

SKRIPSI

**PENGEMBANGAN *WEBSITE* PENJUALAN PAKAIAN
DENGAN PENERAPAN *APPLICATION PROGRAMMING*
INTERFACE PEMBAYARAN DAN ONGKOS KIRIM**

Ditulis untuk memenuhi sebagian persyaratan
akademik guna memperoleh gelar Sarjana Komputer

Oleh:

NAMA : VINCENT SUKAMTO
NPM : 01081190008



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PELITA HARAPAN
JAKARTA
2023**

**UNIVERSITAS PELITA HARAPAN****PERNYATAAN DAN PERSETUJUAN UNGGAH TUGAS AKHIR**

Saya yang bertanda tangan di bawah ini :

Nama – NPM : Vincent Sukamto - 01081190008
Fakultas : Ilmu Komputer
Program Studi : Sistem Informasi
Lokasi Kampus : Tangerang
Jenis Tugas Akhir : Skripsi
Judul : **“PENGEMBANGAN *WEBSITE* PENJUALAN
PAKAIAN DENGAN PENERAPAN API
PEMBAYARAN DAN ONGKOS KIRIM ”**

Menyatakan bahwa:

1. Tugas akhir tersebut adalah benar karya saya dengan arahan dari dosen pembimbing dan bukan merupakan duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai untuk mendapatkan gelar akademik di perguruan tinggi mana pun.
2. Tugas akhir tersebut bukan merupakan plagiat dari hasil karya pihak lain, dan apabila saya/kami mengutip dari karya orang lain maka akan dicantumkan sebagai referensi sesuai dengan ketentuan yang berlaku.
3. Saya memberikan Hak Non eksklusif Tanpa Royalti kepada Universitas Pelita Harapan atas Tugas Akhir tersebut untuk diunggah ke dalam *Repository* UPH.

Apabila di kemudian hari ditemukan pelanggaran Hak Cipta dan Kekayaan Intelektual atau Peraturan Perundang-undangan Republik Indonesia lainnya dan integritas akademik dalam karya saya/kami tersebut, maka saya/kami bersedia menanggung secara pribadi segala bentuk tuntutan hukum dan sanksi akademis yang timbul serta membebaskan Universitas Pelita Harapan dari segala tuntutan hukum yang berlaku.

Dibuat di : Tangerang
Pada Tanggal : 9 Januari 2023
Yang menyatakan,

Tanda Tangan	
Nama	(Vincent Sukamto)



UNIVERSITAS PELITA HARAPAN
FAKULTAS ILMU KOMPUTER

PERSETUJUAN DOSEN PEMBIMBING TUGAS AKHIR

**PENGEMBANGAN *WEBSITE* PENJUALAN PAKAIAN DENGAN
PENERAPAN API PEMBAYARAN DAN ONGKOS KIRIM**

Oleh :

Nama : Vincent Sukamto
NPM : 01081190008
Program Studi : Sistem Informasi
Peminatan : *Web & Mobile Programming*

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam Sidang Tugas Akhir guna memperoleh gelar Strata Satu Sarjana Komputer (S.Kom) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan, Tangerang, Banten.

Tangerang, 9 Januari 2023

Menyetujui :

Pembimbing Pertama

(Calandra Alencia Haryani,
S.E., S.SI., M.T.I.)

Pembimbing Kedua

(Kusno Prasetya, Ph.D.)

Ketua Program Studi Sistem

Informasi



(Arnold Aribowo, S.T., M.T.)

Dekan Fakultas Ilmu Komputer



(Dr. Eng. Ir. Pujianto Yugopuspito, MSc.)




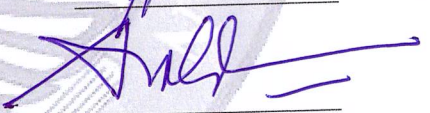
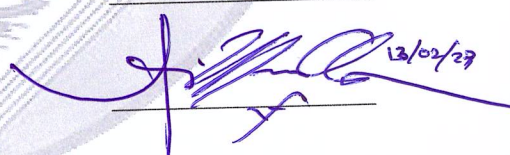
UNIVERSITAS PELITA HARAPAN
FAKULTAS ILMU KOMPUTER

PERSETUJUAN TIM PENGUJI TUGAS AKHIR

Pada hari Kamis, 19 Januari 2023 telah diselenggarakan Sidang Tugas Akhir mahasiswa/i dengan keterangan:

Nama : Vincent Sukamto
NPM : 01081190008
Judul : "PENGEMBANGAN WEBSITE PENJUALAN
PAKAIAN DENGAN PENERAPAN API
PEMBAYARAN DAN ONGKOS KIRIM"

Sebagai persyaratan akademik untuk mencapai gelar Strata Satu Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan, Tangerang, Banten, yang diuji oleh tim penguji yang terdiri dari:

Nama	Status	Tanda tangan
Kusno Prasetya, Ph.D.	sebagai ketua sidang	
Arnold Aribowo, S.T., M.T.	sebagai anggota	
Aditya Rama Mitra, S.Si., M.T.	sebagai anggota	 13/02/23

ABSTRAK

Vincent Sukamto (01081190008)

PENGEMBANGAN *WEBSITE* PENJUALAN PAKAIAN DENGAN PENERAPAN *APPLICATION PROGRAMMING INTERFACE* PEMBAYARAN DAN ONGKOS KIRIM

(XV + 79 halaman; 83 gambar; 15 tabel; 5 lampiran)

Aktivitas belanja melalui toko *online* adalah hal yang kian lumrah di kalangan masyarakat. Salah satu keuntungan yang dapat dinikmati masyarakat dari aktivitas berbelanja dengan mode daring ini adalah tersedianya beragam cara pembayaran praktis yang non konvensional seperti penggunaan dompet digital. Namun saat ini masih sedikit pengusaha atau pemilik UMKM (Usaha Mikro Kecil dan Menengah) di Indonesia yang belum mengoperasikan bisnisnya secara daring. Berkaitan dengan bisnis toko *online*, Levinsky Indonesia (LI) adalah salah satu UMKM yang bergerak di industri pakaian yang membutuhkan digitalisasi untuk mempermudah proses bisnisnya. Sebelumnya, proses bisnis LI masih dilakukan secara konvensional. Hal ini ditandai dengan masih berlakunya transaksi penjualan secara manual.

Untuk mencapai digitalisasi tersebut, LI mempertimbangkan perlunya pengembangan sistem informasi yang memanfaatkan API (*Application Programming Interface*) untuk menghubungkan aplikasi dengan fungsi sistem. Dua API yang dipakai dalam sistem usulan ini adalah API Midtrans yang digunakan untuk memverifikasi pembayaran dan API RajaOngkir untuk mendapatkan perkiraan biaya pengiriman. Pengembangan sistem usulan ini menggunakan metodologi SDLC yang meliputi tahapan perencanaan, analisis, desain dan implementasi. Pengembangan ini juga menggunakan metode *waterfall*, dimana pengembangan sistem direncanakan secara berurutan.

Hasil dari implementasi sistem usulan ini menunjukkan dampak positif sebagaimana diindikasikan melalui peningkatan proses transaksi yang diverifikasi khususnya melalui pemanfaatan API Midtrans. Selain itu, aplikasi berbasis *website* yang dikembangkan ini mempermudah pihak manajemen LI dalam mengoperasikan bisnisnya secara *online* dan juga meningkatkan potensi jangkauan pelanggan baru yang lebih luas.

Kata Kunci : digitalisasi, UMKM, API, Midtrans, RajaOngkir, SDLC, *waterfall*

Referensi : 13 (2001-2022)

ABSTRACT

Vincent Sukamto (01081190008)

CLOTHING WEBSITE DEVELOPMENT WITH PAYMENT AND SHIPPING APPLICATION PROGRAMMING INTERFACE IMPLEMENTATION

(XV + 79 pages; 83 figures; 15 table; 5 appendices)

Shopping activities through online stores are increasingly common among the public. One of the advantages that people can enjoy from online shopping activities is the availability of a variety of practical, non-conventional payment methods like using digital wallets. However, there are still very few entrepreneurs or UMKM owners (Micro, Small and Medium Enterprises) in Indonesia who have not operated their businesses online. Regarding the online shop business, Levinsky Indonesia (LI) is one of the UMKMs engaged in the clothing industry that requires digitization to simplify its business processes. Previously, LI's business processes were done conventionally (manual sales transactions).

To achieve this digitization LI considered developing an information system that utilizes the API (Application Programming Interface) to connect applications with system functions. The two APIs used in this proposed system are the Midtrans API to verify payments and the RajaOngkir API to obtain estimated shipping costs. The development of this proposed system uses the SDLC methodology that includes the stages of planning, analysis, design, and implementation. This development also uses the waterfall method, where system development is planned sequentially.

The results of the implementation of this proposed system show a positive impact indicated by the improvement of the verified transaction process, mainly through the use of the Midtrans API. In addition, the developed website-based application makes it easier for LI management to operate their business online and increases the potential for a wide range of new customers.

Keywords : digitalization, UMKM, API, Midtrans, RajaOngkir, SDLC, *waterfall*

Reference : 13 (2001-2022)

KATA PENGANTAR

Puji syukur kepada Tuhan yang Maha Esa atas segala rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Pengembangan *Website* Penjualan Pakaian Dengan Penerapan *Application Programming Interface* Pembayaran Dan Ongkos Kirim”.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan akademik guna memperoleh gelar Sarjana Komputer Strata Satu Program Studi Sistem Informasi Universitas Pelita Harapan, Karawaci, Tangerang.

Penulis menyadari bahwa tanpa bimbingan, bantuan dan doa dari berbagai pihak, Tugas Akhir ini tidak akan dapat diselesaikan tepat pada waktunya. Maka dari itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam proses penyusunan tugas akhir ini, yaitu kepada:

- 1) Tuhan Yesus Kristus, yang telah menyertai dan memberikan akal budi dalam setiap segi kehidupan penulis.
- 2) Bapak Dr.Eng. Ir. Pujiyanto Yugopuspito, MSc. selaku Dekan Fakultas Ilmu Komputer.
- 3) Bapak Arnold Aribowo, S.T., M.T., selaku Ketua Program Studi Sistem Informasi.
- 4) Ibu Calandra Alencia Haryani, S.SI., S.E., M.T.I. selaku Dosen Pembimbing Utama yang telah meluangkan waktu, tenaga, pikiran, dan dukungan, serta memberikan arahan dan banyak masukan kepada Penulis selama proses pembuatan Tugas Akhir.
- 5) Bapak Kusno Prasetya, Ph.D. selaku Dosen Pembimbing Kedua yang telah meluangkan waktu, tenaga, pikiran, dan dukungan, serta memberikan arahan dan banyak masukan kepada penulis selama proses pembuatan Tugas Akhir.
- 6) Seluruh dosen yang telah mengajar dan memberi nasihat kepada penulis selama menempuh pendidikan di Program Studi Sistem Informasi, Universitas Pelita Harapan.
- 7) Seluruh staf Fakultas Ilmu Komputer Universitas Pelita Harapan yang telah membantu penulis dalam kegiatan administratif.
- 8) Kedua orang tua serta kedua kakak penulis yang telah mendoakan dan mendukung penulis dalam menyelesaikan Tugas Akhir.
- 9) Teman terkasih penulis, Michelyn Niononi yang telah mendoakan, mendukung, membantu, dan memotivasi penulis selama pembuatan Tugas Akhir.
- 10) Sahabat penulis: Jason Otto, Erick Gennady, Gurnita Ambara, Kentvink Teofilos, Glenn Silver, Catherine Pratama, Clara Yauw, Leswanti Tania dan Devi Pundarika yang telah memberikan semangat, hiburan serta mendoakan penulis selama pembuatan Tugas Akhir.

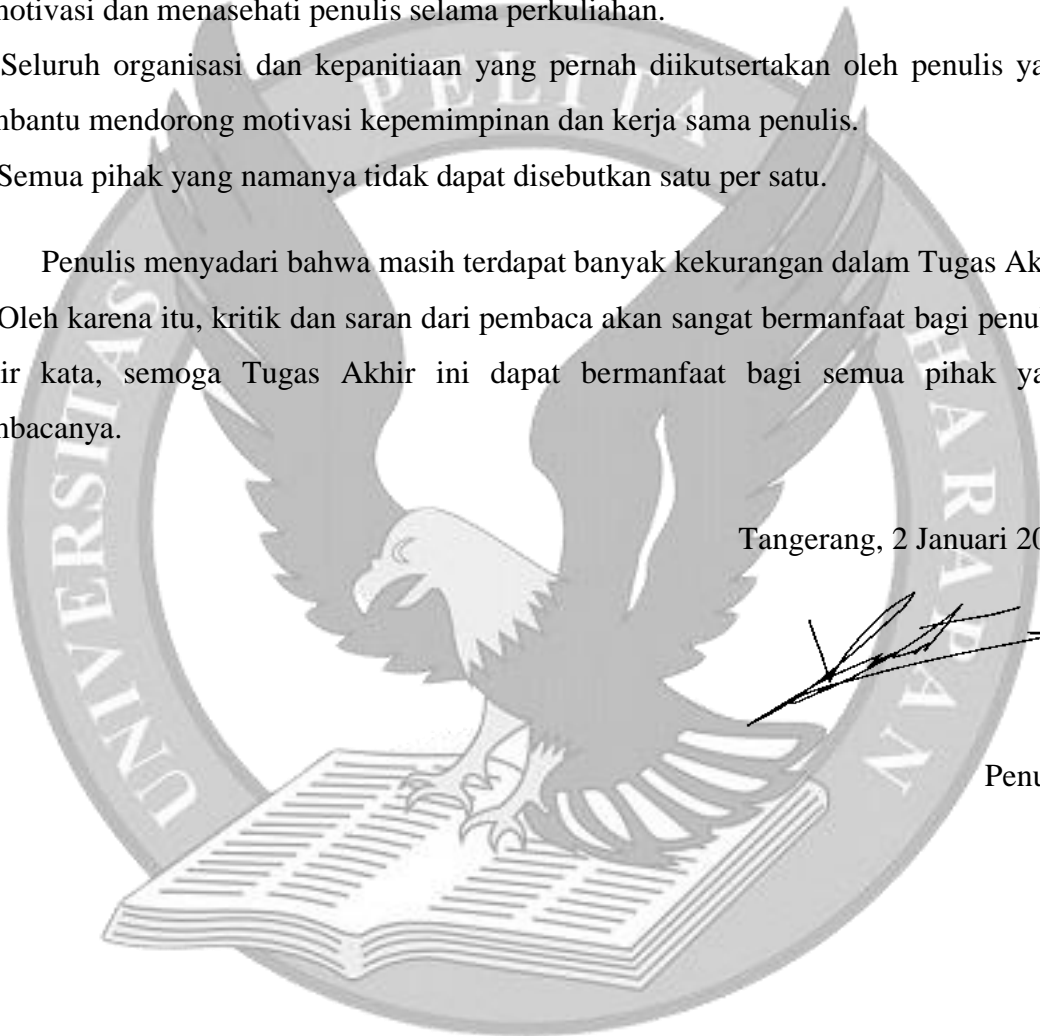
- 11) Teman seangkatan penulis: Albert Steven, Yofan Wijaya, Benny Ebenheizer, Philip Urbanus, Filbert Chan, Adventius Marlie, Davin Feriyadi, Ignasius Migel, Winnie Veronica, Wulan Frederika, Ernest Isthien, David Anthony dan Carlo Toorop yang telah menyemangati dan memotivasi penulis dalam pembuatan Tugas Akhir.
- 12) Mentor penulis selama perkuliahan, Josiah Hans Sujudi yang telah memotivasi dan menasehati penulis selama pembuatan Tugas Akhir.
- 13) Kakak tingkat penulis selama perkuliahan, Valeri Monica, Christian Marsel, Excelcius Roni, Vincent Willington, Abigail Ruth, dan Helbert Jonathan yang telah memotivasi dan menasehati penulis selama perkuliahan.
- 14) Seluruh organisasi dan kepanitiaan yang pernah diikutsertakan oleh penulis yang membantu mendorong motivasi kepemimpinan dan kerja sama penulis.
- 15) Semua pihak yang namanya tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam Tugas Akhir ini. Oleh karena itu, kritik dan saran dari pembaca akan sangat bermanfaat bagi penulis. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang membacanya.

Tangerang, 2 Januari 2023



Penulis



DAFTAR ISI

PERNYATAAN DAN PERSETUJUAN UNGGAH TUGAS AKHIR.....	i
PERSETUJUAN DOSEN PEMBIMBING TUGAS AKHIR.....	ii
PERSETUJUAN TIM PENGUJI TUGAS AKHIR.....	iii
ABSTRAK.....	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan dan Manfaat Penelitian.....	2
1.5 Metodologi Penelitian	2
1.5.1 Metodologi Pengumpulan Data	3
1.5.2 Metodologi Pengembangan Sistem.....	3
1.6 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	4
2.1 Kajian Teori.....	4
2.1.1 Usaha Mikro Kecil dan Menengah (UMKM).....	4
2.1.2 <i>Application Programming Interface</i> (API).....	4
2.1.3 <i>JavaScript Object Notation</i> (JSON)	4
2.1.4 Pengertian Aplikasi Berbasis <i>Web</i>	5
2.1.5 <i>Hypertext Markup Language</i> (HTML).....	5
2.1.6 <i>Cascading Style Sheet</i> (CSS).....	5

2.1.7 Hypertext Preprocessor (PHP)	5
2.1.8 Javascript.....	6
2.1.9 Basis Data (<i>Database</i>).....	6
2.1.10 CodeIgniter	6
2.1.11 <i>Model View Controller (MVC)</i>	6
2.1.12 <i>System Development Life Cycle (SDLC)</i>	7
2.1.13 Metode <i>Waterfall</i>	8
2.1.14 <i>Activity Diagram</i>	8
2.1. 15 <i>Use Case Diagram</i>	10
2.2 Penelitian Terdahulu	11
BAB III SISTEM SAAT INI	13
3.1 Profil Perusahaan	13
3.2 Struktur Organisasi	13
3.3 Analisis Sistem Saat Ini.....	14
3.4 Kendala Sistem Saat Ini	16
BAB IV ANALISIS DAN PERANCANGAN SISTEM USULAN.....	17
4.1 Tahap Perencanaan Sistem.....	17
4.1.1 Kelayakan Teknis (<i>Technical Feasibility</i>).....	17
4.1.2 Kelayakan Ekonomi (<i>Economy Feasibility</i>).....	17
4.1.3 Kelayakan Organisasi (<i>Organization Feasibility</i>)	18
4.2 Tahap Analisis	18
4.2.1 Pengumpulan Kebutuhan Pengguna (<i>User Requirements Gathering</i>).....	18
4.2.2 Pemodelan Fungsional	21
4.2.3 Pemodelan Struktural	32
4.3 Tahap Perancangan	39
4.3.1 Kerangka Sistem (<i>Wireframe</i>)	39
4.3.2 Lapisan Antarmuka (<i>User Interface</i>)	50
4.3.3 Penggunaan API Pihak Ketiga.....	71
4.4 Tahap Pengujian	74

BAB V KESIMPULAN DAN SARAN	76
5.1 Kesimpulan.....	76
5.2 Saran.....	76
DAFTAR PUSTAKA	78



DAFTAR GAMBAR

Gambar 3.1 Struktur Organisasi Levinsky Indonesia	13
Gambar 3.2 <i>Activity Diagram</i> Sistem Saat Ini	15
Gambar 3.3 Pencatatan Pesanan Levinsky dengan Excel (Secara Manual)	16
Gambar 4.1 <i>Use Case Diagram</i> Levinsky	22
Gambar 4.2 <i>Activity Diagram</i> Register Levinsky	23
Gambar 4.3 <i>Activity Diagram</i> Login Levinsky	24
Gambar 4.4 <i>Activity Diagram</i> Forget Password Levinsky.....	25
Gambar 4.5 <i>Activity Diagram</i> Cart Levinsky	26
Gambar 4.6 <i>Activity Diagram</i> Manage Address Levinsky	27
Gambar 4.7 <i>Activity Diagram</i> Receive Notification Levinsky.....	28
Gambar 4.8 <i>Activity Diagram</i> Checkout Levinsky.....	29
Gambar 4.9 <i>Activity Diagram</i> Manage Product Admin Levinsky	30
Gambar 4.10 <i>Activity Diagram</i> Manage Transaction Admin Levinsky	31
Gambar 4.11 <i>Activity Diagram</i> Send Notification Admin Levinsky	32
Gambar 4.12 <i>Class Diagram</i> Levinsky	33
Gambar 4.13 <i>Table Relational Diagram</i> Levinsky	34
Gambar 4.14 <i>Wireframe</i> Halaman Utama User Levinsky	39
Gambar 4.15 <i>Wireframe</i> Login Sidebar User Levinsky	40
Gambar 4.16 <i>Wireframe</i> Register Sidebar User Levinsky	40
Gambar 4.17 <i>Wireframe</i> Halaman Utama Member Levinsky.....	41
Gambar 4.18 <i>Wireframe</i> Menu Sidebar Member Levinsky	41
Gambar 4.19 <i>Wireframe</i> Halaman Change Password Member Levinsky.....	42
Gambar 4.20 <i>Wireframe</i> Halaman View Address Member Levinsky.....	42
Gambar 4.21 <i>Wireframe</i> Halaman Create Address Member Levinsky	43
Gambar 4.22 <i>Wireframe</i> Notification Sidebar Member Levinsky	43
Gambar 4.23 <i>Wireframe</i> Halaman Notification Member Levinsky	44
Gambar 4.24 <i>Wireframe</i> Halaman Product Levinsky	44
Gambar 4.25 <i>Wireframe</i> Cart Sidebar Member Levinsky.....	45
Gambar 4.26 <i>Wireframe</i> Halaman Cart Member Levinsky.....	45
Gambar 4.27 <i>Wireframe</i> Halaman Transaction Member Levinsky	46
Gambar 4.28 <i>Wireframe</i> Halaman Login Admin Levinsky.....	46
Gambar 4.29 <i>Wireframe</i> Halaman Dashboard Admin Levinsky	47
Gambar 4.30 <i>Wireframe</i> Halaman Product Admin Levinsky	47
Gambar 4.31 <i>Wireframe</i> Halaman View Product Detail Admin Levinsky	48
Gambar 4.32 <i>Wireframe</i> Halaman Transaction Admin Levinsky	48
Gambar 4.33 <i>Wireframe</i> Halaman View Transaction Detail Admin Levinsky.....	49
Gambar 4.34 <i>Wireframe</i> Halaman Blast Notification Admin Levinsky	49
Gambar 4.35 <i>Wireframe</i> Halaman Notification Admin Levinsky	50
Gambar 4.36 Halaman Utama Levinsky	51

Gambar 4.37 <i>Sidebar Login</i> Levinsky	52
Gambar 4.38 <i>Sidebar Register</i> Levinsky	52
Gambar 4.39 Aktivasi Akun Levinsky pada Email	53
Gambar 4.40 Halaman Utama <i>Member (Logged In)</i> Levinsky	54
Gambar 4.41 <i>Sidebar Menu Member</i> Levinsky	55
Gambar 4.42 Halaman <i>Change Password Member</i> Levinsky	55
Gambar 4.43 Validasi Kelengkapan <i>Form</i> pada Halaman <i>Change Password</i> Levinsky	56
Gambar 4.44 Validasi Kecocokan <i>Password</i> pada <i>Form Change Password</i> Levinsky	56
Gambar 4.45 Validasi <i>Password</i> Lama pada <i>Form Change Password</i> Levinsky	56
Gambar 4.46 <i>Forget Password</i> pada <i>Email</i>	57
Gambar 4.47 Halaman <i>View Address Member</i> Levinsky	58
Gambar 4.48 Halaman <i>Create Address Member</i> Levinsky	58
Gambar 4.49 Validasi Kelengkapan <i>Form</i> pada Halaman <i>Create Address</i> Levinsky	59
Gambar 4.50 Halaman <i>Shop</i> Levinsky	59
Gambar 4.51 Validasi Pemilihan Varian Produk pada Aktivitas <i>Insert Cart</i> Levinsky	59
Gambar 4.52 Validasi Ketersediaan Produk pada Aktivitas <i>Insert Cart</i> Levinsky	60
Gambar 4.53 Produk Berhasil Ditambahkan pada Aktivitas <i>Insert Cart</i> Levinsky	60
Gambar 4.54 <i>Sidebar Cart Member</i> Levinsky	61
Gambar 4.55 Halaman <i>Cart Member</i> Levinsky	61
Gambar 4.56 Notifikasi Ketersediaan <i>Product Cart</i> Levinsky	62
Gambar 4.57 Halaman <i>Checkout</i> API Midtrans	62
Gambar 4.58 Pembayaran <i>Credit Card</i> API Midtrans	63
Gambar 4.59 OTP Pembayaran API Midtrans	63
Gambar 4.60 Pembayaran Berhasil API Midtrans	63
Gambar 4.61 Notifikasi Transaksi Berhasil Levinsky	64
Gambar 4.62 Halaman Notifikasi <i>Member</i> Levinsky	64
Gambar 4.63 Halaman <i>Login Admin</i> Levinsky	65
Gambar 4.64 Validasi <i>Form Login Admin</i> Levinsky	65
Gambar 4.65 Halaman <i>Dashboard Admin</i> Levinsky	66
Gambar 4.66 Halaman <i>Product Admin</i> Levinsky	66
Gambar 4.67 Validasi Mengaktifkan atau Menonaktifkan Produk <i>Admin</i> Levinsky	67
Gambar 4.68 Halaman <i>Add Product Admin</i> Levinsky	67
Gambar 4.69 Validasi Kelengkapan <i>Form</i> Tambah Produk <i>Admin</i> Levinsky	68
Gambar 4.70 Halaman <i>Edit Product Admin</i> Levinsky	68
Gambar 4.71 Validasi Hapus Produk <i>Admin</i> Levinsky	69
Gambar 4.72 Halaman <i>Transaction Admin</i> Levinsky	69
Gambar 4.73 Halaman Detail Transaksi <i>Admin</i> Levinsky	70
Gambar 4.74 Validasi Penerimaan atau Penolakan Transaksi <i>Admin</i> Levinsky	70
Gambar 4.75 Halaman <i>Blast Notification Admin</i> Levinsky	70
Gambar 4.76 Validasi <i>Form Blast Notification Admin</i> Levinsky	71
Gambar 4.77 <i>Popup Feedback</i> Transaksi <i>Admin</i> Levinsky	71
Gambar 4.78 Mengirim Data <i>Cart</i> Kepada API Midtrans	72

Gambar 4.79 Menampilkan <i>Snap UI</i> Midtrans untuk Pembayaran.....	73
Gambar 4.80 Mengirim <i>CURL</i> API RajaOngkir	74



DAFTAR TABEL

Tabel 2.1 Tabel Penjelasan <i>Activity Diagram</i>	8
Tabel 2.2 Tabel Komponen <i>Use Case Diagram</i>	10
Tabel 2.3 Tabel Penelitian Terdahulu.....	11
Tabel 4.1 Daftar Perangkat Keras Dan Perangkat Lunak Minimum Untuk Sistem Usulan Dari Sisi <i>Server</i>	17
Tabel 4.2 Daftar Perangkat Keras Dan Perangkat Lunak Minimum Untuk Sistem Usulan Dari Sisi <i>Client</i>	18
Tabel 4.3 Tabel <i>user Levinsky</i>	35
Tabel 4.4 Tabel <i>address Levinsky</i>	35
Tabel 4.5 Tabel <i>notification Levinsky</i>	36
Tabel 4.6 Tabel <i>cart Levinsky</i>	36
Tabel 4.7 Tabel <i>cart_detail Levinsky</i>	37
Tabel 4.8 Tabel <i>transaction Levinsky</i>	37
Tabel 4.9 Tabel <i>product Levinsky</i>	37
Tabel 4.10 Tabel <i>product_detail Levinsky</i>	38
Tabel 4.11 Tabel <i>product_image Levinsky</i>	38
Tabel 4.12 Tabel <i>Script #1.1 Login User</i>	75



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	A-1
LAMPIRAN B <i>USE CASE DESCRIPTION</i>	B-1
LAMPIRAN C <i>USER ACCEPTANCE TESTING</i>	C-1
LAMPIRAN D DOKUMENTASI	D-1
LAMPIRAN E FORM LEMBAR MONITORING BIMBINGAN TUGAS AKHIR	E-1



BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada masa ini, hampir seluruh kegiatan sehari-hari masyarakat didukung dengan digitalisasi. Digitalisasi merupakan suatu proses bentuk informasi seperti gambar, suara, angka, kata, data, atau gerakan yang dikembangkan dengan pemrograman ke dalam bentuk *binary* sehingga memungkinkan untuk memanipulasi dan mentransformasi data. [1] Perkembangan digitalisasi ini dapat mendukung transformasi perekonomian di Indonesia dan membantu mempermudah kegiatan masyarakat, salah satunya dengan memanfaatkan penggunaan sistem pembayaran digital. [2] Pembayaran digital yang bervariasi juga berperan penting agar pelanggan dapat membayar dengan metode pembayaran favoritnya.

Saat ini, hanya sekitar 17,1% dari 56 juta Usaha Mikro Kecil dan Menengah (UMKM) di Indonesia yang sudah mulai mendigitalisasikan usahanya. [3] Levinsky Indonesia merupakan perusahaan UMKM yang bergerak di sektor pakaian, perusahaan ini membutuhkan digitalisasi untuk memudahkan proses bisnis mereka yang saat ini masih dicatat secara manual. Metode penjualan yang digunakan oleh Levinsky Indonesia adalah dengan melalui *online shopping* seperti WhatsApp, Instagram, dan platform *online chatting* lainnya serta penjualan melalui *e-commerce* seperti Tokopedia dan Shopee, membuat munculnya permasalahan dengan kekeliruan data transaksi yang dicatat secara manual.

Seperti teknologi lainnya pembayaran digital terus berkembang, saat ini berbagai platform sistem pembayaran telah menerapkan API (*Application Programming Interface*) yang dapat digunakan untuk mengelola pembayaran yang bervariasi baik dari pembayaran kartu kredit, *online payment* (Gopay, OVO, dll.) dan kartu debit dengan akun virtual yang disediakan dari API tersebut. Salah satunya adalah dengan aplikasi Midtrans. Midtrans merupakan suatu platform digital yang dapat memverifikasi pembayaran dengan memanfaatkan API yang telah disediakan. Selain itu untuk mendukung perhitungan ongkos kirim yang lebih akurat diperlukan juga API pengiriman RajaOngkir yang dapat mengestimasi lebih tepat terhadap ongkos yang harus ditagihkan.

Dalam proses digitalisasi usaha ini akan menggunakan sistem aplikasi berbasis *web* yang terintegrasi dengan API Midtrans dan RajaOngkir, sehingga diharapkan dapat mengubah proses bisnis Levinsky Indonesia menjadi lebih efisien dan dapat

meminimalkan tingkat kekeliruan dalam pembayaran dan perhitungan biaya ongkos kirim.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah:

- 1) Bagaimana perancangan ulang sistem informasi untuk Levinsky Indonesia?
- 2) Bagaimana sistem usulan dapat mempermudah proses transaksi pada Levinsky Indonesia?
- 3) Bagaimana API pembayaran dan pengiriman dapat diintegrasikan pada sistem usulan Levinsky Indonesia?

1.3 Batasan Masalah

Batasan masalah dalam pembuatan penelitian ini:

- 1) API Midtrans pada tugas akhir ini hanya menggunakan fitur *sandbox*
- 2) API RajaOngkir pada tugas akhir ini hanya menggunakan tipe akun *starter*
- 3) Beberapa tampilan seperti *carousell* di halaman utama, *feedback* pelanggan dan beberapa tampilan tertentu tidak mencakup *content management system*.

1.4 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah:

Mengembangkan sistem informasi berbasis *website* yang memanfaatkan API pembayaran menggunakan Midtrans dan pengiriman menggunakan RajaOngkir dalam proses transaksi.

Manfaat dari penelitian ini adalah:

Mempermudah perusahaan dalam melakukan transaksi yang dilakukan secara *online* melalui aplikasi berbasis *website* yang terintegrasi dengan API pembayaran menggunakan Midtrans dan pengiriman menggunakan RajaOngkir.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam Tugas Akhir ini yaitu dengan metodologi pengumpulan data dan metodologi pengembangan sistem.

1.5.1 Metodologi Pengumpulan Data

Metodologi pengumpulan data yang digunakan yakni menggunakan metode observasi guna mengetahui semua fitur yang akan diimplementasikan dalam sistem sesuai dengan harapan dari sisi pengguna.

1.5.2 Metodologi Pengembangan Sistem

Metodologi yang digunakan dalam pengembangan sistem usulan ini adalah dengan metode *System Development Life Cycle* (SDLC) dengan menggunakan model *waterfall*. Dalam metode ini terdapat 4 fase, yaitu; *planning*, *analysis*, *design*, dan *implementation*. Keuntungan yang didapat dari penggunaan model *waterfall* adalah ketepatan dalam proses perancangan sistem usulan karena proses lebih berurutan.[4]

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan dari penelitian ini.

BAB 1: PENDAHULUAN

Berisi tentang latar belakang, rumusan dan batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2: LANDASAN TEORI

Berisi tentang teori-teori dasar yang dikutip dari buku serta beberapa jurnal yang berhubungan dengan penelitian ini.

BAB 3: SISTEM SAAT INI

Berisi pembahasan mengenai sistem yang ada saat ini, dan permasalahan yang dihadapi dalam sistem yang telah ada.

BAB 4: ANALISIS DAN PERANCANGAN SISTEM USULAN

Bab ini berisi pembahasan mengenai tahap perencanaan yang berisi identifikasi dan analisis kelayakan sistem yang akan dibangun, tahap analisis dengan *use case diagram*, *activity diagram*, dan *class diagram*, juga tahap perancangan yang terdiri dari *table relationship diagram* dan perancangan *interface*.

BAB 5: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan serta saran mengenai perancangan sistem yang telah dilakukan.

BAB II

LANDASAN TEORI

2.1 Kajian Teori

Berikut merupakan kajian teori berdasarkan literatur yang berkaitan dengan tema penelitian ini.

2.1.1 Usaha Mikro Kecil dan Menengah (UMKM)

Dalam Peraturan Pemerintah No.7 tahun 2021 yang membahas mengenai Kemudahan, Perlindungan, dan Pemberdayaan Koperasi dan Usaha Mikro, Kecil dan Menengah (UMKM), Usaha Mikro Kecil dan Menengah (UMKM) adalah suatu usaha produktif yang dimiliki oleh perorangan atau badan usaha perorangan dengan hasil pendapatan atau kekayaan bersih tahunan yang telah memenuhi kriteria usaha mikro. UMKM juga berperan penting dalam kemajuan perekonomian di Indonesia dengan terciptanya lapangan kerja dan jumlah usaha yang meningkat [5].

2.1.2 *Application Programming Interface* (API)

Application programming interface (API) adalah susunan dokumentasi yang terdiri dari *interface* (tampilan), *function* (fungsi), *class* (kelas), *structure* (struktur) dan lainnya untuk membantu dalam pengembangan *software*. API dapat membantu *programmer* dalam “membongkar” *software* yang selanjutnya dapat diintegrasikan atau dikembangkan dengan *software* lainnya. API juga merupakan penghubung antar aplikasi sehingga dapat membantu *programmer* mengintegrasikan dengan fungsi sistem. Proses ini dijalankan melalui sistem operasi [6].

2.1.3 *JavaScript Object Nation* (JSON)

JSON (*JavaScript Object Nation*) merupakan bentuk pertukaran data yang mudah dikenali, mudah tafsir dan mudah dibentuk oleh komputer atau sistem. JSON adalah salah satu susunan teks yang dapat digunakan oleh bahasa *programming* apa saja karena JSON menggunakan bentuk bahasa yang mudah dipahami oleh *programmer*. JSON juga merupakan bagian dari bahasa pemrograman *JavaScript*. Hal ini membuat JSON sebagai bahasa pemrograman yang ideal untuk bertukar data [6].

