

BAB III

METODE PENELITIAN

3.1. Alat Pendukung Penelitian

Dalam membangun sistem, dibutuhkan peralatan pendukung yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat yang digunakan dalam penelitian ini adalah sebagai berikut:

3.1.1 Perangkat Keras

Adapun perangkat keras yang digunakan dalam perancangan dan pembuatan *website e-commerce* MAXCloth ini, yaitu:

1. *Processor* intel core i3.
2. RAM 4 GB.
3. *Harddisk* 500 GB.
4. Monitor, *keyboard*, dan *mouse*.

3.1.2 Perangkat Lunak

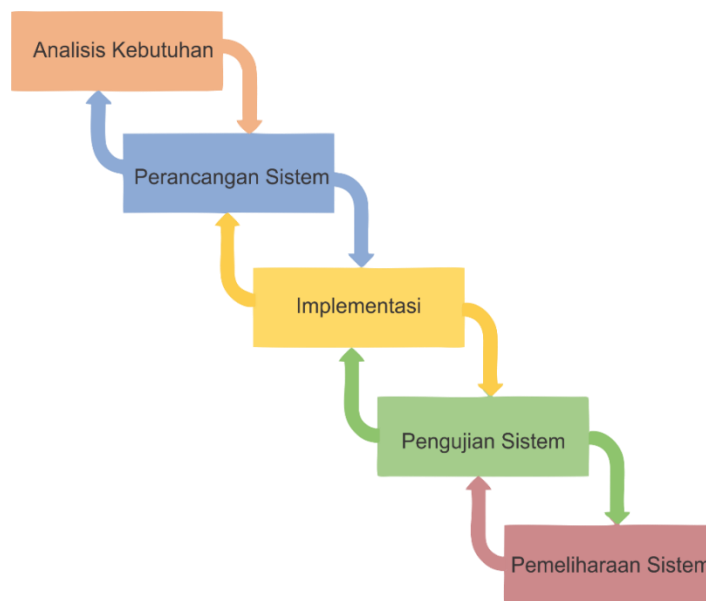
Adapun perangkat lunak yang digunakan dalam perancangan dan pembuatan *website e-commerce* MAXCloth ini, yaitu:

1. Sistem operasi *windows* 8.1.
2. *Tools*: XAMPP
→ Sebagai server yang berdiri sendiri(localhost) yang menghubungkan mysql dengan website yang berada di localdisk.
3. *Web browser*: Mozilla firefox dan Internet Explorer.
→ Aplikasi/perangkat lunak berfungsi untuk menerima dan menyajikan subaer informasi.
4. *Web editor*: Notepad++.
→ Aplikasi atau perangkat lunak yang berfungsi untuk mencatat informasi dan juga sebagai tempat editing web.
5. Bahasa pemrograman: PHP, *Javascript*, HTML, CSS.

→ Merupakan susatu hipunan dari aturan sintaks dan sematik untuk mendefinisikan program komputer.

3.2. Alur Penelitian

Dalam penelitian ini, metode yang digunakan untuk pengembangan sistem *website* e-commerce MAXCloth adalah metode *waterfall*. Model *waterfall* ini sebenarnya adalah “*Linear Sequential Model*”, yang sering juga disebut dengan “*classic life cycle*”. Disebut dengan *waterfall* karena tahap demi tahap yang harus dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Tahapan tersebut dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Langkah Penelitian dengan Metode *Waterfall*

Alur penelitian dilakukan dalam beberapa tahap, sebagai berikut:

1. Analisis Kebutuhan.

Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*.

2. Perancangan Sistem.

Tujuan dalam tahap ini untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam

menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3. Implementasi.

Dalam tahap ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya.

4. Pengujian Sistem.

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

5. Pemeliharaan Sistem

Pada perangkat lunak yang sudah jadi, semuanya dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada tahap sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

3.3. Sistem Sedang Berjalan di E-commerce MAXCloth

3.3.1 Analisis Membership

1. Pelanggan yang ingin membeli produk harus mempunyai akun.
2. Pelanggan melakukan registasi di halaman *register*.
3. Pelanggan menginput nama, *username*, *password*, *email*, nomer hp, dan juga alamat.
4. Setelah menginput data, pelanggan resmi menjadi member di *website* MAXCloth

3.3.2 Analisis Pembelian

1. Setelah memiliki akun, *member* sudah dapat melakukan pembelian secara *online*.
2. Pembayaran dilakukan *via transfer bank*.
3. Setelah *member* melakukan pembayaran, tinggal menunggu konfirmasi dari admin.

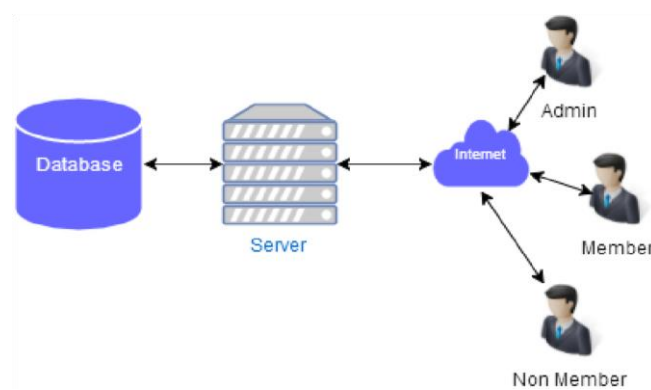
4. Barang yang dipesan akan segera diproses dan dikirim.

3.3.3 Analisis Penjualan

Admin melakukan pengecekan pada laporan penjualan dan *update* data pada stok barang dengan pengurangan atau penambahan barang sesuai kebutuhan.

3.4. Arsitekture

Pada tahap ini dilakukan penentuan perancangan sistem yang akan digunakan. Berikut ini merupakan alur arsitektur pada Gambar 3.2.



