

```

1  #include "brain.h"
2
3  address_baris AdrBaris(infopasien x){
4      address_baris P;
5      P=new ebaris;
6      info(P).nama=x.nama;
7      info(P).sakit=x.sakit;
8      col(P)=NULL;
9      next(P)=NULL;
10     return P;
11 };
12
13 address_kolom AdrKolom(infoobat x){
14     address_kolom P;
15     P=new ekolom;
16     info(P).obat=x.obat;
17     info(P).jenis=x.jenis;
18     next(P)=NULL;
19     return P;
20 }
21
22 bool sudahPunya(List L, string obat, address_baris P){
23     address_kolom Q;
24     if(P!=NULL){
25         Q=col(P);
26         while(Q!=NULL && info(Q).obat!=obat){
27             Q=next(Q);
28         }
29     };
30     return (Q!=NULL && info(Q).obat==obat);
31 };
32
33 void createlist(List &L){
34     first(L)=NULL;
35 };
36 void createlistobat(listobat &L){
37     first(L)=NULL;
38 };
39 void printObat(listobat L, string x){
40     address_kolom P;
41     if((first(L)!=NULL) && x!="NULL"){
42         cout<<"Daftar Obat"<<endl;
43         P=first(L);
44         while(P!=NULL){
45             if(info(P).jenis==x){
46                 cout<<"Nama obat = "<<info(P).obat<<" || Jenis obat =
47                 "<<info(P).jenis<<endl;
48             };
49             P=next(P);
50         };
51         cout<<"======"<<endl;
52     }else if(first(L)!=NULL && x=="NULL"){
53         cout<<"Daftar Obat"<<endl;
54         P=first(L);
55         while(P!=NULL){
56             cout<<"Nama obat = "<<info(P).obat<<" || Jenis obat = "<<info(P).jenis<<endl;
57             P=next(P);
58         };
59     }else{
60         cout<<"Kosong"<<endl;
61     }
62 }
63 void DaftarObat(listobat &L){
64     address_kolom P;
65     infoobat x;
66     if(first(L)==NULL){
67         //data data
68         x.jenis="pusing";
69         x.obat="paracetamol";
70         P=AdrKolom(x);
71         first(L)=P;
72         x.jenis="pusing";
73         x.obat="mortalin";
74         P=AdrKolom(x);
75         next(P)=first(L);
76         first(L)=P;
77         x.jenis="pusing";
78         x.obat="naproxen";
79         P=AdrKolom(x);
80         next(P)=first(L);
81         first(L)=P;
82         x.jenis="pusing";
83         x.obat="aspirin";
84         P=AdrKolom(x);

```

```

84     next(P)=first(L);
85     first(L)=P;
86     x.jenis="flu";
87     x.obat="procolid";
88     P=AdrKolom(x);
89     next(P)=first(L);
90     first(L)=P;
91     x.jenis="flu";
92     x.obat="obh";
93     P=AdrKolom(x);
94     next(P)=first(L);
95     first(L)=P;
96     x.jenis="flu";
97     x.obat="hodrex";
98     P=AdrKolom(x);
99     next(P)=first(L);
100    first(L)=P;
101    x.jenis="flu";
102    x.obat="sanmol";
103    P=AdrKolom(x);
104    next(P)=first(L);
105    first(L)=P;
106    }else{
107        cout<<"Masukan obat yang mau ditambahkan"<<endl;
108        cout<<"Jenis obat = ";
109        cin>>x.jenis;
110        cout<<"Nama obat = ";
111        cin>>x.obat;
112        P=findobat(L,x.obat);
113        if(P==NULL){
114            P=AdrKolom(x);
115            next(P)=first(L);
116            first(L)=P;
117            cout<<"Berhasil ditambahkan!"<<endl;
118        }else{
119            cout<<"Obat dengan nama yang sama sudah terdaftar"<<endl;
120        }
121    }
122    cout<<"menampilkan list obat ? <y/n>"<<endl;
123    char y;
124    cin>>y;
125    if(y=='y'){
126        printObat(L,"NULL");
127    };
128 }
129 address_kolom findJenisObat(listobat L,string x){
130     address_kolom P;
131     P=first(L);
132     while((info(P).jenis!=x)&&(next(P)!=NULL)){
133         P=next(P);
134     }
135     if(info(P).jenis==x){
136         return P;
137     }else{
138         return NULL;
139     }
140 }
141 };
142
143 address_kolom findobat(listobat L,string x){
144     address_kolom P;
145     P=first(L);
146     while((info(P).obat!=x)&&(next(P)!=NULL)){
147         P=next(P);
148     };
149     if(info(P).obat==x){
150         return P;
151     }else{
152         return NULL;
153     }
154 };
155 address_baris findpasien(List L,string x){
156     address_baris P;
157     if(first(L)!=NULL){
158         P=first(L);
159         while((info(P).nama!=x)&&(next(P)!=NULL)){
160             P=next(P);
161         };
162         if(info(P).nama==x){
163             return P;
164         }else{
165             return NULL;
166         };
167     }else{