2.1.4 Pengertian Aplikasi Berbasis Web

Web Application atau biasa disebut aplikasi berbasis *website* merupakan aplikasi yang berjalan pada *web browser* yang dapat tersusun dari sejumlah bahasa pemrograman seperti *Hypertext Preprocessor* (PHP), *Hypertext Markup Language* (HTML), *Javascript*, *Cascading Style Sheet* (CSS), dan lainnya. Aplikasi berbasis *web* menyediakan *resource* yang spesifik seperti layanan dan konten yang disalurkan pada *user interface* dan *web browser*. Aplikasi *website* yang dinamis memiliki proses yang cukup kompleks, sehingga *user* dapat memberi sejumlah perintah yang dikehendaki kemudian akan diproses di *server* dan mengembalikan tampilan yang sesuai dengan permintaan pengguna [6].

2.1.5 Hypertext Markup Language (HTML)

HTML merupakan salah satu bahasa *programming* yang digunakan dalam pembuatan halaman *website* untuk menunjukkan informasi yang dapat berupa gambar maupun tulisan. Hingga saat ini HTML masih terus dikembangkan karena penggunaan internet terus berkembang dan semakin banyak. Oleh sebab itu, HTML masih dapat dikembangkan lagi untuk menciptakan halaman *website* yang lebih berkualitas. *World Wide Web Consortium* (W3C) merupakan organisasi yang dibentuk untuk bertanggung jawab dalam pengembangan bahasa HTML [6].

2.1.6 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) merupakan pengatur tampilan dokumen HTML pada halaman *website*, CSS dapat mengatur teks, warna, jarak antar baris dan format border hingga tampilan gambar pada halaman *website*. *Cascading Style Sheets* (CSS) merupakan salah satu bahasa *programming* pendukung dalam penyusunan *website* untuk mengatur tampilan agar lebih terstruktur dan lebih menarik. Sama layaknya HTML, CSS juga dikembangkan oleh W3C yang bertujuan untuk mempermudah proses pemrograman halaman *web* [6].

2.1.7 Hypertext Preprocessor (PHP)

PHP merupakan salah satu bahasa pemrograman *website* yang berbentuk skrip dan dapat disisipkan dalam *file* HTML. PHP juga sering digunakan untuk pemrograman *website* yang dinamis. PHP menjalankan skrip pada sisi *server* sehingga disebut juga sebagai *server side HTML embedded scripting*. Pengembangan halaman *website* dapat

dimanfaatkan untuk pengembangan bahasa PHP sendiri, digabungkan dengan kode HTML atau digunakan dengan berbagai *framework* [7].

2.1.8 Javascript

Javascript merupakan bahasa pemrograman berbasis skrip (*scripting language*). *Javascript* berisikan serangkaian instruksi berguna untuk mengontrol sistem operasi. *Javascript* dikembangkan untuk berjalan pada *server* maupun dari sisi *client*. *Javascript* dapat membuat beban *server* menjadi lebih ringan dan mempercepat respon halaman *web*. *JavaScript* juga termasuk bahasa pemrograman *case sensitive* yang sangat sensitif dalam penamaan variabel dan fungsi seperti penggunaan huruf besar maupun huruf kecil [6].

2.1.9 Basis Data (Database)

Basis data (*database*) adalah serangkaian data yang disimpan di dalam komputer atau *software* yang kemudian dapat diproses atau dimanipulasi dan kemudian ditampilkan menjadi sebuah informasi [8]. Pengembangan sistem usulan ini akan menggunakan *Framework CodeIgniter* dan MariaDB dalam penyimpanan basis data.

2.1.10 CodeIgniter

CodeIgniter adalah sebuah *framework* pemrograman *website* yang berbasis bahasa pemrograman PHP dan dapat digunakan untuk mengembangkan *website* lebih cepat. *Framework* sendiri merupakan suatu struktur pustaka-pustaka, kelas-kelas dan infrastruktur *run-time* yang telah tersusun dan dapat digunakan oleh *programmer* untuk mengembangkan aplikasi berbasis *website* dengan cepat. Pola desain *CodeIgniter* dalam pengembangan *website* menggunakan pola MVC (*Model View Controller*). Hal ini memisahkan *business logic* dan tampilan aplikasi sehingga *web programmer* dan *web designer* dapat bekerja secara terpisah [9].

2.1.11 Model View Controller (MVC)

Model View Controller (MVC) adalah sarana pengorganisasian situs *website* yang dinamis. Desain ini sudah ada sejak 1979 dan pertama kali dijelaskan oleh orang Norwegia, Trygve Reenskaug [9]. Berikut merupakan penjelasannya:

2.1.11.1 Models

Models merupakan *class* PHP yang didesain untuk bekerja dengan informasi yang ada di dalam basis data seperti operasi menambah, mengubah, menghapus dan menampilkan informasi yang ada di dalam basis data [9].

2.1.11.2 Views

Views bertanggung jawab terhadap tampilan informasi yang diperoleh dari basis data/ *models*. Pada konsep MVC, *views* merupakan bagian presentasi yang berisi kode HTML yang dapat mengubah tampilan sesuai aplikasi yang ingin dibuat [9].

2.1.11.3 Controller

Controller merupakan jantung dari aplikasi *website* yang dibuat, *controller* berisikan logika bisnis yang menghubungkan *models* untuk mengambil informasi dari basis data dan kemudian ditampilkan melalui *views* [9].

2.1.12 System Development Life Cycle (SDLC)

System Development Life Cycle merupakan serangkaian proses sistem informasi untuk mendukung kebutuhan bisnis dengan rancangan sebuah sistem, pembangunan sistem hingga mengantarkan kepada *user*. SDLC sendiri memiliki empat tahapan dasar yang penting yakni tahap *planning*, *analysis*, *design*, dan *implementation*. Setiap tahapan yang ada memiliki tonggak pencapaian yang menghasilkan *deliverable* atau produk dari proses pengembangannya [4].

2.1.12.1 Tahapan Perencanaan (Planning)

Tahap ini merupakan proses untuk mengerti tujuan dibuatnya sebuah sistem dan menentukan bagaimana sistem akan dibangun. Tahap perencanaan terdiri dari inisiasi proyek dan pengelolaan proyek. Inisiasi proyek merupakan penjelasan tentang bagaimana sistem informasi membantu memenuhi kebutuhan bisnis. Pengelolaan proyek merupakan adanya penentuan rencana kerja, pembagian tugas, dan arahan agar proyek dapat dilaksanakan dengan baik [4].

2.1.12.2 Tahapan Analisis (Analysis)

Tahap ini menentukan pihak mana yang menggunakan sistem, hal yang dilakukan oleh sistem, dan bagaimana sistem akan digunakan. Analisis akan dilakukan mengenai sistem yang sudah ada dan berguna untuk mengembangkan sistem yang baru. Tiga tahap analisis meliputi analisis sistem yang telah ada dan membangun sistem baru, *requirement gathering*, dan penyusunan proposal sistem [4].

2.1.12.3 Tahapan Desain (Design)

Pada tahap ini, arsitektur teknis serta desain model akan dibuat. akan dikembangkan strategi perancangan, *basic architecture*, spesifikasi *database*, dan

perancangan program, yang mana akan menghasilkan spesifikasi sistem yang akan digunakan dalam pemrograman sistem [4].

2.1.12.4 Tahapan Implementasi (*Implementation*)

Pada tahap ini, sistem dibangun sejalan dengan spesifikasi sistem pada tahap desain. Terdapat tiga langkah pada tahap implementasi, yaitu *system construction*, *installation*, dan *support plan*. Pertama, *system construction* akan dibangun dan melalui proses pengujian, untuk memastikan bahwa sistem sesuai dengan yang direncanakan. Selanjutnya dilakukan *support plan* untuk membahas sistem yang telah dibangun dan melalui proses pengujian untuk mengidentifikasi apa yang perlu diubah pada sistem [4].

2.1.13 Metode *Waterfall*



Metode *waterfall* merupakan salah satu model *System Development Life Cycle* (SDLC) yang sering digunakan untuk pengembangan *software*. Kelebihan dari metode *waterfall* yakni pada proses yang lebih terstruktur dan berurutan sehingga membuat pengerjaan *software* dapat lebih teratur dan terfokuskan. Hal ini membuat penjadwalan untuk mencapai target menjadi lebih pasti dan proses pemeliharaannya menjadi lebih mudah. Namun metode *waterfall* juga memiliki kelemahan yakni cenderung bersifat kaku sehingga sulit untuk melakukan perbaikan atau perubahan pada sistem. Proses harus dimulai dari awal lagi jika terdapat kekurangan pada sistem [4].

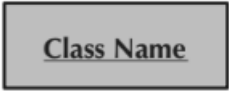




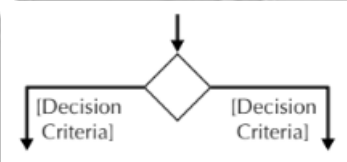
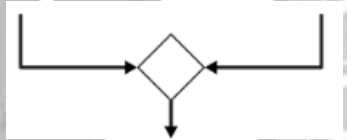
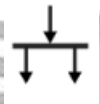
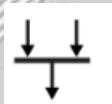

2.1.14 *Activity Diagram*

Activity diagram merupakan instrumen yang digunakan untuk memodelkan perilaku dan kegiatan dalam proses bisnis dengan objek. *Activity diagram* dapat digunakan untuk memodelkan dari cakupan bisnis tingkat tinggi yang prosesnya melibatkan kasus yang berbeda beda, hingga detail kasus penggunaan individu, semuanya berjalan ke rincian secara spesifik dari metode individu [4].

Komponen pembentuk *activity diagram* terdiri dari [4]:

Tabel 2.1 Tabel Penjelasan *Activity Diagram*



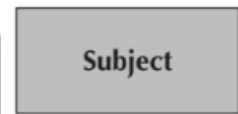

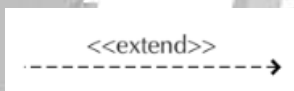
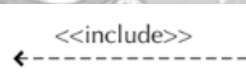

<i>Action:</i> Merupakan tindakan yang sederhana yang tidak dapat diurai.	
<i>Activity:</i> Berguna untuk mewakili kegiatan tertentu.	

<p><i>Object Node:</i></p> <p>Berfungsi untuk merepresentasikan sebuah objek yang terhubung dengan sekumpulan objek yang mengalir.</p>	
<p><i>Control Flow:</i></p> <p>Menunjukkan urutan eksekusi.</p>	
<p><i>Object Flow:</i></p> <p>Menunjukkan aliran suatu objek dari satu aktivitas ke aktivitas lain.</p>	
<p><i>Initial Node:</i></p> <p>Menggambarkan awal dari serangkaian tindakan atau aktivitas.</p>	
<p><i>Final Activity Node:</i></p> <p>Digunakan untuk menghentikan semua aliran kontrol dan aliran objek dalam suatu aktivitas.</p>	
<p><i>Decision Node:</i></p> <p>Digunakan untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalan setelah mewakili kondisi pengujian.</p>	
<p><i>Merge Node:</i></p> <p>Digunakan untuk menyatukan kembali jalur keputusan yang berbeda yang dibuat menggunakan simpul keputusan.</p>	
<p><i>Fork Node:</i></p> <p>Digunakan untuk membagi perilaku menjadi serangkaian aktivitas paralel atau bersamaan.</p>	
<p><i>Join Node:</i></p> <p>Digunakan untuk menyatukan kembali serangkaian aktivitas paralel atau bersamaan</p>	
<p><i>Swimlane:</i></p> <p>Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan aktivitas individu kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas</p>	

2.1.15 Use Case Diagram

Use Case Diagram merupakan diagram yang mencakup kebutuhan bisnis terhadap sistem dan berfungsi sebagai gambaran interaksi antara sistem beserta lingkungannya [4]. Komponen pembentuk *use case diagram* terdiri dari [4]:

Tabel 2.2 Tabel Komponen *Use Case Diagram*

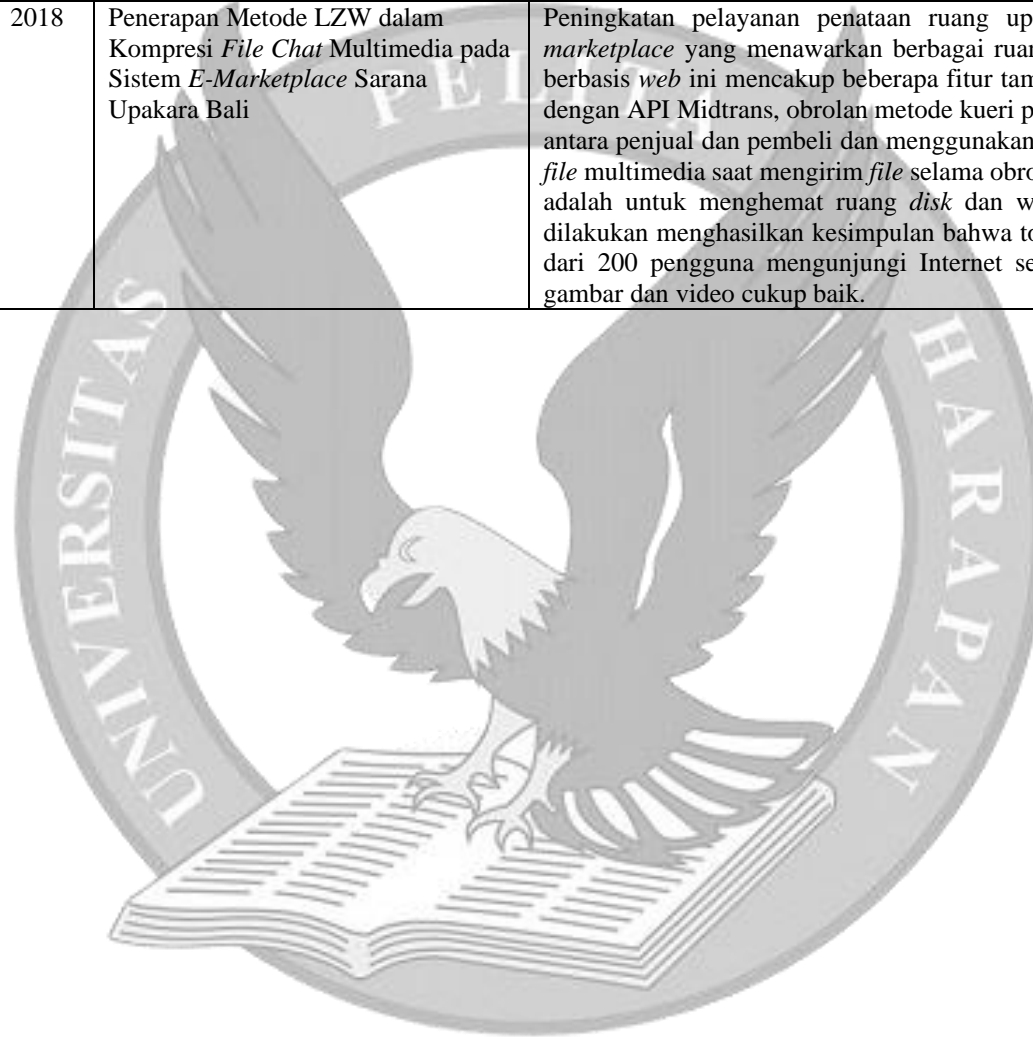
<p><i>Actor:</i> Menggambarkan pihak-pihak yang berperan dalam sistem.</p>	 Actor/Role
<p><i>Use case:</i> Merupakan aktivitas atau sarana yang disiapkan oleh sistem.</p>	 Use Case
<p><i>Subject boundary:</i> Sebuah kotak yang mewakili sebuah sistem (proses bisnis sistem atau individual).</p>	 Subject
<p><i>Association relationship:</i> Aktor yang terlibat dalam <i>use case</i>, dan bagaimana hubungan antara <i>use case</i> dengan <i>use case</i> lainnya.</p>	
<p><i>Extend Relationship:</i> Merupakan perpanjangan dari <i>use case</i> untuk memasukkan perilaku opsional dan digambarkan dengan keterangan <<extend>>.</p>	
<p><i>Include Relationship:</i> Merupakan perwakilan penyertaan fungsionalitas satu <i>use case</i> dalam <i>use case</i> lainnya dan digambarkan dengan keterangan <<include>>.</p>	
<p><i>Generalization Relationship:</i> Mewakili kasus penggunaan khusus ke kasus yang lebih umum.</p>	

2.2 Penelitian Terdahulu

Tabel 2.3 Tabel Penelitian Terdahulu

No.	Nama Peneliti	Tahun	Judul Penelitian	Hasil Penelitian
1.	I Ketut, Widhi Adnyana, Gede Made, Yudi Antara, Dewa Ayu, Putri Wulandari	2021	Pemanfaatan <i>Application Programming Interface</i> Midtrans dan RajaOngkir Untuk Membangun <i>Enterprise Application Integration</i>	Artikel ini mengimplementasikan <i>Application Programming Interface</i> (API) Midtrans sebagai <i>payment gateway</i> , diharapkan pemilik usaha tidak perlu lagi melakukan pengecekan ongkos kirim dan pembayaran secara manual untuk menghitung ongkos kirim ke berbagai wilayah di Indonesia. Metode pengembangan aplikasi adalah metode <i>prototype</i> yaitu metode yang dikembangkan dengan cara berkomunikasi dengan pelanggan agar aplikasi yang dihasilkan sesuai dengan keinginan pelanggan. Hasil uji rekonsiliasi antara biaya pengiriman dengan total biaya yang dibayarkan pelanggan menunjukkan bahwa API Midtrans dapat mendukung pelanggan untuk melakukan pembayaran melalui <i>channel</i> yang bekerja sama dengan Midtrans.
2.	Rolando Alex Richo, I Putu Agus Swastika, Putu Trisna Hady Permana S, Eddy Muntina Dharma	2021	Rancang Bangun Aplikasi <i>Point Of Sales Payment Gateway</i>	Artikel ini menyajikan perancangan aplikasi <i>point of sale</i> untuk kasir yang mengintegrasikan fungsi <i>payment gateway</i> dengan aplikasi <i>point of sale</i> KasirIn. Merancang sistem berbasis aplikasi <i>mobile</i> menggunakan metode pengembangan <i>Scrum</i> . Pemodelan sistem menggunakan <i>Unified Modeling Language</i> (UML) dan bahasa pemrograman antarmuka pengguna menggunakan <i>framework</i> Dart dan Flutter. <i>Backend</i> menggunakan Python Flask yang bertindak sebagai API dan terhubung ke API <i>gateway</i> pembayaran Midtrans melalui <i>database</i> PostgreSQL. Model sistem yang dikembangkan dapat memfasilitasi transaksi penjualan dengan menggunakan metode pembayaran digital. Pengembangan menurut metode <i>Scrum</i> juga sangat efektif untuk siklus hidup pengembangan perangkat lunak karena manajemen kerja dihitung berdasarkan bobot poin tim pengembang.
3.	Erry Julio, Magdalena A Ineke Pakereng	2021	Implementasi API <i>Payment Gateway</i> Menggunakan Arsitektur <i>Microservice</i>	Untuk menyelesaikan integrasi pembayaran <i>online</i> , PT. Emporia Digital Raya perlu membuat aplikasi <i>payment gateway</i> yang dapat mengarahkan pembayaran ke bank tujuan. Arsitektur <i>microservice</i> dapat diterapkan untuk mengatasi permasalahan arsitektur monolitik, dimana pengembangan aplikasi berbentuk <i>web service</i> kecil yang saling berkomunikasi. Menggunakan konsep <i>routing</i> , <i>payment gateway</i> memiliki tugas seperti <i>router</i> , dimana tugas dari <i>payment gateway</i> adalah mencari jalan ke bank mana untuk mengirimkan informasi. <i>Gateway</i> pembayaran memproses permintaan dan membaca bidang kode pesan penerima untuk menemukan rute. Jika bank tujuan tidak ditemukan, <i>payment gateway</i> akan melakukan <i>query</i> dan mencari rute berdasarkan prioritas yang ditetapkan oleh PT. Kerajaan Digital Emporia.
4.	Ivan Reynaldi, Irfan Dwiguna Sumitra	2019	Pembangunan Aplikasi <i>Smart Apotek</i> Menggunakan API Midtrans Sebagai	Dengan menggunakan teknologi Midtrans API, Google Maps dan <i>Location Based Service</i> , ketiga teknologi tersebut dapat digunakan untuk mendapatkan lokasi dan

			<i>Payment Gateway</i> Pada Smartphone Android	menyimpan data pada ponsel Android. Setelah dilakukan pengujian dan penyebaran kuesioner, kami sampai pada kesimpulan bahwa perlunya sebuah aplikasi yang dapat mencari obat dan lokasi apotek terdekat, mengurangi antrian di apotek dan memudahkan pengguna untuk membayar.
5.	I Gede Santi Astawa	2018	Penerapan Metode LZW dalam Kompresi <i>File Chat</i> Multimedia pada Sistem <i>E-Marketplace</i> Sarana Upakara Bali	Peningkatan pelayanan penataan ruang upacara dapat dilakukan melalui <i>e-marketplace</i> yang menawarkan berbagai ruang upacara. <i>E-marketplace</i> Upakara berbasis <i>web</i> ini mencakup beberapa fitur tambahan seperti integrasi pembayaran dengan API Midtrans, obrolan metode kueri panjang untuk memfasilitasi interaksi antara penjual dan pembeli dan menggunakan algoritma LZW untuk mengompres <i>file</i> multimedia saat mengirim <i>file</i> selama obrolan. Tujuan dari proses kompresi ini adalah untuk menghemat ruang <i>disk</i> dan waktu transfer data. Pengujian yang dilakukan menghasilkan kesimpulan bahwa toko <i>online</i> ini optimal ketika kurang dari 200 pengguna mengunjungi Internet secara bersamaan, kualitas kompresi gambar dan video cukup baik.



BAB III

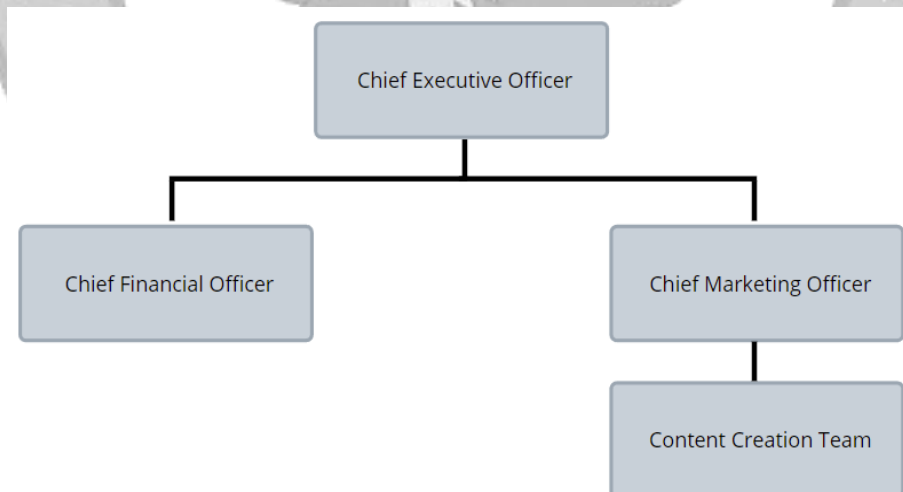
SISTEM SAAT INI

3.1 Profil Perusahaan

Levinsky Indonesia merupakan perusahaan UMKM yang didirikan oleh ketiga pengusaha yang ingin menjalankan bisnis di sektor pakaian yang dimulai sejak bulan Juli 2021. Perusahaan ini berfokus pada produksi pakaian, *marketing*, hingga penjualan dengan skala kecil. Dimulai dari produksi yang dilakukan di tempat yang sederhana dan dengan tim yang sederhana hingga produk seri pertamanya habis terjual dalam kurun waktu 3 bulan. Produksi pakaian ini kemudian berpindah dan mulai bekerja sama dengan salah satu vendor jahit dan sablon yang cukup terkenal. Levinsky Indonesia sangat mengutamakan kualitas dibanding kuantitas dengan menjualnya di harga yang cukup terjangkau, agar target *market* perusahaan ini dapat mencakup seluruh kalangan dengan kualitas yang premium.

3.2 Struktur Organisasi

Struktur organisasi pada perusahaan Levinsky Indonesia pada saat Tugas Akhir ini dibuat adalah sebagai berikut.



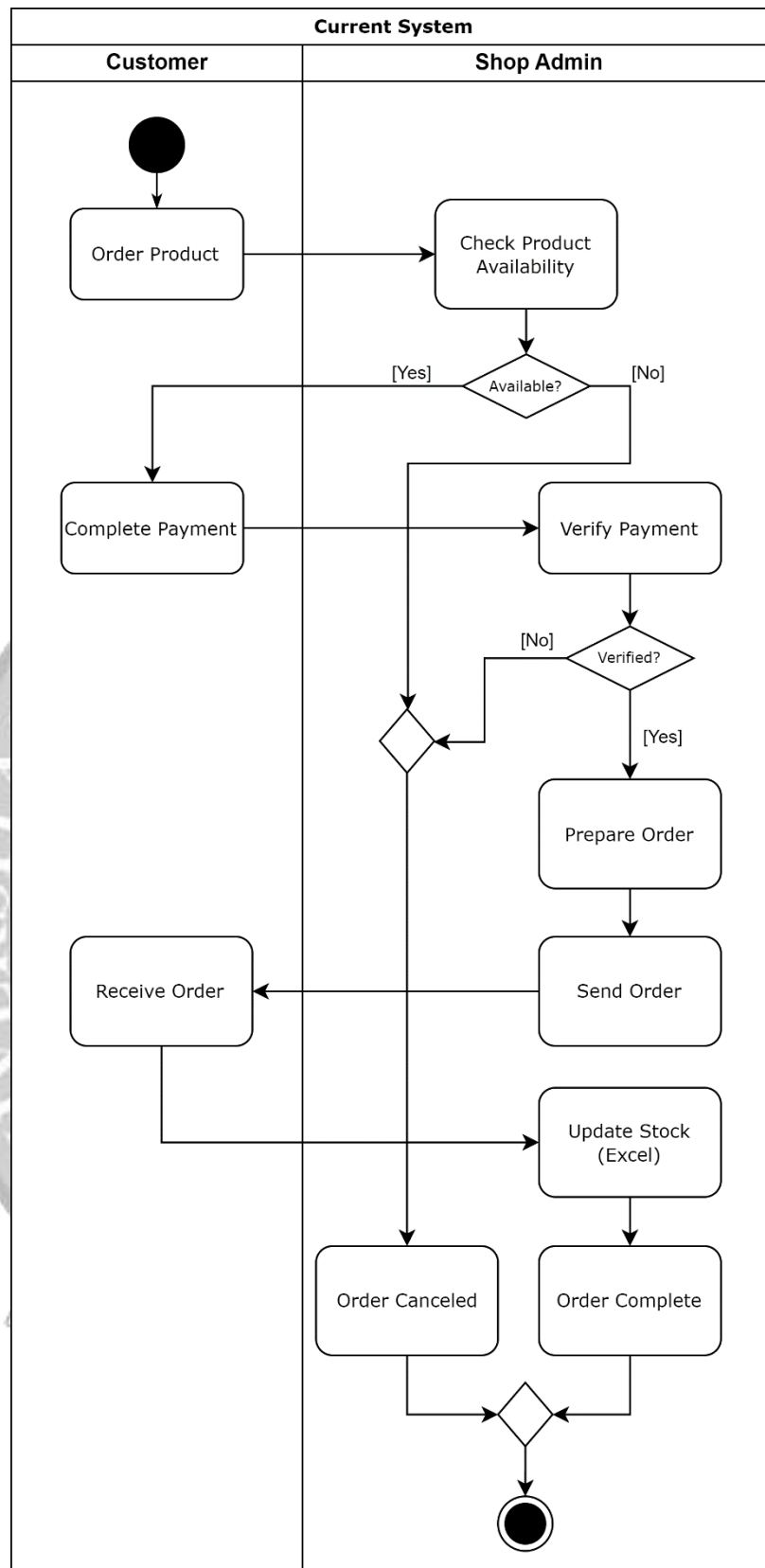
Gambar 3.1 Struktur Organisasi Levinsky Indonesia

- 1) *Chief Executive Officer* (CEO): mengambil keputusan serta mengawasi seluruh kegiatan yang berlangsung pada Levinsky Indonesia. CEO juga yang melaksanakan kegiatan negosiasi dan merancang strategi penjualan dengan sebaik mungkin.

- 2) *Chief Financial Officer* (CFO): mengelola dan menganalisis pergerakan keuangan pada perusahaan Levinsky Indonesia. Selain itu, CFO mencatat laporan jual beli dan mengatur anggaran pada perusahaan Levinsky Indonesia. CFO juga bertugas memeriksa ketersediaan atau stok produk.
- 3) *Chief Marketing Officer* (CMO): melakukan pemasaran seluasnya agar merek Levinsky Indonesia dikenali banyak orang dengan cepat. Sebagai CMO, ia harus membuat target pasaran serta menentukan strategi *marketing* dan produk yang menarik bagi pelanggan. Sehingga CMO juga mengatur kegiatan produksi yang sesuai dengan analisa *trend* pasar yang dilakukan oleh CMO.
- 4) *Content Creation Team*: membuat konten untuk disebar pada sosial media seperti poster, pemotretan model dengan produk baru, dan pembuatan ilustrasi yang digunakan pada berbagai platform digital.

3.3 Analisis Sistem Saat Ini

Proses pembayaran pada sistem saat adanya pesanan melalui *online shopping* pada perusahaan Levinsky Indonesia masih menggunakan metode transaksi secara manual dan memverifikasi pembayaran secara manual.



Gambar 3.2 Activity Diagram Sistem Saat Ini

3.4 Kendala Sistem Saat Ini

Saat ini, proses pemesanan yang terjadi pada Levinsky masih menggunakan cara pencatatan manual pada setiap pemesanan yang dilakukan, sehingga membuat efisiensi dan ketepatan dalam memproses transaksi. Berikut contoh pencatatan pesanan manual Levinsky.

No	Date	Purchase num	Product code	Size		Price	Nama Customer	Catatan
				Left	Right			
31	31/12/2021	031/LVSKY/XII/2021	B[CTC-001+I	M	M		Zufan Wijaya	done tokped
32	31/12/2021	032/LVSKY/XII/2021	B[CTC-001+I	M	M		San	done tokped
33	31/12/2021	033/LVSKY/XII/2021	B[CTC-001+I	XL	L		Valeri Monica	done tokped
34	01/01/2022	034/LVSKY/I/2022	B[CTC-001+I	L	L		Frederika Althea	done tokped
35	01/01/2022	035/LVSKY/I/2022	B[PU(W)-001	M	L		Sofiana	done tokped
36	02/01/2022	036/LVSKY/I/2022	B[PU(W)-001	M	M		Saudara VS	done trf
37	02/01/2022	037/LVSKY/I/2022	B[CTC-001+I	M	S		Temen VS	done trf
38	02/01/2022	038/LVSKY/I/2022	PU(B)-001	XXL			Juan	done trf
39	02/01/2022	039/LVSKY/I/2022	B[PU(W)-001	L	L		Glenn	done trf
40	02/01/2022	040/LVSKY/I/2022	B[PU(W)-001	L	L		VS	done trf
41	02/01/2022	041/LVSKY/I/2022	B[PU(W)-001	L	L		Kent	done
42	03/01/2022	042/LVSKY/I/2022	B[CTC-001+I	XL	S		Saudara Kent	done trf
43	03/01/2022	043/LVSKY/I/2022	B[PU(W)-001	L	L		Saudara Kent	done trf
44	03/01/2022	044/LVSKY/I/2022	B[CTC-001+I	L	M		Saudara Kent	done trf
45	03/01/2022	045/LVSKY/I/2022	B[CTC-001+I	L	XL		Saudara Kent	done trf
46	03/01/2022	046/LVSKY/I/2022	B[CTC-001+I	M	S		Saudara Kent	done trf
47	03/01/2022	047/LVSKY/I/2022	PU(B)-001	L			Saudara Kent	done trf
48	05/01/2022	048/LVSKY/I/2022	B[PU(W)-001	L	M		Asya.	done trf
49	08/01/2022	049/LVSKY/I/2022	B[PU(W)-001	L	L		Michael	done trf
50	09/01/2022	050/LVSKY/I/2022	B[PU(W)-001	XXL	XXL		Calvin	done tokped
51	11/01/2022	051/LVSKY/I/2022	PU(B)-001	L			Javier	done trf
52	13/01/2022	052/LVSKY/I/2022	B[CTC-001+I	L	L		Hans	done trf
53	14/01/2022	053/LVSKY/I/2022	B[CTC-001+I	L	L		Eric	done trf

Gambar 3.3 Pencatatan Pesanan Levinsky dengan Excel (Secara Manual)

Setelah mencatat penjualan yang terjadi baik dari penjualan *online shopping* via WhatsApp maupun melalui *marketplace*, proses verifikasi transaksi dilakukan secara manual. Hal ini dapat mengurangi efisiensi pekerjaan karena harus melakukan verifikasi transaksi per satu.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM USULAN

4.1 Tahap Perencanaan Sistem

Untuk pengembangan sebuah sistem, diperlukan tahap perencanaan sistem agar proses pembangunan sistem berjalan sesuai dengan rencana yang diinginkan. Tahap perencanaan tersebut terdiri dari tahap kelayakan teknis, kelayakan ekonomi dan kelayakan organisasi.

4.1.1 Kelayakan Teknis (*Technical Feasibility*)

Jika ditimbang dari sisi kelayakan teknis, terdapat beberapa risiko yang memungkinkan terjadi:

- 1) Risiko pengenalan sistem di level rendah: pengguna dari sistem ini nantinya adalah masyarakat umum yang biasa berbelanja *online*. Karena sistem ini di desain layaknya aplikasi belanja *online* lainnya, risiko untuk pengenalan sistem ini termasuk golongan rendah.
- 2) Risiko pembangunan aplikasi di level sedang: sistem usulan memiliki berbagai kebutuhan yang harus terpenuhi, seperti memproses pembayaran dengan benar karena merupakan bagian krusial dari transaksi belanja *online*. Maka dari itu, risiko pembangunan aplikasi adalah sedang karena adanya keterbatasan waktu agar dapat memenuhi kebutuhan-kebutuhan tersebut.

4.1.2 Kelayakan Ekonomi (*Economy Feasibility*)

Berikut adalah daftar perangkat keras dan perangkat lunak minimal dari sisi *server* dan *client* yang dapat digunakan dalam penerapan sistem usulan:

Tabel 4.1 Daftar Perangkat Keras Dan Perangkat Lunak Minimum Untuk Sistem Usulan Dari Sisi *Server*

Perangkat Keras	
<i>Processor</i>	2 GHz <i>or faster</i> [10]
RAM	2 GB [10]
<i>Hard Disk</i>	64 GB [10]

Perangkat Lunak	
Sistem Operasi	Windows 2008/Vista [11]

<i>Database</i>	MariaDB 10.4.25 [12]
PHP	7.4 [13]
<i>Web Server</i>	Apache/2.4.54 (Win64) [12]

Tabel 4.2 Daftar Perangkat Keras Dan Perangkat Lunak Minimum Untuk Sistem Usulan Dari Sisi *Client*

Perangkat Keras	
<i>Processor</i>	2 GHz <i>or faster</i> [10]
RAM	2 GB [10]
<i>Hard Disk</i>	32 GB [10]
Perangkat Lunak	
Sistem Operasi	Windows 2008/Vista [11]
<i>Database</i>	MariaDB 10.4.25 [12]
PHP	7.4 [13]

4.1.3 Kelayakan Organisasi (*Organization Feasibility*)

Jika dilihat dari sisi kelayakan organisasi, sistem usulan ini akan memiliki tingkat risiko sedang karena pengguna dari sistem ini merupakan generasi muda yang sudah terbiasa dengan teknologi serta sudah terbiasa memproses pesanan. Namun untuk memaksimalkan pengetahuan dan potensi mengenai sistem, lebih baik diadakan sosialisasi atau pelatihan mengenai penggunaan sistem usulan. Dengan itu diharapkan pengguna dapat menggunakan serta memanfaatkan sistem usulan dengan baik.

4.2 Tahap Analisis

Dalam tahap analisis membahas mengenai kebutuhan pengguna, pemodelan fungsional serta pemodelan struktural.

4.2.1 Pengumpulan Kebutuhan Pengguna (*User Requirements Gathering*)

Pada bagian ini membahas kebutuhan pengguna yang terdiri dari *functional* dan *non-functional requirements*.

