

Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2023-2024 / Genap
Modul : 5 – Structure & Dynamic Array
Nama Asisten / NIM : Jasmine Callista A. I. / 18321008
Nama Praktikan / NIM : Pradigta Hisyam Ramadhan / 18322008

BCL / Log Praktikum

Alasan Pemilihan Soal

Pada praktikum modul ini, penulis memilih soal kedua karena lebih suka soal yang tidak memerlukan perhitungan matematis dalam program. Selain itu, penulis merasa lebih terbiasa dengan soal pemrosesan string, sehingga memilih soal kedua. Namun, penulis mengalami kendala sehingga tidak dapat menyelesaikan soal yang diberikan. Kendala tersebut akan dibahas pada bagian *kejadian saat praktikum*.

Strategi Awal/Rancangan Algoritma

1. Pengambilan input nama file: Pada awalnya, program akan menerima masukan dari pengguna berupa nama file pada direktori yang sama dengan file. Apabila nama file tidak ditemukan pada direktori yang sama, maka akan ditampilkan pesan error.

2. Menampilkan tagihan dari file: Setelah file dibuka, maka program akan menampilkan daftar tagihan berupa jenis menu, banyak pesanan, dan total harga tiap menu. Selain itu, program akan menampilkan total tagihan secara keseluruhan dari menu-menu yang telah dipesan.

3. Pengambilan input dari pengguna: Setelah daftar tagihan ditampilkan, pengguna akan memberi masukan berupa jumlah pemesanan, nama pemesan, dan juga total pesanan tiap orang terhadap tiap menu yang ditampilkan.

4. Pemrosesan *string* untuk menghitung total tagihan: Setelah semua daftar pemesan beserta pesanannya dimasukan oleh pengguna, maka program akan menghitung total tagihan yang harus dibayar oleh setiap pemesan pada akhir program.

Kejadian Saat Praktikum

Selama praktikum, penulis tidak berhasil menyelesaikan soal kedua. Kesulitan penulis dalam **memproses angka dari masukan pengguna dengan data yang ada di dalam file** menjadi **masalah utama** yang menghambat penyelesaian tugas tersebut. Selain itu, penulis menghabiskan banyak waktu untuk memberikan nilai *return* dalam bentuk tipe data *double pointer* pada fungsi yang dibuat penulis. Pengalokasian memori untuk setiap *double pointer* dan *pointer of string* juga menjadi tantangan tersendiri bagi penulis, karena penulis ingin menggunakan array dinamis daripada array statis dalam programnya.

Tahapan Strategi (*milestone*)

1. Pemrosesan string pada file: Untuk mempermudah penulis dalam melakukan pemrosesan *string* pada tiap baris file, penulis melakukan parse pada tiap baris file dan memasukkan *string* pada setiap baris ke dalam array 2-D. Total baris juga dihitung ketika parse baris berlangsung. Setelah itu, pada fungsi *parsePesanan*, array akan dipindahkan ke dalam struktur sesuai dengan keterangan masing-masing data menggunakan fungsi *strtok*. Setelah dipindahkan ke struktur, maka fungsi akan menampilkan data tagihan yang harus dibayarkan.

