**LAPORAN FINAL PROJECT**

**SISTEM OPERASI STUDI KASUS**

**MERCHANT MANAGEMENT SYSTEM**

Mata Kuliah Sistem Operasi

Dosen Pengampu :

Hendra Maulana, S.Kom, M.Kom



Disusun Oleh :

Mochammad Arya Salsabila (19081010001)

Muhammad Lutfi Pratama (19081010049)

Kelas D

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UPN “VETERAN” JAWA TIMUR**

**2020**

STUDI KASUS

Dalam pengerjaan Final Project mata kuliah Sistem Operasi ini kelompok kami memilih studi kasus Merchant Management System. Inti dari program ini dalam menjalankan operasi CRUD (Create, Read, Update, Delete). Sesuai kepanjangannya CRUD ini digunakan untuk membuat, membaca, mengupdate, dan menghapus sebuah data. Dalam program yang kami kembangkan ini kami melakukan CRUD terhadap data barang. Operasi CRUD ini akan memanipulasi data barang yang berupa array kode\_barang, nama\_barang, dan kode\_barang. Data barang yang digunakan ialah kode barang, nama barang, dan harga barang. Program ini tidak hanya bisa membaca semua data yang ada tetapi juga bisa membaca suatu data tertentu. Selain operasi CRUD program ini juga menerapkan beberapa perintah bash seperti menyisipkan suatu teks ke dalam file berekstensi .txt, membuat folder, dan mengecek apakah suatu file dan folder sudah ada atau belum.

PEMBAHASAN SOURCE CODE

A. DEKLARASI ARRAY

declare -a kode\_barang

declare -a nama\_barang

declare -a harga\_barang

Deklarasi array kode\_barang, nama\_barang, harga\_barang secara explicit. Array tersebut dideklarasikan secara explicit karena kita akan menetapkan nilainya di proses setelahnya.

B. DEKLARASI FUNGSI

function is\_laporan\_exists() {

    if [[ ! -e ./laporan/$kode\_laporan.txt ]]

    then

        status=true

    fi

}

Fungsi is\_laporan\_exists() digunakan untuk mengecek apakah file $kode\_laporan.txt sudah ada di dalam folder laporan atau belum, jika belum maka akan mengisi nilai variabel status menjadi true.

function is\_folder\_exists() {

    if [[ ! -d ./laporan ]]

    then

        mkdir laporan

    fi

}

Fungsi is\_folder\_exists() digunakan untuk mengecek apakah folder laporan sudah ada atau belum, jika belum maka akan dibuat folder baru dengan perintah mkdir dan nama laporan.

function is\_barang\_exists() {

    for (( i=0;i<${#kode\_barang[@]};i++ ))

    do

        if [[ $kode\_barang\_in == ${kode\_barang[$i]} ]]

        then

            status=true

            index=$i

            break

        fi

    done

}

Fungsi is\_barang\_exists() melakukan looping sebanyak jumlah elemen array kode\_barang. Jumlah elemen kode barang bisa didapat dengan syntax ${#kode\_barang[@]}. Jika dalam perjalanan looping kode\_barang\_in nilainya sama dengan kode\_barang[$i] maka looping akan dihentikan dengan break lalu mengisi variabel status dengan true dan variabel index dengan $i.

function insert() {

    status=false

    n=${#kode\_barang[@]}

    read -p "Kode barang yang ingin ditambahkan : " kode\_barang\_in

    echo ""

    is\_barang\_exists

    if [[ $status == true ]]

    then

        echo -e "Barang sudah ada..\n"

    else

        echo -e "Barang belum ada, lanjut..\n"

        kode\_barang[$n]=$kode\_barang\_in

        echo "Kode Barang  : ${kode\_barang[$n]}"

        read -p "Nama Barang  : " nama\_barang[$n]

        read -p "Harga Barang : " harga\_barang[$n]

        echo -e "\nBarang berhasil ditambahkan..\n"

    fi

}

Fungsi insert() digunakan untuk menambahkan barang. Diawali dengan input kode barang lalu dicek apakah kode barang tersebut belum ada dengan memanggil fungsi is\_barang\_exists(). Jika status bernilai true maka kode barang sudah ada, sebaliknya jika status bernilai false maka kode barang belum ada dan akan dilanjutkan dengan input nama barang dan harga barang. Kode barang, nama barang, harga barang masing masing akan ditambahkan ke array kode\_barang[$n], nama\_barang[$n], harga\_barang[$n]. n adalah index yang diambil dari panjang array kode\_barang.

