**Laporan Project Bengkel Pemrograman Shell**

**“Aplikasi Pendaftaran Vaksin”**



**Nama: Ahmad Miqdad**

**Kelas: 2 D4 Telekomunikasi A**

**NRP: 2220600008**

**DAFTAR ISI**

**Tujuan 3**

**Dasar Teori 3**

**Program 13**

**Tampilan Project 18**

**Analisa 22**

**Kesimpulan 24**

**Link Video 24**

**Daftar Pustaka 24**

1. **Tujuan**
2. Mahasiswa dapat mengaplikasikan semua modul yang sudah diajarkan dalam satu semester
3. Mahasiswa dapat membuat aplikasi berbasis shell script dengan GUI Zenity
4. Mahasiswa dapat menjalankan program yang telah dibuat
5. **Dasar Teori**

Dalam pembuatan program Aplikasi Pendaftaran Vaksin kita menggunakan perintah-perintah yang ada dalam bash shell script linux seperti, perintah dasar shell linux, pembuatan script, pembuatan dan penggunaan variable, pengkondisian (if dan case), perintah sed, Input/Output(I/O) ,penggunaan ekspresi, dan zenity untuk menampilkan GUI aplikasi.

1. **Eksperesi**

Ekspresi pada bahasa-bahasa tingkat tinggi terbagi atas dua macam yaitu ekspresi kondisi dan ekspresi aritmatika. Kedua macam ekspresi ini menggunakan operator.

**Operator Kondisi**

|| atau (OR)

&& dan (AND)

! tidak(NOT)

Konstanta kondisi ada dua: 0 untuk kondisi salah (FALSE) dan 1 untuk kondisi benar (TRUE). Sedangkan untuk operator relasi yang menghasilkan TRUE atau FALSE sebagai berikut:

**Operator Relasi**

== sama dengan

!= tidak sama dengan

=~ sama dengan untuk string

!~ tidak sama dengan untuk string

<= lebih kecil atau sama dengan

>= lebih besar atau sama dengan

> lebih besar

< lebih kecil

Operator yang berhubungan dengan aritmatika adalah:

Perkalian \*

Pembagian /

Penjumlahan +

Pengurangan –

Modulo (sisa hasil bagi) %

1. Perintah If dan case

* **If**

Statement builtin **if** berfungsi untuk melakukan seleksi berdasarkan suatu kondisi tertentu. Secara umum ada dua bentuk umum sintak perintah if , seperti ditunjukkan dibawah ini

**Sintak :**

**1.**

***if [ kondisi ] then***

*statements*

***fi***

**2.**

***if [ kondisi ] then***

*statements*

***else***

*statements*

***fi***

Perbedaan antara kedua bentuk adalah bentuk pertama mempunyai perintah tunggal jika ekspresi/kondisi yang diuji benar, sedangkan bentuk kedua mempunyai banyak perintah yang akan dijalankan jika ekspresi yang diuji benar.

Contoh bentuk pertama:

*let hasil = "$b \* $c" if [ "$hasil" = 10 ]*

*then*

*echo "Hasil perkalian kedua bilangan = $hasil"*

*fi*

Contoh bentuk kedua:

*let hasil = "$b \* $c" if [ "$hasil" = 10 ]*

*then*

*echo "Hasil perkalian kedua bilangan = $hasil" else*

*echo "selesai"*

*fi*

Kalau diperhatikan, perintah if ini hampir sama dengan perintah if pada bahasa-bahasa tingkat tinggi, seperti Pascal, C, dan juga hampir sama dengan perintah if pada batch file-nya DOS. Pada bentuk pertama maupun bentuk kedua dari sintak diatas adalah statement dalam blok if...fi akan dieksekusi apabila kondisi if terpenuhi. Dari kedua bentuk diatas dapat pula ditambahkan perintah untuk pengecekan kondisi dengan **elif** (else if), contoh sintaknya adalah sebagai berikut:

**3*.***

***if [ kondisi ]; then***

*perintah1;*

***elif [ kondisi2 ];***

***then***

*perintah2;*

***else***

*alternatif\_perintah;*

***fi***

klausa **else** akan dieksekusi jika **if** tidak terpenuhi, sebaliknya jika **if** terpenuhi maka **else** tidak akan dieksekusi.