Gambar 3. 2 Arsitektur Desain

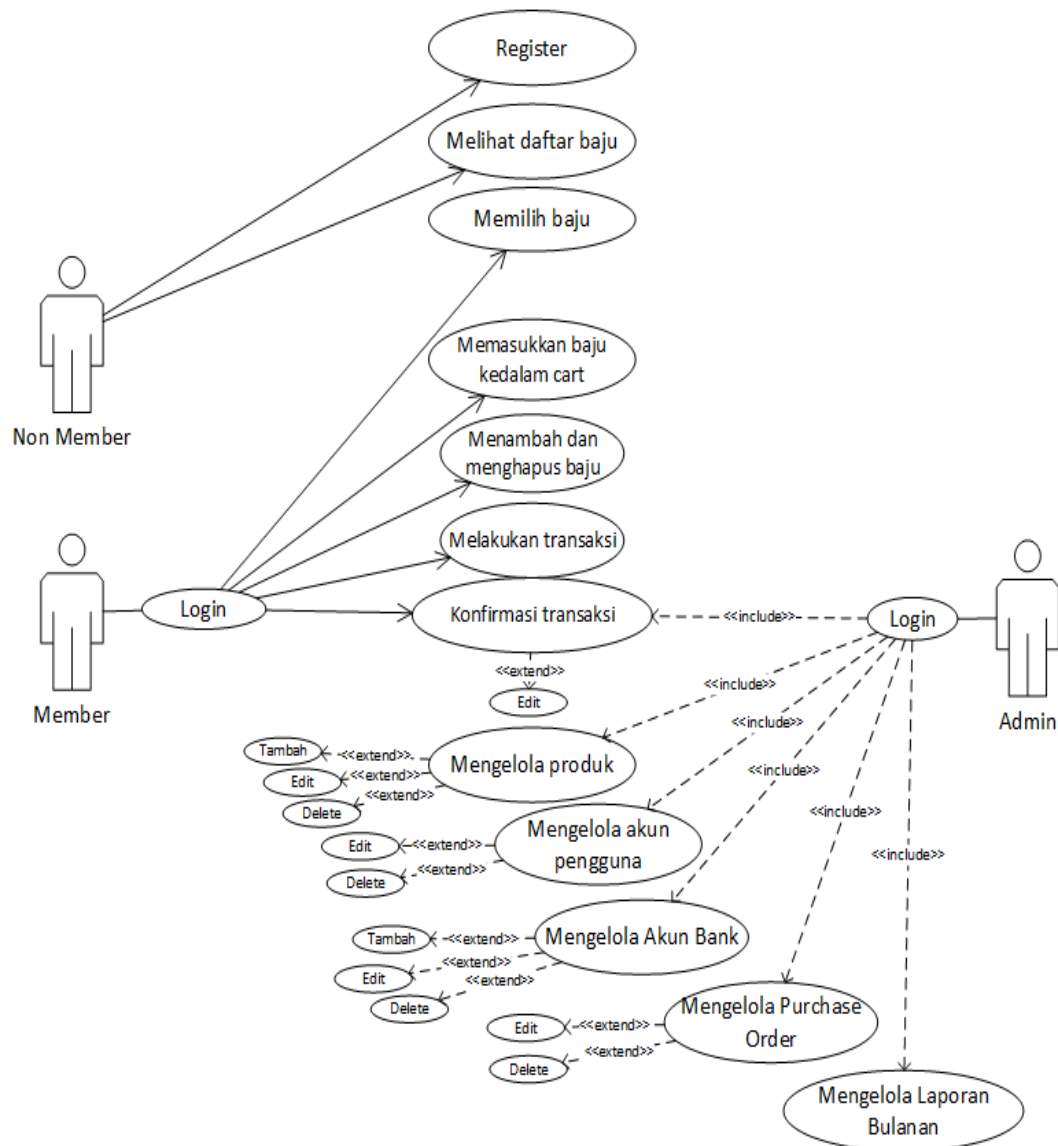
Database server menggunakan *MySQL* dan menggunakan tools *XAMPP*. Hubungan antar pengguna dan *server* menggunakan *internet* dan *web browser* pada perangkat pengguna. Saat pengguna mengakses aplikasi, *server* memuat antar muka dan melakukan pengambilan data yang dilakukan dari *database server* dapat dilihat pada tabel 3.1.

Tabel 3. 1 Keterangan Komponen Arsitektur

	<ol style="list-style-type: none"> 1. Admin 2. Member 3. Non Member
	<ol style="list-style-type: none"> 1. Data Barang 2. Kategori Barang 3. Detail Barang 4. Data Transaksi 5. Slide Banner 6. Data Member 7. Data Pengurus
	Server digunakan untuk menyimpan data-data.
	Jaringan komputer yang saling terhubung

3.5. Use Case diagram

Berdasarkan analisis kebutuhan yang diperlukan maka dibuatlah *use case* diagram untuk membantu perancangan *website e-commerce* MAXCloth. Berikut adalah *use case* yang telah dibuat.



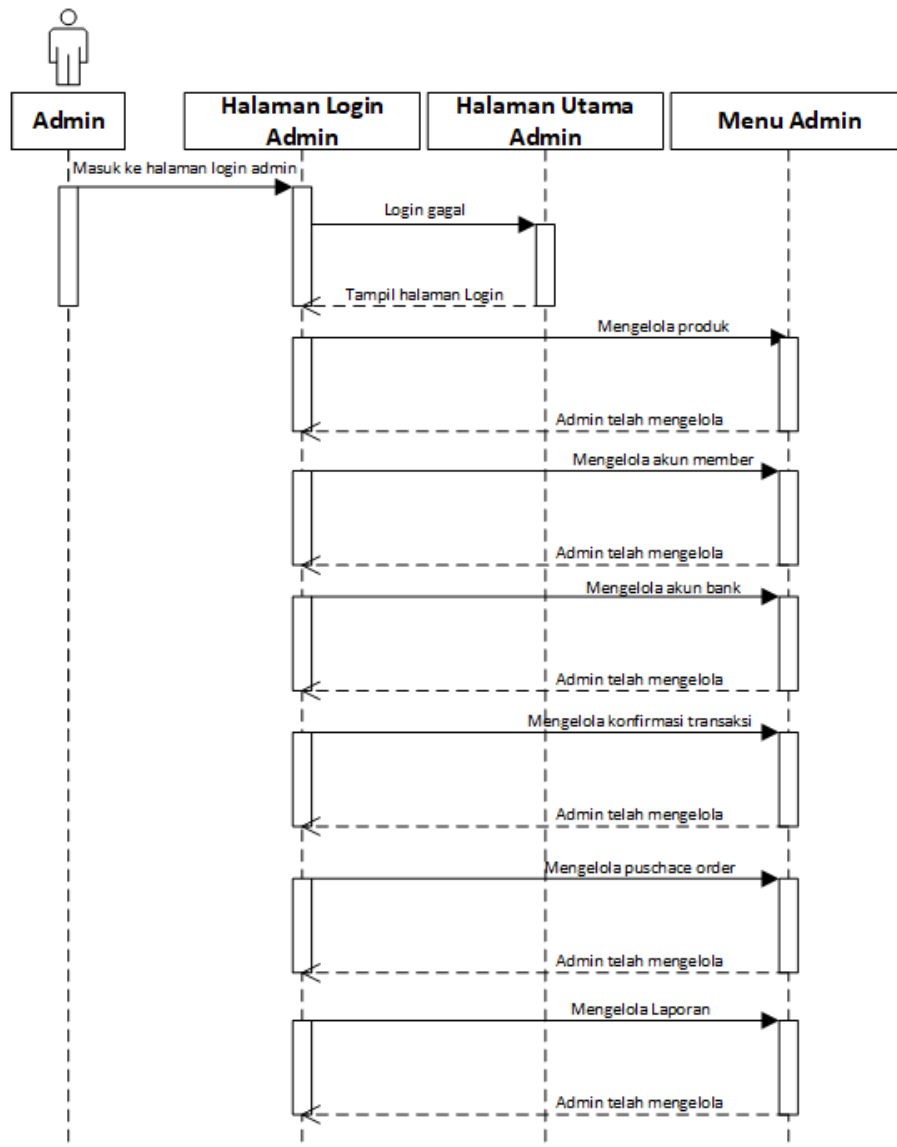
Gambar 3.3 Use Case Diagram MAXCloth

Dari Gambar 3.3 dapat di jelaskan bahwa *admin* berhubungan langsung pada *update* data barang seperti jumlah barang, data pengurus, data *member*, dan data transaksi barang sehingga dapat melakukan pengelolaan data produk, sedangkan *customer* Terdapat 2 *user* yaitu : *member* dan *non member*.

