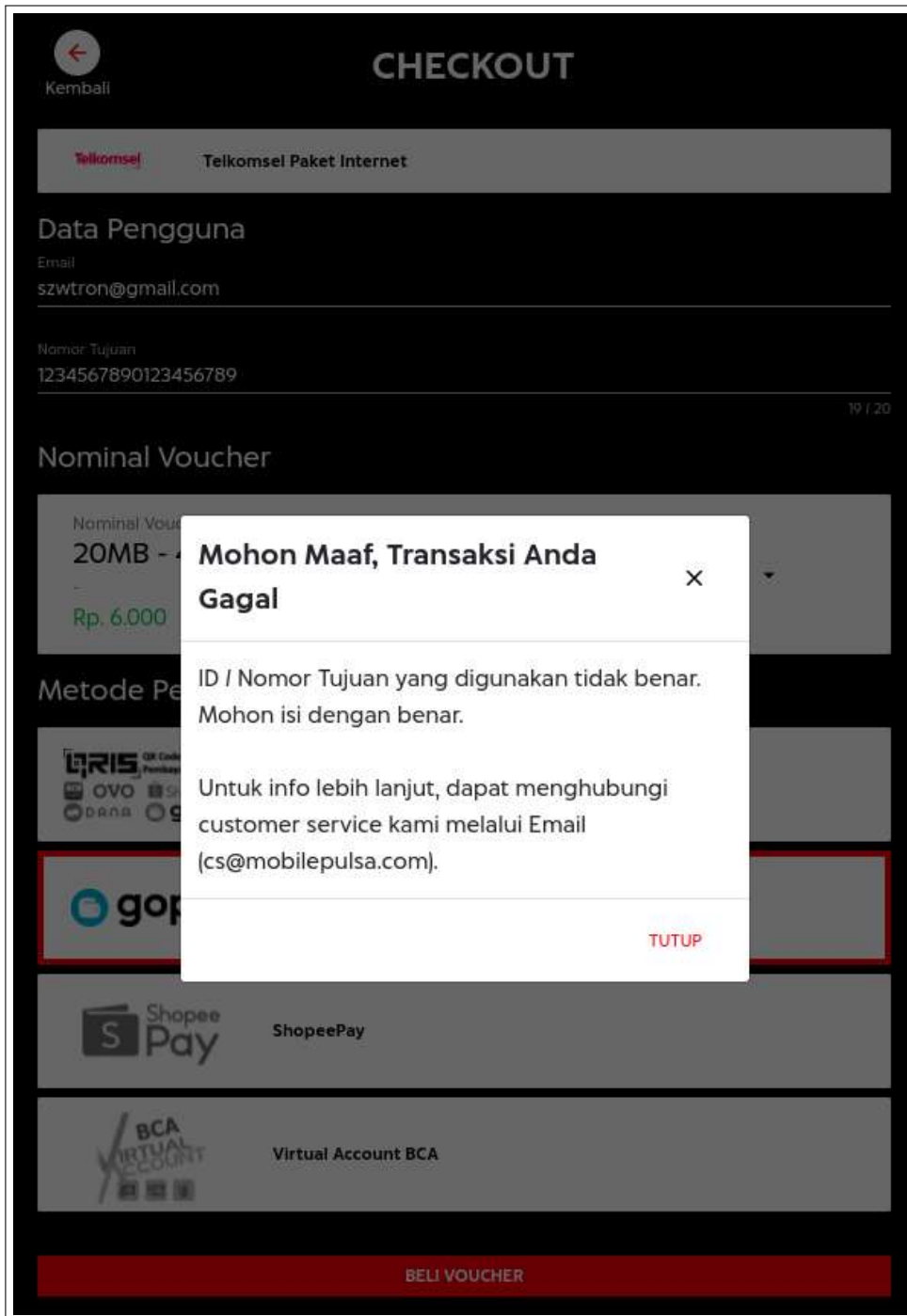
Gambar 3.8. Halaman detail produk

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



Gambar 3.9. Validasi email pada halaman detail produk

MULTIMEDIA
NUSANTARA

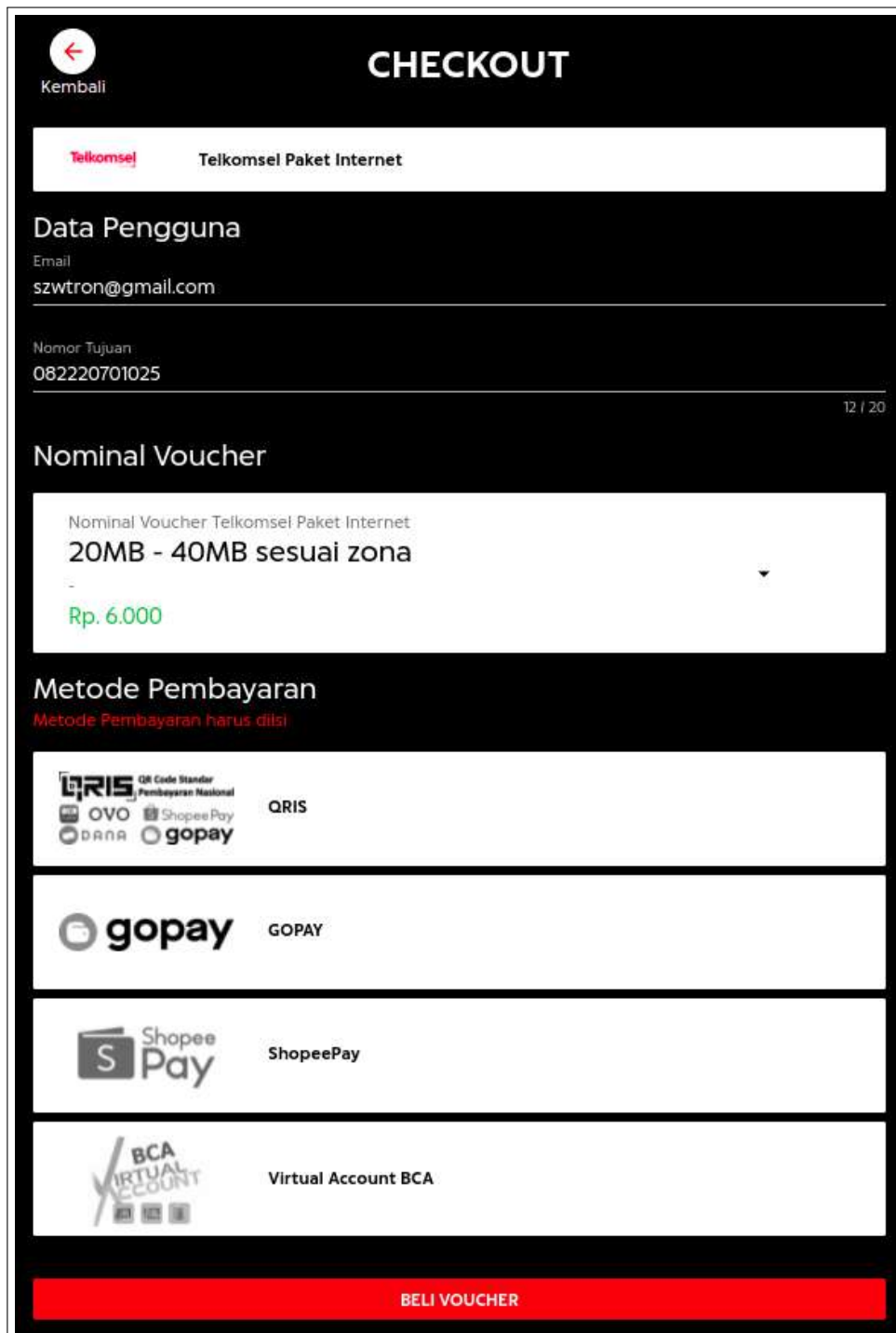


Gambar 3.10. Validasi nomor tujuan pada halaman detail produk

Nominal Voucher Telkomsel Paket Internet	
20MB - 40MB sesuai zona	Rp. 6.000
50MB - 110MB sesuai zona	Rp. 11.500
10GB VideoMax, 7 hari	Rp. 11.500
500 MB semua jaringan + bebas zona	Rp. 14.800
1 GB semua jaringan + bebas zona	Rp. 21.000
200MB - 420MB sesuai zona	Rp. 21.500
750 MB semua jaringan + bebas zona	Rp. 22.500
GamesMAX 500MB Internet + 2.5GB Internet Games Kuota	Rp. 25.000
360MB - 800MB sesuai zona	Rp. 27.000
2 GB semua jaringan + bebas zona	Rp. 45.000
12 GB VideoMax, 30 hari	Rp. 50.000
800MB - 1.5GB sesuai zona	Rp. 52.000
3 GB semua jaringan + bebas zona	Rp. 70.000
12 GB + 2GB VideoMax	Rp. 99.500
50 GB + 2GB VideoMax	Rp. 198.500

