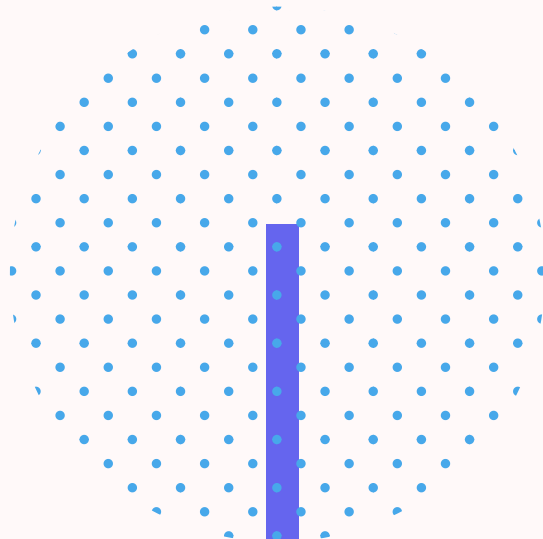




PROGRAM ADMIN SUPERMART

Aaqila Dhiyaanisafa G.
1910130001



ESQ Business School

DESKRIPSI PROGRAM

Program Admin Supermarket merupakan suatu program yang diperuntukan untuk karyawan di supermarket bagian gudang. Program ini dirancang untuk mempermudah pencatatan produk-produk yang masuk gudang berdasarkan kategori produk tersebut dan disertai dengan detail dari masing-masing produk.

Fungsi utama dari program admin supermarket adalah untuk mencatat produk-produk yang masuk ke gudang, yang dibagi berdasarkan kategorinya yang harus ditambahkan oleh user terlebih dahulu. Selain itu, program ini dapat menampilkan seluruh produk yang telah dicatat di dalam program secara detail, mencari produk berdasarkan kategori, merk, supplier, serta bulan kadaluarsa dari produk tersebut, menghapus produk yang sudah kadaluarsa, serta mengurangi maupun menambahkan stok dari produk yang telah dipilih.

Konsep dari program ini menggunakan konsep Linked List, dimana semua data disimpan di dalam list yang sesuai dengan yang dituju. Sebuah kategori produk akan berelasi dengan supplier produk dan tambahannya yaitu detail makanan. Artinya, sebuah kategori produk dapat dikeluarkan oleh beberapa supplier, dan satu supplier dapat menyeter beberapa kategori makanan ke dalam supermarket.

TUJUAN PROGRAM

Adapun tujuan dari perancangan program ini adalah :

- Memenuhi tugas akhir mata kuliah Algoritma dan Struktur Data.
- Memberikan kemudahan untuk karyawan bagian gudang di suatu supermarket dalam pencatatan produk yang masuk.
- Mencatat produk ke dalam daftar produk berdasarkan kategori produk.
- Memudahkan untuk mencari ketersediaan barang serta menyortir produk berdasarkan kategori, merk, supplier, dan bulan kadaluarsa.
- Mengurangi dan menambahkan stok suatu produk.
- Menghapus produk yang sudah kadaluarsa dari daftar produk.



MANFAAT PROGRAM

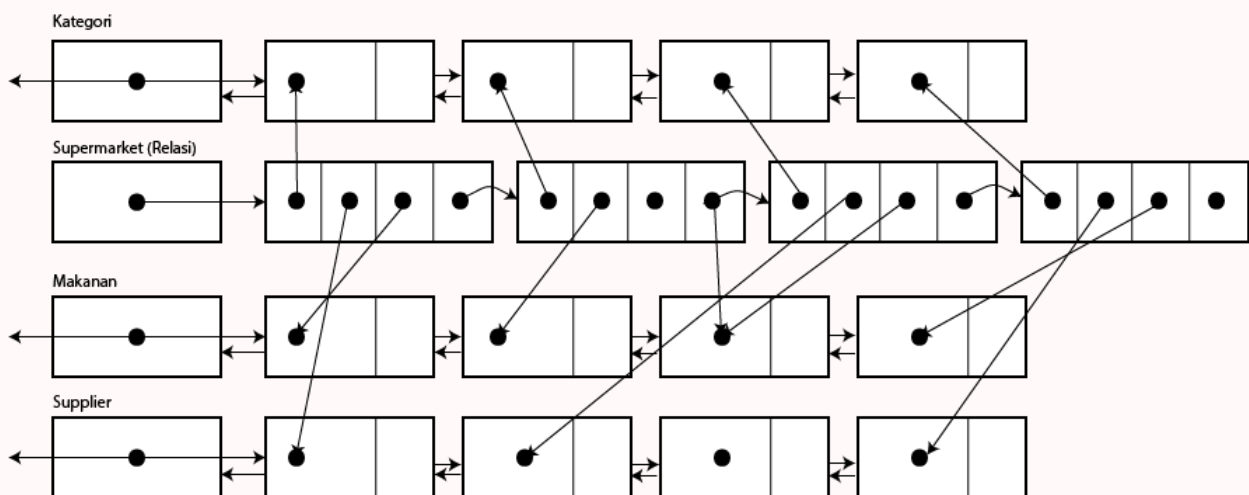
Adapun manfaat dari perancangan program ini adalah :

- Semua data mengenai produk tercatat dan tersusun dengan rapih sehingga data tidak akan hilang.
- Dapat menampilkan semua data mengenai produk dengan mudah dan detail.
- Menghapus produk yang sudah kadaluarsa dari list produk, namun masih meninggalkan jejak sehingga tidak benar-benar terhapus.
- Dapat mencari data produk berdasarkan kategori, merk, supplier, serta bulan kadaluarsa produk tersebut.