function search() {

    status=false

    read -p "Kode barang yang ingin dicari : " kode\_barang\_in

    echo ""

    is\_barang\_exists

    if [[ $status == true ]]

    then

        echo "Kode Barang   : ${kode\_barang[$index]}"

        echo "Nama Barang   : ${nama\_barang[$index]}"

        echo "Harga Barang  : ${harga\_barang[$index]}"

        echo -e "\nBarang berhasil ditemukan..\n"

    else

        echo -e "Barang tidak ditemukan..\n"

    fi

}

Fungsi search() digunakan untuk mencari barang dengan menginputkan kode barang terlebih dahulu lalu dicek apakah kode barang tersebut sudah ada atau belum dengan memanggil fungsi is\_barang\_exists(). Jika kode barang ada yang berarti status bernilai true maka program akan menampilkan kode barang, nama barang, dan harga barang yang dicari. Jika status bernilai false maka barang yang dicari tidak ada atau tidak ditemukan.

function show() {

    echo -e "Barang yang tersedia..\n"

    for (( i=0;i<${#kode\_barang[@]};i++ ))

    do

        echo "Kode Barang   : ${kode\_barang[$i]}"

        echo "Nama Barang   : ${nama\_barang[$i]}"

        echo "Harga Barang  : ${harga\_barang[$i]}"

        echo ""

    done

}

Fungsi show() digunakan untuk menampilkan semua barang yang telah diinputkan oleh user sebelumnya. Fungsi ini diawali menggunakan looping untuk mengambil nilai kode barang, nama barang, dan harga barang yang dimulai dari indeks ke nol sampai panjang array kode\_barang.

function delete() {

    status=false

    read -p "Kode barang yang ingin dihapus : " kode\_barang\_in

    echo ""

    is\_barang\_exists

    if [[ $status == true ]]

    then

        unset kode\_barang[$index]

        unset nama\_barang[$index]

        unset harga\_barang[$index]

        kode\_barang=( "${kode\_barang[@]}" )

        nama\_barang=( "${nama\_barang[@]}" )

        harga\_barang=( "${harga\_barang[@]}" )

        echo -e "Barang telah dihapus..\n"

    else

        echo -e "Barang tidak ditemukan..\n"

    fi

}

Fungsi delete digunakan untuk menghapus barang dari array. Sama seperti fungsi fungsi sebelumnya fungsi ini akan mengecek apakah kode barang yang diinput oleh user untuk dihapus sudah ada atau belum dengan memanggil fungsi is\_barang\_exists(). Jika kode barang tersebut ada yang artinya status bernilai true maka kode barang, nama barang, dan kode barang pada indeks tersebut akan dihapus dengan perintah unset nama\_array[$i]. setelah itu akan dilakukan reindex array dengan perintah nama\_array=( “${nama\_array[@]}” ). Reindex ini digunakan untuk menggeser indeks array yang kosong (akibat ada data yang dihapus pada indeks ke-i) dengan indeks array yang berikutnya. Sebaliknya jika status bernilai false maka artinya barang yang dicari tidak ada.

function update() {

    status=false

    read -p "Kode barang yang ingin diperbarui : " kode\_barang\_in

    echo ""

    is\_barang\_exists

    if [[ $status == true ]]

    then

        read -p "Kode Barang  : " kode\_barang[$index]

        read -p "Nama Barang  : " nama\_barang[$index]

        read -p "Harga Barang : " harga\_barang[$index]

        echo -e "\nBarang berhasil diperbarui..\n"

    else

        echo -e "Barang tidak ditemukan..\n"

    fi

}

Fungsi update digunakan untuk meng-update data barang. Fungsi ini akan mengecek apakah kode barang yang dicari sudah ada atau tidak dengan fungsi is\_barang\_exists(). Jika ternyata kode barang ada atau status bernilai true maka akan dilakukan update kode barang, nama barang, dan harga barang pada indeks tersebut. Namun jika kode barang tidak ada maka akan menampilkan teks barang tidak ditemukan.