4.2.1.1 *Functional Requirements*

Berikut merupakan kebutuhan fungsional (*Functional Requirements*) yang terdapat pada sistem usulan sebagai berikut:

- 1) *User*

User merupakan seluruh pengguna umum yang belum memiliki akun dan melakukan *login* pada sistem usulan.

a. *Login*

Hak akses pada sistem ini berupa *User*, *Member*, dan *Admin*. *Member* dapat mengakses seluruh fitur *member* seperti melakukan transaksi, menerima notifikasi dan lainnya. Selain itu juga terdapat *admin* yang dapat mengelola produk, mengirimkan notifikasi, memproses transaksi dan lainnya. Untuk akses menjadi *member* dan *admin* diperlukan *login* terlebih dahulu dengan mengisi alamat *email* beserta *password*.

b. *Register*

Agar *user* dapat melakukan *login*, *user* harus mendaftarkan akunnya terlebih dahulu. Pada bagian *register* ini diperlukan untuk mengisi formulir pendaftaran berupa *username*, alamat *email*, nomor *handphone*, dan *password*.

c. *Forget Password*

Jika *member* lupa dengan *password* akun yang telah dibuat, *member* dapat mengganti *password* dengan tahapan verifikasi melalui *email* agar *user* dapat melakukan penggantian *password*.

2) *Member*

Member merupakan pengguna umum yang telah memiliki akun dan sudah melakukan *login* pada sistem usulan.

a. Memasukkan produk ke *cart*

Setelah berhasil mendaftar, *member* dapat memanfaatkan fitur *cart* untuk menyimpan pesanan yang ingin dibeli. Sehingga pengguna yang belum sempat melakukan transaksi tidak perlu mengingat pesanan yang ingin dibeli.

b. Melakukan proses *checkout*

Jika *member* sudah selesai berbelanja dengan memasukan produk ke *cart*, penghitungan biaya pengiriman akan dilakukan oleh API RajaOngkir (pihak ketiga) dan *member* dapat melakukan pembayaran dengan melewati API Midtrans (pihak ketiga).

c. Menerima notifikasi

Member dapat menerima notifikasi yang dikirimkan oleh sistem ataupun oleh *admin*, sehingga *member* akan terus mendapatkan informasi terbaru.

d. Mengelola *address*

Member dapat mengelola alamat yang disimpan pada sistem dan memilihnya sebagai alamat utama. Selain itu pada aktivitas ini juga dapat membuat alamat baru dan menghapus alamat

3) *Admin*

Admin merupakan pengguna khusus untuk internal perusahaan dan memiliki akses pada *Content Management System* (CMS) sistem usulan.

a. Mengelola produk

Admin dapat mengelola produk dengan mengubah informasi produk, menghapus produk, serta mengaktifkan dan menonaktifkan produk.

b. Mengelola pesanan

Admin dapat mengelola pesanan yang diterima seperti menerima pesanan, menolak pesanan, dan melihat detail pesanan.

c. Mengirimkan notifikasi

Admin dapat mengirimkan notifikasi kepada *member*, baik kepada seluruh *member* maupun kepada *member* baru saja.

4.2.1.2 Non-Functional Requirements

Berikut merupakan kebutuhan non-fungsional (*Non-Functional Requirements*) yang terdapat pada sistem usulan sebagai berikut:

1) *Operational*

Sistem usulan akan membutuhkan koneksi internet agar dapat beroperasi sebab sistem usulan menggunakan API pihak ketiga dan dapat dijalankan melalui *web browser*.

2) *Performance*

a. Sistem akan terhubung secara *real time* dengan *database web server*.

b. Sistem akan dapat beroperasi secara *online*

3) *Security*

a. Seluruh informasi terkait pengguna (data pribadi, detail transaksi, keranjang belanja dan notifikasi) hanya dapat diakses oleh pengguna ketika pengguna sudah melakukan *login*

- b. Sistem telah menerapkan sistem *hashing password* (kata sandi) satu arah atau MD5

4) *Cultural*

Sistem usulan ini menggunakan bahasa Inggris secara keseluruhan sehingga sistem usulan ini berunsur konsisten dan dapat dimengerti secara internasional sesuai dengan keinginan dari *user*.

4.2.2 Pemodelan Fungsional

Pada tahap pemodelan fungsional, proses kegiatan yang berlangsung pada sistem usulan dari masing-masing pengguna akan dijelaskan melalui *use case diagram* dan *activity diagram*.

4.2.2.1 Use Case Diagram

Use case diagram merupakan pengilustrasian atas fungsi dan aktor dalam sistem usulan. Aktor-aktor dalam sistem usulan ini sebagai berikut:

1) *Admin*

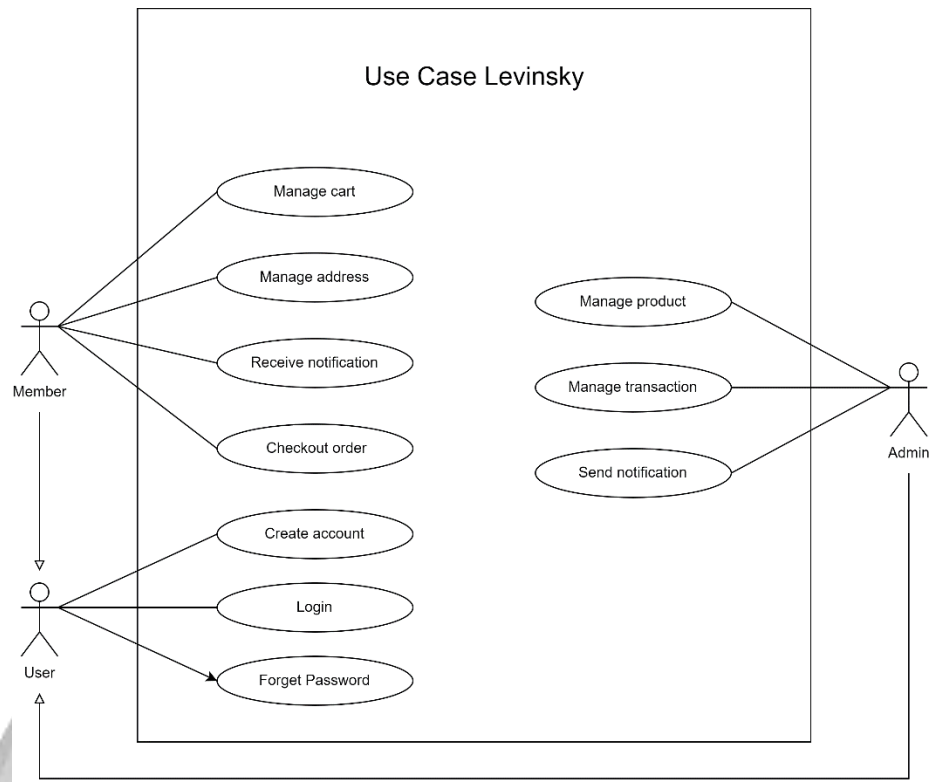
Admin merupakan salah satu jenis pengguna yang memiliki fungsi untuk mengubah informasi apapun terkait *website*. Tugas admin adalah untuk mengelola produk, mengelola transaksi dan mengirimkan notifikasi. Selain itu *admin* juga dapat melakukan verifikasi-verifikasi lainnya dalam sistem. Pada sistem usulan ini, pemilik usaha akan menerima akses sebagai *admin*.

2) *User*

User merupakan seluruh masyarakat umum yang dapat mengakses sistem usulan. Seluruh *user* dapat membuat akun dalam sistem untuk menjadi *member*, *login* menjadi *member*, melihat halaman utama dan melihat halaman produk.

3) *Member*

Member merupakan masyarakat umum atau calon pembeli yang telah melakukan registrasi pembuatan akun dan telah melakukan *login*. Fitur yang dapat diakses oleh pengguna *member* adalah memasukkan produk dalam keranjang belanja, mengelola alamat pengiriman pesanan, menerima notifikasi, dan melakukan *checkout* pesanan.



Gambar 4.1 *Use Case Diagram* Levinsky

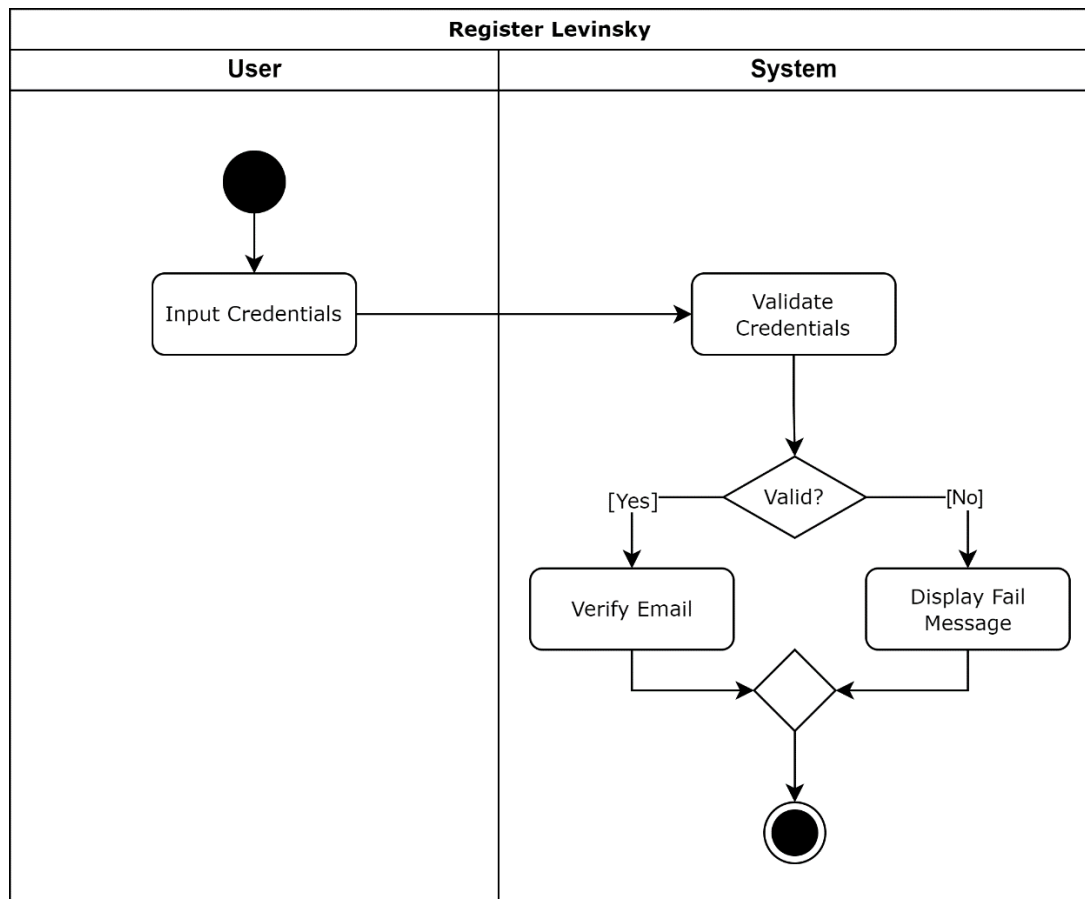
4.2.2.2 *Activity Diagram*

Berikut merupakan *activity diagram* pada sistem usulan Levinsky Indonesia:

1) *Register*

Pada aktivitas berikut, *user* dapat melakukan registrasi pengguna baru untuk *login* menjadi *member*. Adapun alur aktivitas Gambar 4.2 sebagai berikut:

- *User* memasukkan data *form* registrasi berupa *username*, *email*, dan *password*.
- Sistem akan memvalidasi kecocokan *email* pada *database* jika *email* tersebut telah terdaftar.
- Jika berhasil memvalidasi, sistem akan registrasi data sebagai pengguna baru.
- Jika gagal memvalidasi, sistem akan membatalkan registrasi dan menampilkan pesan gagal.

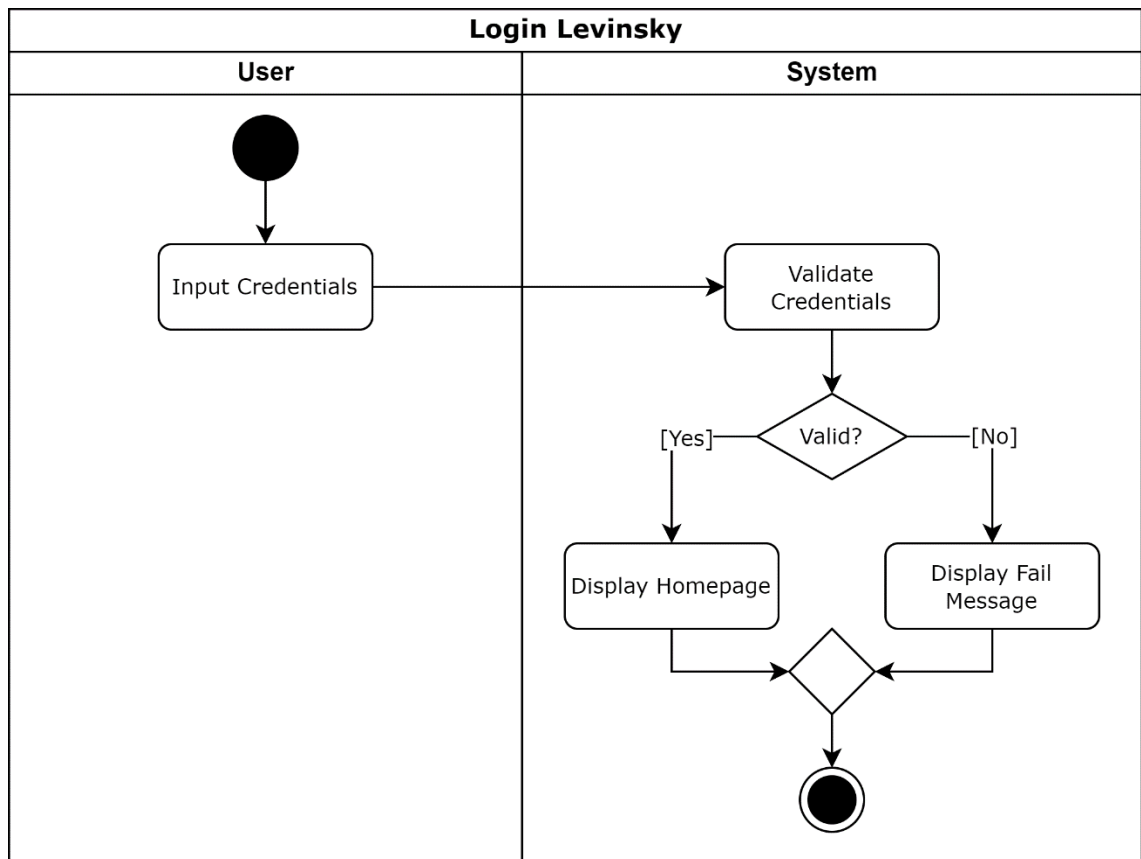


Gambar 4.2 Activity Diagram Register Levinsky

2) Login

Pada aktivitas berikut, *user* dapat melakukan *login* untuk mengubah peran menjadi *member*. Adapun alur aktivitas Gambar 4.3 sebagai berikut:

- *User* memasukkan data formulir *login* berupa *email*, dan *password*.
- Sistem akan memvalidasi kecocokan *email* dan *password* pada *database*.
- Jika berhasil memvalidasi, *user* akan berhasil *login* sebagai *member*.
- Jika gagal memvalidasi, *user* akan gagal *login* dan sistem akan menampilkan pesan kegagalan *login*.

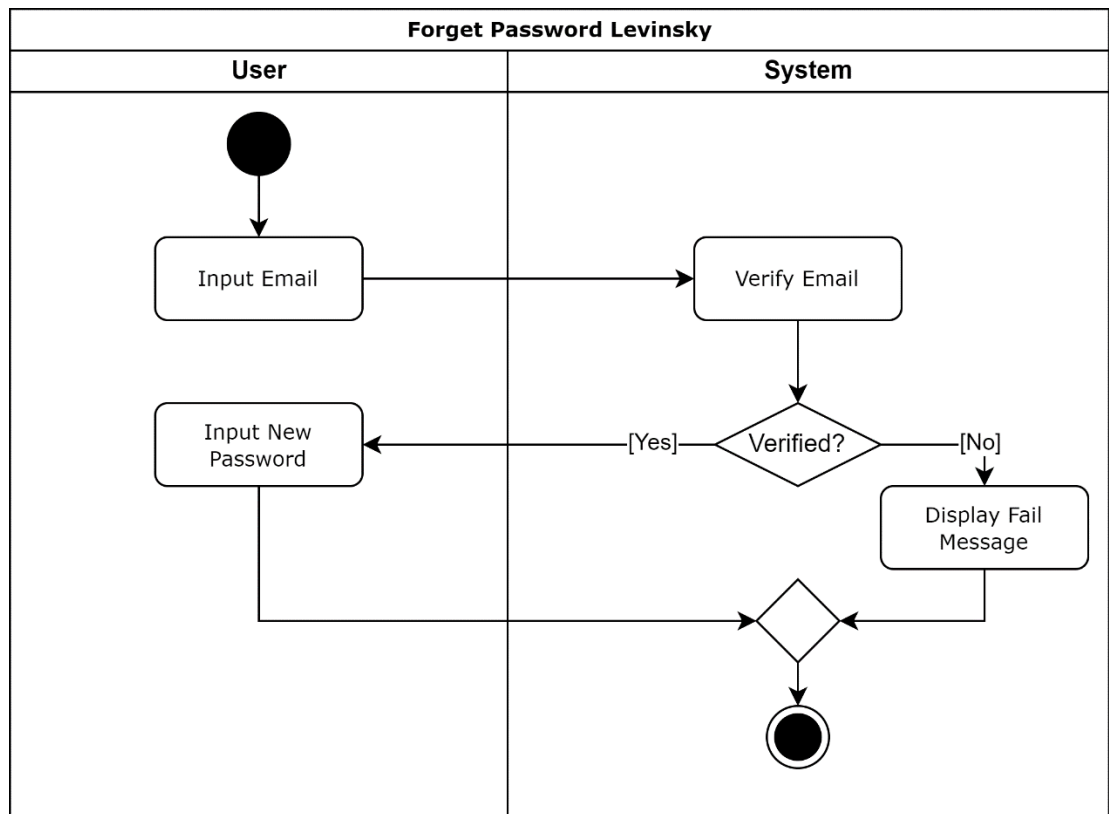


Gambar 4.3 Activity Diagram Login Levinsky

3) *Forget Password*

Pada aktivitas Gambar 4.4, *user* yang lupa *password* akun dapat mengubah *password* nya dengan langkah berikut:

- *User* dapat mengakses halaman *forget password*.
- *User* dapat memasukkan *email* akun pada formulir *forget password*.
- Sistem akan memverifikasi *email* pada *database*.
- Jika sistem berhasil memverifikasi, sistem akan mengirimkan *link* perubahan *password* melalui *email*.
- Jika sistem gagal memverifikasi, sistem akan menampilkan pesan gagal.

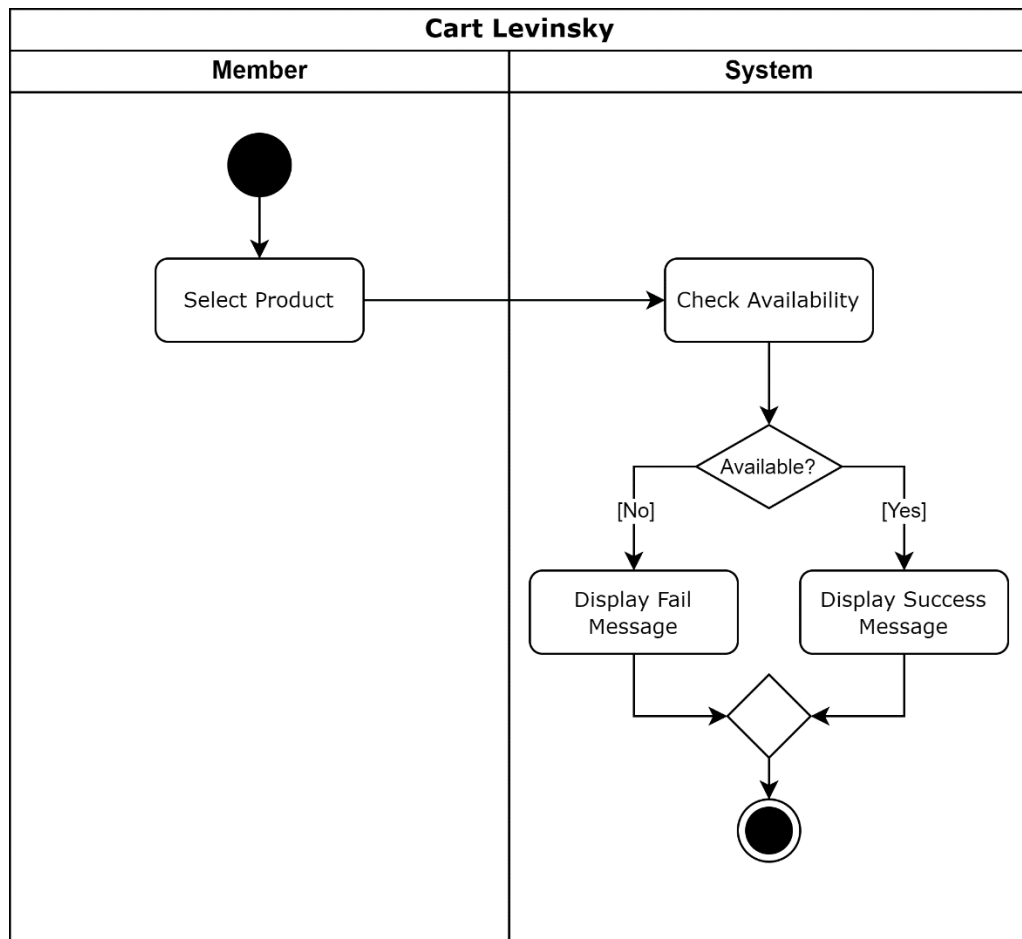


Gambar 4.4 Activity Diagram Forget Password Levinsky

4) *Cart*

Pada aktivitas berikut, *member* dapat memasukkan produk ke dalam *cart*. Langkah-langkahnya aktivitas Gambar 4.5 sebagai berikut:

- *Member* dapat memilih varian produk yang ingin di beli.
- *Member* memilih ukuran dan memasukkan jumlah varian produk.
- Sistem akan memeriksa ketersediaan produk yang diinginkan.
- Jika tidak tersedia, maka produk tidak ditambahkan ke *cart*.
- Jika tersedia, maka produk akan ditambahkan ke *cart*.
- Sistem akan melakukan pengecekan ketersediaan produk setiap *member* membuka halaman *cart*.

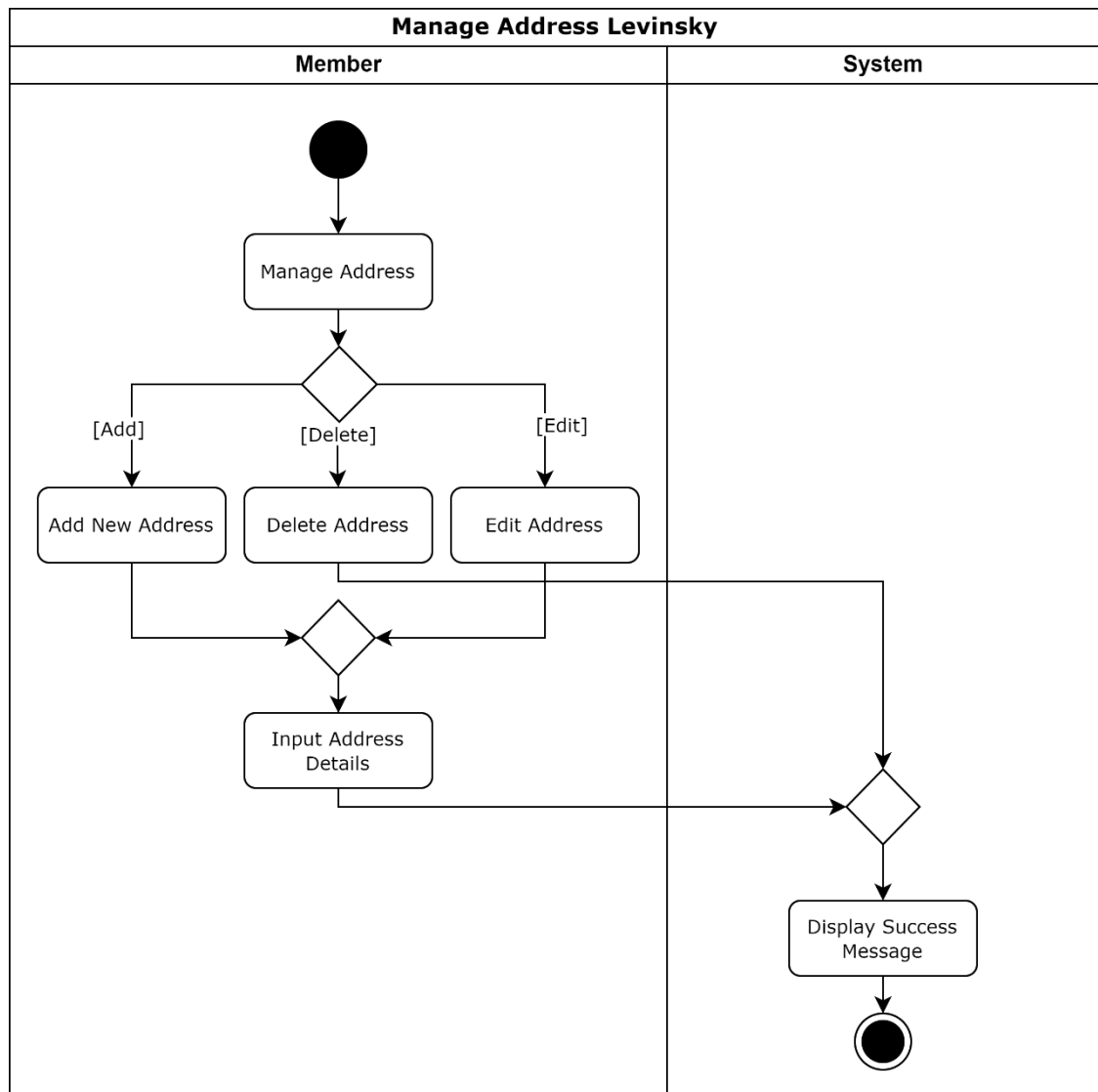


Gambar 4.5 Activity Diagram Cart Levinsky

5) Manage Address

Member dapat mengelola alamat yang disimpan seperti pada Gambar 4.6. Langkah-langkah yang dapat dilaksanakan dalam pengelolaan alamat sebagai berikut:

- *Member* dapat memilih untuk menambahkan alamat baru, menghapus alamat, atau menyunting alamat.
- Jika *member* memilih untuk menambahkan alamat baru, *member* dapat mengisi alamat, memilih provinsi dan kabupaten yang ditampilkan dari API RajaOngkir (pihak ketiga)
- Jika *member* memilih untuk menghapus alamat, sistem akan menghapus alamat dari *database*.
- Jika *member* memilih untuk menyunting alamat, maka *member* dapat mengubah informasi alamat dan sistem akan menyimpan perubahan pada *database*.

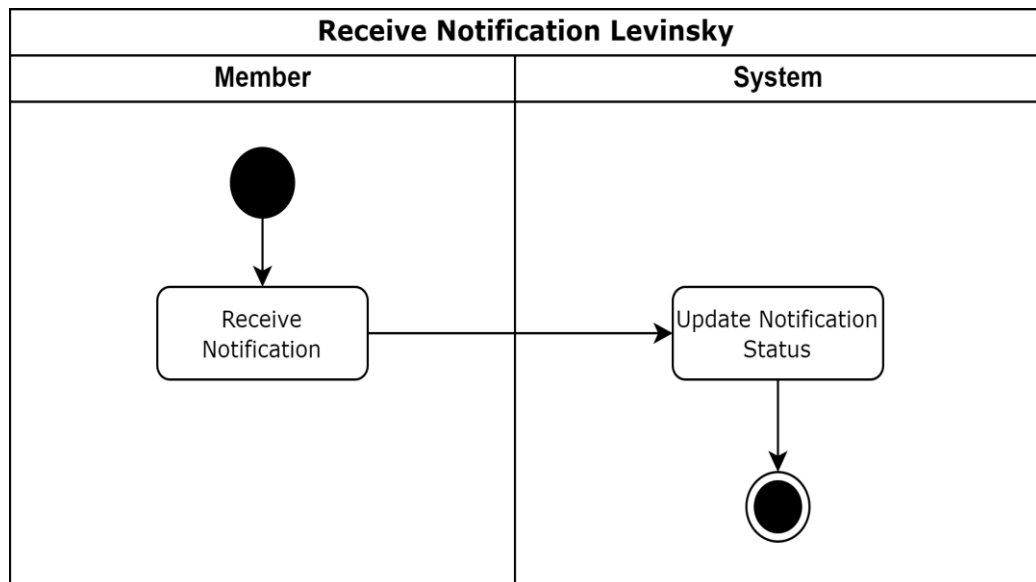


Gambar 4.6 Activity Diagram Manage Address Levinsky

6) Receive Notification

Member dapat menerima notifikasi yang dikirimkan oleh sistem maupun admin, adapun langkah-langkah aktivitas Gambar 4.7 sebagai berikut:

- Sistem atau *admin* mengirimkan notifikasi kepada pengguna.
- *Member* akan menerima notifikasi.
- *Member* dapat melihat notifikasi "*user_notified*" yang berstatus 1 (belum dibaca) dan menekan notifikasi untuk langsung mengalihkan ke halaman yang dituju.
- Sistem akan mengubah status notifikasi "*user_notified*" menjadi 0 yang tandanya sudah dibaca.

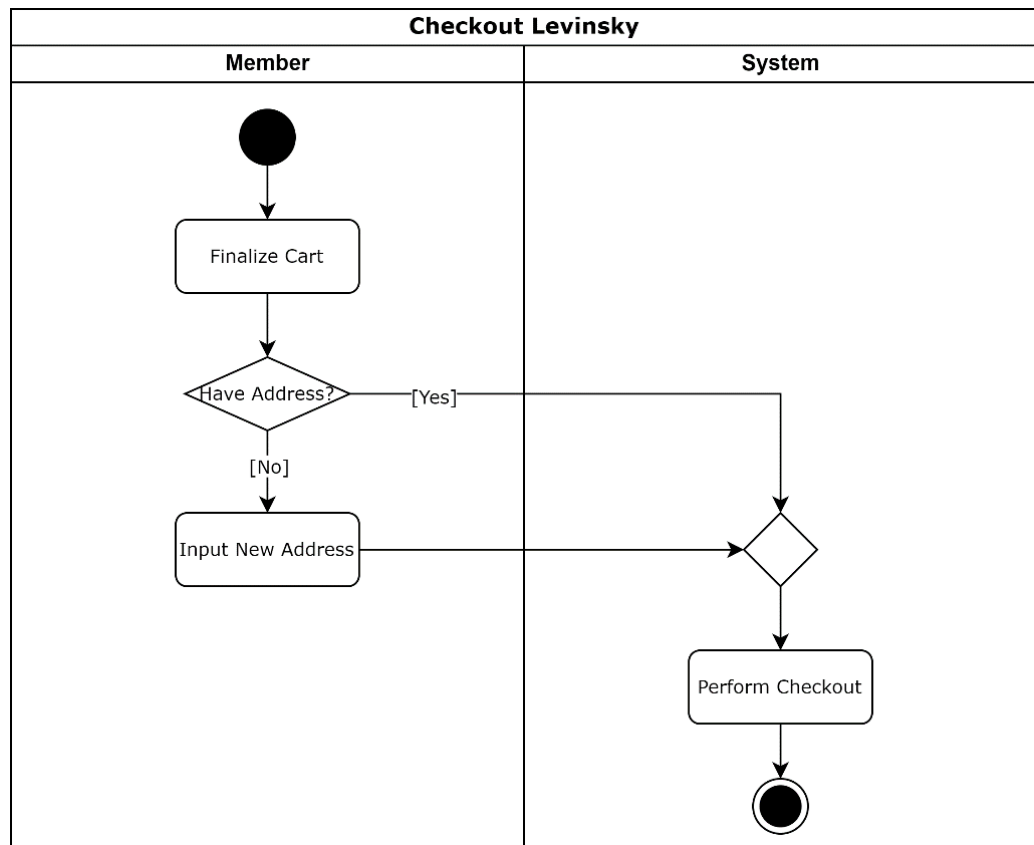


Gambar 4.7 Activity Diagram Receive Notification Levinsky

7) Checkout

Member yang telah menyelesaikan *cart* dan sudah selesai berbelanja dapat melalui tahapan *checkout* seperti pada Gambar 4.8 sebagai berikut:

- *Member* telah selesai memasukkan produk ke *cart*.
- *Member* dapat melakukan *checkout*.
- Jika *member* belum memiliki alamat, *member* harus membuat alamat baru yang dapat didaftarkan dengan mengisi formulir alamat berupa alamat, provinsi, dan kabupaten. Setelah itu alamat secara *default* akan terpilih sebagai alamat utama.
- Jika *member* sudah memiliki alamat, maka sistem dapat melakukan *checkout*.
- Pada tahapan *checkout*, sistem akan menghitung ongkos kirim berdasarkan kabupaten/kota asal yang telah ditentukan sejak awal pada sistem menuju kabupaten/kota pengiriman melalui API RajaOngkir (pihak ketiga). Selanjutnya sistem akan mengkalkulasi total dari biaya yang harus ditanggung oleh pembeli dan kemudian mengirim data berupa total biaya yang harus dibayar, alamat *email member*, dan nomor telepon *member* kepada API Midtrans untuk menyelesaikan pembayaran.

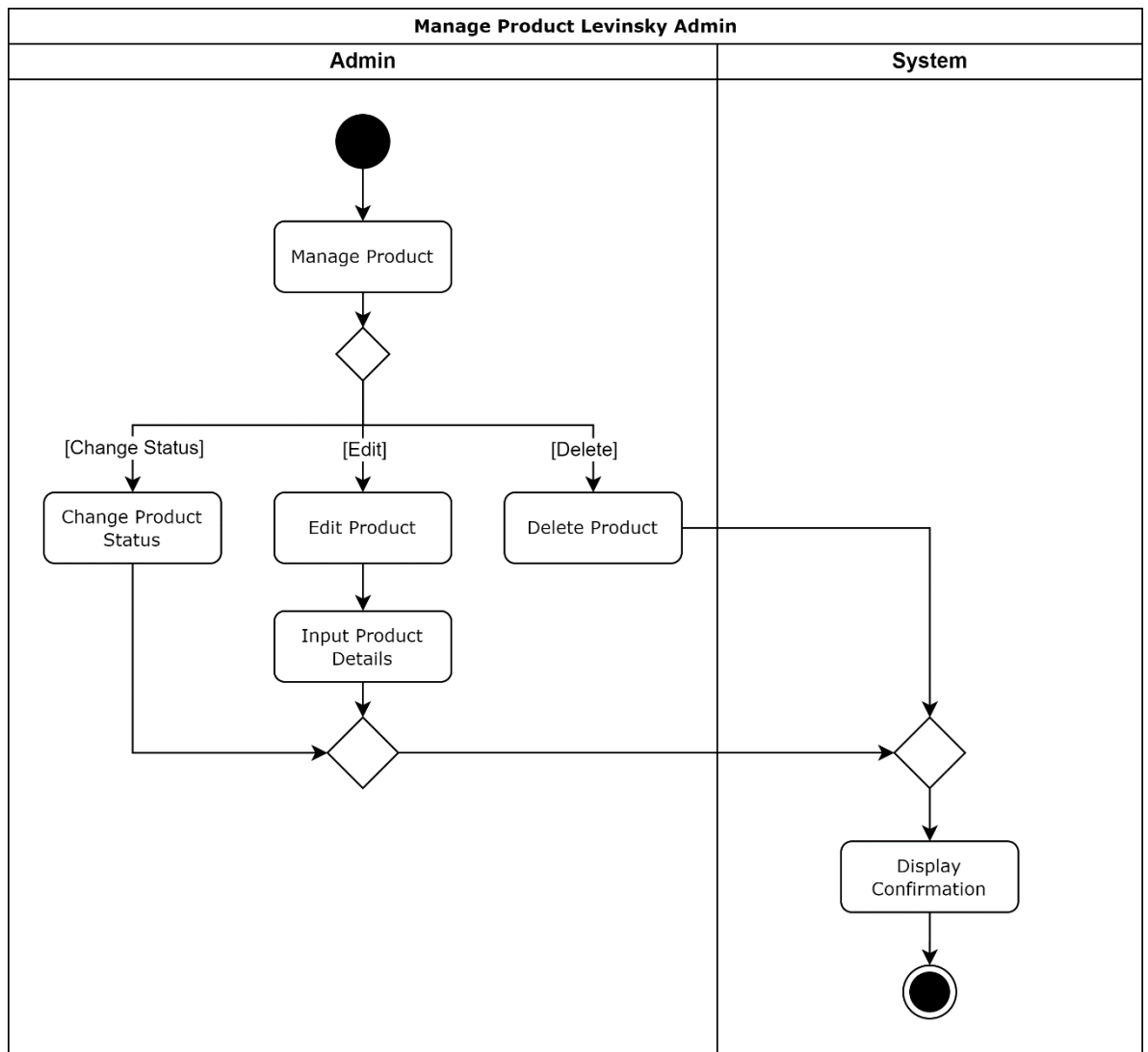


Gambar 4.8 Activity Diagram Checkout Levinsky

8) *Manage Product Admin*

Pada aktivitas Gambar 4.9, *admin* dapat mengelola produk yang ditampilkan pada halaman produk *user*. Langkah aktivitas pengelolaan produk sebagai berikut:

- *Admin* dapat melakukan pengelolaan produk dengan mengganti status produk, mengedit informasi produk, dan menghapus produk.
- Jika *admin* memilih untuk mengubah status produk, sistem akan memperbaharui status produk pada *database*.
- Jika *admin* memilih untuk mengedit informasi produk, setelah admin mengubah informasi produk maka sistem akan memperbaharui detail produk pada *database*.
- Jika *admin* memilih untuk menghapus produk, sistem akan menghapus produk pada *database*.



