

3. ANALISA DAN DESAIN SISTEM

Dalam bab ini dijelaskan mengenai analisa sistem, desain sistem dan desain interface. Pada pembahasan mengenai desain sistem, bagian-bagian yang dibahas meliputi *flowchart*, desain ERD dan database.

3.1. Analisa Sistem

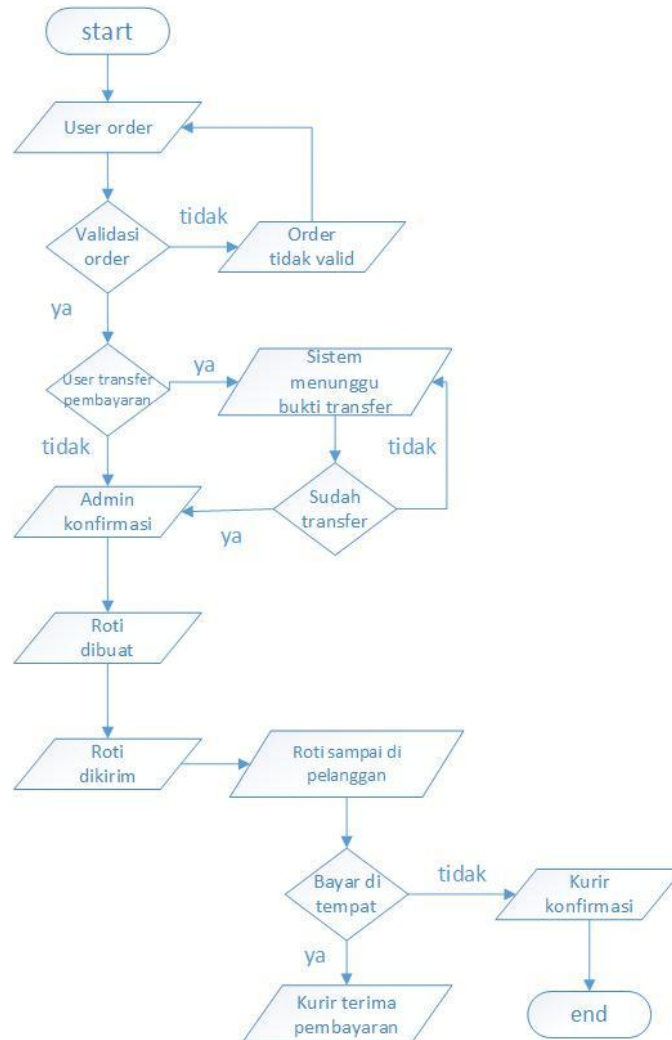
Perangkat lunak yang ingin dibuat adalah transaksi pesan antar atau distribusi *Delivery order* berbasis *web* sebagai *user* dan juga responsive sehingga bisa juga digunakan melalui *mobile* untuk dimanfaatkan bagi masyarakat sekitar Surabaya untuk pemesanan roti pada toko roti Vania Bakery. Aplikasi *Delivery order* ini menolong pengguna memesan roti dengan mudah, praktis serta mengefisienkan waktu dan tenaga. Layanan ini juga menyediakan fitur-fitur yang mempermudah dan memperjelas aplikasi pemesanan roti ini antara lain *tracking area* bagi kurir yang mengirim pesanan agar mempermudah kurir dalam menentukan route pengiriman pesanan, *recent order* yang menyimpan data pemesanan tiap *user* agar *user* tersebut bisa mengetahui riwayat pesannya, fitur notifikasi sebagai tanggapan dari admin atas pesanan *user* seperti *order in processing* atau *order delivered* dimana hal tersebut juga berhubungan bahwa pesanan tidak dapat dibatalkan karena sedang diproses / sedang dikirim, fitur *report* bagi admin untuk melihat hasil penjualan tiap bulannya. Adanya aplikasi ini dapat membantu meningkatkan perekonomian perusahaan Vania Bakery dalam meningkatkan jasa pelayanan dan juga menjaring lebih banyak pelanggan. Beberapa keuntungan layanan *Delivery order* yang ditawarkan dan menjadi konsep dasar dari penyediaan layanan adalah kemudahan dalam melakukan komunikasi pemesanan, kelebihan dalam hal waktu pengiriman.

3.2. Desain Sistem

Pada bagian ini menjelaskan seluruh bagian beserta alurnya dari aplikasi yang dibuat, dan juga penjelasan mengenai tiap bagian beserta proses dan *interfacenya*.