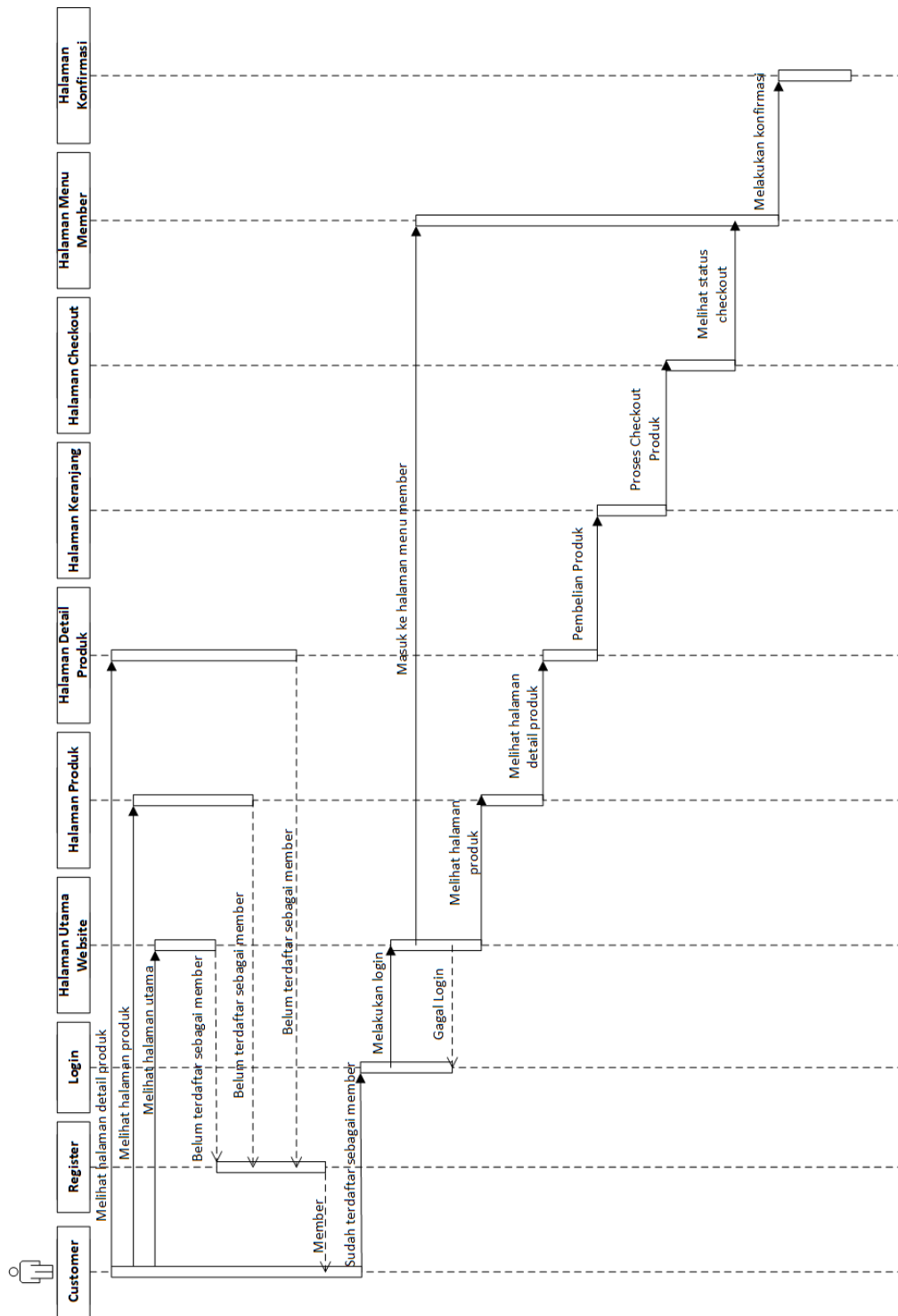
1. *Member* dapat melakukan pemesanan melalui *order* dan melakukan pembayaran secara *transfer*.
2. *Non member* hanya dapat melihat produk yang ada pada *website* dan dapat melakukan registrasi *member* baru.

3.6. Sequence diagram

Gambar 3.4 menunjukkan alur kelola *website e-commerce* MAXcloth. Bermula dari *admin* masuk ke halaman *login* dan juga melakukan *login* di *form login admin*. Setelah melakukan *login*, *admin* dapat mengelola data produk, *member*, akun bang, konfirmasi transaksi, *purchase order* dan juga laporan data *website e-commerce* MAXcloth.



Gambar 3.4 *Sequence diagram admin*



Gambar 3. 5 Sequence diagram customer

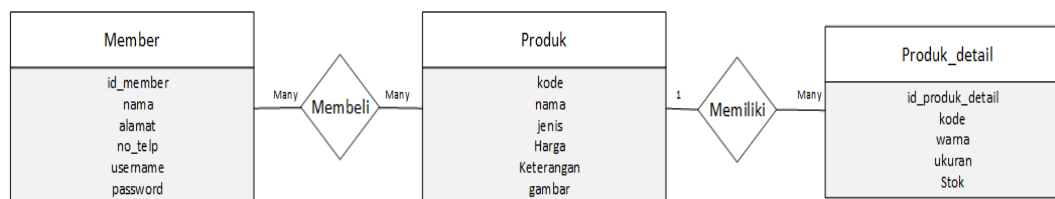
Gambar 3.5 menunjukkan *sequence* diagram pada kegiatan alur proses pemesanan yaitu *customer* menjadi *member* dan dari *member* dapat melakukan pemesanan melalui *order* yang tersedia dengan proses pembayaran. Setelah itu, *member* akan menunggu konfirmasi pemesanan melalui status transaksi yang tersedia di akun *member*.

3.7. Perancangan Basis Data

3.7.1 Entity Relationship Diagram

Pada Gambar 3.14 menjelaskan relasi antar data dalam basis data yang terdapat didalam sistem website *e-commerce* MAXCloth:

Pada entitas *member* memiliki relasi *many-to-many* ke entitas produk, memiliki makna setiap entias dari member dapat berhubungan dengan banyak entitas himpunan entitas produk dan demikian pula sebaliknya artinya semua *member* dapat memilih banyak barang dengan kode yang sama untuk melakukan transaksi, sedangkan pada entitas produk memiliki relasi *one-to-many* ke entitas produk detail memiliki makna setiap produk dalam satu kode produk memiliki banyak detail produk meliputi warna, ukuran dan juga stok.



Gambar 3. 6 ER Diagram

Gambar 3.6 yang menunjukkan *Entity Relationship Diagram* dan memiliki 3 buah *entitas* yaitu:

1. Member
2. Produk
3. Produk_detail

3.7.2. Perancangan Database Logikal

3.7.2.1. Konversi ERD Pada Tabel

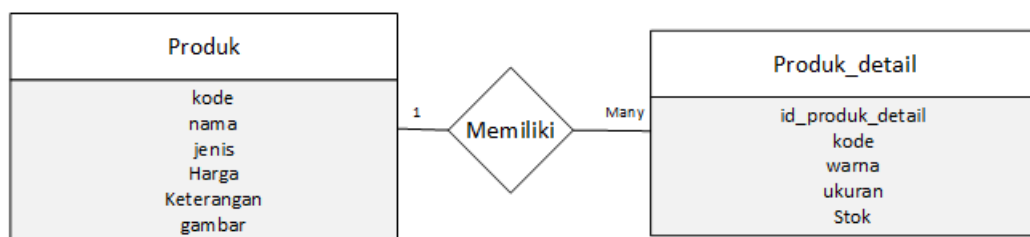
Konversi ERD pada tabel adalah proses memasukkan entitas menjadi nama tabel dan attritube menjadi nama kolom beserta relasinya kedalam sebuah tabel yang saling berhubungan.