```

```

168         return NULL;
169     }
170 };
171
172 void printPenyakit(listobat L){
173     address_kolom P;
174     string x;
175     P=first(L);
176     x=info(P).jenis;
177     int i=1;
178     cout<<"Obat tersedia untuk jenis penyakit"<<endl;
179     cout<<i<<" " <<info(P).jenis<<endl;
180     while (P!=NULL){
181         if (info(P).jenis!=x){
182             i++;
183             cout<<i<<" " <<info(P).jenis<<endl;
184             x=info(P).jenis;
185         };
186         P=next(P);
187     }
188 };
189
190 //-----
191
192 void addFirstBaris(infopasien pasien, List &L){
193     address_baris P;
194     P=AdrBaris(pasien);
195     if (first(L)==NULL){
196         first(L)=P;
197         next(first(L))=NULL;
198     }else{
199         next(P)=first(L);
200         first(L)=P;
201     };
202 };
203
204 void addFirstKolom(infoobat obat, List &L){
205     address_kolom Q;
206     Q=new ekolom;
207     info(Q).obat=obat.obat;
208     info(Q).jenis=obat.jenis;
209     next(Q)=NULL;
210     if (first(L)->col ==NULL){
211         first(L)->col=Q;
212     }else{
213         next(Q)=first(L)->col;
214         first(L)->col=Q;
215     }
216 };
217
218 void printPasienObat(List L){
219     if (first(L)!=NULL){
220         ebaris *P;
221         P=first(L);
222         int i=1;
223         while (P!=NULL){
224             cout<<"===== "<<endl;
225             cout<<i<<" Nama = " <<info(P).nama<<" || Sakit = " <<info(P).sakit<<endl;
226             cout<<" Jenis Obat "<<endl;
227             //ekolom *Q=col(P);
228             address_kolom Q=col(P);
229             while (Q!=NULL){
230                 cout<<" -" <<info(Q).obat<<" Jenis = " <<info(Q).jenis<<endl;
231                 //cout<<" -" <<info(next(Q)).obat<<" Jenis = " <<info(next(Q)).obat<<endl;
232                 Q=next(Q);
233             };
234             Q=NULL;
235             i++;
236             P=next(P);
237         };
238     }else{
239         cout<<"TIDAK ADA"<<endl;
240     };
241 };
242
243 void addTKolom(string pasien, address_kolom Q, List &L){
244     ebaris *P;
245     ekolom *Z;
246     Z=new ekolom;
247     info(Z).obat=info(Q).obat;
248     info(Z).jenis=info(Q).jenis;
249     P=findpasien(L, pasien);
250     next(Z)=P->col;
251     P->col=Z;