3.2.1. Kerangka Aplikasi

Kerangka aplikasi ini menjelaskan mengenai proses dari *user* memesan produk yang ditawarkan hingga selesai. *Design* dapat dilihat pada Gambar 3.1.



Gambar 3.1. *Flowchart* Proses *Delivery order*

Flowchart pada Gambar 3.1 merupakan diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. *Flowchart* ini yang menggambarkan seluruh input ke sistem atau output dari sistem. *Flowchart* memberi gambaran tentang keseluruhan sistem. Sistem *Delivery order* ini terbagi menjadi 3 pengguna, diantaranya adalah admin, *user*, dan kurir / karyawan.

- *User*, pengguna aplikasi

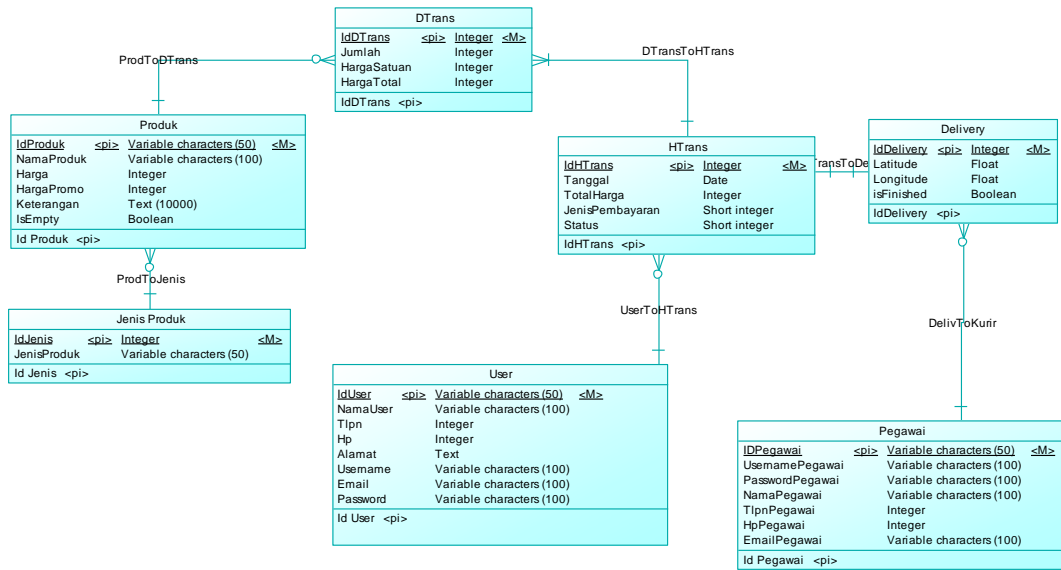
User merupakan target utama pendukung aplikasi ini berjalan dari proses pemesanan roti. Tahap pertama relasi yang terjadi antara *user* dengan

sistem ialah *user* melakukan *order* roti masuk ke dalam sistem, sehingga sistem dapat menerima dan meneruskan kepada pihak admin toko roti untuk memberikan konfirmasi atas pesanan *user*. Jika admin toko roti belum memberikan konfirmasi pada pesanan *user* maka status yang ada pada halaman *recent order* milik *user* adalah *waiting*, tetapi setelah mendapat konfirmasi oleh admin *delivery order*, maka status tersebut diperbarui menjadi *on process* yang artinya pesanan *user* sedang diproses . Kemudian jika pesanan sudah *ready*, akan ada konfirmasi status berubah menjadi *delivered* yang artinya pesanan sudah dikirim. Dan yang terakhir jika pesanan telah tiba dan diterima oleh *user*, maka status tersebut akan diupdate menjadi *accepted*.

