**Aplikasi Pemetaan Pasar**



|  |
| --- |
| LAPORAN TUGAS BESAR  Diajukan untuk memenuhi tugas pada mata kuliah Sturuktur Data |



Disusun oleh:

1. Arief Budhiman (1301154360)
2. Firdausi Aufa Faradhila (1301154500)

|  |
| --- |
| PRODI S1 TEKNIK INFORMATIKA  FAKULTAS INFORMATIKA  UNIVERSITAS TELKOM  BANDUNG  2016 |

# DAFTAR ISI

[DAFTAR ISI i](#_Toc467621799)

[BAB 1 PENDAHULUAN 2](#_Toc467621800)

[1.1 LATAR BELAKANG 2](#_Toc467621801)

[1.2 RUMUSAN MASALAH 2](#_Toc467621802)

[1.3 TUJUAN 2](#_Toc467621803)

[1.4 PEMBAGIAN TUGAS 2](#_Toc467621804)

[BAB 2 PERANCANGAN 3](#_Toc467621805)

[2.1 FITUR APLIKASI 3](#_Toc467621806)

[2.2 FLOWCHART 3](#_Toc467621807)

[2.3 ADT DAN ALGORITMA 3](#_Toc467621808)

[BAB 3 IMPLEMENTASI DAN PENGUJIAN 4](#_Toc467621809)

[3.1 SCREENSHOT APLIKASI 4](#_Toc467621810)

[3.2 BLACK BOX TESTING 4](#_Toc467621811)

[BAB 4 KESIMPULAN DAN SARAN 5](#_Toc467621812)

# PENDAHULUAN

## LATAR BELAKANG

Pasar tidak dapat dipisahkan dari masyarakat. Dari zaman dahulu, pasar menjadi tempat transaksi jual beli, hampir semua kebutuhan ada disana, terutama kebutuhan pangan. Sampai saat ini, pasar masih menjadi tempat transaksi jual beli kebutuhan bagi masyarakat.

Namun, saat ini keadaan pasar dikalahkan oleh supermarket. Beberapa orang memilih untuk berbelanja di supermarket karena supermarket dianggap memiliki tempat yang lebih rapi, nyaman, dan bersih dibandingkan dengan pasar. Tetapi jika dari segi harga, supermarket memiliki harga yang sudah ditetapkan, sedangkan pasar memiliki harga yang lebih terjangkau dan masih bisa ditawar sehingga banyak pula masyarakat yang masih memilih untuk berbelanja di pasar.

Permasalahan lain yang dimiliki pasar adalah proses transaksinya yang masih manual, baik dari segi transaksi jual beli maupun pendataannya. Selain itu, pasar memiliki peletakan tempat yang tidak teratur dan tidak ditempatkan sesuai dengan jenis dagangannya.

Dari masalah diatas kami membuat program untuk menangani masalah tersebut. Antara lain untuk memudahkan pembeli mencari barang yang dibutuhkan. Maka, dalam tugas besar algoritma struktur data ini, kami membuat aplikasi pemetaan pasar.

## RUMUSAN MASALAH

Berdasarkan latar belakang diatas, program aplikasi pemetaan pasar memiliki rumusan masalah. Adapun rumusan masalahnya sebagai berikut :

1. Bagaimana cara mengimplementasikan multi linked list dalam program aplikasi pemetaan pasar?

## TUJUAN

Dari rumusan masalah diatas, tujuan dari aplikasi pemetaan pasar adalah sebagai berikut :

1. Untuk mengimplementasikan multi linked list dalam program aplikasi pemetaan pasar.

# PERANCANGAN

## FITUR APLIKASI

Fitur yang ada pada aplikasi pemetaan pasar adalah :

### Input Penjual

Fitur input penjual ini digunakan untuk menginputkan data penjual pada list penjual yang telah dibuat sebelumnya. Data yang diperlukan adalah nama, barang, dan alamat.

### Input Pembeli

Fitur input pembeli ini digunakan untuk menginputkan data pembeli pada list pembeli yang telah dibuat. Data yang diinputkan antara lain :

1. Nama untuk list pembeli.
2. Barang untuk list barang.
3. Alamat untuk list barang.

### Delete Penjual

Fitur delete penjual ini digunakan untuk menghapus data dari list penjual. Delete penjual ini terdiri dari 3 prosedur, yaitu :

1. Delete First, berfungsi untuk menghapus data yang terletak pada elemen pertama dari list penjual.
2. Delete Last, berfungsi untuk menghapus data yang terletak pada elemen terakhir dari list penjual.
3. Delete, berfungsi untuk menghapus data yang sesuai dengan data yang dicari.