MODEL DATA

Model data yang digunakan merupakan model MultiList. MultiList merupakan suatu struktur yang terdiri dari beberapa buah list yang saling berkaitan/berelasi. Setiap struktur dari list yang berkait memiliki perbedaan karakteristik satu sama lain. Model MultiList yang digunakan adalah model ke-1, dimana terdapat 1 list parent (Kategori), 1 list relasi (Supermarket), dan 2 list child (Makanan dan Supplier).



FUNGSIONALITAS

— Kategori

Kelas ini merupakan kelas objek, dimana kita menyimpan atribut yang sesuai. Kelas Kategori memiliki :

Atribut

- **kategori : String**

Atribut kategori ini menyimpan data nama kategori.

Konstruktor

- **Kategori (kategori:String)**

Merupakan method konstruktor saat memasukkan kategori

Method

- **getKategori() : String**

Digunakan untuk mengambil data kategori yang berupa String.

- **setKategori(kategori:String) : void**

Digunakan untuk menetapkan data kategori yang berupa String.

— NodeKategori

Kelas ini merupakan kelas node untuk kelas *doubly linked list*, dimana semua data kategori yang terdapat pada kelas Kategori itu tersambung satu sama lain. Kelas NodeKategori memiliki **method** :

- **getElement() : Kategori**

Method ini digunakan untuk mengambil elemen/data yang terdapat pada kelas Kategori.

- **getNext() : NodeKategori**

Method ini digunakan untuk mengambil data pada node selanjutnya dari node yang sedang ditunjuk oleh pointer.

- **setNext(next:NodeKategori) : void**

Method ini digunakan untuk menetapkan data node selanjutnya dari node yang sedang ditunjuk oleh pointer.

- **getPrev() : NodeKategori**

Method ini digunakan untuk mengambil data pada node sebelumnya dari node yang sedang ditunjuk oleh pointer.

- **setPrev(prev:NodeKategori) : void**

Method ini digunakan untuk menetapkan data node sebelumnya dari node yang sedang ditunjuk oleh pointer.

- **setKategori(k:Kategori) : void**

Method ini digunakan untuk menetapkan data kategori.

FUNGSIONALITAS

ListKategori

Kelas ini merupakan kelas linked list yang menampung fungsi berupa method. Model linked list yang diimplementasikan dalam kelas ini adalah *doubly linked list*, dimana pointer-nya bisa 2 arah. Kelas ListKategori memiliki :

Atribut

- **choose : Kategori**
Atribut ini digunakan untuk memilih kategori ketika menambahkan data produk.
- **head : NodeKategori**
Atribut ini digunakan sebagai penanda node yang paling awal.
- **tail : NodeKategori**
Atribut ini digunakan sebagai penanda node yang paling akhir.
- **size : int**
Integer yang digunakan sebagai pengukur size dari list.

Method

- **size() : int**
Integer yang digunakan untuk menunjukkan *size* dari list.
- **getChoose() : Kategori**
Digunakan untuk mengambil data yang telah dipilih dari kelas Kategori.
- **setChoose(choose:Kategori) : void**
Digunakan untuk menetapkan data yang telah dipilih dari kelas Kategori.
- **isEmpty() : boolean**
Boolean yang digunakan untuk memperlihatkan apakah kondisi list kosong atau tidak.
- **first() : Kategori**
Menandakan node paling pertama/awal (head) dari list.
- **last() : Kategori**
Menandakan node paling akhir (tail) dari list.
- **addKategori(k:Kategori) : void**
Menambahkan data kategori pada list.
- **showKategori() : void**
Memperlihatkan semua isi list yaitu data kategori.
- **searchKategori(kode:int) : void**
Mencari index dari list dan juga mengimplementasikan method setChoose() agar dapat menetapkan pilihan dari kelas Kategori.

FUNGSIONALITAS

Makanan

Kelas ini merupakan kelas objek, dimana kita menyimpan atribut yang sesuai. Kelas Makanan memiliki :

Atribut

- **merk, stok, berat, expired, varian, harga : String**
Atribut ini menyimpan data yang merupakan detail produk yang terdiri dari merk, stok, berat, bulan kadaluarsa, varian, dan harga produk.

Konstruktor

- **Makanan (merk, stok, berat, expired, varian, harga:String)**
Merupakan method konstruktor saat memasukkan data Makanan.