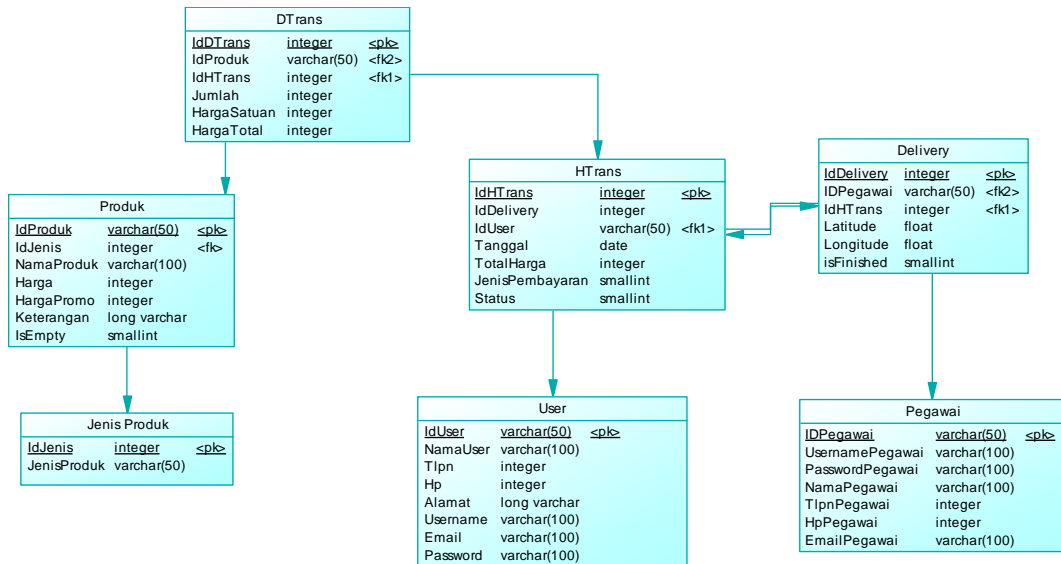
- **Pegawai/Kurir**
Pegawai memiliki peran yaitu mengatur proses transaksi pemesanan yang ada, seperti menanggapi pemesanan dari *user* atas pesannya, mengkonfirmasi atas pesanan *user* yang sudah diterima oleh *user* yang memesan, dan melihat lokasi pembeli melalui GPS.
- **Admin *delivery order***
Sebagai admin secara umum hanya bertugas memonitoring segala aktifitas yang terjadi pada saat proses pemesanan berlangsung, namun hal lain yang admin lakukan terhadap sistem ialah mengupdate informasi seluruh pengguna yang berkaitan pada proses pemesanan apabila sewaktu-waktu dibutuhkan, input data (*add, edit, delete*) roti dan pegawai yang tersedia pada aplikasi, menyimpan semua *log activities* yang terjadi pada saat proses pemesanan berlangsung.

3.2.2. Design ERD (Entity Relationship Diagram)

Perancangan data base ini menggunakan *software Sybase Power Designer*. Tahap dalam pembuatan tabel yang pertama adalah *Conceptual Data Model*. Selanjutnya dari format tersebut diubah menjadi format *Physical Data Model* dan membuat *database* pada phpMyAdmin. ERD dapat dilihat pada Gambar 3.2 dan Gambar 3.3.



Gambar 3.2. Conceptual Data Model



Gambar 3.3. Physical Data Model

Tabel Pegawai yang tertera pada Tabel 3.1 digunakan untuk menyimpan data pegawai yang bekerja di Vania Bakery.

Tabel 3. 1 Pegawai

Nama Field	Tipe Data	Keterangan
IdPegawai	Varchar(50)	Menyimpan id pegawai
NamaPegawai	Varchar(100)	Menyimpan nama pegawai
TlpnPegawai	Int	Menyimpan nomer telepon pegawai
HpPegawai	Int	Menyimpan nomer <i>handphone</i> pegawai
UsernamePegawai	Varchar(100)	Menyimpan <i>username</i> pegawai untuk

		<i>login</i> bagi pegawai
<i>PasswordPegawai</i>	Varchar(100)	Menyimpan <i>password</i> pegawai untuk <i>login</i> bagi pegawai
<i>EmailPegawai</i>	Varchar(100)	Menyimpan <i>email</i> pegawai

Tabel *User* yang tertera pada Tabel 3.2 digunakan untuk menyimpan informasi data dari *user*.

Tabel 3. 2 *User*

Nama Field	Tipe Data	Keterangan
<i>IdUser</i>	Varchar(50)	Menyimpan id <i>user</i>
<i>NamaUser</i>	Varchar(100)	Menyimpan nama <i>user</i>
<i>Tlpn</i>	Int	Menyimpan nomer telepon <i>user</i>
<i>Hp</i>	Int	Menyimpan nomer <i>handphone</i> <i>user</i>
<i>Alamat</i>	Text	Menyimpan alamat <i>user</i>
<i>Username</i>	Varchar(100)	Menyimpan <i>username</i> untuk <i>login</i> bagi <i>user</i>
<i>Password</i>	Varchar(100)	Menyimpan <i>password</i> untuk <i>login</i> bagi <i>user</i>
<i>Email</i>	Varchar(100)	Menyimpan <i>email</i> pegawai

Tabel *Produk* yang tertera pada Tabel 3.3 digunakan untuk menyimpan data produk.

