Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2023-2024 / Genap
Modul : 4 – Recursion
Nama Asisten / NIM : Reynaldo Averill / 13219071
Nama Praktikan / NIM : Pradigta Hisyam Ramadhan / 18322008

BCL / Log Praktikum

Alasan Pemilihan Soal

Pada praktikum modul ini, penulis memilih soal pertama karena program tidak memiliki pengulangan untuk memvalidasi input. Dalam arti lain, program hanya sekali jalan saja. Selain itu, program hanya menerima input *string* saja yang diberikan ke dalam satu baris. Bagian rekursif dari soal nomor satu juga cukup sederhana, yaitu dengan menentukan validitas jumlah kata tiap baris yang mengurut terbesar/sama dengan baris sebelumnya.

Strategi Awal/Rancangan Algoritma

1. Menentukan banyak baris: Pada awalnya, program akan menentukan banyak baris yang akan digunakan penulis untuk membuat puisi

2. Pengambilan input string: Input yang diberikan pengguna kepada program adalah sebuah string sebanyak n-baris.

3. Validasi konten baris: Untuk setiap barisnya, program akan melakukan validasi terhadap input yang dimasukkan oleh pengguna. Validasi yang dilakukan adalah menentukan apakah jumlah kata tiap barisnya selalu lebih dari 1/sama dengan jumlah kata dari baris sebelumnya.

Kejadian Saat Praktikum

Pada saat praktikum dimulai, penulis mengalami kesulitan dalam melakukan pengambilan input *string* dari pengguna. Beberapa fungsi yang disediakan oleh *library string.h* sudah dicoba oleh penulis seperti *fgets*, *gets*, dan *getchar*. Akibatnya, penulis menghabiskan waktu hampir satu jam untuk memikirkan cara menerima input dari pengguna. Masalah dapat terselesaikan setelah asisten praktikum memberikan arahan tentang cara mengambil input sebanyak satu baris dengan menggunakan fungsi *scanf*.

Selain itu, penulis juga cukup kesulitan ketika akan melakukan *passing parameter* berupa *array of integer* dari subprogram ke dalam program utama. Penulis mendapati *error* yang menyebutkan bahwa alamat yang di-*passing* ke dalam program utama adalah memori dari variabel lokal. Untuk menyelesaikan persoalan tersebut, penulis melakukan alokasi memori terhadap *array of integer* yang akan di-*passing*.

Tahapan Strategi (milestone)

1. Pengambilan *input* string tiap baris

Pengambilan input dilakukan dengan menggunakan fungsi *scanf* yang dimodifikasi sehingga dapat menerima input *string* satu baris penuh beserta dengan spasi-nya, alih-alih menerima hanya satu *string* saja. Selanjutnya, *string* pada tiap baris ini akan disebut dengan konten. Banyak konten yang di-*assign* selaras dengan banyak baris yang diberikan pengguna pada awal program.

2. Menyalin string ke dalam array

Konten-konten kemudian di-*assign* ke dalam *arr of strings* untuk mempermudah pemrosesan data. Proses *assign* konten ini dilakukan dalam satu fungsi bernama *getContent*, bersamaan dengan proses pengambilan input *string* oleh pengguna.

3. Menghitung jumlah kata tiap baris

Untuk menghitung jumlah kata tiap konten, penulis membuat sebuah fungsi bernama *countWord* yang menerima sebuah *string* untuk dihitung jumlah katanya. Fungsi ini menggunakan pendekatan iteratif dan untuk memisahkan kata-perkatanya, penulis menggunakan pemisah tanda spasi (‘ ’). Fungsi akan mengembalikan nilai jumlah kata pada *string* dalam tipe data *integer*.

