

LAPORAN TUGAS BESAR
MATA KULIAH ALGORITMA DAN STRUKTUR DATA
“OTOMATISASI PROSES PENJUALAN
UPPEREAST CAFE



Disusun Oleh:

Kelompok 6

| | |
|-------------------------------------|--------------------|
| Benony Gabriel | (105222002) |
| Faris Farhan | (105222013) |
| Sheva Haya Milano | (105222036) |
| Muhammad Rizqi Ferdhinandito | (105222041) |
| Arshanda Geulis Nawajaputri | (105222045) |

PROGRAM STUDI ILMU KOMPUTER

FAKULTAS SAINS DAN ILMU KOMPUTER

UNIVERSITAS PERTAMINA

2024

BAB I

SPESIFIKASI PROGRAM

1.1 Proses Bisnis

UperEast kafe ingin melakukan otomasi proses penjualan. Proses yang ingin di dukung dengan sistem adalah pengelolaan menu dan harga dan pengelolaan stok bahan baku.

Menu yang dijual di kafe adalah produk roti dan kue, minuman, dan makanan utama. Menu ini dikelompokkan berdasarkan tipenya untuk memudahkan pelanggan melakukan pencarian, juga memudahkan karyawan kafe dalam mengelola menu. Berikut ini adalah menu yang dijual di kafe.

| Kategori | Sub Kategori | Item |
|-----------------|---------------------|--|
| Roti dan Kue | Viennoiseries | Croissant, Pain au Chocolat, Brioche |
| | Pastry | Éclair, Mille-feuille, Tarte Tatin, Tarlet |
| | Bread | Baguette, Sourdough, Brioche Long |
| | Bun | Bagel, Burger, Hotdog |
| Minuman | Jus | Apple Juice, Orange Juice, Strawberry Juice |
| | Teh | Earl Grey, Chamomile, English Breakfast, Darjeeling, Oolong, Matcha, Jasmine |
| | Kopi | Long black, Cappuccino, Latte, Espresso, Macchiato, Piccol |
| Makanan Utama | Salad | Caesar Salad, Caprese salad, Waldorf Salad, Nicoise salad |
| | Sup | Pumpkin soup, mushroom soup, Corn Soup |
| | Sandwich | Club Sandwich, Tuna Sandwich |
| | Pasta | Gnocchi, Aglio Olio, Lasagna, Fettucine, Carbonara |

Menu makanan tersebut juga dapat dikelompokkan berdasarkan waktu makannya seperti dibawah ini:

| Event | Item |
|---|--|
| Breakfast | Semua menu Sandwich, bread, bun, minuman |
| Brunch | Semua menu Sandwich, bread, bun, salad, sup, minuman |
| Lunch | Semua menu makanan utama, pastry, minuman |
| Morning Coffe/ Tea time/Afternoon Coffe | Semua menu minuman, viennoiseries, dan pastry |
| Dinner | Semua menu makanan utama, pastry, minuman |

Data menu tersebut harus disimpan menggunakan struktur data yang tepat sehingga nantinya dapat dikelola oleh pegawai kafe dan mudah dicari saat pelanggan melakukan pemesanan. Daftar menu ini dapat berubah, karena itu dibutuhkan fitur untuk mengelola menu yang mencakup penambahan, pengurangan, dan pengaturan ulang kategori, sub kategori, menu item juga event dan menu item.

Perubahan harga bahan baku mempengaruhi harga jual produk. Akibatnya harga tiap item dalam menu juga dapat berubah di kemudian hari. Dibutuhkan fitur untuk mengelola harga tiap menu item.

Menu item yang ada pada sub-kategori viennoiseries, pastry, dan bread dibeli dari supplier lain sedangkan makanan utama dan minuman dibuat langsung di dapur kafe. Setiap harinya kafe hanya membeli roti dan bun dalam jumlah tertentu. Kafe juga hanya menyetok bahan-bahan untuk membuat makanan utama dan minuman dalam jumlah tertentu. Karena itu tiap kafe memiliki jumlah ketersediaan maksimal untuk tiap menu item. Jumlah ini akan berkurang setiap kali pelanggan melakukan pemesanan. Jika ketersediaan 0, maka menu item tersebut tidak bisa dipesan lagi oleh pelanggan. Jumlah ketersediaan ini harus selalu terupdate, sehingga pelayan dapat menerima atau menolak pesanan pelanggan saat melayani pelanggan.

Selanjutnya UperEast kafe juga ingin mengelola stok bahan baku. Untuk memastikan stok produk selalu tersedia setiap hari, UpperEast kafe perlu mencatat stok bahan baku yang tersedia. Setiap menu yang disediakan oleh kafe memiliki daftar bahan

baku dan jumlah bahan baku yang dibutuhkan. Setiap kali ada order untuk menu tertentu, seluruh stok bahan baku yang digunakan untuk membuat menu tersebut akan dikurangi. Pengelola UpperEast ingin ada notifikasi otomatis setiap kali bahan baku sudah melewati batas minimum stok. Notifikasi ini dikirimkan di akhir hari pada pukul 17, berisi daftar bahan baku yang jumlahnya kurang dari minimum stock hanya jika ada bahan baku yang melebihi batas minimum stok.

1.2 Requirements List

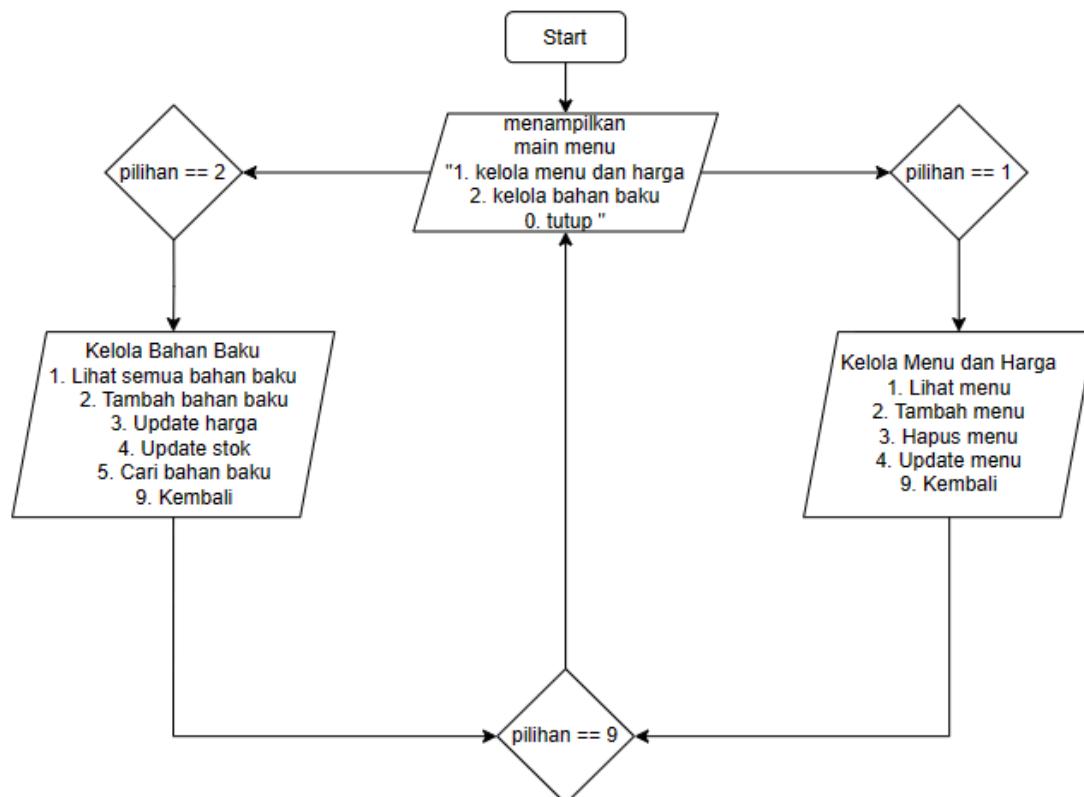
Berdasarkan proses bisnis yang telah dijelaskan di atas, terdapat sejumlah kebutuhan atau requirement dalam aktivitas pengelolaan menu dan harga. Berikut adalah penjabaran dari user requirement menjadi fitur-fitur yang dibutuhkan:

| No | User Requirement | Fitur |
|----|--|---------------------|
| 1 | Program dapat menampilkan semua daftar menu beserta harga dan stok yang tersedia berdasarkan kategori, sub kategori dan item-item | Semua menu |
| 2 | Program akan menampilkan beberapa event yang di dalamnya terdapat kategori, sub kategori dan item-item yang cocok dengan event tersebut | Menu event |
| 3 | Program dapat memberikan akses kepada admin untuk melakukan pencarian terhadap item menu yang ingin dicari | Cari menu |
| 4 | Program dapat memberikan akses kepada admin untuk menambahkan kategori baru ke dalam sistem dengan menginputkan juga sub kategori dan item baru | Tambah kategori |
| 5 | Program dapat memberikan akses kepada admin untuk menambahkan sub kategori baru ke kategori yang sudah didalam menu dengan menginputkan juga item baru | Tambah sub kategori |
| 6 | Jika admin menambahkan item, maka harga dan stok juga akan diminta untuk diinput | Tambah item menu |

| | | |
|----|---|---|
| 7 | Jika admin ingin menambahkan kategori/sub kategori ke dalam event maka kategori tersebut harus ada terlebih dahulu dalam menu | Tambah kategori/sub kategori ke dalam event |
| 8 | Jika admin menghapus kategori, maka subkategori, item, harga dan stok akan otomatis terhapus. | Hapus kategori |
| 9 | Jika admin menghapus sub kategori, maka item, harga, dan stok akan otomatis terhapus. | Hapus sub kategori |
| 10 | Jika admin menghapus item, maka harga dan stok akan otomatis terhapus. | Hapus item menu |
| 11 | Jika admin menghapus kategori/sub kategori dari event, maka item beserta harga dan stok akan otomatis terhapus. | Hapus sub kategori event |
| 12 | Aplikasi dapat memberikan akses kepada admin untuk melakukan pembaruan harga terhadap tiap menu item | Update harga menu |
| 13 | Aplikasi dapat memberikan akses kepada admin untuk melakukan pembaruan stok item | Update stok menu |
| 14 | Stok item dapat diubah atau diperbarui oleh admin kapan saja sesuai kebutuhan. | |
| 15 | Aplikasi dapat memberikan akses kepada admin untuk menampilkan daftar bahan baku dengan harga dan ketersediaan stok | Lihat semua bahan baku |
| 16 | Aplikasi dapat memberikan akses kepada admin untuk melakukan penambahan bahan baku baru | Tambah bahan baku |
| 17 | Aplikasi dapat memberikan akses kepada admin untuk melakukan pembaruan harga bahan baku | Update harga |

| | | |
|----|--|-----------------|
| 18 | Aplikasi dapat memberikan akses kepada admin untuk melakukan pembaruan stok item | |
| 19 | Stok item dapat diubah atau diperbarui oleh admin sesuai kebutuhan. | Update stok |
| 20 | Aplikasi dapat memberikan akses kepada admin untuk melakukan pencarian terhadap bahan baku yang ingin dicari | Cari bahan baku |
| 21 | Jika admin menutup toko, sistem akan mengirim sebuah notifikasi yang berisi daftar bahan baku yang ketersedian stoknya dibawah minimum | Notifikasi |

Merujuk pada requirement tersebut, dibentuklah flowchart yang menggambarkan proses pengelolaan menu, harga, dan pengelolaan bahan baku yang dilakukan oleh admin. Aktor utama dalam sistem ini adalah Admin. Admin adalah orang yang bertanggung jawab untuk mengelola menu, harga, dan stok bahan baku. Sistem adalah perangkat lunak yang menyimpan data yang dikelola oleh admin.



Gambar 1.1 : *Main Function* proses bisnis pengelolaan menu dan pengelolaan bahan baku

Diagram tersebut menggambarkan aktivitas yang dilakukan oleh admin kafe dalam mengelola menu, dan mengelola bahan baku. Pengelolaan menu dan bahan baku yang dilakukan oleh admin berupa penambahan, penghapusan, pencarian dan update menu. Berikut ini adalah penjelasannya secara rinci :

- Lihat Menu: Admin dapat melihat daftar menu yang didalamnya terdapat informasi menu berupa nama, harga dan stok yang tersedia, menu event dan pencarian item dalam menu.
- Tambah Menu: Admin dapat menambahkan menu baru ke dalam sistem dengan menyertakan rincian informasi seperti nama, kategori, sub kategori, harga, ketersediaan stok, dan juga dapat menambahkan kategori/sub kategori ke dalam event.
- Hapus Menu: Admin dapat menghapus menu yang sudah ada dari sistem jika diperlukan.
- Update Menu: Admin dapat melakukan pembaruan harga dan stok untuk tiap menu item.
- Lihat Semua Bahan Baku: Admin dapat melihat daftar bahan baku yang didalamnya terdapat informasi menu berupa nama, harga dan stok yang tersedia.
- Tambah Bahan Baku: Admin dapat menambahkan bahan baku baru ke dalam sistem dengan menyertakan rincian informasi seperti nama, harga, dan ketersediaan stok.
- Update Harga: Admin dapat mengupdate harga bahan baku melalui sistem jika diperlukan.
- Update Stok: Admin dapat memesan bahan baku melalui sistem dengan mengupdate stok berdasarkan kebutuhan atau keadaan stok yang kurang.
- Cari Bahan Baku: Admin dapat melakukan pencarian terhadap bahan baku yang ingin dicari untuk melihat stok dan harga dari daftar bahan baku tersebut.
- Tutup (Notifikasi): Admin akan menerima notifikasi otomatis di akhir hari (penutupan kafe) jika ada bahan baku yang melewati batas minimum stok.

1.3 *Role and Privileges*

Berdasarkan diagram flowchart pada gambar 1.1 tersebut menunjukkan proses pengelolaan menu, harga, dan stok bahan baku yang melibatkan satu entitas yaitu Admin.

- Role Admin: Role ini memiliki hak akses penuh program untuk mengelola menu, harga, dan mengelola bahan baku. Admin dapat menambah, mengubah, menghapus, dan melihat data tentang menu, harga, dan stok.

1.4 Asumsi dan Validasi

- Admin dapat menerima informasi mengenai bahan baku yang dipesan.
- Tipe data yang digunakan pada kategori, sub kategori adalah string untuk penamaan kategori dan subkategori, sedangkan item menggunakan tipe data string untuk nama item, tipe data integer untuk harga item, dan jumlah stok item. Jika admin menginput tidak sesuai dengan tipe data yang diberikan, maka program akan menampilkan silahkan coba lagi dengan tipe data yang benar.
- Harga dan stok makanan tidak bisa diinput jika item tidak ada.
- Pada program kami, sistem akan mengurangi stok bahan baku jika terjadi pembelian.
- Pada program kami, sistem akan mengurangi jumlah stok item jika terjadi pembelian.
- Pada program kami, menu akan tersedia secara langsung di program.
- Stok bisa ditambah dan diubah oleh admin.
- Jika salah satu jumlah stok bahan makanan kosong, maka makanan yang dipesan tidak bisa dibuat.
- Jika tidak ada bahan baku yang kurang dari minimum maka notifikasi tersebut tidak muncul.

BAB II

RANCANGAN PROGRAM

2.1 Struktur Data

Merujuk pada proses bisnis yang tergambar melalui spesifikasi program pada Bab I, rancangan program yang akan dikembangkan dapat dibagi menjadi dua bagian utama, yakni:

- Proses pengelolaan menu dan harga
- Proses pengelolaan bahan baku

Untuk mencapai keberhasilan dalam pengembangan program, harus memperhatikan pemilihan penggunaan struktur data yang tepat agar program yang dibuat dapat berjalan dengan efisien serta memiliki performance yang baik. Berikut ini akan diuraikan penggunaan struktur data pada masing-masing bagian tersebut.

2.1.1 Proses Pengelolaan Menu dan Harga

Pada bagian ini, pengelolaan menu dan harga meliputi penambahan, penghapusan, update dan lihat menu yang biasa dikenal dengan CRUD (*Create, Read, Update, Delete*). Struktur data yang dipakai dalam proses ini adalah struktur data ***Non-Binary Tree***.

Pemilihan struktur data ***Non-Binary Tree*** untuk mengelola menu dan harga didasarkan pada beberapa pertimbangan dan kebutuhan spesifik dari proses bisnis yang dijalankan oleh program ini. Berikut adalah beberapa alasan mengapa struktur data ***non-binary tree*** dipilih:

1. Hierarki Menu:

- Menu kafe seringkali memiliki struktur hierarki, di mana menu terbagi menjadi berbagai kategori dan subkategori. ***Non-binary tree*** memungkinkan representasi yang alami untuk menyajikan hierarki semacam itu.
- Setiap node dalam ***tree*** dapat mewakili kategori atau subkategori menu, dengan cabang-cabangnya merepresentasikan item-menu di bawahnya.

2. Fleksibilitas:

- ***Non-binary tree*** memberikan fleksibilitas dalam menangani perubahan dan penambahan menu. Penambahan kategori, subkategori, atau item-menu baru dapat dilakukan tanpa memerlukan perubahan besar pada keseluruhan struktur.
- Struktur ***tree*** memberikan kemampuan untuk mengorganisasi dan mengelompokkan menu dengan cara yang intuitif dan mudah dipahami.

3. Pencarian Efisien:

- Struktur *tree* memungkinkan pencarian menu berdasarkan kategori, subkategori, atau nama menu menjadi lebih efisien. Proses pencarian dapat dilakukan dengan kompleksitas waktu yang lebih rendah.

4. Representasi Visual yang Jelas:

- Tree menyediakan representasi visual yang jelas dan mudah dipahami untuk struktur hierarki menu. Hal ini membuat pengelolaan menu menjadi lebih intuitif, terutama ketika menu memiliki banyak kategori dan subkategori.

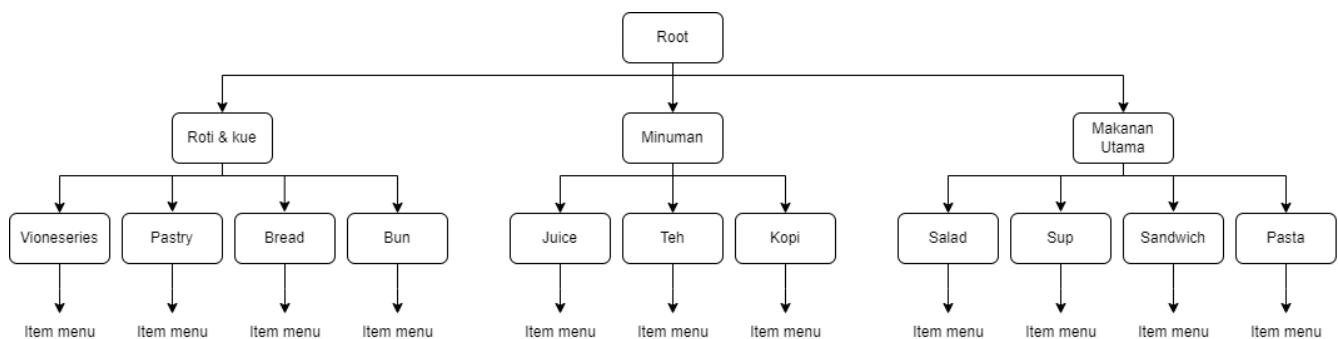
5. Pembaruan Harga dan Menu:

- *Non-binary tree* memudahkan operasi pembaruan harga dan menu. Pembaruan dapat dilakukan dengan mudah pada node-node tertentu tanpa mempengaruhi seluruh struktur, memberikan efisiensi operasional.

6. Pengelolaan Event atau Waktu Makan:

- Menu yang berubah berdasarkan event atau waktu makan dapat diakomodasi dengan baik oleh struktur *tree*. Setiap event dapat diasosiasikan dengan kategori atau subkategori tertentu dalam *tree*.

