**PENDATAAN KELUARGA**

**DI KELURAHAN BOJONGSOANG**

**Laporan**

Diajukan untuk memenuhi tugas pada mata kuliah

Struktur Data

**Oleh:**

Armadhani Hiro Juni Permana (1301190234)

Windy Ramadhanti (1301194002)



**PROGRAM STUDI S-1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**UNIVERSITAS TELKOM**

**BANDUNG 2020**

**PENDATAAN KELUARGA**

**DI KELURAHAN BOJONGSOANG**

Armadhani Hiro Juni Permana#1,Windy Ramadhanti#2

[armadhanihiro@student.telkomuniversity.ac.id](mailto:armadhanihiro@student.telkomuniversity.ac.id) , [windyramadhanti@student.telkomunivrsity.ac.id](mailto:windyramadhanti@student.telkomunivrsity.ac.id)

BAB 1

1.Latar Belakang

Di kelurahan bojongsoang akan melakukan pendataan penduduk.pendataan ini berlaku bagi warga yang sudah memiliki kartu keluarga. Di berlakukannya pendataan tersebut agar dapat mempermudah penghitungan banyak penduduk berdasarkan data keluarga untuk pemilihan umum. Oleh karena itu, agar pendataan penduduk lebi efisien dan dokumen tidak menumpuk Maka, kelurahan bojongsoang akan memasukan pendataan keluarga di daerah tersebut ke suatu system. Sehingga sistem tersebut dapat merekam data penduduknya.jika ada perubahan pada data Kartu keluarga, maka data tersebut akan diperbaharui pada sistem.  
2. Rumusan masalah

* Bagaimana sistem tersebut merekam data-data penduduk kelurahan bojongsoang berdasarkan kartu keluarga?
* Fitur apa saja yang tersedia dalam sistem tersebut

3. Tujuan

* Meng-efisiensikan pendataan secara cepat
* Memudakan mencari suatu data Keluarga
* Memberikan informasi mengenai keluarga yang memiliki anak yang paling sedikit serta paling banyak

4. Pembagian Tugas  
5. Video presentasi (link video youtube)

BAB 2

1. Rancangan List

Ibu

u

ayah

u

Ibu

u

ayah

ibu

u

ayahh

ayah

ayah

ayah

ayah

anak

anak

anak

anak

1. Pada list berwarna biru adalah list parent, yang memuat data orang tua(ayah dan ibu) berisikan nama, tempat tanggal lahir, umur, golongan darah,serta status. List ini bermodelkan single linked list circular.
2. Pada list yang berwarna oren adalah list relasi, yang memuat pointer pointer yang akan menghubungkan relasi anak dengan orang tua .kotak berwarna biru berisikan pointer untuk ayah, hijau untuk ibu, kuning untuk anak. List ini bermodelkan single linked list.
3. Pada list berwarna hijau adalah list anak, yang memuat data anak yang berisikan nama, tempat tanggal lahir,umur, anak keberapa, dan golongan darah.list ini bermodelkan double linked list.
4. Fungsi dan prosedur

