

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metode penelitian yang diterapkan pada perancangan dan pengembangan sistem dengan menggunakan algoritma Apriori dijelaskan sebagai berikut.

3.1.1 Perumusan Masalah

Perumusan masalah adalah tahap awal dalam penelitian yang melibatkan identifikasi dan penjelasan permasalahan yang akan diteliti. Dalam tahap ini, peneliti mencari permasalahan yang terdapat dari proses bisnis melalui wawancara yang telah dilakukan. Permasalahan yang terdapat adalah belum pernah ada sistem *product bundling* pada toko Jumbo dan pemilik toko Jumbo ingin meningkatkan angka penjualan dengan strategi baru, sehingga penelitian ini bertujuan untuk menciptakan sistem rekomendasi *product bundling* agar pemilik toko dapat meningkatkan efisiensi angka penjualan produk dan meningkatkan akurasi pemilihan produk saat melakukan *product bundling*.

3.1.2 Telaah Literatur

Tahapan ini meliputi kegiatan dalam mengumpulkan berbagai informasi yang berkaitan dengan penelitian seperti jurnal, buku, dan referensi. Informasi yang bersangkutan dengan algoritma Apriori, *market basket analysis*, *association rules*, dan *product bundling*

3.1.3 Studi Fisibilitas

Pada tahap ini dilakukan wawancara dengan pemilik toko dan karyawan toko untuk mendapatkan daftar transaksi yang pernah dilakukan dan daftar produk yang dijual oleh toko. Setelah diketahui toko tidak pernah mempunyai promo *bundling product* maka dibutuhkannya rekomendasi *bundling product*.

3.1.4 Analisis dan Perancangan

Dilakukan tahapan analisis dan perancangan proses data mining sebagai berikut.

A. Selection

Data dipilih dari transaksi tanggal 1 Juni 2022 sampai dengan 1 Juni 2023.

B. Pre-processing

Proses *cleaning* dari data yang telah pilih melalui proses *selection* data di cek kembali apakah data memiliki duplikat, *no value*.

C. Transformation

Setelah data dilakukan *cleaning*, data yang masih dalam bentuk nota dimasukkan ke dalam *database* MySQL dirancang menggunakan *Entity Relationship Diagram* (ERD), *database schema*, dan struktur tabel.

D. Data Mining

Data mining dilakukan dengan menggunakan algoritma Apriori yang digambarkan menggunakan *flowchart*. Lalu dilakukan perancangan Antarmuka sistem menggunakan *Balsamiq Cloud*.

E. Evaluation

Evaluasi dilakukan dengan *lift ratio* untuk menentukan apakah hasil *association rules* memenuhi standard nilai *lift ratio* lebih besar dari satu.

3.1.5 Implementasi

Sesuai dengan teori yang sudah dijabarkan sebelumnya, algoritma Apriori memiliki *frequent itemset* yang menjadi hasil akhir untuk digunakan dengan aturan asosiasi (*association rules*). Aturan asosiasi digunakan untuk mencari produk yang sering muncul bersamaan dalam suatu transaksi. Kemudian, implementasi dilakukan dengan bahasa pemrograman PHP, *Framework* CodeIgniter, dan *database* MySQL.

3.1.6 Pengujian Aplikasi

Pengujian aplikasi dilakukan dengan menggunakan perangkat Laptop Omen HP 15 dh0105tx menggunakan metode *White Box Testing*. *White Box Testing* diharapkan dapat membantu peneliti dalam memberikan hasil yang optimal dalam pengecekan fungsionalitas sistem.

3.1.7 Evaluasi

Pada bagian ini evaluasi akan dilakukan dengan *lift ratio* untuk mengevaluasi hasil dari penelitian sesuai dengan yang diharapkan atau tidak. Lalu akan dilakukan evaluasi sistem menggunakan *USE Questionnaire*.

3.1.8 Konsultasi dan Penulisan

Tahap ini mencakup dokumentasi proses-proses yang terlibat dalam penelitian dan menyimpulkan hasil akhir penelitian.

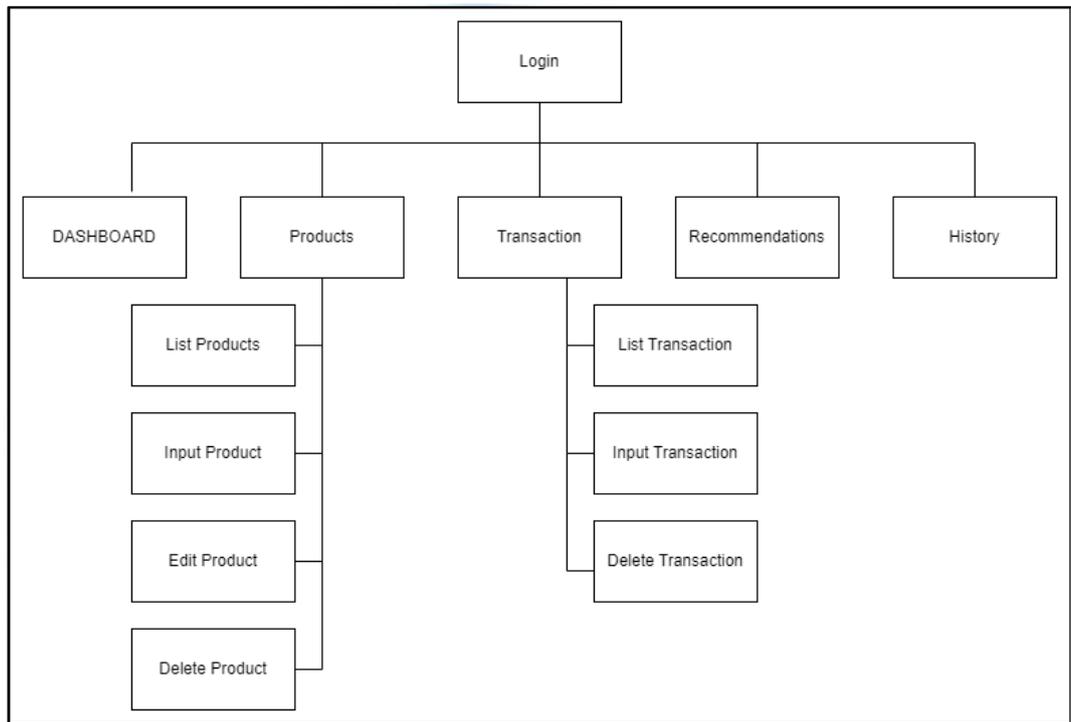
3.2 Perancangan Sistem

Bagian ini menjelaskan perancangan sistem dari penelitian ini. Perancangan sistem ini meliputi pembuatan *Flowchart*, *Database schema*, *Entity Relationship Diagram (ERD)*, *table structure*, dan desain antarmuka pengguna (*user interface*).

3.2.1 Sitemap

Perancangan sistem ini diawali dengan halaman login. Setelah user berhasil memasukkan username dan password, sistem akan mengarahkan user ke halaman dashboard. Pada halaman dashboard memiliki lima menu yang dapat dipilih untuk bernavigasi ke halaman lain seperti halaman *Dashboard*, *Products*, *Transaction*, *Recommendation*, dan *History*. Bentuk alur dari perancangan sistem dapat dilihat pada Gambar 3.1.

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



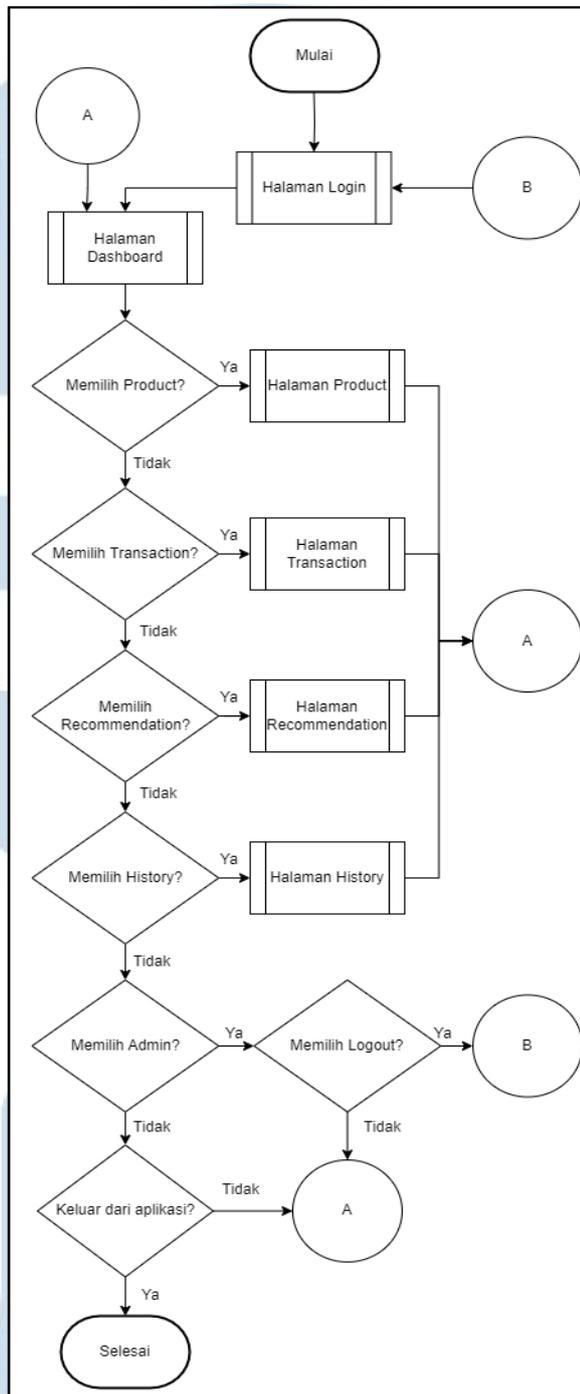
Gambar 3.1 Site Map

3.2.2 Flowchart

Flowchart adalah bentuk alur yang menunjukkan bagaimana program bekerja. Penjelasan *flowchart* sistem rekomendasi *product bundling* sebagai berikut.