Gambar 3.11. Halaman nominal produk

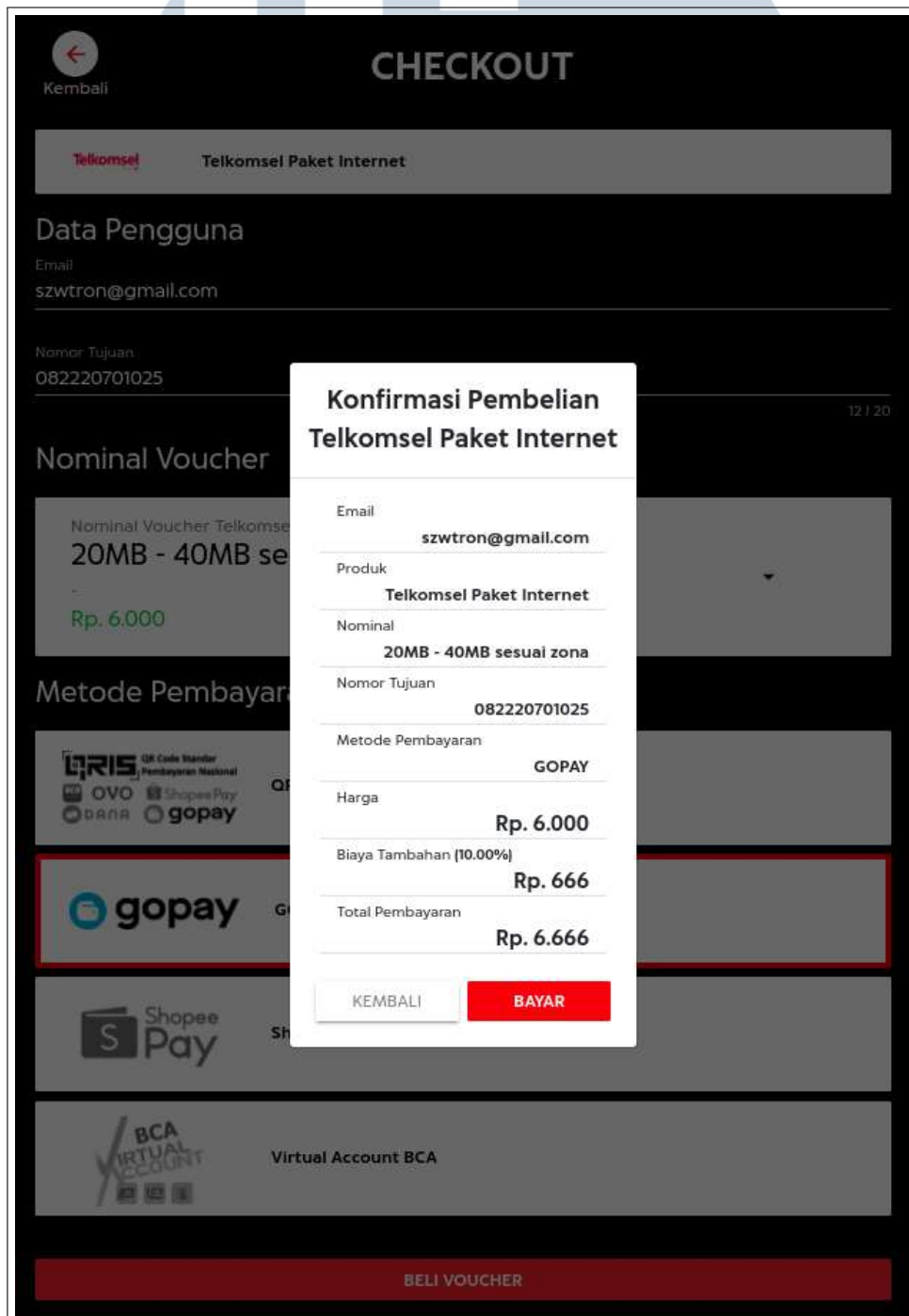
UMMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA



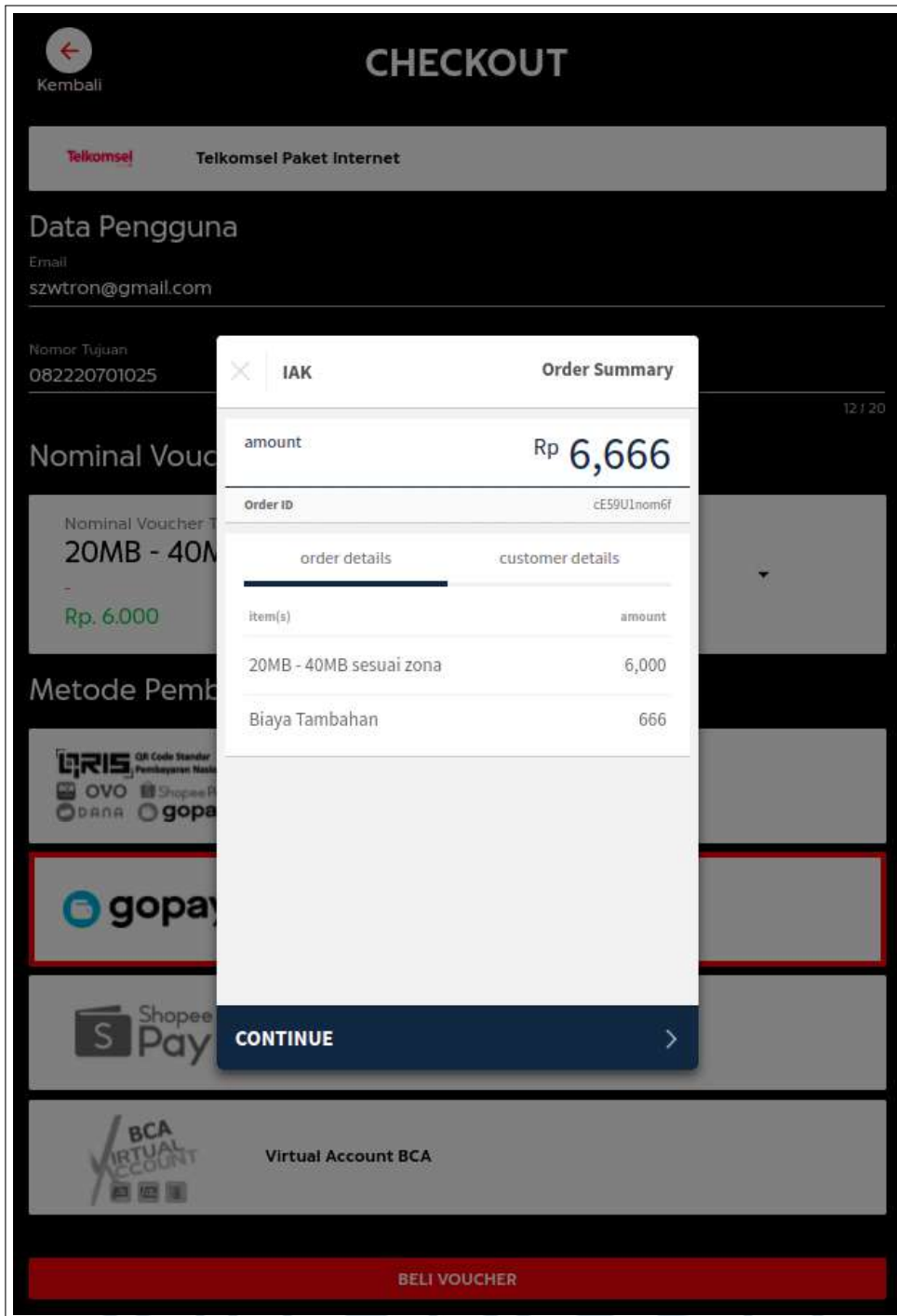
Gambar 3.12. Validasi metode pembayaran pada halaman detail produk

Gambar 3.13 merupakan tampilan dari *modal box* konfirmasi yang muncul setelah seluruh data pengguna berhasil tervalidasi. *modal box* yang muncul berisi ringkasan transaksi yang akan dilakukan yaitu alamat email, produk, nominal, nomor tujuan (bila

ada), metode pembayaran, harga, biaya tambahan, dan total jumlah yang harus dibayarkan. Apabila pengguna mengklik tombol bayar pada *modal box*, maka akan muncul *snap* dari Midtrans seperti pada Gambar 3.14. Melalui *snap* Midtrans, pengguna diminta untuk menyelesaikan pembayaran sesuai dengan jumlah yang tertera.