function print\_out() {

    status=false

    is\_folder\_exists

    read -p "Kode laporan : " kode\_laporan

    echo ""

    is\_laporan\_exists

    if [[ $status == true ]]

    then

        echo "Laporan Penjualan $kode\_laporan" >> ./laporan/$kode\_laporan.txt

        echo $date >> ./laporan/$kode\_laporan.txt

        echo "" >> ./laporan/$kode\_laporan.txt

        total=0

        for (( i=0;i<${#kode\_barang[@]};i++ ))

        do

            echo "Kode Barang   : ${kode\_barang[$i]}" >> ./laporan/$kode\_laporan.txt

            echo "Nama Barang   : ${nama\_barang[$i]}" >> ./laporan/$kode\_laporan.txt

            echo "Harga Barang  : ${harga\_barang[$i]}" >> ./laporan/$kode\_laporan.txt

            echo "" >> ./laporan/$kode\_laporan.txt

            let total+=${harga\_barang[$i]}

        done

        echo "Total Harga   : $total" >> ./laporan/$kode\_laporan.txt

        echo "" >> ./laporan/$kode\_laporan.txt

        cat ./laporan/$kode\_laporan.txt

        unset kode\_barang

        unset nama\_barang

        unset harga\_barang

    else

        echo -e "Laporan sudah ada..\n"

    fi

}

Fungsi print\_out() akan menghasilkan file.txt yang berupa laporan penjualan dengan nama kode\_laporan.txt di dalam folder laporan. Fungsi ini diawali dengan mengecek folder laporan sudah ada atau belum dengan fungsi is\_folder\_exists(). Jika folder laporan belum ada makan akan dibuat folder tersebut. Kemudian akan meminta input berupa kode laporan untuk nama file.txt-nya. Setelah itu mengecek apakah file.txt dengan nama kode laporan yang telah diinputkan sebelumnya sudah ada atau belum. Jika belum ada yang artinya status bernilai true maka akan memasukkan teks “Laporan penjualan $kode\_laporan” dan “$date” (tanggal hari saat itu) ke dalam file kode\_laporan.txt yang berada di dalam folder laporan. Lalu ada inisialisasi total dengan nol dan dilakukan looping sebanyak array kode\_barang untuk memasukkan data kode barang, nama barang, harga barang secara urut mulai indeks ke nol ke dalam file kode\_laporan.txt, jika loop sudah berakhir maka akan memasukkan teks “Total harga : $total” ke dalam file kode\_laporan.txt juga. Yang terakhir ialah menampilkan isi dari laporan yang bernama kode\_laporan.txt dengan perintah cat ./laporan/$kode\_laporan.txt dan menghapus array kode\_barang, nama\_barang, dan harga\_barang dengan perintah unset nama\_array.

function menu() {

    date=`date +"%b %d, %Y %I:%M %p"`

    echo ""

    echo "=============================="

    echo "| Merchant Management System |"

    echo "|   $date    |"

    echo "=============================="

    echo "| Menu :                     |"

    echo "| 1. Tambah Barang           |"

    echo "| 2. Cari Barang             |"

    echo "| 3. Lihat Barang            |"

    echo "| 4. Hapus Barang            |"

    echo "| 5. Perbarui Barang         |"

    echo "| 6. Cetak Laporan           |"

    echo "| 7. Keluar                  |"

    echo "=============================="

    echo ""

    read -p "Masukkan pilihan : " pilihan

    echo ""

    case "$pilihan" in

        1)

            insert

            read -p "Tekan tombol apa saja untuk melanjutkan.." hold

            clear; menu

        ;;

        2)

            search

            read -p "Tekan tombol apa saja untuk melanjutkan.." hold

            clear; menu

        ;;

        3)

            show

            read -p "Tekan tombol apa saja untuk melanjutkan.." hold

            clear; menu

        ;;