Contoh bentuk ketiga:

*if winter*

*then*

*snowremoval*

*weatherstrip*

*elif spring*

*then*

*startgarden*

*mowlawn*

*else*

*echo “Something is wrong”*

*fi*

* **Case**

Case digunakan untuk menyederhanakan pemakaian if yang berantai, sehingga dengan case, kondisi dapat dikelompokkan secara logis dengan lebih jelas dan mudah untuk ditulis. Statement case juga digunakan untuk menyeleksi kondisi majemuk, dibanding if, pemakaian case lebih efisien.

**Sintak :**

***case*** *string* ***in***

*pilihan)*

*commands*

*;;*

*pilihan)*

*commands*

*;;*

*\*)*

*default commands*

*;;*

***esac***

Case diakhiri dengan esac dan pada setiap kelompok instruksi diakhiri dengan ;;. Pada akhir pilihan yaitu \*) yang berarti adalah “default”, bila kondisi tidak memenuhi pola sebelumnya. Contoh:

*let hasil = “$b \* $c”*

***case*** *$hasil* ***in***

*10)*

*echo “Hasil perkalian kedua bilangan = $hasil”*

*;;*

*\*)*

*echo “Selesai”*

***esac***

1. **Variabel**

Secara sederhana variabel adalah pengenal (identifier) berupa satuan dasarpenyimpanan yang isi atau nilainya sewaktu-waktu dapat berubah baik oleheksekusi program (runtime program) ataupun proses lain yang dilakukan sistem operasi. Variabel dibagi menjadi 3 kategori:

1. Environment Variable
2. Positional Parameter
3. User Define Variable

## Environment Variabel

Environment Variabel atau variabel lingkungan adalah variabel yang digunakan khusus oleh shell atau system linux untuk proses kerja system seperti variabel PS1, PS2, HOME, PATH, USER, SHELL, dan sebagainya. Jika variabel ini digunakan akan berdampak pada system, misalkan variabel PS1 yang digunakan untuk mengeset prompt shell pertama yaitu prompt tempat anda mengetikkan perintah - perintah shell (defaultnya "\s-\v\$"), PS2 untuk promptpelengkap perintah, prompt ini akan ditampilkan jika perintah yang dimasukkandianggap belum lengkap oleh shell (defaultnya ">"). Variabel PS1 dan PS2 dapat disetting seperti berikut.

*$PS1LAMA=$PS1*

data string yang diinginkan dimasukkan pada variabel PS1

*$PS1="Ini Promptku>"*

*Ini Promptku>PS1="Cobalah ? "*

maka prompt pertama dan kedua akan berubah, untuk mengembalikan PS1 keprompt semula perintahnya adalah :

*$PS1=$PS1LAMA*

Jika ingin mengkonfigurasi prompt shell, bash telah menyediakan beberapabackslash karakter diantaranya adalah:

|  |  |
| --- | --- |
| \a | ASCII bell character (07) |
| \d | date dengan format "Weekday Month Date" (misalnya "Tue  May 26") |
| \e | ASCII escape character (033) |
| \H | hostname (namahost) |
| \n | newline (karakter baru) |
| \w | Direktory aktif |
| \t | time dalam 24 jam dengan format HH:MM:SS |
| dll | man bash:-) |

contoh pemakaiannya:

*$PS1="[\t][\u@\h:\w]\$"*

agar prompt shell hasil konfigurasi anda dapat tetap berlaku (permanen)sisipkan pada file .bashrc atau .profile

* **Positional Parameter**

Positional Parameter / parameter posisi yaitu variabel yang digunakan shell untuk menampung argumen yang diberikan terhadap shell baik berupa argumen waktu sebuah file dijalankan atau argumen yang dikirim ke subrutin. variabel yang dimaksud adalah 1,2,3,dan seterusnya, lebih jelasnya lihat contohscript berikut :

*echo $1 adalah salah satu $2 populer di $3*

masing-masing argumen pada contoh diatas akan disimpan pada variabel 1,2,3sesuai posisinya. variabel spesial lain yang dapat digunakan yaitu 0, #, dan \* seperti diperlihatkan pada script berikut:

*echo "Nama script anda : $0";*

*echo "Banyak argumen : $#";*

*echo "Argumennya adalah: $\*";*

* **User Defined Variable**

User Defined Variable atau variabel yang didefinisikan sendiri oleh pembuatscript sesuai dengan kebutuhannya, beberapa hal yang perlu diperhatikan dalam mendefenisikan variabel adalah:

dimulai dengan huruf atau underscore

*VARIABLE\_NAME*=*VALUE*

Contoh :

*VARIABLE1=Value*

*echo “My variable holds: $VARIABLE1”*

hindari pemakaian spesial karakter seperti \*,$,#,dan lain-lain. bash bersifat case sensitive, maksudnya membedakan huruf besar dankecil, a berbeda dengan A, nama berbeda dengan Nama, NaMa, dan sebagainya.

Untuk mengeset nilai variabel dengan menggunakan operator assignment(pemberi nilai) "=", contohnya :

*myos="linux" #double-quoted nama='pinguin' #single-quoted*

*hasil=`ls -l`; #back-quoted*

*angka=12*

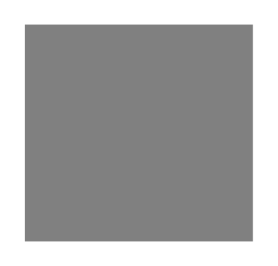
Jika diperhatikan ada 3 tanda kutip yang digunakan untuk memberikan nilai string ke suatu variabel, adapun perbedaannya adalah:

dengan kutip ganda (double-quoted), bash mengizinkan untuk menyisipkanvariabel di dalamnya.

dengan kutip tunggal (single-quoted), akan ditampilkan apa adanya. dengan kutip terbalik (back-quoted), bash menerjemahkan sebagai perintah yang akan dieksekusi.

1. **I/O**

Proses Input dan Output:



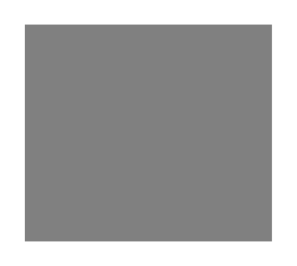
**I**

**n**

**p**

**u**

**t**



**P**

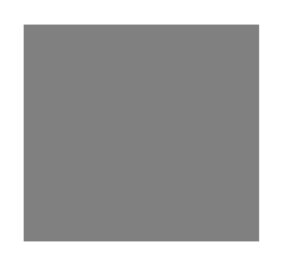
**r**

**o**

**s**

**e**

**s**



**O**

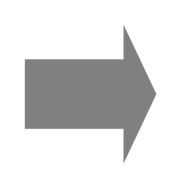
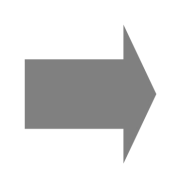
**u**

**t**

**p**

**u**

**t**



Instruksi (command) yang diberikan pada Linux melalui Shell disebut sebagai eksekusi program yang selanjutnya disebut proses. Setiap kali instruksi diberikan, maka Linux kernel akan menciptakan sebuah proses dengan memberikan nomor PID (Process Identity). Proses dalam Linux selalu membutuhkan Input dan menghasilkan suatu

Output.

Dalam konteks Linux input/output adalah :

* + Keyboard (Input)
  + Layar (Output)
  + Files
  + Struktur data kernel

**Simple I/O**

I/O merupakan hal yang mendasar dari kerja komputer karena kapasitas inilah yang membuat komputer begitu berdayaguna. I/O yang dimaksud adalah device yang menangani masukan dan keluaran, baik itu berupa keyboard, floppy, layar monitor,dan sebagainya. Penggunaan proses I/O ini pada contoh sebelumnya seperti statement echo yang digunakan untuk menampilkan teks atau informasi ke layar, atau operasi redirect ke file. selain echo, bash menyediakan perintah builtin printfuntuk mengalihkan keluaran ke output standard, baik ke layar ataupun ke file dengan format tertentu, mirip statement printf kepunyaan bahasaC atau perl. berikut contohnya:

**Output dengan echo**

Perintah yang digunakan adalah : echo.