Tabel 3. 3 *Produk*

Nama Field	Tipe Data	Keterangan
<i>IdProduk</i>	Varchar(50)	Menyimpan id produk
<i>NamaProduk</i>	Varchar(100)	Menyimpan nama tiap produk
<i>Harga</i>	Int	Menyimpan harga tiap produk
<i>HargaPromo</i>	Int	Menyimpan harga promo dari produk yang sedang dalam promo
<i>Keterangan</i>	Text	Menyimpanan keterangan dari tiap produk
<i>IsEmptyy</i>	Boolean	Digunakan untuk menentukan apakah

		produk kosong atau tidak
--	--	--------------------------

Tabel Jenis Produk yang tertera pada Tabel 3.4 digunakan untuk menyimpan jenis produk.

Tabel 3. 4 Jenis Produk

Nama Field	Tipe Data	Keterangan
IdJenis	Int	Menyimpan id jenis dari produk
JenisProduk	Varchar(50)	Menyimpan jenis produk

Tabel *Delivery* yang tertera pada Tabel 3.5 digunakan untuk menyimpan data *delivery*.

Tabel 3. 5 *Delivery*

Nama Field	Tipe Data	Keterangan
<i>IdDelivery</i>	Int	Menyimpan id <i>delivery</i>
Latitude	Float	Menyimpan latitude dari google maps dari <i>handphone</i> kurir
Longitude	Float	Menyimpan longitude dari google maps dari <i>handphone</i> kurir
IsFinished	Boolean	Menentukan apakah proses <i>delivery</i> sudah selesai atau belum

Tabel Header Transaksi yang tertera pada Tabel 3.6 digunakan untuk menyimpan data header transaksi.

Tabel 3. 6 Header Transaksi

Nama Field	Tipe Data	Keterangan
IdHTrans	Int	Menyimpan id header transaksi
Tanggal	Date	Menyimpan tanggal transaksi
TotalHarga	Int	Menyimpan total harga transaksi
JenisPembayaran	Short int	Menyimpan jenis pembayaran
Status	Short int	Menyimpan status transaksi

Tabel Detail Transaksi yang tertera pada Tabel 3.7 digunakan untuk menyimpan data detail transaksi.

Tabel 3. 7 Detail Transaksi

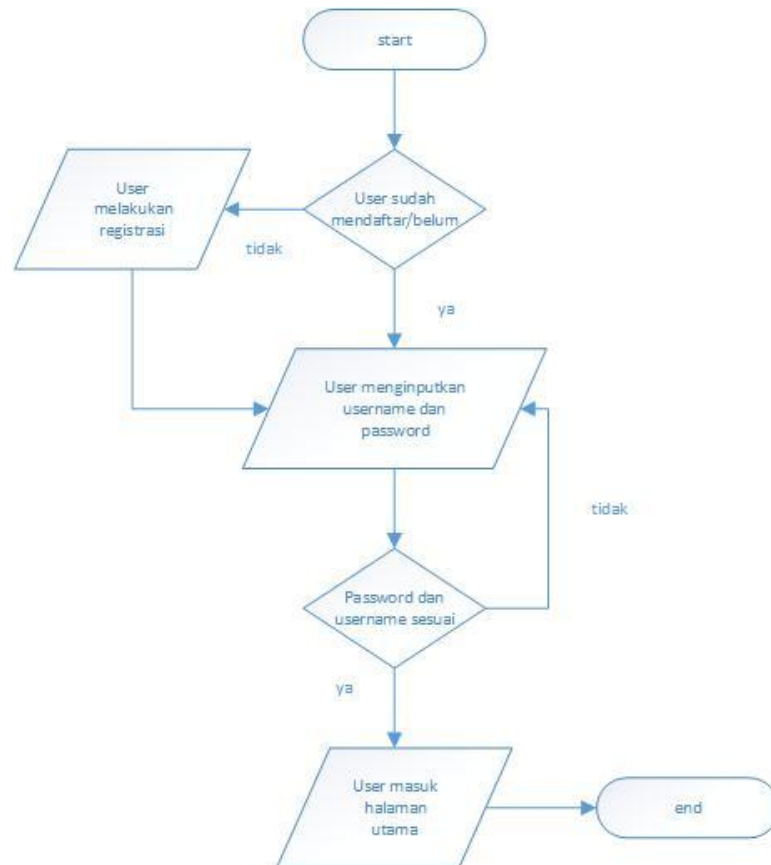
Nama Field	Tipe Data	Keterangan
IdDTrans	int	Menyimpan id detail transaksi
Jumlah	Int	Menyimpan jumlah produk dari transaksi
HargaSatuan	Int	Menyimpan harga satuan dari tiap produk
HargaTotal	Int	Menyimpan harga total dari tiap produk