Dengan demikian, *non-binary tree* dipilih untuk menyajikan menu dan harga karena kemampuannya dalam memodelkan hierarki menu dan menyediakan operasi pengelolaan menu yang efisien dan fleksibel. Berikut ini adalah representasi visual dari struktur menu yang dapat dilihat dari gambar berikut ini:



Setiap child di bawah kategori utama memiliki node yang mewakili subkategori, dan setiap subkategori memiliki node yang mewakili item atau produk. Dalam implementasinya ke kodingan, kami menggunakan *Array / Dynamic node* dan untuk

menangani perubahan jumlah anak/child menggunakan ***Dynamic Array*** / array dinamis (vector). Dalam program ini struct digunakan untuk menampung variabel-variabel yang diperlukan untuk membuat fitur-fitur yang akan diimplementasikan seperti menyimpan informasi menu, informasi bahan baku, menyimpan data event dan sebagainya. Berikut ini adalah penggunaanya masing-masing.

a). Struct ItemMenu:

Digunakan untuk menyimpan informasi item menu seperti nama menu, harga, dan stok:

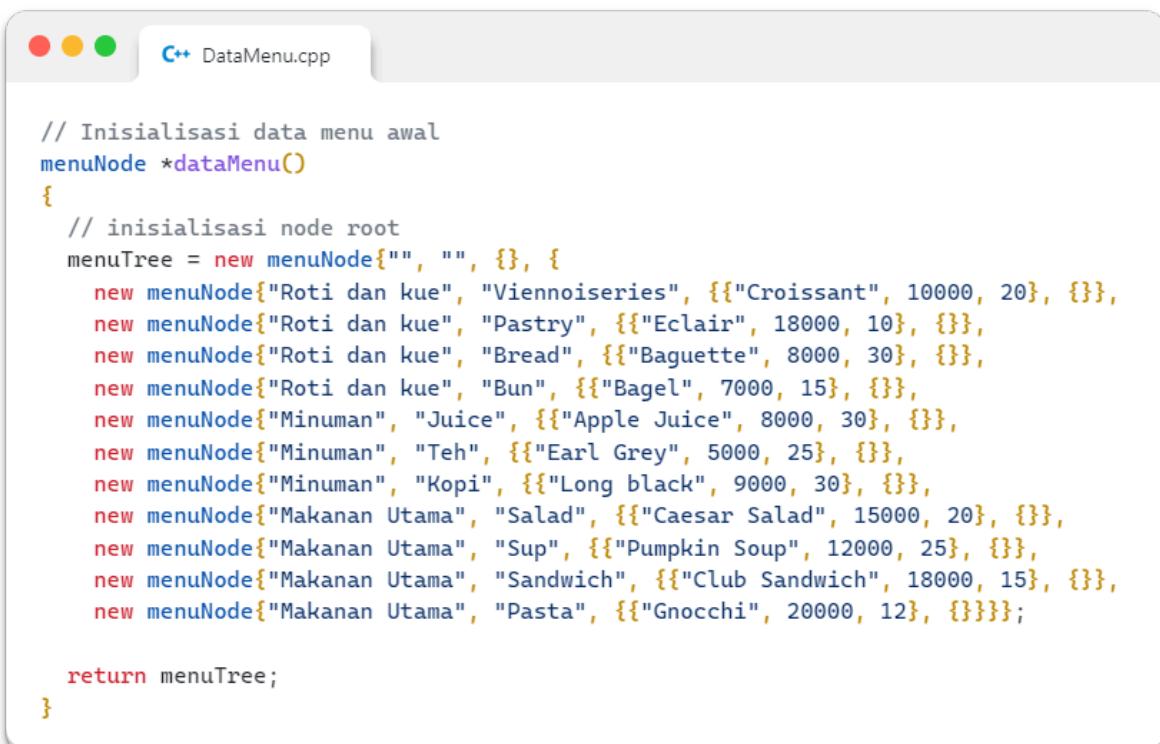
- namaMenu: string
- harga : int
- stok : int

b) Struct Event

Digunakan untuk menyimpan informasi event yang terdiri dari:

- name : string
- categories : vector<string>
- subcategories : vector<string>

Berikut ini adalah gambaran sederhana implementasi struktur menu ke dalam kodingan menggunakan *dynamic array* melalui inisialisasi data menu awal di dalam fungsi **dataMenu()**:



The screenshot shows a code editor window with a tab labeled "C++ DataMenu.cpp". The code itself is as follows:

```
// Inisialisasi data menu awal
menuNode *dataMenu()
{
    // inisialisasi node root
    menuTree = new menuNode{ "", "", {}, {
        new menuNode{"Roti dan kue", "Viennoiseries", {{"Croissant", 10000, 20}, {}},
        new menuNode{"Roti dan kue", "Pastry", {{"Eclair", 18000, 10}, {}},
        new menuNode{"Roti dan kue", "Bread", {{"Baguette", 8000, 30}, {}},
        new menuNode{"Roti dan kue", "Bun", {{"Bagel", 7000, 15}, {}},
        new menuNode{"Minuman", "Juice", {{"Apple Juice", 8000, 30}, {}},
        new menuNode{"Minuman", "Teh", {{"Earl Grey", 5000, 25}, {}},
        new menuNode{"Minuman", "Kopi", {{"Long black", 9000, 30}, {}},
        new menuNode{"Makanan Utama", "Salad", {{"Caesar Salad", 15000, 20}, {}},
        new menuNode{"Makanan Utama", "Sup", {{"Pumpkin Soup", 12000, 25}, {}},
        new menuNode{"Makanan Utama", "Sandwich", {{"Club Sandwich", 18000, 15}, {}},
        new menuNode{"Makanan Utama", "Pasta", {{"Gnocchi", 20000, 12}, {}}}};

    return menuTree;
}
```

Struktur dari tree menu yang diinisialisasi oleh fungsi **dataMenu** adalah sebagai berikut:

- Root Node (Level 0):
 - **kategori**: "" (kosong)
 - **subKategori**: "" (kosong)
 - **items**: {}
 - **children**: [Node Roti dan Kue, Node Minuman, Node Makanan Utama]

- Node Roti dan Kue (Level 1):
 - **kategori**: "Roti dan kue"
 - **subKategori**: "Viennoiseries"
 - **items**: [Croissant, Pain au Chocolate, Brioche]
 - **children**: {}

- Node Minuman (Level 1):
 - **kategori**: "Minuman"
 - **subKategori**: "Juice"
 - **items**: [Apple Juice, Orange Juice, Strawberry Juice]
 - **children**: [Node Teh, Node Kopi]

- Node Makanan Utama (Level 1):
 - **kategori**: "Makanan Utama"
 - **subKategori**: "Salad"
 - **items**: [Caesar Salad, Caprese Salad, Waldorf Salad, Nicoise Salad]
 - **children**: [Node Sup, Node Sandwich, Node Pasta]

Setiap node pada level 1 memiliki anak yang merupakan node pada level 2. Begitu seterusnya. Di setiap level, node dapat memiliki kategori (`**kategori**` dan `**subKategori**`), daftar item (`**items**`), dan anak-anak (`**children**`). Node pada level terendah (misalnya, node yang memiliki item seperti Croissant atau Apple Juice) tidak memiliki anak (**children**).

Dalam program ini, data menu dan data event memiliki hubungan yang erat. Data menu disimpan dalam struktur *non-binary tree* yang terdiri dari node-node yang mewakili kategori, subkategori, dan item-menu. Di sisi lain, data event disimpan menggunakan

array dinamis yang dalam implementasinya menggunakan vektor yang menyimpan informasi tentang nama event dan kategori-kategori yang terkait dengan setiap event.

Hubungan antara data menu dan data event terletak pada kategori dan subkategori menu yang dapat dihubungkan dengan kategori yang terkait dengan setiap event. Sebagai contoh, pada event "Breakfast", kategori-kategori yang terkait adalah "Sandwich", "Bread", "Bun", dan "Minuman". Program dapat menggunakan informasi ini untuk menampilkan menu yang terkait dengan setiap event.

Dalam implementasinya, program dapat melakukan pencarian menu berdasarkan kategori dan subkategori yang terkait dengan event tertentu. Sebagai contoh, jika sedang menampilkan menu untuk event "Breakfast", program dapat mencari dan menampilkan menu-menu yang termasuk dalam kategori-kategori yang terkait dengan event tersebut.

Jadi, data menu dan data event saling terhubung melalui kategori dan subkategori, memungkinkan program untuk menyajikan menu yang spesifik untuk setiap event.

2.1.1 Proses Pengelolaan Bahan Baku

Proses pengelolaan bahan baku yang dibuat dalam program ini meliputi fitur lihat bahan baku, tambah bahan baku, update bahan baku(harga dan stok) dan dilengkapi dengan fitur cari bahan baku untuk memudahkan dalam menemukan ketersediaan bahan baku.

Pada bagian ini kami memilih menggunakan struktur data **hash table** untuk menyimpan data bahan baku karena struktur data ini memiliki beberapa keuntungan tertentu, terutama dalam konteks manajemen stok dan pencarian yang efisien. Berikut adalah alasan-alasan utama memilih hash table untuk tujuan ini:

1. Pencarian Cepat

Hash table menyediakan pencarian konstan ($O(1)$) pada kasus rata-rata. Hal ini berarti ketika kita ingin mencari informasi tentang suatu bahan baku, waktu yang dibutuhkan tidak bergantung pada jumlah bahan baku yang ada.

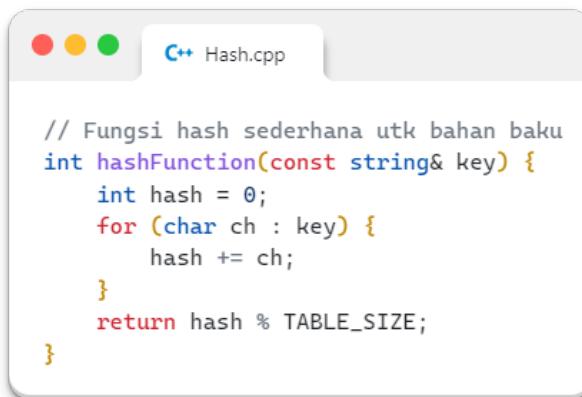
2. Manajemen Stok yang Efisien

Hash table memungkinkan kita dengan mudah mengakses dan memperbarui informasi stok bahan baku. Operasi seperti penambahan, pengurangan, atau pembaruan stok dapat dilakukan dengan efisien.

3. Keberlanjutan Waktu yang Konsisten

Waktu yang diperlukan untuk operasi pada hash table cenderung konsisten, terlepas dari jumlah elemen yang ada. Hal ini berguna untuk memastikan responsivitas aplikasi dalam situasi stok yang besar.

Hashing digunakan untuk mengindex dan mendapatkan kembali key pada hash table, karena lebih cepat untuk mengambil key yang sudah di hash daripada mencarinya menggunakan original value. Selanjutnya hash function dibuat untuk mentransformasikan string menjadi bilangan yang akan menjadi index pada hash table. Berikut ini adalah fungsi hash sederhana untuk mengonversi string (nama bahan baku) menjadi nilai hash.



```
// Fungsi hash sederhana utk bahan baku
int hashFunction(const string& key) {
    int hash = 0;
    for (char ch : key) {
        hash += ch;
    }
    return hash % TABLE_SIZE;
}
```

Fungsi hash tersebut adalah metode sederhana untuk mengonversi string (nama bahan baku) menjadi nilai hash. Setiap karakter dalam string diubah menjadi nilai ASCII, yang kemudian diakumulasikan. Hasil akhirnya diambil modulo dengan ukuran hash table (TABLE_SIZE = 500). Hal ini dilakukan untuk memastikan bahwa nilai hash tetap dalam rentang indeks yang valid untuk hash table.

Ilustrasi penyimpanan bahan baku pada hash table dapat dilihat seperti berikut ini. Misalnya kita asumsikan hasil hashing untuk "Tepung" adalah indeks 3. Maka, pada indeks 3 dari hash table, akan ada satu node (atau lebih jika terjadi *collision*) yang berisi informasi tentang bahan baku "Tepung", begitu dan seterusnya berlaku untuk setiap string bahan baku.

Hash Table:

```
Index 0: [ Susu - Harga: 10000.0 - Stok: 500 ]
Index 1: [ Terigu - Harga: 5000.0 - Stok: 500 ]
Index 2: [ Gula - Harga: 3000.0 - Stok: 500 ]
Index 3: [ Tepung - Harga: 5000.0 - Stok: 100 ]
```

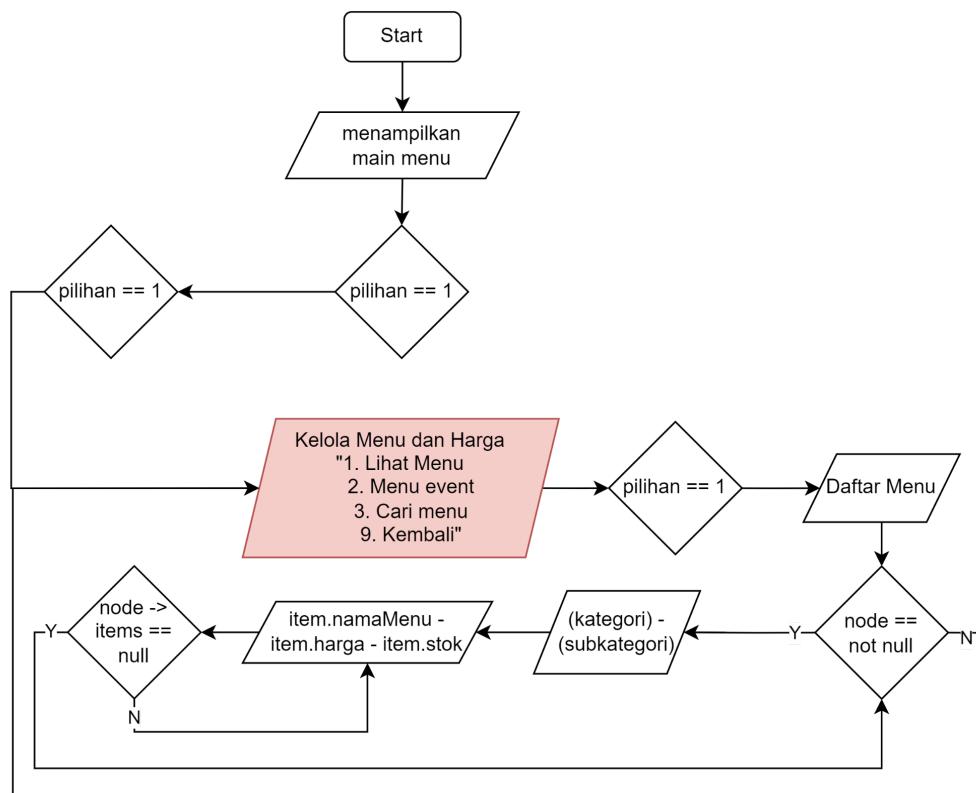
Setiap bahan baku memiliki entri pada indeks hash table yang dihasilkan dari nama bahan baku tersebut.