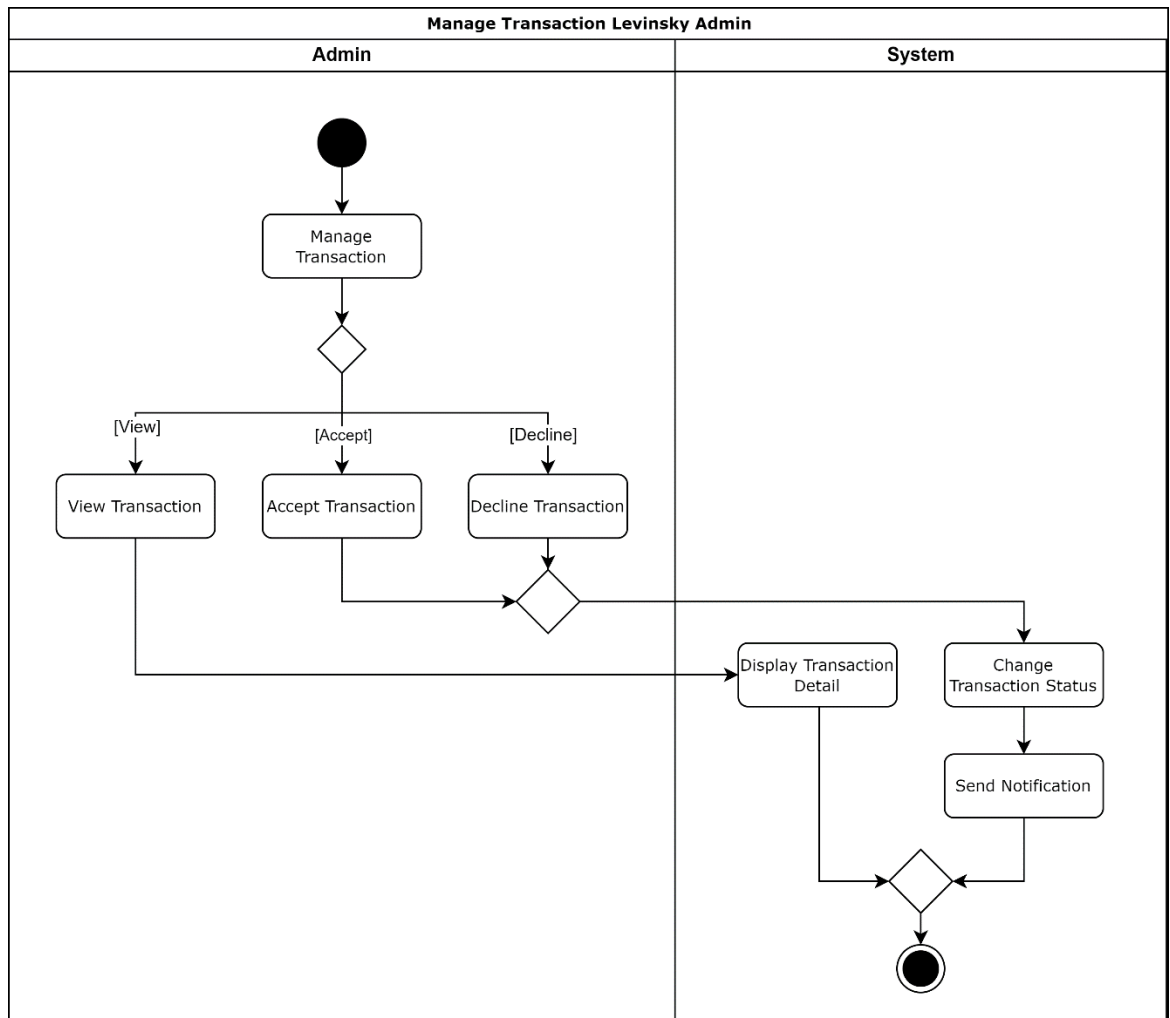
Gambar 4.9 Activity Diagram Manage Product Admin Levinsky

9) Manage Transaction Admin

Pada Gambar 4.10, *admin* dapat mengelola transaksi yang diterima pada sistem. Adapun langkah-langkahnya sebagai berikut:

- *Admin* dapat memilih untuk melihat detail transaksi, menerima transaksi atau menolak transaksi.
- Jika *admin* memilih untuk melihat detail transaksi, maka sistem akan menampilkan detail transaksi beserta detail pesanan yang dilakukan oleh pengguna *member*.
- Jika *admin* memilih untuk menerima transaksi, maka sistem akan memperbarui status transaksi menjadi “*CONFIRMED*”.
- Jika *admin* memilih untuk menolak transaksi, maka sistem akan memperbarui status transaksi menjadi “*CANCELED*”.

- Transaksi yang baru diterima oleh admin secara *default* akan berstatus “*ON PROGRESS*”

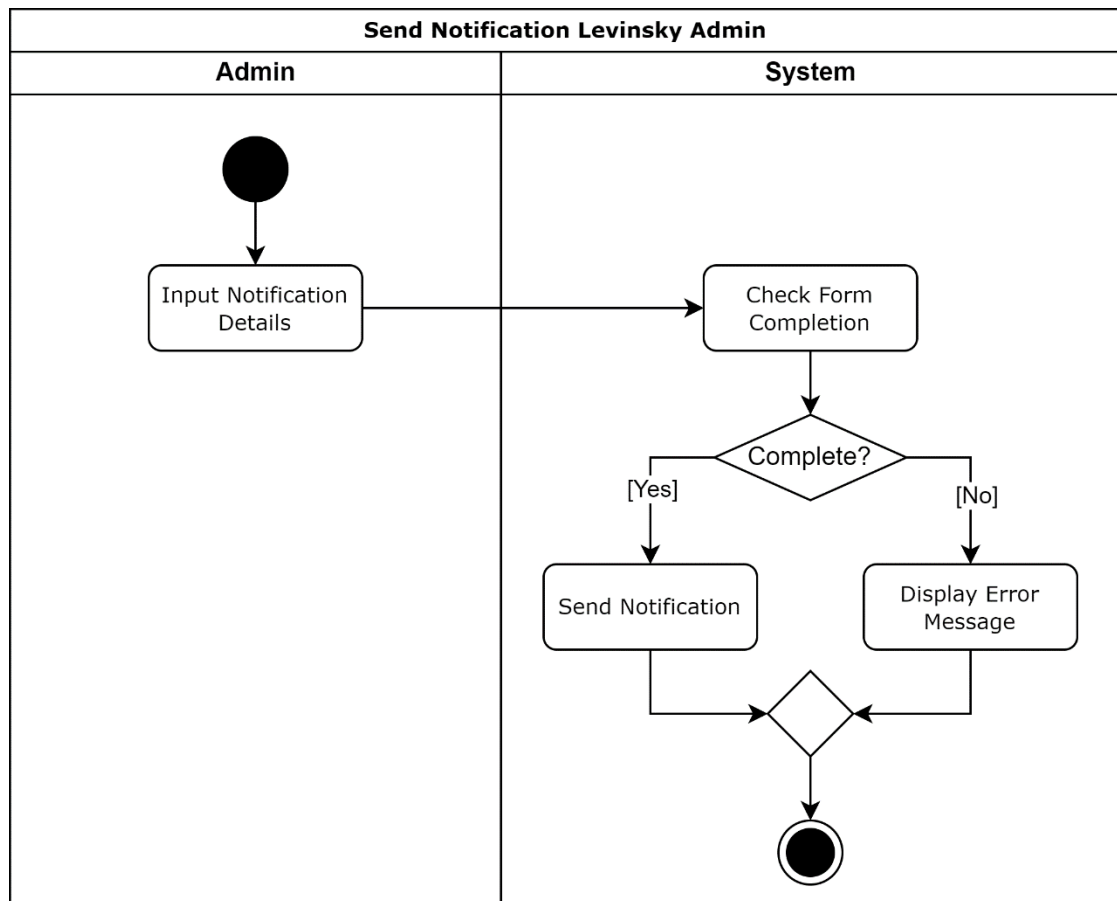


Gambar 4.10 Activity Diagram Manage Transaction Admin Levinsky

10) Send Notification Admin

Pada Gambar 4.11, *admin* dapat mengirimkan notifikasi kepada pengguna *member*, adapun tahapannya sebagai berikut:

- *Admin* memilih fungsi untuk mengirim notifikasi.
- *Admin* mengisi formulir berupa *title*, *message*, *link* (mengalihkan), dan pilihan pengguna yang menerima notifikasi.
- Sistem akan memvalidasi kelengkapan formulir notifikasi.
- Jika formulir tidak lengkap, maka sistem akan memunculkan *error message* pada kolom yang belum terisi.
- Jika formulir sudah lengkap, maka sistem akan menyimpan data notifikasi dan mengirimkan notifikasi kepada pengguna *member* yang dituju.



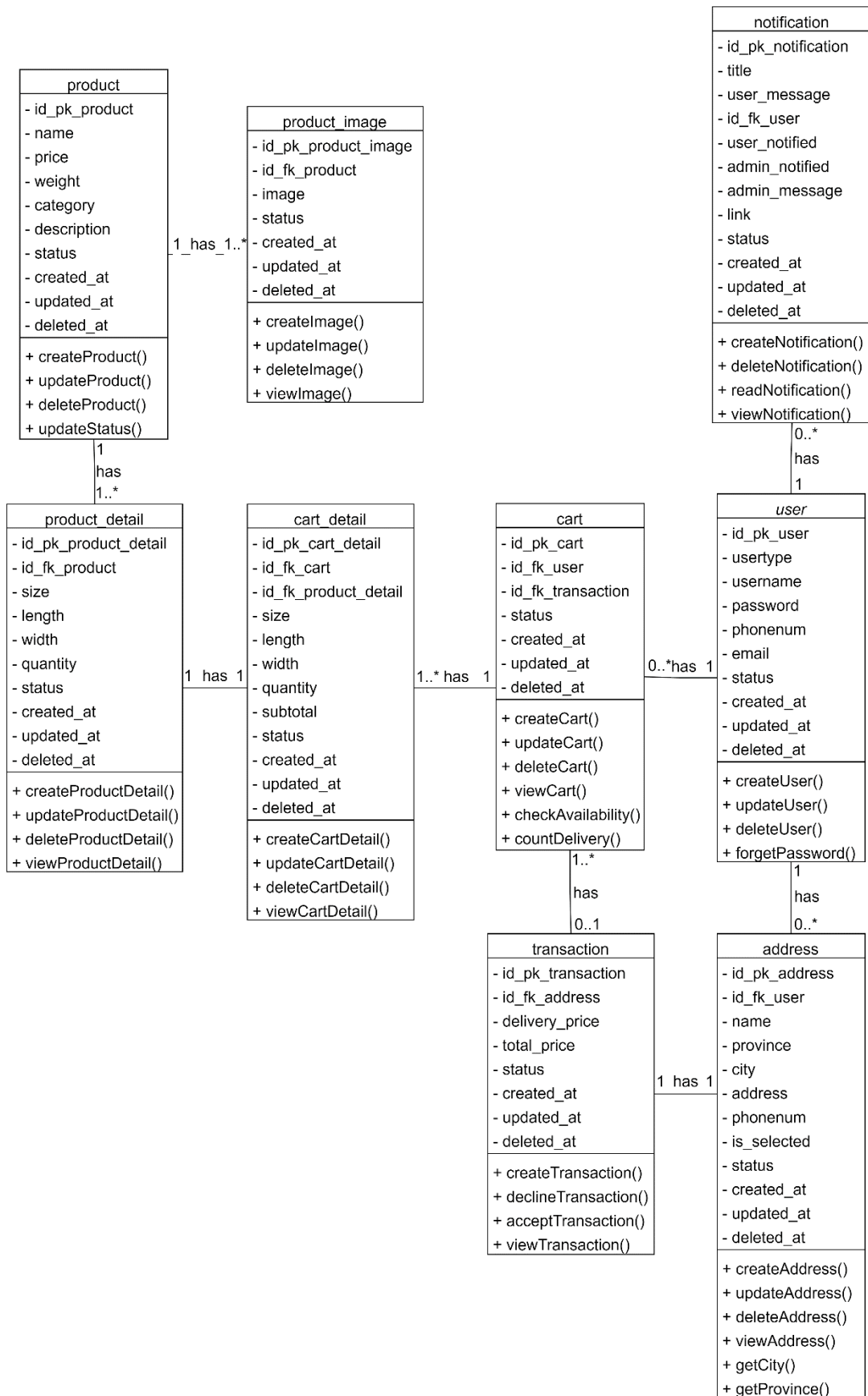
Gambar 4.11 Activity Diagram Send Notification Admin Levinsky

4.2.3 Pemodelan Struktural

Pada tahap pemodelan struktural, struktur sistem usulan dijelaskan melalui *class diagram* dan *table relational diagram*.

4.2.3.1 Class Diagram

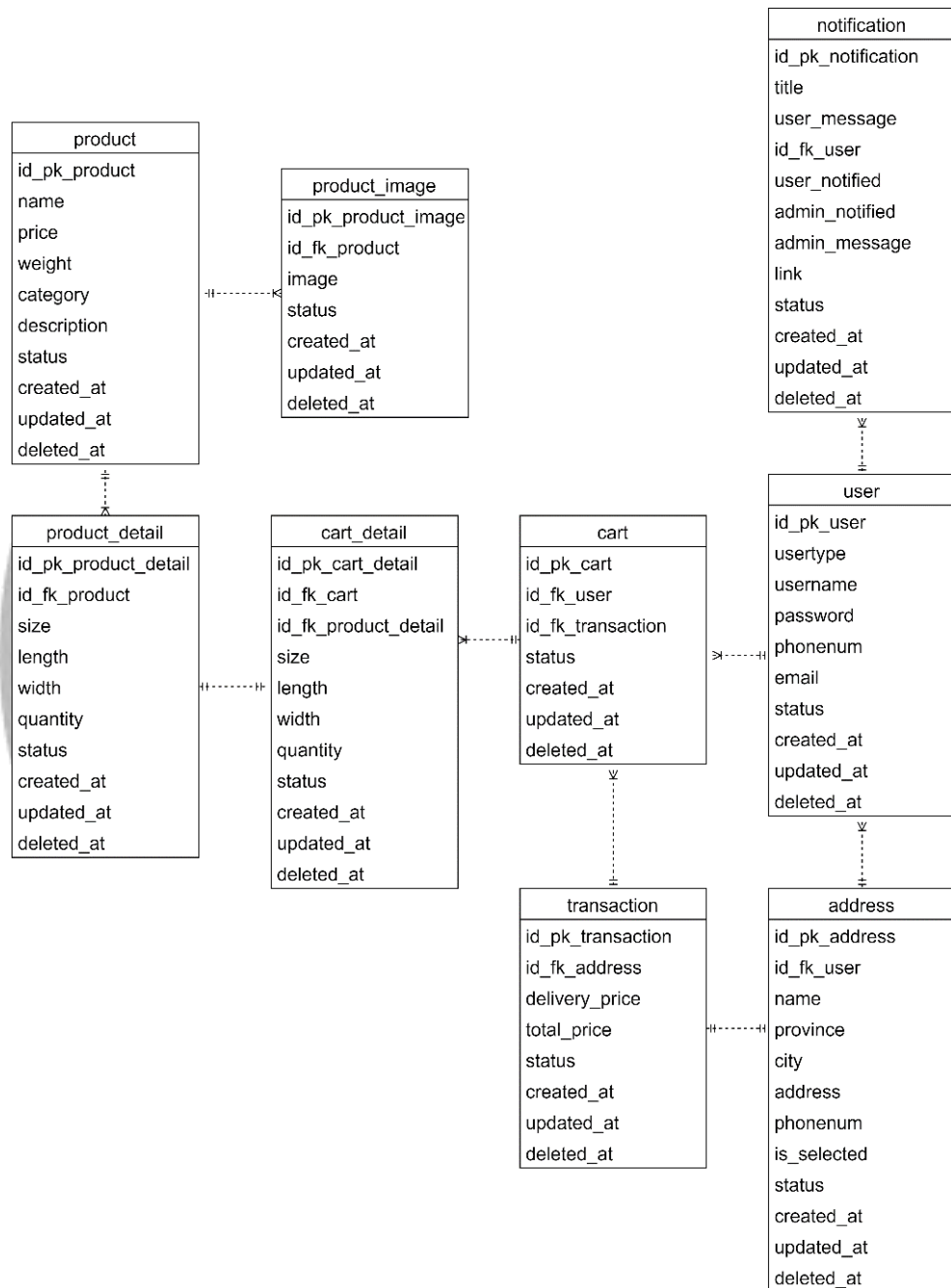
Pada Gambar 4.12 ditunjukkan *class diagram* untuk sistem Levinsky yang menggambarkan kolom dan *class* setiap tabel pada *database* yang digunakan dalam sistem Levinsky.



Gambar 4.12 Class Diagram Levinsky

4.2.3.2 Table Relational Diagram

Pada Gambar 4.13 ditunjukkan *table relational diagram* untuk sistem Levinsky yang menggambarkan relasi antar tabel pada *database* yang digunakan dalam sistem Levinsky.



Gambar 4.13 *Table Relational Diagram* Levinsky

4.2.3.3 Table Description

Berikut merupakan penjelasan detail mengenai jenis data per baris pada tabel yang ada di *database* Levinsky:

1) Tabel *user*

Tabel ini membahas mengenai tabel *user* pada *database* Levinsky yang memiliki *primary key* “*id_pk_user*”.

Tabel 4.3 Tabel *user* Levinsky

Field name	Type	Length	Key
<i>id_pk_user</i>	<i>int</i>		<i>Primary key</i>
<i>usertype</i>	<i>varchar</i>	10	
<i>username</i>	<i>varchar</i>	50	
<i>password</i>	<i>varchar</i>	50	
<i>phonenum</i>	<i>varchar</i>	25	
<i>email</i>	<i>varchar</i>	50	
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

2) Tabel *address*

Tabel ini membahas mengenai tabel *address* pada *database* Levinsky yang memiliki *primary key* “*id_pk_address*” dan hubungan *foreign key* “*id_fk_user*” dengan tabel *user*.

Tabel 4.4 Tabel *address* Levinsky

Field name	Type	Length	Key
<i>id_pk_address</i>	<i>int</i>		<i>Primary key</i>
<i>id_fk_user</i>	<i>int</i>		<i>Foreign key</i>
<i>name</i>	<i>varchar</i>	50	
<i>province</i>	<i>varchar</i>	50	
<i>city</i>	<i>varchar</i>	50	
<i>address</i>	<i>text</i>	100	
<i>phonenum</i>	<i>varchar</i>	25	
<i>is_selected</i>	<i>varchar</i>	10	
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

3) Tabel *notification*

Tabel ini membahas mengenai tabel *address* pada *database* Levinsky yang memiliki *primary key* “*id_pk_address*” dan hubungan *foreign key* “*id_fk_user*” dengan tabel *user*.

Tabel 4.5 Tabel *notification* Levinsky

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>id_pk_notification</i>	<i>int</i>		<i>Primary key</i>
<i>id_fk_user</i>	<i>int</i>		<i>Foreign key</i>
<i>title</i>	<i>varchar</i>	50	
<i>user_message</i>	<i>text</i>	100	
<i>user_notified</i>	<i>varchar</i>	10	
<i>admin_message</i>	<i>text</i>	100	
<i>admin_notified</i>	<i>varchar</i>	10	
<i>link</i>	<i>varchar</i>	50	
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

4) Tabel *cart*

Tabel ini membahas mengenai tabel *cart* pada *database* Levinsky yang memiliki *primary key* “*id_pk_cart*”, hubungan *foreign key* “*id_fk_user*” dengan tabel *user*, dan hubungan *foreign key* “*id_fk_transaction*” dengan tabel *transaction*.

Tabel 4.6 Tabel *cart* Levinsky

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>id_pk_cart</i>	<i>int</i>		<i>Primary key</i>
<i>id_fk_user</i>	<i>int</i>		<i>Foreign key</i>
<i>id_fk_transaction</i>	<i>int</i>		<i>Foreign key</i>
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

5) Tabel *cart_detail*

Tabel ini membahas mengenai tabel *cart_detail* pada *database* Levinsky yang memiliki *primary key* “*id_pk_cart_detail*”, hubungan *foreign key* “*id_fk_cart*”

dengan tabel *cart*, dan hubungan *foreign key* “*id_fk_product_detail*” dengan tabel *product_detail*.

Tabel 4.7 Tabel *cart_detail* Levinsky

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>id_pk_cart_detail</i>	<i>int</i>		<i>Primary key</i>
<i>id_fk_cart</i>	<i>int</i>		<i>Foreign key</i>
<i>id_fk_product_detail</i>	<i>int</i>		<i>Foreign key</i>
<i>size</i>	<i>varchar</i>	10	
<i>length</i>	<i>int</i>		
<i>width</i>	<i>int</i>		
<i>quantity</i>	<i>int</i>		
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

6) Tabel *transaction*

Tabel ini membahas mengenai tabel *transaction* pada *database* Levinsky yang memiliki *primary key* “*id_pk_transaction*” dan hubungan *foreign key* “*id_fk_address*” dengan tabel *address*.

Tabel 4.8 Tabel *transaction* Levinsky

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>id_pk_transaction</i>	<i>int</i>		<i>Primary key</i>
<i>id_fk_address</i>	<i>int</i>		<i>Foreign key</i>
<i>delivery_price</i>	<i>int</i>		
<i>total_price</i>	<i>int</i>		
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

7) Tabel *product*

Tabel ini membahas mengenai tabel *product* pada *database* Levinsky yang memiliki *primary key* “*id_pk_product*”.

Tabel 4.9 Tabel *product* Levinsky

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
-------------------	-------------	---------------	------------

<i>id_pk_product</i>	<i>int</i>		<i>Primary key</i>
<i>name</i>	<i>varchar</i>	50	
<i>price</i>	<i>int</i>		
<i>weight</i>	<i>int</i>		
<i>category</i>	<i>varchar</i>	50	
<i>description</i>	<i>text</i>	100	
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

8) Tabel *product_detail*

Tabel ini membahas mengenai tabel *product_detail* pada database Levinsky yang memiliki *primary key* “*id_pk_product_detail*” dan hubungan *foreign key* “*id_fk_product*” dengan tabel *product*.

Tabel 4.10 Tabel *product_detail* Levinsky

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>id_pk_product_detail</i>	<i>int</i>		<i>Primary key</i>
<i>id_fk_product</i>	<i>int</i>		<i>Foreign key</i>
<i>size</i>	<i>varchar</i>	10	
<i>length</i>	<i>int</i>		
<i>width</i>	<i>int</i>		
<i>quantity</i>	<i>int</i>		
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

9) Tabel *product_image*

Tabel ini membahas mengenai tabel *product_image* pada database Levinsky yang memiliki *primary key* “*id_pk_product_image*” dan hubungan *foreign key* “*id_fk_product*” dengan tabel *product*.

Tabel 4.11 Tabel *product_image* Levinsky

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>id_pk_product_image</i>	<i>int</i>		<i>Primary key</i>
<i>id_fk_product</i>	<i>int</i>		<i>Foreign key</i>

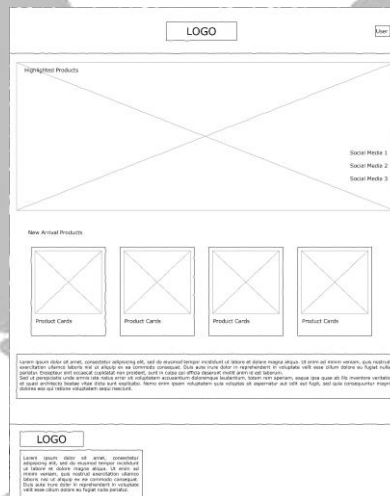
<i>image</i>	<i>varchar</i>	50	
<i>status</i>	<i>varchar</i>	10	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>deleted_at</i>	<i>timestamp</i>		

4.3 Tahap Perancangan

Pada tahap perancangan ini membahas tentang gambaran pembuatan sistem yang dijelaskan melalui *wireframe* dan *user interface*.

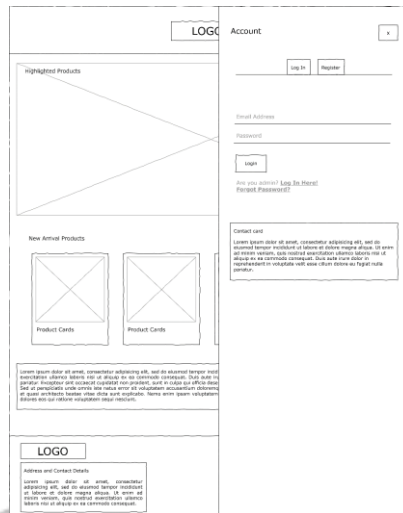
4.3.1 Kerangka Sistem (Wireframe)

Kerangka sistem merupakan sebuah kerangka untuk tataan pada halaman *website* atau aplikasi. Kerangka ini dibuat sebagai gambaran awal pada sistem. Berikut merupakan kerangka untuk sistem usulan:



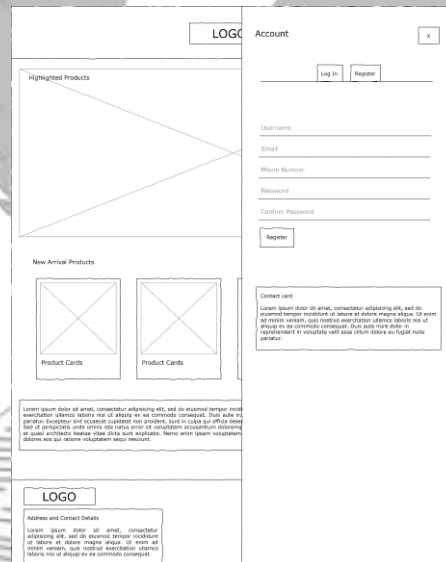
Gambar 4.14 Wireframe Halaman Utama User Levinsky

Pada Gambar 4.14 menunjukkan rencana tampilan halaman utama ketika *user* belum melakukan *login*. Tujuan utama sistem usulan adalah untuk melakukan transaksi, sehingga mayoritas dari isi halaman utama pada sistem usulan menunjukkan produk yang dijual dan informasi terkait perusahaan yang dapat menarik perhatian pembeli.



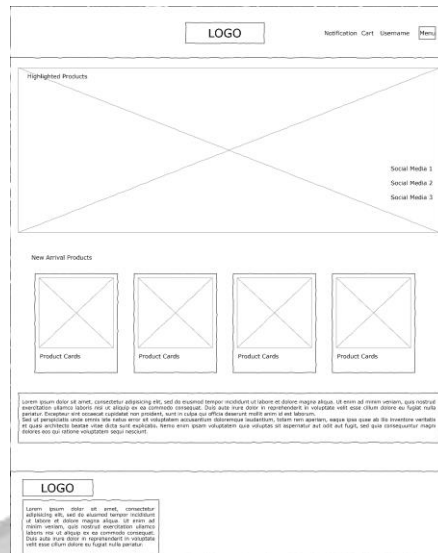
Gambar 4.15 Wireframe Login Sidebar User Levinsky

Ketika ingin melakukan *login*, *user* dapat menekan tombol “*user*” pada Gambar 4.14 yang terletak pada *navigation bar* bagian pojok kanan atas dan kemudian *sidebar login* akan muncul seperti pada Gambar 4.15. *User* dapat mengisi *form login* dan sistem akan memvalidasi data pengguna.



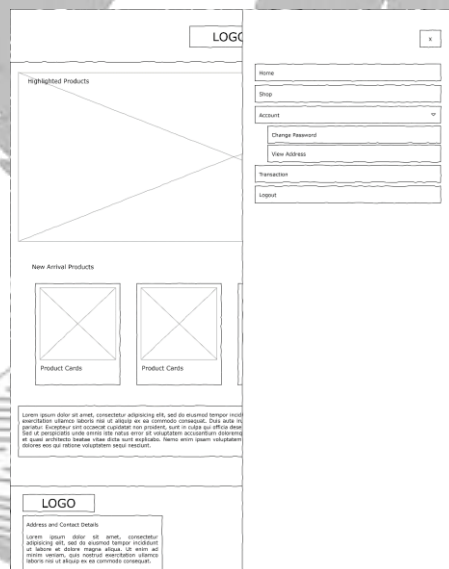
Gambar 4.16 Wireframe Register Sidebar User Levinsky

Jika *user* belum memiliki akun pada sistem, maka *user* perlu mendaftarkan diri terlebih dahulu dengan menekan tombol *register* pada *sidebar “user”* dan sistem akan menampilkan *sidebar register* seperti pada Gambar 4.16.



Gambar 4.17 Wireframe Halaman Utama Member Levinsky

Setelah *user* berhasil *login*, maka level *user* akan berubah menjadi *member* dan *navigation bar* yang dimunculkan akan berbeda menandakan bahwa *member* telah *login*. *Navigation bar* yang muncul berupa: *notification*, *cart*, *username*, dan *menu* seperti pada Gambar 4.17.



Gambar 4.18 Wireframe Menu Sidebar Member Levinsky

Member dapat mengakses seluruh halaman melalui *sidebar menu*. Menu ini dapat menavigasikan ke halaman *shop*, *change password*, *view address*, *transaction*, dan *logout* seperti pada Gambar 4.18.

Gambar 4.19 Wireframe Halaman *Change Password* Member Levinsky

Pada Gambar 4.19, ditunjukkan tampilan halaman *change password* yang berisikan *form* penggantian *password* dan *username*.

Gambar 4.20 Wireframe Halaman *View Address* Member Levinsky

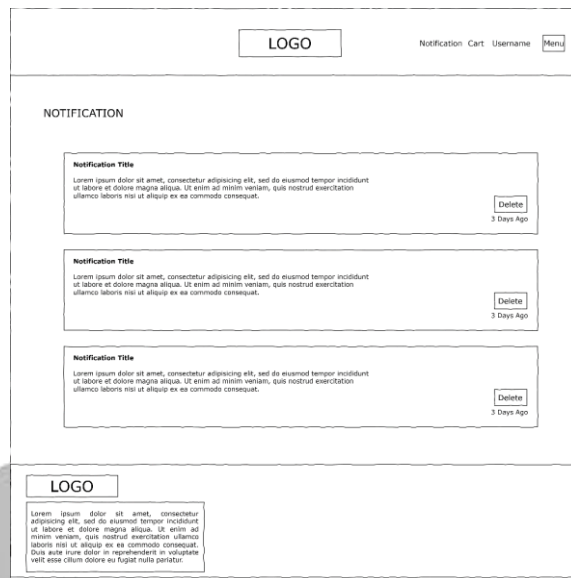
Selanjutnya pada Gambar 4.20, *member* dapat melihat halaman *address* untuk melihat seluruh daftar alamat yang telah didaftarkan pada sistem. Pada halaman ini direncanakan agar *member* dapat menghapus alamat dan memilih alamat sebagai alamat utama.

Gambar 4.21 Wireframe Halaman *Create Address Member* Levinsky

Member dapat membuat alamat baru dengan menekan tombol *add new address* pada Gambar 4.20 dan sistem akan memunculkan halaman *create address* yang berisikan *form* untuk membuat alamat baru seperti pada Gambar 4.21.

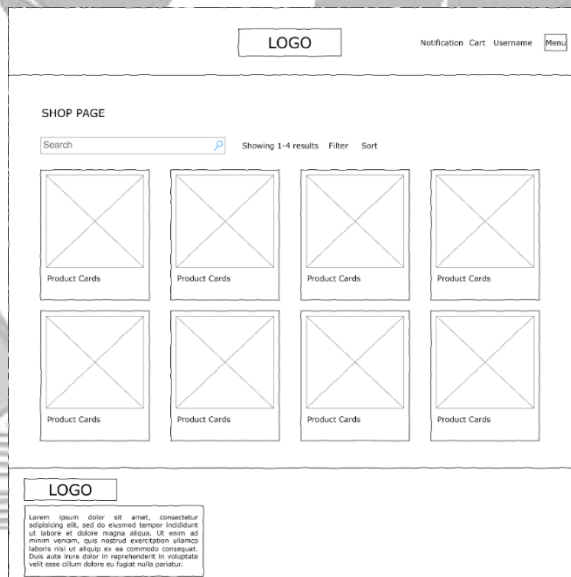
Gambar 4.22 Wireframe *Notification Sidebar Member* Levinsky

Notifikasi *member* direncanakan agar muncul pada *sidebar* sehingga memudahkan akses *member* terhadap notifikasi. *Sidebar notification* akan muncul seperti pada Gambar 4.22 ketika *navigation bar notification* ditekan.



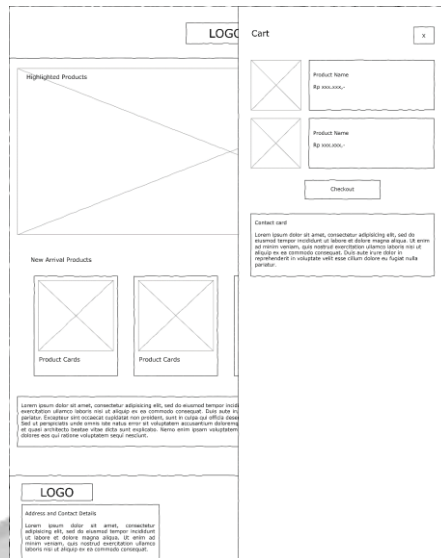
Gambar 4.23 Wireframe Halaman *Notification Member* Levinsky

Jika *member* ingin melihat lebih detail mengenai notifikasi serta *history* dari notifikasi pengguna, maka *member* dapat menekan tombol “*See All Notification*” pada *sidebar* notifikasi dan sistem akan membuka halaman notifikasi seperti pada Gambar 4.23.



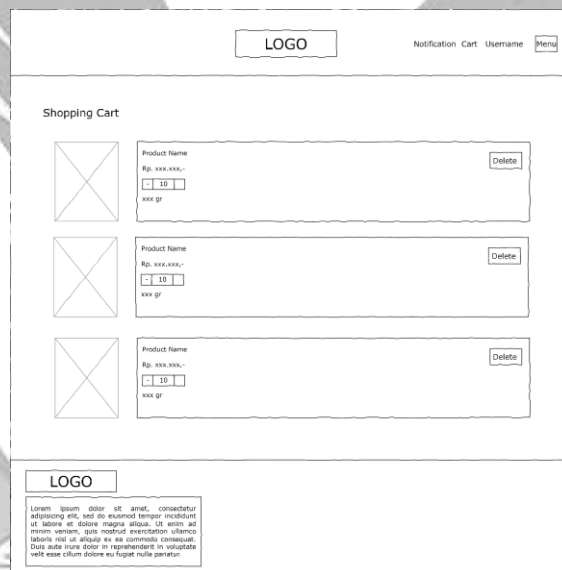
Gambar 4.24 Wireframe Halaman *Product* Levinsky

Halaman *shop* akan ditampilkan ketika *member* mengakses *shop* pada *sidebar menu* seperti pada Gambar 4.24.