3.3. *Desain Aplikasi User*

Pada bagian ini akan membahas semua tampilan pada halaman *user* beserta *flowchart*nya yang menjelaskan cara kerja program dari sebuah proses yang dilakukan oleh aplikasi ini.

3.3.1. *Login User*

Pada saat pertama membuka aplikasi ini, maka akan ada halaman utama tampilan toko roti Vania Bakery, jika *user* ingin melakukan pemesanan maka *user* diharuskan untuk melakukan *login* dengan memasukkan *username* dan *password* bila sudah melakukan register sebelumnya jika belum maka *user* harus melakukan register terlebih dahulu. Setelah itu barulah *user* menginputkan lagi *username* dan *password*, jika *username* dan *password* benar / sesuai maka akan *login* sebagai *user* dan *user* dibawa ke halaman utama *user*. Barulah *user* dapat melakukan transaksi pemesanan. *Design* dapat dilihat pada Gambar 3.4.



Gambar 3.4. *Flowchart Login User*

Gambar 3.4 adalah gambar *flowchart* mengenai alur proses *login* bagi *user* yang sudah melakukan registrasi maupun proses registrasi bagi *user* yang belum mendaftar.



Gambar 3.5. *Login*

Gambar 3.5 adalah tampilan ketika *user* ingin melakukan *login*, *user* dapat langsung mengisi *username* dan *password* dengan benar. Ketika *user* sudah benar

memasukan *username* dan *password* maka akan langsung kembali lagi ke halaman utama.

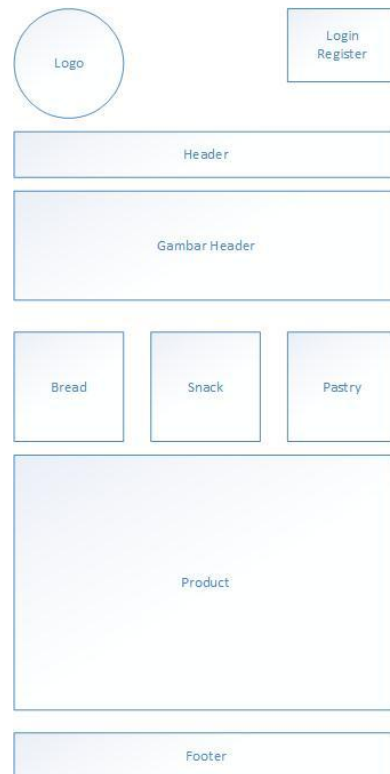
The diagram illustrates the layout of a registration page. At the top is a circular logo labeled 'Logo'. Below it is a rectangular header bar labeled 'Header'. The main content area is a large rectangle containing a list of registration fields: 'Nama', 'Username', 'Password', 'Confirm Password', 'Email', 'Handphone', 'Telephone', and 'Address'. At the bottom is a rectangular footer bar labeled 'Footer'.

Gambar 3.6. *Register*

Gambar 3.6 adalah gambar tampilan registrasi bagi *user* yang ingin melakukan registrasi. *User* yang ingin melakukan registrasi harus mengisi lengkap data yang sudah disediakan pada kolom-kolom registrasi, seperti nama, alamat, *telephone*, *handphone* dan juga *email*.

3.3.2. *Main menu User*

Bagi *user* yang sudah melakukan *login* maka *user* akan memasuki halaman utama toko roti Vania Bakery, disitu *user* langsung dapat melihat beberapa jenis roti beserta harganya dan juga ada 3 katagori penjualan yang ditawarkan oleh Vania Bakery yaitu *bread*, *pastry* dan *snack*.