2.2 Flowchart

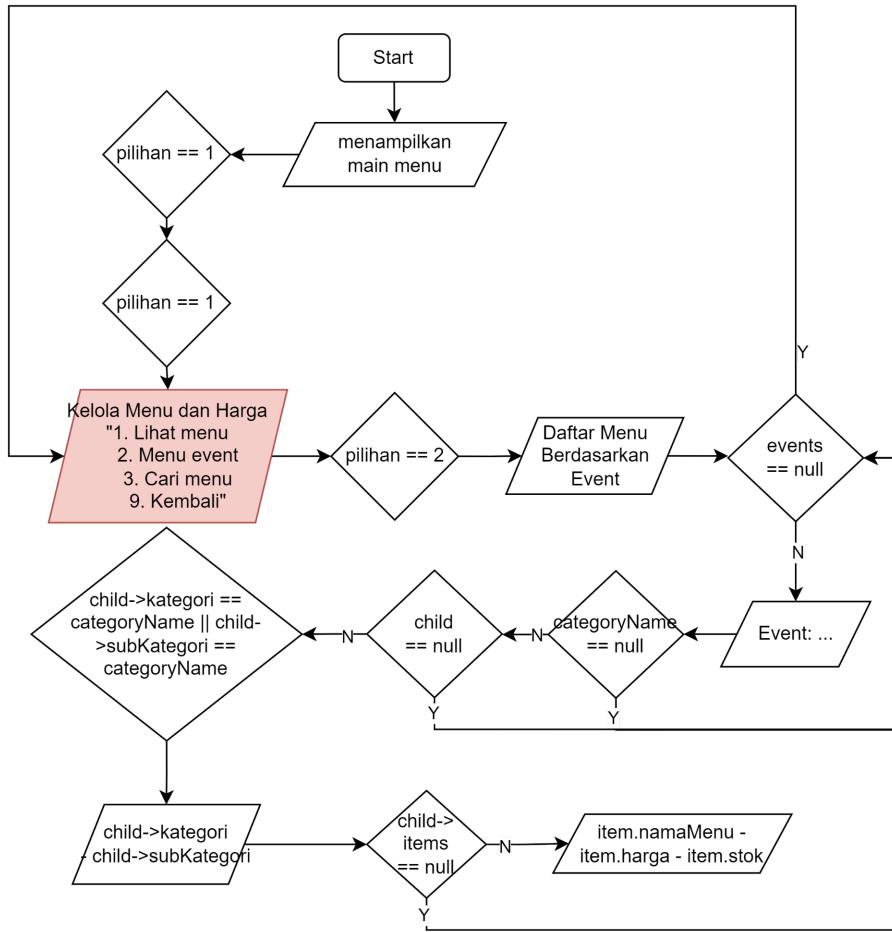
2.2.1 Kelola menu dan harga

2.2.1.1 Lihat menu

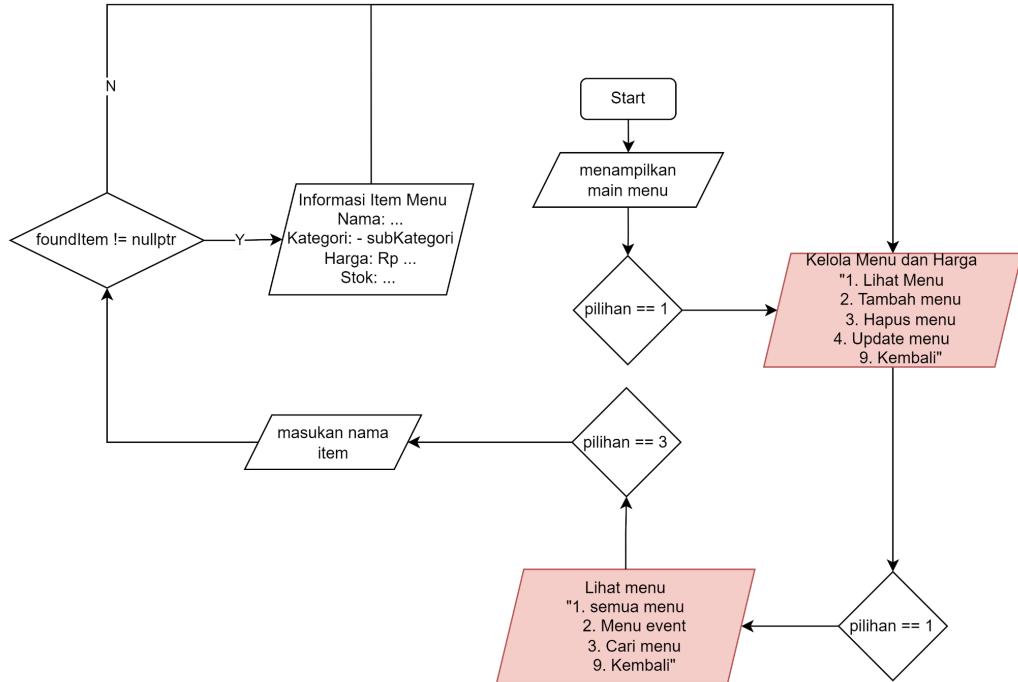
a. Semua menu



b. Menu event

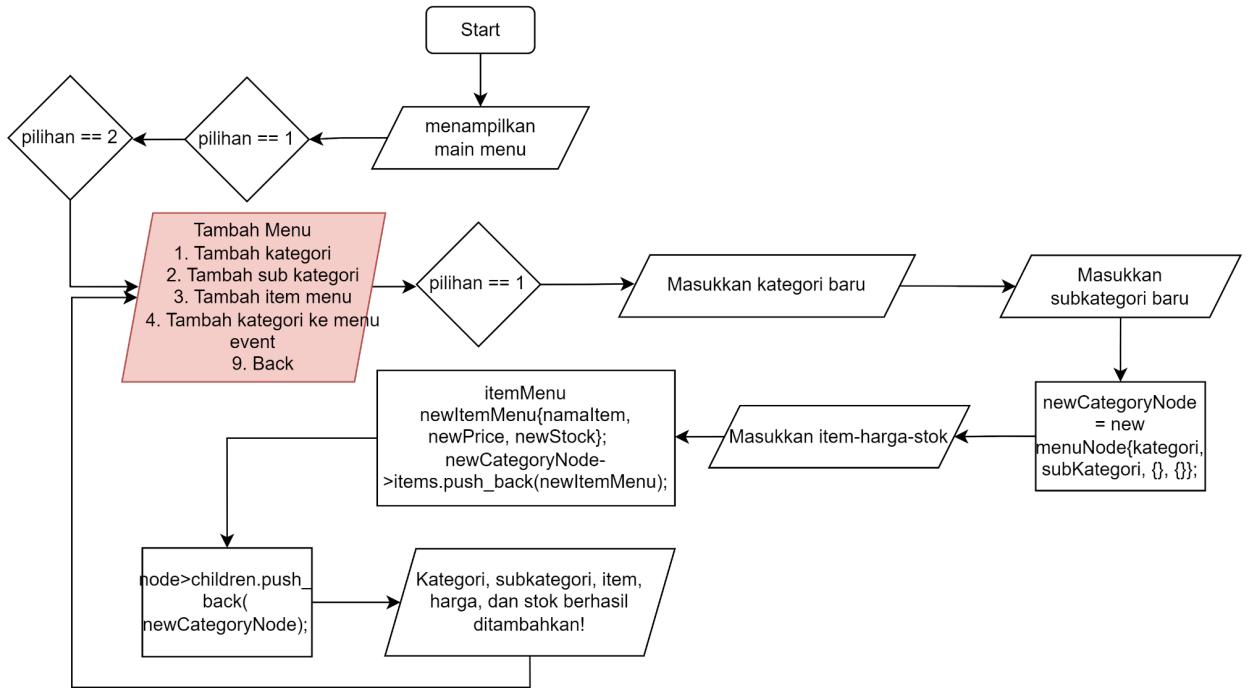


c. Cari menu

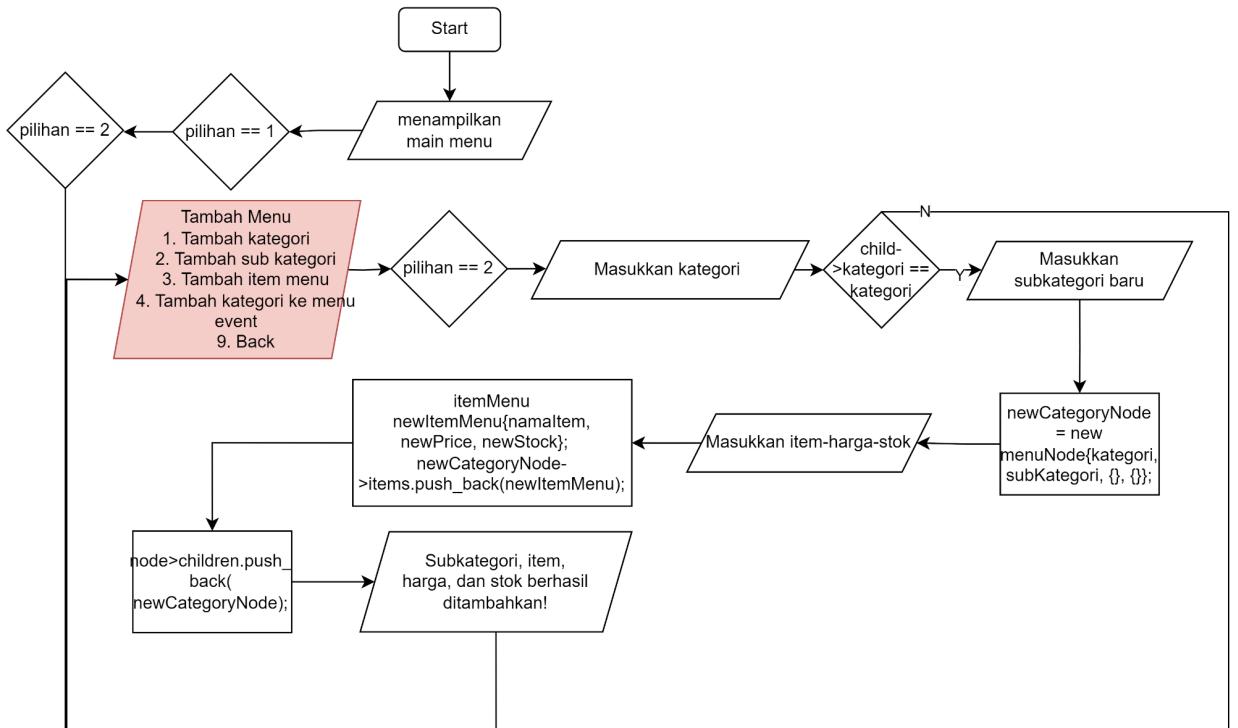


2.2.1.2 Tambah menu

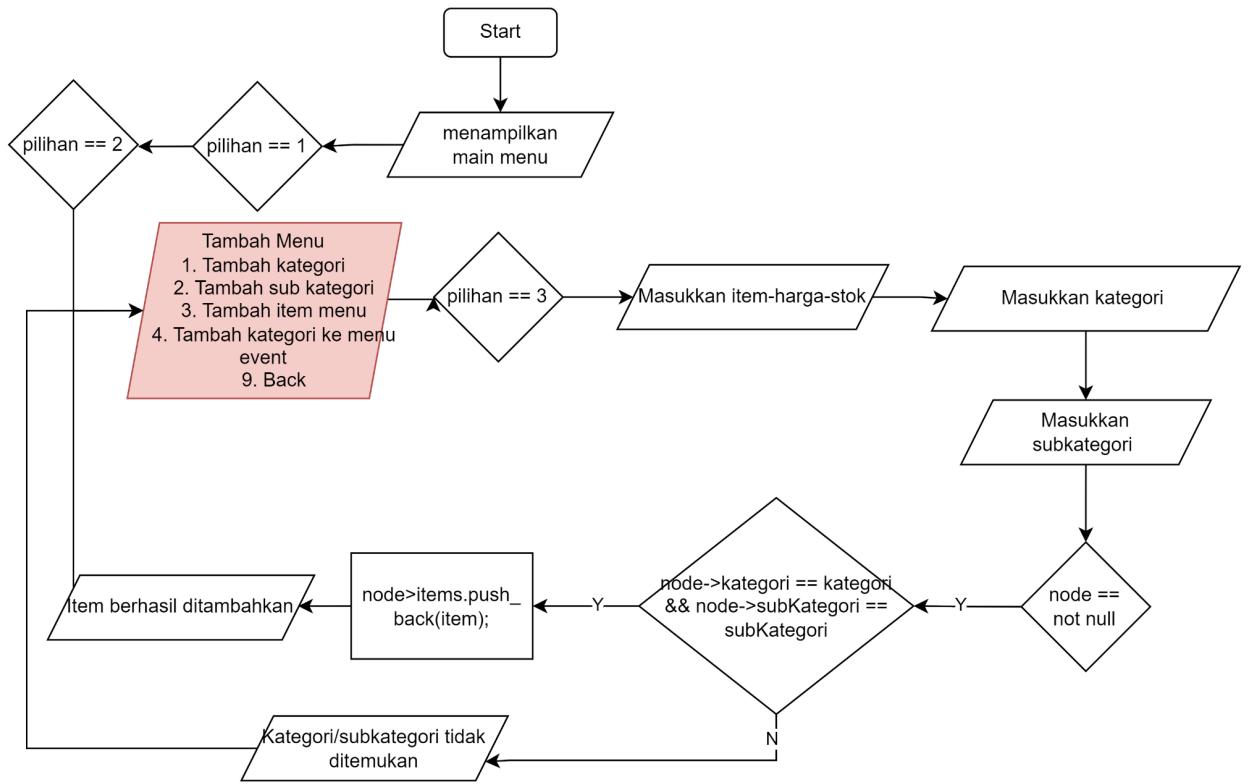
a. Tambah kategori



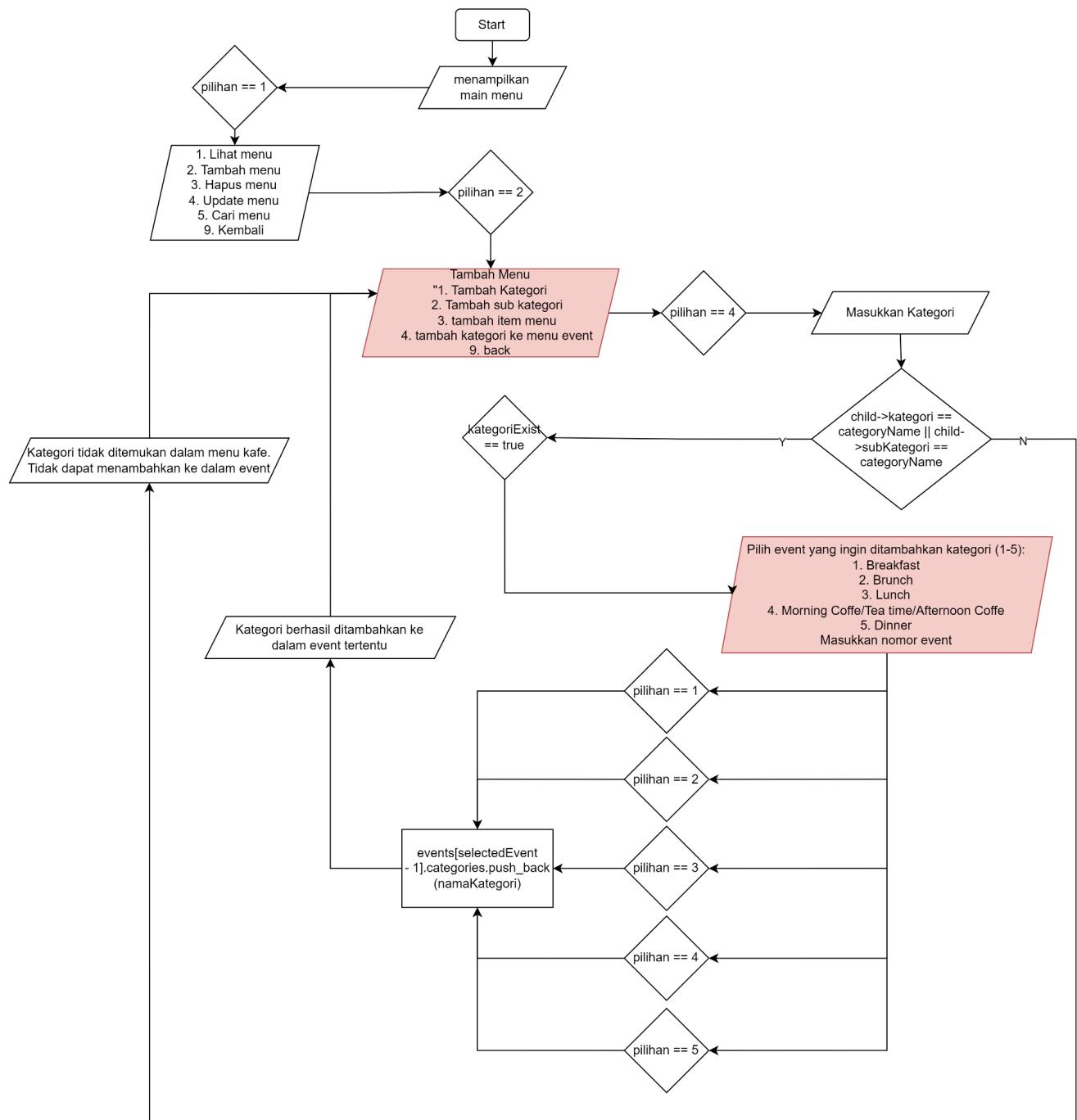
b. Tambah sub kategori



c. Tambah item menu

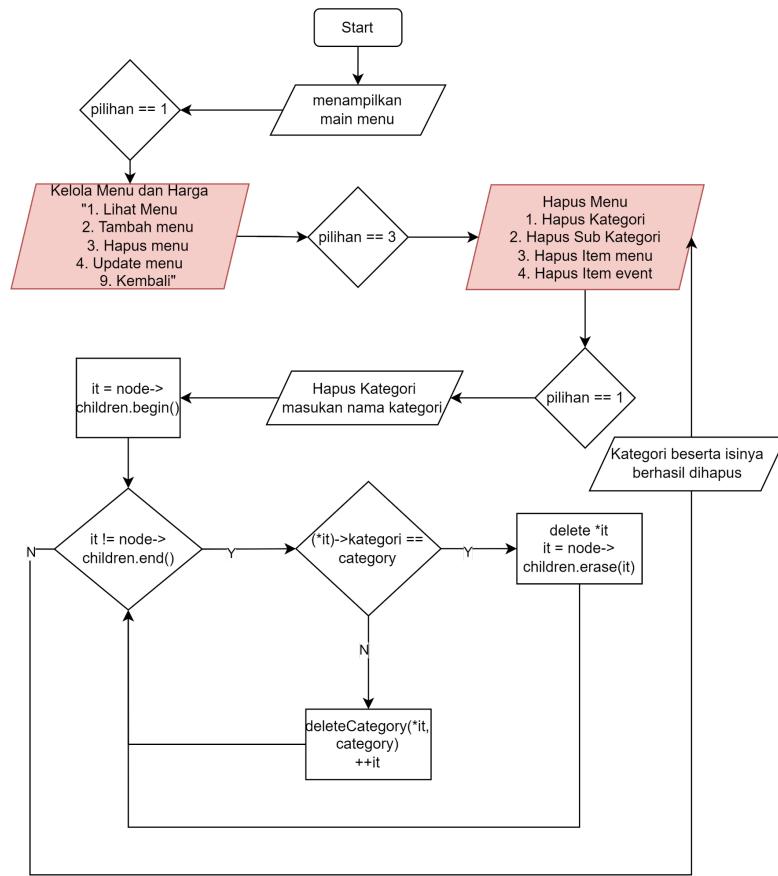


d. Tambah kategori/sub kategori ke menu event