A. Flowchart Utama

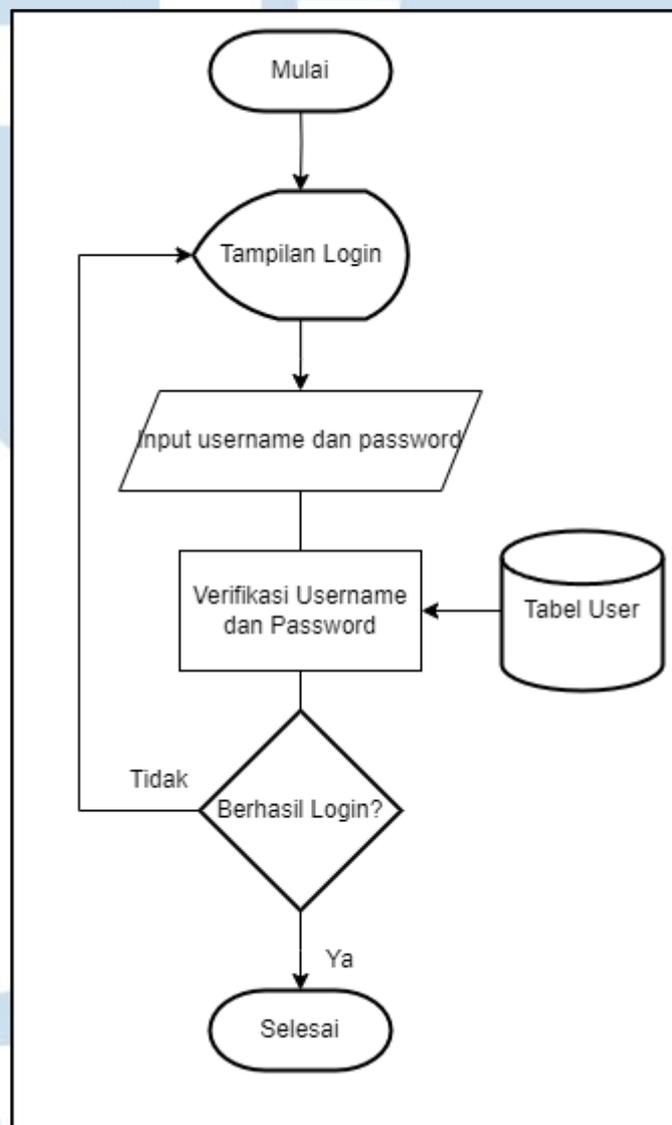
Flowchart Utama menjelaskan proses saat pengguna masuk ke dalam aplikasi. Jika proses login berhasil, maka *user* akan diarahkan ke halaman *Dashboard*. Pada halaman *Dashboard*, terdapat lima opsi untuk pindah halaman melalui *side navigation bar*, yaitu *Dashboard*, *Product*, *Transaction*, *Recommendation*, dan *History*. Selain itu, pada pojok kanan atas terdapat nama admin yang jika dipilih, akan muncul opsi untuk *logout*. Jika admin memilih *logout*, sistem akan mengarahkan *user* kembali ke halaman *login*. Bentuk detail dari *Flowchart* Utama dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart Utama

B. Flowchart Login

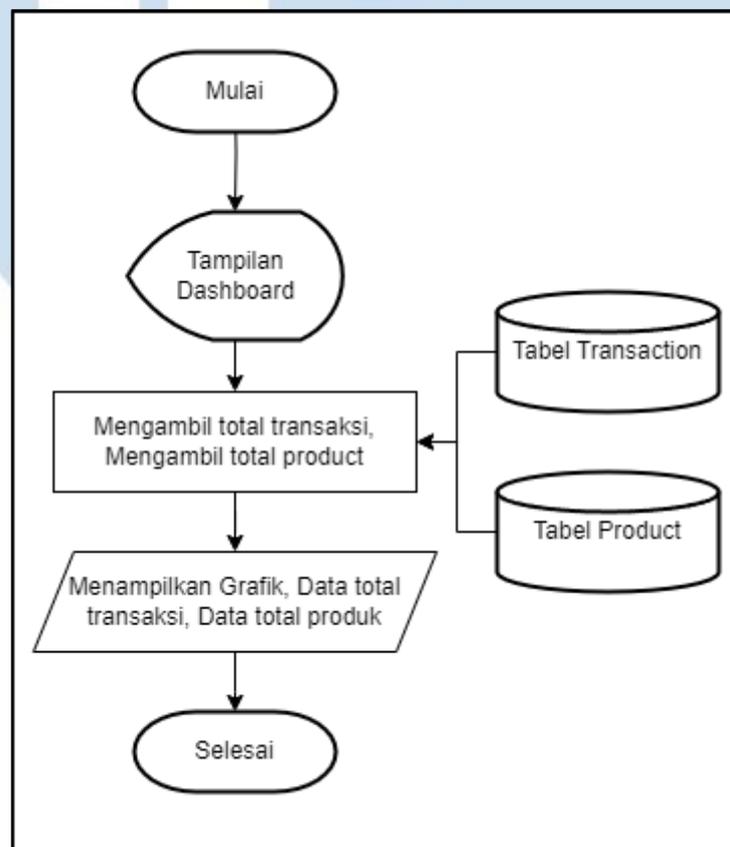
Flowchart Login menjelaskan awal halaman yang dimulai dengan adanya *card* yang berisikan dua elemen *input field* untuk kategori *username* dan *password*. Jika *login* gagal maka akan muncul error “Login Failed”. Apabila login berhasil dilakukan maka user akan diarahkan ke *Dashboard*. Bentuk detail dari *Flowchart Login* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Flowchart Login

C. Flowchart Dashboard

Flowchart Dashboard menjelaskan tentang tampilan pada halaman *Dashboard*. Pertama muncul tampilan untuk halaman *Dashboard*, setelah itu melakukan proses pengambilan data dari tabel *Transaction* dan *Product*. Lalu pada *dashboard* dimunculkan pada bagian kiri data total transaksi yang pernah dilakukan dan pada bagian kanan total produk yang dimiliki. Pada *dashboard* dibawah total transaksi dan total produk ada grafik berupa lima produk teratas yang paling banyak dibeli. Bentuk detail dari *Flowchart Dashboard* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Flowchart Dashboard

D. Flowchart Product

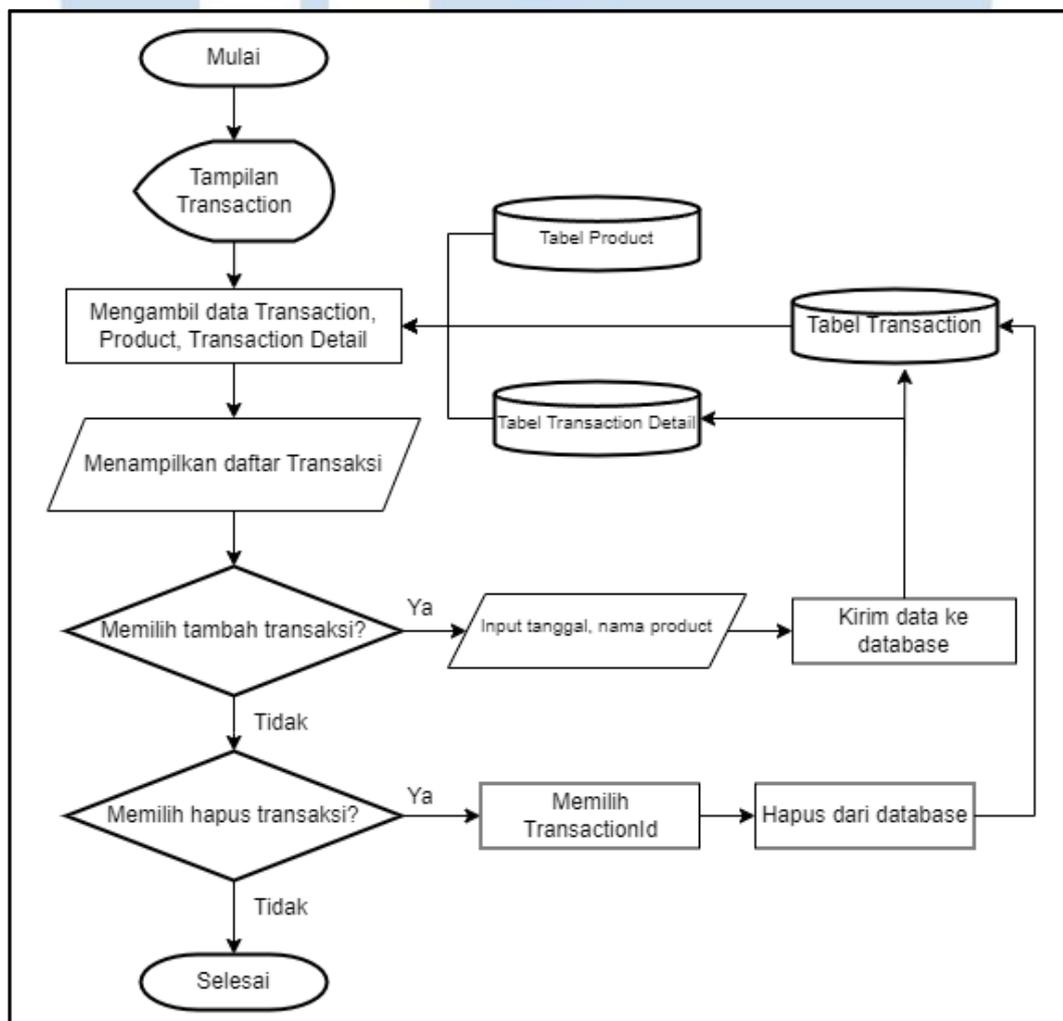
Flowchart Product menjelaskan tentang tampilan pada halaman *Product*. Pertama, dimulai dengan proses pengambilan data nama produk, harga, dan brand dari tabel *Product*. Lalu halaman *Product* menampilkan daftar produk yang dimiliki. Halaman *Product* juga memiliki tombol untuk menambahkan, edit, dan hapus produk yang ada didalam database. Bentuk detail dari *Flowchart Product* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart Product