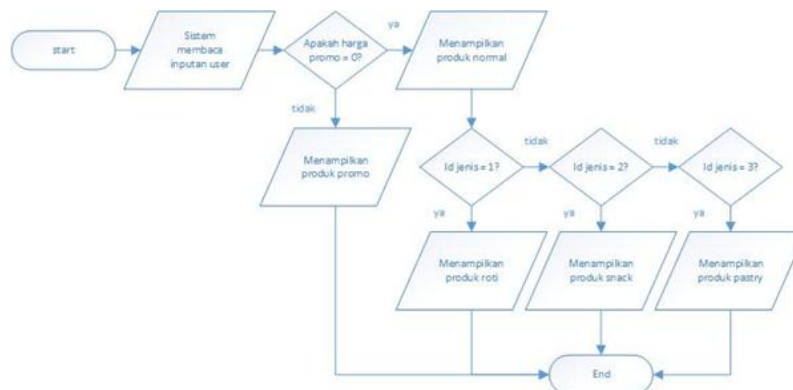


Gambar 3.7. Halaman Utama *User*

Gambar 3.7 adalah gambar halaman utama *user* setelah *user* melakukan *login*. Di halaman utama *user* dapat langsung memilih jenis produk apa yang ingin dipesan yaitu *bread*, *pastry* dan *snack*. Atau *user* dapat memilih langsung beberapa jenis roti yang ditampilkan di halaman utama.

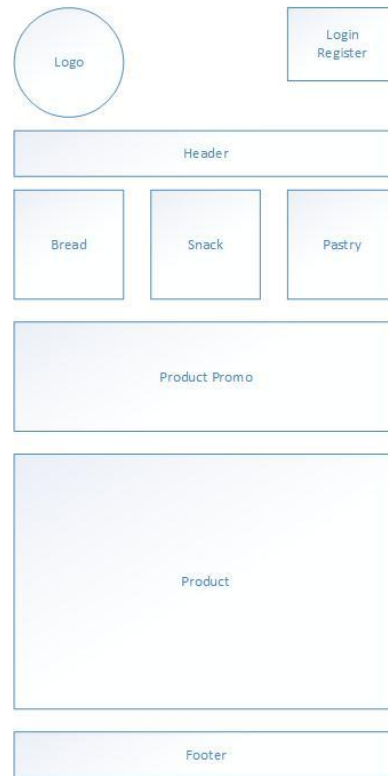
3.3.3. Produk *User*

User yang ingin melihat produk yang ada dapat memilih berdasarkan kategori yang ada yaitu roti, *snack* dan *pastry*.



Gambar 3.8. *Flowchart* Memilih produk

Gambar 3.8 adalah gambar *flowchart* yang menjelaskan proses bagi *user* yang ingin memilih produk.



Gambar 3.9. Halaman Produk

Gambar 3.9 adalah gambar halaman product yang berisi tentang roti yang sedang promo dan juga latest product atau produk terbaru dari Vania Bakery. Di halaman ini *user* juga dapat memilih produk Vania berdasarkan kategori yang sudah ada yaitu *bread*, *pastry*, dan *snack*. *User* juga langsung dapat memilih roti yang lagi promo atau langsung memilih produk yang ada di halaman ini untuk dipesan.

3.3.4. Halaman Roti, *Snack* dan *Pastry User*

Ketika *user* sudah memilih kategori produk yang akan dibeli maka akan masuk kehalaman berdasarkan kategori yang dipilih, jika *user* memilih kategori roti maka akan masuk ke halaman yang menampilkan produk berkategori roti begitu juga dengan *snack* dan *pastry*.



Gambar 3.10. Halaman Roti (*Bread*)

Gambar 3.10 adalah gambaran tampilan utama kategori roti. Pada halaman ini yang ditampilkan hanya semua produk yang jenisnya roti. Jadi jika *user* ingin melihat dan memesan produk yang ditawarkan Vania Bakery yang memiliki kategori/jenis roti saja dapat langsung menuju halaman roti. Begitu juga untuk *snack* dan *pastry*, telah disediakan juga halaman khusus hanya untuk produk kategori *snack* dan juga *pastry*.