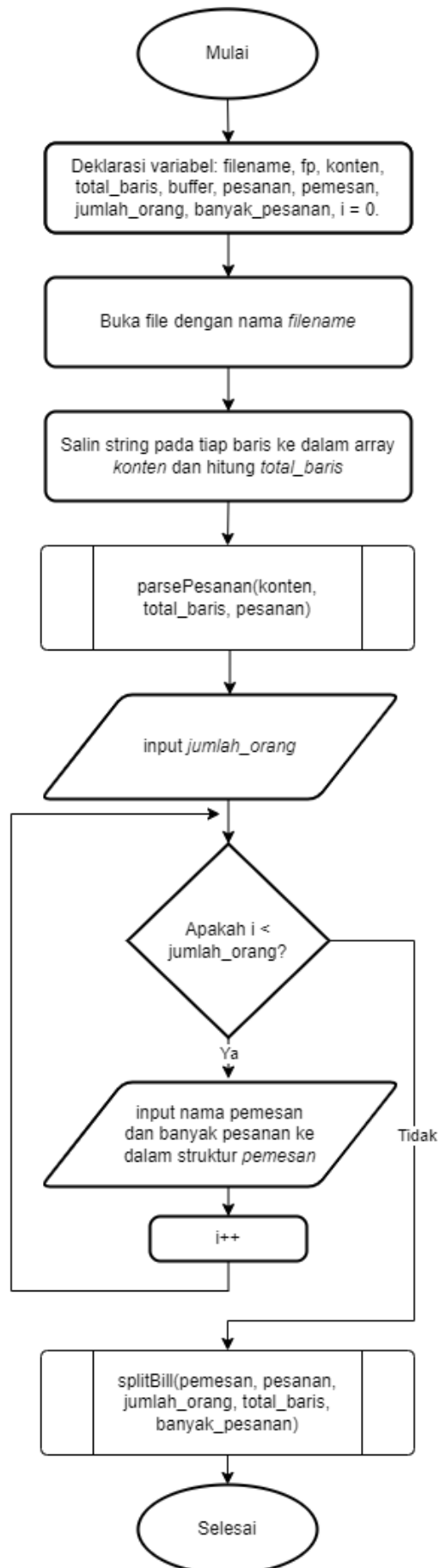
2. Proses penghitungan *split bill*: Setelah struktur terbentuk dan semua data pemesan sudah dimasukan oleh pengguna, maka total tagihan yang harus dibayarkan oleh setiap pemesan akan ditampilkan pada akhir program. Fungsi yang mengakomodasi proses ini adalah fungsi *splitBill*, tetapi

fungsi ini tidak dapat penulis selesaikan karena penulis mengalami kendala dalam memikirkan algoritma untuk menghitung total tagihan tiap orang berdasarkan input masukan pengguna berupa "Nama(spasi)total_pesanan". Awalnya penulis menggunakan algoritma pemrosesan berdasarkan hasil bagi 10, tetapi fungsi belum dapat bekerja secara optimal dan terjadi *error* dari tipe data yang dikembalikan oleh fungsi.

