

Operasi Dasar Singly linked List (Primitive List)



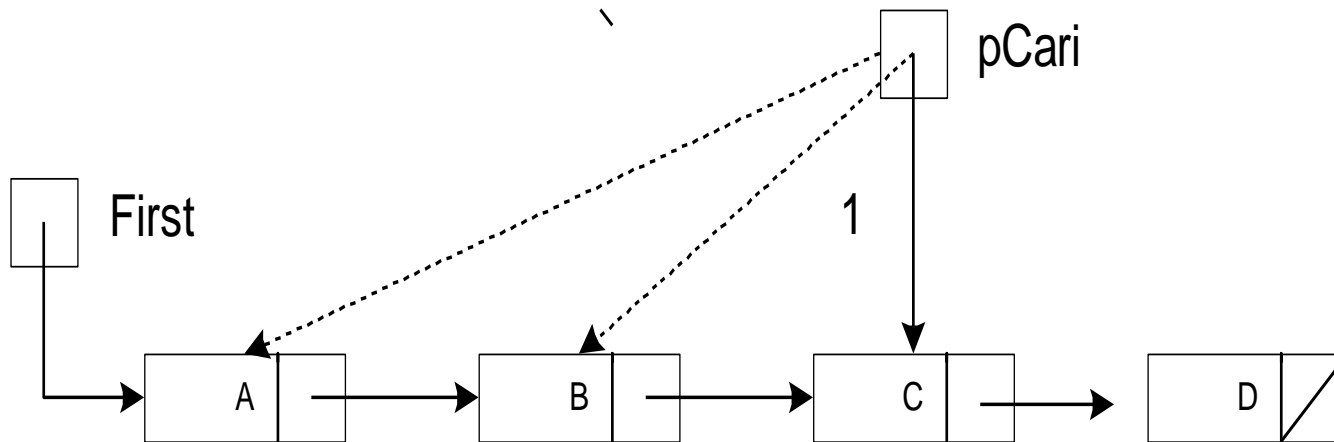
Akmal, S.Si, MT

Mata Kuliah : Struktur Data

Searching (Pencarian)

- Algoritma yang digunakan adalah linear search. Pencarian dilakukan ketika diberikan sebuah list berikut kunci pencariannya. Pencarian dimulai dari elemen pertama dan dilakukan sampai data yang dicari ditemukan **ATAU** data dalam list sudah habis diperiksa. Atau sebaliknya dikatakan bahwa pencarian dilakukan selama data yang dicari belum ditemukan **DAN** data dalam list yang akan diperiksa masih ada { Hukum de Morgan : $(A \text{ or } B)' = A' \text{ and } B'$ dan $(A \text{ and } B)' = A' \text{ or } B'$ } .

Setelah proses pencarian selesai dilakukan maka akan dihasilkan status ditemukan atau tidak ditemukannya elemen yang dicari beserta alamat dari elemen yang dicari tersebut. (NULL jika tidak ditemukan).



Fungsi searching dengan cara linear search

```
void linearSearch(List First,int key, int& found, pointer& pCari){
    found=0;           // belum ditemukan
    pCari=First;       // pencarian mulai dari elemen pertama
    while (found==0 && pCari!=NULL) {    //!found
        if (pCari->info==key)
            found=1;
        else
            pCari=pCari->next;
    }
}
```

Misalkan dalam list First akan dicari elemen dengan kunci = C

```
kunci='C'
linearSearch(First, kunci, ketemu, pCari);
if (ketemu) {
    cout<<"Ditemukan dgn informasinya : "<< pCari->info;
}
else {
    cout<<"Tidak Ditemukan " << endl;
}
```

Insert After (Penyisipan Sesudah Elemen)

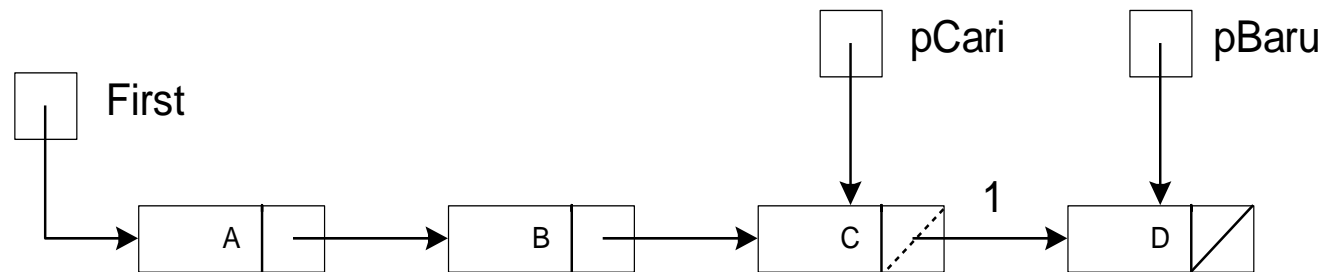
- Penyisipan pada Insert After adalah proses penyisipan sebuah elemen yang baru sesudah suatu elemen tertentu. Alamat elemen dimana elemen akan disisipkan sudah diketahui terlebih dahulu. Untuk mengetahui dan mendapatkan alamat elemen tertentu bisa menggunakan fungsi pencarian linear search.