Method

- **getMerk(), getStok(), getBerat(), getExpired(), getVarian(), getHarga() : String**
Digunakan untuk mengambil data Makanan yang berupa String.
- **setMerk, setStok, setBerat, setExpired, setVarian, setHarga (merk, stok, berat, expired, varian, harga:String) : void**
Digunakan untuk menetapkan data Makanan yang berupa String.

NodeMakanan

Kelas ini merupakan kelas node untuk kelas *doubly linked list*, dimana semua data Makanan yang terdapat pada kelas Makanan itu tersambung satu sama lain. Kelas NodeMakanan memiliki **method** :

- **getElement() : Makanan**
Method ini digunakan untuk mengambil elemen/data yang terdapat pada kelas Makanan.
- **getNext() : NodeMakanan**
Method ini digunakan untuk mengambil data pada node selanjutnya dari node yang sedang ditunjuk oleh pointer.
- **setNext(next:NodeMakanan) : void**
Method ini digunakan untuk menetapkan data node selanjutnya dari node yang sedang ditunjuk oleh pointer.
- **getPrev() : NodeMakanan**
Method ini digunakan untuk mengambil data pada node sebelumnya dari node yang sedang ditunjuk oleh pointer.
- **setPrev(prev:NodeMakanan) : void**
Method ini digunakan untuk menetapkan data node sebelumnya dari node yang sedang ditunjuk oleh pointer.
- **setMakanan(m:Makanan) : void**
Method ini digunakan untuk menetapkan data Makanan.

FUNGSIONALITAS

ListMakanan

Kelas ini merupakan kelas linked list yang menampung fungsi berupa method. Model linked list yang diimplementasikan dalam kelas ini adalah *doubly linked list*, dimana pointernya bisa 2 arah. Kelas ListMakanan memiliki :

Atribut

- **head : NodeMakanan**
Atribut ini digunakan sebagai penanda node yang paling awal.
- **tail : NodeMakanan**
Atribut ini digunakan sebagai penanda node yang paling akhir.
- **size : int**
Integer yang digunakan sebagai pengukur size dari list.

Method

- **size() : int**
Integer yang digunakan untuk menunjukkan *size* dari list.
- **isEmpty() : boolean**
Boolean yang digunakan untuk memperlihatkan apakah kondisi list kosong atau tidak.
- **first() : Makanan**
Menandakan node paling pertama/awal (head) dari list.
- **last() : Makanan**
Menandakan node paling akhir (tail) dari list.
- **addMakanan(m:Makanan) : void**
Menambahkan data dari kelas Makanan pada list.
- **showMakanan() : void**
Memperlihatkan semua isi list yaitu data makanan (detail).
- **removeExp(exp:String) : void**
Menghapus data dari kelas Makanan sesuai dengan *keyword* yang diinput oleh *user*.
- **kurangiStok(merk:String, stok:Int)**
Mengurangi jumlah stok yang dicari dengan nama merk dari produk tersebut.
- **tambahStok(merk:String, stok:Int)**
Menambahkan jumlah stok yang dicari dengan nama merk dari produk tersebut.

FUNGSIONALITAS

Supplier

Kelas ini merupakan kelas objek, dimana kita menyimpan atribut yang sesuai. Kelas Supplier memiliki :

Atribut

- **namaPerusahaan, area, cp : String**

Atribut ini menyimpan data yang merupakan detail supplier produk yang berupa nama perusahaan, area perusahaan, dan *contact person*.

Konstruktor

- **Supplier (namaPerusahaan, area, cp:String)**

Merupakan method konstruktor saat memasukkan atribut di Supplier.

Method

- **getNamaPerusahaan(), getArea(), getCp() : String**

Digunakan untuk mengambil data Supplier yang berupa String.

- **setNamaPerusahaan, setArea, setCp : void**

Digunakan untuk menetapkan data Supplier yang berupa String.

NodeSupplier

Kelas ini merupakan kelas node untuk kelas *doubly linked list*, dimana semua data Supplier yang terdapat pada kelas Supplier itu tersambung satu sama lain. Kelas NodeMakanan memiliki **method** :

- **getElement() : Supplier**

Method ini digunakan untuk mengambil elemen/data yang terdapat pada kelas Supplier.

- **getNext() : NodeSupplier**

Method ini digunakan untuk mengambil data pada node selanjutnya dari node yang sedang ditunjuk oleh pointer.

- **setNext(next:NodeSupplier) : void**

Method ini digunakan untuk menetapkan data node selanjutnya dari node yang sedang ditunjuk oleh pointer.

- **getPrev() : NodeSupplier**

Method ini digunakan untuk mengambil data pada node sebelumnya dari node yang sedang ditunjuk oleh pointer.

- **setPrev(prev:NodeSupplier) : void**

Method ini digunakan untuk menetapkan data node sebelumnya dari node yang sedang ditunjuk oleh pointer.

- **setSupplier(s:Supplier) : void**

Method ini digunakan untuk menetapkan data Supplier.