Fungsi/ prosedur list parent

|  |  |
| --- | --- |
| Fungsi / prosedur | keterangan |
| void createList(List\_parent &L);  I.S : list parent belum terbentuk  F.S : List parent sudah terbentuk | Prosedur ini berguna untuk meng-inisialisasikan dalam membuat list parent |
| void insertFirst(List\_parent &L, address\_parent P);  I.S : list mungkin kosong, P sudah dialokasikan  F.S : menempatkan elemen P sebagai elemen pertama pada list parent | Prosedur ini berguna untuk memasukan atau menambah elemen baru, sebagai elemen pertama pada list parent. Setiap akan menambah list parent, elemen yang akan di tambahkan akan bertambah sebanyak 2 elemen. 2 elemen ini di gunakan untuk data ayah dan data ibu |
| void insertAfter(List\_parent &L, address\_parent Prec, address\_parent P);  I.S : list mungkin kosong, P dan prec alamat salah satu elemen list  F.S : menempatkan elemen beralamatkan p sesudah elemen beralamat prec pada list parent | Prosedur ini berguna untuk memasukan atau menambah elemen baru, elemen baru akan di tambahkan setelah alamat prec pada list parent. Setiap akan menambah list parent, elemen yang akan di tambahkan akan bertambah sebanyak 2 elemen. 2 elemen ini di gunakan untuk data ayah dan data ibu |
| void insertLast(List\_parent &L, address\_parent P, address\_parent Q);  I.S : list mungkin kosong, P dan Q sudah dialokasikan  F.S : menempatkan elemen P sebagai elemen terakhir dan menempatkan Q pada next(P) pada list parent | Prosedur ini berguna untuk memasukan atau menambah elemen baru, sebagai elemen terakhir pada list parent. Setiap akan menambah list parent, elemen yang akan di tambahkan akan bertambah sebanyak 2 elemen. 2 elemen ini di gunakan untuk data ayah dan data ibu |
| void deleteFirst(List\_parent &L, address\_parent &P);  I.S : List tidak kosong  F.S : P adalah alamat elemen pertama list sebelum elemen pertama list di hapus, elemen pertama list hilang dan list mungkin menjadi kosong. | Prosedur ini berguna untuk menghapus data parent pada elemen pertama list parent. Karena saat proses insert parent akan bertambah 2 elemen , maka pada elemen parent pun akan menghapus 2 elemen . elemen ayah dan elemen ibu |
| void deleteLast(List\_parent &L, address\_parent &P);  I.S : list tidak kosong  F.S : : P adalah alamat elemen terakhir list sebelum elemen pertama list di hapus, elemen terakhir list hilang dan list mungkin menjadi kosong. | Prosedur ini berguna untuk menghapus data parent pada elemen terakhir list parent. Karena saat proses insert parent akan bertambah 2 elemen , maka pada elemen parent pun akan menghapus 2 elemen , elemen ayah dan elemen ibu. |
| Void deleteAfter(List\_parent &L, address\_parent Prec, address\_parent &P);  I.S : list tidak kosong , prec alamat salah satu elemen list  F.S: P adalah alamat dari next(prec) , menghapus next prec dari list . | Prosedur ini berguna untuk menghapus data parent pada elemen setelah prec, Karena saat proses insert parent akan bertambah 2 elemen , maka pada elemen parent pun akan menghapus 2 elemen . elemen ayah dan elemen ibu |
| address\_parent alokasiParent(string nama, string NIK, string peranan, string tempat, string tanggal, string agama, string endidikan, string pekerjaan);  / mengirimkan address dari alokasi sebuah elemen parent jika alokasi berhasil, dan nilai address tidak NULL dan jika gagal , maka sebaliknya. | Fungdi ini berguna untuk megisi info yang ada pada list .user akan diminta mengisi nama,nik,agama,Pendidikan, dan pekerjaan. |
| Address\_parent findElm(List\_parent L, string x, string y);  / mencari apakah ada elemen list parent yang yang berinfo x dan y jika ada, mengembalikan address elemen tersebut, NULL jika tidak ada . | Fungsi ini berguna untuk mencari elemen yang berada pada list parent. Pencarian dilakukan berdasarkan inputan user. x sebagai pencarian nama ayah dan y sebagai nama ibu. User akan diminta memasukan nama ayah dan ibu yang akan di cari |
| void printInfo(List\_parent L);  I.S : list mungkin kosong  FS : jika list tidak kosong , akan menampilkan semua info yang ada di list parent | Prosedur ini berguna untuk mencetak atau mengoutputkan info yang ada pada list parent |