4. Validasi jumlah kata dalam *array of integer*

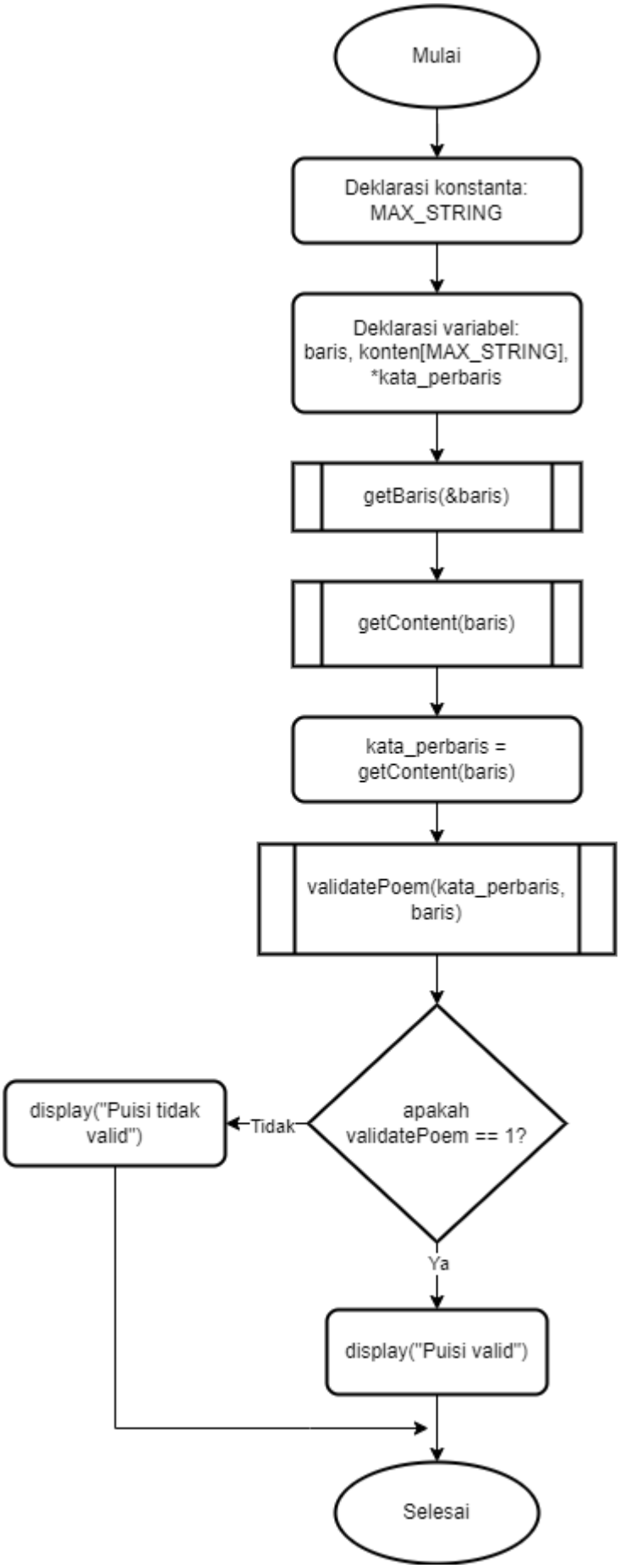
Untuk menentukan validitas konten yang diberikan oleh pengguna, program akan melakukan pengulangan untuk mengecek keterurutan jumlah kata pada konten. Apabila jumlah kata mengurut dari kecil ke besar, maka masukan valid. Fungsi yang digunakan untuk mengimplementasi fungsional ini adalah fungsi *validatePoem*. Fungsi ini menggunakan pendekatan rekursif dan akan mengembalikan nilai satu apabila baris terurut dari kecil ke besar. Apabila tidak urut/urutan terputus, maka akan mengembalikan nilai nol.

Proses Debugging

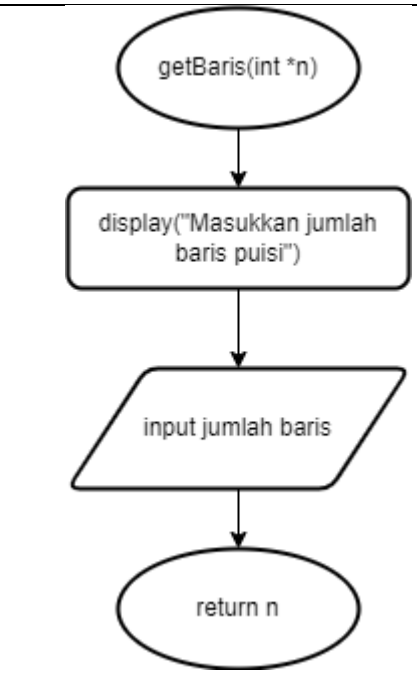
Sebelum melakukan *commit* ke repo github, penulis melakukan *debugging* untuk mengecek nilai *string* yang tersimpan pada *array of strings* setelah diberi masukan oleh pengguna. Selain itu, penulis juga melakukan *debugging* pada fungsi *getContent* yang sempat bermasalah akibat alamat parameter yang di-*passing* ke dalam fungsi utama belum dialokasikan memorinya.

Setelah algoritma utama program selesai dan penulis telah melakukan *commit* ke repo github, penulis melakukan *debugging* pada kondisional fungsi *validatePoem* karena terdapat beberapa *test case* yang belum dievaluasi dengan benar. Fungsi ini juga sempat mengalami *infinte loop* karena kesalahan kasus basis pada fungsi.