E. Flowchart Transaction

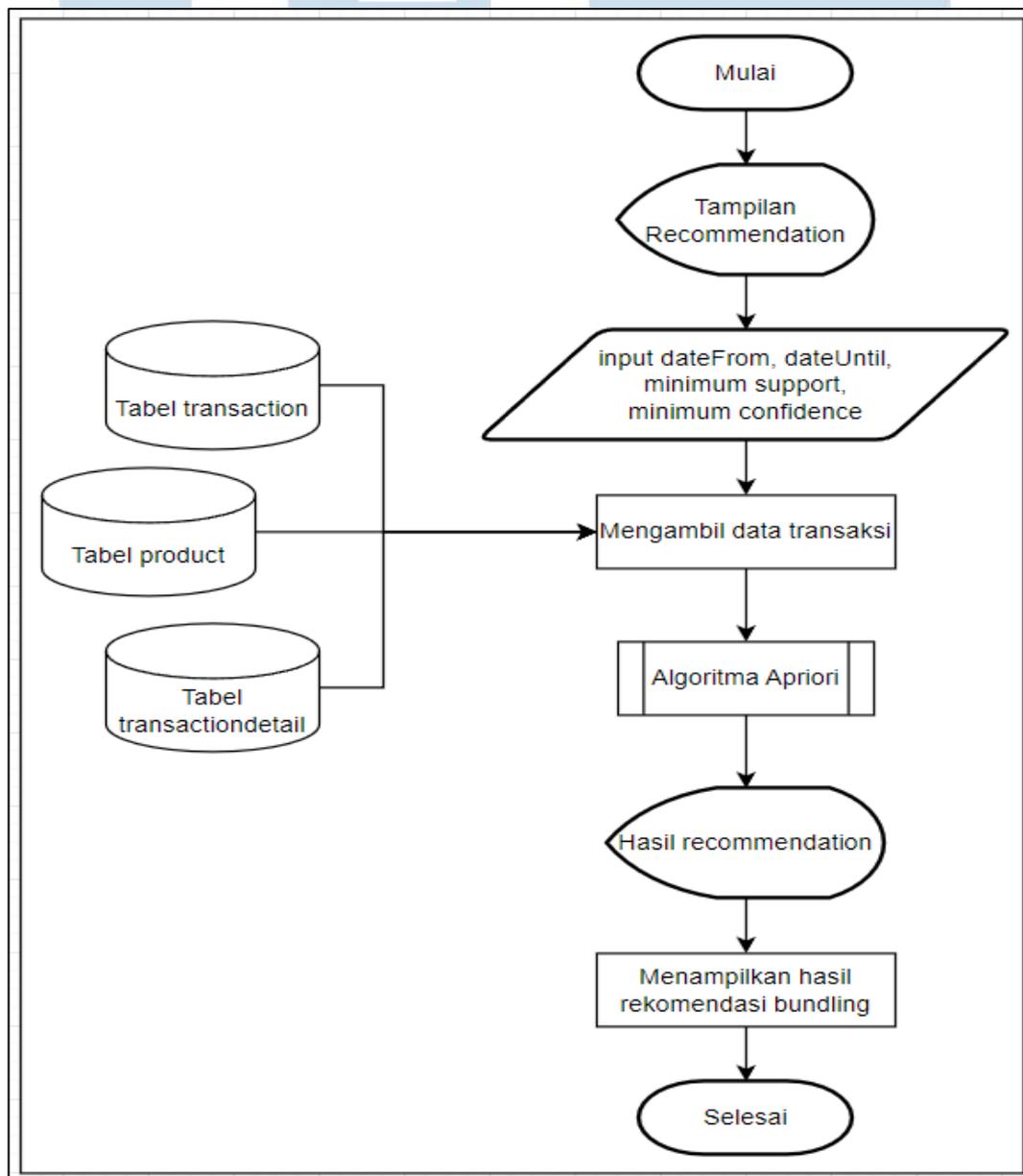
Flowchart Transaction menjelaskan tentang tampilan pada halaman *Transaction* yang dimulai dengan proses pengambilan data nama produk, id transaksi, dan tanggal transaksi dari tabel *Product*, *Transaction*, *Transactiondetail*. Lalu halaman *Transaction* akan menampilkan daftar transaksi yang pernah dilakukan. Halaman *Transaction* juga memiliki tombol untuk menambahkan, dan hapus transaksi yang ada didalam database. Bentuk detail dari *Flowchart Transaction* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Flowchart Transaction

F. Flowchart Recommendation

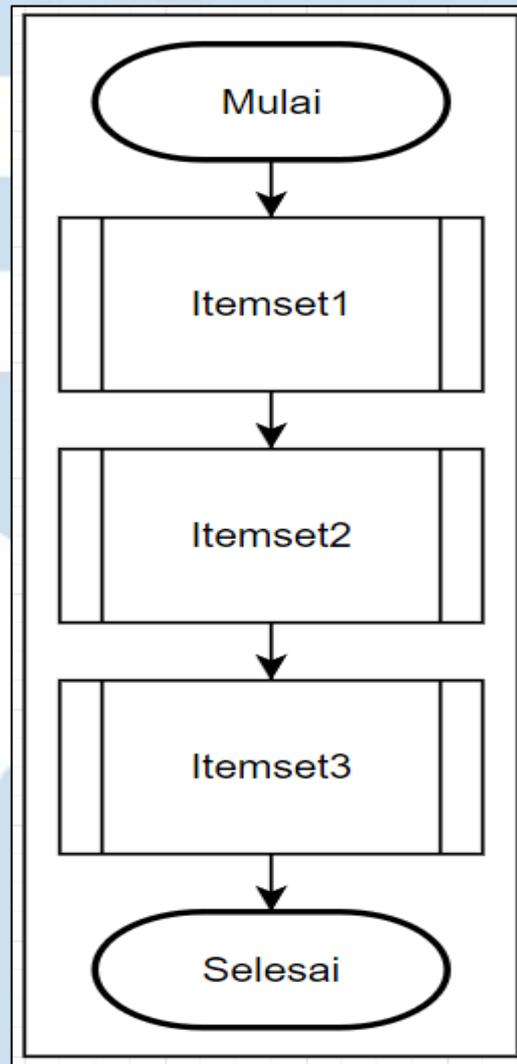
Flowchart Recommendation menjelaskan tentang tampilan pada halaman *Recommendation* yang dimulai dengan memilih *range* tanggal transaksi (*dateFrom*, *dateUntil*), *input minimum support*, dan *minimum confidence*. Kemudian dilanjutkan dengan proses algoritma Apriori yang akan menunjukkan hasil rekomendasi *product bundling*. Bentuk detail dari *Flowchart Recommendation* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Flowchart Recommendation

G. Flowchart Algoritma Apriori

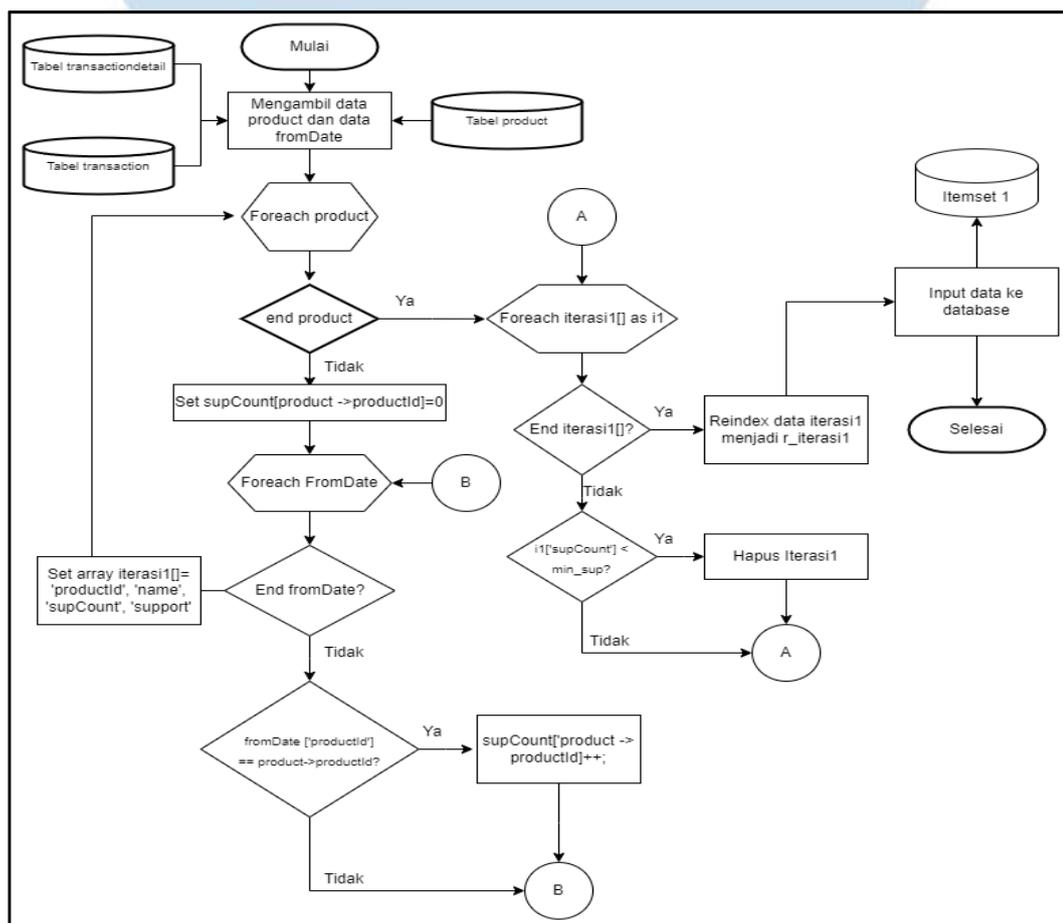
Flowchart Algoritma Apriori menjelaskan tentang proses pembentukan itemset1, itemset2, dan itemset3 menggunakan algoritma Apriori. Pada *Flowchart* Algoritma Apriori terdapat 3 modul yang berupa Itemset1, Itemset2, dan Itemset3. Bentuk detail dari Flowchart Algoritma Apriori dapat dilihat pada Gambar 3.8.