```

```

252     Z=NULL;
253 };
254
255 void DelFirst(List &L) {
256     if(first(L) != NULL) {
257         address_baris P=first(L);
258         first(L)=next(first(L));
259         next(P)=NULL;
260         delete P;
261     } else {
262         cout<<"LIST KOSONG"<<endl;
263     }
264 };
265
266 void DelLast(List &L) {
267     if(first(L) != NULL) {
268         address_baris P=first(L);
269         while(next(next(P)) != NULL) {
270             P=next(P);
271         };
272         address_baris Q=P;
273         Q=next(P);
274         next(P)=NULL;
275         delete Q;
276     } else {
277         cout<<"LIST KOSONG"<<endl;
278     }
279 };
280
281 void DelTengah(List &L, string x) {
282     if(first(L) != NULL) {
283         address_baris P=first(L);
284         while((P != NULL) && (info(next(P)).nama != x)) {
285             P=next(P);
286         };
287         address_baris Q=next(P);
288         next(P)=next(Q);
289         next(Q)=NULL;
290         delete Q;
291     } else {
292         cout<<"LIST KOSONG"<<endl;
293     }
294 };
295
296 void CekPosisiDel(List &L, address_baris P) {
297     if(P != NULL) {
298         string x=info(P).nama;
299         if(P != first(L) && (next(P) != NULL)) {
300             //data berada ditengah
301             DelTengah(L, x);
302         } else if(P == first(L)) {
303             //data berada di awal;
304             DelFirst(L);
305         } else if(next(P) == NULL) {
306             //data berada di akhir
307             DelLast(L);
308         }
309     } else {
310         cout<<"DATA TIDAK DITEMUKAN"<<endl;
311     }
312 };
313
314 void ShowSatuPasien(List L, address_baris P) {
315     if(first(L) != NULL) {
316         if(P != NULL) {
317             address_kolom Q;
318             Q=col(P);
319             cout<<"Nama Pasien = "<<info(P).nama<<endl;
320             cout<<"Daftar Obat"<<endl;
321             int i=1;
322             while(Q != NULL) {
323                 cout<<i<<" . "<<info(Q).obat<<" "<<info(Q).jenis<<endl;
324                 Q=next(Q);
325                 i++;
326             };
327         } else {
328             cout<<"Nama Pasien Tidak terdaftar"<<endl;
329         }
330     } else {
331         cout<<"LIST KOSONG"<<endl;
332     }
333 };
334
335

```

```

336 void coutObatBanyak(List L, listobat O) {
337     address_baris P=first(L);
338     address_kolom Q;
339     const int NMAX=100;
340     struct TbL{
341         int akhir;
342         string obat[NMAX];
343         int jumlah[NMAX];
344     };
345
346     TbL Tabel;
347     Tabel.akhir=-1;
348
349     //memasukan obat yang digunakan pasien kedalam array
350     while (P!=NULL) {
351         Q=col(P);
352         while (Q!=NULL) {
353             Tabel.akhir++;
354             Tabel.obat[Tabel.akhir]=info(Q).obat;
355             Tabel.jumlah[Tabel.akhir]=0;
356             Q=next(Q);
357         };
358         P=next(P);
359     };
360
361     //menghitung jumlah penggunaan obat tertentu di array
362     int cek=0;
363     while (cek<=Tabel.akhir) {
364         int pos=cek;
365         while (pos<=Tabel.akhir) {
366             if (Tabel.obat[cek]==Tabel.obat[pos]) {
367                 Tabel.jumlah[cek]++;
368             };
369             pos++;
370         };
371         cek++;
372     };
373
374
375     //mencari indeks dengan jumlah terbanyak
376     int show=0;
377     int bandingIndeks=0;
378     int indx=0;
379     while (show<=Tabel.akhir) {
380         if (bandingIndeks<Tabel.jumlah[show]) {
381             bandingIndeks=Tabel.jumlah[show];
382             indx=show;
383         };
384         show++;
385     };
386
387     cout<<"======"<<endl;
388     cout<<"Obat yang populer digunakan Pasien"<<endl;
389     cout<<"Obat = "<<Tabel.obat[indx]<<endl;
390     cout<<"Total pengguna = "<<Tabel.jumlah[indx]<<endl;
391     cout<<"======"<<endl;
392 };
393
394 void printObat3(List L) {
395     address_baris P=first(L);
396     address_kolom Q;
397     address_kolom Z;
398     bool ada=false;
399     int jumlah=0;
400     while (P!=NULL) {
401         Q=col(P);
402         jumlah=0;
403         while (Q!=NULL) {
404             jumlah++;
405             Q=next(Q);
406         };
407         if (jumlah>=3) {
408             ada=true;
409             cout<<"======"<<endl;
410             cout<<"Nama pasien = "<<info(P).nama<<endl;
411             cout<<"Obat yang Digunakan = ";
412             Z=col(P);
413             while (Z!=NULL) {
414                 cout<<" "<<info(Z).obat;
415                 Z=next(Z);
416             };
417         };
418         P=next(P);
419     };

```

```
420     if(!ada){
421         cout<<"Tidak ada pasien yang menggunakan obat lebih dari 3"<<endl;
422     }
423 };
424
```