### Delete Pembeli

Fitur delete pembeli ini digunakan untuk membatalkan semua pemesanan barang dari pembeli yang dimaksud.

### Delete Barang

Fitur delete barang ini digunakan untuk menghapus data dari list barang.

### Update Data Penjual

Fitur ini update data penjual ini digunakan untuk mengupdate data dari list penjual. Data yang akan diupdate antara lain :

1. Barang
2. Alamat

## PEMBAGIAN TUGAS

Arief Budhiman :

1. Hapus list penjual
2. Hapus list pembeli
3. Hapus list barang
4. Input list pembeli

Firdausi Aufa Faradhila :

1. Input list penjual
2. Tampilkan list penjual
3. Tampilkan list pembeli
4. Edit data penjual

## FLOWCHART

### Input Penjual



Input Penjual berfungsi untuk menginputkan elemen ke dalam list penjual.

Prosesnya, user menginputkan data dengan menggunakan insertlast.

### Input Pembeli



Input Pembeli berfungsi untuk menginputkan data pembeli pada list pembeli yang telah dibuat.

Prosesnya yaitu user menginputkan data pembeli, kemudian selama i<n, user menginputkan data barang yang akan dibeli.

### Delete Penjual



Delete Penjual berfungsi untuk menghapus data penjual yang sudah pernah diinputkan oleh user.

Prosesnya yaitu : Inputkan Alamat Penjual, jika data ditemukan, maka data yang sudah diinputkan sebelumnya akan diproses oleh program untuk dihapus dari list, jika data tidak ditemukan, maka program akan keluar dari menu Delete Penjual.

### Delete Pembeli



Delete Pembeli berfungsi untuk menghapus data pembeli yang sudah pernah diinputkan oleh user.

Prosesnya yaitu : Inputkan Nama Pembeli, jika data ditemukan, maka data yang sudah diinputkan sebelumnya akan diproses oleh program untuk dihapus dari list, jika data tidak ditemukan, maka program akan keluar dari menu Delete Pembeli.

### Delete Barang



Delete Barang berfungsi untuk menghapus data barang yang sudah pernah diinputkan oleh user.

Prosesnya yaitu : Inputkan Nama Pembeli dan Nama Barang, jika data ditemukan, maka data yang sudah diinputkan sebelumnya akan diproses oleh program untuk dihapus dari list, jika data tidak ditemukan, maka program akan keluar dari menu Delete Barang.

### Tampilkan Data Penjual



Tampilkan Data Penjual berfungsi untuk menampilkan data yang terdapat pada list penjual yang telah diinputkan sebelumnya.

Prosesnya, user mengetikkan angka yang terdapat pada menu utama, lalu data penjual akan ditampilkan.

### Tampilkan Data Pembeli



Tampilkan Data Pembeli berfungsi untuk menampilkan data yang terdapat pada list pembeli yang telah diinputkan sebelumnya.

Prosesnya, user mengetikkan angka yang terdapat pada menu utama, lalu data pembeli akan ditampilkan.

### Edit Data Penjual

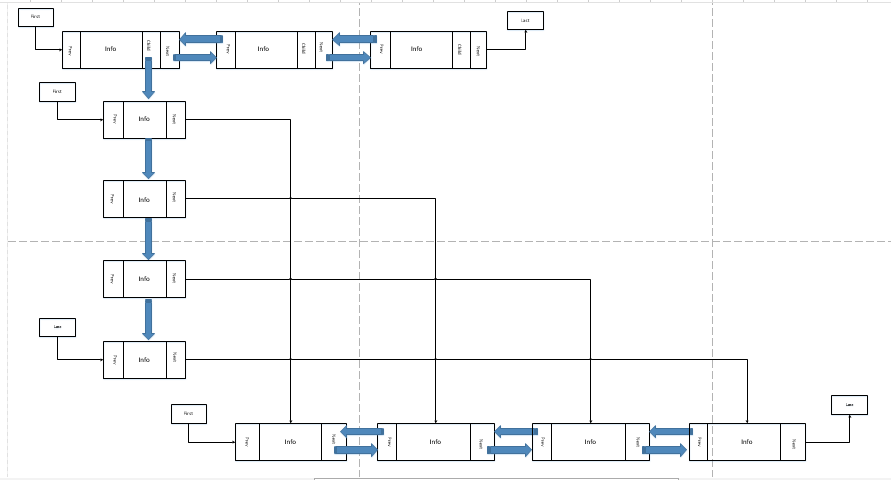


Edit Data Penjual berfungsi untuk mengubah data penjual pada list penjual yang telah dibuat.