Gambar 3.8 Flowchart Algoritma Apriori

H. Flowchart Itemset1

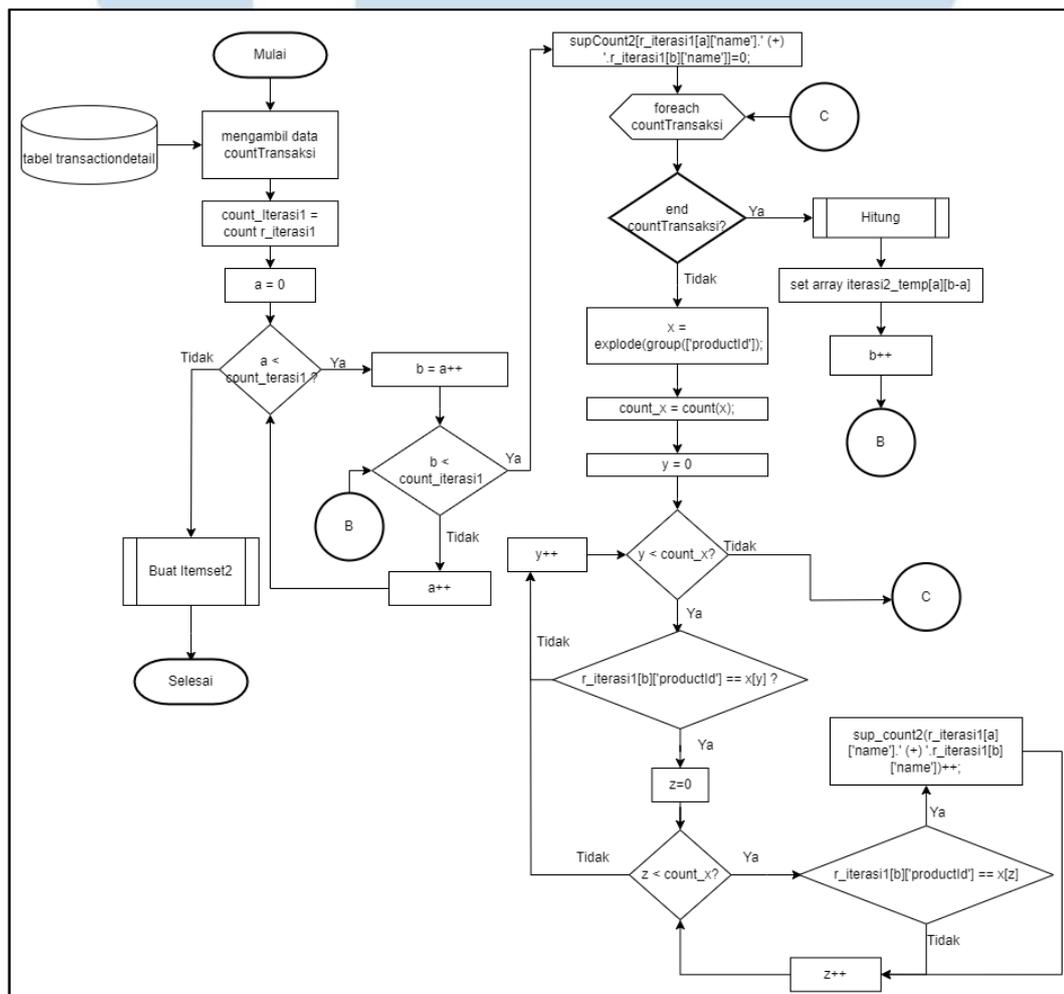
Flowchart Itemset1 menjelaskan tentang proses pembetulan itemset1. Langkah awal melibatkan pengambilan data dari tabel *product* dan data *input fromDate*. Proses ini melibatkan dua *loop*, yaitu *loop* pertama untuk data *product* dan *loop* kedua untuk data *fromDate*, dengan tujuan menghitung *support* dari satu kombinasi *product*. Setelah *loop* pada *fromDate* selesai, data akan dimasukkan ke dalam sebuah *array* yang dinamakan *iterasi1*. Jika *support* kurang dari *minimum support* yang telah ditentukan, data dalam *iterasi1* akan dihapus. Setelah mendapatkan nilai *support* yang memenuhi *minimum support*, proses selanjutnya *loop* untuk membuat *index array* yang lebih teratur. Selanjutnya, data akan disimpan ke dalam database *itemset1*. Bentuk detail dari Flowchart Itemset1 dapat dilihat pada Gambar 3.9.



Gambar 3.9 Flowchart Itemset1

I. Flowchart Itemset2

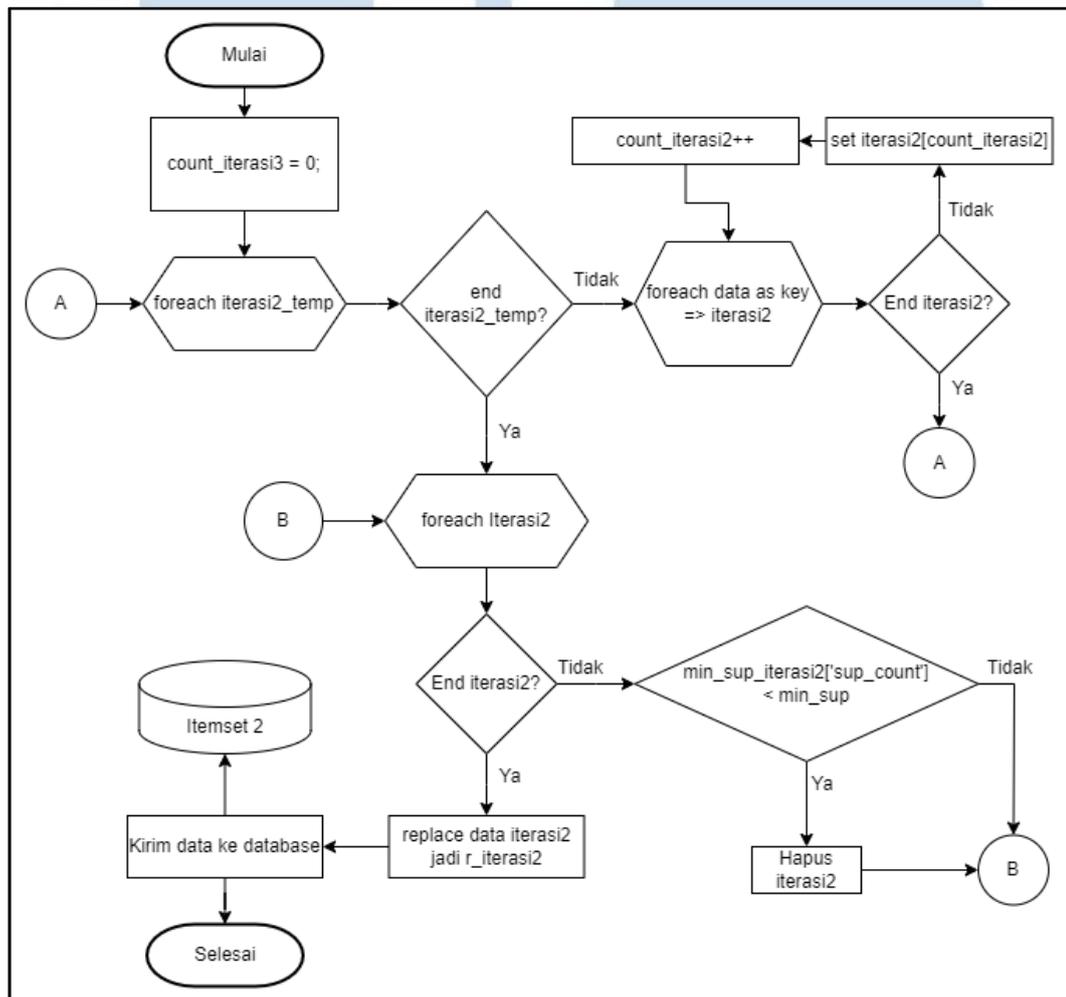
Flowchart Itemset2 menjelaskan tentang pembentukan itemset2 dengan cara mendapatkan data kombinasi dua *product* yang memiliki nilai *support* di atas *minimum support* yang telah ditentukan. Proses dimulai dengan mengambil data total transaksi dari tabel *transactiondetail*. Selanjutnya, data total transaksi dikirim ke satu *array*, kemudian dibandingkan dengan *productid* pada data itemset1 dari Gambar 3.9 Flowchart Itemset1. Setelah *loop* selesai, data akan masuk ke dalam proses Hitung. Selanjutnya, data akan dimasukkan ke dalam proses Buat Itemset2 untuk menghilangkan item yang memiliki nilai *support* kurang dari *minimum support* yang telah ditentukan. Bentuk detail dari Flowchart Itemset2 dapat dilihat pada Gambar 3.10.



Gambar 3.10 Flowchart Itemset2

J. Flowchart Buat Itemset2

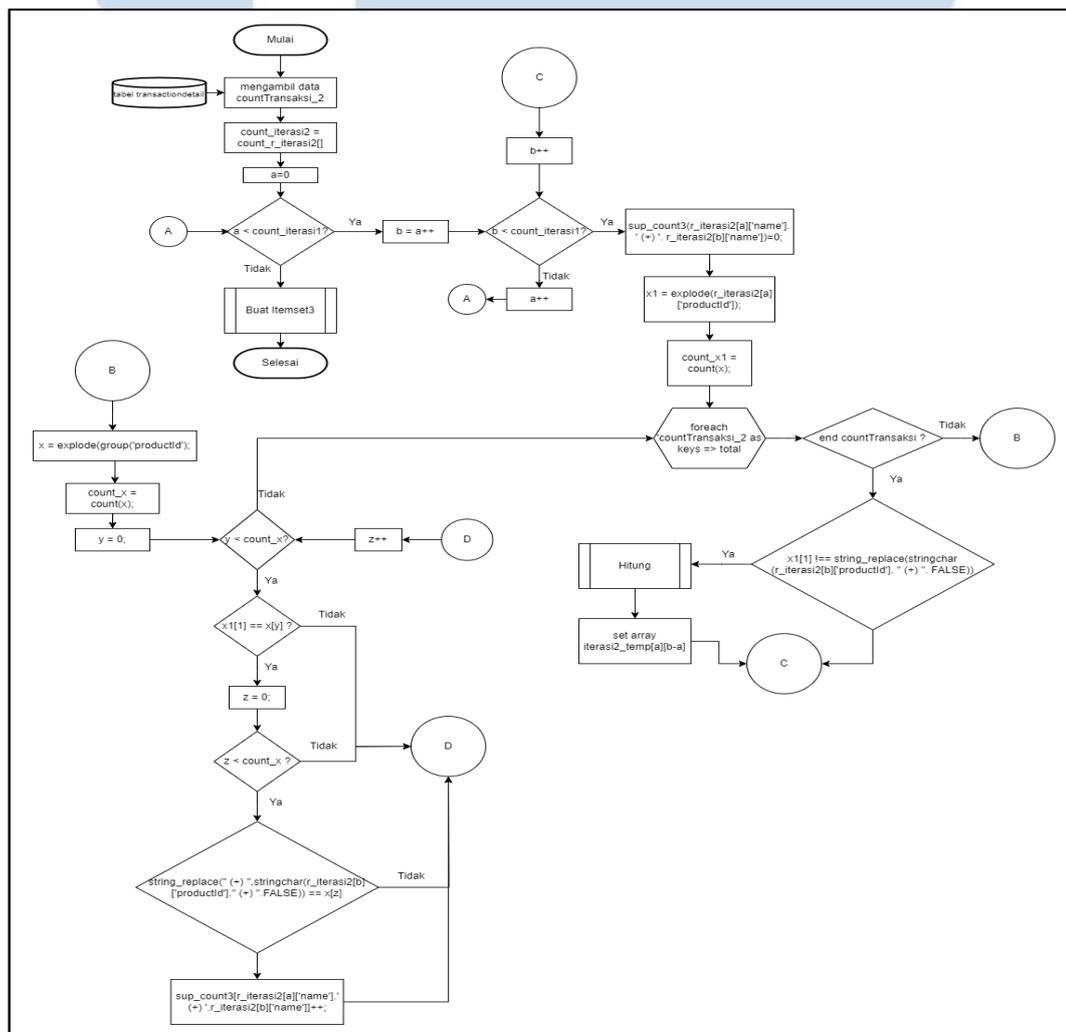
Flowchart Buat Itemset2 menjelaskan proses pembuatan data di ubah dari *iterasi2_temp* menjadi data untuk *itemset2* yang sudah di validasi memiliki nilai *support* lebih dari *minimum support*. Setelah proses validasi hampir selesai, data dikirim ke database *itemset 2*. Bentuk detail dari Flowchart Buat Itemset2 dapat dilihat pada Gambar 3.11.



