

Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2023-2024 / Genap
Modul : 3 – Strings & External Files
Nama Asisten / NIM : M. Rafli / 18321012
Nama Praktikan / NIM : Pradigta Hisyam Ramadhan / 18322008

BCL / Log Praktikum

Alasan Pemilihan Soal

Pada awalnya, soal nomor dua dipilih oleh penulis karena penulis berpikir bahwa kompleksitas algoritma yang diperlukan tidak terlalu rumit dan deklarasi variabel pada tiap fungsi tidak terlalu banyak. Selain itu, soal dipilih karena pada saat merancang strategi awal/algoritma, hanya memerlukan lebih sedikit fungsi daripada soal nomor satu dan pemrosesan string yang tidak terlalu rumit. Meskipun pada saat pengerjaan *source code*, terdapat beberapa kendala yang akan dibahas pada subbagian *Kejadian Saat Praktikum*.

Strategi Awal/Rancangan Algoritma

1. Pembacaan File: Program akan membaca nama file sesuai dengan masukkan pengguna dan memvalidasi ada tidaknya file dalam direktori yang sama dengan *source code*. Jika nama file tidak ada pada direktori yang sama dengan *source code*, maka program akan berhenti dan menampilkan pesan error. Sebaliknya, jika nama file ditemukan, maka pemrosesan string akan dilakukan.

2. Algoritma Pemrosesan String: Awalnya, setiap string pada file akan ditampung pada sebuah variabel dengan tipe data *array of characters* dengan ukuran dua dimensi, dimensi pertama untuk banyak baris pada file dan dimensi kedua untuk banyak karakter pada tiap barisnya. Kemudian, setiap elemen *array of characters* (tiap baris pada file) akan diproses menggunakan fungsi bawaan dari *header string.h* dan *stdlib.h*.

Kejadian Saat Praktikum

Pada saat penulis akan menyalin string pada baris pertama ke dalam variabel lokal fungsi *getDate*, terdapat kendala untuk menghilangkan karakter titik ('.'). Awalnya penulis akan menggunakan iterasi untuk menghilangkan karakter tersebut, akan tetapi asisten memberi saran untuk menggunakan fungsi *strtok* sehingga karakter yang di-assign pada variabel lokal akan berhenti sebelum karakter titik ('.'). Hal ini membuat fungsi *getDate* jauh lebih ringkas dan mudah untuk dipahami. Karena saran inilah penulis mendapatkan ide untuk menampung semua semua *string* yang akan diproses ke dalam variabel bertipe *pointers of char* dengan menggunakan fungsi *strtok*.

Beberapa masalah muncul ketika akan melakukan perbandingan dua string yang merepresentasikan jam dan menit untuk menentukan masuk-tidaknya barang ke dalam toko pada fungsi *getSembako*. Masalah utama dari fungsi ini terletak pada logika kondisional yang digunakan, yaitu dengan membandingkan secara langsung nilai waktu dalam format (HH.MM). Sehingga, segera setelah bertanya dengan asisten dan penulis disarankan untuk memisahkan jam dan waktu untuk dibandingkan, permasalahan ini dapat diselesaikan. Akan tetapi, masih terdapat masalah pada *if-else statement* fungsi sehingga ada *overlap* dan tidak menghasilkan apapun pada terminal output. Fungsi *getSembako* merupakan inti dari program penulis dan ditulis dalam durasi yang cukup lama karena mempertimbangkan berbagai macam isi test file.

Tahapan Strategi (Milestone)

1. Pengambilan input nama file

Pada awal program, pengguna diminta untuk menuliskan nama file yang berada di direktori yang sama dengan program. Kemudian, program akan memvalidasi keberadaan file tersebut. Jika file tidak ditemukan, program akan menampilkan pesan error dan keluar dari program.

2. Penyalinan string pada tiap baris di file ke dalam array

Setelah file berhasil dibuka, program akan memindahkan string yang ada pada setiap baris ke dalam array karakter dua dimensi dengan indeks yang sesuai dengan baris string di file. Tujuan pemindahan ini adalah untuk mempermudah penulis dalam melakukan pemrosesan string.

3. Menampilkan tanggal masuknya stok sembako

Tiap file studi kasus memiliki format yang sama satu sama lainnya, yaitu terletak pada baris pertama di dalam filenya. Oleh karena itu, untuk mengakses tanggal masuk stok sembako, pengguna membuat sebuah fungsi yang akan menerima string dari elemen pertama array yang berisikan konten file. Karena string yang dipindahkan dari file masih mengandung karakter '.' di dalamnya, maka digunakan fungsi *strtok* untuk memperoleh string elemen array dengan delimiter berupa '.'.