Prosesnya yaitu :

1. User menginputkan data nama dan alamat penjual, kemudian jika hasilnya valid, maka user akan menginputkan data untuk merevisi barang dan alamat, namun jika tidak valid maka program akan kembali ke menu utama.
2. Setelah data berhasil di revisi, maka program akan melanjutkan proses untuk menginput data barang yang baru ke dalam list.

## BENTUK MULTI LINKED LIST



Jadi, dari implementasi di atas terdapat list, dimana list induknya merupakan list pembeli dan list anaknya merupakan list barang dimana list barang tersebut memiliki address yang nantinya akan diisi dan dihubungkan ke list penjual.

## ADT DAN ALGORITMA

### ADT Program Pemetaan Pasar

Type infopembeli <

nama : string

>

Type infoalamat <

barang : string

alamat : string

>

Type infopenjual <

nama : string

barang : string

alamat : string

hitung integer

>

Type addresspembeli : Pointer to pembelilist

Type addressalamat : Pointer to alamatbarang

Type addresspenjual : Pointer to penjuallist

Type pembelilist <

info : infopembeli

next : addresspembeli

prev : addresspembeli

anak : addressalamat

>

Type alamatbarang <

info : infoalamat

next : addressalamat

prev : addressalamat

>

Type penjuallist <

info : infopenjual

next : addresspenjual

prev : addresspenjual

>

Type listpembeli <

first : addresspembeli

last : addresspembeli

>

Type listbarang <

first : addressalamat

last : addressalamat

>

Type listpenjual <

first : addresspenjual

last : addresspenjual

>

Procedure createlistpembeli(input/output L : listpembeli)

Procedure createlistbarang(input/output L : listbarang)

Procedure createlistpenjual(input/output L : listpenjual)

### 2.5.1.1 PEMBELI

Function alokasipembeli(input x : infopembeli) : addresspembeli

Procedure dealokasipembeli(input/output L : addresspembeli)

Procedure insertlastpembeli(input/output L : listpembeli, input P : addresspembeli)

Procedure deletepembeli(input/output L : listpembeli, input/output P : addresspembeli)

Procedure deletefirstpembeli(input/output L : listpembeli)

Function findelemenpembeli(input L : listpembeli, input x : infopembeli) : addresspembeli

Procedure tampilkanpembeli(input/output L : listpembeli)

### 2.5.1.2 BARANG

Function alokasibarang(input x : infoalamat) : addressalamat

Procedure dealokasibarang(input/output L : addressalamat)

Procedure insertlastbarang(input/output Q : addresspembeli, input/output L : listbarang, input P : addressalamat)

Procedure deletebarang(input/output L : listbarang, input/output P : addressalamat)

Procedure deletefirstbarang(input/output L : listbarang)

Function findelemenbarang(input L : listbarang, input x : infoalamat) : addressalamat

### 2.5.1.3 PENJUAL

Function alokasipenjual(input x : infopenjual) : addresspenjual

Procedure dealokasipenjual(input/output L : addresspenjual)

Procedure insertlastpenjual(input/output L : listpenjual, input P : addresspenjual)

Procedure deletepenjual(input/output L : listpenjual, input/output P : addresspenjual)

Function findelemenpenjual(input L : listpenjual, input x : infopenjual) : addresspenjual

Procedure tampilkanpenjual(input/output L : listpenjual)

### 2.5.1.4 MENU

Procedure menu();

Procedure inputpenjual(input/output L : listpenjual)

Procedure inputpembeli(input/output LP : listpenjual, input/output L : listpembeli, input/output LB : listbarang)

Procedure hapuspenjual(input/output L : listpenjual)

Procedure hapuspembeli(input/output L : listpembeli, input/output LB : listbarang, input/output LPP : listpenjual)