Gambar 4.25 Wireframe Cart Sidebar Member Levinsky

Ketika *member* telah memasukkan produk ke keranjang, *member* dapat mengakses *cart* pada *sidebar cart* seperti pada Gambar 4.25.



Gambar 4.26 Wireframe Halaman Cart Member Levinsky

Jika *member* ingin melihat detail *cart*, maka *member* dapat membuka halaman *cart* dengan menekan tombol *checkout* pada Gambar 4.25 dan sistem akan memunculkan halaman seperti Gambar 4.26.

LOGO Notification Cart Username Menu

TRANSACTIONS

Transaction Number: 2022/12/12/X
Name: [Name]
Status: ON PROGRESS
Address: [Address] [10]
Contact: [Contact]
Delivery Price: Rp. xxx,xxx,-
Total Price: Rp. xxx,xxx,-

View details

Product name
XL x 2 Rp. xxx,xxx,-

Transaction Number: 2022/12/12/X
Name: [Name]
Status: ON PROGRESS
Address: [Address] [10]
Contact: [Contact]
Delivery Price: Rp. xxx,xxx,-
Total Price: Rp. xxx,xxx,-

View details

LOGO
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Gambar 4.27 Wireframe Halaman *Transaction Member* Levinsky

Setelah *member* melakukan *checkout*, maka *member* dapat melihat status transaksi pada halaman *transaction* yang ditunjukkan pada Gambar 4.27.

LOGO

Admin Login Session

Email

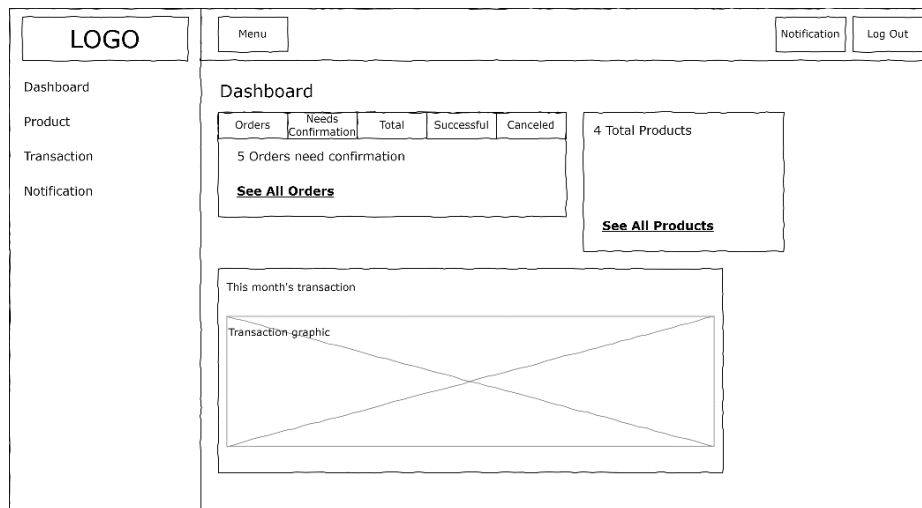
Password

Sign In

Back to Home Page

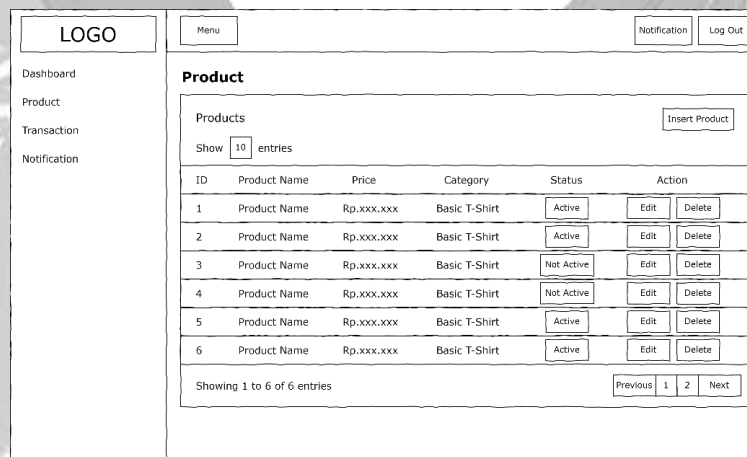
Gambar 4.28 Wireframe Halaman *Login Admin* Levinsky

Jika *user* ingin *login* sebagai *admin*, maka *user* perlu melakukan *login* pada halaman *login admin* seperti yang ditunjukkan pada Gambar 4.28.



Gambar 4.29 Wireframe Halaman *Dashboard Admin* Levinsky

Setelah masuk sebagai *admin*, maka akan muncul halaman *dashboard admin* yang berisikan informasi terkait sistem usulan seperti yang ditunjukkan pada Gambar 4.29.



Gambar 4.30 Wireframe Halaman *Product Admin* Levinsky

Admin dapat mengelola produk halaman ini, *admin* dapat memilih untuk mengaktifkan atau menonaktifkan produk, menghapus produk dan mengedit produk seperti pada Gambar 4.50.

LOGO

Dashboard
Product
Transaction
Notification

Menu

Notification

Log Out

Product Detail

Product name: Levinsky Product

ID

1

Name

Levinsky Product

Price

Rp. xxx.xxx

Description

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Category

Basic t-shirt

Status

Active

Weight

100 gr

Product Stock

Size	Quantity	Width	Length	Action
S	10	45	55	Delete

Add Variant

Save

Gambar 4.31 Wireframe Halaman *View Product Detail Admin Levinsky*

Pada Gambar 4.51 ditunjukkan halaman *view product detail* untuk melihat detail produk sekaligus mengedit informasi produk.

LOGO

Dashboard
Product
Transaction
Notification

Menu

Notification

Log Out

Transaction

Show

10

entries

Search

No	Customer Name	Destination Address	Shipping Cost	Total Price	Date Time	Status	
1	Customer Name	Jalan Raya Serpong	Rp.x.xxx	Rp.x.xxx	2022/12/12	CONFIRMED	View Decline
2	Customer Name	Jalan Raya Serpong	Rp.x.xxx	Rp.x.xxx	2022/12/12	CANCELED	View
3	Customer Name	Jalan Raya Serpong	Rp.x.xxx	Rp.x.xxx	2022/12/12	ON PROGRESS	View Accept
							Delete
4	Customer Name	Jalan Raya Serpong	Rp.x.xxx	Rp.x.xxx	2022/12/12	CONFIRMED	View Delete
5	Customer Name	Jalan Raya Serpong	Rp.x.xxx	Rp.x.xxx	2022/12/12	CONFIRMED	View Delete
6	Customer Name	Jalan Raya Serpong	Rp.x.xxx	Rp.x.xxx	2022/12/12	CONFIRMED	View Delete

Showing 1 to 6 of 6 entries

Previous

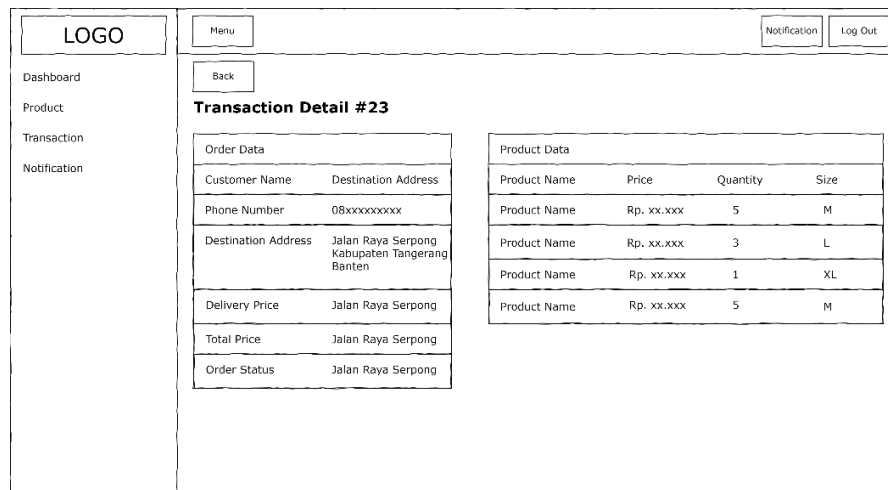
1

2

Next

Gambar 4.32 Wireframe Halaman *Transaction Admin Levinsky*

Pada Gambar 4.52 ditunjukkan halaman *transaction*. Pada halaman ini *admin* dapat mengelola transaksi seperti melihat detail transaksi dan memperbarui status transaksi.



Logo Menu Notification Log Out

Dashboard
Product
Transaction
Notification

Back

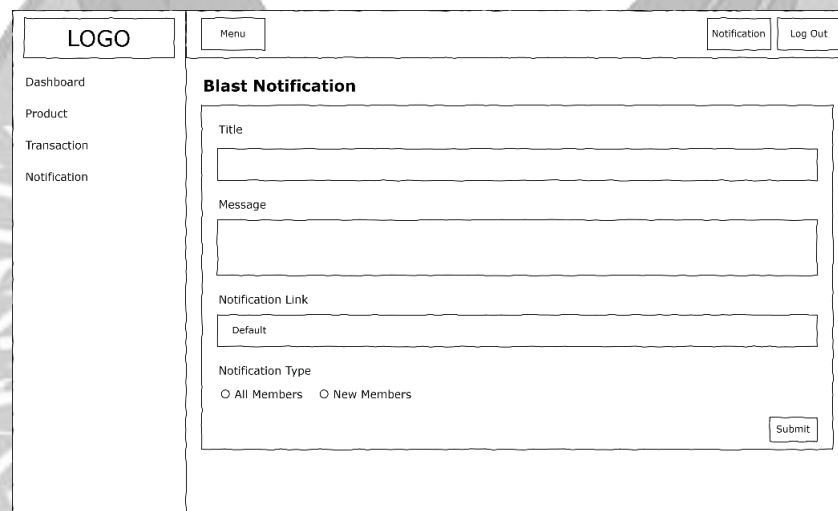
Transaction Detail #23

Order Data	
Customer Name	Destination Address
Phone Number	08xxxxxxxxx
Destination Address	Jalan Raya Serpong Kabupaten Tangerang Banten
Delivery Price	Jalan Raya Serpong
Total Price	Jalan Raya Serpong
Order Status	Jalan Raya Serpong

Product Data			
Product Name	Price	Quantity	Size
Product Name	Rp. xx.xxx	5	M
Product Name	Rp. xx.xxx	3	L
Product Name	Rp. xx.xxx	1	XL
Product Name	Rp. xx.xxx	5	M

Gambar 4.33 Wireframe Halaman View Transaction Detail Admin Levinsky

Jika *admin* memilih untuk melihat detail transaksi, maka sistem akan menampilkan halaman berisi informasi detail seperti pada Gambar 4.53.



Logo Menu Notification Log Out

Dashboard
Product
Transaction
Notification

Blast Notification

Title

Message

Notification Link

Default

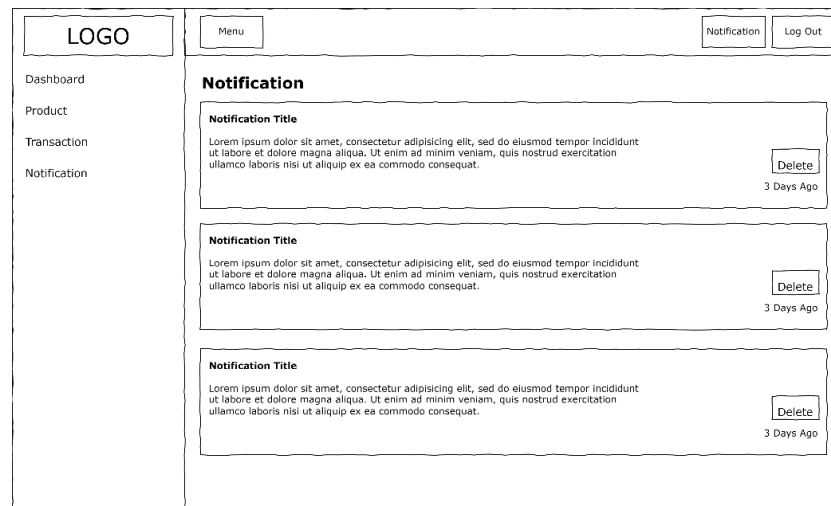
Notification Type

☐ All Members ☐ New Members

Submit

Gambar 4.34 Wireframe Halaman Blast Notification Admin Levinsky

Admin dapat mengirimkan *blast notification* dengan membuka halaman *notification* seperti pada Gambar 4.54 dan mengisi *form* untuk melakukan *blast notification*.

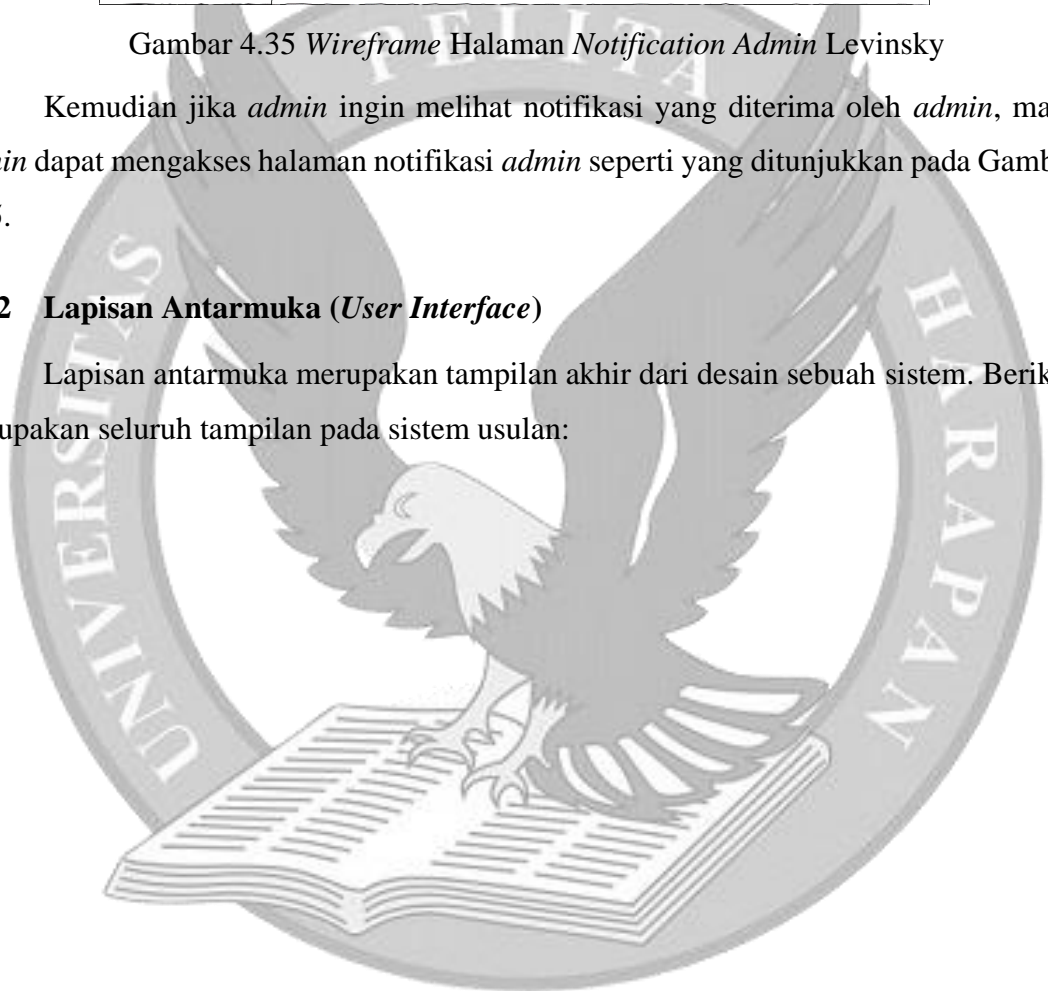


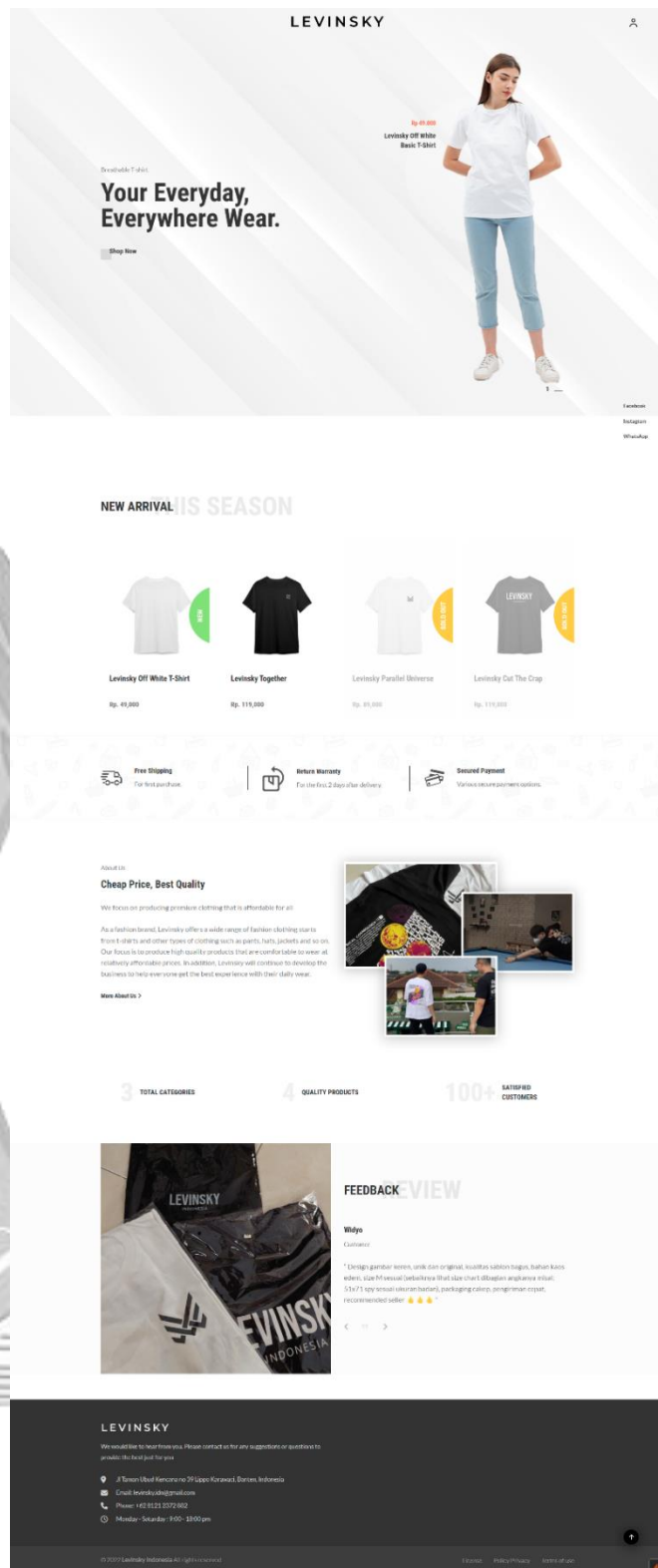
Gambar 4.35 Wireframe Halaman *Notification Admin* Levinsky

Kemudian jika *admin* ingin melihat notifikasi yang diterima oleh *admin*, maka *admin* dapat mengakses halaman notifikasi *admin* seperti yang ditunjukkan pada Gambar 4.55.

4.3.2 Lapisan Antarmuka (*User Interface*)

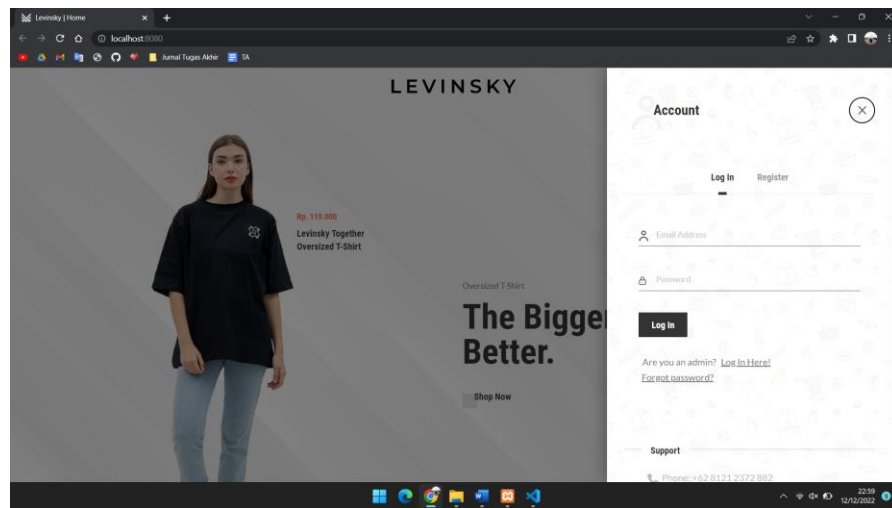
Lapisan antarmuka merupakan tampilan akhir dari desain sebuah sistem. Berikut merupakan seluruh tampilan pada sistem usulan:





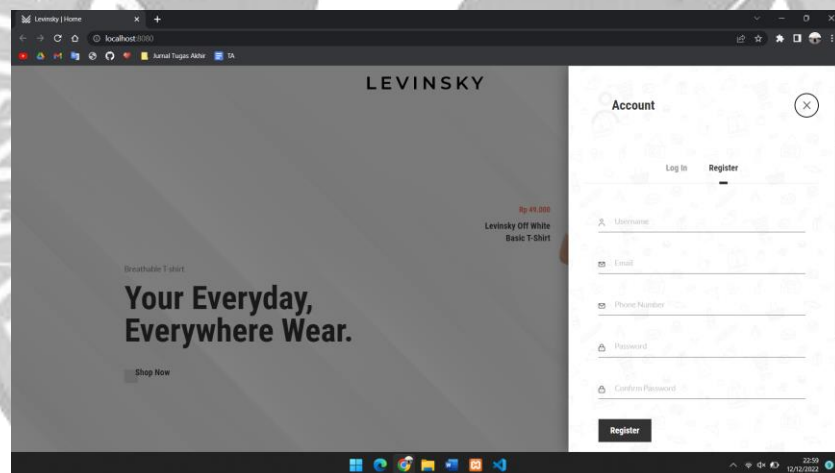
Gambar 4.36 Halaman Utama Levinsky

Gambar 4.36 merupakan tampilan utama yang dilihat oleh *user* ketika mengakses sistem Levinsky. Pada halaman ini *user* dapat melihat produk terbaru dari Levinsky, melihat informasi singkat terkait perusahaan Levinsky, dan melihat beberapa *feedback* dari *customer* tentang pembelian sebelumnya.



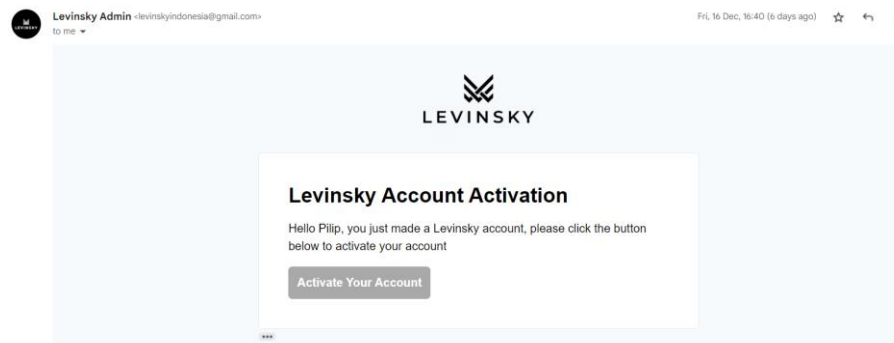
Gambar 4.37 Sidebar Login Levinsky

Ketika *user* menekan tombol akun pada pojok kanan atas, maka *sidebar* akun akan terbuka sehingga *user* dapat melakukan *login* dengan menginput data berupa *email* dan *password*, dan kemudian dapat menekan tombol *login* untuk melakukan autentikasi pengguna *member*.



Gambar 4.38 Sidebar Register Levinsky

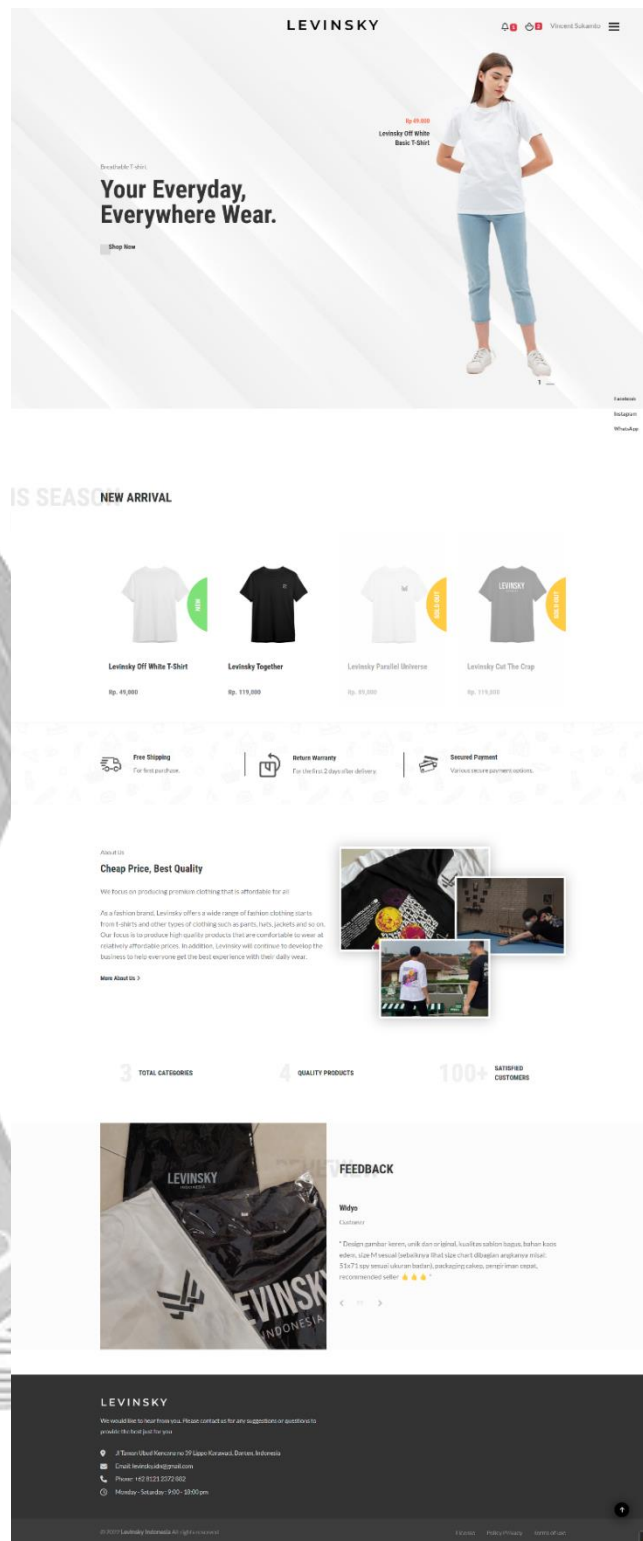
Selain dapat melakukan *login* pada *sidebar* akun, *user* juga dapat melakukan registrasi pengguna *member* dengan memasukkan *username*, *email*, nomor *handphone*, *password* dan *confirm password*. Pada *form* ini, terdapat validasi berupa kelengkapan *form*, keunikan *email* dan kecocokan *password* dengan *confirm password*.



Gambar 4.39 Aktivasi Akun Levinsky pada Email

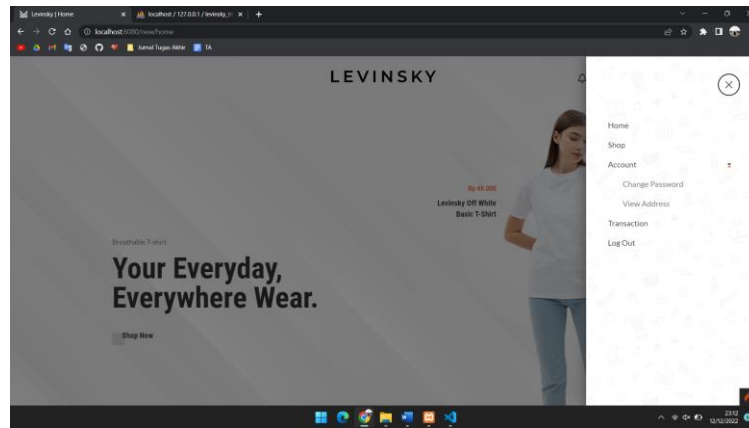
Setelah *user* melakukan registrasi, *user* dapat mengaktifkan akun pada *email* yang diregistrasikan untuk mengaktifkan akun dengan menekan tombol “*Activate Your Account*” seperti pada Gambar 4.39.





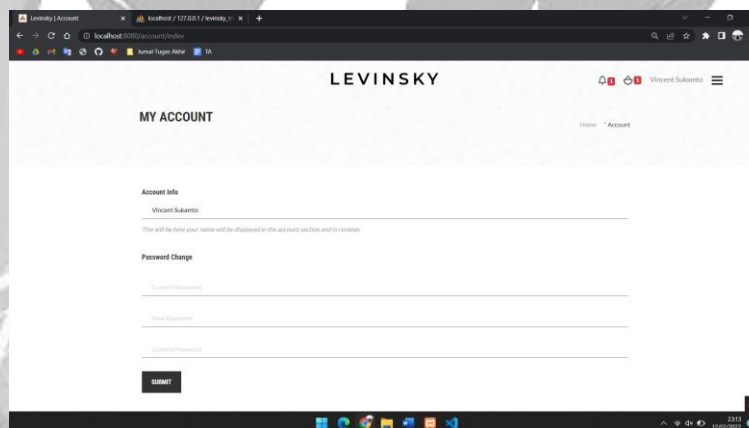
Gambar 4.40 Halaman Utama *Member (Logged In)* Levinsky

Pada Gambar 4.40, ditunjukkan halaman utama setelah *user* melakukan *login* menjadi *member*. Terdapat perbedaan pada *navigation bar* yang sebelumnya hanya berupa *icon* akun untuk melakukan *login* menjadi beberapa *navigation* seperti *notification*, *cart*, dan *menu*.



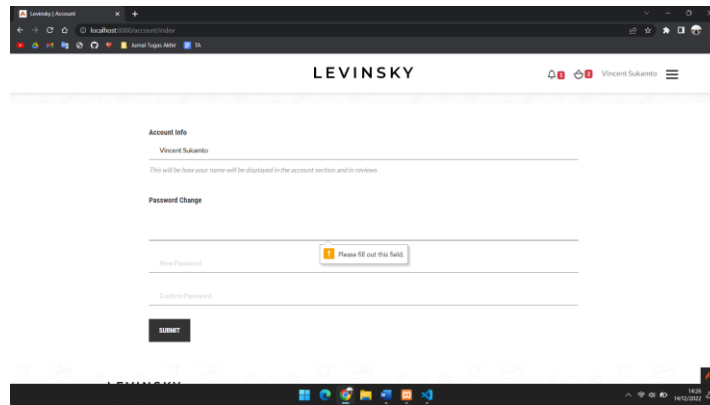
Gambar 4.41 *Sidebar Menu Member Levinsky*

Member dapat menekan tombol *menu* di pojok kanan dan kemudian *menu sidebar* akan muncul dan menampilkan seluruh halaman yang dapat diakses oleh *member*. Terdapat *menu home* untuk navigasi ke halaman *home*, *shop* untuk navigasi ke halaman produk, *account* yang berisikan *change password* untuk mengganti *password* dan *view address* untuk navigasi ke halaman pengaturan alamat, *transaction* untuk navigasi ke halaman transaksi, dan *logout* untuk menonaktifkan sesi *member*.



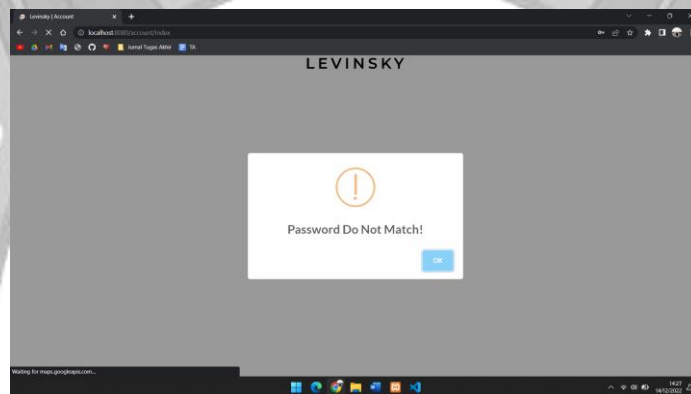
Gambar 4.42 *Halaman Change Password Member Levinsky*

Jika *member* ingin mengubah *password*, maka halaman *change password* akan tampil setelah *member* menekan halaman pada *menu sidebar*. Pada halaman ini, *member* dapat mengubah *username* dan mengganti *password* lama dengan memasukkan *password* lama beserta *password* baru yang akan kecocokannya. Jika sudah sesuai, maka *password* akan diubah.



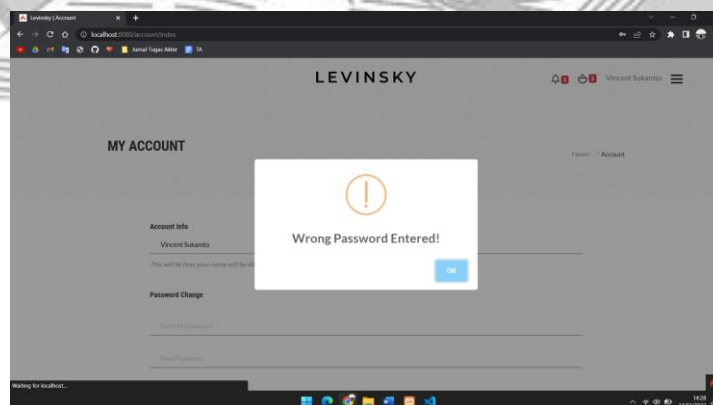
Gambar 4.43 Validasi Kelengkapan *Form* pada Halaman *Change Password* Levinsky

Jika *form* tidak terisi dengan lengkap maka akan ada peringatan untuk memenuhi *form* penggantian *password* pada kolom yang belum terisi seperti pada Gambar 4.43.



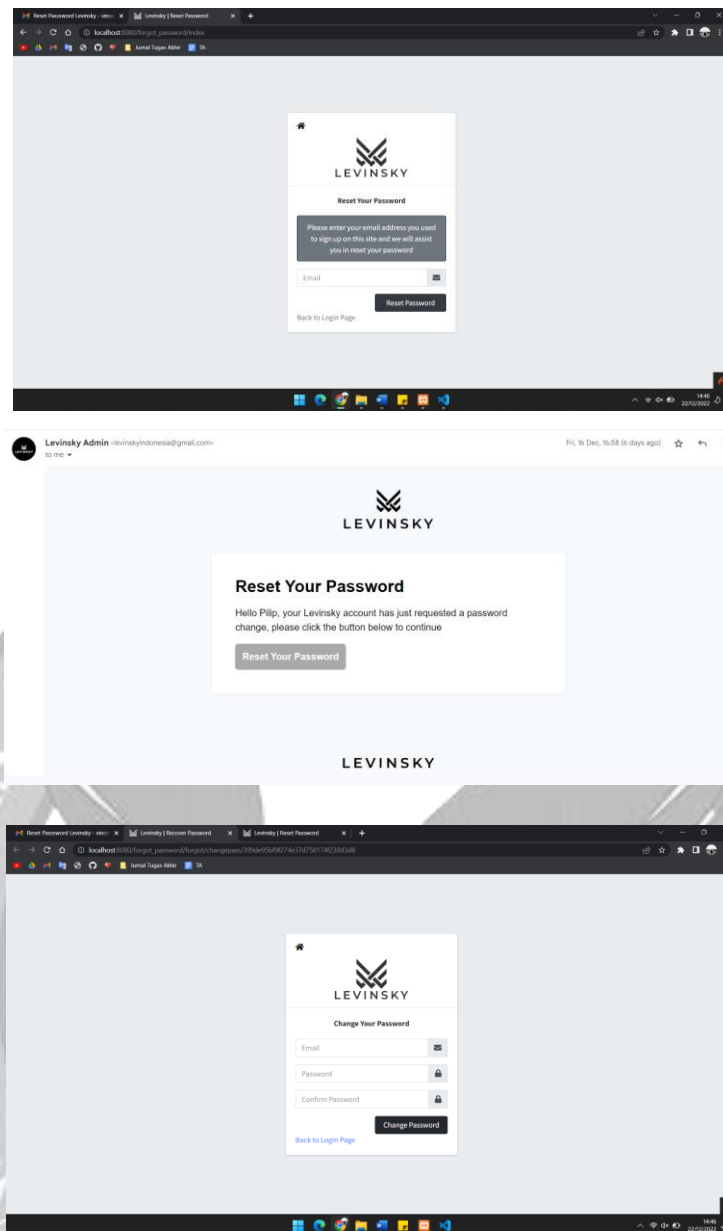
Gambar 4.44 Validasi Kecocokan *Password* pada *Form Change Password* Levinsky

Jika *password* baru yang dimasukkan tidak cocok dengan *confirm password*, maka akan muncul peringatan seperti pada Gambar 4.44 dan penggantian *password* akan dibatalkan.