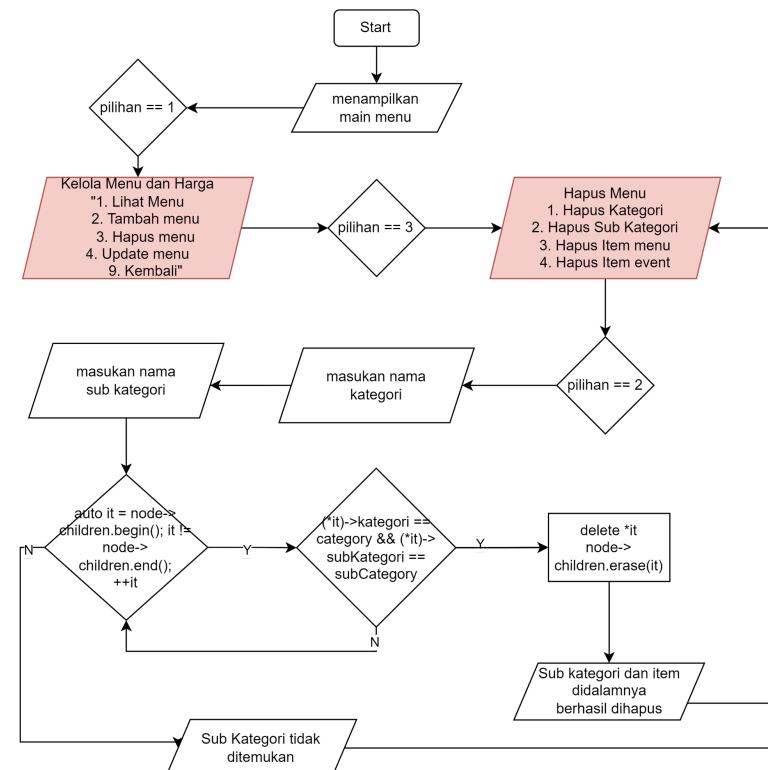


2.2.1.3 Hapus menu

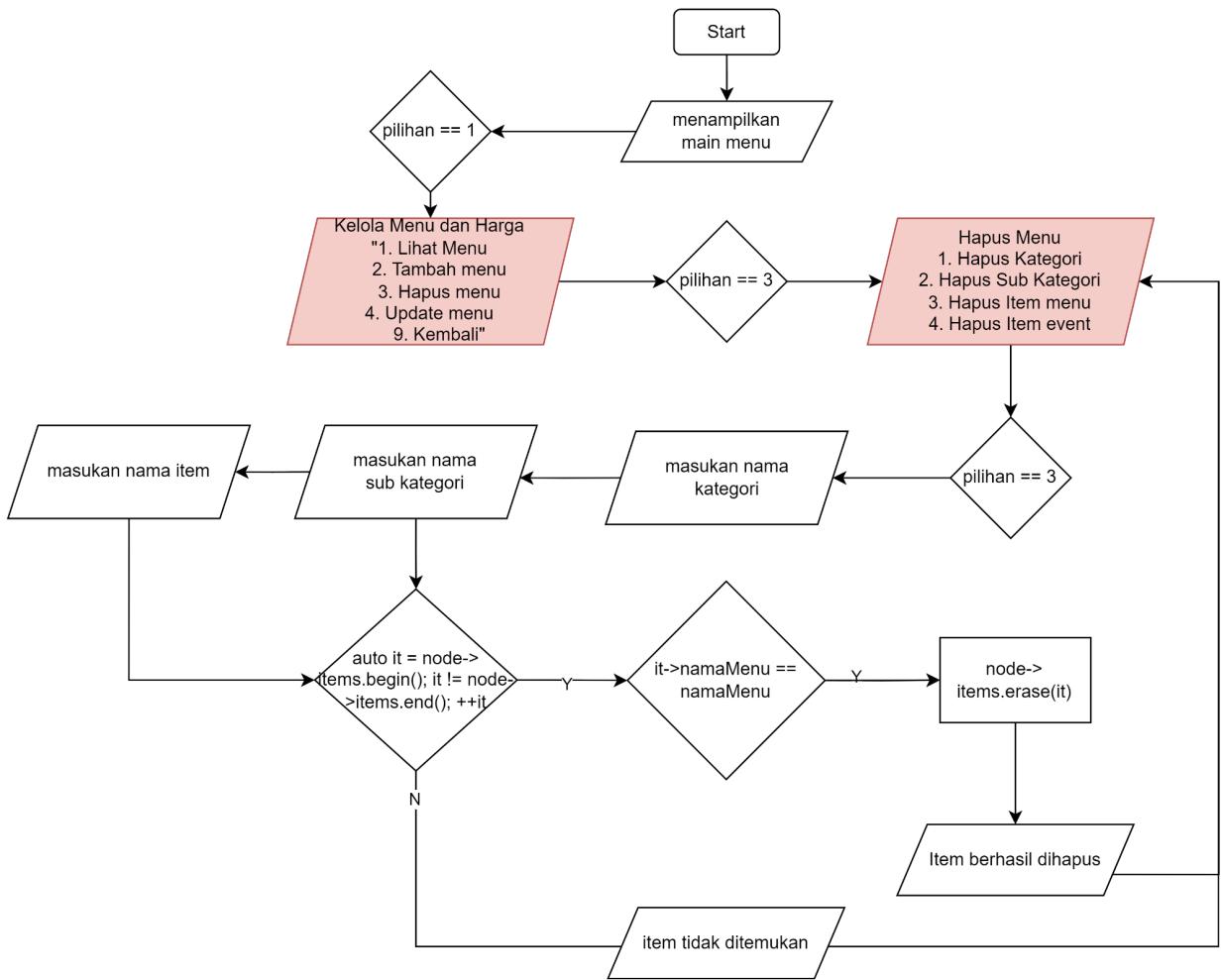
a. Hapus kategori



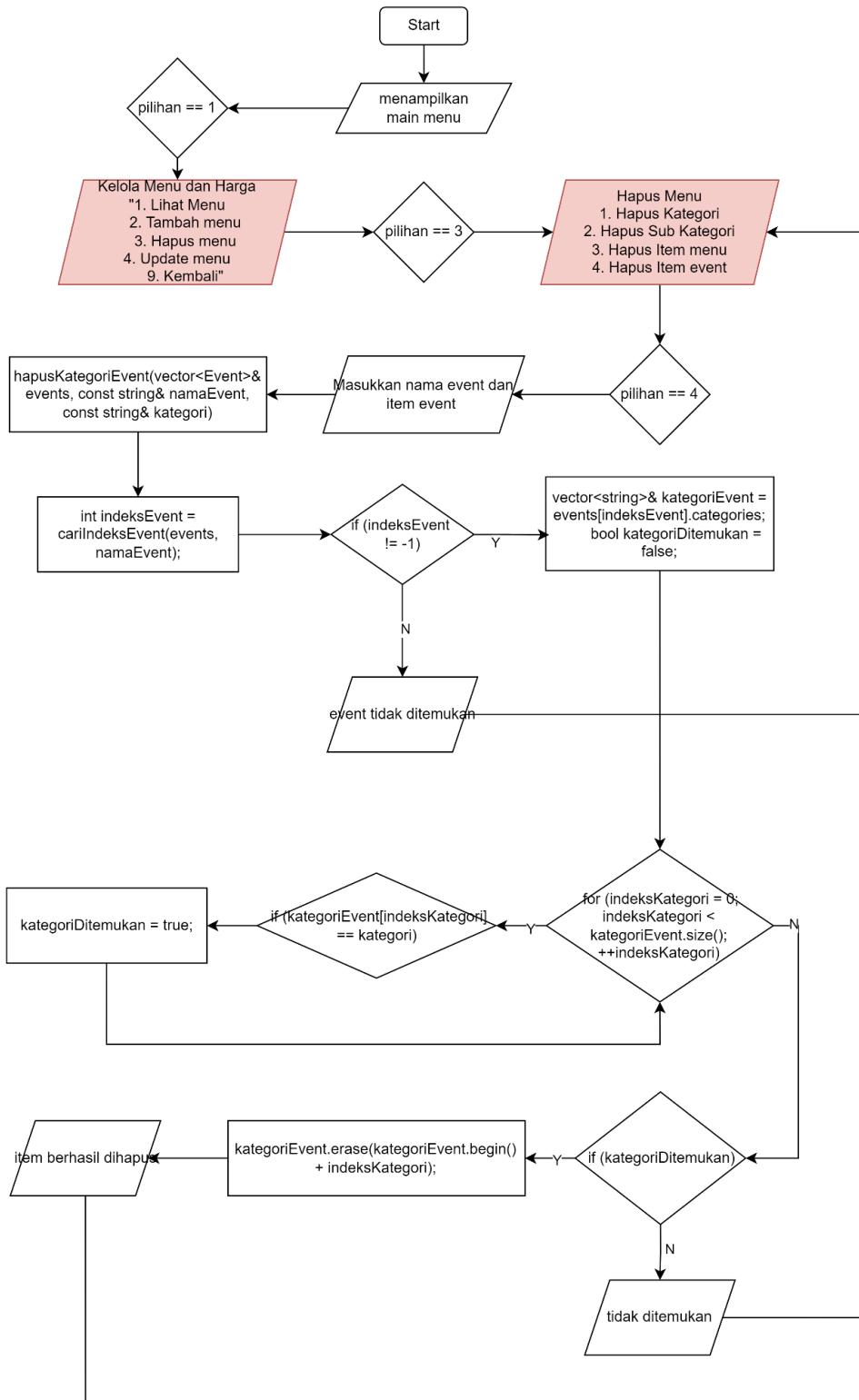
b. Hapus sub kategori



c. Hapus item menu

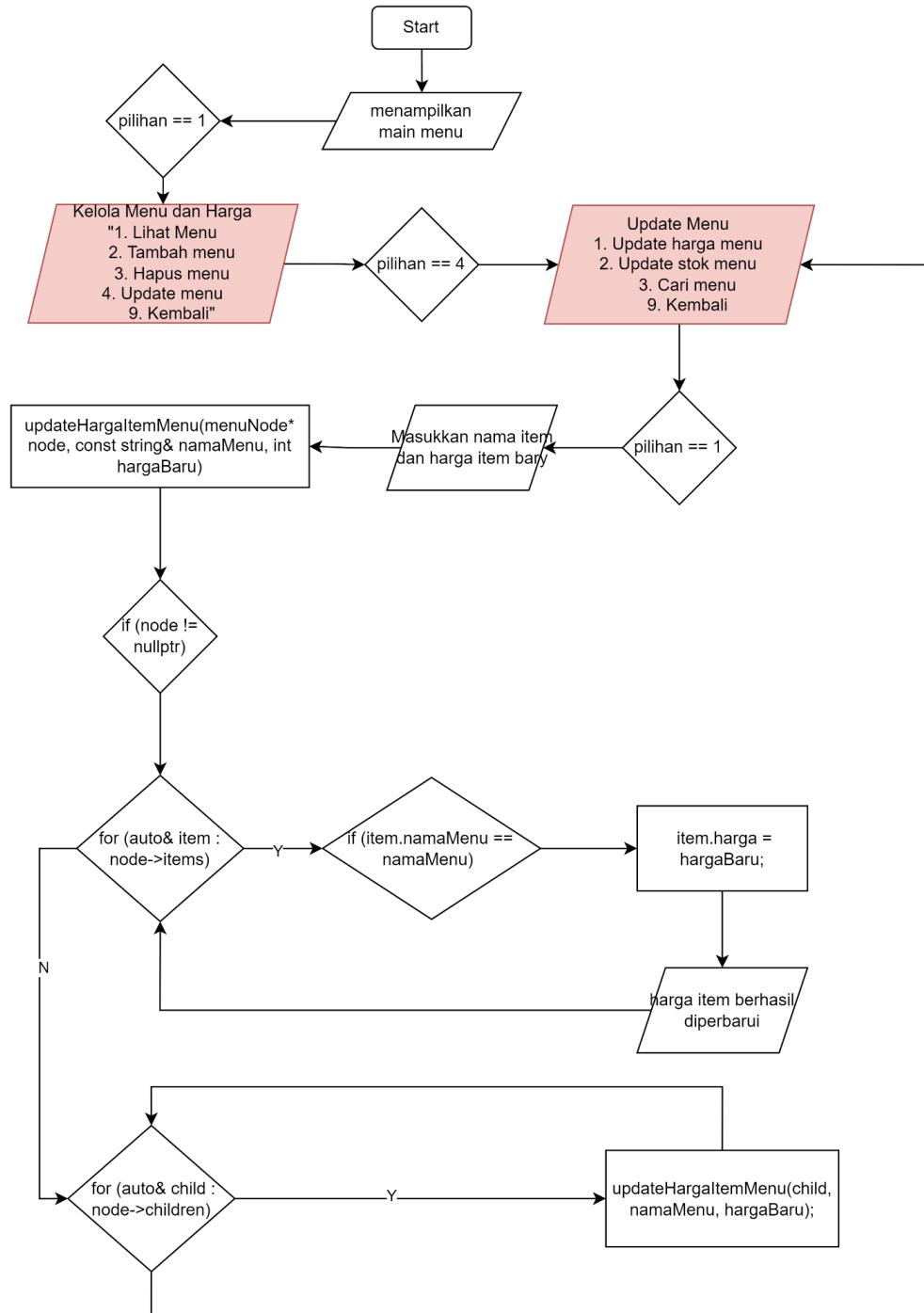


d. Hapus item event

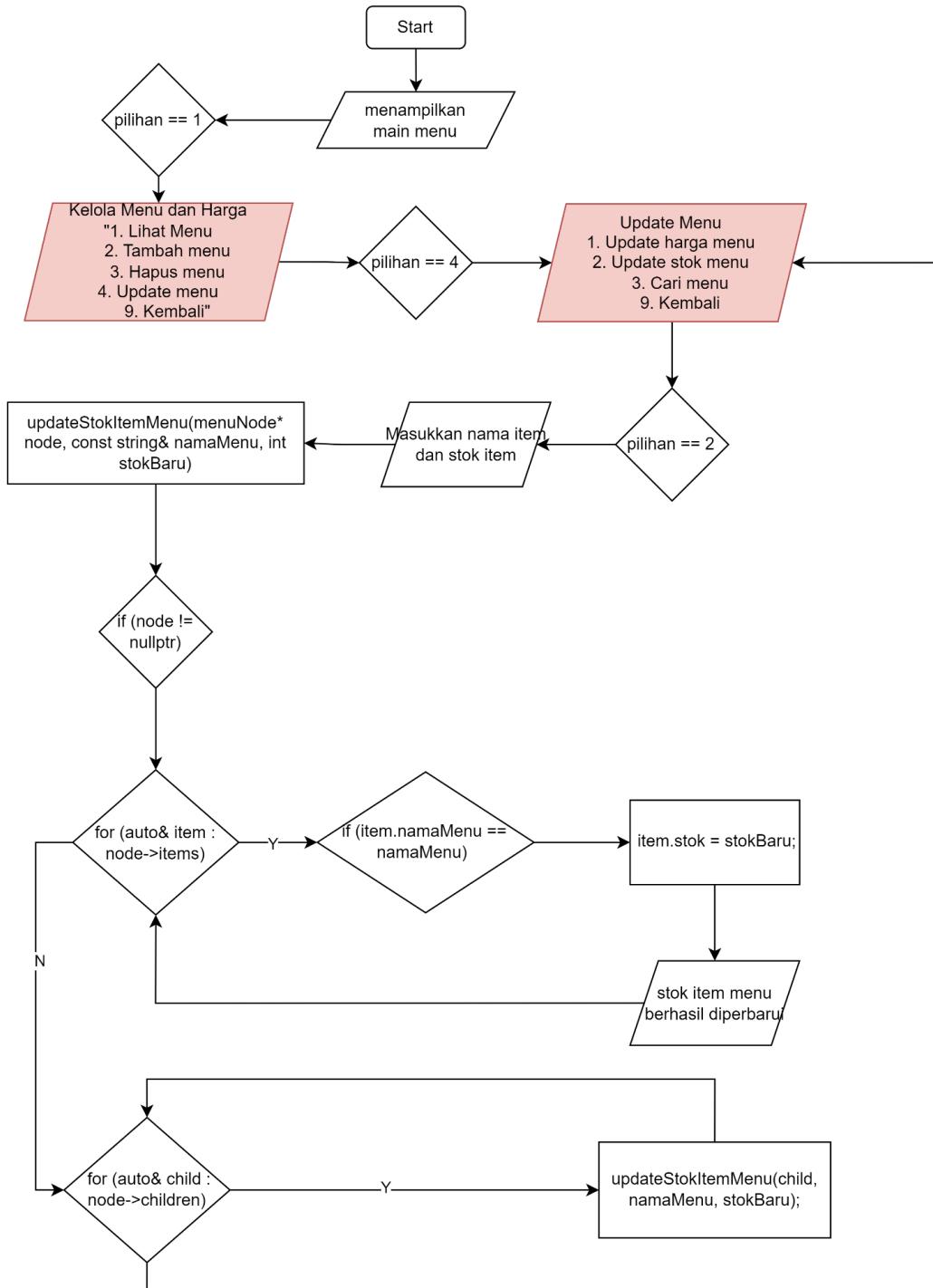


2.2.1.4 Update menu

a. Update harga menu



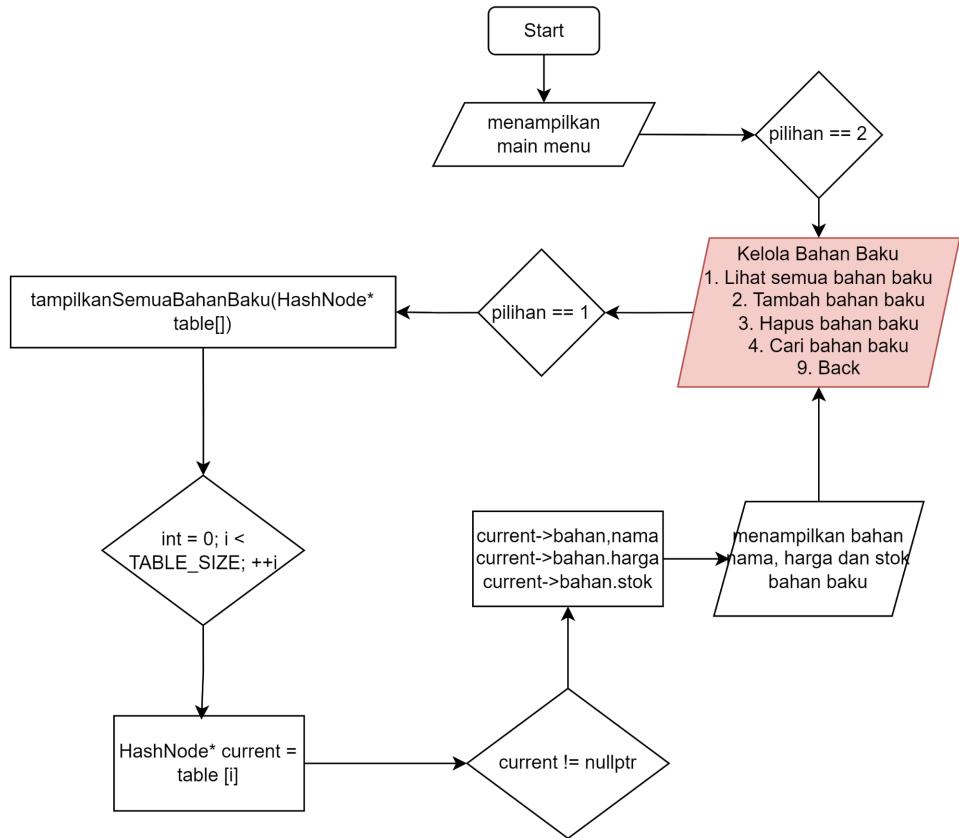
b. Update stok menu



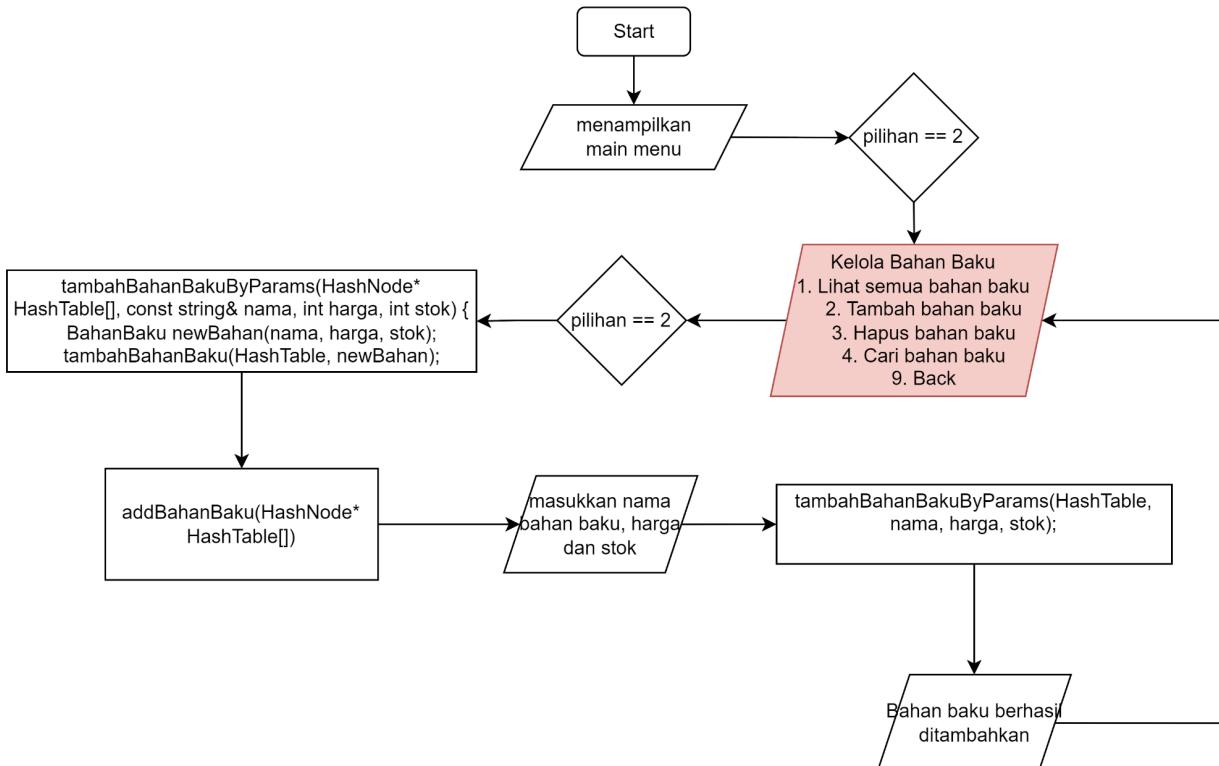
c. Cari menu (sama dengan 2.2.1.1 c)