Procedure hapusbarang(input L : listpembeli, input/output LB : listbarang)

Function cekpenjual(input/output L listpenjual, input Q : addresspenjual) : boolean

Procedure updatedataPenjual(input/output L : listpenjual)

### ALGORITMA PSEUDO CODE

#### Menu Input Penjual

Procedure inputpenjual(input/output L : listpenjual)

Kamus

x : infopenjual

P : addresspenjual

Algoritma

Output("Masukkan nama penjual = ")

Input(x.nama)

Output("masukkan nama barang = ")

Input(x.barang)

Output("masukkan alamat penjual = ")

Input(x.alamat)

P 🡨 alokasipenjual(x)

If (cekpenjual(L,P) = false) then

insertlastpenjual(L,P)

output("--data berhasil diinputkan--")

else

output("alamat sudah di tempati")

output("masukkan alamat lain")

info(P).hitung 🡨 0

#### Menu Input Pembeli

Procedure inputpembeli(input/output LP : listpenjual, input/output L : listpembeli, input/output LB : listbarang)

Kamus

x : infopembeli

bny, i : integer

y : infoalamat

Q : addressalamat

Algoritma

addresspenjual R 🡨 first(LP)

output("masukkan nama pembeli = ")

input(x.nama)

addresspembeli P 🡨 alokasipembeli(x)

insertlastpembeli(L,P)

system("cls")

output("masukkan banyak barang yang ingin dibeli = ")

input(bny)

for (i🡨1 to i<=bny) do

output("masukkan barang yang ingin dibeli = ")

input(y.barang)

while (R <> nil) do

if (y.barang = info(R).barang) then

y.alamat 🡨 info(R).alamat

info(R).hitung 🡨 info(R).hitung + 1

break

else

R 🡨 next(R)

Q 🡨 alokasibarang(y)

insertlastbarang(P,LB,Q)

R 🡨 first(LP)

Output("--data berhasil diinputkan--")

#### **Menu Hapus Penjual**

Procedure hapuspenjual(input/output L : listpenjual, input/output LPP : listbarang)

Kamus

x : infopenjual

Q : addressalamat

Algoritma

Output("========hapus penjual==========")

Output("masukkan alamat = ")

Cinput(x.alamat)

addresspenjual P 🡨 first(L)

while (P <> nil) do

if (info(P).alamat <> x.alamat) then

P 🡨 next(P)

else if (info(P).alamat = x.alamat) then

Q 🡨 first(LPP)

While (Q <> nil) do

If (info(Q).alamat = info(P).alamat) then

info(Q).alamat 🡨 "\0"

break;

else

Q 🡨 next(Q)

deletepenjual(L,P)

output("--data berhasil dihapus--")

break;

if (P = nil) then

output("--data tidak dapat dihapus--")

#### Menu Hapus Pembeli

Procedure hapuspembeli(input/output L : listpembeli, input/output LB : listbarang, input/output LPP : listpenjual)

Kamus

x : infopembeli

Algoritma

Output("==========hapus pembeli==========")

Output("masukkan nama pembeli = ")

Input(x.nama)

addresspembeli P 🡨 first(L)

while (P <> nil) do

if (info(P).nama <> x.nama) then

P 🡨 next(P)

else if (info(P).nama = x.nama) then

while (first(LB) <> nil) do

addressalamat R 🡨 first(LB)

addresspenjual Q 🡨 first(LPP)

while (Q <> nil)do

if (info(Q).alamat = info(R).alamat) then

info(Q).hitung 🡨 info(Q).hitung - 1

break

else

Q 🡨 next(Q)

deletebarang(LB,P,R)

deletepembeli(L,P)

output("--data berhasil dihapus--")

break

if (P = nil) then

output("--data tidak dapat dihapus--")

#### Menu Hapus Barang

Procedure hapusbarang(input/output L : listpembeli, input/output LB : listbarang, input/output LPP : listpenjual)

Kamus

x : infopembeli

y : infoalamat

Algoritma

addresspenjual R 🡨 first(LPP)

output("masukkan nama pembeli = ")

input(x.nama)

addresspembeli P 🡨 first(L)

while (P <> nil) do

if (info(P).nama = x.nama) then

break

else

P 🡨 next(P)

Output("masukkan nama barang yang ingin di hapus = ")