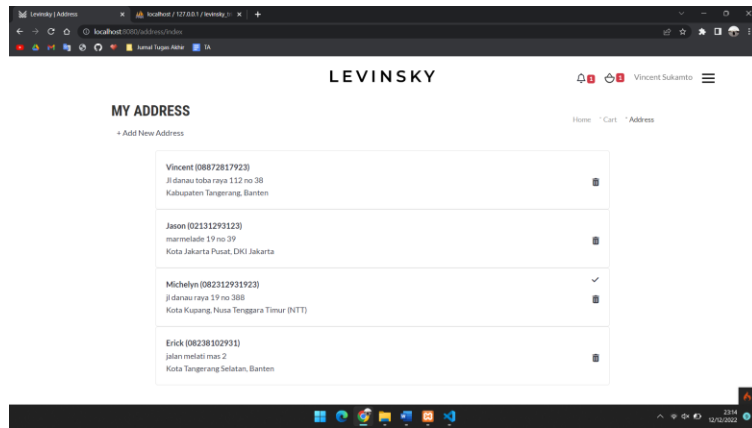
Gambar 4.45 Validasi *Password* Lama pada *Form Change Password* Levinsky

Jika *password* lama yang di *input* oleh *member* salah, maka akan muncul peringatan seperti pada Gambar 4.45 dan penggantian *password* akan dibatalkan.



Gambar 4.46 *Forget Password* pada *Email*

Jika user lupa dengan *password* akunnya, maka user dapat mengirimkan email untuk melakukan penggantian *password* dan dapat menekan tombol “*Reset Your Password*” pada email dan mengisi *form recover password* seperti pada Gambar 4.46.

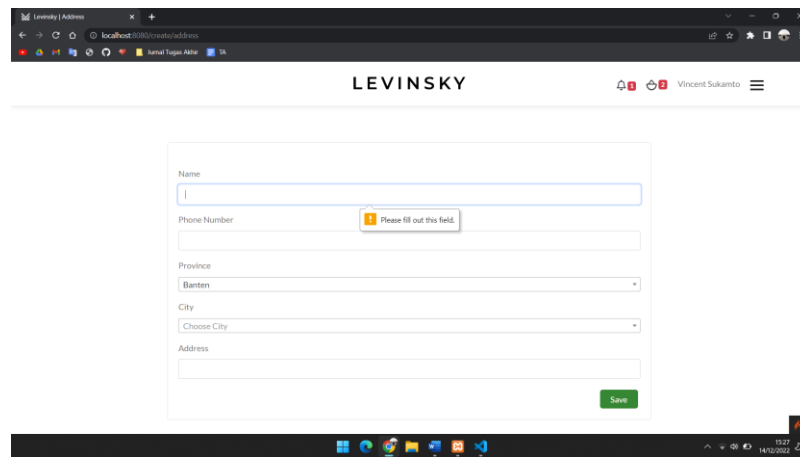


Gambar 4.47 Halaman *View Address Member* Levinsky

Pada halaman ini, *member* dapat melihat seluruh alamat yang pernah disimpan pada akun *member* seperti pada Gambar 4.47. Selain itu *member* juga dapat menghapus alamat dan memilih alamat sebagai alamat utama untuk pengiriman pada transaksi selanjutnya.

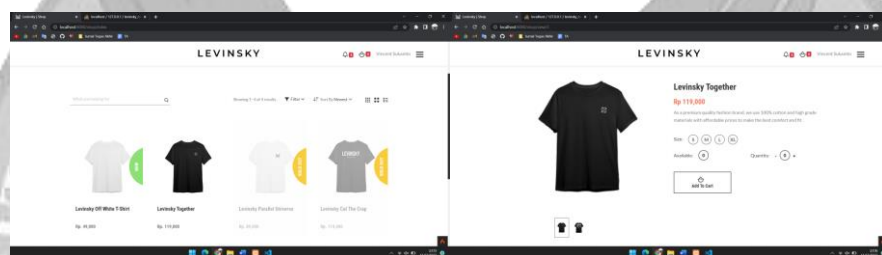
Gambar 4.48 Halaman *Create Address Member* Levinsky

Pada halaman berikut terdapat *form* pembuatan alamat yang dapat di *input* oleh *member* berupa nama penerima alamat, nomor *handphone* penerima, provinsi, kota/kabupaten, dan detail alamat penerima.



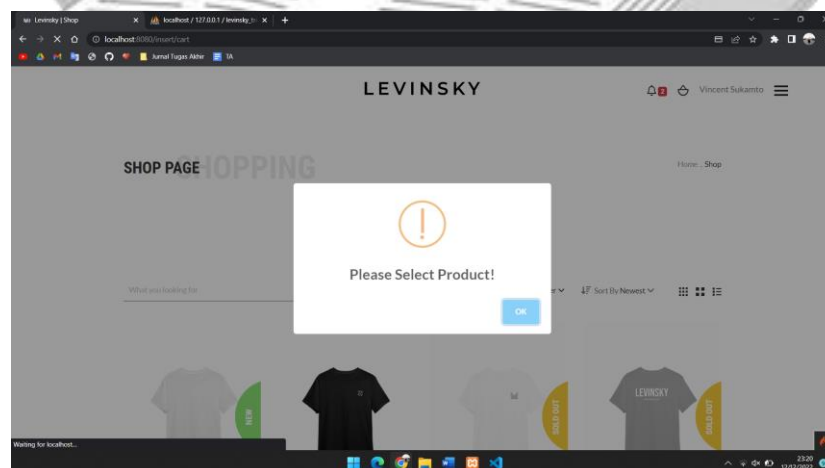
Gambar 4.49 Validasi Kelengkapan *Form* pada Halaman *Create Address* Levinsky

Jika *member* tidak melengkapi *form* dengan lengkap, maka akan muncul peringatan seperti pada Gambar 4.49.



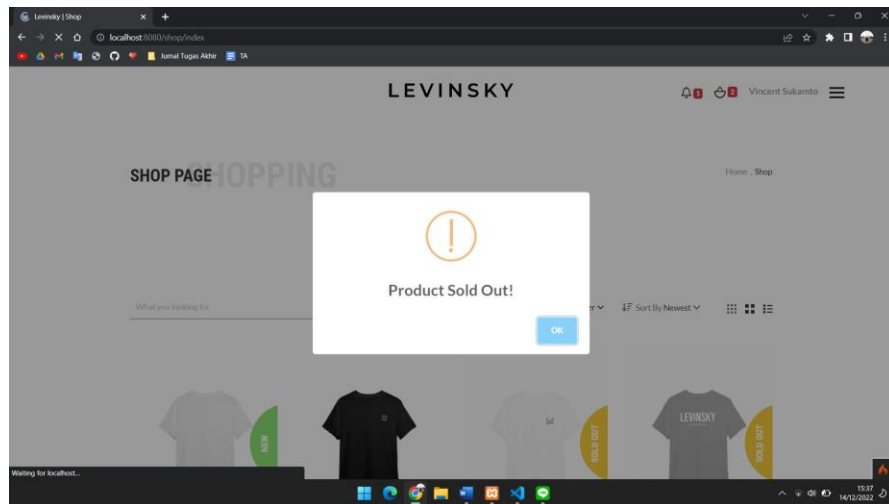
Gambar 4.50 Halaman *Shop* Levinsky

Setelah *member* menekan *menu shop* pada *sidebar menu*, sistem akan menampilkan produk-produk yang tersedia sesuai dengan yang tersimpan pada sistem seperti pada Gambar 4.50. *Member* dapat memilih produk yang diinginkan dan produk akan ditampilkan secara mendetail. Kemudian *member* dapat memilih varian produk yang diinginkan seperti ukuran dan jumlah yang diinginkan.



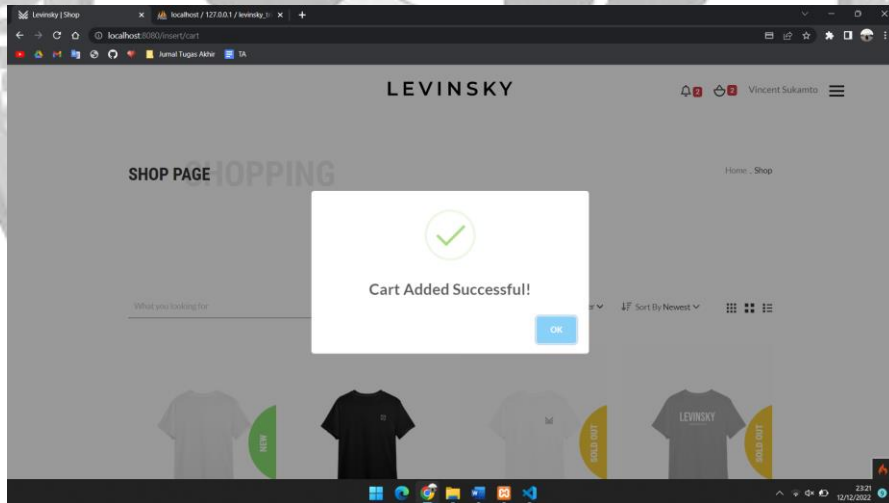
Gambar 4.51 Validasi Pemilihan Varian Produk pada Aktivitas *Insert Cart* Levinsky

Jika *member* tidak memilih varian pada halaman *shop view*, maka akan muncul peringatan seperti pada Gambar 4.51 dan sistem akan membatalkan penyimpanan data produk pada *cart*.



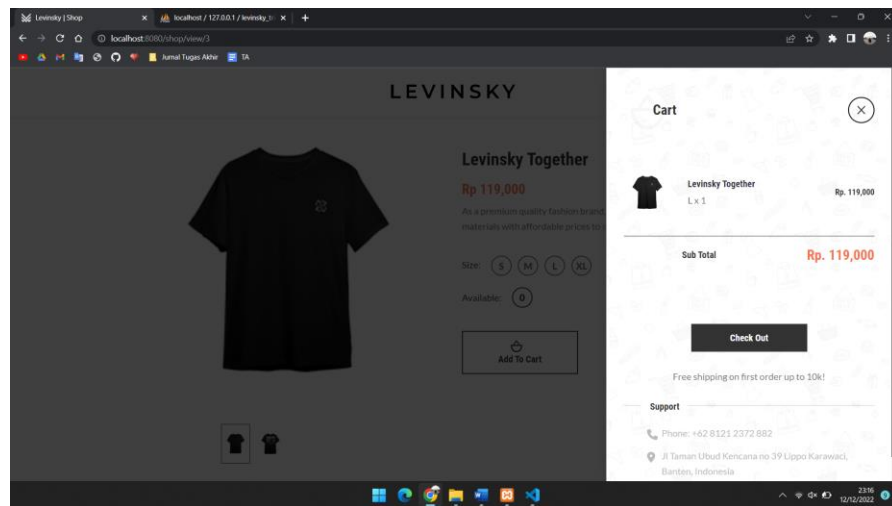
Gambar 4.52 Validasi Ketersediaan Produk pada Aktivitas *Insert Cart* Levinsky

Jika *member* memilih varian yang tidak tersedia pada halaman *shop view*, maka akan muncul peringatan seperti pada Gambar 4.52 dan sistem akan membatalkan penyimpanan data produk pada *cart*.



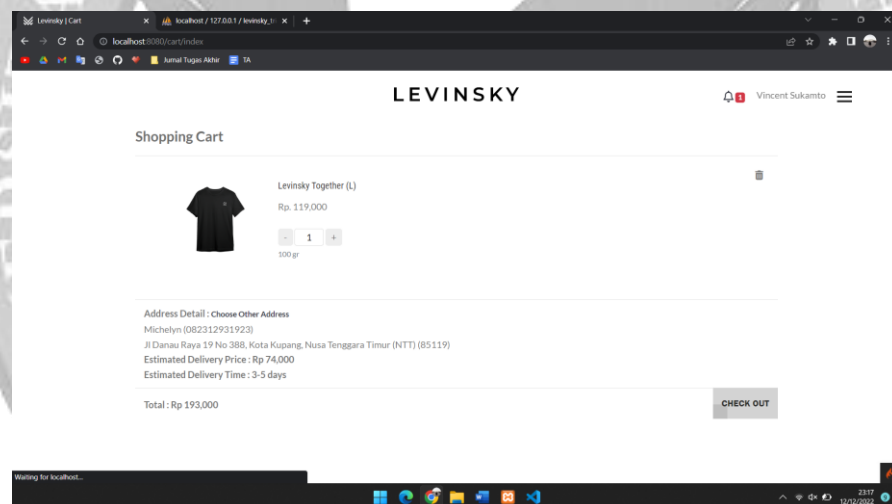
Gambar 4.53 Produk Berhasil Ditambahkan pada Aktivitas *Insert Cart* Levinsky

Jika *member* memilih varian yang tersedia pada halaman *shop view* dan sesuai dengan kecocokan yang ada pada sistem, maka akan muncul peringatan berhasil seperti pada Gambar 4.53 dan sistem akan menambahkan data produk pada *cart*.



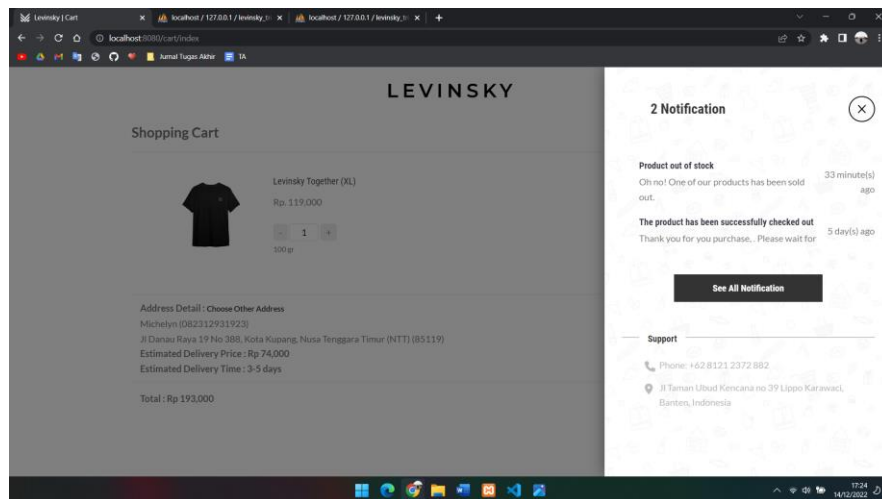
Gambar 4.54 *Sidebar Cart Member Levinsky*

Pada Gambar 4.54 ditunjukkan *sidebar cart* yang berisikan produk yang telah dimasukkan pada *cart* sebelumnya.



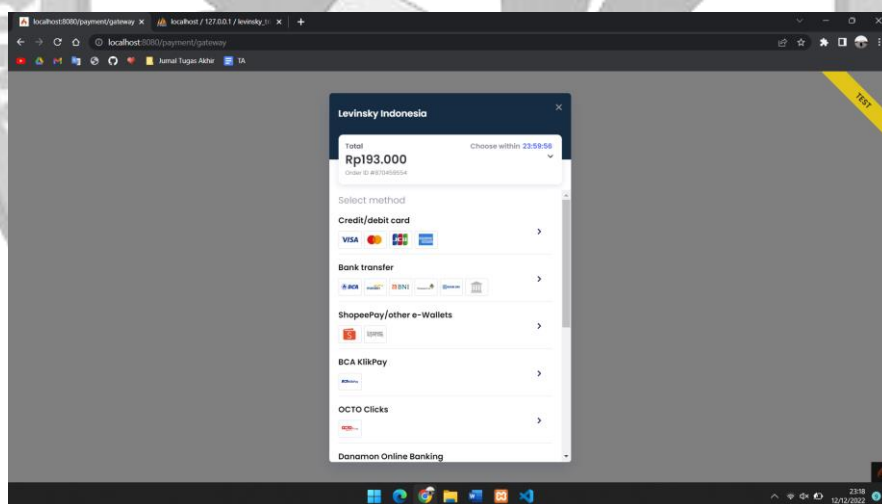
Gambar 4.55 *Halaman Cart Member Levinsky*

Ketika *member* menekan tombol *checkout* pada *sidebar cart*, maka halaman cart akan tampil seperti pada Gambar 4.55. Saat halaman ini dibuka, maka akan ada pengecekan ketersediaan varian produk.



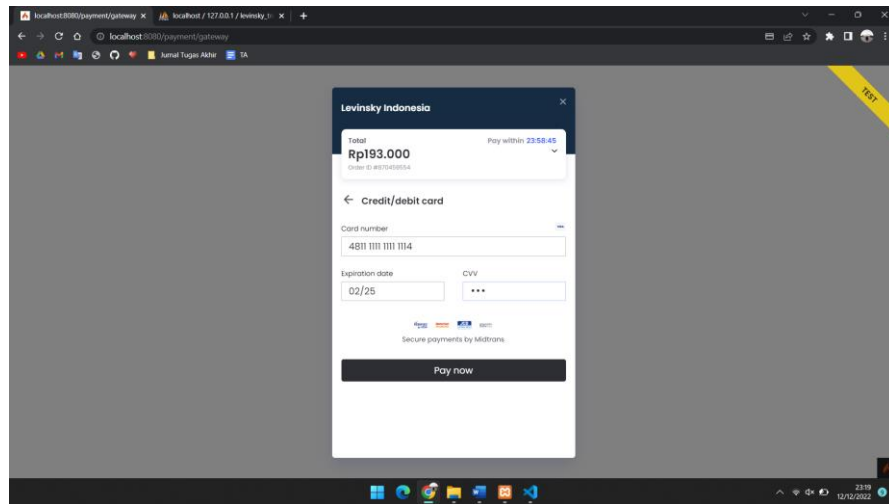
Gambar 4.56 Notifikasi Ketersediaan *Product Cart* Levinsky

Jika produk tidak tersedia ketika *member* melakukan *checkout*, maka produk akan dihapus dari *cart* dan sistem akan mengirimkan notifikasi produk yang tidak tersedia seperti pada Gambar 4.56. Kemudian pada halaman *cart*, sistem melakukan perhitungan ongkos kirim sesuai dengan jumlah berat produk dan alamat pengiriman dengan API RajaOngkir.



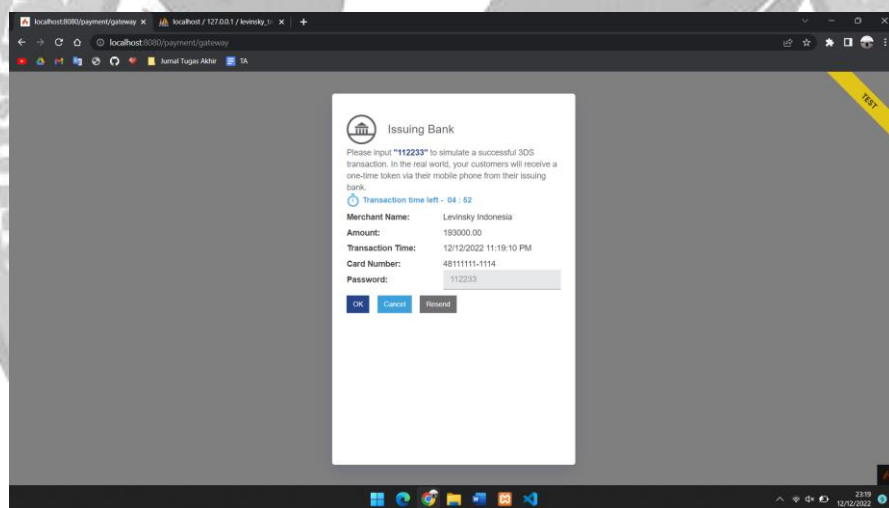
Gambar 4.57 Halaman *Checkout* API Midtrans

Setelah *member* menekan tombol *checkout* pada halaman *cart*, maka sistem akan mengalihkan ke halaman API Midtrans untuk melanjutkan pembayaran. Pada saat ini, sistem mengirimkan informasi berupa jumlah yang harus dibayar, *email* dan nomor *handphone*. Pada pengembangan sistem usulan ini menggunakan pembayaran *testing* (percobaan) dengan metode pembayaran *credit card*.



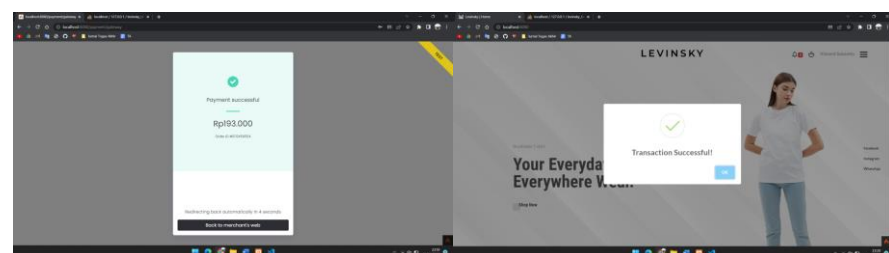
Gambar 4.58 Pembayaran *Credit Card* API Midtrans

Setelah *member* memilih metode *credit card*, maka akan muncul tampilan seperti pada Gambar 4.58. *Member* dapat menginput data *card number* “4811 1111 1111 1114”, *expiration date* “02/25”, dan CVV “123” sesuai dengan dokumentasi pada <https://docs.midtrans.com/en/snap/interactive-demo>.



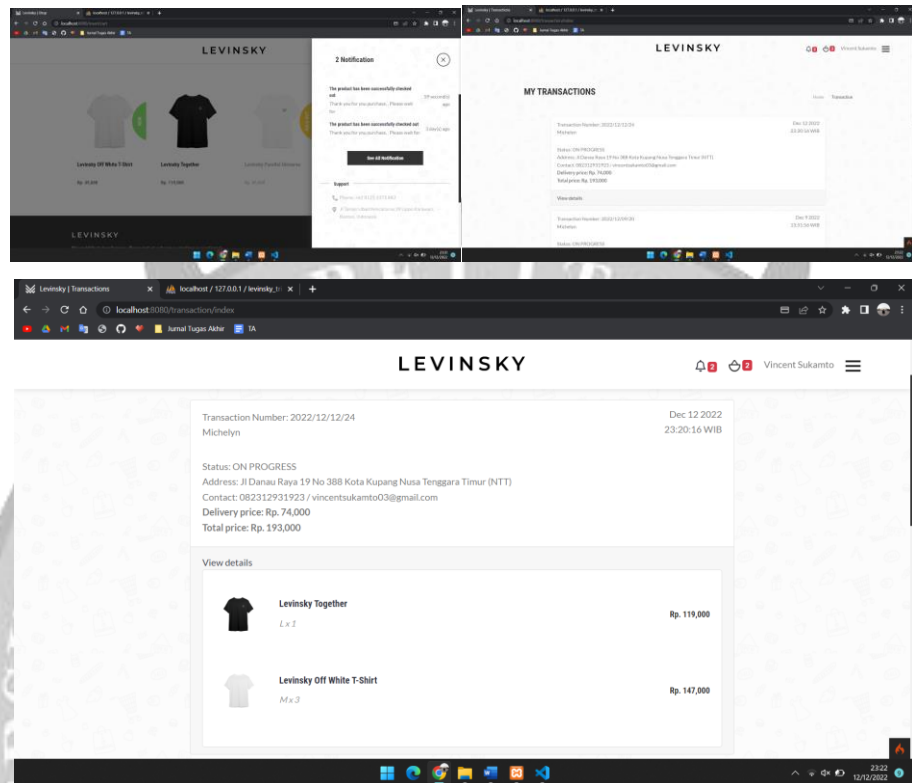
Gambar 4.59 OTP Pembayaran API Midtrans

Kemudian *member* dapat memasukkan OTP sesuai dengan instruksi yang diberikan oleh UI API Midtrans seperti pada Gambar 4.59.



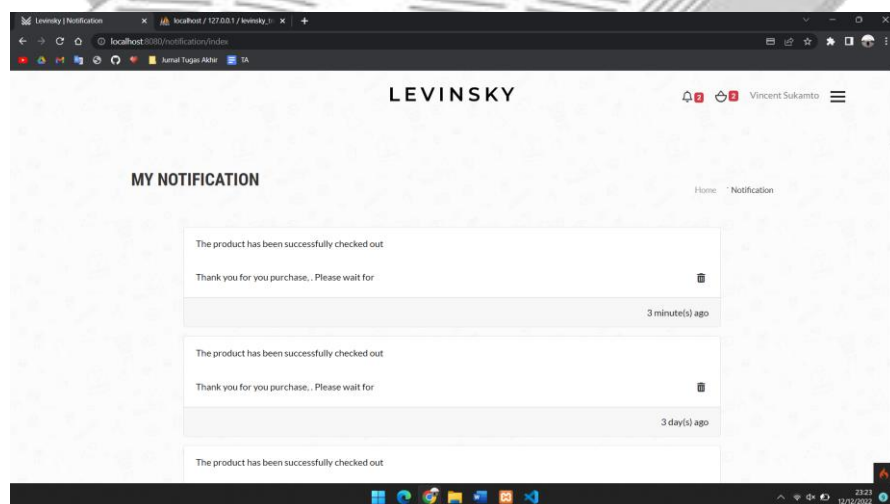
Gambar 4.60 Pembayaran Berhasil API Midtrans

Setelah pembayaran berhasil maka sistem akan mencatat transaksi yang telah dibuat dan menonaktifkan *cart* yang telah ditransaksikan. Kemudian sistem akan memunculkan *feedback* transaksi berhasil seperti pada Gambar 4.60.



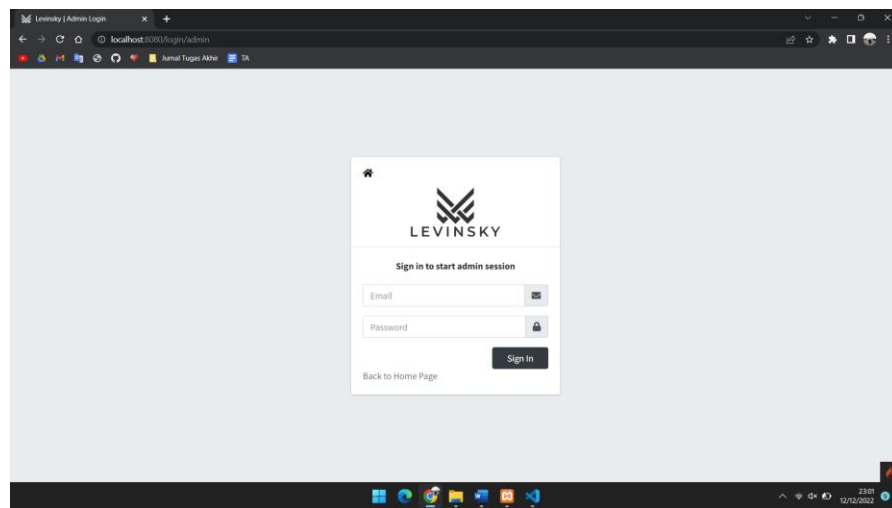
Gambar 4.61 Notifikasi Transaksi Berhasil Levinsky

Setelah transaksi berhasil, sistem akan memberikan notifikasi keberhasilan transaksi dan *member* dapat memantau kembali transaksi yang telah dilakukan pada sistem dan melihat detail transaksi yang telah dilakukan.



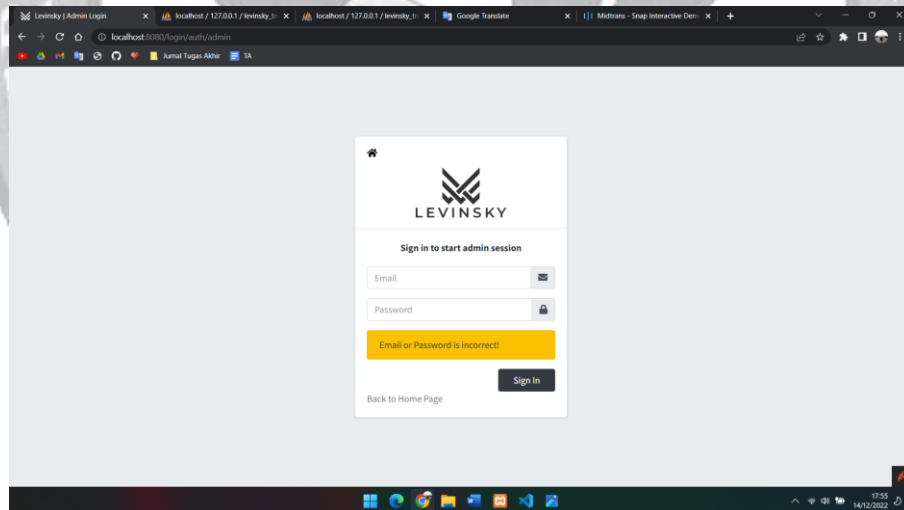
Gambar 4.62 Halaman Notifikasi *Member* Levinsky

Jika *member* ingin melihat kembali notifikasi yang telah diterima selama *member* menggunakan layanan sistem, maka sistem akan menampilkan notifikasi sebelumnya yang pernah diterima oleh *member*.



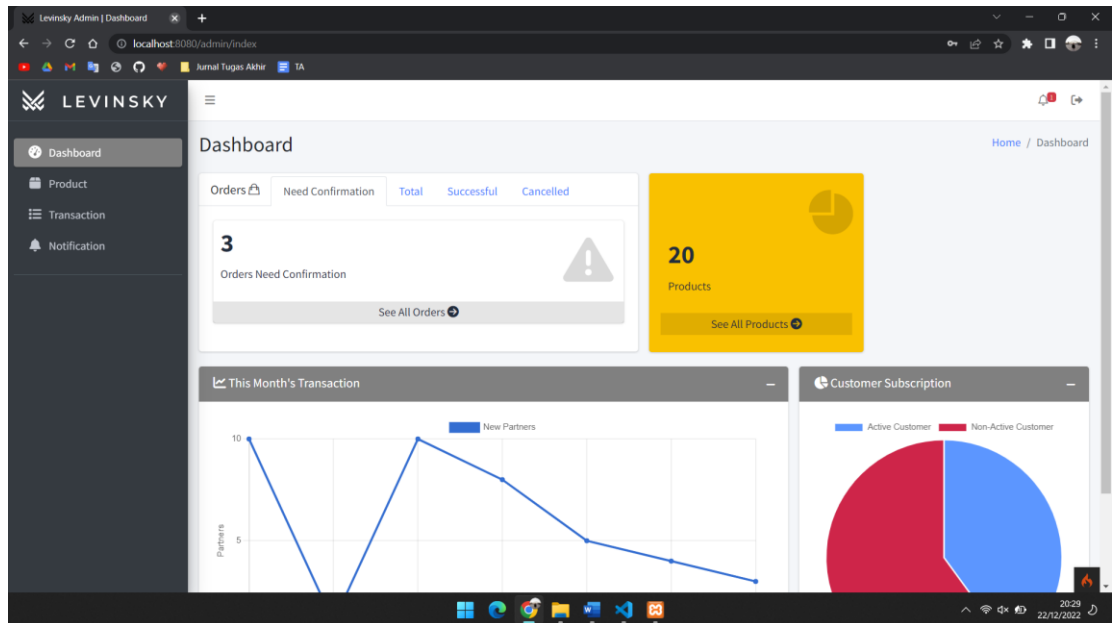
Gambar 4.63 Halaman *Login Admin* Levinsky

Ketika *user* ingin *login* sebagai *admin*, maka sistem akan mengarahkan *user* ke halaman *login admin* seperti pada Gambar 4.63 dan *user* diminta untuk memasukkan *email* serta *password admin* untuk memperoleh akses *admin*.



Gambar 4.64 Validasi *Form Login Admin* Levinsky

Sistem akan memvalidasi kecocokan data *login admin*, jika tidak cocok maka sistem akan memberikan peringatan seperti pada Gambar 4.64 dan menggagalkan *login admin*.



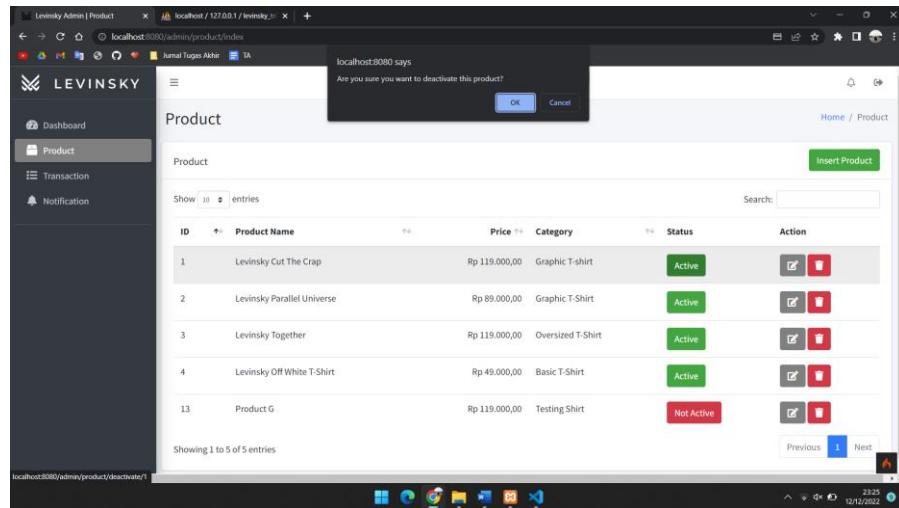
Gambar 4.65 Halaman *Dashboard Admin* Levinsky

Setelah berhasil *login* sebagai *admin*, maka sistem akan menampilkan halaman *dashboard admin* seperti pada Gambar 4.65. Halaman ini menampilkan sejumlah informasi terkait transaksi maupun produk pada Levinsky. Selain itu juga terdapat *sidebar menu* yang berupa *dashboard* yang menampilkan halaman *dashboard*, *product* yang menampilkan halaman pengelolaan produk, *transaction* yang menampilkan halaman pengelolaan transaksi, dan *notification* yang menampilkan halaman untuk mengirimkan *blast notification* pada *member*.

ID	Product Name	Price	Category	Status	Action
1	Levinsky Cut The Crap	Rp 119.000,00	Graphic T-Shirt	Active	
2	Levinsky Parallel Universe	Rp 89.000,00	Graphic T-Shirt	Active	
3	Levinsky Together	Rp 119.000,00	Oversized T-Shirt	Active	
4	Levinsky Off White T-Shirt	Rp 49.000,00	Basic T-Shirt	Active	
13	Product G	Rp 119.000,00	Testing Shirt	Not Active	

Gambar 4.66 Halaman *Product Admin* Levinsky

Pada halaman ini, *admin* dapat mengelola produk seperti mengaktifkan atau menonaktifkan produk, menyunting produk dan menghapus produk.



Gambar 4.67 Validasi Mengaktifkan atau Menonaktifkan Produk *Admin* Levinsky

Sistem akan menampilkan validasi untuk kepastian mengaktifkan atau menonaktifkan produk seperti pada Gambar 4.67, jika *admin* memilih “OK” maka sistem akan mengaktifkan atau menonaktifkan produk. Jika *admin* memilih “Cancel” maka sistem akan membatalkan pengaktifan atau penonaktifan produk.