        4)

            delete

            read -p "Tekan tombol apa saja untuk melanjutkan.." hold

            clear; menu

        ;;

        5)

            update

            read -p "Tekan tombol apa saja untuk melanjutkan.." hold

            clear; menu

        ;;

        6)

            print\_out

            read -p "Tekan tombol apa saja untuk melanjutkan.." hold

            clear; menu

        ;;

        7)

            exit 0

        ;;

        \*)

            echo -e "Pilihan salah\n"

            read -p "Tekan tombol apa saja untuk melanjutkan.." hold

            clear; menu

        ;;

    esac

}

Fungsi menu() adalah fungsi terakhir yang dideklarasikan pada program merchant management system ini. Fungsi menu ini digunakan untuk menampilkan menu yang tersedia di dalam program, menu yang tersedia diantaranya :

1. Tambah barang

2. Cari barang

3. Lihat barang

4. Hapus barang

5. Perbarui barang

6. Cetak laporan

7. Keluar

Kemudian mengambil input dari user berupa no menu yang dipilih dan dilakukan pengecekan dengan switch case. Dengan begitu program akan menyeleksi no menu yang dipilih untuk dilakukan proses selanjutnya yaitu memanggil fungsi sesuai dengan yang tertera di source code di atas. Jika user memilih menu no 1, 2, 3, 4, 5, 6, dan 7 diakhir pemanggilan fungsi program akan selalu memanggil fungsi menu() lagi. Jika user memasukkan angka selain 1, 2, 3, 4, 5, 6, dan 7 maka akan masuk ke default yang tetap akan menampilkan teks “Pilihan salah” lalu memanggil fungsi menu() lagi untuk input menu. Jika user memasukkan angka 7 maka program akan berhenti atau keluar karena di dalam case 7 itu dideklarasikan perintah exit 0. Perintah clear digunakan untuk membersihkan output sebelumnya pada terminal sehingga saat memanggil fungsi menu() lagi tampilannya terlihat bersih atau clear.

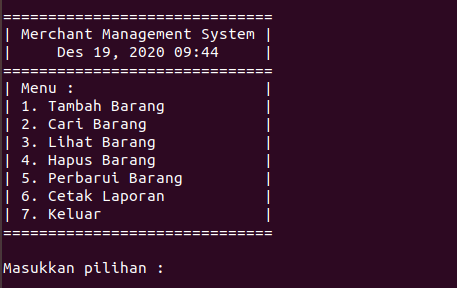
C. PEMANGGILAN FUNGI MENU

menu

Fungsi menu() dipanggil agar program menampilkan menu yang tersedia dari program merchant management system ini saat program ini pertama kali dijalankan sebenarnya pemanggilan fungsi menu ini mirip seperti fungsi main pada kebanyak program karena ini merupakan inti dari jalannya program ini.

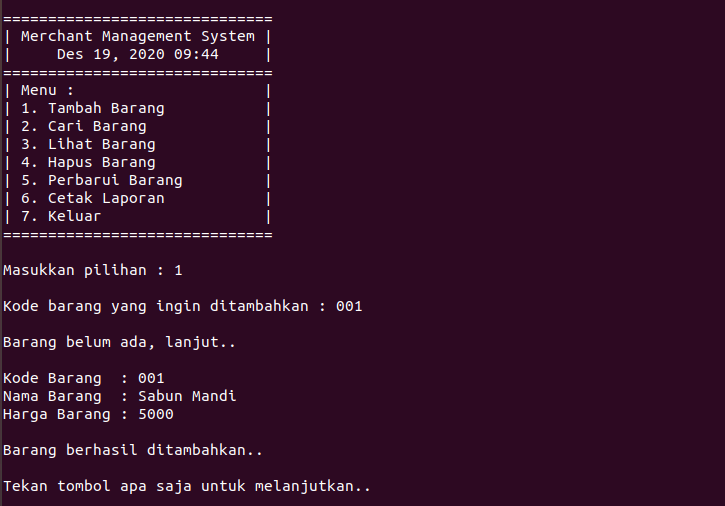
OUTPUT PROGRAM

Menajalankan program menggunakan perintah bash nama\_file.sh pada terminal Linux.