1. Konversi dengan relasi 1:M (Many to Many)

Konversi entitas yang memiliki relasi 1:M (1 to many) yaitu menghubungkan dua entitas, diwujudkan dalam bentuk tabel (file data) khusus yang memiliki field (foreign key) yang berasal dari kunci-kunci (key) dari entitas yang dihubungkan sehingga menciptakan tabel baru. Pada ERD yang ditunjukkan pada gambar 3.14 terdapat relasi many to many yaitu antara entitas member dengan entitas produk. Dari entitas-entitas tersebut akan direpresentasikan kedalam tabel, dapat dilihat pada gambar – sebagai berikut.

a. Relasi Entitas Produk dengan Produk Detail

Entitas produk memiliki relasi *one-to-many* ke entitas produk detail memiliki makna setiap produk dalam satu kode produk memiliki banyak detail produk meliputi warna, ukuran dan juga stok, dapat dilihat pada gambar 3.7.



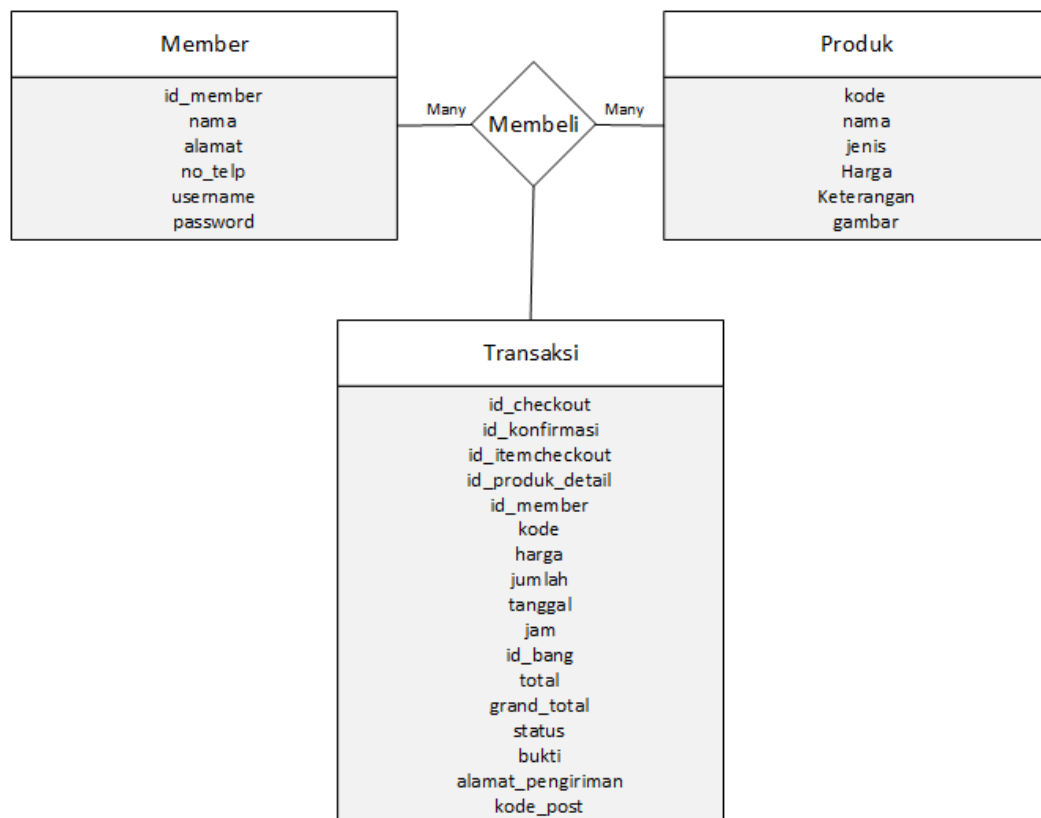
Gambar 3. 7 Relasi Produk dengan Produk Detail

2. Konversi dengan relasi M:M (One to Many)

Konversi entitas yang memiliki relasi M:M (many to many) yaitu menghubungkan dua entitas, diwujudkan dalam bentuk tabel (file data) khusus yang memiliki field (foreign key) yang berasal dari kunci-kunci (key) dari entitas yang dihubungkan sehingga menciptakan tabel baru. Pada ERD yang ditunjukkan pada gambar 3.14 terdapat relasi many to many yaitu antara entitas member dengan entitas produk. Dari entitas-entitas tersebut akan direpresentasikan kedalam tabel, dapat dilihat pada gambar – sebagai berikut.

b. Relasi Entitas *Member* dengan Produk

Entitas member memiliki relasi *many to many* dengan entitas produk sehingga tercipta tabel baru yaitu tabel *checkout*. *Member* yang telah memilih produk akan menuju ke tahap *checkout*, dapat dilihat pada gambar 3.8.



Gambar 3. 8 Relasi Produk dengan Produk Detail

3. Normalisasi hasil relasi entitas member dengan produk

Normalisasi adalah untuk memperoleh skema relasi mejadi bentuk normal tertinggi, sehingga tidak terjadi pengulangan data yang tidak perlu dan menghilangkan anomali.

a. Bentuk Normalisasi Pertama (1NF)

Suatu relasi dikatakan sudah memenuhi bentuk normal kesatu (1NF) bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data. Pada Tabel 3.2 tidak terjadi perubahan tabel.

Transaksi
id_checkout
id_konfirmasi
id_itemcheckout
id_produk_detail
id_member
kode
harga
jumlah
tanggal
jam
id_bang
total
grand_total
status
bukti
alamat_pengiriman
kode_post

Tabel 3. 2 Bentuk Normalisasi Pertama (1NF)

b. Bentuk Normalisasi Kedua (2NF)

Suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu, dan atribut yang bukan key sudah tergantung penuh terhadap keynya. Akan tetapi masih memiliki anomali yang secara praktis tidak dapat di terima, dapat dilihat pada tabel 3.3.

Item_Checkout	Checkout
id_item_checkout id_produk_detail id_checkout kode harga jumlah	id_checkout id_konfirmasi id_member kode tanggal jam id_bang total grand_total status bukti alamat_pengiriman kode_post

Tabel 3. 3 Bentuk Normalisasi Kedua (2NF)

c. Bentuk Normalisasi Ketiga (3NF)

Suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap keynya dan sudah tidak memiliki anomali. Pada Tabel 3.4 sudah terbentuk masing – masing tabel dengan keynya dan sudah di sesuaikan kedalam tabel masing masing.

Item_Checkout	Checkout	Konfirmasi
id_item_checkout id_produk_detail id_checkout kode harga jumlah	id_checkout id_member kode tanggal jam id_bang total grand_total status alamat_pengiriman kode_post	id_konfirmasi id_checkout id_member tanggal bukti

Tabel 3. 4 Bentuk Normalisasi Ketiga (3NF)

3.7.2.2. Struktur Data

Struktur data merupakan struktur spesifik *database* yang menunjukkan panjang elemen data disertai jenis data dalam *database*. Urutan isi atau data item merupakan urutan dalam struktur data dalam *website*, dijelaskan dalam tabel berikut ini.