Fungsi / prosedur list relasi

|  |  |
| --- | --- |
| Fungsi/ prosedur | keterangan |
| void createList(List\_relasi &L);  I.S : list relasi belum terbentuk  F.S : List relasi sudah terbentuk | Prosedur ini berguna untuk meng-inisialisasikan dalam membuat list relasi |
| void insertFirst(List\_relasi &L, address\_relasi P);  I.S : list mungkin kosong, P sudah dialokasikan  F.S : menempatkan elemen P sebagai elemen pertama pada list relasi | Prosedur ini berguna untuk memasukan atau menambah elemen baru, sebagai elemen pertama pada list relasi |
| void insertLast(List\_relasi &L, address\_relasi P);  I.S : list mungkin kosong, P sudah dialokasikan  F.S : menempatkan elemen P sebagai elemen terakhir pada list relasi | Prosedur ini berguna untuk memasukan atau menambah elemen baru, sebagai elemen terakhir pada list parent |
| void insertAfter(address\_relasi Prec, address\_relasi P);  I.S : list mungkin kosong, P dan prec alamat salah satu elemen list  F.S : menempatkan elemen beralamatkan p sesudah elemen beralamat prec pada list relasi | Prosedur ini berguna untuk memasukan atau menambah elemen baru, elemen baru akan di tambahkan setelah alamat prec pada list relasi |
| void deleteFirst(List\_relasi &L, address\_relasi &P);  I.S : List tidak kosong  F.S : P adalah alamat elemen pertama list sebelum elemen pertama list di hapus, elemen pertama list hilang dan list mungkin menjadi kosong. | Prosedur ini berguna untuk menghapus data relasi pada elemen pertama list relasi |
| void deleteLast(List\_parent &L, address\_parent &P);  I.S : list tidak kosong  F.S : : P adalah alamat elemen terakhir list sebelum elemen pertama list di hapus, elemen terakhir list hilang dan list mungkin menjadi kosong. | Prosedur ini berguna untuk menghapus data relasi pada elemen terakhir list relasi |
| void deleteAfter(List\_relasi &L, address\_relasi Prec, address\_relasi &P);  Prec, address\_parent &P);  I.S : list tidak kosong , prec alamat salah satu elemen list  F.S: P adalah alamat dari next(prec) , menghapus next prec dari list . |  |
| address\_relasi alokasi(address\_parent P, address\_child C);  / mengirimkan address dari alokasi sebuah elemen relasi jika alokasi berhasil, dan nilai address tidak NULL dan jika gagal , maka sebaliknya. |  |
| void printAllInfo(List\_relasi L);  I.S : list mungkin kosong  FS : jika list tidak kosong , akan menampilkan semua info yang ada di list relasi |  |
| void printInfoAnak(List\_relasi L, string ayah, string ibu);  I.S : list mungkin kosong  FS : jika list tidak kosong , akan menampilkan relasi anak di list relasi |  |
| void printInfoInduk(List\_relasi L, string x);  I.S : list mungkin kosong  FS : jika list tidak kosong , akan menampilkan relasi orang tua di list relasi |  |
| void printIndukMinMax(List\_relasi L);  I.S : list mungkin kosong  FS : jika list tidak kosong , akan menampilkan jumlah relasi terbesar atau terkecil pada list relasi |  |
| void deleteAnak(List\_parent K, List\_relasi L, List\_child M, string anak); |  |

Fungsi/ prosedur list child

|  |  |
| --- | --- |
|  |  |
| void createList(List\_child &L); | Prosedur ini berguna untuk meng-inisialisasikan dalam membuat list anak |
| void insertFirst(List\_child &L, address\_child P); | Prosedur ini berguna untuk memasukan atau menambah elemen baru, sebagai elemen pertama pada list anak |
| void insertLast(List\_child &L, address\_child P); | Prosedur ini berguna untuk memasukan atau menambah elemen baru, sebagai elemen terakhir pada list anak |
| void insertAfter(address\_child Prec, address\_child P); | Prosedur ini berguna untuk memasukan atau menambah elemen baru, elemen baru akan di tambahkan setelah alamat prec pada list anak |
| void deleteFirst(List\_child &L, address\_child &P); | Prosedur ini berguna untuk menghapus data anak pada elemen pertama list anak |
| void deleteLast(List\_child &L, address\_child &P); |  |
| void deleteAfter(List\_child &L, address\_child Prec, address\_child &P); |  |
| address\_child alokasiChild(string nama, string NIK, string peranan, string tempat, string tanggal, string agama, string pendidikan, string pekerjaan); |  |
| address\_child findElm(List\_child L, string anak); |  |
| void printInfo(List\_child L); |  |

3 .Pengujian

Bab 3  
kesimpulan dan saran