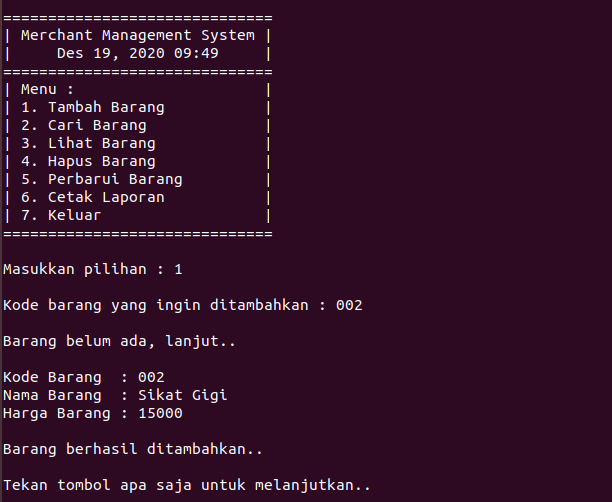


Gambar 1 : Tampilan menu utama

Menu Tambah Barang. Sebagai contoh, ditambahkan 3 barang yang berbeda.



Gambar 2 : Menambahkan barang ke-1

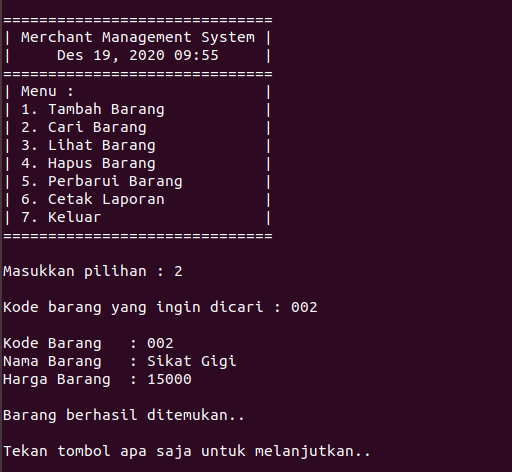


Gambar 3 : Menambahkan barang ke-2

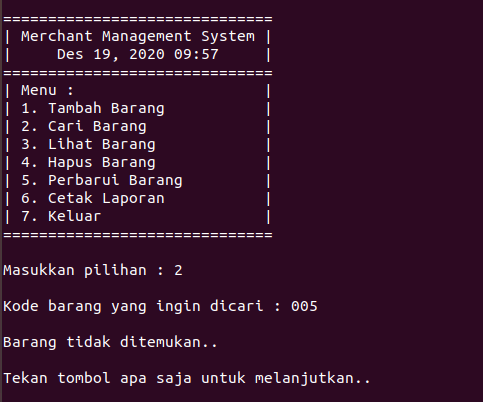


Gambar 4 : Menambahkan barang ke-3

Menu Cari Barang.



Gambar 5 : Mencari barang dengan kode barang 002 (Input benar atau kode tersebut tersedia)



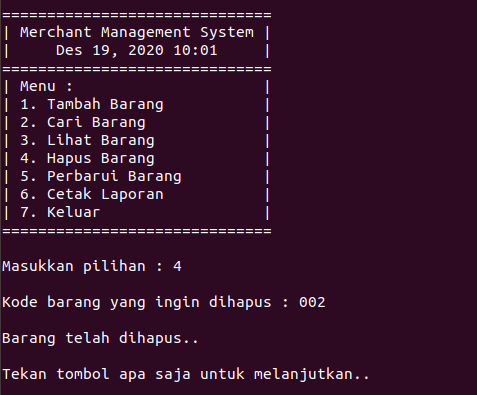
Gambar 6 : Mencari barang dengan kode barang 005 (Input salah atau kode tersebut tidak tersedia)

Menu Lihat Barang.