Tabel 3. 5 Tabel Member

No	Nama	Tipe	Null	Extra
1	Id_member	Int(10)	No	Auto Increment
2	Nama	Varchar(30)	No	
3	Alamat	Varchar(50)	No	
4	No_telp	Varchar(15)	No	
5	Username	Varchar(15)	No	
6	Password	Varchar(15)	No	
7	Gambar	Varchar(40)	No	

Tabel 3. 6 Tabel Produk

No	Nama	Type	Null	Extra
1	Kode	Int(10)	No	Auto Increment
2	Nama	Varchar(30)	No	
3	Jenis	Varchar(10)	No	
4	Harga	Int(10)	No	
5	Keterangan	Varchar(500)	No	
6	Gambar	Varchar(40)	No	

Tabel 3. 7 Tabel Konfirmasi

No	Nama	Type	Null	Extra
1	Id_kon	Int(5)	No	Auto Increment
2	Id_checkout	Int(5)	No	
3	Id_member	Int(10)	No	
4	Tanggal	Date	No	
5	Bukti_transfer	Varchar(50)	No	

Tabel 3. 9 Tabel Item Checkout

No	Nama	Type	Null	Extra
1	Id_item_checkout	Int(5)	No	Auto Increment
2	Id_checkout	Int(5)	No	
3	Kode	Double	No	
4	harga	Int(5)	No	
5	Jumlah	Int(5)	No	
6	Id_produk_detail	Int(5)	No	

Tabel 3. 8 Tabel Detail Produk

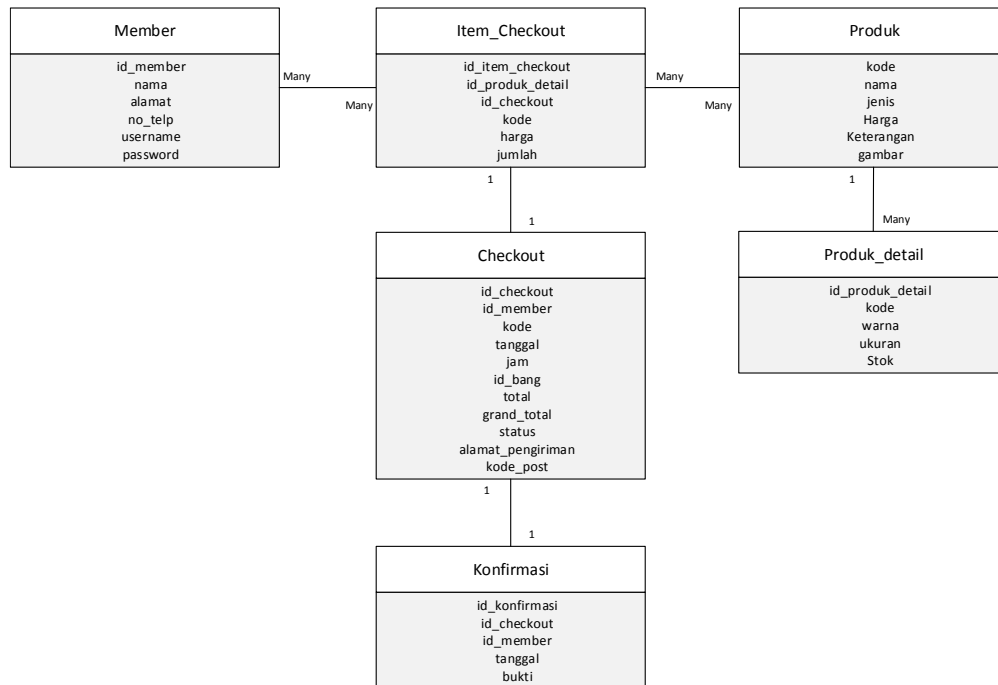
No	Nama	Type	Null	Extra
1	Id_detail_produk	Int(11)	No	Auto Increment
2	kode	Int(10)	No	
3	ukuran	Int(11)	No	
4	Stok	Stok(5)	No	
5	Warna	Varchar(25)	No	

Tabel 3. 10 Tabel Checkout

No	Nama	Type	Null	Extra
1	Id_checkout	Int(5)	No	Auto Increment
2	Id_member	Int(5)	No	
3	Kode_unik	Int(5)	No	
4	Tanggal	Date	No	
5	Jam	Time	No	
6	Id_bank	Int(5)	No	
7	Total	Double	No	
8	Grand_total	Double	No	
9	Status	Varchar(25)	No	
10	Alamat_pengiriman	Varchar(100)	No	
11	Kode_post	Int(11)	No	

3.7.2.3. Relasi Antar Tabel

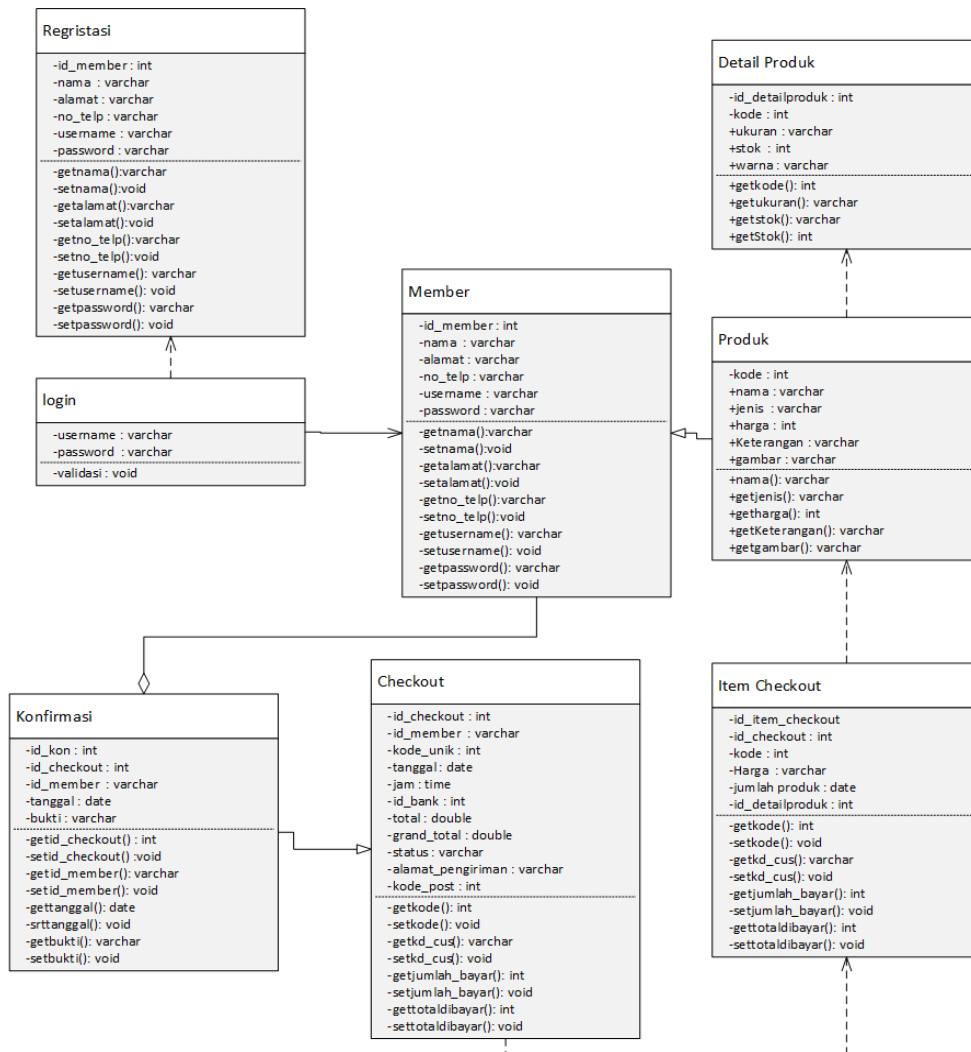
Setelah dilakukan konversi ERD kedalam tabel maka akan didapatkan relasi antar tabel yang dapat dilihat paa gambar 3.9 sebagai berikut:



Gambar 3. 9 Relasi Antar Tabel

3.8. Class Diagram

Berdasarkan analisis kebutuhan dibuatlah *class* diagram untuk membantu perancangan *website e-commerce* MAXCloth. Berikut adalah *class* diagram pada Gambar 3.10.