Gambar 3.13. *Modal box* konfirmasi transaksi



Gambar 3.14. Tampilan *snap* Midtrans

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

Gambar 3.15 merupakan tampilan dari *receipt* transaksi sukses yang dikirimkan melalui email pengguna setelah seluruh proses transaksi berhasil. pada *receipt* terdapat beberapa informasi seperti tanggal transaksi, metode pembayaran, kode transaksi, status transaksi, email, nomor tujuan (bila ada), nominal produk, *serial number* (bila sukses), harga produk, biaya tambahan, dan total pembayaran. Apabila transaksi mengalami kegagalan, maka pengguna akan menerima *receipt* transaksi gagal melalui email (seperti pada gambar 3.16).





BUKTI TRANSAKSI PEMBELIAN TELKOMSEL PAKET INTERNET

TANGGAL TRANSAKSI : 25 May 2022 10:13:52

METODE PEMBAYARAN : GOPAY

KODE TRANSAKSI : cE59U1nom6f

STATUS TRANSAKSI : SUKSES

EMAIL : szwtron@gmail.com

NOMOR TUJUAN : 082220701025

NOMINAL : 20MB - 40MB sesuai zona

SERIAL NUMBER : 123456789

HARGA PRODUK : **Rp. 6.000**

BIAYA TAMBAHAN : **Rp. 666**

TOTAL PEMBAYARAN : **RP. 6.666**

Ada pertanyaan, kritik, dan saran mengenai transaksi?
Hubungi kami melalui
Email (cs@mobilepulsa.com)

**Terima kasih telah bertransaksi di
WWW.GAMEFINITY.ID**

Gambar 3.15. *Receipt* transaksi sukses



BUKTI TRANSAKSI PEMBELIAN TELKOMSEL PAKET INTERNET

TANGGAL TRANSAKSI : 25 May 2022 10:18:30

METODE PEMBAYARAN : GOPAY

KODE TRANSAKSI : cE59U1nom6f

STATUS TRANSAKSI : GAGAL

EMAIL : szwtron@gmail.com

NOMOR TUJUAN : 082220701025

NOMINAL : 20MB - 40MB sesuai zona

HARGA PRODUK : **Rp. 6.000**

BIAYA TAMBAHAN : **Rp. 666**

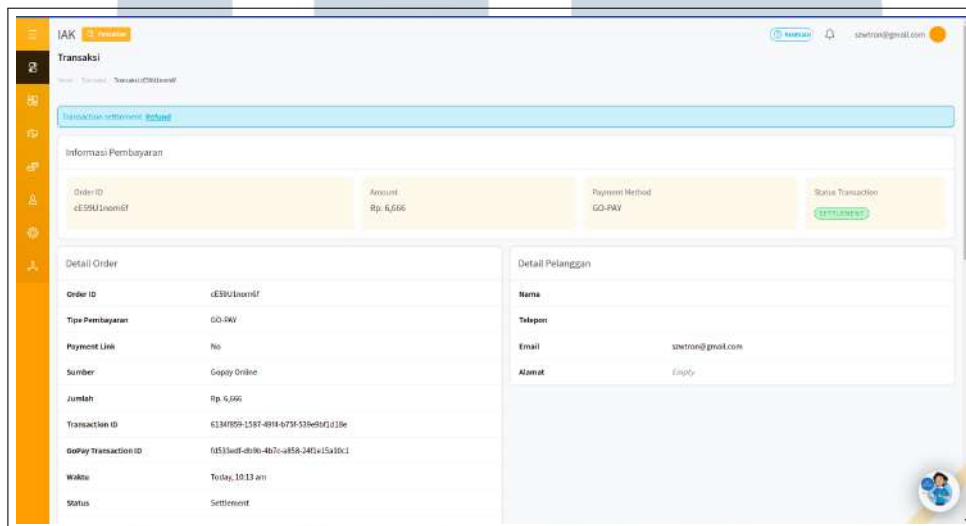
TOTAL PEMBAYARAN : **RP. 6.666**

Transaksi Gagal?
Hubungi kami melalui
Email (cs@mobilepulsa.com)