Kalimat yang mengikuti echo akan ditampilkan di layar, kemudian kursor pindah baris berikutnya tanda (opsi) **–n** antara echo dengan kalimat digunakan agar kursor tidak pindah ke baris berikutnya, Contoh :

#!/bin/bash

#myinfo

#membersihkantampilanlayar clear

#menampilkaninformasi echo-n"Waktusystem :";date

echo-n"Anda: ";whoami

echo-n"Banyakpemakai:";who|wc–l

tentunya layout diatas akan disesuaikan dengan system yang digunakan statement echo dengan opsi -n akan membuat posisi kursor untuk tidak berpindah ke baris baru karena secara default statement echo akan mengakhiri proses pencetakan ke standar output dengan karakter baris baru (newline), dapat juga dicoba tanpa menggunakan opsi -n, dan lihat perbedaannya. opsi lain yang dapat digunakan adalah -e(enable), memungkinkan penggunaan backslash karakter atau karakter sekuen seperti pada bahasa C atau perl, misalkan :

**echo-e"\abunyikanbell"**

jika dijalankan akan mengeluarkan bunyi bell, informasi opsi pada statement echo dan backslash karakter selengkapnya dapat dilihat via man di prompt shell. **$manecho**

**Output dengan printf**

#!/bin/bash

#pr1

url="pemula.linux.or.id"; angka=32;

printf"Hi,PakeprintfalaC\n\t\adibash\n";

printf"Myurl%s\n%ddecimal=%ooctal\n"$url

$angka$angka;

printf"%ddecimaldalamfloat=%.2f\n"

$angka $angka

Untuk menggunakan format kontrol disertakan simbol %, bash akan mensubtitusikan format tsb dengan isi variabel yang berada di posisi kanan sesuai dengan urutannya jika lebih dari satu variabel, \n \t \a adalah karakter sekuen lepas newline,tab, dan bell, seperti terlihat pada tabel berikut:

|  |  |
| --- | --- |
| Format control | keterangan |
| %d | untuk format data integer |
| %o | octal |
| %f | float atau decimal |
| %x | Hexadecimal |

1. **Zenity**

Zenity mempunyai peran sebagai pembuat GUI pada shell programing atau yang disebut *Grafik User Interface* dimana mempunyai fungsi mempermudah user untuk mengoperasikan programnya melalui grafis secara interaktif. Zenity mempunyai banyak opsi seperti:

--entry : meminta input dari keyboard

--info : menampilkan text yang dimana berfungsi sebagai sebuah info

--list : membuat list berdasarkan column dan row secara tertabel dan lain-lain

Dengan opsi seperti diatas zenity dapat digunakan untuk membuat sebuah *question dialog box*. Disamping itu zenity juga dapat digunakan untuk aplikasi yang lain seperti *calendar, entry, error, info, file selection, list, notification, progress, warning, scale* dan *text info*. Pada bab ini akan di ilustrasikan bagaimana membuat aplikasi zenity dialog.

**Kalender Dialog**

Dengan menggunakan option –-calendar dapat dibuat sebuah kalender dialog. Pada kalender dialog ini user diijinkan untuk memilih inisial yang khusus pada perintah date yaitu dengan menggunakan option :

*--text=text*

Spesikasi “text” digunakan untuk menampilkan teks pada kalender dialog.

*--day=day*

Spesifikasi “day” digunakan untuk memilih tanggal di dalam kalender dialog. Day harus diberi nilai angka untuk tanggal dari 1 sampai 31.

*--month=month*

Spesifikasi “month” digunakan untuk memilih bulan di dalam kalender dialog. Month harus diberi nilai untuk bulan dari 1 sampai 12.

*--year=year*

Spesifikasi “year” digunakan untuk memilih tahun di dalam kalender dialog.

*--date-format=format*

Menentukan kembali format dari dialog kalender setelah seleksi tanggal (date). Format standar tergantung pada system lokal dan format yang dapat diterima oleh fungsi ***strftime*** misalnya %A %d/%m/%y.

**File Selection Dialog**

Untuk membuat *file selection dialog* digunakan opsi *--file-selection*. *Zenity* akan melakukan seleksi file atau direktori ke output standard. Mode default *file* *selection dialog* adalah buka file. *File selection dialog* memiliki beberapa opsi:

*--filename=*filename

Menentukan file atau direktori yang dipilih pada dialog pemilihan file ketika dialog yang pertama ditampilkan.

*--multiple*

Menentukan pemilihan beberapa nama file dalam dialog pemilihan file.

*--directory*

Menentukan pilihan direktori pada dialog pemilihan file.