Gambar 3. 10 Class diagram

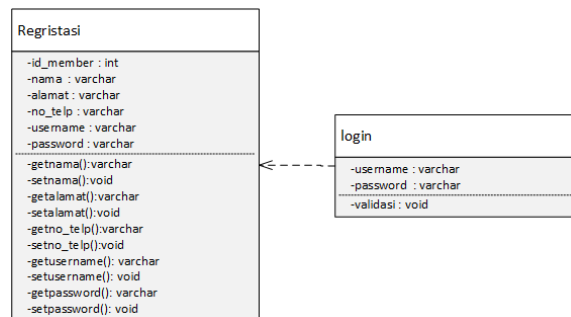
Berikut adalah penjelasan *class* diagram pada gambar 3.10 :

- Pada class *register*, memiliki fungsi untuk mendaftar sebagai *member* untuk bisa *login* kedalam *website*.
- Pada class *login*, memiliki fungsi untuk masuk sebagai *member* pada *website*.
- Pada class *member*, memiliki fungsi untuk melihat produk, memilih produk, melakukan order, dan melakukan transaksi.

- d. Pada *class* produk, memiliki fungsi sebagai produk yang dijual
- e. Pada *class* detail produk, memiliki fungsi menampung data dari ukuran, stok, dan juga warna dari produk.
- f. Pada *class item checkout*, memiliki fungsi untuk menampung data dari banyaknya checkout.
- g. Pada *class checkout*, memiliki fungsi untuk proses pemesanan yang berisi *detail* pesanan dan data pengiriman.
- h. Pada *class* konfirmasi, memiliki fungsi untuk melanjutkan ke tahap konfirmasi setelah melakukan checkout.

Pada gambar 3.10 mempunyai beberapa *class* yang saling berhubungan. Kelas-kelas yang saling berhubungan memiliki arti antara hubungan yang satu dengan yang lain. Kelas-kelas yang berelasi antara lain:

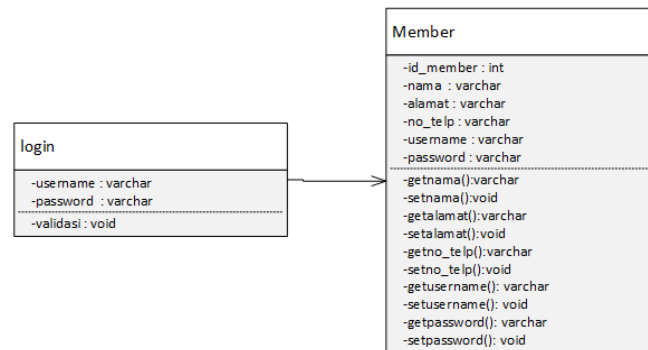
- a. Relasi antara *class login* dan *class* registrasi.



Gambar 3.11 Relasi antara *class login* dan *class* registrasi

Pada Gambar 3.11 menjelaskan relasi antara *class login* dengan *class* registrasi dihubungkan dengan relasi *dependency* atau dalam bahasa indonesianya relasi dependensi merupakan Relasi antar kelas dengan makna ketergantungan antar kelas. Jadi ketika pengguna akan melakukan *login*, maka pengguna tersebut harus terdaftar terlebih dahulu yaitu dengan mendaftarkan dirinya pada *class* registrasi untuk mendapatkan *username* dan *password* untuk *login*.

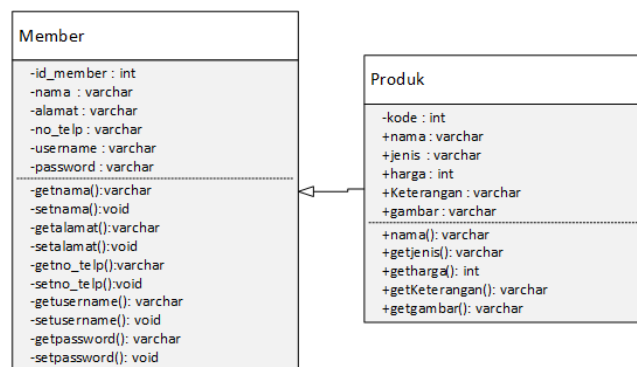
b. Relasi antara *class login* dan *class member*.



Gambar 3. 12 Relasi antara *class login* dan *class member*

Pada Gambar 3.12 menjelaskan relasi antara *class login* dan *class member* dihubungkan dengan relasi *Directed Association* yaitu relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain. *Class login* digunakan untuk login oleh *member* dengan proses *validasi*.

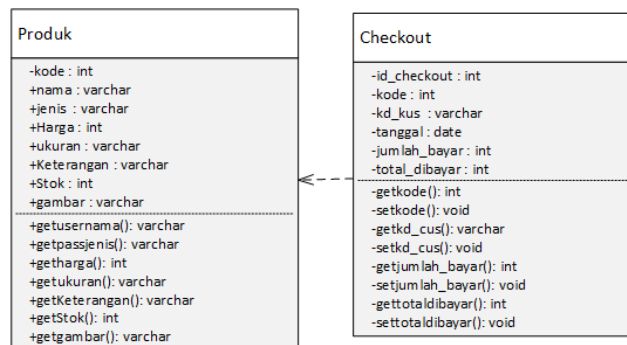
c. Relasi antara *member* dan produk.



Gambar 3. 13 Relasi antara *class produk* dan *class member*

Pada Gambar 3.13 menjelaskan relasi antara *class member* dengan *class produk* terhubung dengan relasi generalisasi. Produk akan bisa terbuka dan dapat memiliki akses lebih dalam memilih produk sampai checkout ketika member telah melakukan *login* dan sukses jika posisinya adalah pengguna masuk langsung sebagai *member*.

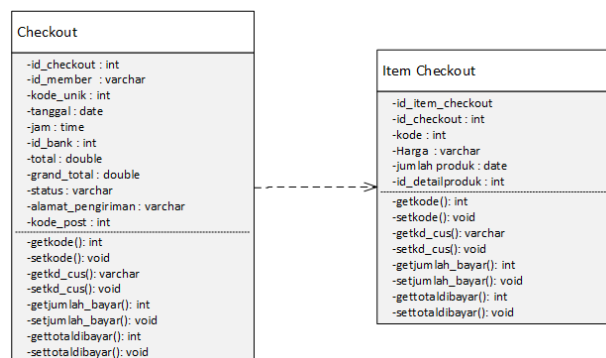
d. Relasi antara *class* produk dan *class item checkout*



Gambar 3. 14 Relasi antara *class* produk dan *class checkout*

Pada Gambar 3.14 menjelaskan relasi antara *class* produk dengan *class item checkout* dihubungkan dengan relasi *dependency* atau dalam bahasa indonesiaanya relasi dependensi merupakan Relasi antar kelas dengan makna ketergantungan antar kelas. Jadi ketika produk tersebut tidak ada maka member tidak dapat masuk kedalam *item checkout*.