 The screenshot shows the 'Add Product' form in the Levinsky Admin. The form includes input fields for Product Name, Price, Picture (with a 'Choose File' button and 'Browse' link), Category, Description, and Weight. Below these fields is a 'Product Stock' section with a table for adding variants. The table has columns for Size, Quantity, Width, Length, and Action. At the bottom of the form is an 'Add Variant' button and a 'Submit' button.

Gambar 4.68 Halaman *Add Product Admin* Levinsky

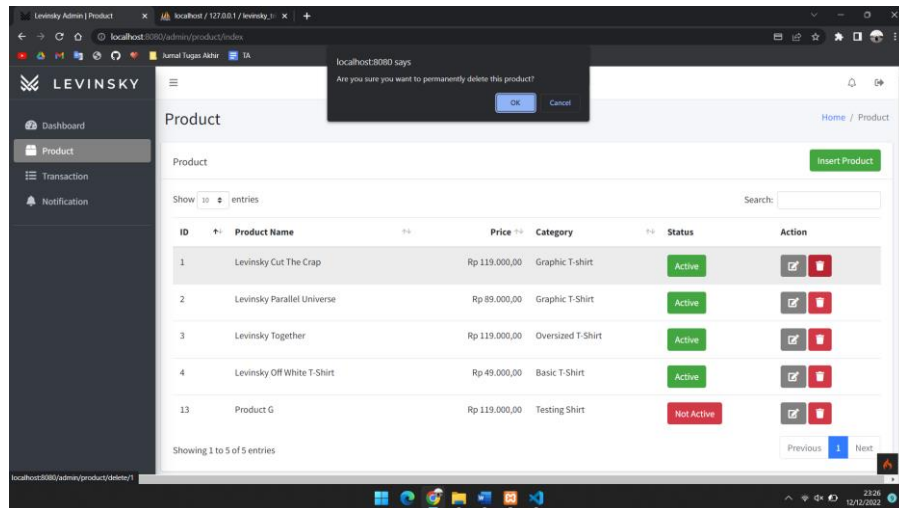
Ketika admin memilih “*Insert Product*” pada halaman *product admin*, maka sistem akan menampilkan halaman *form* tambah produk baru seperti pada Gambar 4.68.

Gambar 4.69 Validasi Kelengkapan *Form* Tambah Produk Admin Levinsky

Admin dapat mengisi *form* tambah produk dan sistem akan memvalidasi kelengkapan *form* tambah produk dan menampilkan peringatan pada *form* yang belum terisi seperti pada Gambar 4.69.

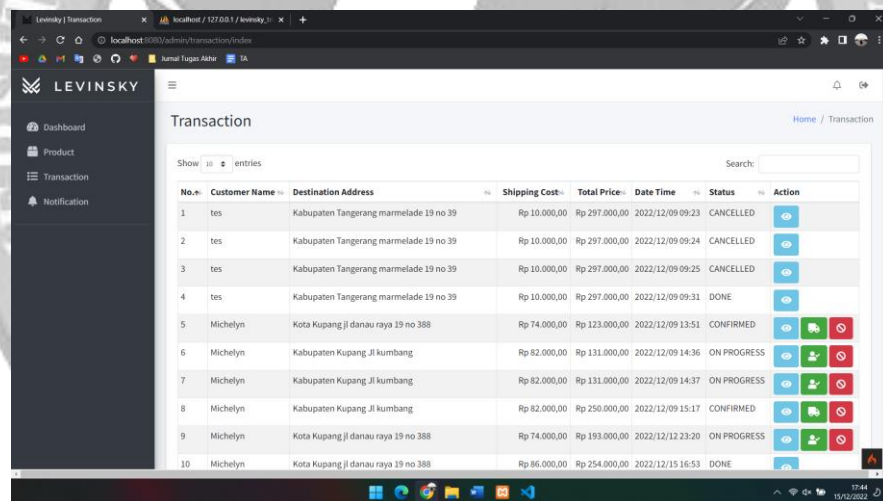
Gambar 4.70 Halaman *Edit Product* Admin Levinsky

Jika *admin* memilih untuk menyunting produk pada halaman mengelola produk, maka akan muncul tampilan seperti pada Gambar 4.70 dan *admin* dapat mengubah detail informasi produk dan kemudian menekan tombol “*Save*”.



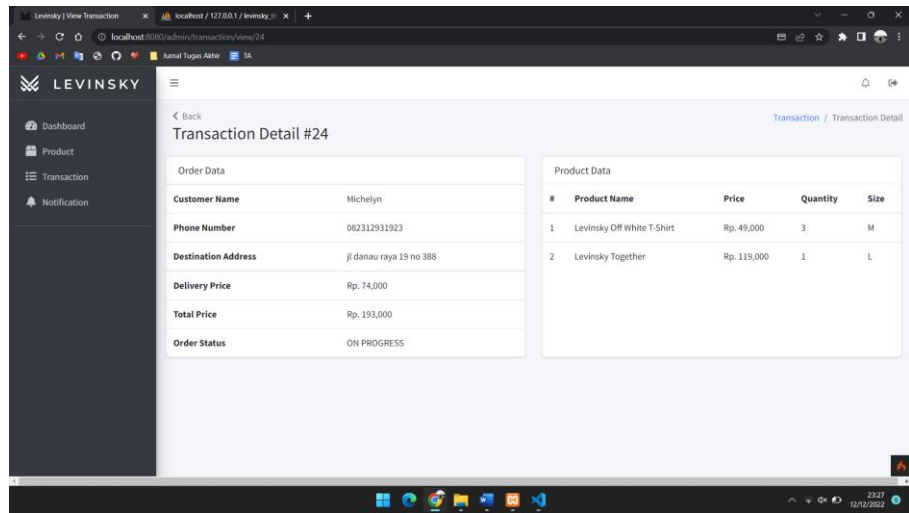
Gambar 4.71 Validasi Hapus Produk Admin Levinsky

Jika *admin* memilih untuk menghapus produk dari *list* produk yang ada, maka *admin* dapat menekan tombol hapus dan akan muncul peringatan seperti pada Gambar 4.71, jika *admin* memilih “OK” maka sistem akan menghapus produk. Jika *admin* memilih “Cancel” maka sistem akan membatalkan penghapusan produk.



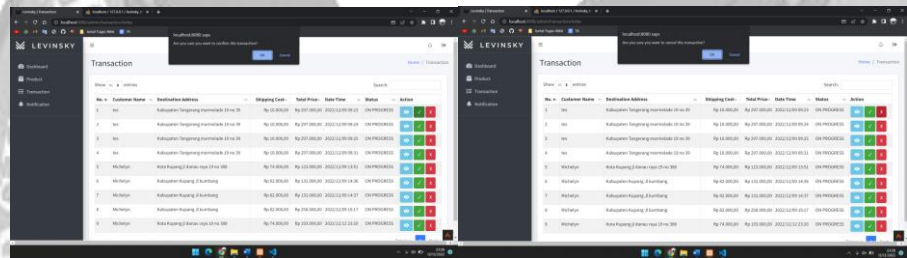
Gambar 4.72 Halaman Transaction Admin Levinsky

Pada halaman ini, *admin* dapat mengelola transaksi seperti melihat detail transaksi, menerima transaksi dan menolak transaksi jika status transaksi adalah “ON PROGRESS”, membatalkan transaksi jika status transaksi adalah “CONFIRMED”.



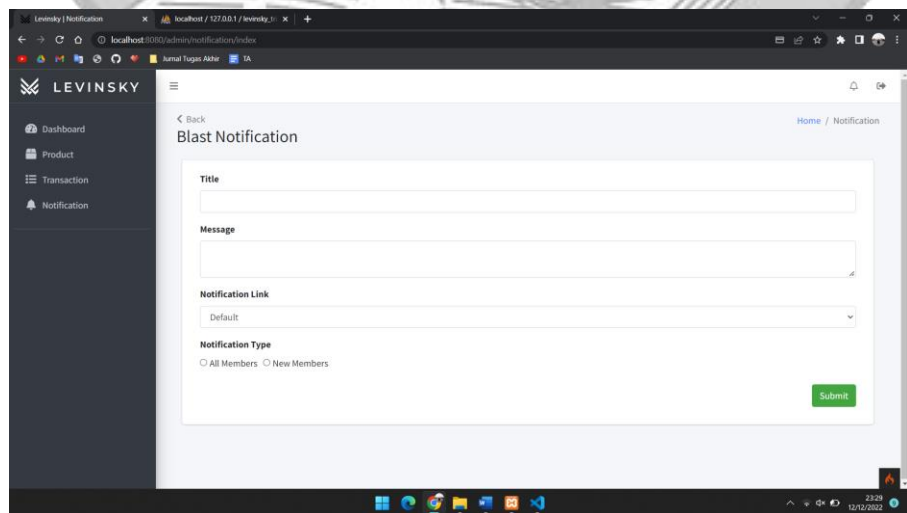
Gambar 4.73 Halaman Detail Transaksi *Admin* Levinsky

Admin dapat melihat detail transaksi setelah memilih transaksi dan akan muncul tampilan seperti pada Gambar 4.73.



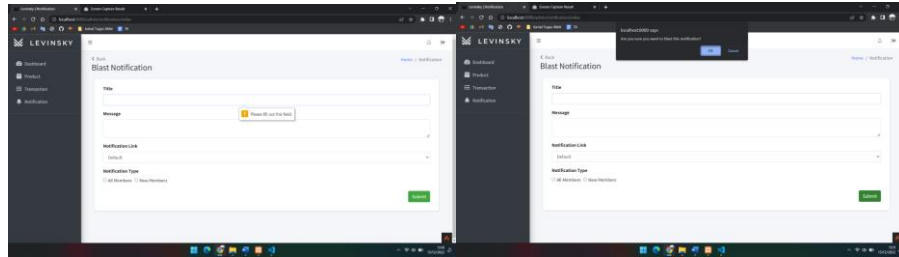
Gambar 4.74 Validasi Penerimaan atau Penolakan Transaksi *Admin* Levinsky

Jika *admin* memilih untuk menerima pesanan atau membatalkan pesanan, maka sistem akan menampilkan peringatan seperti pada Gambar 4.74 untuk memastikan bahwa *admin* ingin menerima atau menolak transaksi.



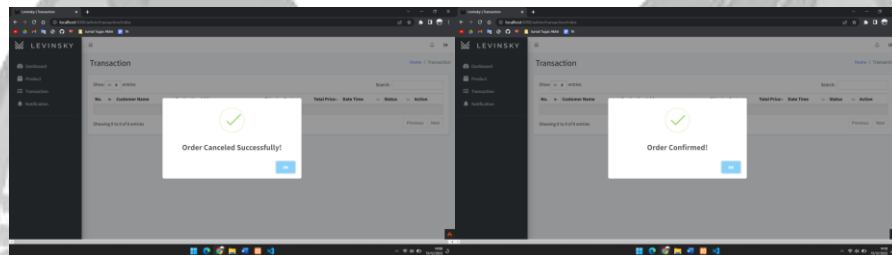
Gambar 4.75 Halaman *Blast Notification* *Admin* Levinsky

Admin dapat melakukan *blast notification* pada *member* baru maupun seluruh *member* dengan memenuhi *form* notifikasi seperti pada Gambar 4.75.

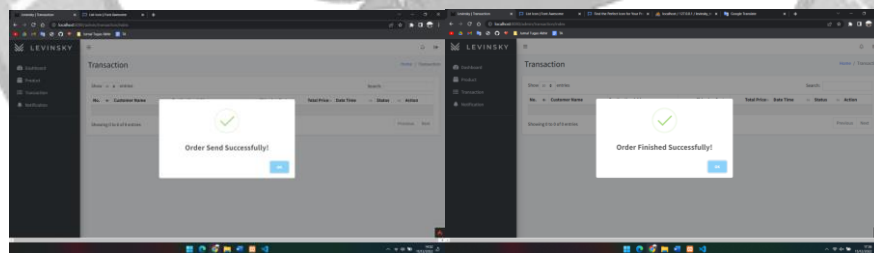


Gambar 4.76 Validasi *Form Blast Notification* Admin Levinsky

Ketika *admin* ingin mengirimkan notifikasi, maka akan muncul peringatan untuk melengkapi *form* dan memastikan bahwa notifikasi yang ingin dikirim sudah benar seperti pada Gambar 4.76.



Gambar 4.77 *Popup Feedback* Transaksi Admin Levinsky



Gambar 4.77 *Popup Feedback* Transaksi Admin Levinsky (lanjutan)

Setiap kali *admin* melakukan konfirmasi pesanan, mengirim pesanan, menyelesaikan pesanan dan membatalkan pesanan maka sistem akan menampilkan *popup feedback* seperti pada Gambar 4.77.

4.3.3 Penggunaan API Pihak Ketiga

Sistem usulan pada Tugas Akhir ini memanfaatkan API pembayaran dan ongkos kirim untuk membantu proses transaksi. API yang digunakan untuk pembayaran adalah

Midtrans sedangkan API yang digunakan untuk menghitung ongkos kirim adalah RajaOngkir. Berikut penjelasan mengenai penggunaan API dalam sistem usulan:

4.3.3.1 API Midtrans

Midtrans merupakan solusi penerimaan pembayaran yang mudah digunakan dan sudah berlisensi dari Bank Indonesia. Midtrans menyediakan *Open API* yang dapat digunakan secara gratis dengan mendaftarkan akun pada *website* Midtrans.

```
public function pay()
{
    $formData = $this->request->getPost();
    // Set your Merchant Server Key
    \Midtrans\Config::$serverKey = 'SB-Mid-server-211Bno0FaoKbNsgqITYUAbw7';
    // Set to Development/Sandbox Environment (default). Set to true for Production Environment (accept real transaction).
    \Midtrans\Config::$isProduction = false;
    // Set sanitization on (default)
    \Midtrans\Config::$isSanitized = true;
    // Set 3DS transaction for credit card to true
    \Midtrans\Config::$is3ds = true;

    $params = array(
        'transaction_details' => array(
            'order_id' => rand(),
            'gross_amount' => $this->request->getPost('subtotal'),
        ),
        'customer_details' => array(
            'first_name' => session()->get('name'),
            'last_name' => '',
            'email' => session()->get('email'),
            'phone' => session()->get('phonenumber'),
        ),
    );

    $data['data'] = [
        'token' => \Midtrans\Snap::getSnapToken($params),
        'form_data' => $formData,
        'name' => session()->get('name'),
        'email' => session()->get('email'),
        'phone' => session()->get('phonenumber'),
    ];

    return view('payment/index', $data);
}
```

Gambar 4.78 Mengirim Data *Cart* Kepada API Midtrans

Penggunaan dari API Midtrans membutuhkan data dari detail transaksi yang dikirimkan dari sistem usulan. Data yang dikumpulkan berupa total harga, nama pelanggan, *email*, dan *phone number* seperti pada Gambar 4.78.

```

<!-- @TODO: replace SET_YOUR_CLIENT_KEY_HERE with your client key -->
<script type="text/javascript"
  src="https://app.sandbox.midtrans.com/snap/snap.js"
  data-client-key="SB-Mid-client-we70k6Hb62D7fQU7">
</script>
<!-- Note: replace with src="https://app.midtrans.com/snap/snap.js" for Production environment -->

<script type="text/javascript">
  // For example trigger on button clicked, or any time you need
  var payButton = document.getElementById('pay-button');
  function payment() {
    // Trigger snap popup. @TODO: Replace TRANSACTION_TOKEN_HERE with your transaction token
    window.snap.pay('<?php echo $data['token']?>', {
      onSuccess: function(result){
        /* You may add your own implementation here */
        alert("payment success!"); console.log(result);
        document.getElementById("myForm").submit();
      },
      onPending: function(result){
        /* You may add your own implementation here */
        alert("wating your payment!"); console.log(result);
      },
      onError: function(result){
        /* You may add your own implementation here */
        alert("payment failed!"); console.log(result);
      },
      onClose: function(){
        /* You may add your own implementation here */
        alert('you closed the popup without finishing the payment');
      }
    })
  };
</script>
<script>
  payment();
</script>

```

Gambar 4.79 Menampilkan *Snap UI* Midtrans untuk Pembayaran

Pengiriman data tersebut dioleh sebagai bentuk *token*, kemudian *request* dari hasil *token* tersebut akan mengembalikan respon berupa tampilan *Snap UI*. Hal ini ditampilkan melalui *script* dan mengambil data *token* dari *request* sebelumnya seperti pada Gambar 4.79.

4.3.3.2 API RajaOngkir

RajaOngkir merupakan API yang dapat membantu perhitungan ongkos kirim secara otomatis. Saat dibuatnya Tugas Akhir ini, RajaOngkir memiliki beberapa tipe akun yang dapat digunakan yaitu diantaranya *Starter*, *Basic* dan *Pro*. Pada Tugas Akhir ini, API RajaOngkir yang digunakan adalah tipe akun *starter* (tidak membayar). Walaupun tipe akun ini merupakan fitur terbatas, namun kebutuhan perhitungan ongkos kirim pada perusahaan Levinsky sebagai UMKM sudah tercukupi.

```

public function getService($destination_id, $weight, $courier)
{
    $curl = curl_init();

    curl_setopt_array($curl, array(
        CURLOPT_URL => "https://api.rajaongkir.com/starter/cost",
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => "",
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 30,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => "POST",
        CURLOPT_POSTFIELDS => "origin=457&destination=" . $destination_id . "&weight=" . $weight . "&courier=" . $courier,
        CURLOPT_HTTPHEADER => array(
            "content-type: application/x-www-form-urlencoded",
            "key: $this->api_key"
        ),
    ));
    $response = curl_exec($curl);
    $err = curl_error($curl);

    curl_close($curl);
}

```

Gambar 4.80 Mengirim *CURL* API RajaOngkir

Untuk menggunakannya sistem dapat menggunakan *curl* untuk mendapatkan harga ongkos kirim, yakni dengan memasukkan *id* kota asal, *id* kota pengiriman, berat dalam gram, dan jenis kurir seperti pada Gambar 4.80. Hasil dari *request* tersebut mengembalikan harga ongkos kirim yang dibutuhkan.

4.4 Tahap Pengujian

Pada tahap ini setiap pengguna dapat menggunakan skrip *test case* untuk melakukan pengujian terhadap sistem usulan untuk menguji kelayakan sistem dan dapat memastikan bahwa sistem memuaskan bagi setiap pengguna.

Contoh skrip *test case* untuk sistem usulan berikut:

SCENARIO #1: LOGIN

Scenario Description:

Skenario ini membahas mengenai proses *user login* ke dalam sistem

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Login User*;

Use Case:

- 1) *Login*;

User groups:

- 1) *User*;

Script #1.1: Login User

Script Description:

Test script ini membahas mengenai proses *user login* ke sistem

Setup:

- 1) *User* mengakses halaman *login*

Script Steps

Tabel 4.12 Tabel *Script #1.1 Login User*

<i>Step #</i>	<i>Test Action</i>	<i>Expected Result</i>	<i>Status</i>	<i>Reference</i>
1	<i>User</i> menekan tombol akun	Sistem menampilkan sidebar account	<i>Pass</i>	Gambar 4.37
2	<i>User</i> memasukkan <i>email</i> dan <i>password</i> lalu menekan tombol “ <i>Log In</i> ”	Sistem menampilkan halaman utama	<i>Pass</i>	Gambar 4.40

Skip *test case* selanjutnya dilampirkan pada Lampiran C (*User Acceptance Testing*).



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Hasil akhir dari pengerjaan Tugas Akhir ini berupa *website* penjualan pakaian yang memanfaatkan API pembayaran Midtrans dan ongkos kirim RajaOngkir. Sistem ini dibuat berdasarkan hasil wawancara terhadap kebutuhan pemilik usaha pakaian Levinsky. Adapun hasil dari wawancara dengan pemilik pada Lampiran A (Hasil Wawancara). Kesimpulan yang didapatkan dari hasil pengerjaan Tugas Akhir ini sebagai berikut:

- 1) Perancangan ulang sistem informasi perusahaan dengan menambahkan validator pembayaran melalui API Midtrans dan perhitungan ongkos kirim yang akurat melalui API RajaOngkir yang mempercepat proses transaksi pada perusahaan Levinsky.
- 2) Sistem usulan berhasil mengimplementasikan API pembayaran dan pengiriman dalam proses transaksi.
- 3) Transaksi yang dilakukan oleh pelanggan disimpan dan ditampilkan sehingga pelanggan dapat melihat status transaksi lebih jelas yang membuat pelanggan dan penjual merasa aman dengan pembayaran yang dilakukan.
- 4) Sistem usulan dapat membantu penjual untuk mengelola produk seperti menyimpan data produk, mengedit produk, dan menghapus produk pada sistem.
- 5) Sistem usulan dapat membantu penjual untuk mengelola transaksi seperti menyimpan data transaksi, menerima transaksi, dan membatalkan transaksi pada sistem.
- 6) Sistem usulan dapat membantu penjual untuk mengirimkan informasi dan berita mengenai perusahaan melalui notifikasi pada *member*.
- 7) Sistem usulan dapat membantu pembeli untuk melihat dan membeli produk.
- 8) Sistem usulan sudah cukup layak untuk digunakan berdasarkan *User Acceptance Testing* (UAT) yang terlampir pada Lampiran III (Dokumentasi Pengujian)

5.2 Saran

Saran yang dapat diusulkan untuk pengembangan berikutnya adalah:

- 1) Memperluas sistem informasi yang ada dengan menambahkan integrasi stok *marketplace* melalui pemanfaatan API *marketplace* ketika perusahaan telah terverifikasi sebagai *official store* pada *marketplace* sehingga saat sistem melakukan perubahan stok, stok pada *marketplace* akan ikut diperbaharui.

- 2) Menambahkan personalisasi seperti tampilan produk favorit berdasarkan data pelanggan dengan memanfaatkan *big data*.
- 3) Menambahkan fitur *fitting* pakaian dengan *artificial reality* (AR) atau *virtual reality* (VR) untuk memudahkan pembeli dalam menentukan ukuran yang sesuai dengan varian produk yang ingin dibeli.



DAFTAR PUSTAKA

- [1] Wuryanti and AG. Eka Wenats, "Digitalisasi Masyarakat: Menilik Kekuatan Dan Kelemahan Dinamika Era Informasi Digital Dan Masyarakat Informasi," *Jurnal ILMU KOMUNIKASI*, vol. 1, no. 2, 2013.
- [2] Erwin Haryono, "DIGITALISASI SISTEM PEMBAYARAN UNTUK KEMANFAATAN MASYARAKAT," 2022.
https://www.bi.go.id/id/publikasi/ruang-media/news-release/Pages/sp_2417922.aspx (accessed Dec. 24, 2022).
- [3] H. Wijoyo, H. Vensuri, and Musnaini, *Digitalisasi UMKM*. 2020.
- [4] A. Dennis, "Systems Analysis & Design An Object-Oriented Approach with UML," 2015.
- [5] "Kemudahan, Pelindungan, dan Pemberdayaan Koperasi dan Usaha Mikro, Kecil, dan Menengah." <https://peraturan.bpk.go.id/Home/Details/161837/pp-no-7-tahun-2021#:~:text=PP%20No.%207%20Tahun%202021,dan%20Menengah%20%5BJDIH%20BPK%20RI%5D&text=LN.2021%2FNo.17,go.id%20%3A%2092%20hlm.> (accessed Dec. 24, 2022).
- [6] J. C. Jackson, *Web Technologies A Computer Science Perspective*. Pearson Education, Inc. 2007.
- [7] L. Welling and Laura. Thomson, *PHP and MySQL Web Development*. Sams, 2001.
- [8] Adani M R, "Pengenaln Big Data: Pengertian Fungsi, Manfaat, dan Tools," *sekawanmedia.co.id*, 2020.
- [9] D. Upton, *CodeIgniter for Rapid PHP Application Development*. 2007.
- [10] "Install Windows Server 2008 and Windows Server 2008 R2 - Hardware Requirements." <https://learn.microsoft.com/en-us/iis/install/installing-iis-7/install-windows-server-2008-and-windows-server-2008-r2> (accessed Dec. 24, 2022).
- [11] "PHP Install Requirements." <https://www.php.net/manual/en/install.windows.requirements.php> (accessed Dec. 24, 2022).
- [12] "New XAMPP release 7.4.30." https://www.apachefriends.org/blog/new_xampp_20221001.html (accessed Dec. 24, 2022).

[13] “CodeIgniter *Server Requirements*.”

https://codeigniter.com/user_guide/intro/requirements.html (accessed Dec. 24, 2022).



LAMPIRAN A

HASIL WAWANCARA

P: Apa kesulitan yang sedang dialami perusahaan Levinsky Indonesia selama beroperasi?

N: Untuk kesulitan sebenarnya cukup banyak, saat ini kami sedang kesulitan saat mengecek transaksi yang diterima di luar *e-commerce*, karena masih banyak juga pelanggan kita yang tidak beli produk kami di *e-commerce* dan mereka masih banyak yang mau membeli itu via WhatsApp saja atau secara langsung datang ke tempat untuk beli, jadi sebenarnya terkadang kita juga kan harus ngecekin dulu apakah transferannya sudah masuk atau belum, lalu belakangan ini kan juga banyak transaksi bodong ya yang memanfaatkan *seller seller* kaya kita ini untuk melakukan transaksi palsu lalu kita diminta *refund* uang. Jadi menurut kami untuk saat ini keamanan dari transaksi itu cukup krusial juga.

P: Bagaimana sistem usulan dapat membantu mempermudah proses bisnis perusahaan Levinsky Indonesia?

N: Mungkin dengan membuat sistem sendiri semisal nya *website* atau aplikasi yang aman, mudah, dan cepat jadi kita bisa memproses pesanan tuh jadi lebih cepat dan lebih percaya diri aja dengan pelanggan karena sistem yang aman gitu. Lebih baik lagi kalau sistemnya bisa mempermudah kelangsungan bisnis nya ya seperti penyimpanan transaksi, *input* stok produk, menurut saya itu berguna sekali karena kita sekarang ini masih pake catatan manual. Karena manual ini juga kita sering bingung kalau produk nya kok sisa segini padahal kemarin catat harusnya masih ada segini gitu. Nah terus kita juga biasa bingung nih gimana cara kita *channeling* ke pelanggan pelanggan kita yang udah udah untuk menarik mereka kembali belanja di Levinsky. Mungkin dari saya itu dulu saja udah cukup.

P: Baik, terima kasih banyak atas waktunya akan saya usahakan untuk membuat sistem yang layak untuk digunakan.

LAMPIRAN B

USE CASE DESCRIPTION

1 Create Account

<i>Use Case Name: Create Account</i>	ID: 1	<i>Importance Level: High</i>
<i>Primary Actor: User</i>	<i>Use Case Type: Detailed, Essential</i>	
<i>Stakeholder and Interest:</i> <i>User : untuk mendaftarkan pada sistem sebagai Member</i>		
<i>Brief Description:</i> <i>Use case ini membahas mengenai cara pembuatan akun pada website Levinsky</i>		
<i>Trigger: User mengakses formulir register</i> <i>Type: External</i>		
<i>Relationship:</i> <ul style="list-style-type: none">• <i>Association: User</i>• <i>Include:</i>• <i>Extend:</i>• <i>Generalization:</i>		
<i>Normal Flow of Events:</i> <ol style="list-style-type: none">1. <i>User memasukkan data formulir registrasi berupa username, email, dan password</i>2. <i>Sistem akan memvalidasi kecocokan email pada database jika email tersebut telah terdaftar.</i><ul style="list-style-type: none">➤ <i>Jika sistem berhasil memvalidasi</i> S-1: Registrasi sebagai pengguna baru➤ <i>Jika sistem gagal memvalidasi</i> S-2: Sistem akan membatalkan registrasi dan menampilkan pesan gagal		
<i>SubFlows:</i> S-1: Registrasi data sebagai pengguna baru S-2: Sistem akan membatalkan registrasi dan menampilkan pesan gagal		
<i>Alternate/Exception flows:</i>		

2 Login

Use Case Name: Login	ID: 2	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholder and Interest: User : untuk mendapatkan akses pada sistem sebagai Member		
Brief Description: Use case ini membahas mengenai cara login sebagai member pada website Levinsky		
Trigger: User mengakses formulir login Type: External		
Relationship: <ul style="list-style-type: none">• Association: User• Include:• Extend:• Generalization:		
Normal Flow of Events: <ol style="list-style-type: none">1. User memasukkan data form login berupa email, dan password2. Sistem akan memvalidasi kecocokan email dan password pada database.<ul style="list-style-type: none">➤ Jika berhasil memvalidasi S-1: User berhasil login sebagai member.➤ Jika gagal memvalidasi S-2: User gagal login dan sistem akan menampilkan pesan gagal login.		
SubFlows: S-1: User berhasil login sebagai member. S-2: User gagal login dan sistem akan menampilkan pesan gagal login.		
Alternate/Exception flows:		

3 Forgot Password

<i>Use Case Name: Forgot Password</i>	ID: 3	<i>Importance Level: High</i>
<i>Primary Actor: User</i>	<i>Use Case Type: Detailed, Essential</i>	
<i>Stakeholder and Interest:</i> <i>User : untuk mengubah password yang terlupakan oleh user</i>		
<i>Brief Description:</i> <i>Use case ini membahas mengenai cara mengganti password pada website Levinsky</i>		
<i>Trigger: User mengakses forget password</i> <i>Type: External</i>		
<i>Relationship:</i> <ul style="list-style-type: none">• <i>Association: User</i>• <i>Include:</i>• <i>Extend:</i>• <i>Generalization:</i>		
<i>Normal Flow of Events:</i> <ol style="list-style-type: none">1. <i>User dapat mengakses halaman forget password</i>2. <i>User dapat memasukkan email akun pada formulir forget password</i>3. <i>Sistem akan memverifikasi email pada database</i><ul style="list-style-type: none">➤ <i>Jika sistem berhasil memverifikasi</i> S-1: <i>Sistem akan mengirimkan link perubahan password melalui email</i>➤ <i>Jika sistem gagal memverifikasi</i> S-2: <i>Sistem akan menampilkan pesan gagal</i>		
<i>SubFlows:</i> S-1: <i>Sistem akan mengirimkan link perubahan password melalui email</i> S-2: <i>Sistem akan menampilkan pesan gagal</i>		
<i>Alternate/Exception flows:</i>		

4 Cart

<i>Use Case Name: Cart</i>	ID: 4	<i>Importance Level: High</i>
<i>Primary Actor: Member</i>	<i>Use Case Type: Detailed, Essential</i>	
<i>Stakeholder and Interest:</i>		
<i>Member:</i> untuk memasukkan daftar belanjaan produk pada <i>cart</i>		
<i>Brief Description:</i>		
<i>Use case</i> ini membahas mengenai cara untuk memasukkan daftar belanjaan produk pada <i>cart website</i> Levinsky		
<i>Trigger: Member</i> mengakses halaman <i>cart</i>		
<i>Type: External</i>		
<i>Relationship:</i>		
<ul style="list-style-type: none">• <i>Association: Member</i>• <i>Include:</i>• <i>Extend:</i>• <i>Generalization:</i>		
<i>Normal Flow of Events:</i>		
<ol style="list-style-type: none">1. Sistem akan melakukan pengecekan ketersediaan produk setiap <i>member</i> membuka halaman <i>cart</i>2. <i>Member</i> dapat memilih varian produk yang ingin di beli3. <i>Member</i> memilih ukuran dan memasukkan jumlah varian produk4. Sistem akan memeriksa ketersediaan produk yang diinginkan<ul style="list-style-type: none">➤ Jika tersedia S-1: Sistem akan menambahkan produk ke <i>cart</i>➤ Jika tidak tersedia S-2: Sistem tidak menambahkan produk ke <i>cart</i>		
<i>SubFlows:</i>		
S-1: Sistem akan menambahkan produk ke <i>cart</i>		
S-2: Sistem tidak menambahkan produk ke <i>cart</i>		
<i>Alternate/Exception flows:</i>		

5 Manage Address

Use Case Name: Manage Address	ID: 5	Importance Level: High
Primary Actor: Member	Use Case Type: Detailed, Essential	
Stakeholder and Interest:		
Member: untuk mendaftarkan alamat pengiriman pada website Levinsky		
Brief Description:		
Use case ini membahas mengenai cara untuk mengelola daftar alamat pengiriman member pada website Levinsky		
Trigger: Member mengakses halaman alamat		
Type: External		
Relationship:		
<ul style="list-style-type: none">• Association: Member• Include:• Extend:• Generalization:		
Normal Flow of Events:		
<div>1. Member dapat memilih untuk menambahkan alamat baru, menghapus alamat, atau menyunting alamat.</div> <div><div>➤ Jika member memilih untuk menambahkan alamat baru</div><div>S-1: Member dapat mengisi alamat, memilih provinsi dan kabupaten yang ditampilkan dari API RajaOngkir (pihak ketiga)</div><div>➤ Jika member memilih untuk menghapus alamat</div><div>S-2: Sistem akan menghapus alamat dari database.</div><div>➤ Jika member memilih untuk menyunting alamat</div><div>S-3: Member dapat mengubah informasi alamat dan sistem akan menyimpan perubahan pada database.</div></div>		
SubFlows:		
<div>S-1: Member dapat mengisi alamat, memilih provinsi dan kabupaten yang ditampilkan dari API RajaOngkir (pihak ketiga)</div> <div>S-2: Sistem akan menghapus alamat dari database</div> <div>S-3: Member dapat mengubah informasi alamat dan sistem akan menyimpan perubahan pada database.</div>		
Alternate/Exception flows:		

6 Receive Notification

Use Case Name: <i>Receive Notification</i>	ID: 6	Importance Level: <i>High</i>
Primary Actor: <i>Member</i>	Use Case Type: <i>Detailed, Essential</i>	
Stakeholder and Interest: <i>Member</i> : untuk menerima notifikasi pada <i>website</i> Levinsky		
Brief Description: <i>Use case</i> ini membahas mengenai cara <i>member</i> menerima notifikasi pada <i>website</i> Levinsky		
Trigger: - Type: <i>External</i>		
Relationship: <ul style="list-style-type: none">• Association: <i>Member</i>• Include:• Extend:• Generalization:		
Normal Flow of Events: <ol style="list-style-type: none">1. Sistem atau <i>admin</i> mengirimkan notifikasi kepada pengguna2. <i>Member</i> akan menerima notifikasi3. <i>Member</i> dapat melihat notifikasi “<i>user_notified</i>” yang berstatus 1 (belum dibaca) dan menekan notifikasi untuk langsung mengalihkan ke halaman yang dituju4. Sistem akan mengubah status notifikasi “<i>user_notified</i>” menjadi 0 yang tandanya sudah dibaca		
SubFlows:		
Alternate/Exception flows:		

7 Checkout

Use Case Name: Checkout	ID: 7	Importance Level: High
Primary Actor: Member	Use Case Type: Detailed, Essential	
Stakeholder and Interest:		
Member: untuk melakukan checkout pada website Levinsky		
Brief Description:		
Use case ini membahas mengenai cara untuk melakukan checkout pada website Levinsky		
Trigger: Member mengakses halaman checkout		
Type: External		
Relationship:		
<ul style="list-style-type: none">• Association: Member• Include:• Extend:• Generalization:		
Normal Flow of Events:		
<ol style="list-style-type: none">1. Member telah selesai memasukkan produk ke cart2. Member dapat melakukan checkout3. Pada tahapan checkout, sistem akan menghitung ongkos kirim berdasarkan kabupaten/kota asal yang telah ditentukan sejak awal pada sistem menuju kabupaten/kota pengiriman melalui API RajaOngkir (pihak ketiga). Selanjutnya sistem akan mengkalkulasikan total dari biaya yang harus ditanggung oleh pembeli dan kemudian mengirim data berupa total biaya yang harus dibayar, alamat email member, dan nomor telepon member kepada API Midtrans untuk menyelesaikan pembayaran.<ul style="list-style-type: none">➤ Jika member belum memiliki alamat S-1: Member harus membuat alamat baru yang dapat didaftarkan dengan mengisi formulir alamat berupa alamat, provinsi, dan kabupaten. Setelah itu alamat secara default akan terpilih sebagai alamat utama➤ Jika member sudah memiliki alamat S-2: Sistem dapat melakukan checkout		
SubFlows:		
<p>S-1: Member harus membuat alamat baru yang dapat didaftarkan dengan mengisi formulir alamat berupa alamat, provinsi, dan kabupaten. Setelah itu alamat secara default akan terpilih sebagai alamat utama</p> <p>S-2: Sistem dapat melakukan checkout</p>		
Alternate/Exception flows:		