*--save*

Set dialog pemilihan file ke mode save.

--separator=*separator*

Menentukan string yang digunakan untuk membagi kembali daftar nama file.

**List Dialog**

Menggunakan opsi --*list*  Untuk membuat *list dialog.* Zenity akan mengembalikan entry dalam kolom pertama baris teks yang dipilih ke output standar. Data untuk dialog kolom harus ditentukan menurut kolom, baris demi baris. Data dapat disediakan untuk dialog melalui input standar. Setiap entri harus dipisahkan oleh karakter baris baru. Jika menggunakan opsi --checklist atau --radiolist, setiap baris harus dimulai dengan ’TRUE’ atau ’FALSE’. List dialog memiliki opsi berikut:

*--column=*column

Menentukan header kolom yang ditampilkan dalam list dialog. Harus ditentukan opsi –colomn untuk setiap kolom yang ingin ditampilkan dalam dialog.

*--checklist*

Menentukan bahwa kolom pertama pada list dialog berisi kotak cek.

*--radiolist*

Menentukan bahwa kolom pertama pada list dialog berisi kotak radio.

*--editable*

Memungkinkan ditampilkan item yang akan diedit.

*--separator=*separator

Menentukan string apa yang digunakan ketika dialog mengembalikan daftar entry yang dipilih.

--print-column=*column*

Menentukan apakah kolom harus dicetak pada seleksi. Default kolom adalah '1 '. ‘ALL’ yang dapat digunakan untuk encetak semua kolom dalam list. Untuk melihat penjelasan lebih detail tentang penggunaan zenity, dapat dilihat di terminal dengan mengetik ***man zenity*,** maka akan muncul petunjuk penggunaan zenity.

1. **Program**

#!/bin/bash

menuawal()

{

ans=$(zenity --list --text "Pilih Menu" --column "Opsi" Daftar Cek Exit);

if [ -z $ans ];

then

peringatan;

fi

case $ans in

"Daftar") daftar;;

"Cek") cek;;

"Exit") keluar;;

"") menuawal;;

esac

}

daftar()

{

jawabv=$(zenity --list --text "Pilih Vaksin ke-" --radiolist --column "Pilih" --column "Vaksin" 1 Pertama 2 Kedua);

if [ "$?" == 1 ];

then

menuawal;

fi

if [ -z $jawabv ];

then

peringatan;

fi

case $jawabv in

"Pertama") tanggal;;

"Kedua") tanggal;;

"") daftar;;

esac

}

cek()

{

if ( test -z $lokasi1 )

then

peringatancek;

menuawal;

else

hasil;

kumpulandata;

fi

}

peringatancek()

{

(zenity --warning --title "Error" --text "Daftar Vaksin Dahulu")

}

keluar()

{

clear;

(zenity --info --title "Terimakasih!!" --text "Jangan Lupa Tanggal Vaksin ^\_^" --width 430)

exit 0;

}

keluar2()

{

clear;

exit 0;

}

peringatanlokasi()

{

zenity --warning --title "Error" --text "Pilih Puskesmas Terlebih Dahulu"

}

lokasi()

{

lokasi1=$(zenity --list --text "Pilih Lokasi" --width 300 --height 250 --column "Opsi" "Puskesmas Asemworo (Surabaya Barat)" "Puskesmas Benowo (Surabaya Barat)" "Puskesmas Balongsari (Surabaya Barat)" "Puskesmas Gading (Surabaya Timur)" "Puskesmas Kalijudan (Surabaya Timur)" "Puskesmas Tenggilis (Surabaya Timur)" "Puskesmas Kenjeran (Surabaya Utara)" "Puskesmas Dupak (Surabaya Utara)" "Puskesmas Sidotopo (Surabaya Utara)" "Puskesmas Jagir (Surabaya Selatan)" "Puskesmas Pakis (Surabaya Selatan)" "Puskesmas Jemursari (Surabaya Selatan)")

if [ "$?" == 1 ];

then

tanggal;

fi

if [ -z $lokasi1 ];

then

peringatanlokasi;

fi

case $lokasi1 in

"Puskesmas Asemworo (Surabaya Barat)") menuawal;;

"Puskesmas Benowo (Surabaya Barat)") menuawal;;

"Puskesmas Balongsari (Surabaya Barat)") menuawal;;