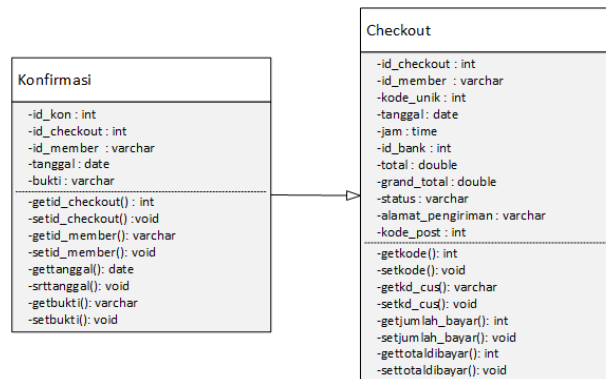
e. Relasi antara *class item checkout* dan *class checkout*



Gambar 3. 15 Relasi antara *class item checkout* dan *class checkout*

Pada Gambar 3.15 menjelaskan relasi antara *class item checkout* dengan *class checkout* dihubungkan dengan relasi *dependency* merupakan Relasi antar kelas dengan makna ketergantungan antar kelas. Jadi ketika *class item checkout* tersebut tidak ada maka member tidak dapat melakukan *checkout*.

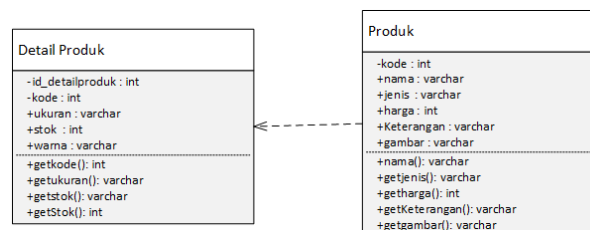
f. Relasi antara *class checkout* dan *class konfirmasi*



Gambar 3. 16 Relasi antara *class checkout* dan *class konfirmasi*

Pada Gambar 3.16 menjelaskan relasi antara *class checkout* dengan *class konfirmasi* terhubung dengan relasi generalisasi. *Konfirmasi* akan bisa dilakukan ketika member telah melakukan *checkout*.

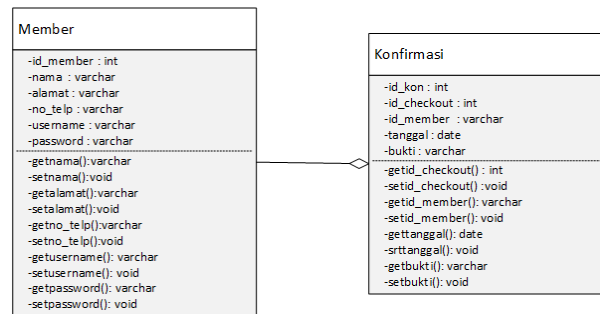
g. Relasi antara *class produk* dan *class detail produk*



Gambar 3. 17 Relasi antara *class checkout* dan *class konfirmasi*

Pada Gambar 3.17 menjelaskan Relasi antara *class produk* dengan *class detail produk* terhubung dengan relasi dependency. Jika detail produk tidak ada maka produk tersebut tidak bisa dibeli .

h. Relasi antara *class member* dan *class konfirmasi*



Gambar 3. 18 Relasi antara *class member* dan *class konfirmasi*

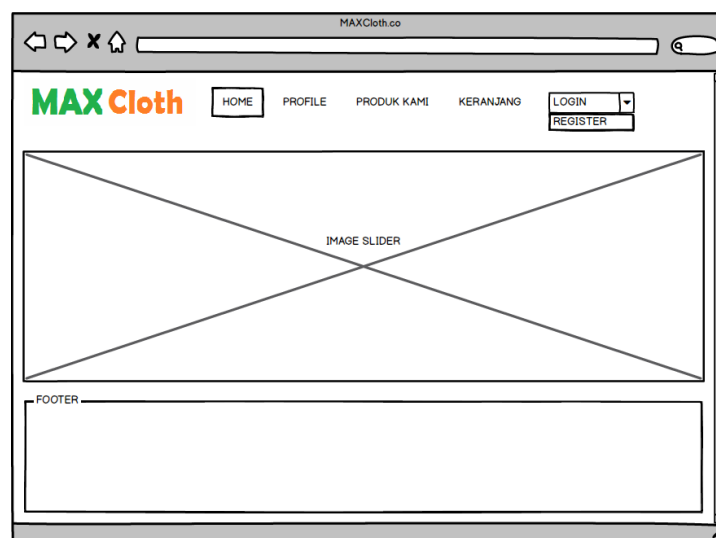
Pada Gambar 3.18 menjelaskan Relasi antara *class member* dengan *class konfirmasi* dihubungkan dengan relasi agregasi. *Class member* merupakan *main class* dari *subclass konfirmasi*.

3.9. Perancangan Interface

Perancangan *interface* berguna dalam pembuatan website dan perancangan tampilan yang akan dibuat menjadi *website*. *Interface* untuk sistem yang akan dibuat adalah sebagai berikut.

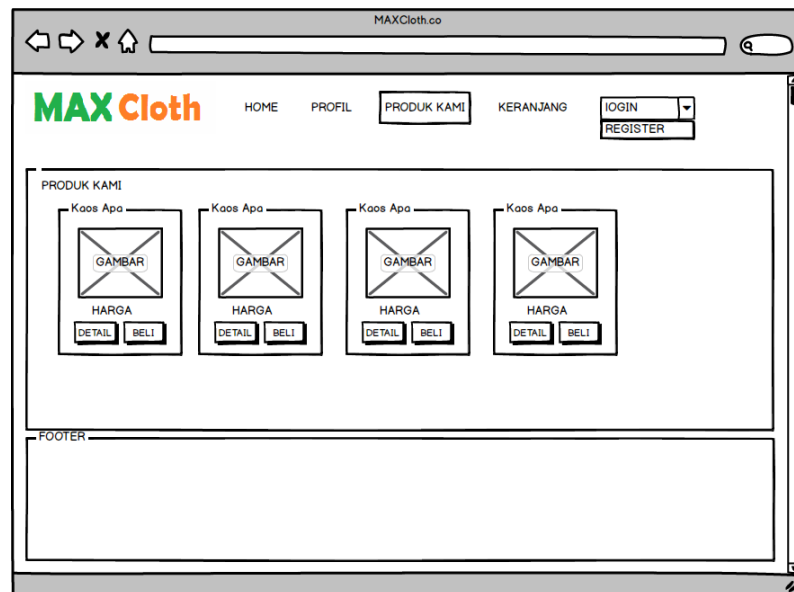
3.9.1. Rancangan Interface Halaman Utama

Halaman utama adalah halaman *website* yang terdiri dari beberapa pilihan kategori. Rancangan *interface* halaman utama dapat dilihat oleh *admin*, *member*, dan *non-member*.



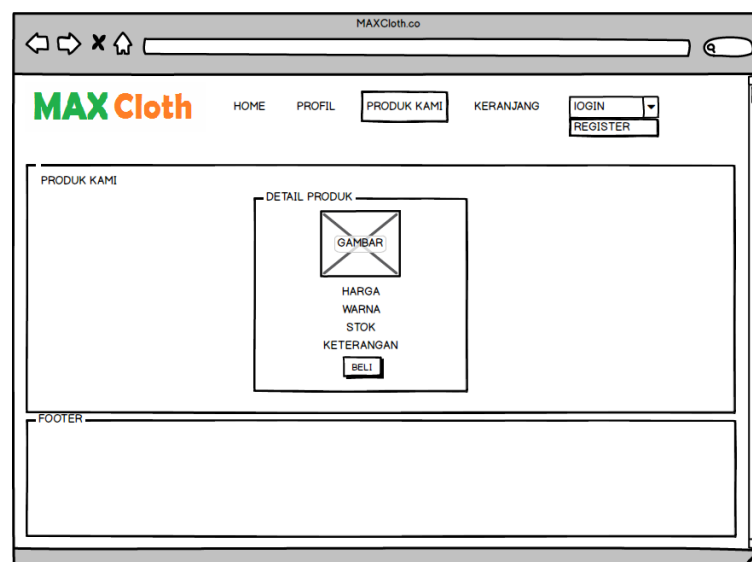
3.9.2. Rancangan Interface Halaman Produk

Rancangan *interface* halaman produk berisi tentang produk yang berkaitan dengan barang yang dijual. Pelanggan dapat melihat beberapa jenis produk baju beserta *detail*.