8 Manage Product

Use Case Name: Manage Product	ID: 8	Importance Level: High
Primary Actor: Admin	Use Case Type: Detailed, Essential	
Stakeholder and Interest: Admin: untuk mengelola produk pada website Levinsky		
Brief Description: Use case ini membahas mengenai cara untuk mengelola produk pada website Levinsky		
Trigger: Admin mengakses halaman produk admin Levinsky Type: External		
Relationship: <ul style="list-style-type: none">• Association: Admin• Include:• Extend:• Generalization:		
Normal Flow of Events: <ol style="list-style-type: none">Admin dapat melakukan pengelolaan produk dengan mengganti status produk, mengedit informasi produk, dan menghapus produk.<ul style="list-style-type: none">Jika admin memilih untuk mengubah status produk S-1: Sistem akan memperbaharui status produk pada databaseJika admin memilih untuk mengedit informasi produk S-2: Setelah admin mengubah informasi produk maka sistem akan memperbaharui detail produk pada databaseJika admin memilih untuk menghapus produk S-3: Sistem akan menonaktifkan produk pada database		
SubFlows: S-1: Sistem akan memperbaharui status produk pada database S-2: Setelah admin mengubah informasi produk maka sistem akan memperbaharui detail produk pada database S-3: Sistem akan menonaktifkan produk pada database		
Alternate/Exception flows:		

9 Manage Transaction

Use Case Name: Manage Transaction	ID: 9	Importance Level: High
Primary Actor: Admin	Use Case Type: Detailed, Essential	
Stakeholder and Interest:		
Admin: untuk mengelola transaksi pada website Levinsky		
Brief Description:		
Use case ini membahas mengenai cara untuk mengelola transaksi pada website Levinsky		
Trigger: Admin mengakses halaman transaksi admin Levinsky		
Type: External		
Relationship:		
<ul style="list-style-type: none">• Association: Admin• Include:• Extend:• Generalization:		
Normal Flow of Events:		
<ol style="list-style-type: none">1. Transaksi yang baru diterima oleh admin secara default akan berstatus “ON PROGRESS”2. Admin dapat memilih untuk melihat detail transaksi, menerima transaksi atau menolak transaksi<ul style="list-style-type: none">➤ Jika admin memilih untuk melihat detail transaksi S-1: Sistem akan menampilkan detail transaksi beserta detail pesanan yang dilakukan oleh pengguna member➤ Jika admin memilih untuk menerima transaksi S-2: Sistem akan memperbarui status transaksi menjadi “CONFIRMED”➤ Jika admin memilih untuk menolak transaksi S-3: Sistem akan memperbarui status transaksi menjadi “CANCELED”		
SubFlows:		
<p>S-1: Sistem akan menampilkan detail transaksi beserta detail pesanan yang dilakukan oleh pengguna member</p> <p>S-2: Sistem akan memperbarui status transaksi menjadi “CONFIRMED”</p> <p>S-3: Sistem akan memperbarui status transaksi menjadi “CANCELED”</p>		
Alternate/Exception flows:		

10 Send Notification

Use Case Name: Send Notification	ID: 10	Importance Level: High
Primary Actor: Admin	Use Case Type: Detailed, Essential	
Stakeholder and Interest:		
Admin: untuk mengirim notifikasi pada website Levinsky		
Brief Description:		
Use case ini membahas mengenai cara untuk mengirim notifikasi pada website Levinsky		
Trigger: Admin mengakses halaman notifikasi admin Levinsky		
Type: External		
Relationship:		
<ul style="list-style-type: none">Association: AdminInclude:Extend:Generalization:		
Normal Flow of Events:		
<ol style="list-style-type: none">Admin memilih fungsi untuk mengirim notifikasiAdmin mengisi formulir berupa title, message, link (mengalihkan), dan pilihan pengguna yang menerima notifikasiSistem akan memvalidasi kelengkapan formulir notifikasi<ul style="list-style-type: none">Jika formulir tidak lengkap S-1: Sistem akan memunculkan error message pada kolom yang belum terisiJika formulir sudah lengkap S-2: Sistem akan menyimpan data notifikasi dan mengirimkan notifikasi kepada pengguna member yang dituju		
SubFlows:		
<p>S-1: Sistem akan memunculkan error message pada kolom yang belum terisi</p> <p>S-2: Sistem akan menyimpan data notifikasi dan mengirimkan notifikasi kepada pengguna member yang dituju</p>		
Alternate/Exception flows:		

LAMPIRAN C

USER ACCEPTANCE TESTING

SCENARIO #1: LOGIN

Scenario Description:

Skenario ini membahas mengenai proses *user login* kedalam sistem

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Login User*;

Use Case:

- 1) *Login*;

User groups:

- 1) *User*;

Script #1.1: Login User

Script Description:

Test script ini membahas mengenai proses *user login* ke sistem

Setup:

- 1) *User* mengakses halaman *login*

Script Steps

Script #1.1 Login User

Step #	Test Action	Expected Result	Status	Reference
1	<i>User</i> menekan tombol akun	Sistem menampilkan <i>sidebar account</i>	<i>Pass</i>	Gambar 4.36
2	<i>User</i> memasukkan <i>email</i> dan <i>password</i> lalu menekan tombol “ <i>Log In</i> ”	Sistem menampilkan halaman utama	<i>Pass</i>	Gambar 4.37

SCENARIO #2: REGISTER

Scenario Description:

Skenario ini membahas mengenai proses *user* melakukan registrasi pada sistem

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Register User*;

Use Case:

- 1) *Register*;

User groups:

- 1) *User*;

Script #2.1: Register User

Script Description:

Test script ini membahas mengenai proses *user* melakukan registrasi pada sistem

Setup:

- 1) *User* mengakses halaman *register*

Script Steps

Script #2.1 Register User

Step #	Test Action	Expected Result	Status	Reference
1	<i>User</i> menekan tombol akun	Sistem menampilkan <i>sidebar account</i>	<i>Pass</i>	Gambar 4.37
2	<i>User</i> menekan tombol “ <i>Register</i> ”	Sistem menampilkan <i>form register</i>	<i>Pass</i>	Gambar 4.38
3	<i>User</i> memasukkan <i>username</i> , <i>email</i> , <i>phone number</i> , <i>password</i> ,dan <i>confirm password</i> lalu menekan tombol “ <i>Register</i> ”	Sistem mengirimkan <i>email</i> konfirmasi	<i>Pass</i>	Gambar 4.38
4	<i>User</i> menekan tombol “ <i>Activate Your Account</i> ” pada <i>email</i> konfirmasi yang diterima	Sistem menampilkan halaman utama	<i>Pass</i>	Gambar 4.39

SCENARIO #3: FORGET PASSWORD

Scenario Description:

Skenario ini membahas mengenai proses *user* melakukan *forget password* dan mengganti *password* baru.

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Forget Password User*;

Use Case:

- 1) *Forget Password*;

User groups:

- 1) *User*;

Script #3.1: Forget Password User

Script Description:

Test script ini membahas mengenai proses *user* melakukan *forget password* dan mengganti *password* baru.

Setup:

- 1) *User* mengakses halaman *forget password*

Script Steps

Script #3.1 Forget Password User

<i>Step #</i>	<i>Test Action</i>	<i>Expected Result</i>	<i>Status</i>	<i>Reference</i>
1	<i>User</i> menekan tombol akun	Sistem menampilkan <i>sidebar account</i>	<i>Pass</i>	Gambar 4.37
2	<i>User</i> menekan tombol “ <i>Forget Password?</i> ”	Sistem menampilkan <i>form reset password</i>	<i>Pass</i>	Gambar 4.46
3	<i>User</i> memasukkan <i>email</i> lalu menekan tombol “ <i>Reset Password</i> ”	Sistem mengirimkan <i>email reset password</i>	<i>Pass</i>	Gambar 4.46
4	<i>User</i> menekan tombol “ <i>Reset Your Password</i> ” pada <i>email reset password</i> yang diterima	Sistem menampilkan halaman <i>recover password</i>	<i>Pass</i>	Gambar 4.46
5	<i>User</i> memasukan <i>email</i> yang terdaftar, <i>new password</i> dan <i>confirm password</i>	Sistem menampilkan halaman <i>login</i>	<i>Pass</i>	Gambar 4.46

SCENARIO #4: MANAGE CART

Scenario Description:

Skenario ini membahas mengenai proses *member* mengelola *cart*

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Add Cart*;
- 2) *Delete Cart*;
- 3) *Edit Cart*;

Use Case:

- 1) *Manage Cart*;

User groups:

- 1) *Member*;

Script #4.1: Add Cart

Script Description:

Test script ini membahas mengenai proses *member* memasukkan produk belanja ke *cart*.

Setup:

- 1) *Member* mengakses halaman *shop*
- 2) *Member* memasukkan produk ke dalam *cart*

Script Steps

Script #4.1 Add Cart

<i>Step #</i>	<i>Test Action</i>	<i>Expected Result</i>	<i>Status</i>	<i>Reference</i>
1	<i>Member</i> menekan tombol <i>menu</i>	Sistem menampilkan <i>sidebar menu</i>	<i>Pass</i>	Gambar 4.41
2	<i>Member</i> menekan tombol " <i>Shop</i> "	Sistem menampilkan halaman <i>shop</i>	<i>Pass</i>	Gambar 4.50
3	<i>Member</i> memilih jenis produk yang diinginkan	Sistem menampilkan halaman <i>view shop</i>	<i>Pass</i>	Gambar 4.50
4	<i>Member</i> memasukkan varian dan jumlah yang diinginkan kemudian menekan tombol " <i>Add To Cart</i> "	Sistem menampilkan halaman <i>shop</i> dan menampilkan <i>popup "Cart Added Successful!"</i>	<i>Pass</i>	Gambar 4.53

Script #4.2: Delete Cart

Script Description:

Test script ini membahas mengenai proses *member* menghapus produk pada *cart*.

Setup:

- 1) *Member* mengakses halaman *check out*
- 2) *Member* menghapus produk dari *cart*

Script Steps

Script #4.2 Delete Cart

Step #	Test Action	Expected Result	Status	Reference
1	<i>Member</i> menekan tombol <i>cart</i>	Sistem menampilkan <i>sidebar cart</i>	<i>Pass</i>	Gambar 4.54
2	<i>Member</i> menekan tombol “ <i>Check Out</i> ”	Sistem menampilkan halaman <i>check out</i>	<i>Pass</i>	Gambar 4.55
3	<i>Member</i> menekan tombol <i>delete cart</i>	Sistem menampilkan <i>popup “Cart Deleted”</i>	<i>Pass</i>	Gambar 4.55

Script #4.3: Edit Cart

Script Description:

Test script ini membahas mengenai proses *member* mengubah jumlah varian produk pada *cart*.

Setup:

- 1) *Member* mengakses halaman *check out*
- 2) *Member* mengubah jumlah varian produk pada *cart*

Script Steps

Script #4.3 Edit Cart

Step #	Test Action	Expected Result	Status	Reference
1	<i>Member</i> menekan tombol <i>cart</i>	Sistem menampilkan <i>sidebar cart</i>	<i>Pass</i>	Gambar 4.54
2	<i>Member</i> menekan tombol “ <i>Check Out</i> ”	Sistem menampilkan halaman <i>check out</i>	<i>Pass</i>	Gambar 4.55
3	<i>Member</i> menekan tombol “-” atau “+” atau	Sistem mengubah jumlah varian produk pada <i>cart</i>	<i>Pass</i>	Gambar 4.55

SCENARIO #5: MANAGE ADDRESS

Scenario Description:

Skenario ini membahas mengenai proses *member* mengelola alamat.

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Add Address*;
- 2) *Delete Address*;
- 3) *Edit Address*;

Use Case:

- 1) *Manage Address*;

User groups:

- 1) *Member*;

Script #5.1: Add Address

Script Description:

Test script ini membahas mengenai proses *member* menambahkan alamat baru.

Setup:

- 1) *Member* mengakses halaman *address*
- 2) *Member* menambahkan alamat baru

Script Steps

Script #5.1 Add Address

Step #	Test Action	Expected Result	Status	Reference
1	<i>Member</i> menekan tombol <i>menu</i>	Sistem menampilkan <i>sidebar menu</i>	<i>Pass</i>	Gambar 4.41
2	<i>Member</i> menekan tombol “ <i>View Address</i> ”	Sistem menampilkan halaman <i>address</i>	<i>Pass</i>	Gambar 4.47
3	<i>Member</i> menekan tombol “ <i>Add New Address</i> ”	Sistem menampilkan halaman <i>create address</i>	<i>Pass</i>	Gambar 4.48
4	<i>Member</i> memasukkan <i>name</i> , <i>phone number</i> dan memilih <i>province</i> , <i>city</i> , dan memasukkan detail alamat lalu menekan tombol “ <i>Save</i> ”	Sistem menampilkan halaman <i>shop</i> dan menampilkan popup “ <i>Address Created</i> ”	<i>Pass</i>	Gambar 4.50

Script #5.2: Delete Address

Script Description:

Test script ini membahas mengenai proses *member* menghapus alamat.

Setup:

- 1) *Member* mengakses halaman *address*

- 2) *Member* menghapus alamat

Script Steps

Script #5.2 Delete Address

Step #	Test Action	Expected Result	Status	Reference
1	<i>Member</i> menekan tombol <i>menu</i>	Sistem menampilkan <i>sidebar menu</i>	<i>Pass</i>	Gambar 4.41
2	<i>Member</i> menekan tombol “ <i>View Address</i> ”	Sistem menampilkan halaman <i>address</i>	<i>Pass</i>	Gambar 4.47
3	<i>Member</i> menekan tombol hapus	Sistem menampilkan popup “ <i>Address Deleted</i> ”	<i>Pass</i>	Gambar 4.47

Script #5.3: Edit Address

Script Description:

Test script ini membahas mengenai proses *member* mengedit alamat *primary address*.

Setup:

- 1) *Member* mengakses halaman *address*
- 2) *Member* memilih alamat sebagai *primary address*

Script Steps

Script #5.3 Add Address

Step #	Test Action	Expected Result	Status	Reference
1	<i>Member</i> menekan tombol <i>menu</i>	Sistem menampilkan <i>sidebar menu</i>	<i>Pass</i>	Gambar 4.41
2	<i>Member</i> menekan tombol “ <i>View Address</i> ”	Sistem menampilkan halaman <i>address</i>	<i>Pass</i>	Gambar 4.47
3	<i>Member</i> memilih salah satu alamat yang ada	Sistem menampilkan popup “ <i>Primary Address Changed</i> ”	<i>Pass</i>	Gambar 4.47

SCENARIO #6: RECEIVE NOTIFICATION

Scenario Description:

Skenario ini membahas mengenai proses *member* menerima notifikasi.

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Receive Notification*;

Use Case:

- 1) *Receive Notification*;

User groups:

- 1) *Member*;

Script #6.1: Receive Notification

Script Description:

Test script ini membahas mengenai proses *member* menerima notifikasi.

Setup:

- 1) *Member* mengakses halaman notifikasi

Script Steps

Script #6.1 Receive Notification

Step #	Test Action	Expected Result	Status	Reference
1	<i>Member</i> menekan tombol <i>notification</i>	Sistem menampilkan <i>sidebar notification</i>	<i>Pass</i>	Gambar 4.56
2	<i>Member</i> menekan tombol “ <i>See All Notification</i> ”	Sistem menampilkan halaman <i>notification</i>	<i>Pass</i>	Gambar 4.62

SCENARIO #7: CHECKOUT ORDER

Scenario Description:

Skenario ini membahas mengenai proses *member* melakukan *checkout*.

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Checkout Order*;

Use Case:

- 1) *Checkout Order*;

User groups:

- 1) *Member*;

Script #7.1: Checkout Order

Script Description:

Test script ini membahas mengenai proses *member* melakukan *checkout*

Setup:

- 1) *Member* mengakses halaman *checkout*

Script Steps

Script #7.1 Checkout Order

<i>Step #</i>	<i>Test Action</i>	<i>Expected Result</i>	<i>Status</i>	<i>Reference</i>
1	<i>Member</i> menekan tombol “ <i>Check Out</i> ”	Sistem menampilkan <i>halaman check out</i>	<i>Pass</i>	Gambar 4.55
2	<i>Member</i> menekan tombol “ <i>Check Out</i> ” pada halaman <i>check out</i>	Sistem menampilkan pembayaran Midtrans	<i>Pass</i>	Gambar 4.57
3	<i>Member</i> menyelesaikan pembayaran	Sistem menampilkan <i>popup “Transaction Successful!”</i>	<i>Pass</i>	Gambar 4.60

SCENARIO #8: MANAGE PRODUCT

Scenario Description:

Skenario ini membahas mengenai proses *admin* mengelola produk.

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Change Status Product*;
- 2) *Edit Product*;
- 3) *Delete Product*;

Use Case:

- 1) *Manage Product*;

User groups:

- 1) *Admin*;

Script #8.1: Change Status Product

Script Description:

Test script ini membahas mengenai proses *admin* mengubah status keaktifan produk

Setup:

- 1) *Admin* mengakses halaman *product*
- 2) *Admin* mengubah status produk

Script Steps

Script #8.1 Change Status Product

Step #	Test Action	Expected Result	Status	Reference
1	Admin menekan tombol "Product"	Sistem menampilkan halaman <i>product</i>	Pass	Gambar 4.66
2	Admin menekan tombol "Active/Non Active" pada halaman <i>product</i>	Sistem menampilkan konfirmasi pengubahan status dan menampilkan <i>popup</i> "Update Product Successful!"	Pass	Gambar 4.67

Script #8.2: Edit Product

Script Description:

Test script ini membahas mengenai proses *admin* mengubah informasi detail produk

Setup:

- 1) Admin mengakses halaman *product*
- 2) Admin mengubah mengedit produk

Script Steps

Script #8.2 Edit Product

Step #	Test Action	Expected Result	Status	Reference
1	Admin menekan tombol "Product"	Sistem menampilkan halaman <i>product</i>	Pass	Gambar 4.66
2	Admin menekan tombol <i>edit</i> pada halaman <i>product</i>	Sistem menampilkan halaman <i>edit</i> produk	Pass	Gambar 4.70
3	Admin mengubah data detail produk dan menekan tombol "Save"	Sistem menampilkan halaman <i>product</i> dan menampilkan <i>popup</i> "Update Product Successful!"	Pass	Gambar 4.70

Script #8.3: Delete Product

Script Description:

Test script ini membahas mengenai proses *admin* menghapus produk

Setup:

- 1) Admin mengakses halaman *product*
- 2) Admin menghapus produk

Script Steps

Script #8.3 Delete Product

Step #	Test Action	Expected Result	Status	Reference
--------	-------------	-----------------	--------	-----------

1	<i>Admin</i> menekan tombol “ <i>Product</i> ”	Sistem menampilkan halaman <i>product</i>	<i>Pass</i>	Gambar 4.66
2	<i>Admin</i> menekan tombol hapus pada halaman <i>product</i>	Sistem menampilkan halaman <i>product</i> dan menampilkan <i>popup</i> “ <i>Delete Product Successful!</i> ”	<i>Pass</i>	Gambar 4.71

SCENARIO #9: MANAGE TRANSACTION

Scenario Description:

Skenario ini membahas mengenai proses *admin* mengelola transaksi.

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *View Transaction*;
- 2) *Accept Transaction*;
- 3) *Decline Transaction*;

Use Case:

- 1) *Manage Transaction*;

User groups:

- 1) *Admin*;

Script #9.1: View Transaction

Script Description:

Test script ini membahas mengenai proses *admin* melihat detail transaksi tertentu

Setup:

- 1) *Admin* mengakses halaman *transaction*
- 2) *Admin* melihat detail transaksi

Script Steps

Script #9.1 View Transaction

Step #	Test Action	Expected Result	Status	Reference
1	<i>Admin</i> menekan tombol “ <i>Transaction</i> ”	Sistem menampilkan halaman <i>transaction</i>	<i>Pass</i>	Gambar 4.72
2	<i>Admin</i> menekan tombol <i>view</i> pada halaman <i>transaction</i>	Sistem menampilkan halaman <i>view detail transaction</i>	<i>Pass</i>	Gambar 4.73

Script #9.2: Accept Transaction

Script Description:

Test script ini membahas mengenai proses admin menerima transaction

Setup:

- 1) *Admin mengakses halaman transaction*
- 2) *Admin menerima transaction*

Script Steps

Script #9.2 Accept Transaction				
<i>Step #</i>	<i>Test Action</i>	<i>Expected Result</i>	<i>Status</i>	<i>Reference</i>
1	<i>Admin menekan tombol "Transaction"</i>	<i>Sistem menampilkan halaman transaction</i>	<i>Pass</i>	<i>Gambar 4.72</i>
2	<i>Admin menekan tombol accept pada halaman transaction</i>	<i>Sistem menampilkan popup "Order Confirmed!"</i>	<i>Pass</i>	<i>Gambar 4.77</i>

Script #9.3: Decline Transaction

Script Description:

Test script ini membahas mengenai proses admin membatalkan transaction

Setup:

- 1) *Admin mengakses halaman transaction*
- 2) *Admin membatalkan transaction*

Script Steps

Script #9.3 Decline Transaction				
<i>Step #</i>	<i>Test Action</i>	<i>Expected Result</i>	<i>Status</i>	<i>Reference</i>
1	<i>Admin menekan tombol "Transaction"</i>	<i>Sistem menampilkan halaman transaction</i>	<i>Pass</i>	<i>Gambar 4.72</i>
2	<i>Admin menekan tombol decline pada halaman transaction</i>	<i>Sistem menampilkan popup "Order Canceled Successfully!"</i>	<i>Pass</i>	<i>Gambar 4.77</i>

SCENARIO #10: SEND NOTIFICATION

Scenario Description:

Skenario ini membahas mengenai proses admin mengirimkan notifikasi.

Test Scripts:

Skenario ini terdiri dari *test script* berikut:

- 1) *Send Notification*;

Use Case:

- 1) *Send Notification*;

User groups:

- 1) *Admin*;

Script #10.1: Send Notification

Script Description:

Test script ini membahas mengenai proses *admin* melihat detail transaksi tertentu

Setup:

- 1) *Admin* mengakses halaman *notification*
- 2) *Admin* mengirim notifikasi

Script Steps

Script #10.1 Send Notification

<i>Step #</i>	<i>Test Action</i>	<i>Expected Result</i>	<i>Status</i>	<i>Reference</i>
1	<i>Admin</i> menekan tombol “ <i>Notification</i> ”	Sistem menampilkan halaman <i>notification</i>	<i>Pass</i>	Gambar 4.75
2	<i>Admin</i> mengisi form notifikasi dan menekan tombol “ <i>Submit</i> ”	Sistem menampilkan <i>popup</i> “ <i>Notification Sent!</i> ”	<i>Pass</i>	Gambar 4.76

Penguji 1 (*User*)



Erick Gennady

Penguji 2 (*Member*)



Michelyn Niononi

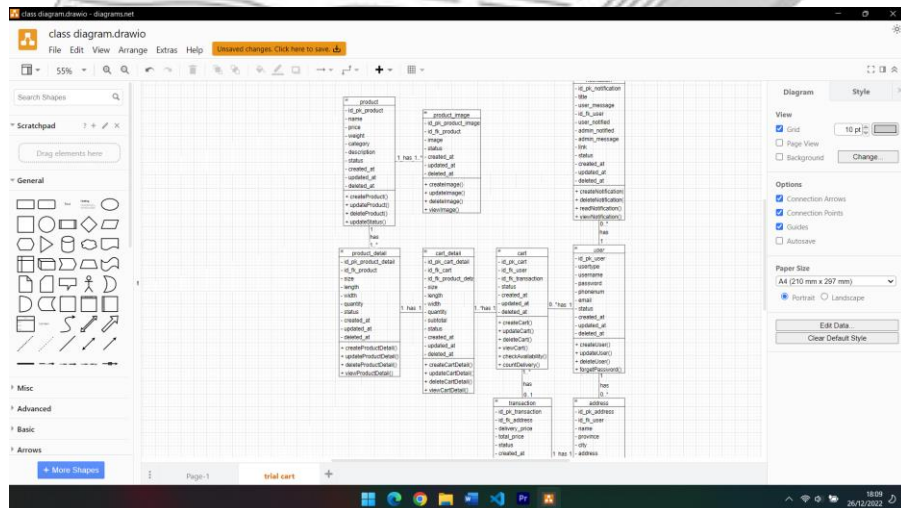
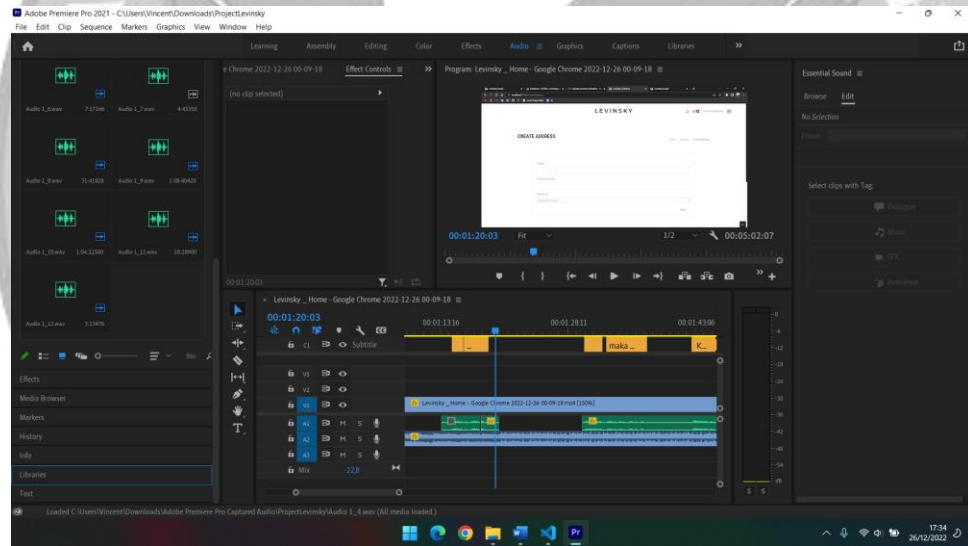
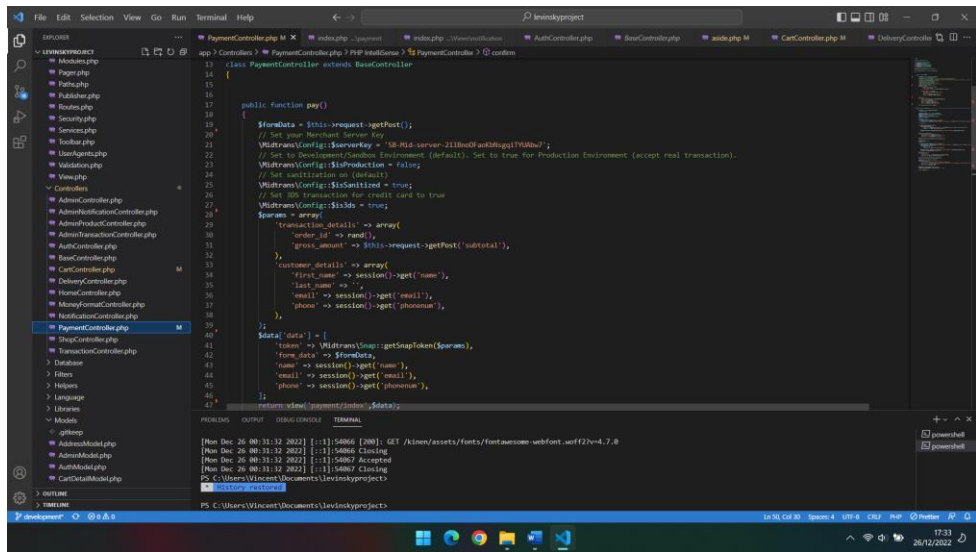
Penguji 3 (*Admin*)



Kentvink Teofilos
Anson Sebastian

LAMPIRAN D

DOKUMENTASI



MAP | Midtrans

dashboard.sandbox.midtrans.com/transactions

Book of Sales Design Ins

Levinsky Indonesia Pencarian

Transaksi

Transaksi

Pencarian Kirim ke Email

PAYMENT TYPE	TANGGAL & WAKTU	ID PESANAN	EMAIL PELANGGAN	JUMLAH	STATUS
Credit Card	Today, 00:12	875753734	vincentsukanto03@gmail.com	Rp. 116,000	SUCCESS
Credit Card	15/12/2022 18:10	1971503134	vincentsukanto03@gmail.com	Rp. 116,000	SETTLEMENT
Credit Card	15/12/2022 17:18	786958046	vincentsukanto03@gmail.com	Rp. 178,000	SETTLEMENT
Credit Card	15/12/2022 17:15	1352193333	vincentsukanto03@gmail.com	Rp. 123,000	SETTLEMENT
Credit Card	15/12/2022 16:53	343764696	vincentsukanto03@gmail.com	Rp. 254,000	SETTLEMENT
Credit Card	12/12/2022 23:19	870459554	vincentsukanto03@gmail.com	Rp. 193,000	SETTLEMENT
Credit Card	12/12/2022 23:19	870459554	vincentsukanto03@gmail.com	Rp. 193,000	FAILURE
Credit Card	09/12/2022 15:17	2107956244	michelynniononi1271@gmail.com	Rp. 250,000	SETTLEMENT
Credit Card	09/12/2022 14:37	650848124	michelynniononi1271@gmail.com	Rp. 131,000	SETTLEMENT

localhost / 127.0.0.1 / levinsky

localhost/phpmyadmin/index.php?route=/sql&db=levinsky_trial&table=cart&pos=0

phpMyAdmin

Recent Favorites

Information_schema

levinsky

levinsky_trial

New

address

cart

cart_detail

notification

product

product_detail

product_image

transaction

user

mysql

performance_schema

phpmyadmin

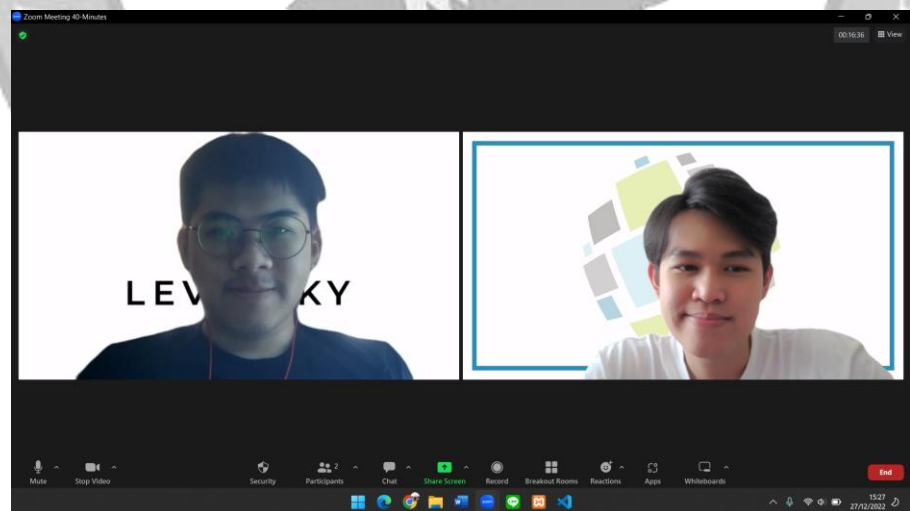
test

Extra options


id_pk_cart id_fk_user id_fk_transaction status created_at updated_at deleted_at

<input type="checkbox"/>	Edit Copy Delete	1	1	19 0	2022-12-08 16:50:38	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	1	20 0	2022-12-09 13:01:38	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	1	24 0	2022-12-09 13:59:00	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	6	22 0	2022-12-09 14:35:19	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	6	6	23 0	2022-12-09 14:39:40	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	7	1	25 0	2022-12-12 23:21:00	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	8	1	26 0	2022-12-15 17:06:22	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	9	1	27 0	2022-12-15 17:16:43	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	10	1	28 0	2022-12-15 18:10:12	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	11	1	NULL 1	2022-12-16 22:17:52	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	12	9	NULL 1	2022-12-26 00:00:19	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	13	10	29 0	2022-12-26 00:10:44	NULL	NULL









Check all With selected Edit Copy Delete Export

















LAMPIRAN E
FORM LEMBAR MONITORING
BIMBINGAN TUGAS AKHIR

	UNIVERSITAS PELITA HARAPAN SISTEM PENJAMINAN MUTU INTERNAL	No. Dok. : FOR03-M/PRO07/STA05/SPMI-UPH
		Revisi : 01
	FORM LEMBAR MONITORING BIMBINGAN TUGAS AKHIR	Tanggal : 2 Januari 2023
		Halaman : 1 dari 1

Nama Mahasiswa / *Student Name* : Vincent Sukamto
 Nama Dosen Pembimbing : Calandra Alencia Haryani, SE, S.SI., M.TI. &
/Final Project Advisor Kusno Prasetya Ph.D.
 NPM / *Student ID Number* : 01081190008
 Program Studi / *Study Program* : Sistem Informasi
 Fakultas / *Faculty* : Ilmu Komputer
 Lokasi Kampus / *Campus Location* : UPH Lippo Village
 Semester : 10

Minggu Ke- Week of-	Hari Day	Tanggal Date	Materi yang didiskusikan The Material Discussed	Tanda Tangan Dosen Pembimbing Final Project Advisor Signature		Catatan Remarks
				Calandra Alencia Haryani, S.E, S.SI., M.TI.	Kusno Prasetya Ph.D.	
1	Senin	22/8/2022	Membahas topik dan gambaran besar Tugas Akhir bersama dosen pembimbing			- Menetapkan judul Tugas Akhir - Menetapkan untuk menggunakan API Midtrans dan API RajaOngkir
2	Jumat	9/9/2022	Membahas bab 1 dan bab 2 Tugas Akhir bersama dosen pembimbing			- Memperbaiki bagian latar belakang, - Memperbaiki penggunaan bahasa yang lebih jelas pada bab 1 dan bab 2
3	Jumat	21/10/2022	Melakukan <i>weekly progress update</i>			- Menyelesaikan halaman utama dan halaman produk
4	Jumat	28/10/2022	Melakukan <i>weekly progress update</i> dan mengubah template UI sistem usulan			- Menyelesaikan fitur <i>cart</i> dan <i>checkout</i> - Mengubah template UI sistem usulan

5	Rabu	16/11/2022	Melakukan <i>weekly progress update</i> dan membahas mengenai tahap pemodelan fungsional			- Membuat <i>use case</i> dan <i>activity diagram</i>
6	Senin	21/11/2022	Melakukan <i>weekly progress update</i>			- Memperbaiki <i>use case</i> dan <i>activity diagram</i> - Membuat pemodelan struktural
7	Jumat	25/11/2022	Melakukan <i>weekly progress update</i> dan mengerjakan bab 4			- Menyelesaikan <i>class diagram</i> , TRD, <i>table description</i> dan <i>use case description</i>
8	Jumat	9/12/2022	Melakukan <i>weekly progress update</i> dan menyelesaikan bab 4			- Menyelesaikan tahapan perancangan (<i>Wireframe</i> , UI dan penjelasan API)
9	Jumat	16/12/2022	Melakukan <i>weekly progress update</i> dan mengerjakan bab 5			- Mengerjakan kesimpulan dan saran
10	Senin	19/12/2022	Melakukan <i>weekly progress update</i> dan membuat daftar isi, daftar tabel dan daftar gambar			- Membuat daftar isi, daftar tabel dan daftar gambar
11	Kamis	22/12/2022	Melakukan <i>weekly progress update</i> dan merapikan struktur buku TA			- Merapikan seluruh struktur buku TA

Karawaci, 2 Januari 2023

Disetujui oleh/*Approved by:*
Dosen Pembimbing 1/*Final*
Project Advisor 1



Calandra Alencia Haryani

Tanda Tangan & Nama

Lengkap/*Signature & Full Name*

Disetujui oleh/*Approved by:*
Dosen Pembimbing 2/*Final*
Project Advisor 2



Kusno Prasetya

Tanda Tangan & Nama

Lengkap/*Signature & Full Name*

Diketahui oleh/*Acknowledged by:*
Kaprod/Department Chair



Arnold Aribowo

Tanda Tangan & Nama

Lengkap/*Signature & Full Name*

True Knowledge, Faith in Christ, Godly Character