"Puskesmas Gading (Surabaya Timur)") menuawal;;

"Puskesmas Kalijudan (Surabaya Timur)") menuawal;;

"Puskesmas Tenggilis (Surabaya Timur)") menuawal;;

"Puskesmas Kenjeran (Surabaya Utara)") menuawal;;

"Puskesmas Dupak (Surabaya Utara)" ) menuawal;;

"Puskesmas Sidotopo (Surabaya Utara)") menuawal;;

"Puskesmas Jagir (Surabaya Selatan)") menuawal;;

"Puskesmas Pakis (Surabaya Selatan)") menuawal;;

"Puskesmas Jemursari (Surabaya Selatan)") menuawal;;

"") lokasi;;

esac

}

peringatantanggal()

{

zenity --warning --title "Error" --text "Pilih tanggal terlebih dahulu"

}

tanggal()

{

tanggal1=$(zenity --calendar --text "Pilih Tanggal Pendaftaran Vaksin" --title "Tanggal Pendaftaran Vaksin")

if [ "$?" == 1 ];

then

peringatantanggal;

tanggal;

fi

lokasi;

}

peringatan()

{

zenity --warning --title "Error" --text "Pilih menu terlebih dahulu"

}

hasil()

{

echo "Nama: $nama\nVaksin: $jawabv\nLokasi: $lokasi1 \nTanggal Pendaftaran: $tanggal1\n" >> data;

}

kumpulandata()

{

data=data

cetak1=$(sed -n /$nama/p $data)

kumpulandata1=$(zenity \

--text="$cetak1" \

--info \

--width=500 \

--height=200 \ )

menuawal;

}

clear

zenity --title "Pendaftaran Vaksin dan Cek Vaksin"

while [ -z $nama ];

do

nama=$(zenity --entry --text "Masukkan Nama Anda: " --title "Nama Untuk Daftar Vaksin" --width 500);

if [ "$?" == 1 ];then

(zenity --info --title "Keluar" --text "Terima kasih !!")

keluar2;

elif [ -z $nama ]; then

zenity --warning --title "Error" --text "Isi Nama Terlebih Dahulu"

fi

done

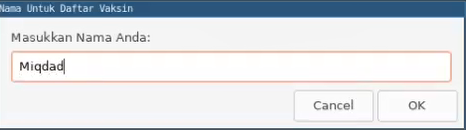
(zenity --info --title "Selamat Datang!" --text "Halo $nama Selamat Datang!" --width 430)

clear

menuawal;

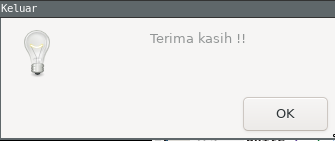
1. **Tampilan Project**

Gambar 1

****

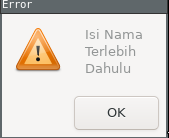
Tampilan Program Saat awal dijalankan, kita memasukkan nama yang akan didaftarkan ke program vaksin.

Gambar 2



Ketika user menekan tombol cancel maka akan muncul window keluar dan keluar dari program.

Gambar 3



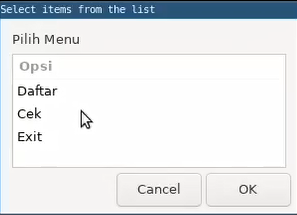
Peringatan Ketika user tidak mengisikan nama.

Gambar 4



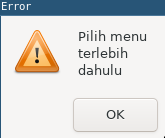
Tampilan program setelah mengisi nama, berisikan ucapan selamat dating.

Gambar 5



Tampilan Menu Utama.

Gambar 6



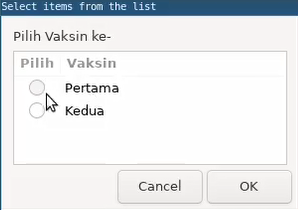
Tampilan peringatan Ketika Tidak memilih menu

Gambar 7



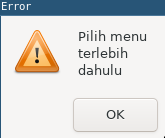
Tampilan Peringatan Ketika langsung memilih menu cek sebelum daftar vaksin