2.2.2 Kelola bahan baku

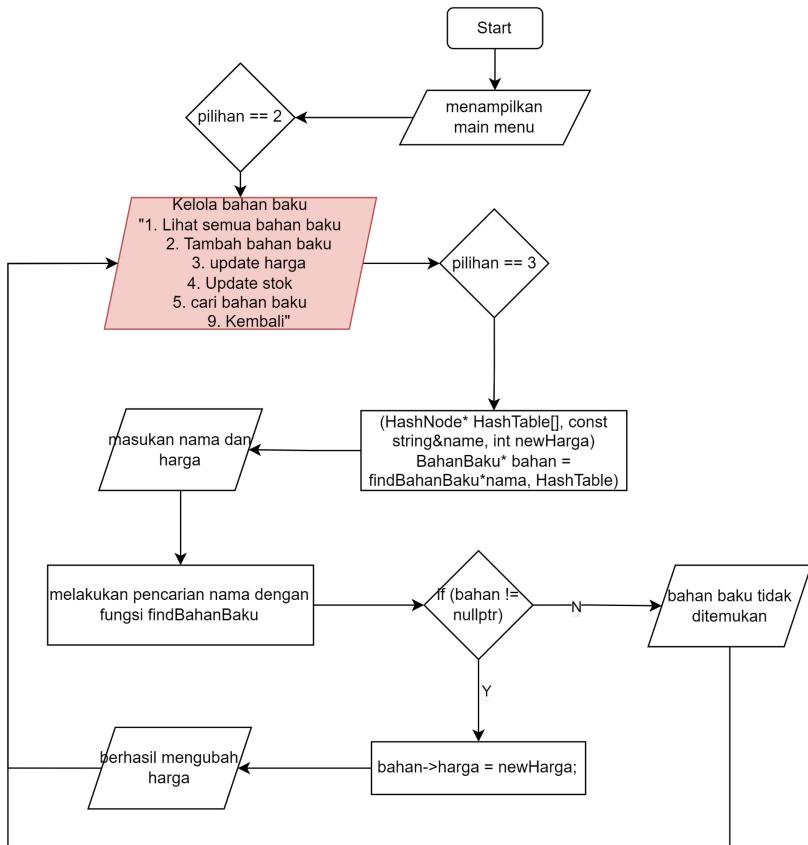
2.2.2.1 Lihat semua bahan baku



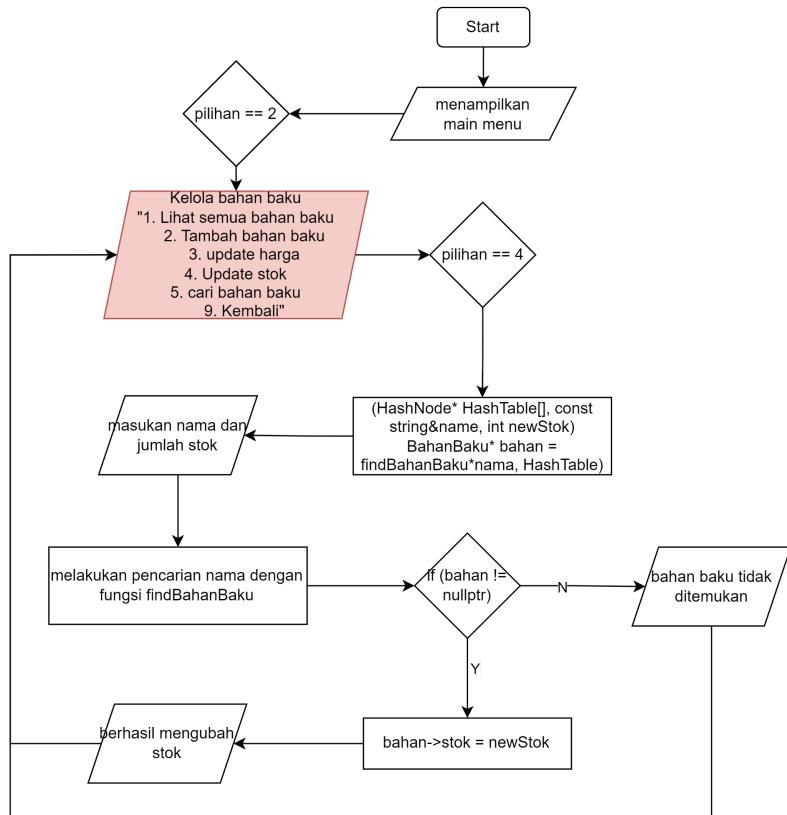
2.2.2.2 Tambah bahan baku



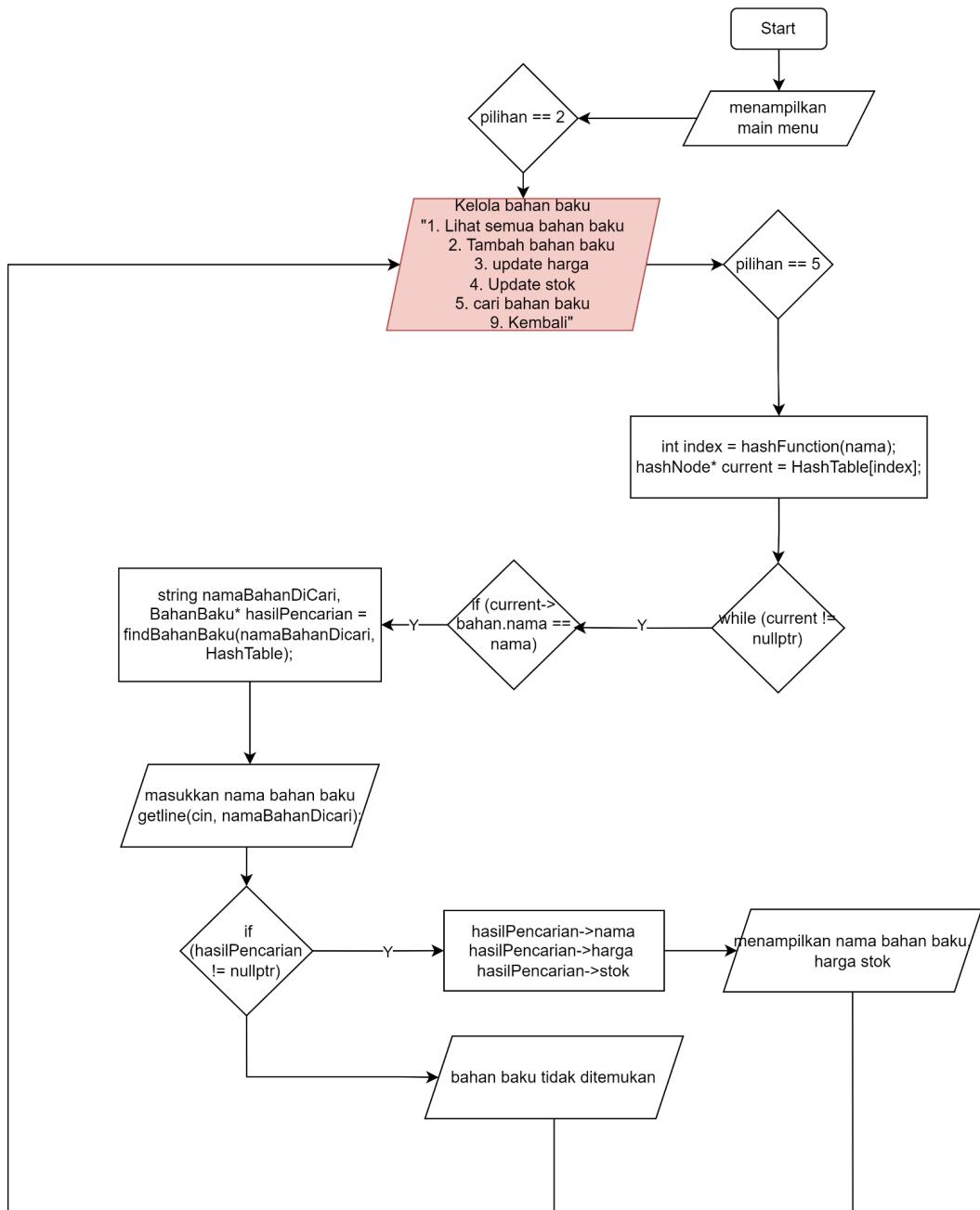
2.2.2.3 Update harga



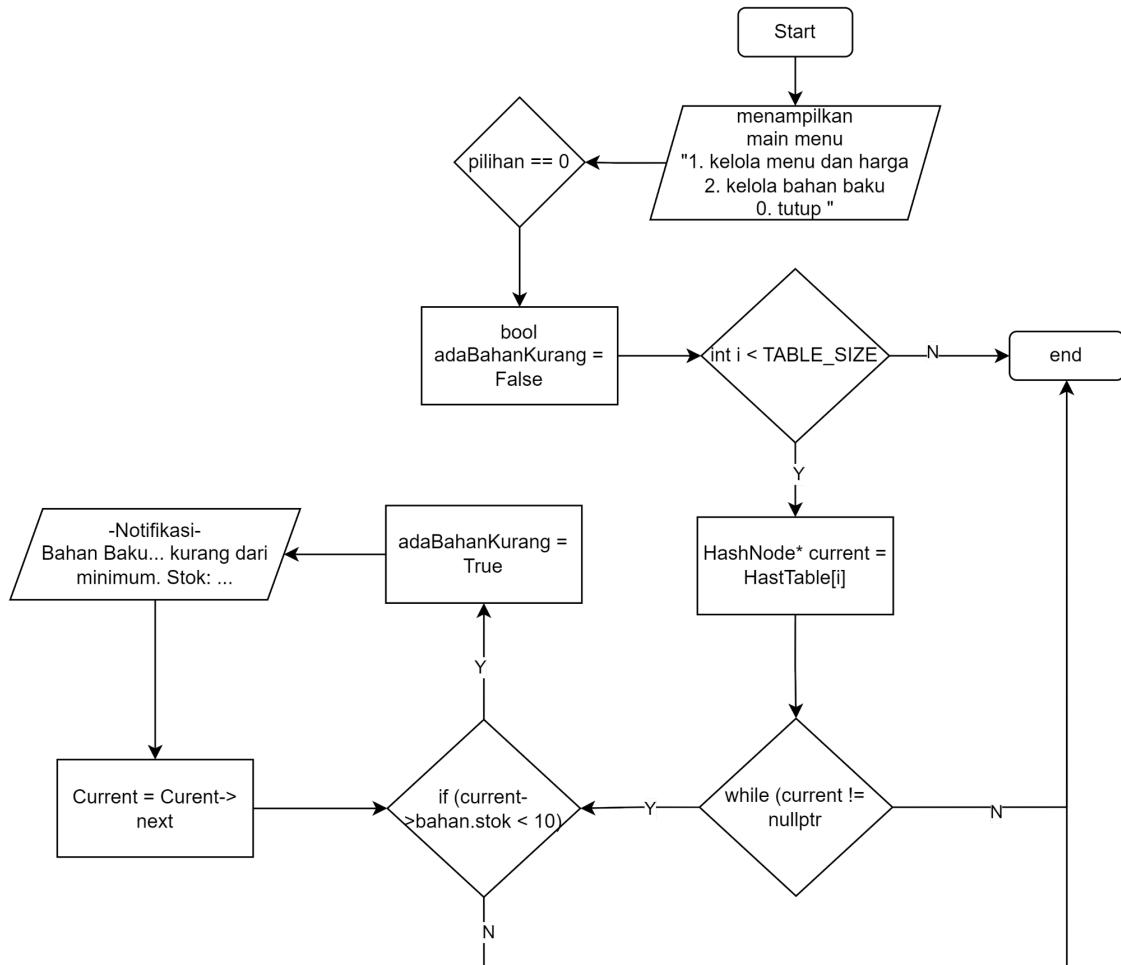
2.2.2.4 Update stok



2.2.2.5 Cari bahan baku



2.2.3 Notifikasi



2.3 Struktur Program

| No | Nama subprogram | Deskripsi | Input Parameter | Output Parameter | Kamus Data (local variabel) |
|----|-----------------------|---|-----------------|------------------|-----------------------------|
| 1 | Fungsi: dataMenu | merupakan fungsi untuk menginisialisasi data menu awal pada kafe(fungsi ini bertipe *menuNode). | - | - | -menuTree -menuNode |
| 2 | Fungsi: initEvents | digunakan untuk menginisialisasi data event yang berkaitan | - | - | - |

| | | | | | |
|---|--------------------------------------|--|---|--|---|
| | | dengan kategori menu pada kafe. | | | |
| 3 | Fungsi: hashFunction | merupakan implementasi dari fungsi hash sederhana yang digunakan untuk menghasilkan nilai hash dari suatu string (key) dlm hal ini bahan baku | -const string& key | -int: mengembalikan nilai hash berupa integer yang merupakan hasil dari proses hash terhadap string input | -int hash |
| 4 | Prosedur: tambahBahanBaku | bertujuan untuk menambahkan bahan baku ke dalam tabel hash (hashtable). Jika bahan baku dengan nama yang sama sudah ada dalam hashtable, maka stok dan harga bahan baku tersebut akan diperbarui. Jika belum ada, bahan baku baru akan ditambahkan ke dalam hashtable. | - hashNode* hashTable[] - const BahanBaku& bahan | Tidak ada (void). Fungsi ini tidak mengembalikan nilai. | - int index - hashNode* current - hashNode* newNode |
| 5 | Prosedur: tambahBahanBakuByParams | digunakan untuk menambahkan atau memperbarui bahan baku ke dalam hashtable. | - hashTable - nama - harga - stok | Tidak ada (void). Fungsi ini tidak mengembalikan nilai. | - newBahan |

| | | | | | |
|----|--------------------------------------|--|--|---|--|
| 6 | Prosedur: addBahanBaku | digunakan untuk menambahkan bahan baku ke dalam hashtable | - hashTable | Tidak ada (void). Fungsi ini tidak mengembalikan nilai. | - nama string - harga int - stok int |
| 7 | Prosedur: inisialisasiBahanBaku | untuk inisialisasi dan menambahkan bahan baku awal ke dalam hash table | - hashTable | - | - |
| 8 | Prosedur: tampilkanSemuaBahanBaku | untuk menampilkan semua bahan baku yang ada dalam hash table. | - table | Tidak ada (void). Fungsi ini tidak mengembalikan nilai. | - |
| 9 | Prosedur: findBahanBaku | untuk mencari bahan baku dalam hash table berdasarkan nama. | - nama - HashTable | return pointer ke bahanBaku | - index |
| 10 | Prosedur: cariBahanBaku_Cari | untuk melakukan pencarian bahan baku dalam hash table berdasarkan nama dengan input dari user. | - | - | - |
| 11 | Prosedur: updateHargaBahanBaku | untuk mengupdate harga bahan baku dalam hash table berdasarkan nama | - hashTable[] - nama - newHarga | - | - *bahan |
| 12 | Prosedur: | untuk mengupdate harga | - hashTable[] | - | - *bahan |

| | | | | | |
|----|----------------------------|---|--|---|---|
| | updateStokBahanBaku | bahan baku dalam hash table berdasarkan nama | - nama - newStok | | |
| 13 | Prosedur: cekStokBahanBaku | digunakan untuk mengecek stok bahan baku dalam hash table dan memberikan notifikasi jika stoknya kurang dari 10 | - hashTable[] | - | - bool : adaBahanBaku |
| 14 | Prosedur: addMenu | digunakan untuk menambahkan item menu ke dalam struktur tree menu. | - node - kategori - subKategori - item | - | - |
| 15 | Prosedur: deleteItem | digunakan untuk menghapus item menu dari struktur pohon menu. | - node - kategori - subKategori - namaMenu | - | - |
| 16 | Prosedur: addCategory | digunakan untuk menambahkan kategori baru beserta subkategori, item, harga, dan stok ke dalam struktur tree menu. | -node: Pointer menuNode yang akan menjadi parent dari kategori dan subkategori baru yang akan ditambahkan. | - | kategori: String subKategori: String namaItem: String newPrice: Integer newStock: Integer |
| 17 | Prosedur: addSubcategory | digunakan untuk menambahkan subkategori dan item ke dalam kategori yang | node: Pointer menuNode yang akan dicari kategori yang | - | kategori: String subKategori: String namaItem: |

| | | | | | |
|----|--|--|---|---|--|
| | | sudah ada dalam struktur pohon menu. | sudah ada dan di dalamnya ditambahkan subkategori dan item baru. | | String newPrice: Integer newStock: Integer |
| 18 | Prosedur: tambahKategoriSubkategoriEvent | bertujuan untuk menambahkan kategori atau subkategori ke dalam event pada menu kafe. | events : vector struktur data event node : pointer menuNode | - | namaKategori: String kategoriExist: Boolean selectedEvent: Integer |
| 19 | Prosedur: deleteCategory | bertujuan untuk menghapus kategori beserta semua subkategori dan item yang terkait dari struktur menu kafe | node: Pointer menuNode category: String yang menyimpan nama kategori | - | it : iterasi digunakan utk traversing |
| 20 | Prosedur: deleteSubCategory | berfungsi untuk menghapus subkategori beserta semua item yang terkait dari suatu kategori di dalam struktur menu kafe. | node: Pointer menuNode category: String subCategory: String | - | it : iterasi digunakan utk traversing |
| 21 | Prosedur: hapusKategoriEvent | digunakan untuk menghapus kategori dari suatu event dalam struktur data events. | -events: Vector yang menyimpan struktur data event. -namaEvent: String yang menyimpan nama | -indeksEvent: Integer -kategoriEvent: Reference ke vector string -kategoriDitem | |

| | | | | | |
|----|----------------------------------|--|---|---|--|
| | | | event yang ingin dihapus kategori. -kategori: String yang menyimpan nama kategori yang ingin dihapus dari event. | | ukan: Boolean -indeksKategori: Integer |
| 22 | Fungsi: cariItemMenu | digunakan untuk mencari item menu berdasarkan nama pada struktur data menu yang terorganisir dalam bentuk pohon. | node: Pointer ke menuNode namaMenu: String kategori: String subKategori: String | Jika item tidak ditemukan, mengembalikan nullptr. | item: Variabel iterasi yang menyimpan informasi item menu pada suatu node. |
| 23 | Prosedur: updateHargaItemMenu | untuk memperbarui harga item menu berdasarkan nama pada struktur data menu yang terorganisir dalam bentuk pohon. | node: Pointer ke menuNode namaMenu: String hargaBaru: Integer | - | item: Variabel iterasi yang menyimpan informasi item menu pada node. |
| 24 | Prosedur: updateStokItemMenu | untuk memperbarui stok item menu berdasarkan nama pada struktur data menu yang terorganisir dalam bentuk pohon. | node: Pointer ke menuNode namaMenu: String stokBaru: Integer | - | item: Variabel iterasi yang menyimpan informasi item menu pada node. |
| 25 | Prosedur: lihatMenu | digunakan untuk menampilkan daftar menu beserta informasi harga dan stoknya. | node: Pointer ke menuNode, merupakan node saat ini dalam pencarian. | - | item: Variabel iterasi yang menyimpan informasi item menu pada |

| | | | | | |
|----|-----------------------------------|--|---|---|--|
| | | | | | node. |
| 26 | Prosedur: lihatMenuBy Event | untuk menampilkan daftar menu berdasarkan kategori atau subkategori yang terkait dengan suatu event. | events: Referensi vektor Event node: Pointer ke menuNode | - | -event: Variabel iterasi -categoryName : Variabel iterasi -child: Variabel iterasi |

BAB III

TEST CASE DAN TEST SCENARIO

3.1 Semua menu

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---|---|--|---------------------------------|-------------|
| 1 | Inputan yang dimasukan dalam program sesuai dan tipe data INT positif | 1. Kelola Menu dan Harga 2. Lihat menu 3. Semua menu 4. Menampilkan semua menu | - inputan pertama = 1 - inputan kedua = 1 - inputan ketiga = 1 | Menampilkan semua menu yang ada | PASS |
| 2 | Inputan dalam bentuk INT | | - inputan pertama = 1 | “Pilihan tidak tersedia, coba | PASS |

| | | | | | |
|---|-----------------------------|--|----------------------|--------------------------------------|-------------------|
| | negatif | | - inputan kedua = -1 | lagi!” | |
| 3 | Inputan dalam bentuk string | | “View Menu” | “Pilihan tidak tersedia, coba lagi!” | FAIL (Looping) |

3.2 Menu event

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---|---|--|---|-------------------|
| 1 | Inputan dalam pilihan menu sesuai fitur (INT) | | - inputan pertama = 1 - inputan kedua = 1 - inputan ketiga = 2 | Menampilkan semua menu event | PASS |
| 2 | Jika event (e.g Breakfast) kosong | 1. Kelola Menu dan Harga 2. Lihat menu 3. Menu event 4. Menampilkan menu event | - inputan pertama = 1 - inputan kedua = 1 - inputan ketiga = 2 | Menampilkan semua menu event “ Event: Breakfast Event: Lunch Makanan Utama - Sup ” | PASS |
| 3 | Inputan dalam bentuk string | | “Menu Event” | “Pilihan tidak tersedia, coba lagi!” | FAIL (Looping) |

3.3 Cari menu

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|-------------------------------------|--------------------------|--|---|-------------|
| 1 | Item ada di dalam array | | - inputan pertama = 1 - inputan kedua = 1 - inputan ketiga = 3 - nama item (Burger) | “Burger Kategori: Roti dan Kue - Bun Harga: ... Stok: ...” | PASS |
| 2 | Item yang dicari tidak ada di dalam | 1. Kelola Menu dan Harga | - inputan pertama = 1 - inputan | “Item menu Bubur tidak ditemukan” | PASS |

| | | | | | |
|---|--|--|--|------------------------------------|------|
| | array | 2. Lihat menu 3. Cari menu 4. Masukan nama item yang dicari: | kedua = 1 - inputan ketiga = 3 - nama item (Bubur) | | |
| 3 | Huruf awal item yg diinput tidak kapital | | - inputan pertama = 1 - inputan kedua = 1 - inputan ketiga = 3 - nama item (burger) | “Item menu burger tidak ditemukan” | PASS |
| 4 | Jika array item kosong (Kategori dan sub kategori masih ada) | | - inputan pertama = 1 - inputan kedua = 1 - inputan ketiga = 3 - nama item (Burger) | “Item menu Burger tidak ditemukan” | PASS |

3.4 Tambah kategori

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--------------------|--|--|---|-------------|
| 1 | Kategori belum ada | 1. Kelola Menu dan Harga | Kategori = “Dessert” Subkategori = “Puding” Item = “Puding Coklat” Harga = 8000 Stok = 10 | Kategori berhasil ditambahkan | PASS |
| 2 | Kategori sudah ada | 2. Tambah Menu 3. Tambah kategori 4. Masukan kategori baru 5. Masukan subkategori baru 6. Masukan nama item baru 7. Masukan harga item baru | Kategori = “Makanan Utama” Subkategori = “Nasi” Item = “Nasi uduk” Harga = 10000 Stok = 10 | Kategori “Makanan Utama” tidak terduplikat. Data baru yang dimasukan akan masuk ke kategori yang sudah ada. | PASS |

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|----------------|---------------------------|-------|--|-------------|
| 3 | Inputan kosong | 8. Masukan stok item baru | “ ” | Tidak berpengaruh apa-apa ke dalam program | PASS |

3.5 Tambah sub kategori

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--|--|---|---|-------------|
| 1 | Kategori sudah ada, dan sub kategori belum ada | 1. Kelola Menu dan Harga 2. Tambah Menu 3. Tambah sub kategori 4. Masukan kategori yang sudah ada 5. Masukan sub kategori baru 6. Masukan nama item baru 7. Masukan harga item baru 8. Masukan stok item baru | Kategori = "Roti dan kue" Sub kategori = "Puding" Item "Puding Coklat" Harga = 10000 Stok = 7 | Berhasil menambahkan sub kategori baru serta item, harga dan stok | PASS |
| 2 | Kategori sudah ada dan sub kategori sudah ada | | Kategori = "Roti dan kue" Sub kategori = "Pastry" Item "Puding Coklat" Harga = 10000 Stok = 7 | Menampilkan "Subkategori yang dimasukan sudah ada" dan tidak berpengaruh terhadap program | FAIL |
| 3 | Kategori belum ada | | Kategori = "Dessert" Sub kategori = "Puding" Item "Puding Coklat" Harga = 10000 Stok = 7 | Menampilkan "Kategori yang dimasukan belum ada" dan tidak berpengaruh terhadap program | PASS |

3.6 Tambah item menu

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--|---|--|---|-------------|
| 1 | Item belum ada (Kategori dan Sub kategori sudah ada) | | Item = “Mango Juice” Harga = 10000 Stok = 6 Kategori = “Minuman” Sub kategori = “Juice” | Menampilkan “Item berhasil ditambahkan” | PASS |
| 2 | Item sudah ada (Kategori dan Sub kategori yang sesuai) | 1. Kelola Menu dan Harga 2. Tambah Menu 3. Tambah item menu 4. Masukan nama item menu 5. Masukan harga 6. Masukan stok 7. Masukan kategori 8. Masukan sub kategori | Item = “Burger” Harga = 10000 Stok = 6 Kategori = “Roti dan kue” Sub kategori = “Bun” | Menampilkan “Item yang dimasukkan sudah ada” dan tidak berpengaruh terhadap program | FAIL |
| 3 | Item belum ada (Kategori, sub kategori tidak sesuai) | | Item = “Es campur” Harga = 10000 Stok = 6 Kategori = “Minuman Manis” Sub kategori = “Es” | Item baru tidak ditambahkan ke program | PASS |
| 4 | Inputan kosong | | “ ” | Error | PASS |

3.7 Tambah kategori/subkategori ke menu event

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---|--|--|---|-------------|
| 1 | Jika kategori/sub kategori tidak ada dalam event tersebut | 1. Kelola Menu dan Harga 2. Tambah Menu 3. Tambah kategori ke menu event 4. Masukan nama kategori/sub kategori yang ingin ditambahkan 5. Pilih event yang ingin ditambahkan kategori (1-5): 1. Breakfast 2. Brunch 3. Lunch 4. Morning Coffe/Tea time/Afternoon Coffe 5. Dinner | Kategori = “Roti dan kue” Event = “5” | “Kategori/sub kategori berhasil ditambahkan ke dalam event Dinner!” | PASS |
| 2 | Jika kategori/sub kategori tidak ada dalam menu | | Kategori = “Dessert” | “Kategori tidak ditemukan dalam menu kafe” dan tidak berpengaruh terhadap program | PASS |
| 3 | Jika inputan event diluar 1-5 | | Kategori = “Roti dan kue” Event = “6” | Looping “Masukkan nomor event (1-5)” sampai benar | PASS |
| 4 | Jika kategori/sub kategori ada dalam event tersebut | | Kategori = “Roti dan kue” Event = “1” | “Kategori/sub kategori sudah ada dalam event” | FAIL |