Gambar 3.11 Flowchart Buat Itemset2

K. Flowchart Itemset3

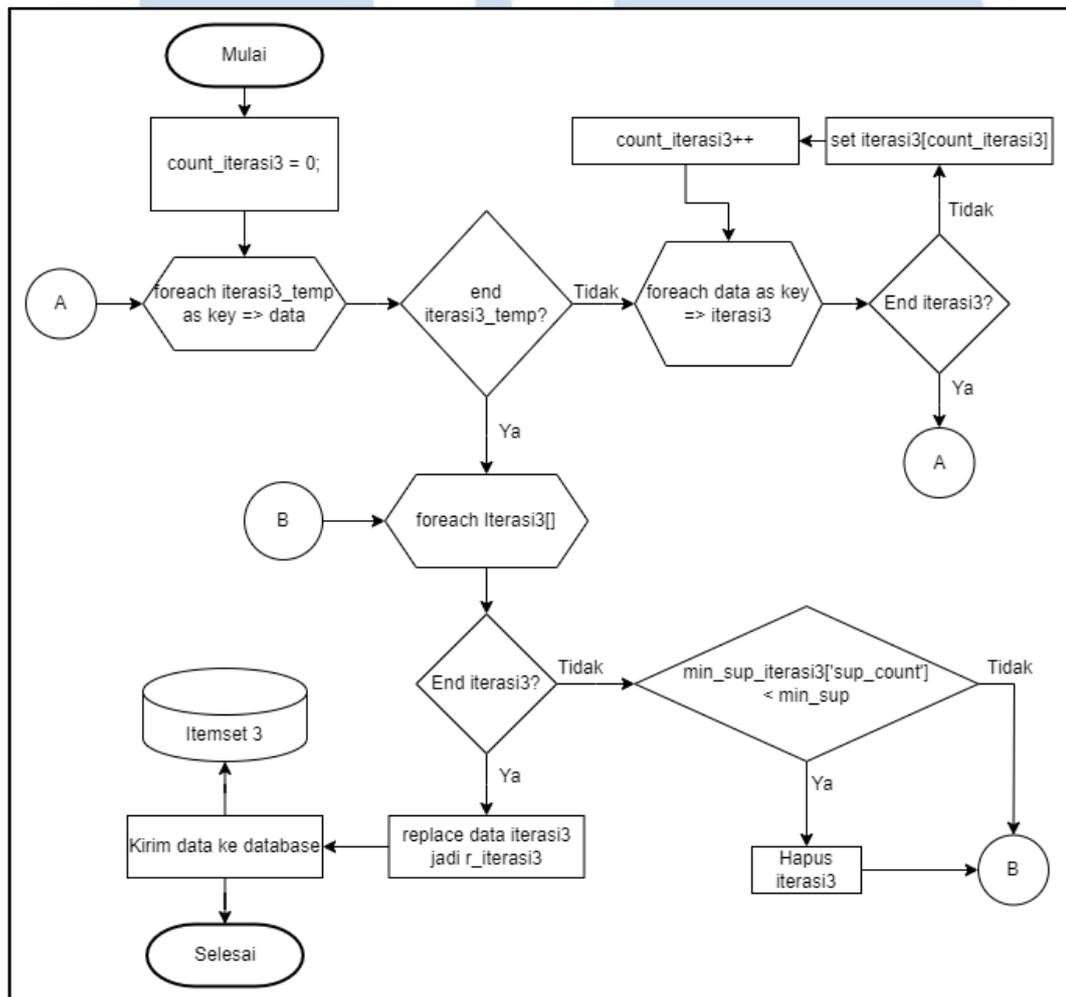
Flowchart Itemset3 menjelaskan tentang pembetulan data itemset 3. Proses ini mendapatkan tiga kombinasi produk yang memiliki nilai diatas *minimum support*. Proses diawali dengan pengambilan data dari tabel *transactiondetail* setelah itu, data total transaksi yang dimasukkan kedalam satu array setelah itu, dibandingkan *productid* dengan data itemset2. Hal ini bertujuan untuk menambahkan nilai *support count* untuk menghitung nilai *support* kombinasi item tersebut. Setelah *loop* berhasil dilakukan, data akan masuk ke proses Hitung. Setelah itu akan masuk kedalam proses Buat Itemset3 untuk melakukan penghapusan data iterasi3_temp yang memiliki nilai *support* kurang dari nilai *minimum support*. Bentuk detail dari Flowchart Itemset3 dapat dilihat pada Gambar 3.12.



Gambar 3.12 Flowchart Itemset3

L. Flowchart Buat Itemset3

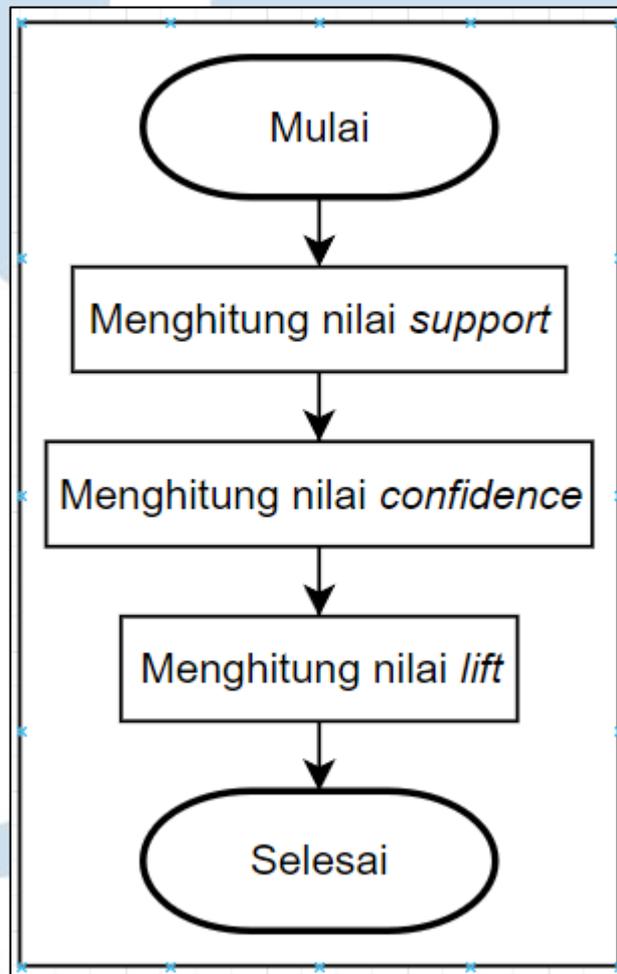
Flowchart Buat Itemset3 menjelaskan tentang pembuatan Itemset3. Pada proses ini data di ubah dari iterasi3_temp menjadi data untuk itemset2 yang sudah di validasi memiliki nilai *support* lebih dari *minimum support*. Setelah proses validasi hampir selesai, data dikirim ke database *itemset 3*. Bentuk detail dari Flowchart Buat Itemset3 Apriori dapat dilihat pada Gambar 3.13.



Gambar 3.13 Flowchart Buat Itemset3

M. Flowchart Hitung

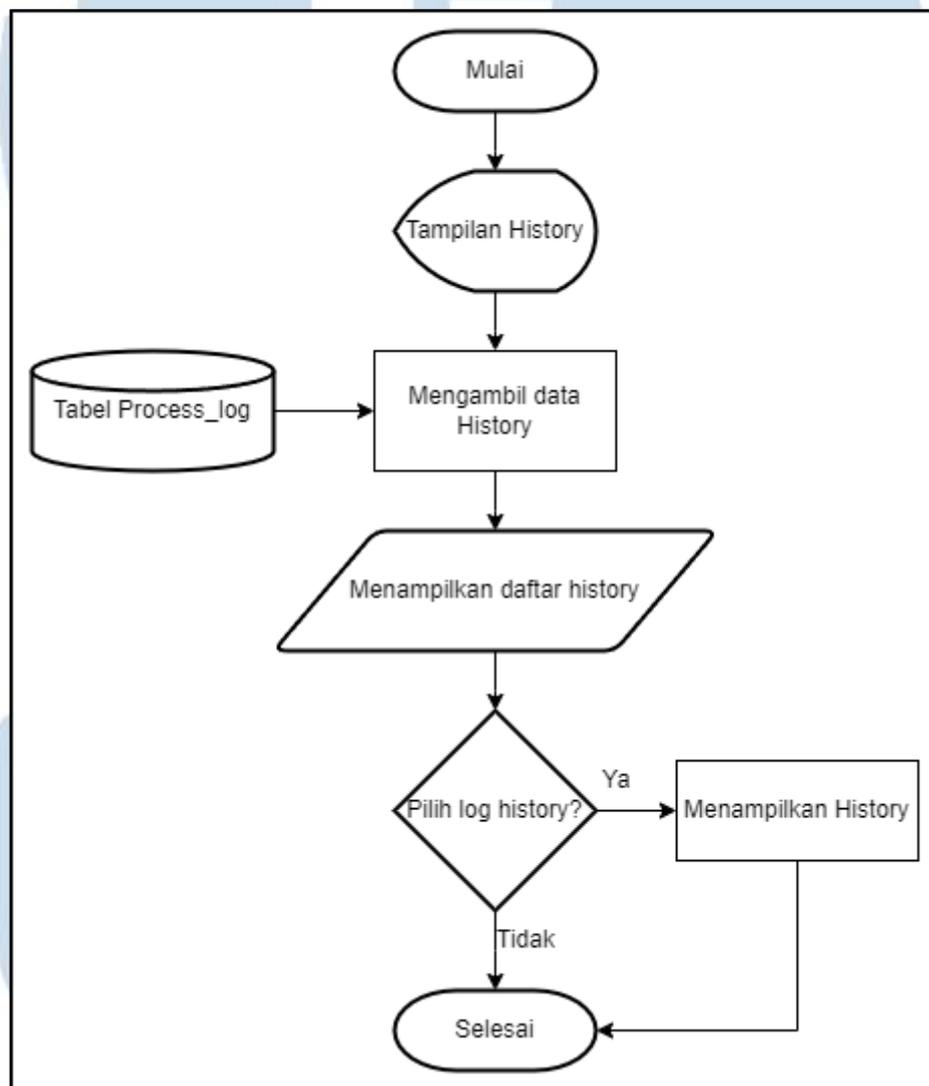
Flowchart Hitung menjelaskan tentang proses perhitungan yang digunakan untuk melakukan seleksi terhadap data yang telah dimasukkan ke dalam array. Untuk menghitung nilai *Support* diambil nilai total dari jumlah transaksi mengandung produk A dibagi total transaksi seperti pada Rumus 2.1. Selanjutnya untuk menghitung nilai *Confidence* diambil dari jumlah transaksi mengandung kombinasi produk A & B dapat dilihat pada Rumus 2.2. Setelah itu dilakukan perhitungan nilai *lift* dengan menggunakan Rumus 2.3. Bentuk detail dari Flowchart Hitung dapat dilihat pada Gambar 3.14.