Input(y.barang)

addressalamat Q 🡨 first(LB)

while (Q <> nil) do

if (info(Q).barang = y.barang) then

while (R <> nil) do

if (info(R).alamat = info(Q).alamat) then

info(R).hitung 🡨 info(R).hitung -1

break

else

R 🡨 next(R)

break

else

Q 🡨 next(Q)

deletebarang(LB,P,Q)

output("--data berhasil dihapus--")

#### Menu Tampilkan Penjual

Procedure tampilkanpenjual(input/output L : listpenjual)

Kamus

Algoritma

addresspenjual P 🡨 first(L)

while (P <> nil) do

output("Nama Penjual = ",info(P).nama)

output("Barang yang dijual = ",info(P).barang)

output("Alamat = ", info(P).alamat)

output("hitung = ")

output(info(P).hitung)

P 🡨 next(P)

#### Menu Tampilkan Pembeli

Procedure tampilkanpembeli(input/output L : listpembeli)

Kamus

Algoritma

addresspembeli P 🡨 first(L)

while (P <> nil) do

addressalamat Q 🡨 anak(P)

while (Q <> nil) do

output("nama = ",info(P).nama)

output("barang = ",info(Q).barang)

output("alamat = ",info(Q).alamat)

Q 🡨 next(Q)

P 🡨 next(P)

#### Menu Edit Data Penjual

Procedure updatedataPenjual(input/output L : listpenjual)

Kamus

x : infopenjual

Algoritma

addresspenjual P 🡨 first(L)

output("masukkan alamat penjual yang ingin di update = ")

input(x.alamat)

output("masukkan nama penjual yang ingin di update = ")

input(x.nama)

while (P <> nil) do

if ((x.alamat = info(P).alamat) and (x.nama = info(P).nama)) then

output("Masukkan Data Baru")

output("Masukkan nama barang = ")

input(x.barang)

output("Masukkan alamat penjual = ")

input(x.alamat)

x.hitung 🡨 0

addresspenjual Q 🡨 alokasipenjual(x)

next(Q) 🡨 next(P)

prev(Q) 🡨 prev(P)

info(P).barang 🡨 info(Q).barang

info(P).alamat 🡨 info(Q).alamat

break

else

output("data tidak ditemukan")

P 🡨 next(P)

# IMPLEMENTASI DAN PENGUJIAN

## SCREENSHOT APLIKASI

|  |  |
| --- | --- |
| 1. INPUT PENJUAL | |
|  | User menginputkan angka 1 untuk mengakses fungsionalitas Input Penjual. |
|  | Kemudian user akan menginputkan beberapa data yang diperlukan, dalam hal ini user perlu menginputkan Nama Penjual, Nama Barang, dan Alamat Penjual pada tampilan yang tersedia. |
| 1. INPUT PEMBELI | |
|  | User menginputkan angka 2 untuk mengakses fungsionalitas Input Pembeli. |
|  | Kemudian user akan menginputkan data yang diperlukan, yaitu Nama Pembeli. |

|  |  |  |
| --- | --- | --- |
|  | Kemudian program akan menampilkan data berupa banyak barang yang ingin dibeli beserta jenis barangnya yang akan diinputkan oleh user. | |
| 1. HAPUS PENJUAL | | |
|  | User menginputkan angka 3 untuk mengakses fungsionalitas Hapus Penjual. | |
|  | Kemudian user akan menginputkan data yang diperlukan, yaitu Alamat Penjual, jika data ditemukan, maka data yang sudah diinputkan sebelumnya akan terhapus. | |
| 1. HAPUS PEMBELI | | |
|  | User menginputkan angka 4 untuk mengakses fungsionalitas Hapus Pembeli. | |
|  | Kemudian user akan menginputkan data yang diperlukan, yaitu Nama Pembeli, jika data ditemukan, maka data yang sudah diinputkan sebelumnya akan terhapus. | |
| 1. HAPUS BARANG | | |
|  | | User menginputkan angka 5 untuk mengakses fungsionalitas Hapus Barang. |
|  | | Kemudian user akan menginputkan data yang diperlukan, yaitu Nama Pembeli dan Nama Barang yang ingin dihapus. Dan selanjutnya data yang sudah diinputkan sebelumnya akan terhapus. |
| 1. TAMPILKAN PENJUAL | | |
|  | | User menginputkan angka 6 untuk mengakses fungsionalitas Tampilkan Penjual. |
|  | | Kemudian user akan menampilkan data yang sudah diinputkan sebelumnya, dan menghitung jumlah data yang diinputkan. |