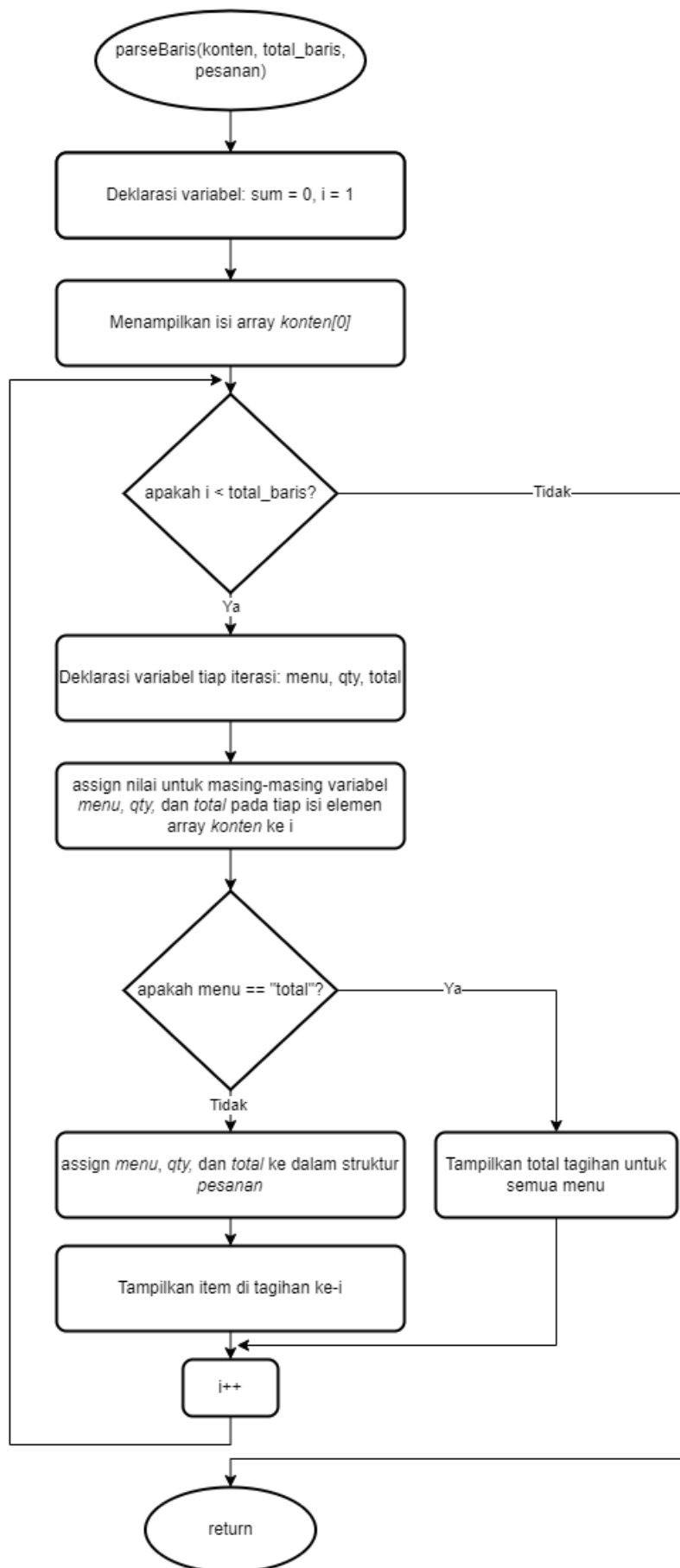
Proses Debugging

Penulis tidak sempat melakukan proses debugging untuk menguji tiap *test case autograder* karena waktu penulis terbuang cukup banyak pada awal praktikum untuk menyelesaikan persoalan jenis tipe data balikan pada fungsi *parseBaris*. Proses *debugging* yang dilakukan oleh penulis terbatas hanya pada cara program untuk dapat menampilkan tagihan pada awal program, memindahkan data dari array ke dalam struktur, dan menghitung total tagihan oleh setiap orang.