Gambar 3.14 Flowchart Hitung

N. Flowchart History

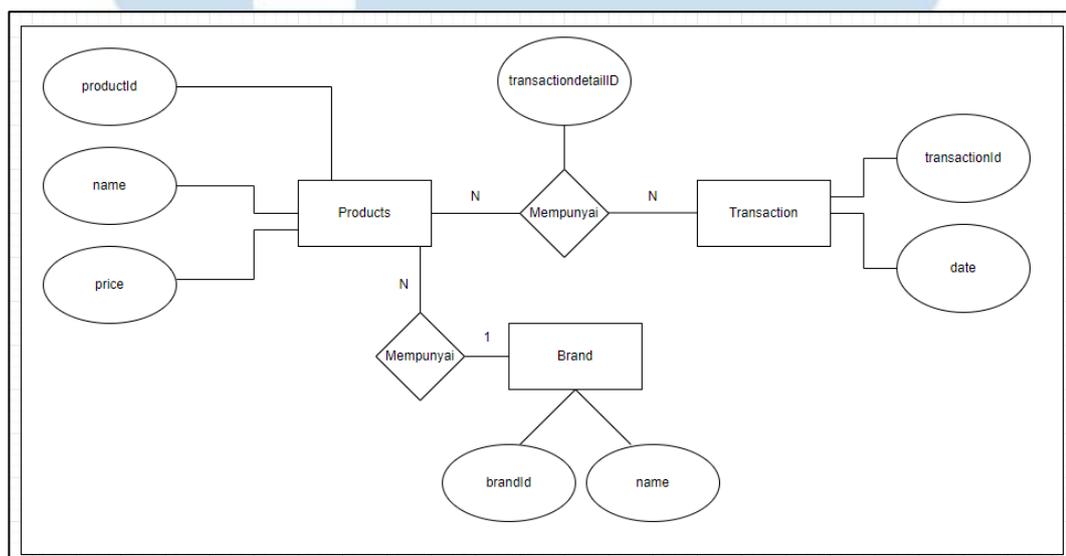
Flowchart History menjelaskan tentang tampilan pada halaman History dimulai dengan proses pengambilan data History dari tabel `process_log`. Lalu halaman *History* akan muncul tampilan daftar proses rekomendasi yang pernah dilakukan oleh *admin*. Jika tombol dipilih maka *admin* dapat melihat perincian history tertentu. Bentuk detail dari Flowchart History dapat dilihat pada Gambar 3.15.



Gambar 3.15 Flowchart History

3.2.3 Entity Relationship Diagram (ERD)

Pada bagian ini menjelaskan *Entity Relationship Diagram* (ERD) yang berisi empat tabel yang memiliki hubungan relasi antar tabel. Tabel *product* memiliki relasi *many to many* dengan tabel *Transaction* dan *many to one* dengan tabel *Brand*. Dikarenakan relasi tabel *transaction* adalah *many to many* sehingga banyak produk bisa memiliki banyak transaksi dengan memiliki *transactiondetailId*. Dan relasi tabel *product* dengan tabel *brand* adalah *many to one* dimana banyak produk hanya bisa memiliki satu *brand*. Bentuk dari *Entity Relationship Diagram* dapat dilihat pada Gambar 3.16.



Gambar 3.16 *Entity Relationship Diagram*

3.2.4 Database Schema

Berdasarkan *Entity Relationship Diagram* (ERD) pada Gambar 3.16 *Entity Relationship Diagram* dapat dijabarkan *database schema* yang menunjukkan relasi antar tabel. *Database schema* pada Gambar 3.17 menunjukkan sistem memiliki empat tabel yang memiliki relasi antar tabel, meliputi tabel *product*, *brand*, *transaction*, dan *transactiondetail*.



Gambar 3.17 Database Schema

3.2.5 Struktur Tabel

Struktur tabel pada Sistem Rekomendasi *Product Bundling* adalah sebagai berikut:

A. Tabel Brand

Tabel 3.1 merupakan struktur tabel dari *Brand*. Tabel ini mempunyai dua kolom yaitu: *brandId* sebagai *PRIMARY KEY*, dan *name*. Tabel ini berfungsi untuk menyimpan kode *brand* dan nama yang dimiliki suatu *brand*. Keterangan untuk tabel *brand* bisa dilihat pada Tabel 3.1

Tabel 3.1 Struktur Tabel Brand

Nama	Tipe Data	Keterangan
brandId	varchar(64)	PRIMARY KEY
name	varchar(64)	Nama dari brand

B. Tabel Product

Tabel 3.2 merupakan struktur dari tabel *Product*. Tabel ini mempunyai empat kolom yaitu: *productId* sebagai *PRIMARY KEY*, *name*, *price*, dan *brandId* sebagai *FOREIGN KEY*. Keterangan untuk tabel *Product* dapat dilihat pada Tabel 3.2

Tabel 3.2 Struktur Tabel Product

Nama	Tipe Data	Keterangan
productId	int(64)	PRIMARY KEY
name	varchar(64)	Nama dari produk
price	int(64)	Harga dari produk
brandId	varchar(64)	FOREIGN KEY

C. Tabel Transaction

Tabel 3.3 merupakan struktur dari tabel *transaction*. Tabel ini mempunyai dua kolom yaitu: *transactionId* sebagai *PRIMARY KEY*, dan *date*. Tabel ini berfungsi untuk menyimpan data transaksi. Keterangan untuk tabel *transaction* dapat bisa di Tabel 3.3

Tabel 3.3 Struktur Tabel Transaction

Nama	Tipe Data	Keterangan
<i>transactionId</i>	int(64)	<i>PRIMARY KEY</i>
<i>date</i>	date	Tanggal transaksi

D. Tabel Users

Tabel 3.4 merupakan struktur dari tabel User. Tabel ini mempunyai enam kolom. Kolom *id* sebagai *PRIMARY KEY*. Tabel ini berfungsi sebagai tempat menyimpan data *user admin*. Keterangan untuk tabel ini bisa dilihat di Tabel 3.4

Tabel 3.4 Struktur Tabel User

Nama	Tipe Data	Keterangan
id	int(11)	PRIMARY KEY
username	varchar(200)	Username admin
nama	varchar(200)	Nama admin
password	text	Password admin yang di enkripsi dengan md5
last_login	datetime	Mencatat kapan admin terakhir login
inactive	tinyint(4)	Mencatat apakah admin aktif atau tidak

E, Tabel Process_log

Tabel 3.5 merupakan struktur dari tabel *Process_log*. Tabel ini mempunyai sembilan kolom. Kolom *id* sebagai *PRIMARY KEY*. Tabel ini berfungsi sebagai tempat penyimpanan data riwayat rekomendasi. Keterangan untuk tabel ini bisa dilihat pada Tabel 3.5

Tabel 3.5 Struktur tabel Process_log

Nama	Tipe Data	Keterangan
id	int(11)	<i>PRIMARY KEY</i>
start_date	date	menampung nilai dari tanggal
end_date	date	menampung nilai sampai tanggal
min_support	double	menampung nilai <i>minimum support</i>
min_confidence	double	menampung nilai <i>minimum confidence</i>

F. Tabel Itemset1

Tabel 3.6 merupakan struktur dari tabel *Itemset1*. Tabel ini mempunyai lima kolom. Kolom *proc_id* adalah *PRIMARY KEY*. Tabel ini berfungsi sebagai tempat penyimpanan data yang memiliki nilai *support* lebih dari *input*. Keterangan untuk tabel ini bisa dilihat pada Tabel 3.6

Tabel 3.6 Struktur Tabel Itemset1

Nama	Tipe Data	Keterangan
Proc_id	int(11)	<i>PRIMARY KEY</i>
atribut	varchar(200)	menampung nama produk
jumlah	int(11)	menampung nilai total kemunculan produk
support	double	menampung nilai support produk
lolos	tinyint(4)	menampung validasi apabila itemset valid atau tidak

G. Tabel Itemset2

Tabel 3.7 merupakan struktur dari tabel Itemset2. Tabel ini mempunyai enam kolom. Kolom *proc_id* adalah *PRIMARY KEY*. Tabel ini berfungsi sebagai tempat penyimpanan data yang memiliki nilai *support* lebih dari *input*. Keterangan untuk tabel ini bisa dilihat pada Tabel 3.7

Tabel 3.7 Struktur Tabel Itemset2

Nama	Tipe Data	Keterangan
Proc_id	int(11)	PRIMARY KEY
atribut1	varchar(200)	menampung nama produk A
atribut2	varchar(200)	menampung nama produk B
jumlah	int(11)	menampung nilai total kemunculan produk
support	double	menampung nilai support
lolos	tinyint(4)	menampung validasi apabila itemset valid atau tidak

H. Tabel Itemset3

Tabel 3.8 merupakan struktur dari tabel Itemset3. Tabel ini mempunyai enam kolom. Kolom *proc_id* adalah *PRIMARY KEY*. Tabel ini berfungsi sebagai tempat penyimpanan data yang memiliki nilai *support* lebih dari *input*. Keterangan untuk tabel ini bisa dilihat pada Tabel 3.8

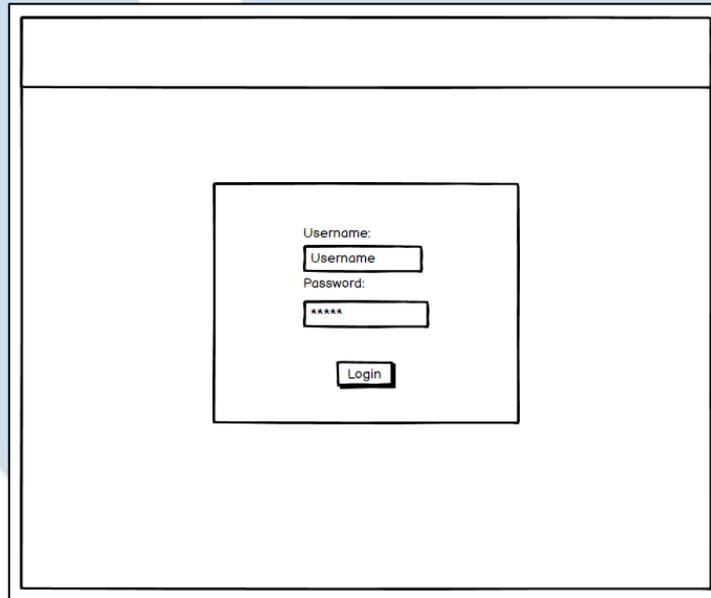
Tabel 3.8 Struktur Tabel Itemset3

Nama	Tipe Data	Keterangan
Proc_id	int(11)	PRIMARY KEY
atribut1	varchar(200)	menampung nama produk A
atribut2	varchar(200)	menampung nama produk B
jumlah	int(11)	menampung nilai total kemunculan produk
support	double	menampung nilai support
lolos	tinyint(4)	menampung validasi apabila itemset valid atau tidak

3.2.6 Rancangan Antarmuka

Bagian ini menjabarkan tentang bentuk rancangan antarmuka sistem rekomendasi *product bundling* berbasis web sebagai berikut.