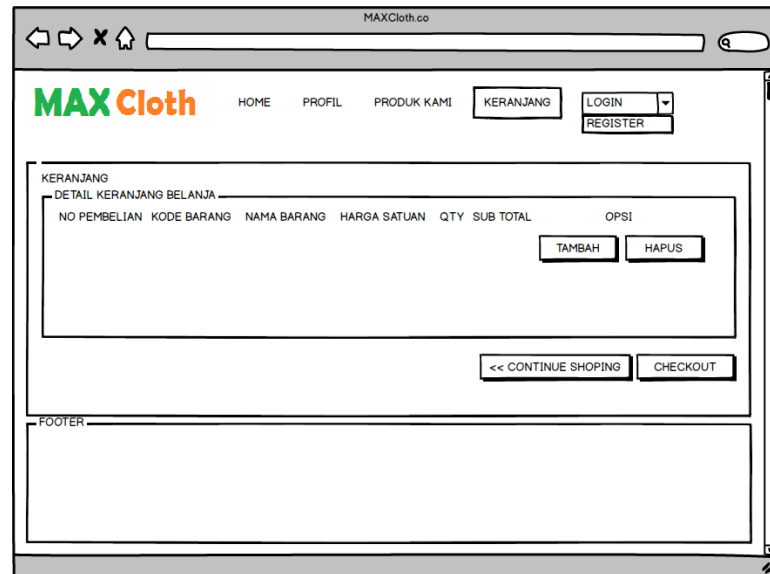
3.9.3. Rancangan Interface Halaman Detail Produk

Rancangan *interface* halaman *detail* produk berisi rincian informasi mengenai produk yang dijual. Gambaran rancangan *interface* halaman *detail* produk terdapat pada gambar berikut.



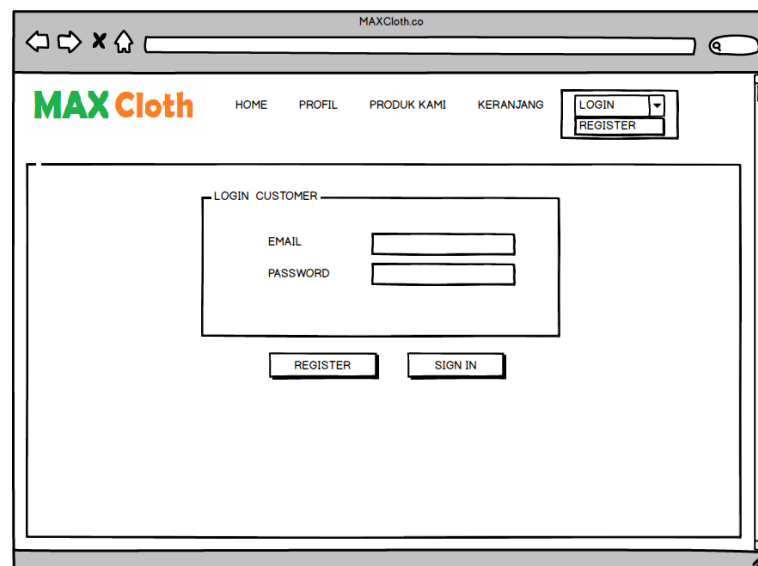
3.9.4. Rancangan *Interface* Halaman Keranjang Transaksi

Rancangan *interface* halaman *detail* pesanan terdapat data pesanan. *Member* dapat melakukan pemesanan baru atau langsung ke proses pembayaran dan *member* juga dapat menghapus pesanan. Gambaran rancangan *interface* halaman keranjang terdapat pada gambar berikut.



3.9.5. Rancangan *Interface* Halaman Login User

Rancangan *interface* halaman *login user* bertujuan agar user dapat masuk ke dalam website sebagai member dari *onlineshop* MAXCloth, dan mempermudah untuk proses pembayaran.



3.9.6. Rancangan *Interface* Halaman Register User

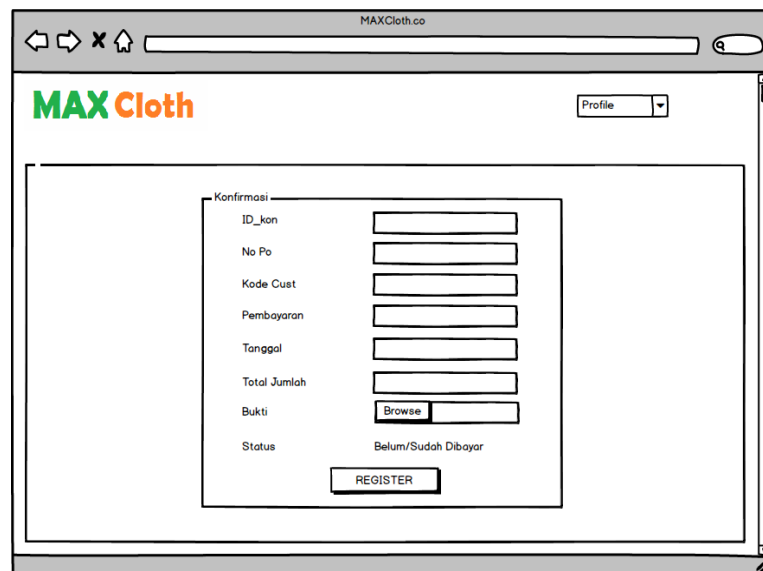
Rancangan *interface* halaman *register user* bertujuan agar *user* yang belum menjadi *member* mengisi data sebagai *member* dan dapat melakukan login.



The screenshot shows a web browser window with the URL 'MAXCloth.co'. The page features the 'MAX Cloth' logo and navigation links: HOME, PROFIL, PRODUK KAMI, and KERANJANG. A dropdown menu is open, showing 'LOGIN' and 'REGISTER' options. The main content area is titled 'REGISTER' and contains a form with the following fields: NAMA LENGKAP, ALAMAT LENGKAP, NO TELP, EMAIL, PASSWORD, and CONFIRM PASSWORD. A 'REGISTER' button is located at the bottom of the form.

3.9.7. Rancangan *Interface* Halaman Konfirmasi Transaksi

Rancangan *interface* halaman *Konfirmasi Transaksi* bertujuan agar *member* dapat melakukan konfirmasi dari barang yang telah di belinya.



The screenshot shows a web browser window with the URL 'MAXCloth.co'. The page features the 'MAX Cloth' logo and a 'Profile' dropdown menu. The main content area is titled 'Konfirmasi' and contains a form with the following fields: ID_kon, No Po, Kode Cust, Pembayaran, Tanggal, Total Jumlah, Bukti (with a 'Browse' button), and Status (with a dropdown menu showing 'Belum/Sudah Dibayar'). A 'REGISTER' button is located at the bottom of the form.