FUNGSIONALITAS

— ListSupplier

Kelas ini merupakan kelas linked list yang menampung fungsi berupa method. Model linked list yang diimplementasikan dalam kelas ini adalah *doubly linked list*, dimana pointer-nya bisa 2 arah. Kelas ListSupplier memiliki :

Atribut

- **head : NodeSupplier**
Atribut ini digunakan sebagai penanda node yang paling awal.
- **tail : NodeSupplier**
Atribut ini digunakan sebagai penanda node yang paling akhir.
- **size : int**
Integer yang digunakan sebagai pengukur size dari list.

Method

- **size() : int**
Integer yang digunakan untuk menunjukkan *size* dari list.
- **isEmpty() : boolean**
Boolean yang digunakan untuk memperlihatkan apakah kondisi list kosong atau tidak.
- **first() : Supplier**
Menandakan node paling pertama/awal (head) dari list.
- **last() : Supplier**
Menandakan node paling akhir (tail) dari list.
- **addSupplier(s:Supplier) : void**
Menambahkan data dari kelas Supplier pada list.
- **showSupplier() : void**
Memperlihatkan semua isi list yaitu data supplier.

FUNGSIONALITAS

— NodeSupermarket

Kelas ini merupakan kelas node untuk kelas *single linked list*, dimana semua data Supplier yang terdapat pada kelas Supermarket itu tersambung satu sama lain. Kelas NodeSupermarket memiliki :

Konstruktor

- **Supermarket (k:Kategori, m:Makanan, s:Supplier)**
Merupakan method konstruktor untuk menyambungkan semua kelas (Kategori, Makanan, Supplier)

Method

- **getElementK() : Kategori**
Method ini digunakan untuk mengambil elemen/data yang terdapat pada kelas Kategori.
- **getElementM() : Makanan**
Method ini digunakan untuk mengambil elemen/data yang terdapat pada kelas Makanan.
- **getElementS() : Supplier**
Method ini digunakan untuk mengambil elemen/data yang terdapat pada kelas Supplier.
- **getNext() : NodeSupermarket**
Method ini digunakan untuk mengambil data pada node selanjutnya dari node yang sedang ditunjuk oleh pointer.
- **setNext(n:NodeSupermarket) : void**
Method ini digunakan untuk menetapkan data node selanjutnya dari node yang sedang ditunjuk oleh pointer.

FUNGSIONALITAS

— ListSupermarket

Kelas ini merupakan kelas linked list yang menampung fungsi berupa method. Kelas ini juga merupakan kelas relasi, dimana kelas ini adalah tempat perhubungan antar list parent (ListKategori) dan list child (ListMakanan dan ListSupplier). Model linked list yang diimplementasikan dalam kelas ini adalah *single linked list*. Kelas ListSupermarket memiliki :

Atribut

- **head : NodeSupermarket**
Atribut ini digunakan sebagai penanda node yang paling awal.
- **tail : NodeSupermarket**
Atribut ini digunakan sebagai penanda node yang paling akhir.
- **size : int**
Integer yang digunakan sebagai pengukur size dari list.

Method

- **size() : int**
Integer yang digunakan untuk menunjukkan *size* dari list.
- **isEmpty() : boolean**
Boolean yang digunakan untuk memperlihatkan apakah kondisi list kosong atau tidak.
- **first() : NodeSupermarket**
Menandakan node paling pertama/awal (head) dari list.
- **last() : NodeSupermarket**
Menandakan node paling akhir (tail) dari list.
- **addSupply(s:Kategori, m:Makanan, s:Supplier) : void**
Menambahkan data dari kelas Supply pada list.
- **printAllProduct() : void**
Memperlihatkan semua isi list.
- **searchByKategori() : void**
Mencari isi list berdasarkan kategori.
- **searchByMerk() : void**
Mencari isi list berdasarkan merk.
- **searchBySupplier() : void**
Mencari isi list berdasarkan nama perusahaan.
- **searchByExpired() : void**
Mencari isi list berdasarkan bulan kadaluarsa.

FUNGSIONALITAS

AdminSupermarket

Kelas ini merupakan kelas main, tempat dimana mengatur tampilan UI dan memanggil method-method dari kelas linked list sesuai fungsi. Method dari kelas linked list dipanggil dengan membuat objek baru terlebih dahulu, untuk menginisialisasikan kelas yang dibutuhkan. Di kelas main disini membutuhkan *input user* sehingga kita menggunakan kelas Scanner.

```
Selamat Datang di Supermart!
-----
|  Apa yang ingin anda lakukan?  |
-----
| 1. Tambahkan Kategori         |
| 2. Tambahkan Produk           |
| 3. Cari Produk                |
| 4. Lihat Semua Produk         |
| 5. Lihat Kategori             |
| 6. Lihat Supplier             |
| 7. Tambah/Kurangi Stok Produk |
| 8. Hapus Produk Expired       |
| 0. Selesai                    |
-----
| Masukkan pilihanmu :          |
```

Tampilan yang dimunculkan pertama kali adalah tampilan seperti menu, dimana user bisa menginput perintah sesuai dengan urutan angkanya. Jika user memasukkan input 1, maka akan memanggil method dari list kategori yaitu `addKategori` yang parameternya diisi dengan objek yang sudah dibuat. Dengan itu, data berhasil ditambahkan ke dalam node list kategori. Begitupula jika user memasukkan input 2, user akan diarahkan untuk mengisi detail produk dari list Makanan dan dengan memanggil method `addMakanan`, dan detail supplier dengan method `addSupplier`.