3.8 Hapus kategori

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--|---|----------------------|--------------------------------------|-------------|
| 1 | Kategori yang akan dihapus ada | 1. Kelola Menu dan Harga 2. Hapus menu 3. Hapus kategori 4. Masukan nama kategori yang ingin dihapus | Kategori = “Minuman” | Kategori tersebut dihapus dalam menu | PASS |
| 2 | Kategori yang akan dihapus tidak ada | | Kategori = “Cemilan” | Tidak ada Perubahan dalam menu | PASS |
| 3 | Nama awalan kategori selalu huruf besar diikuti dengan huruf kecil selanjutnya | | “MINUMA N” | Tidak ada Perubahan dalam menu | PASS |

3.9 Hapus sub kategori

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--|---|--|---|-------------|
| 1 | Kategori dan sub kategori sudah ada dalam menu | 1. Kelola Menu dan Harga 2. Hapus menu 3. Hapus sub kategori 4. Masukan kategori 5. Masukan Sub kategori yang ingin dihapus | Kategori = “Roti dan kue” Subkategori = “Viennoiseries” | Menampilkan “Sub kategori “Viennoiseries” telah di hapus” | PASS |
| 2 | Kategori sudah sesuai, tetapi Sub kategori yang dicari tidak ada atau tidak sesuai | | Kategori = “Minuman” Subkategori = “Jamu” | Menampilkan “Sub kategori tidak ditemukan” | PASS |
| 3 | Nama awalan sub kategori selalu huruf besar diikuti dengan huruf kecil selanjutnya | | Kategori = “Roti dan kue” “VIENNOISERIES” | Menampilkan “Sub kategori tidak ditemukan” | PASS |
| 4 | Kategori dan subkategori tidak ada dalam program | | Kategori = “Dessert” Subkategori = “Puding” | Menampilkan “Sub kategori tidak ditemukan” | PASS |

3.10 Hapus Item

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--|--|---|--|-------------|
| 1 | Item ada di dalam menu | 1. Kelola Menu dan Harga 2. Hapus menu 3. Hapus item menu 4. Masukan kategori 5. Masukan subkategori 6. Masukan nama item | Kategori = “Roti dan Kue” Subkategori = “Bun” Item = “Burger” | Menampilkan “Item Burger berhasil dihapus” | PASS |
| 2 | Item yang akan dihapus tidak ada | 1. Kelola Menu dan Harga 2. Hapus menu 3. Hapus item menu 4. Masukan kategori 5. Masukan subkategori 6. Masukan nama item | Kategori = “Minuman” Subkategori = “Juice” Item = “Mango Juice” | Menampilkan “Item tidak ditemukan” | FAIL |
| 3 | Nama awalan kategori selalu huruf besar diikuti dengan huruf kecil selanjutnya | 1. Kelola Menu dan Harga 2. Hapus menu 3. Hapus item menu 4. Masukan kategori 5. Masukan subkategori 6. Masukan nama item | “Roti dan Kue” “Bun” “BURGER” | Menampilkan “Item tidak ditemukan” | FAIL |

3.11 Update stok menu

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---------------------------------|--|--|--|----------------|
| 1 | inputan stok berupa int positif | 1. Kelola Menu dan Harga | Item = “Carbonara” Jumlah stok baru = 20 | Menampilkan “Stok item Carbonara berhasil diperbarui menjadi 20” | PASS |
| 2 | inputan stok berupa float | 1. Kelola Menu dan Harga 2. Update menu 3. Update stok menu 4. Masukan nama item 5. Masukan jumlah stok baru | Item = “Carbonara” Jumlah stok baru = 7,5 | Inputan yang dikenali program hanya angka yang di depan saja, yaitu 7, sehingga akan menampilkan “Stok item menu Carbonara | PASS (looping) |

| | | | | |
|----|--|---|--|------|
| | | | berhasil diperbarui menjadi 7 | |
| 3 | inputan stok berupa string | Item = “Carbonara” Jumlah stok baru = “tujuh” | Eror | PASS |
| 4 | Item yang di update tidak ada di dalam program | Item = “Puding” Jumlah stok baru = 7 | Menampilkan “Item tidak ada dalam program” | FAIL |
| 5. | inputan berupa int negatif | Item = “Puding” Jumlah stok baru = -7 | Menampilkan “Inputan tidak sesuai” | FAIL |

3.12 Update harga item

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---------------------------------------|--|---|---|-------------|
| 1 | Inputan harga menggunakan tipe INT | 1. Kelola Menu dan Harga 2. Update menu 3. Update stok menu 4. Masukan nama item 5. Masukan harga baru | Item = “Carbonara” Jumlah stok baru = 25000 | Menampilkan “Harga item menu Carbonara berhasil diperbarui menjadi 25000” | PASS |
| 2 | Item yang dicari tidak ada | | Item = “Pesto” Jumlah stok baru = 20000 | Menampilkan “Item tidak ada” | FAIL |
| 3 | Inputan harga berupa tipe data string | | Item = “Carbonara” Jumlah stok baru = “dua puluh ribu” | Eror | PASS |
| 4 | Inputan harga berupa int negatif | | Item = “Carbonara” Jumlah stok baru = -20000 | Menampilkan “Inputan tidak sesuai” | FAIL |

3.13 Hapus subkategori/kategori dalam event

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---|--|--|---|-------------|
| 1 | event dan subkategori yang akan dihapus ada di dalam program | 1. Kelola Menu dan Harga 2. Hapus menu 3. Hapus item event 4. Masukan event 5. Masukan nama subkategori yang ingin dihapus | Event = “Breakfast” Subkategori = “Bread” | Menampilkan “Subkategori ‘Bread’ berhasil dihapus dari event ‘Breakfast’” | PASS |
| 2 | event yang akan dihapus tidak ada di program | | Event = “Christmas” | Menampilkan “Event tidak ditemukan” | PASS |
| 3 | Nama awalan subkategori selalu huruf besar diikuti dengan huruf kecil selanjutnya | | Event = “CHRISTMAS” | Menampilkan “Event tidak ditemukan” | PASS |
| 4 | Event ada dalam program, tetapi subkategori yang akan dihapus tidak ada | | Event = “Brunch” Subkategori = “Pastry” | Menampilkan “Subkategori ‘Pastry’ tidak ditemukan pada event ‘Brunch’ ” | PASS |

3.14 Lihat semua bahan baku

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---------------|--------|-------|-----------------|-------------|
| | | | | | |

| | | | | | |
|---|---|---|--|---------------------------------------|------|
| 1 | Inputan yang dimasukan dalam program sesuai dan tipe data INT positif | 1. Kelola bahan baku 2. Lihat semua bahan baku 3. Menampilkan daftar bahan baku | - inputan pertama = 2 - inputan kedua = 1 | Menampilkan semua bahan baku yang ada | PASS |
| 2 | Inputan dalam bentuk INT negatif | | - inputan pertama = -2 - inputan kedua = -1 | “Pilihan tidak tersedia, coba lagi!” | PASS |
| 3 | Inputan dalam bentuk string | | “View bahan baku” | “Pilihan tidak tersedia, coba lagi!” | FAIL |

3.15 add bahan baku

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|----------------------|---|---|---|-------------|
| 1 | Bahan baku belum ada | 1. Kelola bahan baku 2. Tambah bahan baku 3. Masukan nama bahan baku 4. Masukan harga bahan baku 5. Masukan stok bahan baku | nama bahan baku = “Mayonaise” harga bahan baku = 6000 stok bahan baku = 5 | Menampilkan “Bahan baku berhasil ditambahkan” | PASS |
| 2 | Bahan baku sudah ada | | nama bahan baku = “Bacon” harga bahan baku = 10000 stok bahan baku = 20 | Menampilkan “Bahan baku berhasil ditambahkan” dan akan menimpa ke bahan baku yang sudah | PASS |
| 3 | Inputan kosong | | “ ” | Error | FAIL |

3.16 Update stok bahan baku

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---|---|---|--|----------------|
| 1 | nama bahan baku terdapat dalam program dan tipe data stok INT | 1. Kelola bahan baku 2. Update stok 3. Masukan nama bahan baku 5. Masukan jumlah stok baru | nama bahan baku = “Bacon” jumlah stok baru = 1000 | Menampilkan “Jumlah stok bahan baku ‘Bacon’ berhasil diupdate menjadi 1000” | PASS |
| 2 | nama bahan baku tidak terdapat dalam program | | nama bahan baku = “Paper” jumlah stok baru = 1000 | Menampilkan “Bahan baku dengan nama ‘Paper’ tidak ditemukan” | PASS |
| 3 | nama bahan baku terdapat dalam program, tetapi inputan stok berupa float | | Bahan baku = “Bacon” Jumlah stok baru = 7,5 | Inputan yang dikenali program hanya angka yang di depan saja, yaitu 7, sehingga akan menampilkan “Stok bahan baku ‘Bacon’ berhasil diupdate menjadi 7” | PASS (looping) |
| 4 | nama bahan baku terdapat dalam program, tetapi inputan stok berupa string | | Bahan baku = “Bacon” Jumlah stok baru = “tujuh” | Eror | FAIL |

| | | | | | |
|---|---|--|--|------------------------------------|------|
| 5 | nama bahan baku terdapat dalam program, tetapi inputan berupa int negatif | | Bahan baku = “Bacon” Jumlah stok baru = -7 | Menampilkan “Inputan tidak sesuai” | FAIL |
|---|---|--|--|------------------------------------|------|

3.17 Update harga bahan baku

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|---|--|---|--|----------------|
| 1 | nama bahan baku terdapat dalam program dan tipe data harga INT | 1. Kelola bahan baku 2. Update harga 3. Masukan nama bahan baku 5. Masukan harga baru | nama bahan baku = “Bacon” harga = 30000 | Menampilkan “Harga bahan baku 'Bacon' berhasil diupdate menjadi Rp30000” | PASS |
| 2 | nama bahan baku tidak terdapat dalam program | | nama bahan baku = “Paper” jumlah stok baru = 10000 | Menampilkan “Bahan baku dengan nama ‘Paper’ tidak ditemukan” | PASS |
| 3 | nama bahan baku terdapat dalam program, tetapi inputan harga berupa float | | Item = “Bacon” Harga baru = 25,000 | Inputan yang dikenali program hanya angka yang di depan saja, yaitu 7, sehingga akan menampilkan “Harga bahan baku ‘Bacon’ berhasil diupdate menjadi Rp25” | PASS (looping) |

| | | | | | |
|---|--|--|---|------|------|
| 4 | nama bahan baku terdapat dalam program, tetapi inputan harga berupa string | | tem = "Carbonara" Harga baru = "tujuh" | Eror | FAIL |
|---|--|--|---|------|------|

3.18 Cari bahan baku

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--|--|--|--|-------------|
| 1 | Bahan baku yang ingin dicari ada di dalam program | | - inputan pertama = 2 - inputan kedua = 5 - inputan - nama item "Bacon" | Menampilkan "Bahan baku ditemukan: Bacon - Harga: Rp10000 - Stok: 500" | PASS |
| 2 | Item yang dicari tidak ada di dalam array | 1. Kelola Bahan baku 2. Cari bahan baku 3. Masukan nama bahan baku yang ingin dicari | - inputan pertama = 2 - inputan kedua = 5 - inputan - nama item "Paper" | Menampilkan "Bahan baku dengan nama 'Paper' tidak ditemukan." | PASS |
| 3 | Huruf awal item yg diinput tidak kapital | | - inputan pertama = 2 - inputan kedua = 5 - inputan - nama item "bacon" | Menampilkan "Bahan baku dengan nama 'bacon' tidak ditemukan." | PASS |
| 4 | Jika array item kosong (Kategori dan sub kategori masih ada) | | - inputan pertama = 2 - inputan kedua = 5 - nama item = "Bacon" | Menampilkan "Bahan baku dengan nama 'Bacon' tidak ditemukan." | PASS |

3.19 Notifikasi

| No. | Pre-condition | Proses | Input | Expected result | Real Result |
|-----|--------------------------------|--|----------------------|--|-------------|
| 1 | Inputan pada tampilan awal = 0 | <ol style="list-style-type: none"> 1. Menampilkan tampilan awal program 2. Memasukan inputan = 0 3. Menampilkan notifikasi dan akan menutup program | - inputan = 0 | <p>Menampilkan “Bahan baku ‘...’ stoknya kurang dari 10. Stok sekarang: 5</p> <p>Terima kasih, sampai jumpa!”</p> | PASS |
| 2. | Inputan tampilan awal bukan 0 | | - inputan = 1, 1/2/3 | <p>- inputan = 1, maka akan masuk ke tampilan ‘Kelola menu dan harga’</p> <p>- inputan = 2, maka akan masuk ke tampilan ‘Kelola bahan baku’</p> <p>- inputan = 3, menampilkan ‘Pilihan tidak tersedia, coba lagi!’</p> | PASS |
| 3 | Inputan merupakan INT negatif | | inputan = -1 | Menampilkan “Pilihan tidak tersedia, coba lagi!” | PASS |

BAB IV

MANUAL DOKUMEN

4.1 Kelola menu dan harga

Kelola menu dan harga terdapat 5 fitur yang berisi,

A. Lihat menu

Fitur ini menampilkan semua daftar menu, event, dan mencari menu

a. Semua menu

Fitur ini menampilkan semua kategori, subkategori dan juga Item

Langkah-langkah :

1. Saat program dijalankan, pada tampilan awal “welcome” masukan pilihan = 1 untuk dalam tampilan “ Kelola menu dan harga”
2. Sistem akan menampilkan beberapa pilihan pada tampilan “Kelola menu dan harga”, kemudian masukkan pilihan = 1 untuk dalam tampilan “Lihat menu”.
3. Setelah itu sistem akan menampilkan dalam tampilan “Lihat menu”. Kemudian masukkan pilihan = 1 untuk dalam tampilan “Semua menu”, program akan menampilkan seluruh kategori, subkategori, dan item menu yang ada

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

----Kelola Menu dan Harga----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 1

----Lihat Menu----
1. Semua menu
2. Menu event
3. Cari menu
9. Kembali
Masukkan pilihan: 1
```

Output:

| ~Semua menu~ | |
|---|--|
| Daftar Menu | Daftar Menu |
| - : | Minuman - Teh: |
| Daftar Menu | - Earl Grey (Rp5000) - Stok : 25 |
| Roti dan kue - Viennoiseries: | - Chamomile (Rp4000) - Stok : 20 |
| - Croissant (Rp10000) - Stok : 20 | - English Breakfast (Rp6000) - Stok : 22 |
| - Pain au Chocolate (Rp12000) - Stok : 25 | - Darjeeling (Rp5500) - Stok : 18 |
| - Brioche (Rp15000) - Stok : 25 | - Oolong (Rp7000) - Stok : 15 |
| Daftar Menu | - Matcha (Rp8000) - Stok : 12 |
| Roti dan kue - Pastry: | - Jasmine (Rp6500) - Stok : 20 |
| - Eclair (Rp18000) - Stok : 10 | Daftar Menu |
| - Mille-feuille (Rp20000) - Stok : 12 | Minuman - Kopi: |
| - Tarte Tatin (Rp25000) - Stok : 8 | - Long black (Rp9000) - Stok : 30 |
| - Tarlet (Rp15000) - Stok : 20 | - Cappuccino (Rp10000) - Stok : 25 |
| Daftar Menu | - Latte (Rp11000) - Stok : 22 |
| Roti dan kue - Bread: | - Espresso (Rp8000) - Stok : 28 |
| - Baguette (Rp8000) - Stok : 30 | - Machiato (Rp9500) - Stok : 20 |
| - Sourdough (Rp10000) - Stok : 25 | - Piccolo (Rp10500) - Stok : 18 |
| - Brioche Long (Rp15000) - Stok : 20 | Daftar Menu |
| Daftar Menu | Makanan Utama - Salad: |
| Roti dan kue - Bun: | - Caesar Salad (Rp15000) - Stok : 20 |
| - Bagel (Rp7000) - Stok : 15 | - Caprese Salad (Rp14000) - Stok : 18 |
| - Burger (Rp12000) - Stok : 20 | - Waldorf Salad (Rp16000) - Stok : 15 |
| - Hotdog (Rp9000) - Stok : 18 | - Nicoise Salad (Rp17000) - Stok : 12 |
| Daftar Menu | Daftar Menu |
| Minuman - Juice: | Makanan Utama - Sup: |
| - Apple Juice (Rp8000) - Stok : 30 | - Pumpkin Soup (Rp12000) - Stok : 25 |
| - Orange Juice (Rp7000) - Stok : 25 | - Mushroom Soup (Rp13000) - Stok : 22 |
| - Strawberry Juice (Rp9000) - Stok : 20 | - Corn Soup (Rp11000) - Stok : 28 |

| Daftar Menu |
|---------------------------------------|
| Makanan Utama - Sandwich: |
| - Club Sandwich (Rp18000) - Stok : 15 |
| - Tuna Sandwich (Rp16000) - Stok : 20 |
| Daftar Menu |
| Makanan Utama - Pasta: |
| - Gnocchi (Rp20000) - Stok : 12 |
| - Aglio Olio (Rp15000) - Stok : 18 |
| - Lasagna (Rp22000) - Stok : 15 |
| - Fettucine (Rp16000) - Stok : 20 |
| - Carbonara (Rp18000) - Stok : 18 |

Press any key to continue . . . □

b. Menu event

Fitur ini menampilkan semua kategori, subkategori dan juga Item

Langkah-langkah :

1. Saat program dijalankan, pada tampilan awal “welcome” masukan pilihan = 1 untuk dalam tampilan “ Kelola menu dan harga”.
2. Sistem akan menampilkan beberapa pilihan pada tampilan “Kelola menu dan harga”, kemudian masukkan pilihan = 1 untuk dalam tampilan “Lihat menu”.
3. Setelah itu sistem akan menampilkan dalam tampilan “Lihat menu”. Kemudian masukkan pilihan = 2 untuk dalam tampilan “Menu event”, program akan menampilkan seluruh daftar menu berdasarkan waktu atau event

Contoh *input* dan *output* pada program Menu event:

```

=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

----Kelola Menu dan Harga----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 1