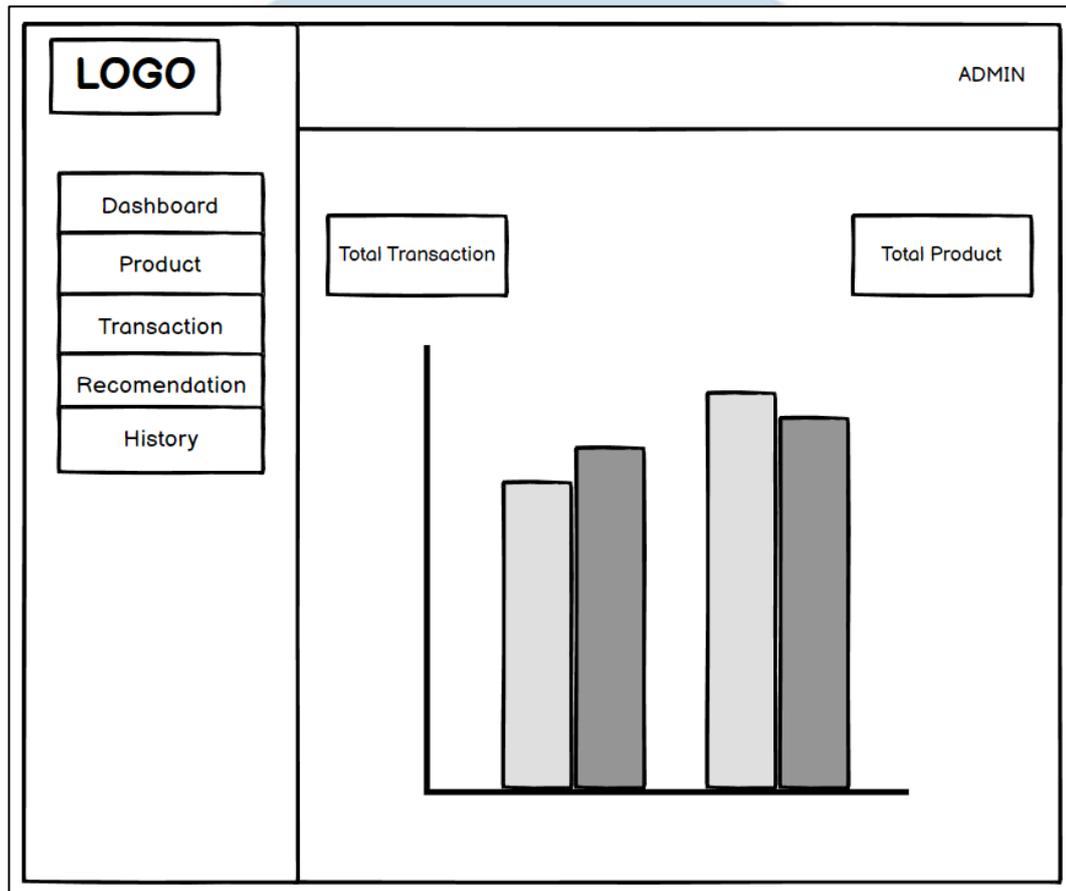
A. Antarmuka Halaman *Login*



Gambar 3.18 Rancangan Antarmuka Halaman *Login*

Gambar 3.18 adalah Rancangan Antarmuka Halaman *Login*. Ada dua kolom *input* yang berfungsi sebagai *input field* untuk *username* dan *password*. Lalu ada tombol *Login* yang berfungsi untuk validasi data *username* dan *password* yang di *input* sesuai dengan *database* atau tidak. Jika sesuai maka *user* akan diarahkan ke halaman *Dashboard*

B. Antarmuka Halaman *Dashboard*

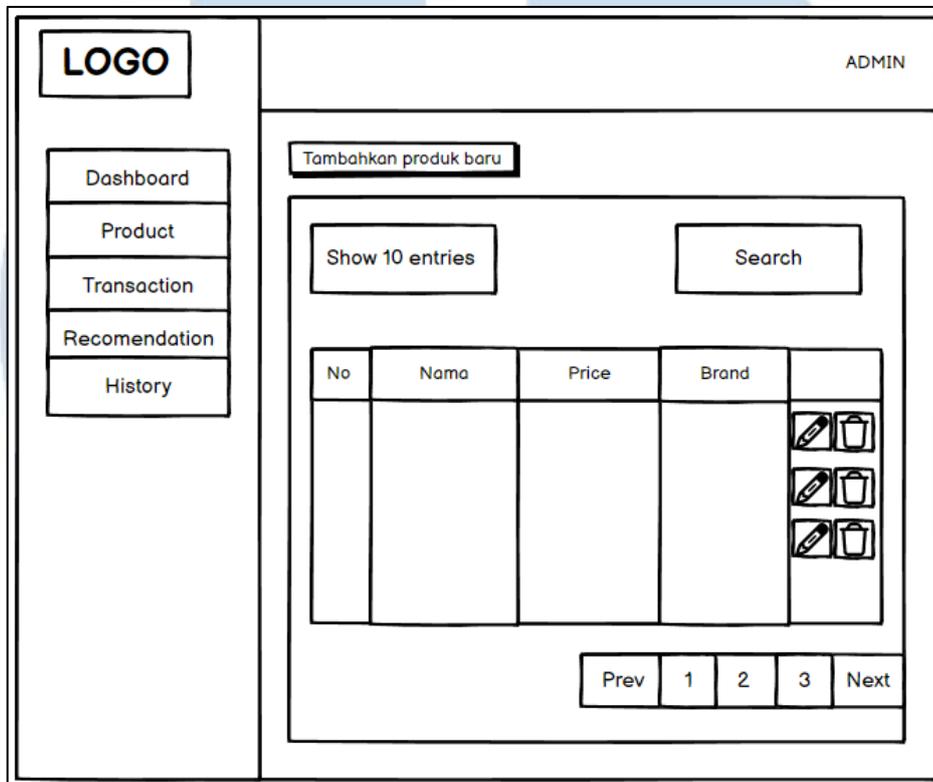


Gambar 3.19 Rancangan Antarmuka *Dashboard*

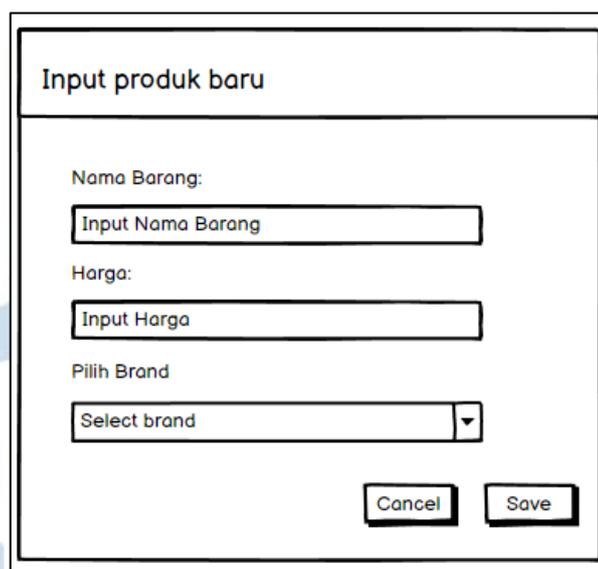
Gambar 3.19 adalah Rancangan Antarmuka *Dashboard*. Di bagian atas terdapat logo dari toko Jumbo, dibagian samping kiri terdapat menu navigasi *side navigation bar* untuk navigasi ke halaman *Dashboard*, *Product*, *Transaction*, *Recommendation*, *History*. Pada pojok kanan atas terdapat menu *admin* yang jika diklik akan muncul bar untuk keluar dari sistem dan kembali ke halaman *Login*. Di halaman *Dashboard* juga terdapat *card* yang menunjukkan total transaksi dan total produk. Serta ada grafik yang menunjukkan data lima produk yang paling sering terjual.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

C. Antarmuka Halaman *Product*



Gambar 3.20 Rancangan Antarmuka *Product*

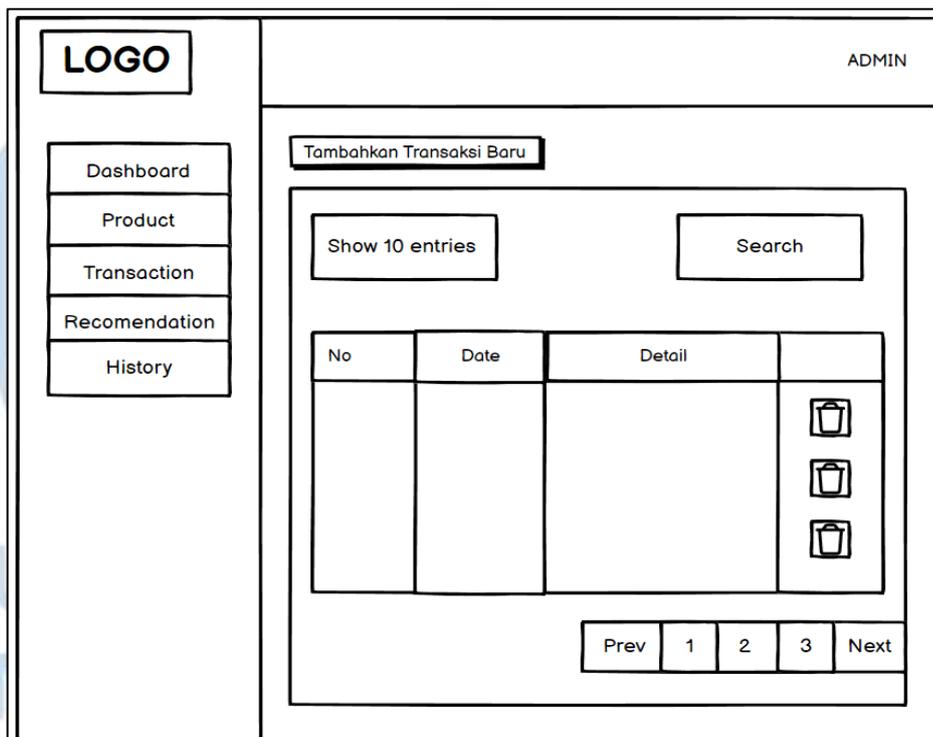


Gambar 3.21 Antarmuka *Modal Input Product*

Gambar 3.20 adalah Rancangan Antarmuka *Product*. Mulai dari bagian atas ada *header* dan pada bagian kiri ada *side navigation bar* untuk navigasi menu. Dibawah *header* terdapat tombol tambahkan produk baru yang akan memunculkan Antarmuka *Modal Input Product* seperti pada Gambar 3.21 Antarmuka *Modal Input Product*. Pada halaman ini juga menampilkan nama produk, harga produk, dan *brand* produk. Di samping itu, terdapat fungsi pencarian (*search*) untuk mencari produk, serta opsi *show entries* untuk menentukan jumlah *product* yang ingin ditampilkan. Dan ada juga tombol berbentuk *icon pencil* dan *trash can* yang berfungsi sebagai tombol *edit* dan *delete* produk yang dipilih.