Ketika user memasukkan input 3, user akan diarahkan untuk memilih ingin mencari produk berdasarkan keyword apa. Disini, telah disediakan 4 pilihan yaitu berdasarkan merk produk, kategori, nama perusahaan supplier, serta bulan kadaluarsa produk. Setelah memilih salah satu dari keempat pilihan yang dibuat dengan konsep if-else, jika salah satu dari kondisi if-elsenya terpenuhi, maka akan memanggil method yang sesuai dengan kondisi yang terpenuhi.

```
-----
| Cari berdasarkan apa?         |
-----
| 1. Merk                      |
| 2. Kategori                  |
| 3. Nama Perusahaan           |
| 4. Bulan Kadaluarsa          |
| 5. Menu Awal                 |
| Masukkan pilihanmu :        |
```

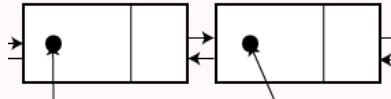
Selanjutnya, saat user memasukkan input 4/5/6, maka method yang akan terpanggil adalah method `printAllProduct()` yang berfungsi untuk menampilkan seluruh isi list supermarket, atau `showKategori()`/`showSupplier()` untuk menampilkan seluruh isi list kategori/supplier.

User juga dapat mengurangi atau menambahkan stok dari suatu produk yang sudah diinput sebelumnya oleh user dengan memasukkan input 7, lalu memasukkan merk dari produk yang ingin dikurangi/ditambahkan stoknya. Setelah itu, user akan ditanya apakah ingin mengurangi atau menambah stok pada produk beserta jumlahnya, dan pada kelas main akan terpanggil method `kurangiStok()` jika ingin mengurangi, dan `tambahStok()` jika ingin ditambahkan.

Selain itu, produk yang telah expired juga dapat dihapus oleh user dengan memasukkan input 8, lalu method `removeExp()` yang akan terpanggil setelah user memasukkan bulan expired. Untuk mengakhiri program, user dapat menginput 0 dan program akan berhenti dengan sendirinya.

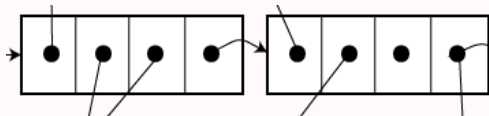
TEKNIK PEMROGRAMAN

Linked List Parent & Child



Linked List yang digunakan untuk list parent & child (list kategori, makanan, dan supplier) adalah menggunakan model doubly linked list, dimana list memiliki pointer penunjuk 2 arah, yakni ke arah node sebelum (previous/prev) dan node sesudah (next).

Linked List Relasi



Linked List yang digunakan untuk list relasi (supermarket) adalah menggunakan model single linked list, dimana list hanya memiliki pointer penunjuk 1 arah, yakni ke arah node sesudah (next) saja.

Penambahan kategori, detail produk, dan detail supplier

```
public void addKategori(Kategori k) { // adds element e to the end of the list
    NodeKategori newest = new NodeKategori();
    boolean ketemu = false;
    NodeKategori current = head;

    for(int i = 0; i < size; i++){
        if(k.getKategori().toLowerCase().equalsIgnoreCase(current.getElement().getKategori())){
            ketemu = true;
        }
        current = current.getNext();
    }
}
```

Pada ketiga method ini, user diharuskan untuk memasukkan input kategori, input detail produk (merk, stok, varian, berat, harga, bulan kadaluarsa), dan input detail supplier (nama perusahaan, area perusahaan, contact person). Disini, saya menggunakan konsep addLast, dimana next data terbaru adalah null dan prev adalah head. Dengan kata lain, data terbaru menjadi tail dari node. Khusus untuk add kategori, sebelum data dimasukkan ke node selanjutnya, terjadi pengecekan terlebih dahulu apakah datanya double atau tidak. Jika tidak, maka data akan dimasukkan ke node sedangkan jika iya, akan muncul pesan bahwa kategori sudah terdaftar.

Penambahan supply (addSupply pada kelas relasi)

```
public void addSupply(Kategori k, Makanan m, Supplier s) {
    NodeSupermarket newest = new NodeSupermarket(k, m, s, null);

    if (isEmpty()) {
        head = newest;
    }
    else {
        tail.setNext(newest);
    }
    tail = newest;
    size++;
}
```

Method ini digunakan untuk mengumpulkan/membuat data yang telah diinput dengan method add__ dalam satu list. Disini, saya menggunakan konsep addLast, dimana kita set data setelah node tail adalah data terbaru, sehingga data terbaru menjadi tail baru di list tersebut.