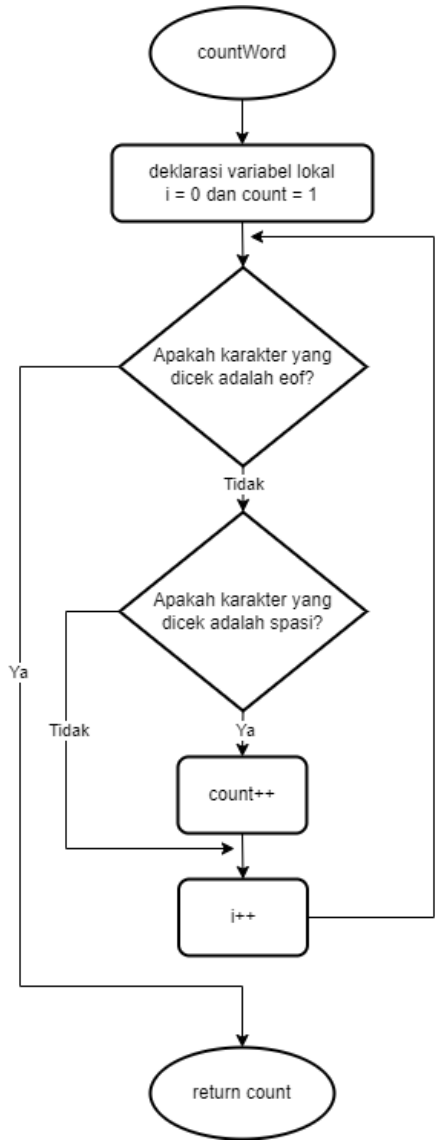
Diagram Alir



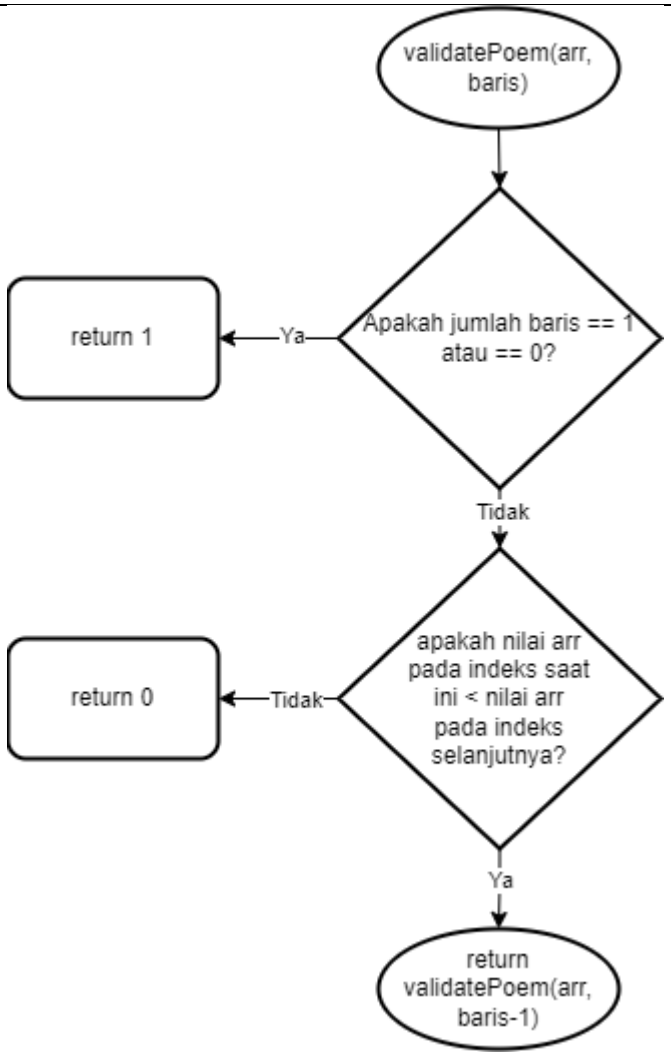
Gambar 1. Diagram alir fungsi utama



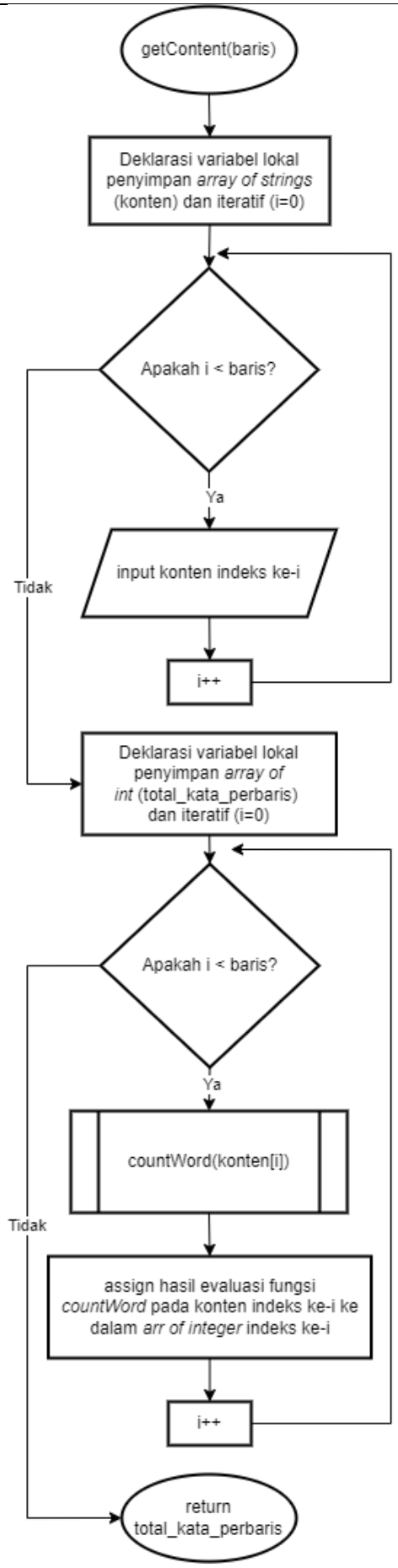
Gambar 2. Diagram alir fungsi *getBaris*



Gambar 3. Diagram alir fungsi *countWord*