Gambar 8



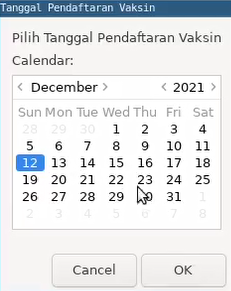
Tampilan Menu vaksin

Gambar 9



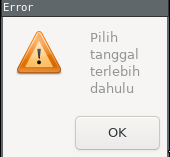
Tampilan Ketika tidak memilih

Gambar 10



Tampilan Tanggal

Gambar 11



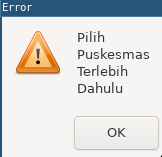
Tampilan Ketika tidak memilih tanggal

Gambar 12



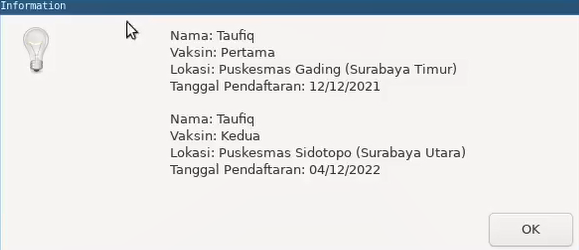
Tampilan Menu Lokasi Puskesmas

Gambar 13



Tampilan Ketika tidak memilih lokasi

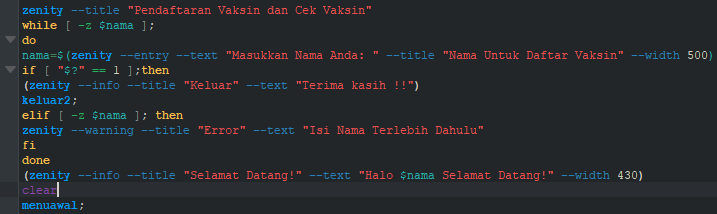
Gambar 14



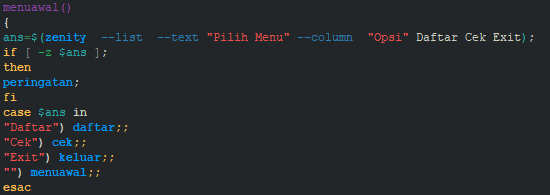
Tampilan Menu cek Ketika sudah mendaftar vaksin

1. **Analisa**

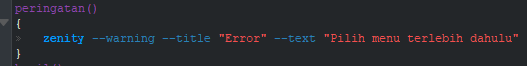
Pada saat awal program dijalankan muncul sebuah text box yang digunakan untuk mengisi nama untuk mendaftar vaksin. Ketika nama tidak di isi maka akan menampilkan error “Isi Nama Terlebih Dahulu” Ketika tombol cancel maka akan muncul info “Terimakasih”. Ketika nama di isi maka akan muncul window selamat datang $nama



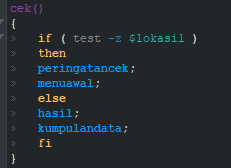
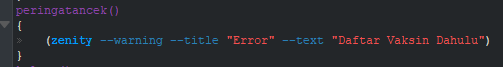
Kemudian setelah tampil window selamat datang maka akan muncul menu utama yang berisikan menu Daftar, Cek, dan Exit.



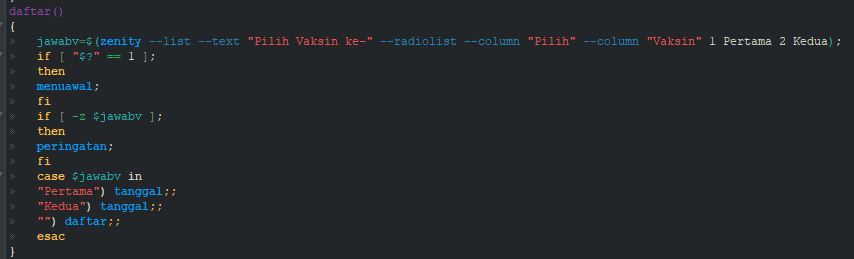
Ketika user tidak memilih menu maka akan muncul fungsi peringatan.