TEKNIK PEMROGRAMAN

Mencari data berdasarkan keyword (kategori, merk, supplier, bulan kadaluarsa)

```
else if(data.getElementN() == null){
    if(data.getElementK().getKategori().toLowerCase().equals(kategori.toLowerCase())){
        System.out.println("| Kategori " + kategori + " telah ditemukan. ");
        System.out.println("| ");
        System.out.println("| Merk : " + data.getElementM().getMerk());
        System.out.println("| Varian Rasa : " + data.getElementM().getVarian());
        System.out.println("| Stok : " + data.getElementM().getStok());
        System.out.println("| Harga : " + data.getElementM().getHarga());
        System.out.println("| Berat : " + data.getElementM().getBerat());
        System.out.println("| Bulan Kadaluarsa : " + data.getElementM().getExpired());
        System.out.println("|-----|");
    }
    else{
        System.out.println(" Maaf, produk tidak ditemukan");
    }
}
else{
    while(data.getElementN() != null){
        if(data.getElementK().getKategori().toLowerCase().equals(kategori.toLowerCase())) {
            System.out.println("| Kategori " + kategori + " telah ditemukan. ");
            System.out.println("| ");
            System.out.println("| Merk : " + data.getElementM().getMerk());
            System.out.println("| Varian Rasa : " + data.getElementM().getVarian());
            System.out.println("| Stok : " + data.getElementM().getStok());
            System.out.println("| Harga : " + data.getElementM().getHarga());
            System.out.println("| Berat : " + data.getElementM().getBerat());
            System.out.println("| Bulan Kadaluarsa : " + data.getElementM().getExpired());
            System.out.println("|-----|");

            data = data.getElementN();
        }
        data = data.getElementN();
        i++;
    }
}
```

Pada method ini, saya menggunakan if - else if - else, yang dimana pada masing-masingnya terdapat while dan atau if else lagi. Di method ini, if-else yang digunakan pertama adalah untuk mengecek apakah kelas nodeSupermarket itu kosong atau tidak. Jika kosong, maka akan keluar pesan bahwa produk tidak ada. Kedua adalah else if dengan kondisi data setelah head adalah null, dengan dirangkap lagi dengan kondisi jika data head sama dengan keyword yang diinput oleh user, maka akan ditampilkan data sesuai keyword yang user cari. Terakhir adalah else, dimana jika kedua kondisi diatas tidak terpenuhi maka ketika data setelah head bukan null, dengan kondisi lagi jika data head sama dengan keyword yang user input, maka akan ditampilkan data yang sesuai. Kedua kondisi diatas juga di looping, looping pertama dalam kondisi if dan looping kedua pada kondisi while agar data akan terus di cek sampai kondisi-kondisi utama tidak terpenuhi.

TEKNIK PEMROGRAMAN

Menampilkan seluruh isi list kategori, supplier, makanan, supermarket (supply)

```
else if(data.getElementN() == null){
    if(data.getElementK().getKategori().toLowerCase().equals(kategori.toLowerCase())){
        System.out.println("| Kategori " + kategori + " telah ditemukan. ");
        System.out.println("| ");
        System.out.println("| Merk : " + data.getElementM().getMerk());
        System.out.println("| Varian Rasa : " + data.getElementM().getVarian());
        System.out.println("| Stok : " + data.getElementM().getStok());
        System.out.println("| Harga : " + data.getElementM().getHarga());
        System.out.println("| Berat : " + data.getElementM().getBerat());
        System.out.println("| Bulan Kadaluarsa : " + data.getElementM().getExpired());
        System.out.println("|-----|");
    }
    else{
        System.out.println(" Maaf, produk tidak ditemukan");
    }
}
else{
    while(data.getElementN() != null){
        if(data.getElementK().getKategori().toLowerCase().equals(kategori.toLowerCase())) {
            System.out.println("| Kategori " + kategori + " telah ditemukan. ");
            System.out.println("| ");
            System.out.println("| Merk : " + data.getElementM().getMerk());
            System.out.println("| Varian Rasa : " + data.getElementM().getVarian());
            System.out.println("| Stok : " + data.getElementM().getStok());
            System.out.println("| Harga : " + data.getElementM().getHarga());
            System.out.println("| Berat : " + data.getElementM().getBerat());
            System.out.println("| Bulan Kadaluarsa : " + data.getElementM().getExpired());
            System.out.println("|-----|");

            data = data.getElementN();
        }
        data = data.getElementN();
        i++;
    }
}
```

Method ini memiliki konsep yang sama persis dengan menampilkan data, hanya berbeda di bagian keyword karena user tidak perlu memasukkan keyword. Data yang akan ditampilkan dengan method ini adalah keseluruhan isi list.