4. Pemrosesan String konten

Setelah mendapatkan tanggal kedatangan sembako, setiap baris dari file hanya akan berisi konten dengan format yang sama; HH.MM barang-jumlah barang. Oleh karena itu, penulis menggunakan for-loop untuk mengakses array buatan dari urutan kedua sampai akhir untuk mendapatkan nilai jam, menit, jenis barang, dan jumlah barang dengan menggunakan fungsi *strtok*. Kemudian, fungsi akan melakukan proses kondisional untuk memeriksa validitas jam pengiriman sembako, dan apabila kondisi terpenuhi, maka akan ditampilkan jenis barang dan jumlahnya. Terakhir, fungsi akan menampilkan total sembako dan jenisnya di akhir fungsi. Pada kasus lain, yaitu apabila jam pengiriman tidak valid, maka jenis dan jumlah barang hanya ditampilkan di akhir program. Beberapa pernyataan pengkondisian if-else juga digunakan untuk menangani beberapa kasus khusus.

Proses Debugging

Pada praktikum modul 3 ini, penulis melakukan proses *debugging* untuk mengoptimalkan fungsi-fungsi yang dibuat sehingga dapat memenuhi *test case* yang diberikan. Setelah dirasa cukup memenuhi tiga *test case* pertama yang diberikan, penulis melakukan *commit* dan mendapati bahwa terdapat masalah pada *test case* kedua dan kesembilan. Untuk *test case* kedua, awalnya penulis mengira bahwa jika tidak ada barang sembako yang dikirim pada tanggal tersebut, maka akan muncul sebagai error. Oleh karena itu, untuk menangani permasalahan tersebut, penulis mengubah syntax untuk keluar dari program dari

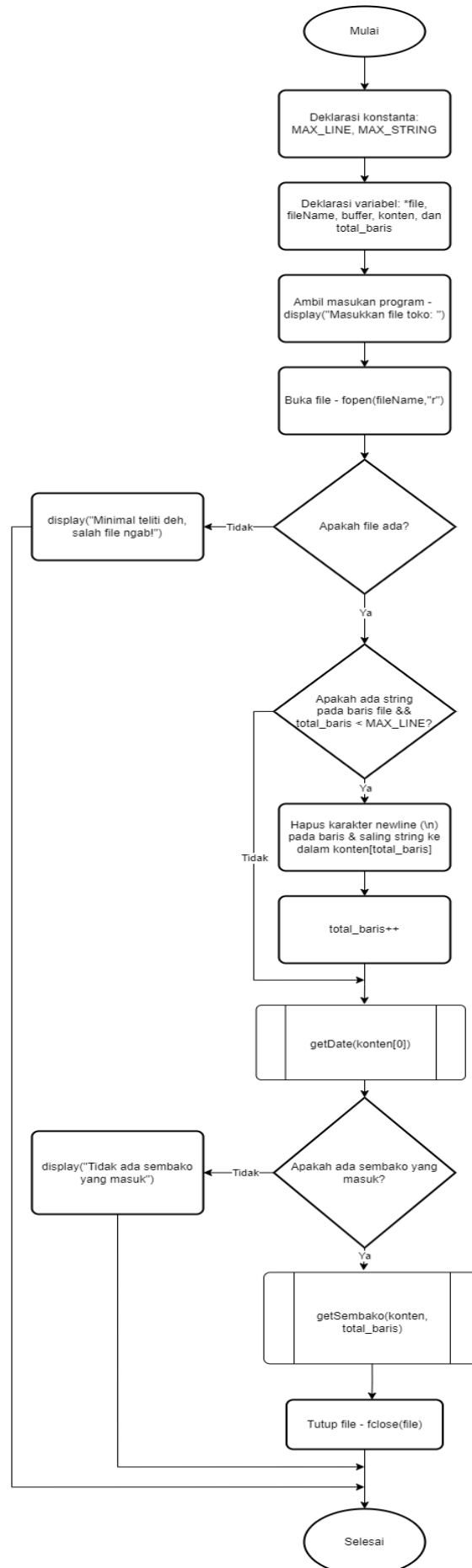
return 1;

menjadi

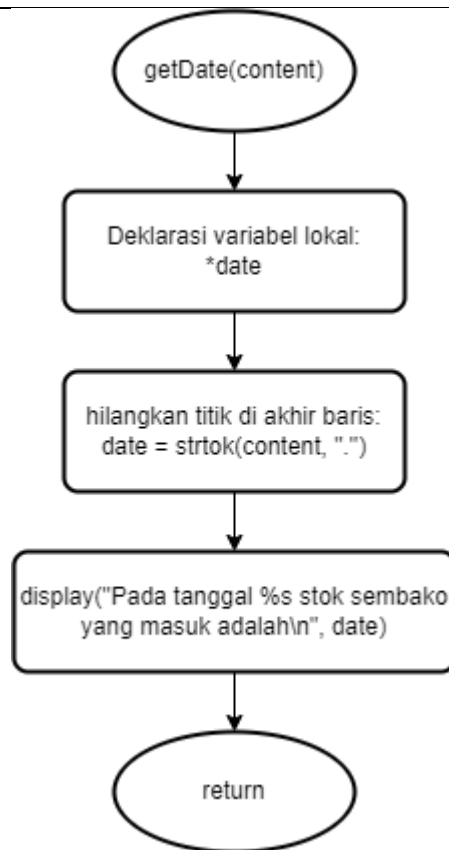
return 0;

Untuk *test case* terakhir, jika hanya terdapat dua barang yang ditolak akibat tutupnya toko, maka akan digunakan pengkondisian lain untuk kasus tersebut.

