```cpp
#include "brain.h"

address_baris AdrBaris(infopasien x){
    address_baris P;
    P=new ebaris;
    info(P).nama=x.nama;
    info(P).sakit=x.sakit;
    col(P)=NULL;
    next(P)=NULL;
    return P;
};

address_kolom AdrKolom(infoobat x){
    address_kolom P;
    P=new ekolom;
    info(P).obat=x.obat;
    info(P).jenis=x.jenis;
    next(P)=NULL;
    return P;
}

bool sudahPunya(List L,string obat,address_baris P){
    address_kolom Q;
    if(P!=NULL){
        Q=col(P);
        while(Q!=NULL && info(Q).obat!=obat){
            Q=next(Q);
        }
    };
    return (Q!=NULL && info(Q).obat==obat);
};

void createlist(List &L){
    first(L)=NULL;
};
void createlistobat(listobat &L){
    first(L)=NULL;
}
void printObat(listobat L,string x){
    address_kolom P;
    if((first(L)!=NULL)&& x!="NULL"){
        cout<<"Daftar Obat"<<endl;
        P=first(L);
        while(P!=NULL){
            if(info(P).jenis==x){
                cout<<"Nama obat = "<<info(P).obat<<" || Jenis obat =
"<<info(P).jenis<<endl;
            };
            P=next(P);
        };
        cout<<"==========="<<endl;
    }else if(first(L)!=NULL && x=="NULL"){
        cout<<"Daftar Obat"<<endl;
        P=first(L);
        while(P!=NULL){
            cout<<"Nama obat = "<<info(P).obat<<" || Jenis obat = "<<info(P).jenis<<endl;
            P=next(P);
        };
    }else{
        cout<<"kosong"<<endl;
    }
}
void DaftarObat(listobat &L){
    address_kolom P;
    infoobat x;
    if(first(L)==NULL){
        //data awal
        x.jenis="pusing";
        x.obat="paracetamol";
        P=AdrKolom(x);
        first(L)=P;
        x.jenis="pusing";
        x.obat="mortin";
        P=AdrKolom(x);
        next(P)=first(L);
        first(L)=P;
        x.jenis="pusing";
        x.obat="naproxen";
        P=AdrKolom(x);
        next(P)=first(L);
        first(L)=P;
        x.jenis="pusing";
        x.obat="aspirin";
        P=AdrKolom(x);
        next(P)=first(L);
        first(L)=P;
        x.jenis="flu";
        x.obat="procold";
        P=AdrKolom(x);
        next(P)=first(L);
        first(L)=P;
        x.jenis="flu";
        x.obat="obh";
        P=AdrKolom(x);
        next(P)=first(L);
        first(L)=P;
        x.jenis="flu";
        x.obat="bodrex";
        P=AdrKolom(x);
        next(P)=first(L);
        first(L)=P;
        x.jenis="flu";
        x.obat="sanmol";
        P=AdrKolom(x);
        next(P)=first(L);
        first(L)=P;
    }else{
        cout<<"Masukan obat yang mau ditambahkan"<<endl;
        cout<<"Jenis obat = ";
        cin>>x.jenis;
        cout<<"Nama obat = ";
        cin>>x.obat;
        P=findobat(L,x.obat);
        if(P==NULL){
            P=AdrKolom(x);
            next(P)=first(L);
            first(L)=P;
            cout<<"Berhasil ditambahkan!"<<endl;
        }else{
            cout<<"Obat dengan nama yang sama sudah terdaftar"<<endl;
        }
    }
    cout<<"menampilkan list obat ? <y/n>"<<endl;
    char y;
    cin>>y;
    if(y=='y'){
        printObat(L,"NULL");
    };
}
address_kolom findJenisObat(listobat L,string x){
    address_kolom P;
    P=first(L);
    while((info(P).jenis!=x)&&(next(P)!=NULL)){
        P=next(P);

    }
    if(info(P).jenis==x){
        return P;
    }else{
        return NULL;
    }
};

address_kolom findobat(listobat L,string x){
    address_kolom P;
    P=first(L);
    while((info(P).obat!=x)&&(next(P)!=NULL)){
        P=next(P);
    };
    if(info(P).obat==x){
        return P;
    }else{
        return NULL;
    }
};
address_baris findpasien(List L,string x){
    address_baris P;
    if(first(L)!=NULL){
        P=first(L);
        while((info(P).nama!=x)&&(next(P)!=NULL)){
            P=next(P);
        };
        if(info(P).nama==x){
            return P;
        }else{
            return NULL;
        };
    }else{
        return NULL;
    }
};

void printPenyakit(listobat L){
    address_kolom P;
    string x;
    P=first(L);
    x=info(P).jenis;
    int i=1;
    cout<<"Obat tersedia untuk jenis penyakit"<<endl;
    cout<<i<<". "<<info(P).jenis<<endl;
    while(P!=NULL){
        if(info(P).jenis!=x){
            i++;
            cout<<i<<". "<<info(P).jenis<<endl;
            x=info(P).jenis;
        };
        P=next(P);
    }
};

//-------------------------------------------

void addFirstBaris(infopasien pasien, List &L){
    address_baris P;
    P=AdrBaris(pasien);
    if(first(L)==NULL){
        first(L)=P;
        next(first(L))=NULL;
    }else{
        next(P)=first(L);
        first(L)=P;
    };
};

void addFirstKolom(infoobat obat,List &L){
    address_kolom Q;
    Q=new ekolom;
    info(Q).obat=obat.obat;
    info(Q).jenis=obat.jenis;
    next(Q)=NULL;
    if(first(L)->col ==NULL){
        first(L)->col=Q;
    }else{
        next(Q)=first(L)->col;
        first(L)->col=Q;
    }
};

void printPasienObat(List L){
    if(first(L)!=NULL){
        ebaris *P;
        P=first(L);