TEKNIK PEMROGRAMAN

Menghapus produk dengan keyword bulan kadaluarsanya

```
Makanan removeExp = data.getElement();
NodeMakanan nextData = data.getNext();
NodeMakanan beforeData = data.getPrev();

//if remove account in head
if(beforeData == null){

    //when next data didnt null
    if(nextData != null)
        nextData.setPrev(null);

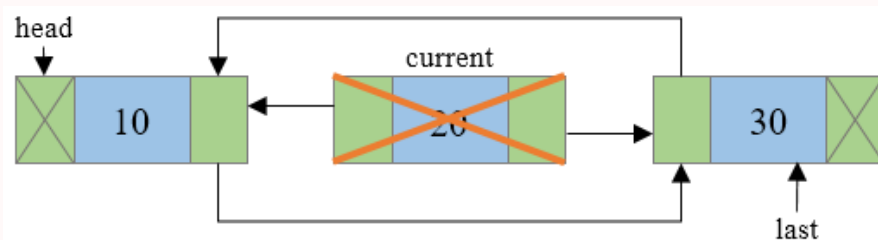
    size--;

    //if size 1
    if(size == 1){
        head = nextData;
        tail = nextData;
    }

    //else
    else{
        head = nextData;
    }
}

//if remove tail data
else if(nextData == null){

    if(beforeData != null)
        beforeData.setNext(null);
```



Method ini menggunakan konsep removeCustom, dimana user harus memasukkan bulan kadaluarsa produk tersebut, lalu akan dicari dengan menggunakan if else untuk menemukan index dari data di list yang ingin kita cari. Konsep dari remove ini bukanlah benar-benar menghapus node, melainkan memutuskan panahnya sehingga node yang telah dihapus tidak ada hubungannya lagi dengan node sebelum dan selanjutnya.

TEKNIK PEMROGRAMAN

Mengurangi/menambahkan stok dari suatu produk

```
else{
    NodeMakanan data = head;
    int i = 0;

    System.out.println(data.getElement().getMerk());
    while (!data.getElement().getMerk().toLowerCase().equals(merk.toLowerCase())) {
        data = data.getNext();
        i++;
        if(i + 1 > size)
            break;
    }

    if( i + 1 > size){
        System.out.println(" Maaf, produk tidak ditemukan");
        System.out.println("");
    }

    //ditemukan
    else if (data.getElement().getMerk().toLowerCase().equals(merk.toLowerCase())){
        int stoknow = data.getElement().getStok() - stok;
        data.getElement().setStok(stoknow);
        System.out.println(" Stok dari produk merk " + merk + " berhasil dikurangi!");
        System.out.println(" Stok " + merk + " sekarang : " + data.getElement().getStok());
    }
}
```

Method ini hanya menggunakan konsep mengambil data; disini kita mengambil data stok dari suatu produk. User akan diawali dengan ditanyakan merk dari produk yang ingin dioperasikan. Setelah itu, user akan diminta untuk memasukkan perintah kurangi/tambah serta jumlah, sehingga operasi serta hasil yang ditampilkan akan sesuai dengan kondisi yang user pilih. Di dalam method ini, kita hanya perlu mengambil data stok produk, lalu data yang telah kita dapat dioperasikan dengan integer yang telah diinput oleh user,

DOKUMENTASI

```
Selamat Datang di Supermart!
-----
|   Apa yang ingin anda lakukan?   |
|-----|
| 1. Tambahkan Kategori           |
| 2. Tambahkan Produk             |
| 3. Cari Produk                  |
| 4. Lihat Semua Produk           |
| 5. Lihat Kategori              |
| 6. Lihat Supplier              |
| 7. Tambah/Kurangi Stok Produk   |
| 8. Hapus Produk Expired         |
| 0. Selesai                     |
|-----|
| Masukkan pilihanmu :            |
```

Tampilan Utama

```
Selamat Datang di Supermart!
-----
|   Apa yang ingin anda lakukan?   |
|-----|
| 1. Tambahkan Kategori           |
| 2. Tambahkan Produk             |
| 3. Cari Produk                  |
| 4. Lihat Semua Produk           |
| 5. Lihat Kategori              |
| 6. Lihat Supplier              |
| 7. Tambah/Kurangi Stok Produk   |
| 8. Hapus Produk Expired         |
| 0. Selesai                     |
|-----|
| Masukkan pilihanmu : 1          |
|-----|
|
|-----|
| Masukkan kategori : susu        |
| Kategori susu berhasil ditambahkan. |
|-----|
```

Tambah Kategori

```
|-----|
| Masukkan pilihanmu : 3          |
|-----|
|
|-----|
| Cari berdasarkan apa?          |
|-----|
| 1. Merk                        |
| 2. Kategori                    |
| 3. Nama Perusahaan            |
| 4. Bulan Kadaluarsa           |
| 5. Menu Awal                  |
| Masukkan pilihanmu : 1          |
|
| Masukkan merk produk : jan      |
|
| Maaf, produk tidak ditemukan    |
|-----|
```

Cari Produk (Merk)

```
| Masukkan pilihanmu : 2          |
|-----|
|
|-----|
| Kategori yang tersedia :       |
|-----|
| Kategori Supermart            |
|-----|
| 1.susu                        |
|-----|
| Jumlah kategori : 1           |
|
| Masukkan kode : 1             |
|-----|
```

```
|-----|
| Detail produk                  |
|-----|
| Merk : indomilk                |
| Varian Rasa : coklat           |
| Stok : 100                     |
| Harga : 10000                  |
| Berat (g) : 100                |
| Bulan Kadaluarsa : des         |
|-----|
| Detail supplier                |
|-----|
| Nama Perusahaan : indofood     |
| Area : jakarta                 |
| Contact Person : yusuf         |
|-----|
| Tambahkan produk ini ke supermarket? |
| (ya/tidak)ya                   |
|
| Produk ini berhasil ditambahkan! |
| Kembali ke menu awal...         |
|-----|
```

