**PRAKTIKUM KECERDASAN BUATAN**

**SEMESTER GASAL 2018/2019**

**MODUL 03**

**Sequential File dan Data Base**

****

**OLEH:**

**JO VIANTO/ 170709173**

**C**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS ATMA JAYA YOGYAKARTA**

**2018**

Praktikum kali ini bertujuan untuk mempelajari file handling pada prolog. Terdapat dua jenis file handling yang dipelajari. Yang pertama berupa file biasa dan yang kedua berupa basis data. Kedua jenis file tersebut mempunyai kapasitas masing-masing, dengan kelebihan dan kekurangan masing-masing.

Pada laporan ini saya tidak akan mengulang pembahasan yang sudah dibahas pada laporan sebelumnya seperti cara membuat menu, menyeleksi pilihan dan perulangan, serta lainnya. Laporan ini akan menitik-beratkan kepada pembahasan file sekuensial dan basis data.

**Admin**

Pertama-tama, buatlah predicates isi\_db, loaddb, savedb. Ketiga predicates tersebut tidak memperlukan parameter. Setiap predicates ini akan berfungsi sebagai pengontrol data antara program dengan pangkalan data. savedb digunakan untuk data ke database. loaddb digunakan untuk menginisialisasi data, dan isi\_db untuk menginisialisasi loaddb dan savedb. Kemudian siapkan domains nama, kategori sebagai string, npm = integer, file = fin;fout. Setelah itu deklarasikan

facts- datamenu

data\_menu(nama, harga, kategori)

hal ini memberitahu kompiler jika database datamenu merupakan objek data\_menu dengan data-data berupa nama, harga, dan kategori.

Klausa isi\_db berfungsi untuk memanggil loaddb dan savedb maka clausa tersebut ditulis seperti isi\_db :- loaddb, savedb. Untuk loaddb sendiri berisi {{ assertz(data\_pribadi("soto", 15000, "makanan"),datapribadi), }} sebanyak data yang ada untuk diinisialisasikan. assertz akan mengisi data ke bagian terakhir dari list data setiap kali dipanggil. Diakhir data jangan lupa assertz data dengan nilai 0 semua, sebagai sebuah poin agar kita tahu kapan harus berhenti menelusuri database. klausa savedb hanya terdapat pemanggilan sebuah fungsi bawaan save("D:\menuresto\_9173.dat",datapribadi). Sehingga data-data yang ada pada database datapribadi akan disimpan kedalam data menuresto\_9173.dat.

Sekarang, buatlah sebuah predikat dan klausa-klausa untuk cekLogin sebagai berikut :

cekLogin(U,Y):-U="admin",Y="9173", go.

cekLogin(U, Y):-U="waiter",Y="3719",go1.

cekLogin(\_, \_):- write("GAGAL"),login.

cekLogin akan mengalihkan program ke klausa go apabila kondisi U, dan Y sesuai, jika tidak maka akan melakukan pengecekan pada klausa kedua, hingga yang terakhir yang merupakan bagian terkecuali dan menampilkan error dan meminta user untuk input login lagi.

Klausa menambah\_data diawali dengan meminta inputan nama, harga, dan kategori kepada user. Setelah itu panggil asserta(data\_menu(Nama, Harga, Kategori)), penggunaan asserta sendiri bertujuan untuk memasukkan data dari atas, sehingga data paling terakhir tetap data sign dengan nilai 0 sebelumnya. Setelah itu simpan ke file dengan memanggil klausa savedb.

Pada klausa menampilkan\_data(N, H, K) yang berasal dari predikat menampilkan\_data(nama, harga, kategori). Ketika klausa ini dipanggil, maka ambil data\_pribadi(N, H, K) maka program akan mengembalikan nilai data\_pribadi yang sedang ditunjuk oleh kursornya. Selanjutnya lakukan cek apakah N <> “0” dan lakukan pemanggilkan menampilkan\_data(“”, \_, \_). Pada klausa menampilkan\_data yang kedua adalah ketika kondisi N = “0” maka itu mengindikasikan database telah selesai dibaca dan tidak perlu memanggil menampilkan\_data lagi.

Penghapusan menu dilakukan dengan memanggil fungsi retractall(facts, database) dimana facts adalah objek beserta kolom yang akan kita jadikan acuan untuk menghapus data yang bernilai sama dengan kolom yang ada di database, contoh untuk menghapus data nama maka menggunakan retractall(data\_pribadi(Nama,\_,\_), datapribadi).

**Waiter**

Pada menu menambahkan pesanan(waiter) disyaratkan menggunakan file sekuensial. Buatlah inputan untuk nama makanan dan minuman, setelah itu openread file dan pindahkan writedevice ke fout. setiap inputan makanan dan minuman maka lakukan pemanggilan cari\_data(B,\_, D, \_), cari\_data akan mencetak nilai yang ditemukan ke file dan akan mengembalikan nilai harga pada variabel ketiga, nilai tersebut ditampung kemudian di hitung dan cetak ke file juga. Sehingga akan menjadi seperti begini :

openwrite(fout, "D:\order\_9173.txt"),

write("masukkan nama :"), readln(A),

write("makanan :"), readln(B), write("minuman :"), readln(C),

writedevice(fout),

...

cari\_data(B ,\_, D, \_),nl , cari\_data(C, \_, E, \_), nl,F = D+E,

...

closefile(fout), writedevice(screen),go1.

Cari\_data sendiri sama seperti menampilkan\_data hanya saja terdapat 1 tambahan parameter sebagai nilai komparasi, apabila nilai tersebut sama dengan nama produk yang sedang ditelusuri maka program akan mencetak nilai dan akan dicetak ke file.

Pada menu menampilkan pesanan, kita lakukan dulu pembacaan file tujuan, dengan fungsi openread(pointer, lokasi), setelah itu ganti readdevice(pointer) disini dengan tujuan agar ketika kita memanggil fungsi inputan, maka otomatis akan membaca dari file ynag bersangkutan. Maka dengna menggunakan perulangan telusuri setiap baris kata di file hingga kondisi **eof**, dimana fungsi tersebut:

while(Finput) :-

not(eof(Finput)),

readln(Dtread),

write(Dtread),nl,

while(Finput).