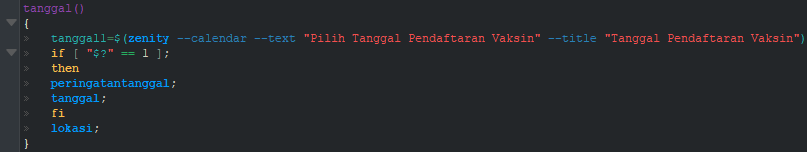
Ketika user memilih menu cek sebelum mendaftar maka akan memanggil fungsi cek yang kemudian diseleksi kondisi Ketika variable $tanggal1 kosong, maka akan memanggil fungsi peringatancek yang kemudian Kembali ke menu awal

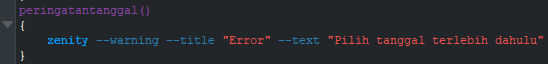
Kemudian Ketika user memilih menu Daftar maka akan memanggil fungsi daftar



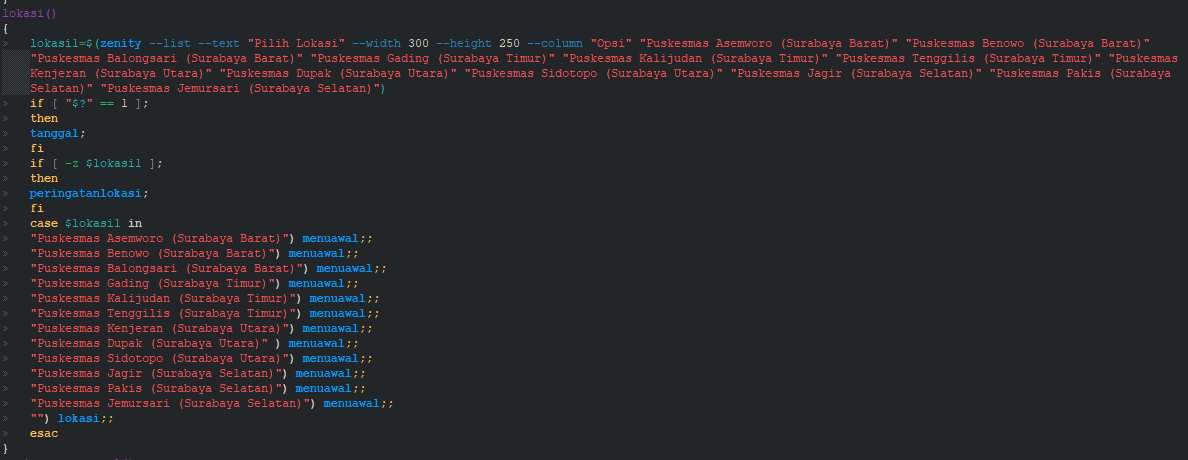
Ketika user tidak memilih menu maka akan muncul peringatan, jika user menekan tombol cancel maka akan dikembalikan ke menuawal. Untuk dapat melakukan beberapa kondisi di atas kita menggunakan perintah if dan case. Ketika user memilih menu pertama atau kedua maka akan dipanggil fungsi tanggal.



Ketika user tidak memilih tanggal maka akan muncul peringatan yang ada dalam fungsi peringatantanggal dan akan tetap dalam window tanggal



jika memilih tanggal maka akan muncul window lokasi dengan memanggil fungsi lokasi.



Ketika user memilih lokasi maka akan Kembali ke menu awal dengan memanggil fungsi menuawal.

Ketika user tidak memilih lokasi akan dipanggil fungsi peringatanlokasi dan akan tetap dalam window lokasi.



Ketika user memilih menu cek Kembali maka akan muncul window kumpulandata yang berisikan dari fungsi hasil. Dalam fungsi hasil data disimpan dalam file data.

1. **Kesimpulan**

Program aplikasi pendaftaran berfungsi untuk mendaftarkan diri untuk mendaftar vaksin pada lokasi dan tanggal tertentu. Setelah mendaftar vaksin pertama jeda antara vaksin pertama dan kedua adalah 2-3 minggu jadi untuk mengisi vaksin kedua kita memilih tanggal 2-3 minggu setelah minggu pertama. Dengan GUI dapat memudahkan user untuk berinteraksi dengan menu-menu yang tersedia.

1. **Link Video**

Berikut ini adalah link video dan file script project:

<https://drive.google.com/drive/folders/1nw5Vt5ijAqVJhUReOEm9QpA_25cFoRTb?usp=sharing>

1. **Daftar Pustaka**

* Modul Praktikum Bengkel Pemrograman Shell, Teknik Telekomunikasi PENS