Tambah Produk

```
|-----|
| Nama Perusahaan : indofood     |
| Area : jakarta                 |
| Contact Person : yusuf         |
|-----|
| Jumlah Supplier : 2            |
|
| Nama Perusahaan : indo         |
| Area : ciamhi                  |
| Contact Person : a             |
|-----|
| Jumlah Supplier : 2            |
|-----|
```

Lihat Supplier

DOKUMENTASI

```
|-----|
| Masukkan pilihanmu : 5 |
|-----|
|
|-----|
|           Kategori Supermart           |
|-----|
| 1.susu |
|-----|
| Jumlah kategori : 1 |
|-----|
```

Lihat Kategori

```
|-----|
| Masukkan pilihanmu : 7 |
|-----|
|
|-----|
| Masukkan merk produk : jan |
| Tambah/Kurang? tambah |
| Masukkan jumlah stok : 10 |
|-----|
indomilk
Maaf, produk tidak ditemukan
```

Tambah Stok

```
|-----|
| Masukkan pilihanmu : 7 |
|-----|
|
|-----|
| Masukkan merk produk : indomilk |
| Tambah/Kurang? kurang |
| Masukkan jumlah stok : 2 |
|-----|
indomilk
Stok dari produk merk indomilk berhasil dikurangi!
Stok indomilk sekarang : 8
|-----|
|
|-----|
| Apa yang ingin anda lakukan? |
|-----|
| 1. Tambahkan Kategori |
| 2. Tambahkan Produk |
| 3. Cari Produk |
| 4. Lihat Semua Produk |
| 5. Lihat Kategori |
| 6. Lihat Supplier |
| 7. Tambah/Kurangi Stok Produk |
| 8. Hapus Produk Expired |
| 0. Selesai |
|-----|
| Masukkan pilihanmu : 4 |
|-----|
|
|-----|
| Merk : indomilk |
| Varian Rasa : coklat |
| Stok : 8 |
| Harga : 10000 |
| Berat (g) : 100 |
| Bulan Kadaluarsa : des |
|-----|
```

Kurangi Stok

```
| Masukkan pilihanmu : 4 |
|-----|
|
|-----|
| Merk : indomilk |
| Varian Rasa : coklat |
| Stok : 10 |
| Harga : 10000 |
| Berat (g) : 100 |
| Bulan Kadaluarsa : des |
|-----|
Jumlah Produk : 2
| Merk : frisian flag |
| Varian Rasa : coconut |
| Stok : 100 |
| Harga : 10000 |
| Berat (g) : 100 |
| Bulan Kadaluarsa : jan |
|-----|
Jumlah Produk : 2
|-----|
|
|-----|
| Lihat riwayat/detail produk? (Ya/Tidak) |
| ya |
|-----|
|
|-----|
| Produk 1 |
|-----|
|
|-----|
| Kategori : susu |
| Merk : indomilk |
| Varian Rasa : coklat |
| Stok : 10 |
| Harga : 10000 |
| Berat : 100 |
| Bulan Kadaluarsa : des |
|-----|
|
|-----|
| Nama Perusahaan : indofood |
| Area : jakarta |
| Contact Person : yusuf |
|-----|
```

Lihat Produk

```
|-----|
| Masukkan pilihanmu : 8 |
|-----|
|
|-----|
| Masukkan bulan kadaluarsa : jan |
| Berhasil menghapus produk kadaluarsa bulan jan dari list. |
|-----|
|
|-----|
| Apa yang ingin anda lakukan? |
|-----|
| 1. Tambahkan Kategori |
| 2. Tambahkan Produk |
| 3. Cari Produk |
| 4. Lihat Semua Produk |
| 5. Lihat Kategori |
| 6. Lihat Supplier |
| 7. Tambah/Kurangi Stok Produk |
| 8. Hapus Produk Expired |
| 0. Selesai |
|-----|
| Masukkan pilihanmu : 4 |
|-----|
|
|-----|
|
|-----|
| Maaf, belum ada produk. |
|-----|
```

Hapus Produk Expired

KESIMPULAN

Program Admin Supermarket merupakan program yang dirancang untuk memberikan kemudahan bagi petugas supermarket bagian gudang dalam pencatatan produk yang masuk ke dalam gudang. Rancangan ini terbentuk dalam 4 kelas Linked List Model, 4 Node, 3 kelas atribut, dan 1 kelas main. Pada program ini pun diterapkan juga single linked list dan doubly linked list.

Model yang digunakan dalam perancangan program ini adalah model Multi List, dimana terdapat 1 list parent (kategori), 2 list child (makanan & supplier), dan 1 list relasi sebagai penghubung antara list parent dan list child.