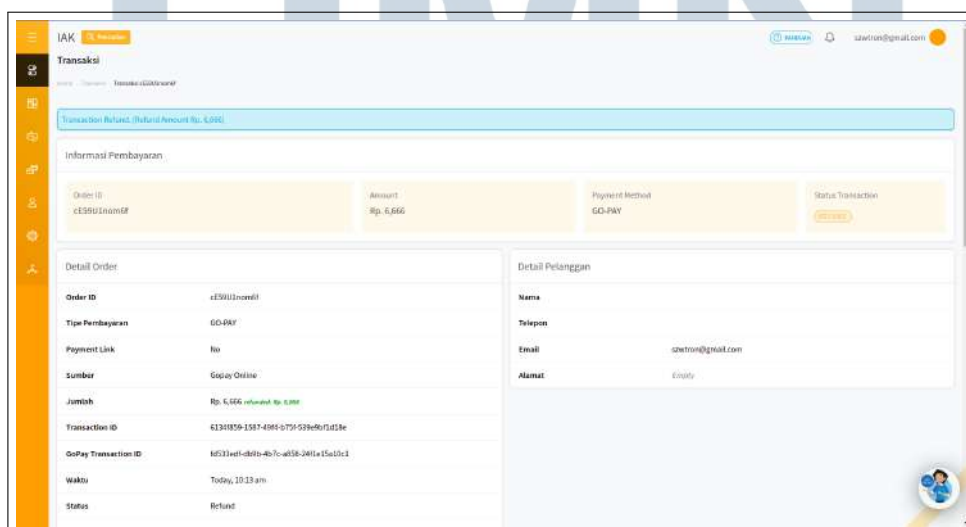
Terima kasih telah bertransaksi di
WWW.GAMEFINITY.ID

Gambar 3.16. *Receipt* transaksi gagal

Gambar 3.17 merupakan tampilan dari Midtrans apabila pengguna sudah berhasil membayar melalui Midtrans. Status transaksi apabila pengguna sudah berhasil melakukan pembayaran melalui Midtrans adalah *settlement*. Kemudian, apabila pada saat melakukan *top-up* melalui API IAK hasil yang dikembalikan adalah transaksi gagal. Maka, akan dilakukan proses *direct refund* secara otomatis. Setelah *direct refund* berhasil dilakukan, maka status transaksi pada Midtrans akan berubah menjadi *refund* seperti pada gambar 3.18.



Gambar 3.17. Transaksi *settlement* pada Midtrans



Gambar 3.18. Transaksi *refund* pada Midtrans

D. Testing

Sebelum *website* Gamefinity di luncurkan secara global, dilakukan serangkaian percobaan atau *testing* yang dilakukan oleh pihak internal dan eksternal. Percobaan dengan pihak internal dilakukan dengan metode *White Box Testing* sebelum dilakukan UAT (*User Acceptance Testing*). Tujuan dari dilakukan *White Box Testing* adalah untuk memastikan *logic* dan algoritma yang digunakan dalam pengembangan *website* sudah berjalan atau berfungsi dengan benar.

User Acceptance Testing dilakukan dengan melibatkan pihak eksternal (klien dan tim pemasaran) dan pihak internal (tim API). *User Acceptance Testing* dilakukan dengan menggunakan metode *Black Box Testing*. Tujuan dari dilakukan UAT adalah untuk memastikan *behaviour* dan alur dari *website* Gamefinity sudah sesuai dengan permintaan. Pada saat UAT pertama dilakukan, ditemukan beberapa ketidaksesuaian yang tergolong minor seperti penggunaan warna dan tata letak komponen. Setelah dilakukan perbaikan terhadap temuan UAT sebelumnya, dilakukan UAT kedua dan tidak ditemukan ketidaksesuaian pada *website* Gamefinity (sudah memenuhi standar yang ditetapkan).

3.4. Kendala dan Solusi yang Ditemukan

Kendala yang ditemukan selama menjalani kerja magang pada PT. Indobest Artha Kreasi adalah sebagai berikut.

1. *Tech stack* yang digunakan untuk membangun *website* Gamefinity adalah Vuejs, Laravel, dan tailwind-css. Dimana sebagian besar dari *tech stack* yang digunakan merupakan sebuah hal yang baru dan jarang digunakan selama masa perkuliahan. Solusi yang dilakukan adalah membaca dokumentasi yang tersedia untuk masing-masing *tech stack* dan meminta arahan atau saran dari rekan kerja.
2. Kurang memahami model bisnis yang diterapkan pada PT. Indobest Artha Kreasi untuk produk prabayar. Solusi yang dilakukan adalah berdiskusi dan bertanya kepada supervisor dan sesama rekan kerja.