----Lihat Menu----
1. Semua menu
2. Menu event
3. Cari menu
9. Kembali
Masukkan pilihan: 2

```

Gambar 4.1.2 tampilan input output pada Menu event

| | |
|---|--|
| <pre> ~Menu Event~ ----- Daftar Menu Berdasarkan Waktu atau Event ----- Event: Breakfast Kategori: Makanan Utama - Sandwich - Club Sandwich (Rp18000) - Stok: 15 - Tuna Sandwich (Rp16000) - Stok: 20 Kategori: Roti dan kue - Bread - Baguette (Rp8000) - Stok: 30 - Sourdough (Rp10000) - Stok: 25 - Brioche Long (Rp15000) - Stok: 20 Kategori: Roti dan kue - Bun - Bagel (Rp7000) - Stok: 15 - Burger (Rp12000) - Stok: 20 - Hotdog (Rp9000) - Stok: 18 Kategori: Minuman - Juice - Apple Juice (Rp8000) - Stok: 30 - Orange Juice (Rp7000) - Stok: 25 - Strawberry Juice (Rp9000) - Stok: 20 Kategori: Minuman - Teh - Earl Grey (Rp5000) - Stok: 25 - Chamomile (Rp4000) - Stok: 20 - English Breakfast (Rp6000) - Stok: 22 - Darjeeling (Rp5500) - Stok: 18 - Oolong (Rp7000) - Stok: 15 - Matcha (Rp8000) - Stok: 12 - Jasmine (Rp6500) - Stok: 20 Kategori: Minuman - Kopi - Long black (Rp9000) - Stok: 30 - Cappuccino (Rp10000) - Stok: 25 - Latte (Rp11000) - Stok: 22 - Espresso (Rp8000) - Stok: 28 - Machiato (Rp9500) - Stok: 20 - Piccolo (Rp10500) - Stok: 18 </pre> | <pre> Event: Brunch Kategori: Makanan Utama - Sandwich - Club Sandwich (Rp18000) - Stok: 15 - Tuna Sandwich (Rp16000) - Stok: 20 Kategori: Roti dan kue - Bread - Baguette (Rp8000) - Stok: 30 - Sourdough (Rp10000) - Stok: 25 - Brioche Long (Rp15000) - Stok: 20 Kategori: Roti dan kue - Bun - Bagel (Rp7000) - Stok: 15 - Burger (Rp12000) - Stok: 20 - Hotdog (Rp9000) - Stok: 18 Kategori: Makanan Utama - Salad - Caesar Salad (Rp15000) - Stok: 20 - Caprese Salad (Rp14000) - Stok: 18 - Waldorf Salad (Rp16000) - Stok: 15 - Nicoise Salad (Rp17000) - Stok: 12 Kategori: Makanan Utama - Sup - Pumpkin Soup (Rp12000) - Stok: 25 - Mushroom Soup (Rp13000) - Stok: 22 - Corn Soup (Rp11000) - Stok: 28 Kategori: Minuman - Juice - Apple Juice (Rp8000) - Stok: 30 - Orange Juice (Rp7000) - Stok: 25 - Strawberry Juice (Rp9000) - Stok: 20 Kategori: Minuman - Teh - Earl Grey (Rp5000) - Stok: 25 - Chamomile (Rp4000) - Stok: 20 - English Breakfast (Rp6000) - Stok: 22 - Darjeeling (Rp5500) - Stok: 18 - Oolong (Rp7000) - Stok: 15 - Matcha (Rp8000) - Stok: 12 - Jasmine (Rp6500) - Stok: 20 Kategori: Minuman - Kopi - Long black (Rp9000) - Stok: 30 - Cappuccino (Rp10000) - Stok: 25 - Latte (Rp11000) - Stok: 22 - Espresso (Rp8000) - Stok: 28 - Machiato (Rp9500) - Stok: 20 - Piccolo (Rp10500) - Stok: 18 </pre> |
|---|--|

| |
|---|
| Event: Lunch |
| Kategori: Makanan Utama - Salad |
| - Caesar Salad (Rp15000) - Stok: 20 |
| - Caprese Salad (Rp14000) - Stok: 18 |
| - Waldorf Salad (Rp16000) - Stok: 15 |
| - Nicoise Salad (Rp17000) - Stok: 12 |
| Kategori: Makanan Utama - Sup |
| - Pumpkin Soup (Rp12000) - Stok: 25 |
| - Mushroom Soup (Rp13000) - Stok: 22 |
| - Corn Soup (Rp11000) - Stok: 28 |
| Kategori: Makanan Utama - Sandwich |
| - Club Sandwich (Rp18000) - Stok: 15 |
| - Tuna Sandwich (Rp16000) - Stok: 20 |
| Kategori: Makanan Utama - Pasta |
| - Gnocchi (Rp20000) - Stok: 12 |
| - Aglio Olio (Rp15000) - Stok: 18 |
| - Lasagna (Rp22000) - Stok: 15 |
| - Fettucine (Rp16000) - Stok: 20 |
| - Carbonara (Rp18000) - Stok: 18 |
| Kategori: Roti dan kue - Pastry |
| - Eclair (Rp18000) - Stok: 10 |
| - Mille-feuille (Rp20000) - Stok: 12 |
| - Tarte Tatin (Rp25000) - Stok: 8 |
| - Tarlet (Rp15000) - Stok: 20 |
| Kategori: Minuman - Juice |
| - Apple Juice (Rp8000) - Stok: 30 |
| - Orange Juice (Rp7000) - Stok: 25 |
| - Strawberry Juice (Rp9000) - Stok: 20 |
| Kategori: Minuman - Teh |
| - Earl Grey (Rp5000) - Stok: 25 |
| - Chamomile (Rp4000) - Stok: 20 |
| - English Breakfast (Rp6000) - Stok: 22 |
| - Darjeeling (Rp5500) - Stok: 18 |
| - Oolong (Rp7000) - Stok: 15 |
| - Matcha (Rp8000) - Stok: 12 |
| - Jasmine (Rp6500) - Stok: 20 |
| Kategori: Minuman - Kopi |
| - Long black (Rp9000) - Stok: 30 |
| - Cappuccino (Rp10000) - Stok: 25 |
| - Latte (Rp11000) - Stok: 22 |
| - Espresso (Rp8000) - Stok: 28 |
| - Machiato (Rp9500) - Stok: 20 |
| - Piccolo (Rp10500) - Stok: 18 |

| |
|--|
| Event: Morning Coffe/Tea time/Afternoon Coffe |
| Kategori: Minuman - Juice |
| - Apple Juice (Rp8000) - Stok: 30 |
| - Orange Juice (Rp7000) - Stok: 25 |
| - Strawberry Juice (Rp9000) - Stok: 20 |
| Kategori: Minuman - Teh |
| - Earl Grey (Rp5000) - Stok: 25 |
| - Chamomile (Rp4000) - Stok: 20 |
| - English Breakfast (Rp6000) - Stok: 22 |
| - Darjeeling (Rp5500) - Stok: 18 |
| - Oolong (Rp7000) - Stok: 15 |
| - Matcha (Rp8000) - Stok: 12 |
| - Jasmine (Rp6500) - Stok: 20 |
| Kategori: Minuman - Kopi |
| - Long black (Rp9000) - Stok: 30 |
| - Cappuccino (Rp10000) - Stok: 25 |
| - Latte (Rp11000) - Stok: 22 |
| - Espresso (Rp8000) - Stok: 28 |
| - Machiato (Rp9500) - Stok: 20 |
| - Piccolo (Rp10500) - Stok: 18 |
| Kategori: Roti dan kue - Viennoiseries |
| - Croissant (Rp10000) - Stok: 20 |
| - Pain au Chocolate (Rp12000) - Stok: 25 |
| - Brioche (Rp15000) - Stok: 25 |
| Kategori: Roti dan kue - Pastry |
| - Eclair (Rp18000) - Stok: 10 |
| - Mille-feuille (Rp20000) - Stok: 12 |
| - Tarte Tatin (Rp25000) - Stok: 8 |
| - Tarlet (Rp15000) - Stok: 20 |

```

Event: Dinner
Kategori: Makanan Utama - Salad
- Caesar Salad (Rp15000) - Stok: 20
- Caprese Salad (Rp14000) - Stok: 18
- Waldorf Salad (Rp16000) - Stok: 15
- Nicoise Salad (Rp17000) - Stok: 12
Kategori: Makanan Utama - Sup
- Pumpkin Soup (Rp12000) - Stok: 25
- Mushroom Soup (Rp13000) - Stok: 22
- Corn Soup (Rp11000) - Stok: 28
Kategori: Makanan Utama - Sandwich
- Club Sandwich (Rp18000) - Stok: 15
- Tuna Sandwich (Rp16000) - Stok: 20
Kategori: Makanan Utama - Pasta
- Gnocchi (Rp20000) - Stok: 12
- Aglio Olio (Rp15000) - Stok: 18
- Lasagna (Rp22000) - Stok: 15
- Fettucine (Rp16000) - Stok: 20
- Carbonara (Rp18000) - Stok: 18
Kategori: Roti dan kue - Pastry
- Eclair (Rp18000) - Stok: 10
- Mille-feuille (Rp20000) - Stok: 12
- Tarte Tatin (Rp25000) - Stok: 8
- Tarlet (Rp15000) - Stok: 20
Kategori: Minuman - Juice
- Apple Juice (Rp8000) - Stok: 30
- Orange Juice (Rp7000) - Stok: 25
- Strawberry Juice (Rp9000) - Stok: 20
Kategori: Minuman - Teh
- Earl Grey (Rp5000) - Stok: 25
- Chamomile (Rp4000) - Stok: 20
- English Breakfast (Rp6000) - Stok: 22
- Darjeeling (Rp5500) - Stok: 18
- Oolong (Rp7000) - Stok: 15
- Matcha (Rp8000) - Stok: 12
- Jasmine (Rp6500) - Stok: 20
Kategori: Minuman - Kopi
- Long black (Rp9000) - Stok: 30
- Cappuccino (Rp10000) - Stok: 25
- Latte (Rp11000) - Stok: 22
- Espresso (Rp8000) - Stok: 28
- Machiato (Rp9500) - Stok: 20
- Piccolo (Rp10500) - Stok: 18

```

c. Cari menu

Fitur ini menampilkan semua kategori, subkategori dan juga Item

Langkah-langkah :

1. Saat program dijalankan, pada tampilan awal “welcome” masukan pilihan = 1 untuk dalam tampilan “ Kelola menu dan harga”.
2. Sistem akan menampilkan beberapa pilihan pada tampilan “Kelola menu dan harga”, kemudian masukkan pilihan = 1 untuk dalam tampilan “Lihat menu”.
3. Setelah itu sistem akan menampilkan dalam tampilan “Lihat menu”. Kemudian masukkan pilihan = 3 untuk dalam tampilan “Cari menu”, program akan menampilkan pencarian pada sebuah item.

Contoh *input* dan *output* pada program Menu event:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

----Kelola Menu dan Harga----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 1

----Lihat Menu----
1. Semua menu
2. Menu event
3. Cari menu
9. Kembali
Masukkan pilihan: 3
~Cari Menu~

Masukan nama item yang dicari: Cappucino
----- Informasi Item Menu -----
Nama: Cappucino
Kategori: Minuman - Kopi
Harga: Rp10000
Stok: 25
```

B. Tambah menu

- a. Tambah kategori

Fitur ini digunakan untuk menambahkan kategori baru ke dalam program.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Tambah menu.
Masukan pilihan = 2 , lalu masukan pilihan = 1 untuk masuk ke fitur tambah kategori.
3. Masukan kategori baru
4. Masukan subkategori baru
5. Masukan nama item baru
6. Masukan harga item baru
7. Masukan stok item baru
8. Setelah itu, program akan dijalankan kan akan mengecek apakah kategori baru yang ditambahkan sudah ada atau belum. Jika sudah ada maka kategori tersebut tidak akan disimpan oleh program, jika belum makan program akan menyimpan kategori tersebut.

Contoh pengisian:

```
-----WELCOME-----  
1. Kelola menu dan harga  
2. Kelola bahan baku  
0. Tutup  
Masukkan pilihan [1/2/0] : 1  
  
----Kelola Menu dan Harga----  
1. Lihat menu  
2. Tambah menu  
3. Hapus menu  
4. Update menu  
9. Kembali  
Masukkan pilihan: 2  
  
----Tambah Menu----  
1. Tambah kategori  
2. Tambah sub kategori  
3. Tambah item menu  
4. Tambah kategori/subkategori ke menu event  
9. Back  
Masukkan pilihan: 1  
Tambah Kategori  
Tambah kategori, subkategori, item, harga, dan stok  
Masukkan kategori baru: Dessert  
Masukkan subkategori baru: Puding  
Masukkan nama item baru: Puding Coklat  
Masukkan harga item baru: 15000  
Masukkan stok item baru: 10
```

Output:

```
Daftar Menu  
Dessert - Puding:  
- Puding Coklat (Rp15000) - Stok : 10
```

Catatan:

1. Kategori harus bertipe data STRING, dengan awalan huruf kapital dan belum terdapat di program
 2. Sub kategori harus bertipe data STRING, dengan awalan huruf kapital, dan belum ada di program.
 3. Nama item harus bertipe data STRING, dengan awalan huruf kapital
 4. Harga harus bertipe data INT
 5. Stok item harus bertipe data INT
- b. Tambah sub kategori

Fitur ini digunakan untuk menambahkan subkategori baru ke dalam kategori yang sudah ada di dalam program.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Tambah menu.
Masukan pilihan = 2 , lalu masukan pilihan = 2 untuk masuk ke fitur tambah subkategori.
3. Masukan kategori yang sudah ada
4. Masukan subkategori baru
5. Masukan nama item baru
6. Masukan harga item baru
7. Setelah itu, program akan dijalankan kan akan mengecek apakah item baru yang ditambahkan sudah ada atau belum. Jika sudah ada maka item tersebut tidak akan disimpan oleh program, jika belum maka program akan menyimpan item tersebut.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

----Kelola Menu dan Harga----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 2

----Tambah Menu----
1. Tambah kategori
2. Tambah sub kategori
3. Tambah item menu
4. Tambah kategori/subkategori ke menu event
9. Back
Masukkan pilihan: 2
Tambah sub kategori
Tambah subkategori, item, harga, dan stok
Masukkan kategori yang sudah ada: Minuman
Masukkan subkategori baru: Es
Masukkan nama item baru: Es Podeng
Masukkan harga item baru: 7000
Masukkan stok item baru: 10
```

Output:

```
Daftar Menu
Minuman - Es:
- Es Podeng (Rp7000) - Stok : 10
Pilih menu
```

Catatan:

1. Kategori harus bertipe data STRING, dengan awalan huruf kapital dan sudah terdapat di program
 2. Sub kategori harus bertipe data STRING, dengan awalan huruf kapital, dan belum ada dalam program.
 3. Nama item harus bertipe data STRING, dengan awalan huruf kapital.
 4. Harga harus bertipe data INT
 5. Stok item harus bertipe data INT
- c. Tambah item menu

Fitur ini digunakan untuk menambah item baru pada program.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Tambah menu.
Masukan pilihan = 2 , lalu masukan pilihan = 3 untuk masuk ke fitur tambah item menu.
3. Masukan nama item menu
4. Masukan harga
5. Masukan kategori
6. Masukan subkategori
7. Setelah itu, program akan dijalankan dan akan mengecek apakah i baru yang ditambahkan sudah ada atau belum. Jika sudah ada maka subkategori tersebut tidak akan disimpan oleh program, jika belum maka program akan menyimpan subkategori tersebut.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

----Kelola Menu dan Harga----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 2

----Tambah Menu----
1. Tambah kategori
2. Tambah sub kategori
3. Tambah item menu
4. Tambah kategori/subkategori ke menu event
9. Back
Masukkan pilihan: 3
Tambah item
Masukkan nama item menu : Watermelon juice
Masukkan harga : 14000
Masukkan stok : 10
Masukkan ke kategori : Minuman
Masukkan ke subkategori : Juice
```

Output:

```
Daftar Menu
Minuman - Juice:
- Apple Juice (Rp8000) - Stok : 30
- Orange Juice (Rp7000) - Stok : 25
- Strawberry Juice (Rp9000) - Stok : 20
- Watermelon juice (Rp14000) - Stok : 10
```

Catatan:

1. Kategori harus bertipe data STRING, dengan awalan huruf kapital dan sudah terdapat di program
 2. Sub kategori harus bertipe data STRING, dengan awalan huruf kapital, dan sudah terdapat dalam program.
 3. Nama item harus bertipe data STRING, dengan awalan huruf kapital
 4. Harga harus bertipe data INT
 5. Stok item harus bertipe data INT
-
- d. Tambah kategori/subkategori ke menu event

Fitur ini digunakan untuk menambahkan kategori atau subkategori ke dalam event yang sudah tersedia dalam program.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Tambah menu.
Masukan pilihan = 2 , lalu masukan pilihan = 4 untuk masuk ke fitur tambah kategori/subkategori ke menu event.
3. Masukan kategori/subkategori yang ingin ditambahkan
4. Pilih event mana yang ingin ditambahkan kategori/subkategori
5. Setelah itu, program akan dijalankan dan akan mengecek apakah kategori/subkategori baru yang ditambahkan sudah ada atau belum.
Jika sudah ada maka inputan tersebut tidak akan disimpan oleh program,
jika belum maka program akan menyimpan inputan tersebut.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

-----Kelola Menu dan Harga-----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 2

-----Tambah Menu-----
1. Tambah kategori
2. Tambah sub kategori
3. Tambah item menu
4. Tambah kategori/subkategori ke menu event
9. Back
Masukkan pilihan: 4
Tambah kategori/subkategori ke menu event
Masukkan nama kategori/subkategori yang ingin ditambahkan: Salad
Pilih event yang ingin ditambahkan kategori (1-5):
1. Breakfast
2. Brunch
3. Lunch
4. Morning Coffe/Tea time/Afternoon Coffe
5. Dinner
Masukkan nomor event (1-5): 1
Kategori/subkategori berhasil ditambahkan ke dalam event Breakfast!
```

Output:

```

Event: Breakfast
Kategori: Makanan Utama - Sandwich
- Club Sandwich (Rp18000) - Stok: 15
- Tuna Sandwich (Rp16000) - Stok: 20
Kategori: Roti dan kue - Bread
- Baguette (Rp8000) - Stok: 30
- Sourdough (Rp10000) - Stok: 25
- Brioche Long (Rp15000) - Stok: 20
Kategori: Roti dan kue - Bun
- Bagel (Rp7000) - Stok: 15
- Burger (Rp12000) - Stok: 20
- Hotdog (Rp9000) - Stok: 18
Kategori: Minuman - Juice
- Apple Juice (Rp8000) - Stok: 30
- Orange Juice (Rp7000) - Stok: 25
- Strawberry Juice (Rp9000) - Stok: 20
Kategori: Minuman - Teh
- Earl Grey (Rp5000) - Stok: 25
- Chamomile (Rp4000) - Stok: 20
- English Breakfast (Rp6000) - Stok: 22
- Darjeeling (Rp5500) - Stok: 18
- Oolong (Rp7000) - Stok: 15
- Matcha (Rp8000) - Stok: 12
- Jasmine (Rp6500) - Stok: 20
Kategori: Minuman - Kopi
- Long black (Rp9000) - Stok: 30
- Cappuccino (Rp10000) - Stok: 25
- Latte (Rp11000) - Stok: 22
- Espresso (Rp8000) - Stok: 28
- Machiato (Rp9500) - Stok: 20
- Piccolo (Rp10500) - Stok: 18
Kategori: Makanan Utama - Salad
- Caesar Salad (Rp15000) - Stok: 20
- Caprese Salad (Rp14000) - Stok: 18
- Waldorf Salad (Rp16000) - Stok: 15
- Nicoise Salad (Rp17000) - Stok: 12

```

Catatan:

1. Kategori/subkategori harus bertipe data STRING, dengan awalan huruf kapital dan belum masuk ke dalam event yang dituju.

C. Hapus menu

- a. Hapus kategori

Fitur ini digunakan untuk menghapus kategori yang sudah ada dalam program beserta subkategori dan item.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.

2. Sistem akan menampilkan beberapa pilihan, termasuk Hapus menu.
Masukan pilihan = 3 , lalu masukan pilihan = 1 untuk masuk ke fitur Hapus kategori.
3. Masukan nama kategori yang ingin dihapus.
4. Setelah itu, program akan dijalankan dan akan mengecek apakah kategori yang ingin dihapus ada atau tidak. Jika ada maka kategori tersebut akan dihapus oleh program, jika tidak maka program tidak menghapus apa-apa.

Contoh pengisian:

```
=====WELCOME=====  
1. Kelola menu dan harga  
2. Kelola bahan baku  
0. Tutup  
Masukkan pilihan [1/2/0] : 1  
  
-----Kelola Menu dan Harga-----  
1. Lihat menu  
2. Tambah menu  
3. Hapus menu  
4. Update menu  
9. Kembali  
Masukkan pilihan: 3  
  
-----Hapus Menu-----  
1. Hapus kategori  
2. Hapus sub kategori  
3. Hapus item menu  
4. Hapus item event  
9. Back  
Masukkan pilihan: 1  
Hapus Kategori  
Masukkan nama kategori yang ingin dihapus: Minuman
```

Output:

```
~Semua menu~

Daftar Menu
- :
Daftar Menu
Roti dan kue - Viennoiseries:
- Croissant (Rp10000) - Stok : 20
- Pain au Chocolate (Rp12000) - Stok : 25
- Brioche (Rp15000) - Stok : 25
Daftar Menu
Roti dan kue - Pastry:
- Eclair (Rp18000) - Stok : 10
- Mille-feuille (Rp20000) - Stok : 12
- Tarte Tatin (Rp25000) - Stok : 8
- Tarlet (Rp15000) - Stok : 20
Daftar Menu
Roti dan kue - Bread:
- Baguette (Rp8000) - Stok : 30
- Sourdough (Rp10000) - Stok : 25
- Brioche Long (Rp15000) - Stok : 20
Daftar Menu
Roti dan kue - Bun:
- Bagel (Rp7000) - Stok : 15
- Burger (Rp12000) - Stok : 20
- Hotdog (Rp9000) - Stok : 18
Daftar Menu
Makanan Utama - Salad:
- Caesar Salad (Rp15000) - Stok : 20
- Caprese Salad (Rp14000) - Stok : 18
- Waldorf Salad (Rp16000) - Stok : 15
- Nicoise Salad (Rp17000) - Stok : 12
```