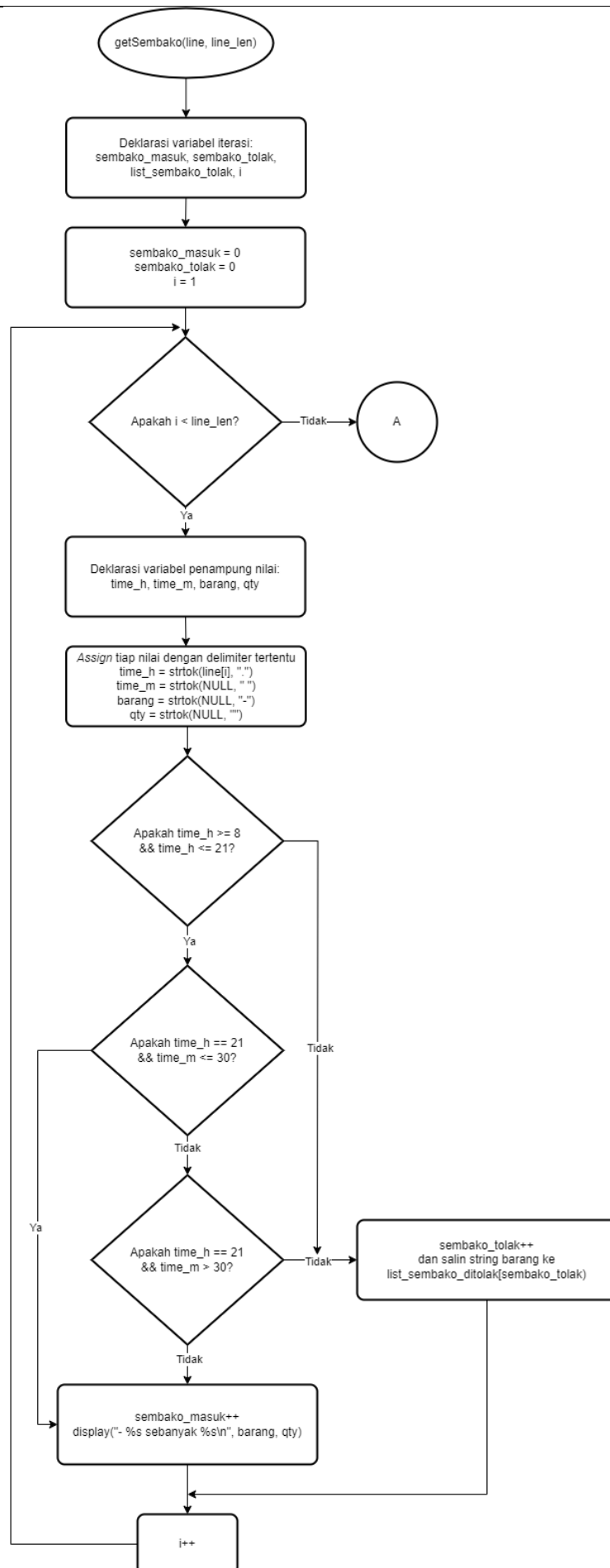
Diagram Alir



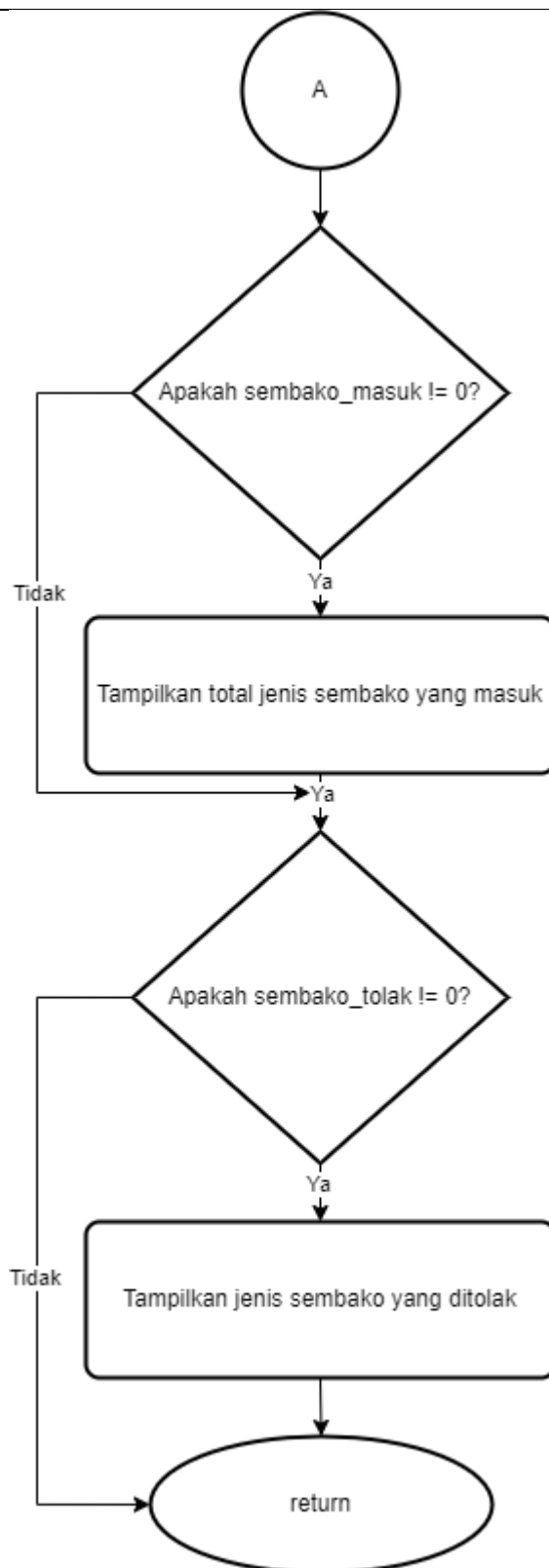
Gambar 1. Diagram alir program utama



Gambar 2. Diagram alir fungsi *getDate*

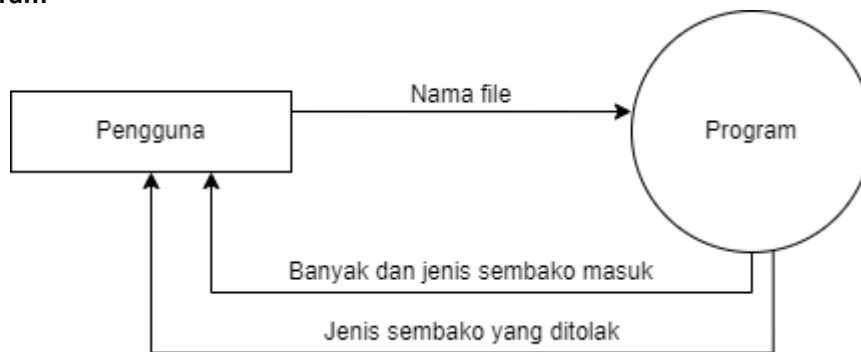