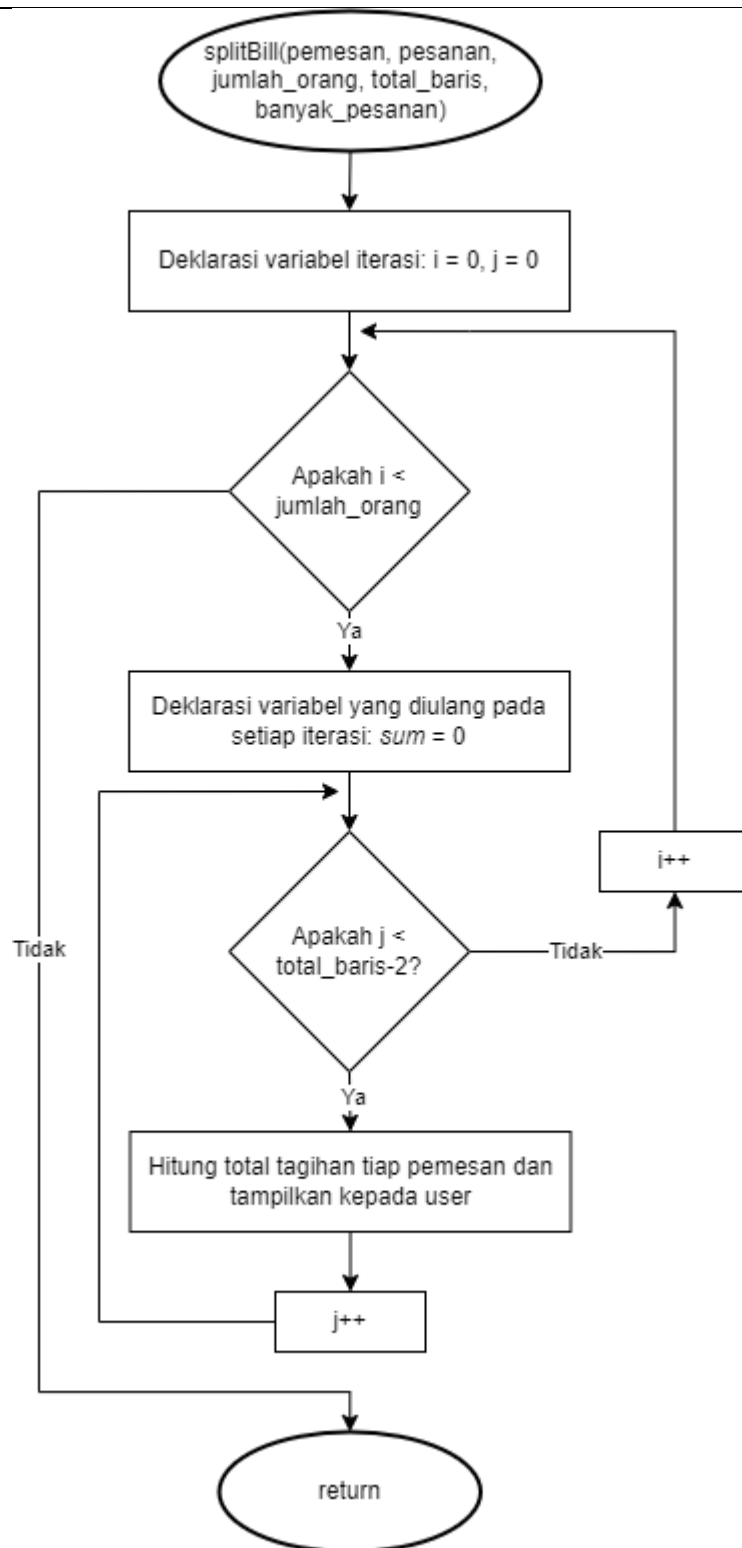
Diagram Alir/flowchart



Gambar 1. Diagram alir fungsi utama

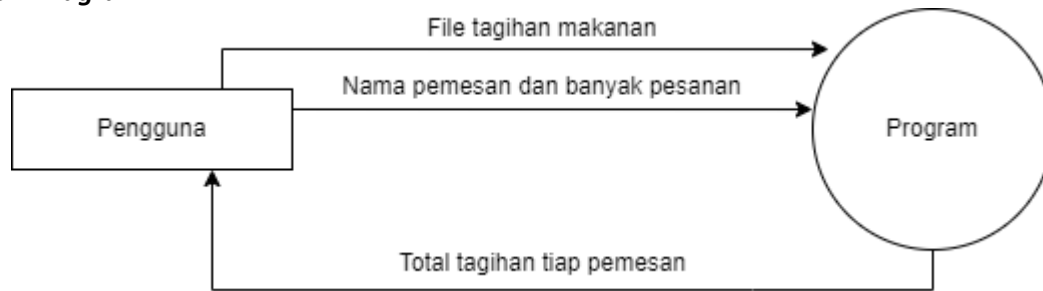


Gambar 2. Diagram alir fungsi *parseBaris*

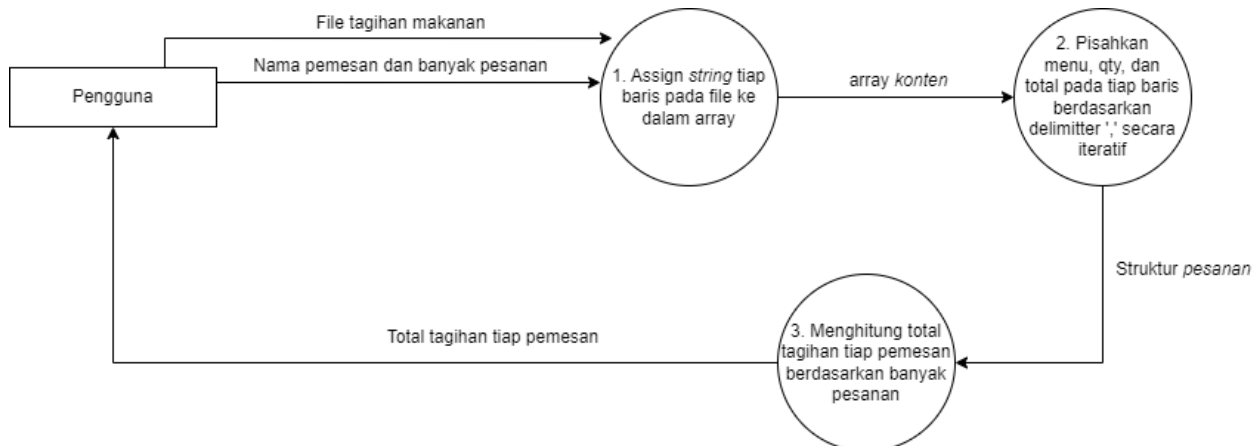


Gambar 3. Diagram alir fungsi *splitBill*

Data Flow Diagram



Gambar 4. Data flow diagram level 0



Gambar 5. Data flow diagram level 1