|  |  |
| --- | --- |
| 1. TAMPILKAN PEMBELI | |
|  | User menginputkan angka 7 untuk mengakses fungsionalitas Tampilkan Pembeli. |
|  | Kemudian user akan menampilkan data yang sudah diinputkan sebelumnya, dan menghitung jumlah data yang diinputkan. |
| 1. EDIT DATA PENJUAL | |
|  | User menginputkan angka 8 untuk mengakses fungsionalitas Edit Data Penjual. |
|  | Kemudian user akan menginputkan data dengan mengisi Alamat Penjual dan Nama Penjual yang sudah diinputkan sebelumnya. Lalu, menginputkan kembali data Nama Barang yang akan diupdate beserta Alamat Penjual yang baru. Selanjutkan data akan diupdate. |

|  |  |
| --- | --- |
| 1. EXIT | |
|  | User menginputkan angka 9 untuk mengakses fungsionalitas Exit. |

## BLACK BOX TESTING

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Procedure | Deskripsi | Inputan | Keluaran yang Diharapkan | Kriteria Evaluasi Hasil | Kesimpulan |
| Input Penjual | Menginputkan penjual terbaru di akhir list. | Nama penjual, Barang yang dijual, alamat (kompleks) pasar | Muncul keluaran berupa “--data berhasil diinputkan--” dan data tampil saat mengakses procedure tampilkan penjual | Elemen masuk pada list penjual | Sesuai |
| Input Pembeli | Menginputkan pembeli terbaru di akhir list. | Nama pembeli, banyak barang yang ingin dibeli, nama barang yang ingin dibeli. | Muncul keluaran berupa “--data berhasil diinputkan--” dan data tampil saat mengakses procedure tampilkan pembeli | Elemen masuk pada list pembeli. | Sesuai |
| Hapus Penjual | Menghapus data penjual dan menghapus alamat yang terdapat pada list barang. | Alamat penjual | Muncul keluaran “--data berhasil dihapus--” dan data tidak muncul saat mengakses procedure tampilkan penjual. | Elemen terhapus dari list penjual | Sesuai |
| Hapus Pembeli | Menghapus data pembeli dan mengurangi counter pedagang yang akan dituju oleh pembeli tersebut. | Nama pembeli | Muncul keluaran “--data berhasil dihapus--” dan data tidak muncul saat mengakses procedure tampilkan pembeli. | Elemen terhapus dari list pembeli | Sesuai |
| Hapus Barang | Menghapus data barang pada pembeli yang ditentukan | Nama pembeli, nama barang yang akan dihapus. | Muncul keluaran “--data berhasil dihapus--” dan data tidak akan muncul saat mengakses procedure tampilkan pembeli. | Elemen terhapus dari list barang | Sesuai |
| Tampikan Penjual | Menampilkan penjual yang sedang berjualan di pasar. | - | Muncul data penjual saat mengakses procedure ini. | Seluruh elemen dari list penjual dikeluarkan. | Sesuai |
| Tampilkan Pembeli | Menampilkan pembeli yang sedang aktif berbelanja | - | Muncul data pembeli dan barang yang dibelinya saat mengakses procedure ini. | Seluruh elemen dari list pembeli dan list barang dikeluarkan | Sesuai |
| Edit Data Penjual | Mengubah isi dari suatu elemen dari list penjual | Alamat pedagang, nma pedagang. | Mengganti data yang sudah ada dengan data baru. | Elemen dari suatu list penjual akan diganti dengan elemen baru yang diinputkan | Sesuai |

# KESIMPULAN DAN SARAN

## Kesimpulan

## Berdasarkan program yang kami buat, kami dapat menyimpulkan bahwa program yang kami buat adalah program pemetaan pasar dengan menggunakan konsep multi linked list dan memiliki 8 fitur yang dapat memudahkan user untuk mencari barang yang dibutuhkan.

## Saran

## Saran yang dapat kami sampaikan adalah agar program bisa lebih berguna dimanfaatkan lagi oleh masyarakat dan batasan pada program yang ada lebih diperkecil lagi untuk meminimalisir kemungkinan error dalam programnya.