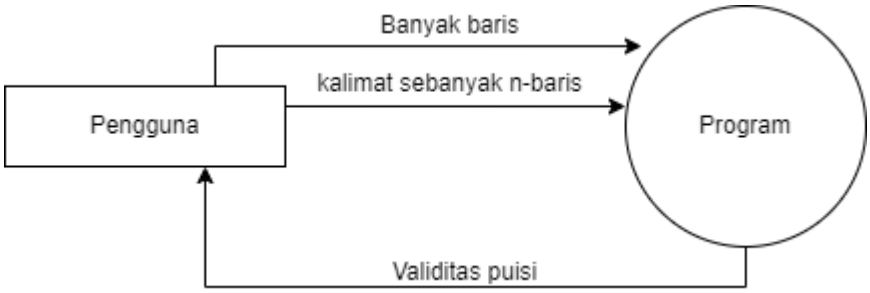


Gambar 4. Diagram alir fungsi *validatePoem*

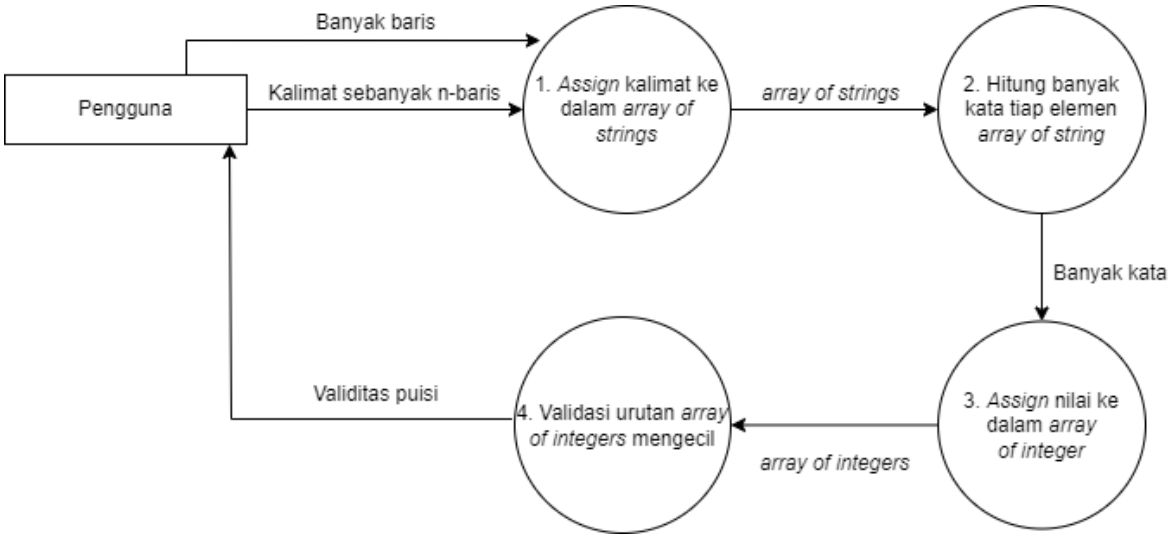


Gambar 5. Diagram alir fungsi *getContent*

Data Flow Diagram



Gambar 6. *Data flow diagram* level 0



Gambar 7. *Data flow diagram* level 1