Gambar 7 : Menampilkan barang atau lihat barang yang telah ditambahkan sebelumnya

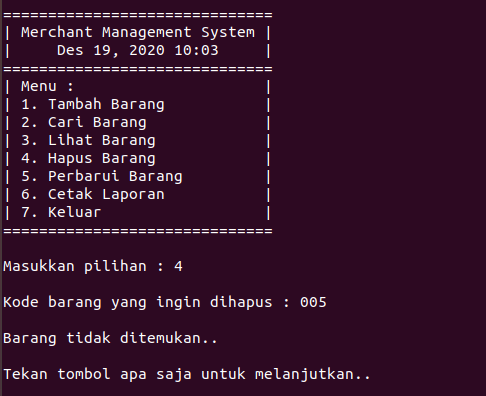
Menu Hapus Barang.



Gambar 8 : Menghapus barang dengan kode barang 002

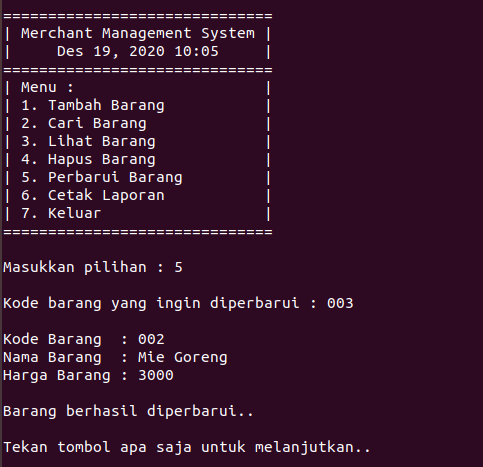


Gambar 9 : Melihat barang setelah dihapus



Gambar 10 : Menghapus barang dengan kode barang 005 (Input salah atau kode tersebut tidak tersedia)

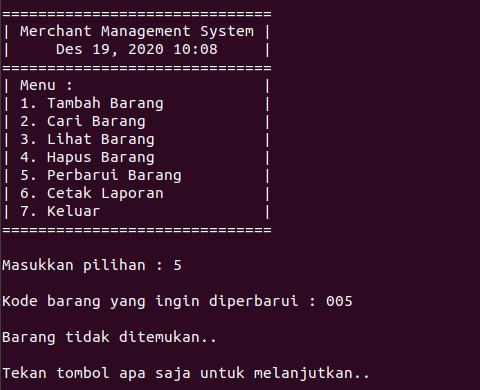
Menu Perbarui Barang.



Gambar 11 : Memperbarui barang dengan kode barang 003



Gambar 12 : Melihat barang setelah diperbarui

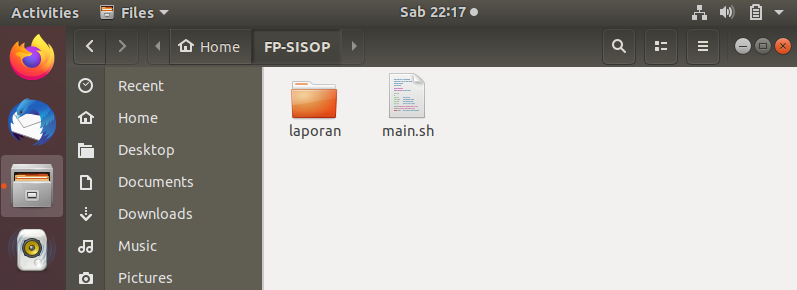


Gambar 13 : Memperbarui barang dengan kode barang 005 (Input salah atau kode tersebut tidak tersedia)

Menu Cetak Laporan.



Gambar 14 : Mencetak laporan dengan kode laporan 01



Gambar 15 : Dibuat folder baru yang bernama “laporan”

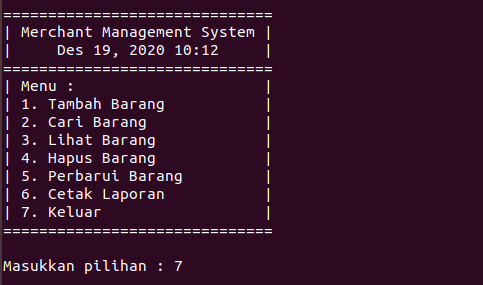


Gambar 16 : Folder “laporan” berisi file “01.txt”



Gambar 17 : Isi dari file yang bernama “01.txt”

Menu Keluar.



Gambar 18 : Keluar dari program