Gambar 3. Diagram Alir fungsi *getSembako* (1)

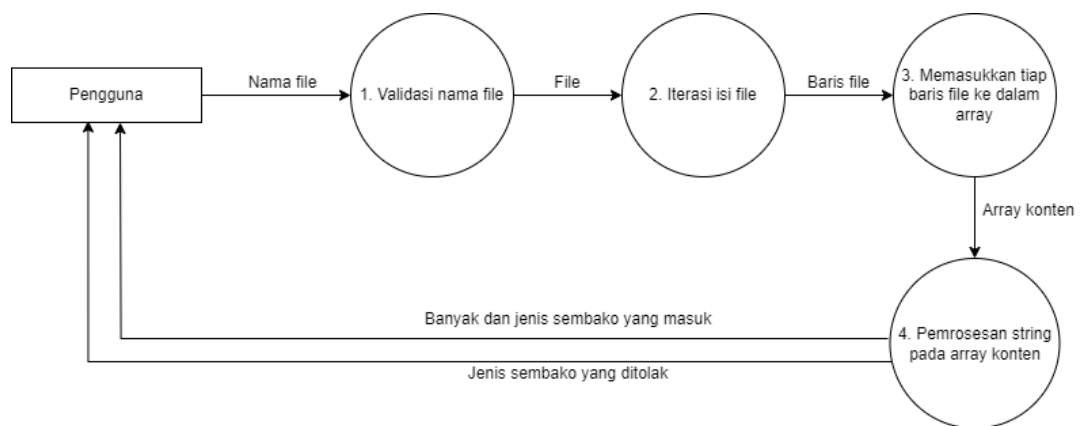


Gambar 4. Diagram alir fungsi *getSembako* (2)

Data Flow Diagram



Gambar 5. Data flow diagram level 0



Gambar 6. Data flow diagram level 1