        int i=1;
        while(P!=NULL){
            cout<<"====================="<<endl;
            cout<<i<<" Nama = "<<info(P).nama<<" || Sakit = "<<info(P).sakit<<endl;
            cout<<"  Jenis Obat"<<endl;
            //ekolom *Q=col(P);
            address_kolom Q=col(P);
            while(Q!=NULL){
                cout<<"  -"<<info(Q).obat<<"  Jenis = "<<info(Q).jenis<<endl;
                //cout<<"  -"<<info(next(Q)).obat<<"  Jenis = "<<info(next(Q)).obat<<endl;
                Q=next(Q);
            };
            Q=NULL;
            i++;
            P=next(P);
        };
    }else{
        cout<<"TIDAK ADA"<<endl;
    };
};

void addTKolom(string pasien,address_kolom Q,List &L){
    ebaris *P;
    ekolom *Z;
    Z=new ekolom;
    info(Z).obat=info(Q).obat;
    info(Z).jenis=info(Q).jenis;
    P=findpasien(L,pasien);
    next(Z)=P->col;
    P->col=Z;
    Z=NULL;
};

void DelFirst(List &L){
    if(first(L)!=NULL){
        address_baris P=first(L);
        first(L)=next(first(L));
        next(P)=NULL;
        delete P;
    }else{
        cout<<"LIST KOSONG"<<endl;
    }
};

void DelLast(List &L){
    if(first(L)!=NULL){
        address_baris P=first(L);
        while(next(next(P))!=NULL){
            P=next(P);
        };
        address_baris Q=P;
        Q=next(P);
        next(P)=NULL;
        delete Q;
    }else{
        cout<<"LIST KOSONG"<<endl;
    }
};

void DelTengah(List &L,string x){
    if(first(L)!=NULL){
        address_baris P=first(L);
        while((P!=NULL)&&(info(next(P)).nama!=x)){
            P=next(P);
        };
        address_baris Q=next(P);
        next(P)=next(Q);
        next(Q)=NULL;
        delete Q;

    }else{
        cout<<"LIST KOSONG"<<endl;
    }
};

void CekPosisiDel(List &L,address_baris P){
    if(P!=NULL){
        string x=info(P).nama;
        if(P!=first(L)&&(next(P)!=NULL)){
            //data berada ditengah
            DelTengah(L,x);
        }else if(P==first(L)){
            //data berada di awal;
            DelFirst(L);
        }else if(next(P)==NULL){
            //data berada di akhir
            DelLast(L);
        }
    }else{
        cout<<"DATA TIDAK DITEMUKAN"<<endl;
    }
};

void ShowSatuPasien(List L,address_baris P){
    if(first(L)!=NULL){
        if(P!=NULL){
            address_kolom Q;
            Q=col(P);
            cout<<"Nama Pasien = "<<info(P).nama<<endl;
            cout<<"Daftar Obat"<<endl;
            int i=1;
            while(Q!=NULL){
                cout<<i<<" . "<<info(Q).obat<<" "<<info(Q).jenis<<endl;
                Q=next(Q);
                i++;
            };
        }else{
            cout<<"Nama Pasien Tidak terdaftar"<<endl;
        }
    }else{
        cout<<"LIST KOSONG"<<endl;
    }
};

```

```
void coutObatBanyak(List L,listobat O){
    address_baris P=first(L);
    address_kolom Q;
    const int NMAX=100;
    struct TbL{
        int akhir;
        string obat[NMAX];
        int jumlah[NMAX];
    };

    TbL Tabel;
    Tabel.akhir=-1;

    //memasukan obat yang digunakan pasien kedalam array
    while(P!=NULL){
        Q=col(P);
        while(Q!=NULL){
            Tabel.akhir++;
            Tabel.obat[Tabel.akhir]=info(Q).obat;
            Tabel.jumlah[Tabel.akhir]=0;
            Q=next(Q);
        };
        P=next(P);
    };

    //menghitung jumlah penggunaan obat tertentu di array
    int cek=0;
    while(cek<=Tabel.akhir){
        int pos=cek;
        while(pos<=Tabel.akhir){
            if(Tabel.obat[cek]==Tabel.obat[pos]){
                Tabel.jumlah[cek]++;
            };
            pos++;
        };
        cek++;
    };


    //mencari indeks dengan jumlah terbanyak
    int show=0;
    int bandingIndeks=0;
    int indx=0;
    while(show<=Tabel.akhir){
        if(bandingIndeks<Tabel.jumlah[show]){
            bandingIndeks=Tabel.jumlah[show];
            indx=show;
        };
        show++;
    };

    cout<<"==================================="<<endl;
    cout<<"Obat yang populer digunakan Pasien"<<endl;
    cout<<"Obat = "<<Tabel.obat[indx]<<endl;
    cout<<"Total pengguna = "<<Tabel.jumlah[indx]<<endl;
    cout<<"==================================="<<endl;
};

void printObat3(List L){
    address_baris P=first(L);
    address_kolom Q;
    address_kolom Z;
    bool ada=false;
    int jumlah=0;
    while(P!=NULL){
        Q=col(P);
        jumlah=0;
        while(Q!=NULL){
            jumlah++;
            Q=next(Q);
        };
        if(jumlah>=3){
            ada=true;
            cout<<"================================"<<endl;
            cout<<"Nama pasien = "<<info(P).nama<<endl;
            cout<<"Obat yang Digunakan = ";
            Z=col(P);
            while(Z!=NULL){
                cout<<" "<<info(Z).obat;
                Z=next(Z);
            };
        };
        P=next(P);
    };
    if(!ada){
        cout<<"Tidak ada pasien yang menggunakan obat lebih dari 3"<<endl;
    }
};
```