```
Daftar Menu
Makanan Utama - Sup:
- Pumpkin Soup (Rp12000) - Stok : 25
- Mushroom Soup (Rp13000) - Stok : 22
- Corn Soup (Rp11000) - Stok : 28
Daftar Menu
Makanan Utama - Sandwich:
- Club Sandwich (Rp18000) - Stok : 15
- Tuna Sandwich (Rp16000) - Stok : 20
Daftar Menu
Makanan Utama - Pasta:
- Gnocchi (Rp20000) - Stok : 12
- Aglio Olio (Rp15000) - Stok : 18
- Lasagna (Rp22000) - Stok : 15
- Fettucine (Rp16000) - Stok : 20
- Carbonara (Rp18000) - Stok : 18
```

Catatan:

1. Kategori harus bertipe data STRING, dengan awalan huruf kapital dan terdapat di program

b. Hapus sub kategori

Fitur ini digunakan untuk menghapus subkategori yang ada didalam program beserta item yang terdapat didalamnya.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Hapus menu.
Masukan pilihan = 3 , lalu masukan pilihan = 2 untuk masuk ke fitur Hapus item menu.
3. Masukan kategori
4. Masukan subkategori
5. Setelah itu, program akan dijalankan dan akan mengecek apakah subaktegori yang ingin dihapus ada atau tidak. Jika ada maka itesubkategori tersebut akan dihapus oleh program, jika tidak maka program tidak menghapus apa-apa.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] :
1

-----Kelola Menu dan Harga-----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 3

-----Hapus Menu-----
1. Hapus kategori
2. Hapus sub kategori
3. Hapus item menu
4. Hapus item event
9. Back
Masukkan pilihan: 2
Hapus sub kategori
Hapus subkategori, item, harga, dan stok

Hapus sub kategori dan semua item di dalamnya
Masukkan kategori: Roti dan kue
Masukkan sub kategori yang ingin dihapus beserta semua isinya: Bun
Sub kategori tidak ditemukan!
Sub kategori tidak ditemukan!
Sub kategori tidak ditemukan!
Sub kategori Bun dan semua item di dalamnya berhasil dihapus!
```

Output:

```
~Semua menu~
Daftar Menu
- :
Daftar Menu
Roti dan kue - Viennoiseries:
- Croissant (Rp10000) - Stok : 20
- Pain au Chocolate (Rp12000) - Stok : 25
- Brioche (Rp15000) - Stok : 25
Daftar Menu
Roti dan kue - Pastry:
- Eclair (Rp18000) - Stok : 10
- Mille-feuille (Rp20000) - Stok : 12
- Tarte Tatin (Rp25000) - Stok : 8
- Tarlet (Rp15000) - Stok : 20
Daftar Menu
Roti dan kue - Bread:
- Baguette (Rp8000) - Stok : 30
- Sourdough (Rp10000) - Stok : 25
- Brioche Long (Rp15000) - Stok : 20
```

Catatan:

1. Kategori harus bertipe data STRING, dengan awalan huruf kapital dan terdapat di program
 2. Sub kategori harus bertipe data STRING, dengan awalan huruf kapital, dan terdapat dalam program.
- c. Hapus item menu
- Fitur ini digunakan untuk menghapus item yang ada didalam program beserta stok dan harganya.
- Langkah-langkah:
1. Ketika pada tampilan awal, pilih 1.
 2. Sistem akan menampilkan beberapa pilihan, termasuk Hapus menu.
Masukan pilihan = 3 , lalu masukan pilihan = 3 untuk masuk ke fitur Hapus kategori.
 3. Masukan kategori
 4. Masukan subkategori
 5. Masukan nama item
 6. Setelah itu, program akan dijalankan dan akan mengecek apakah item yang ingin dihapus ada atau tidak. Jika ada maka item tersebut akan dihapus oleh program, jika tidak maka program tidak menghapus apa-apa.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

-----Kelola Menu dan Harga-----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 3

-----Hapus Menu-----
1. Hapus kategori
2. Hapus sub kategori
3. Hapus item menu
4. Hapus item event
9. Back
Masukkan pilihan: 3
Hapus item
Masukkan kategori: Makanan Utama
Masukkan subkategori: Pasta
Masukkan nama item: Carbonara
Item Carbonara berhasil dihapus
```

Output:

```
Daftar Menu
Makanan Utama - Sandwich:
- Club Sandwich (Rp18000) - Stok : 15
- Tuna Sandwich (Rp16000) - Stok : 20
Daftar Menu
Makanan Utama - Pasta:
- Gnocchi (Rp20000) - Stok : 12
- Aglio Olio (Rp15000) - Stok : 18
- Lasagna (Rp22000) - Stok : 15
- Fettucine (Rp16000) - Stok : 20
```

Catatan:

1. Kategori harus bertipe data STRING, dengan awalan huruf kapital dan terdapat di program
2. Sub kategori/kategori harus bertipe data STRING, dengan awalan huruf kapital, dan terdapat dalam program.
3. Item harus bertipe data STRING, dengan awalan huruf kapital, dan terdapat dalam program.

d. Hapus subkategori/kategori dalam event

Fitur ini digunakan untuk menghapus subkategori yang ada didalam program beserta item yang terdapat didalamnya.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Hapus menu.
Masukan pilihan = 3 , lalu masukan pilihan = 4 untuk masuk ke fitur Hapus kategori.
3. Masukan event
4. Masukan nama subkategori/kategori yang ingin dihapus
5. Setelah itu, program akan dijalankan dan akan mengecek apakah subkategori/kategori yang ingin dihapus ada atau tidak. Jika ada maka inputan tersebut akan dihapus oleh program, jika tidak maka program tidak menghapus apa-apa.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

-----Kelola Menu dan Harga-----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 3

-----Hapus Menu-----
1. Hapus kategori
2. Hapus sub kategori
3. Hapus item menu
4. Hapus item event
9. Back
Masukkan pilihan: 4
Hapus subkategori dalam event
Masukkan event: Breakfast
Masukkan nama item event yang ingin dihapus: Bread
Kategori 'Bread' berhasil dihapus dari event 'Breakfast'.
```

Output:

```
*Semua menu*
Daftar Menu
- :
Daftar Menu
Roti dan kue - Viennoiseries:
- Croissant (Rp10000) - Stok : 20
- Pain au Chocolate (Rp12000) - Stok : 25
- Brioche (Rp15000) - Stok : 25
Daftar Menu
Roti dan kue - Pastry:
- Eclair (Rp18000) - Stok : 10
- Mille-feuille (Rp20000) - Stok : 12
- Tarte Tatin (Rp25000) - Stok : 8
- Tarlet (Rp15000) - Stok : 20
Daftar Menu
Roti dan kue - Bread:
Breadstick (Rp2000) - Stok : 20
```

Catatan:

1. Event harus bertipe data STRING, dengan awalan huruf kapital, dan terdapat dalam program.
2. Subkategori/kategori harus bertipe data STRING, dengan awalan huruf kapital dan terdapat di program

D. Update menu

- a. Update harga menu

Fitur ini digunakan untuk memperbarui harga menu yang ada didalam program.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Update menu.
Masukan pilihan = 4, lalu masukan pilihan = 1 untuk masuk ke fitur Update harga menu.
3. Masukan nama item
4. Masukan harga baru
5. Setelah itu, program akan dijalankan dan akan mengecek apakah item yang ingin diupdate ada atau tidak. Jika ada maka harga item tersebut akan diperbarui oleh program, jika tidak maka program tidak memperbarui apa-apa.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

-----Kelola Menu dan Harga-----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 4

-----Update Menu-----
1. Update harga menu
2. Update stok menu
3. Cari menu
9. Kembali
Masukkan pilihan: 1
Update Harga Menu
Masukkan nama item: Latte
Masukkan harga baru: 20000
Harga item menu Latte berhasil diperbarui menjadi Rp20000.
```

Output:

Awal

```
Daftar Menu
Minuman - Kopi:
- Long black (Rp9000) - Stok : 30
- Cappucino (Rp10000) - Stok : 25
- Latte (Rp11000) - Stok : 22
- Espresso (Rp8000) - Stok : 28
- Machiato (Rp9500) - Stok : 20
- Piccolo (Rp10500) - Stok : 18
```

Akhir

```
Daftar Menu
Minuman - Kopi:
- Long black (Rp9000) - Stok : 30
- Cappucino (Rp10000) - Stok : 25
- Latte (Rp20000) - Stok : 22
- Espresso (Rp8000) - Stok : 28
- Machiato (Rp9500) - Stok : 20
- Piccolo (Rp10500) - Stok : 18
```

Catatan:

1. Item harus bertipe data STRING, dengan awalan huruf kapital, dan terdapat dalam program
2. Harga harus bertipe data INT.

b. Update stok menu

Fitur ini digunakan untuk memperbarui stok menu yang ada didalam program.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Update menu.
Masukan pilihan = 4, lalu masukan pilihan = 2 untuk masuk ke fitur Update stok menu.
3. Masukan nama item
4. Masukan jumlah stok baru
5. Setelah itu, program akan dijalankan dan akan mengecek apakah item yang ingin diupdate ada atau tidak. Jika ada maka stok item tersebut akan diperbarui oleh program, jika tidak maka program tidak memperbarui apa-apa.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

-----Kelola Menu dan Harga-----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 4

-----Update Menu-----
1. Update harga menu
2. Update stok menu
3. Cari menu
9. Kembali
Masukkan pilihan: 2
Update Stok Menu
Masukkan nama item: Latte
Masukkan jumlah stok baru: 30
Stok item menu Latte berhasil diperbarui menjadi 30.
```

Output:

Awal

```
Daftar Menu
Minuman - Kopi:
- Long black (Rp9000) - Stok : 30
- Cappuccino (Rp10000) - Stok : 25
- Latte (Rp11000) - Stok : 22
- Espresso (Rp8000) - Stok : 28
- Machiato (Rp9500) - Stok : 20
- Piccolo (Rp10500) - Stok : 18
Pilih menu:
```

Akhir

```
Daftar Menu
Minuman - Kopi:
- Long black (Rp9000) - Stok : 30
- Cappuccino (Rp10000) - Stok : 25
- Latte (Rp20000) - Stok : 30
- Espresso (Rp8000) - Stok : 28
- Machiato (Rp9500) - Stok : 20
- Piccolo (Rp10500) - Stok : 18
Pilih menu:
```

Catatan:

1. Item harus bertipe data STRING, dengan awalan huruf kapital, dan terdapat dalam program
 2. Stok harus bertipe data INT.
- c. Cari menu

Fitur ini digunakan untuk mencari menu yang ada didalam program.

Langkah-langkah:

1. Ketika pada tampilan awal, pilih 1.
2. Sistem akan menampilkan beberapa pilihan, termasuk Update menu.
Masukan pilihan = 4, lalu masukan pilihan = 3 untuk masuk ke fitur Cari menu.
3. Masukan nama item yang dicari
4. Setelah itu, program akan dijalankan dan akan mengecek apakah item yang ingin dicari ada atau tidak. Jika ada maka item tersebut akan ditampilkan oleh program, jika tidak maka program akan menampilkan "Item menu dengan nama Roti dan kue tidak ditemukan"

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 1

----Kelola Menu dan Harga----
1. Lihat menu
2. Tambah menu
3. Hapus menu
4. Update menu
9. Kembali
Masukkan pilihan: 4

----Update Menu----
1. Update harga menu
2. Update stok menu
3. Cari menu
9. Kembali
Masukkan pilihan: 3
Cari Menu

Masukan nama item yang dicari: Latte
```

Output:

```
----- Informasi Item Menu -----
Nama: Latte
Kategori: Minuman - Kopi
Harga: Rp20000
Stok: 30
```

Catatan:

1. Item harus bertipe data STRING, dengan awalan huruf kapital, dan terdapat dalam program.

4.2 Kelola bahan baku

Kelola bahan baku terdapat 6 fitur yang berisi,

A. Lihat semua bahan baku

Fitur ini digunakan untuk menampilkan semua bahan baku yang ada dalam program kami.

Langkah-langkah :

1. Saat program dijalankan, pada tampilan awal masukan pilihan = 2 untuk masuk ke dalam tampilan “Kelola bahan baku”
2. Masukan pilihan = 1 untuk menampilkan semua bahan baku yang ada.
3. Setelah itu, program akan menampilkan daftar bahan baku yang ada

Contoh pengisian :

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 2

-----Kelola Bahan Baku-----
1. Lihat semua bahan baku
2. Tambah bahan baku
3. Update harga
4. Update stok
5. Cari bahan baku
9. Back
Masukkan pilihan: 1
```

Output:

```
===== Daftar Bahan Baku =====
Paper - Harga: 2000 - Stok: 20
Lemon - Harga: 10000 - Stok: 500
Daun Bawang - Harga: 10000 - Stok: 500
Tomat - Harga: 10000 - Stok: 500
Daun teh english - Harga: 10000 - Stok: 500
Telur - Harga: 10000 - Stok: 500
Sosis - Harga: 10000 - Stok: 500
Daging ayam - Harga: 10000 - Stok: 500
Teh jasmine - Harga: 10000 - Stok: 500
Labu kuning - Harga: 10000 - Stok: 500
Olive hitam - Harga: 10000 - Stok: 500
Bungan chamomile - Harga: 10000 - Stok: 500
Salada - Harga: 10000 - Stok: 500
Selada - Harga: 10000 - Stok: 500
Roti hotdog - Harga: 10000 - Stok: 500
Roti burger - Harga: 10000 - Stok: 500
Buncis - Harga: 10000 - Stok: 500
Santan - Harga: 10000 - Stok: 500
Kismis - Harga: 10000 - Stok: 500
Terigu - Harga: 5000 - Stok: 500
Tepung - Harga: 5000 - Stok: 100
Wortel - Harga: 10000 - Stok: 500
Kacang hijau - Harga: 10000 - Stok: 500
```

B. Tambah bahan baku

Fitur ini digunakan untuk menambahkan bahan baku baru ke dalam program.

Langkah-langkah:

1. Saat program dijalankan, pada tampilan awal masukan pilihan = 2 untuk masuk ke dalam tampilan “Kelola bahan baku”
2. Pada tampilan “Kelola bahan baku”, Masukan pilihan = 2 untuk menambahkan bahan baku
3. Masukan nama bahan baku yang ingin ditambahkan
4. Masukan harga bahan baku
5. Masukan stok bahan baku
6. Jika data yang dimasukan sudah sesuai, maka program akan menyimpan data baru tersebut dan akan mengeluarkan tampilan “Bahan baku berhasil ditambahkan ke dalam hashtable”

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 2

-----Kelola Bahan Baku-----
1. Lihat semua bahan baku
2. Tambah bahan baku
3. Update harga
4. Update stok
5. Cari bahan baku
9. Back
Masukkan pilihan: 2
Tambah Bahan Baku
Masukkan nama bahan baku: Kemangi
Masukkan harga bahan baku: 5000
Masukkan stok bahan baku: 50
```

Output:

```
Kacang hijau - Harga: 10000 - Stok: 500
Es batu - Harga: 10000 - Stok: 500
Bawang merah - Harga: 10000 - Stok: 500
Jagung pipil - Harga: 10000 - Stok: 500
Bawang putih - Harga: 10000 - Stok: 500
Stalk saleri - Harga: 10000 - Stok: 500
Peterseli cincang - Harga: 10000 - Stok: 500
Kemangi - Harga: 5000 - Stok: 50
Susu steamed - Harga: 10000 - Stok: 500
Mentega - Harga: 10000 - Stok: 500
```

Catatan :

1. Nama bahan baku harus bertipe data STRING dan Kapital pada huruf awalnya
2. Harga bahan baku harus bertipe data INT

3. Stok bahan baku harus bertipe data INT
4. Setiap habis memasukan inputan, tekan enter pada keyboard anda.

C. Update harga

Fitur ini digunakan untuk memperbarui harga bahan baku yang ada pada program.

Langkah-langkah :

1. Saat program dijalankan, pada tampilan awal masukan pilihan = 2 untuk masuk ke dalam tampilan “Kelola bahan baku”
2. Pada tampilan “Kelola bahan baku”, Masukan pilihan = 3 untuk mengupdate harga
3. Masukan nama bahan baku
4. Masukan harga baru
5. Setelah itu, program akan dijalankan dan akan mengupdate harga baru pada bahan baku yang sudah ada.

Contoh pengisian program:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 2

----Kelola Bahan Baku----
1. Lihat semua bahan baku
2. Tambah bahan baku
3. Update harga
4. Update stok
5. Cari bahan baku
9. Back
Masukkan pilihan: 3
Update Harga Bahan Baku
Masukkan nama bahan baku: Bacon
Masukkan harga baru: 35000
```

Output:

Awal

```
Buah jeruk - Harga: 10000 - Stok: 500
Kopi bubuk - Harga: 10000 - Stok: 500
Bacon - Harga: 10000 - Stok: 500
Dressing caesar - Harga: 10000 - Stok: 500
Saus sambal - Harga: 10000 - Stok: 500
```

Akhir

```
Buah jeruk - Harga: 10000 - Stok: 500
Kopi bubuk - Harga: 10000 - Stok: 500
Bacon - Harga: 35000 - Stok: 500
Dressing caesar - Harga: 10000 - Stok: 500
Sirup gula - Harga: 10000 - Stok: 500
```

Catatan :

1. Nama bahan baku harus bertipe data STRING dan awalan kata pada nama bahan baku harus kapital, nama bahan baku harus sudah ada dalam program, jika tidak maka nama bahan baku tidak ditemukan atau bahkan akan eror.
2. Stok bahan baku harus bertipe data INT, jika tidak maka output yang ditampilkan tidak sesuai atau bisa terjadi eror.
3. Harga bahan baku harus bertipe data INT, jika tidak output yang ditampilkan tidak sesuai.

D. Update stok

Fitur ini digunakan untuk memperbarui stok bahan baku yang ada dalam program.

Langkah-langkah:

1. Saat program dijalankan, pada tampilan awal masukan pilihan = 2 untuk masuk ke dalam tampilan “Kelola bahan baku”
2. Pada tampilan “Kelola bahan baku”, Masukan pilihan = 4 untuk mengupdate stok bahan baku
3. Masukan nama bahan baku
4. Masukan stok bahan baku
5. Setelah itu, program akan dijalankan dan akan mengupdate stok bahan baku pada bahan baku yang sudah ada.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 2

-----Kelola Bahan Baku-----
1. Lihat semua bahan baku
2. Tambah bahan baku
3. Update harga
4. Update stok
5. Cari bahan baku
9. Back
Masukkan pilihan: 4
Update Stok Bahan Baku
Masukkan nama bahan baku: Bacon
Masukkan jumlah stok baru: 60
```

Output:

Awal

```
Buah jeruk - Harga: 10000 - Stok: 500
Kopi bubuk - Harga: 10000 - Stok: 500
Bacon - Harga: 10000 - Stok: 500
Dressing caesar - Harga: 10000 - Stok: 500
Sirup gula - Harga: 10000 - Stok: 500
```

Akhir

```
Buah jeruk - Harga: 10000 - Stok: 500
Kopi bubuk - Harga: 10000 - Stok: 500
Bacon - Harga: 10000 - Stok: 300
Dressing caesar - Harga: 10000 - Stok: 500
Sirup gula - Harga: 10000 - Stok: 500
```

Catatan:

1. Nama bahan baku harus bertipe data STRING dan awalan kata pada nama bahan baku harus kapital, nama bahan baku harus sudah ada dalam program, jika tidak maka nama bahan baku tidak ditemukan atau bahkan akan eror.
2. Stok bahan baku harus bertipe data INT, jika tidak maka output yang ditampilkan tidak sesuai atau bisa terjadi error.

E. Cari bahan baku

Fitur ini digunakan untuk mencari nama bahan baku yang ada pada program.

Langkah-langkah:

1. Saat program dijalankan, pada tampilan awal masukan pilihan = 2 untuk masuk ke dalam tampilan “Kelola bahan baku”
2. Pada tampilan “Kelola bahan baku”, Masukan pilihan = 5 untuk mencari nama bahan baku yang akan dicari.
3. Masukan nama bahan baku yang ingin dicari.
4. Program akan dijalankan dan akan mengecek apakah, inputan yang dimasukan ada atau tidak dalam program.

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] :
2

-----Kelola Bahan Baku-----
1. Lihat semua bahan baku
2. Tambah bahan baku
3. Update harga
4. Update stok
5. Cari bahan baku
9. Back
Masukkan pilihan: 5
Cari Bahan Baku
Masukkan nama bahan baku yang ingin dicari: Bacon
```

Output:

```
Bahan baku ditemukan: Bacon - Harga: Rp35000 - Stok: 60
```

Catatan:

1. Nama bahan baku harus bertipe data STRING, awalan kata pada nama bahan baku harus kapital, dan nama bahan baku harus sudah ada dalam program, jika tidak maka nama bahan baku tidak ditemukan atau bahkan akan eror.

4.3 Tutup

Fitur ini digunakan untuk menutup program, jika sudah selesai dijalankan

Langkah-langkah:

1. Saat program dijalankan, pada tampilan awal “WELCOME” masukan pilihan = 3 untuk masuk ke dalam tampilan “Tutup”.
2. Program akan dijalankan dan memberikan sebuah notifikasi yang berisi bahan baku yang stoknya berkurang, dan jumlah stoknya

3. Kemudian program berhenti

Contoh pengisian:

```
=====WELCOME=====
1. Kelola menu dan harga
2. Kelola bahan baku
0. Tutup
Masukkan pilihan [1/2/0] : 0
```

Output:

```
===== NOTIFIKASI =====
Bahan baku 'Spageti' stoknya kurang dari 10. Stok sekarang: 5
Terima kasih, sampai jumpa!
```

Catatan:

1. Nama bahan baku harus bertipe data STRING, awalan kata pada nama bahan baku harus kapital, dan nama bahan baku harus sudah ada dalam program, jika tidak maka nama bahan baku tidak ditemukan atau bahkan akan eror.