3.3.5. Halaman Tampilan Utama Produk *User*

Ketika *user* sudah memasuki halaman produk berdasarkan kategori dan sudah memilih produk yang ingin dibeli maka *user* akan masuk utama produk itu untuk melanjutkan proses pembelian dengan cara memasukkan jumlah produk yang ingin dibeli dan menekan tombol *cart* untuk dimasukkan halaman pembelian / *cart*. *Design* dapat dilihat seperti pada gambar 3.11.



Gambar 3.11. Tampilan Utama Produk

Gambar 3.11 adalah gambar utama tiap jenis produk yang ada di dalam aplikasi. Pada halaman tersebut berisi nama produk, deskripsi tiap produk, harga normal dan harga promo jika sedang ada promo. Jika *user* ingin membeli produk tersebut tinggal menentukan jumlah produk yang ingin di beli pada kolom *quantity* setelah itu menekan tombol *add to cart* yang sudah disediakan.

3.3.6. Halaman *Cart*

Halaman *cart* adalah halaman tempat produk-produk yang sudah dipilih *user* untuk dibeli. Pada halaman ini *user* dapat menghapus produk yang sudah dipilih sebelumnya. *Design* dapat dilihat pada Gambar 3.12.

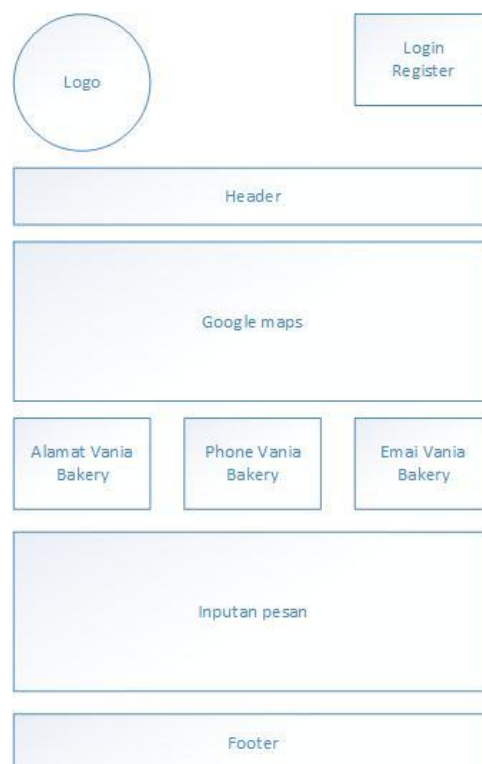


Gambar 3.12. Halaman *Cart*

Gambar 3.12 adalah gambar halaman *cart*. Setelah *user* yakin untuk membeli produk yang sudah dipilih *user* tinggal memilih jenis pembayar yang sudah disediakan yaitu pembayaran ditempat atau melalui *transfer*.

3.3.7. Halaman *Contact*

Halaman *contact* berisi tentang alamat, nomer *telephone* dan *email* dari Vania Bakery. Di halaman *contact* *user* juga dapat melihat lokasi dari toko roti Vania Bakery pada *google maps* dan juga dapat mengirim pesan ke Vania Bakery. *Design* dapat dilihat pada Gambar 3.13.



Gambar 3.13. Halaman *Contact*

Gambar 3.13 adalah halaman *contact* yang berguna untuk *user* mengetahui data-data Vania Bakery seperti lokasi pada *google maps*, alamat, nomer *telephone* dan *email*. Dan juga *user* yang ingin mengirim saran ke Vania Bakery melalui *email* juga dapat melalui halaman ini.

3.3.8. Halaman *Tracking*

Halaman *tracking* berguna bagi *user* untuk mengetahui dimana letak pesannya sekarang.



Gambar 3.14. *Tracking*

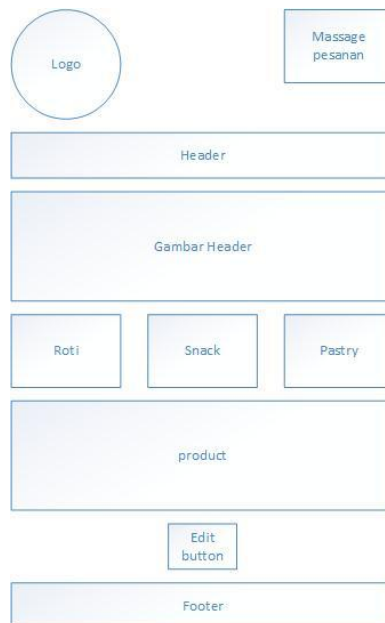
Gambar 3.14 adalah gambar halaman *tracking* untuk mengetahui posisi kurir yang sedang mengantar pesanan dari *user*.