Gambar 3.21 merupakan Antarmuka *Modal Input Product*. Di bagian ini terdapat *input box* untuk memasukkan nama produk, harga produk, dan *brand* produk. Di bagian bawah terdapat tombol *cancel* dan *save*. Apabila tombol *cancel* dipilih maka akan mengembalikan user ke halaman *product* dan apabila tombol *save* dipilih maka akan dilakukan proses *input data* ke dalam *database*.

D. Antarmuka Halaman *Transaction*



Gambar 3.22 Rancangan Antarmuka *Transaction*

The image shows a modal window with a white background and a black border. At the top, the title 'Input transaksi baru' is centered. Below the title, there are two input fields. The first is a text input field with the label 'Pilih tanggal transaksi:' and the placeholder text 'MM/DD/YYYY'. The second is a dropdown menu with the label 'Pilih produk:' and the placeholder text 'Pilih Produk'. At the bottom right of the modal, there are two buttons: 'Cancel' and 'Save'.

Gambar 3.23 Antarmuka Modal *Input Transaction*

Gambar 3.22 merupakan Rancangan Antarmuka *Transaction*. Di bagian atas terdapat *header* dan pada bagian kiri terdapat *side navigation bar* untuk navigasi menu. Dibawah *header* terdapat tombol tambahkan transaksi baru yang akan memunculkan Antarmuka *Modal Input Transaction* seperti pada Gambar 3.23 Antarmuka *Modal Input Transaction*. Pada halaman ini juga menampilkan tanggal transaksi dan detail transaksi. Di samping itu, terdapat fungsi pencarian (*search*) untuk mencari produk, serta opsi *show entries* untuk menentukan jumlah *transaction* yang ingin ditampilkan. Dan ada juga tombol berbentuk *icon trash can* yang berfungsi sebagai tombol *delete* transaksi yang dipilih.

Gambar 3.23 merupakan Antarmuka *Modal Input Transaction*. Di bagian ini terdapat *input box* untuk *input* tanggal transaksi dan *drop box* yang berisi nama produk. Di bagian bawah terdapat tombol *cancel* dan *save*. Apabila tombol *cancel* dipilih maka akan mengembalikan user ke halaman *transaction* dan apabila tombol *save* dipilih maka akan dilakukan proses *input* data ke dalam *database*.

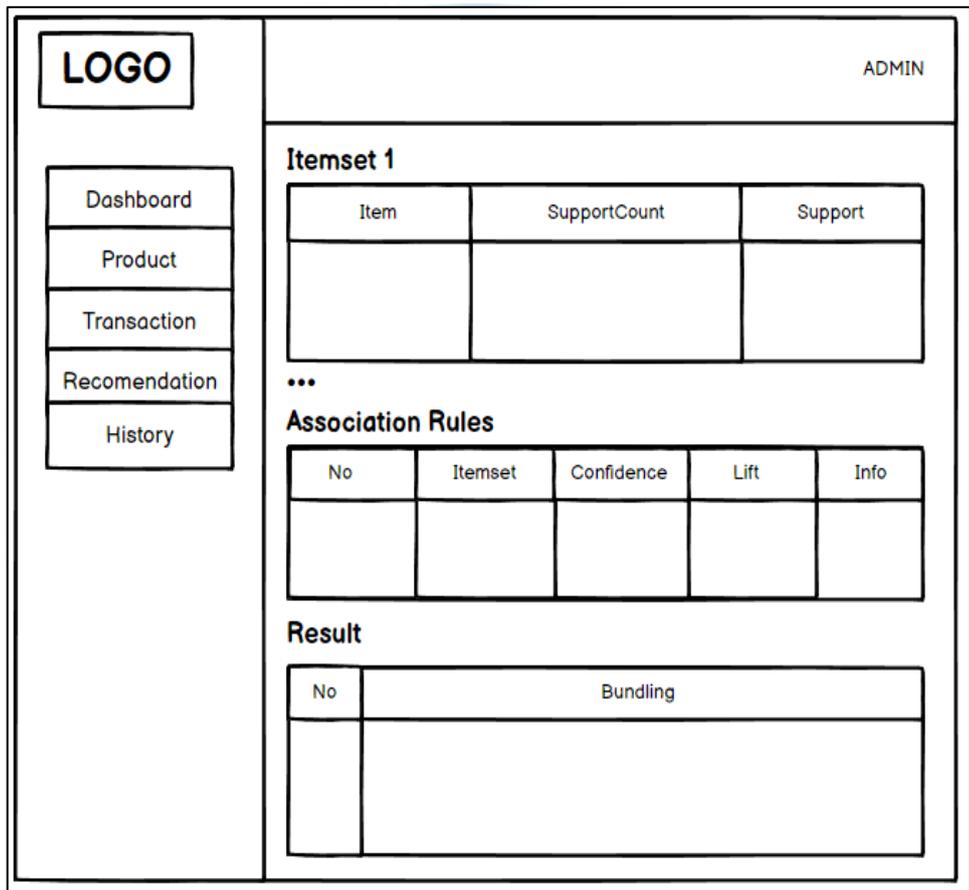
E. Antarmuka Halaman Recommendation

The image shows a web interface for the Recommendation page. On the left is a sidebar menu with a 'LOGO' box at the top, followed by buttons for 'Dashboard', 'Product', 'Transaction', 'Recommendation', and 'History'. The main content area on the right is titled 'ADMIN' and contains the following elements:

- 'Range tanggal transaksi:' with an input box containing the placeholder 'MM/DD/YYYY - MM/DD/YYYY'.
- 'Nilai Minimum Support:' with an input box containing the placeholder 'Input Minimum Support'.
- 'Nilai Minimum Confidence' with an input box containing the placeholder 'Input Minimum Confidence'.
- A 'Hitung' button at the bottom.

Gambar 3.24 Rancangan Antarmuka *Recommendation*

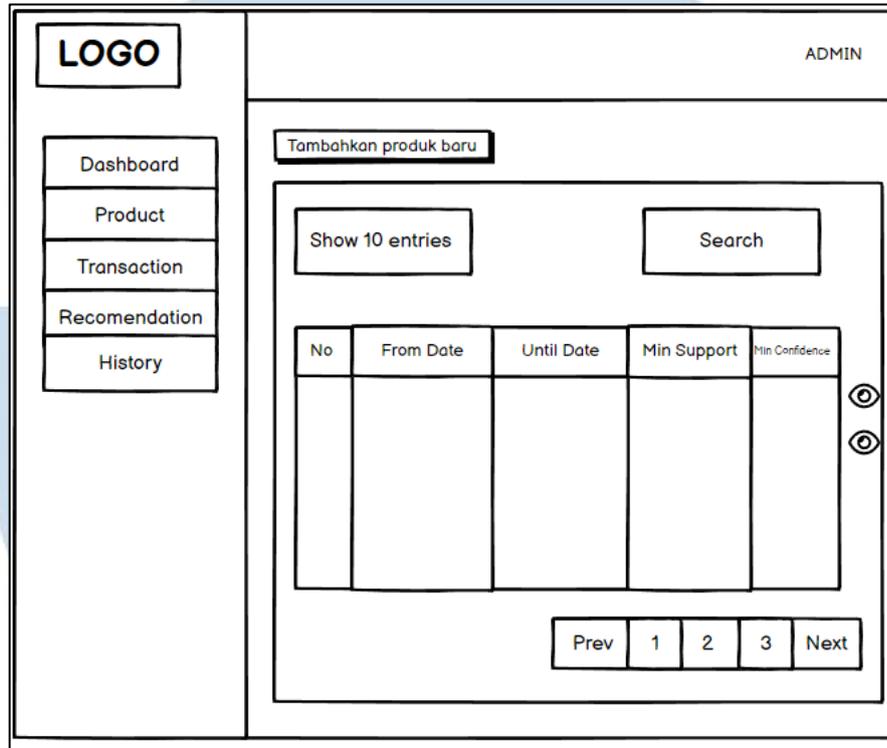
Gambar 3.24 merupakan Rancangan Antarmuka Recommendation. Di bagian ini terdapat *input box* untuk *input* tanggal transaksi, *input box minimum support*, dan *input box minimum confidence*. Di bagian bawah terdapat tombol Hitung. Apabila tombol Hitung dipilih maka akan memunculkan Rancangan Antarmuka Halaman Hasil *Recommendation* seperti pada Gambar 3.25 Rancangan Antarmuka Halaman Hasil *Recommendation*.



Gambar 3.25 Rancangan Antarmuka Halaman Hasil *Recommendation*

Gambar 3.25 merupakan Rancangan Antarmuka Halaman Hasil *Recommendation*. Pada bagian ini menunjukkan hasil proses dari algoritma Apriori yang berupa nilai *support count*, nilai *support*, nilai *confidence*, nilai *lift*, *itemset*, dan *result*. Pada bagian *result* akan memunculkan hasil *output* berupa *product bundling* dari hasil algoritma Apriori.

F. Antarmuka Halaman History



Gambar 3.26 Rancangan Antarmuka *History*

Gambar 3.26 merupakan Rancangan Antarmuka *History*. Di bagian atas terdapat *header* dan pada bagian kiri terdapat *side navigation bar* untuk navigasi menu. Pada halaman ini menampilkan tanggal transaksi dari dan sampai berdasarkan *input* dari *admin*, *minimum support*, dan *minimum confidence*. Di samping itu, terdapat fungsi pencarian (*search*) untuk mencari riwayat, serta opsi *show entries* untuk menentukan jumlah *riwayat* yang ingin ditampilkan. Dan ada juga tombol berbentuk *icon eye* yang berfungsi untuk memunculkan hasil rekomendasi yang pernah dibuat seperti pada Gambar 3.25 Rancangan Antarmuka Hasil *Recommendation*