3.4. Design Aplikasi Admin

Pada bagian ini akan membahas semua tampilan pada halaman admin.

3.4.1. Main menu Admin

Halaman *main menu* pada admin mirip dengan halaman utama pada *user* tetapi pada halaman *main menu* admin memiliki fungsi *edit* pada tampilan produk yang ada pada halaman tersebut.



Gambar 3.15. *Main menu Admin*

Gambar 3.15 adalah gambar halaman utama pada admin yang memiliki fungsi *edit* untuk mengedit tampilan produk pada halaman utama *user*.

3.4.2. Produk Admin

Halaman produk pada admin memiliki tampilan yang sama pada halaman produk pada *user*, tetapi halaman produk pada admin memiliki fungsi *edit* sehingga admin dapat melakukan *edit* pada tampilan pada halaman produk.



Gambar 3.16. Produk Admin

Gambar 3.16 adalah gambar halaman produk bagi admin, pada halaman ini juga terdapat fungsi *edit* untuk melakukan *edit* produk yang ada dan juga *delete* untuk menghapus produk yang ada.

3.4.3. Halaman Kategori

Halaman kategori pada admin menampilkan produk berdasarkan kategori yaitu *roti*, *snack* dan *pastry*. Pada halaman tersebut juga memiliki fungsi *insert* untuk menambah jenis produk pada tiap kategori, fungsi *edit* untuk mengedit semua produk yang sudah ada dan fungsi *delete* untuk menghapus produk yang ada.



Gambar 3.17. Halaman Kategori

Gambar 3.17 adalah gambar halaman kategori bagi admin yang berfungsi untuk *edit* dan *delete* data produk berdasarkan kategori dan juga fungsi *insert* produk berdasarkan kategori tersebut.

3.4.4. Halaman Pesanan

Halaman pesanan berisi tentang semua pesanan dari *user* yang masuk, pada halaman ini admin juga dapat memberi notifikasi pada *user* untuk menanggapi pesanan *user*.



Gambar 3.18. Halaman Pesanan

Gambar 3.18 adalah halaman pesanan dari semua *user* yang sudah memesan produk tersebut dihalaman ini admin juga bisa memberi tanggapan pada *user* mengenai pesannya berupa notifikasi.

3.4.5. Halaman Utama Produk Admin

Halaman utama produk pada admin memiliki tampilan yang juga mirip dengan *user* hanya saja pada halaman utama produk admin memiliki fitur *edit* yang berfungsi untuk mengedit data produk tersebut jika ada perubahan.



Gambar 3.19. Gambar Utama produk

Gambar 3.19 adalah gambar halaman utama dari tiap produk yang ada di Vania Bakery, disini terdapat fungsi *edit* yang berguna bagi admin untuk mengedit data dari tiap produk yang ada seperti harga dan deskripsi produk.

3.4.6. Halaman Pegawai

Halaman pegawai berguna supaya admin dapat melihat data diri dari tiap pegawai yang bekerja di Vania Bakery. Di halaman ini juga terdapat fungsi *insert*, *edit* dan *delete*.



Gambar 3.20. Halaman Pegawai

Gambar 3.20 adalah gambar dari halaman pegawai bagi admin, disini admin dapat melihat data-data pegawai yang masih bekerja, selain itu admin juga dapat merubah data pegawai tersebut, *mendelete* pegawai yang sudah tidak bekerja dan juga menambah data pegawai baru.

3.4.7. Halaman Tracking Admin

Halaman *tracking* berguna agar admin dapat mengetahui lokasi dari kurir yang sedang mengirim pesan dari *user*.



Gambar 3.21. Halaman Tracking

Gambar 3.21 adalah gambar halaman *tracking* bagi admin untuk melihat lokasi dari kurir yang mengantarkan pesanan *user*.

3.4.8. Halaman *Report*

Halaman *report* berisi tentang keuntungan Vania Bakery tiap bulannya dan juga admin dapat melihat produk yang paling banyak dipesan tiap bulannya.



Gambar 3.22. Halaman *Report*

Gambar 3.22 adalah gambar halaman *report* bagi admin untuk melihat hasil keuntungan penjualan tiap bulannya dan juga produk apa yang paling banyak dipesan tiap bulannya.