Langkah-langkah penyisipan :

Temukan suatu elemen dimana elemen yang baru akan disisipkan sesudah elemen tersebut.

Jika ditemukan, maka ada 2 kasus yang mungkin terjadi yaitu

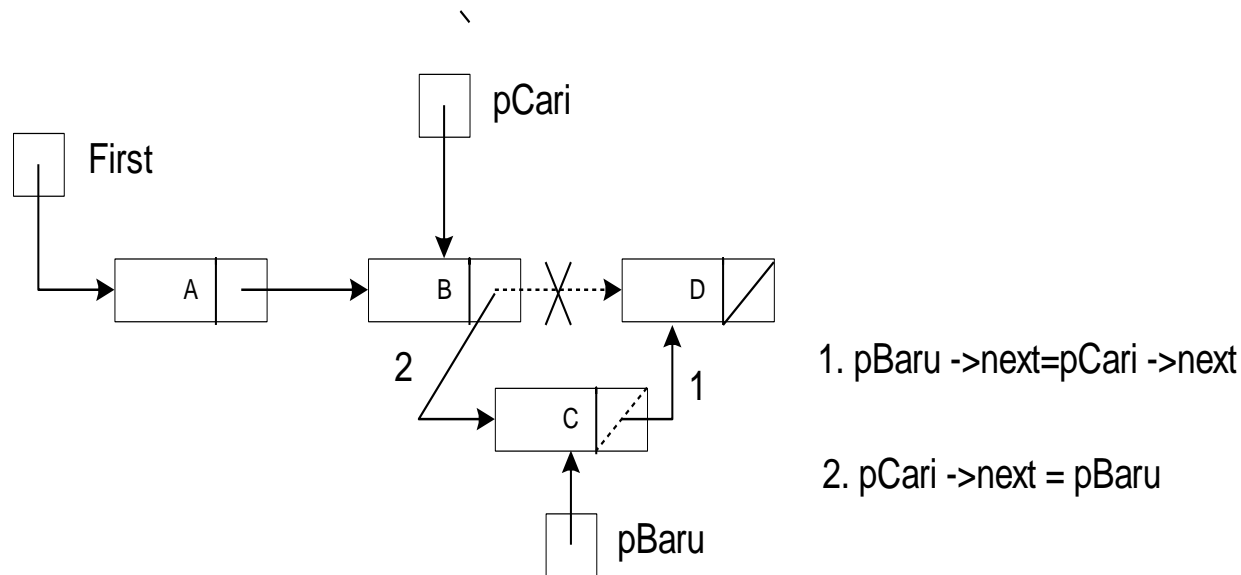
Elemen yang dicari berada di belakang sehingga akan terjadi penyisipan di belakang (insert Last)



1. pCari -> next = pBaru

Insert After (Penyisipan Sesudah Elemen)

- Elemen yang dicari berada di tengah sehingga elemen yang baru akan berada diantara 2 buah elemen



Sambungkan antara elemen yang baru dengan elemen sesudah pCari
($pBaru \rightarrow next = pCari \rightarrow next$)

Sambungkan antara elemen yang ditunjuk oleh pCari dengan elemen yang baru ($pCari \rightarrow next = pBaru$)

Fungsi Insert After

```
void insertAfter(List& First,pointer pCari,pointer pBaru){  
    //I.S : List tidak mungkin kosong, pCari sudah  
    //      diketahui, pBaru sudah terdefinisi  
    //F.S : pBaru disisipkan sesudah pCari  
  
    if (pCari->next==NULL)                // elemen terakhir  
        pCari->next=pBaru;  
    else {                                // bukan terakhir  
        pBaru->next=pCari->next;  
        pCari->next=pBaru;  
    }  
}
```

Pemanggilan fungsi insertAfter

```
cout<<"Masukkan kunci pencarian : "; cin>>key;    //B
linearSearch (Ma21,key,ketemu,pCari);
if (ketemu){
    cout<<" Ditemukan yang dicari "<<endl;
    createElement (p);                               //C
    insertAfter (Ma21,pCari,p);                       // Sisipkan C sesudah B
    traversal (Ma21);                                  //A  B  C  D
}
else {
    cout<<" Tidak Ditemukan yang dicari "<<endl;
}
```

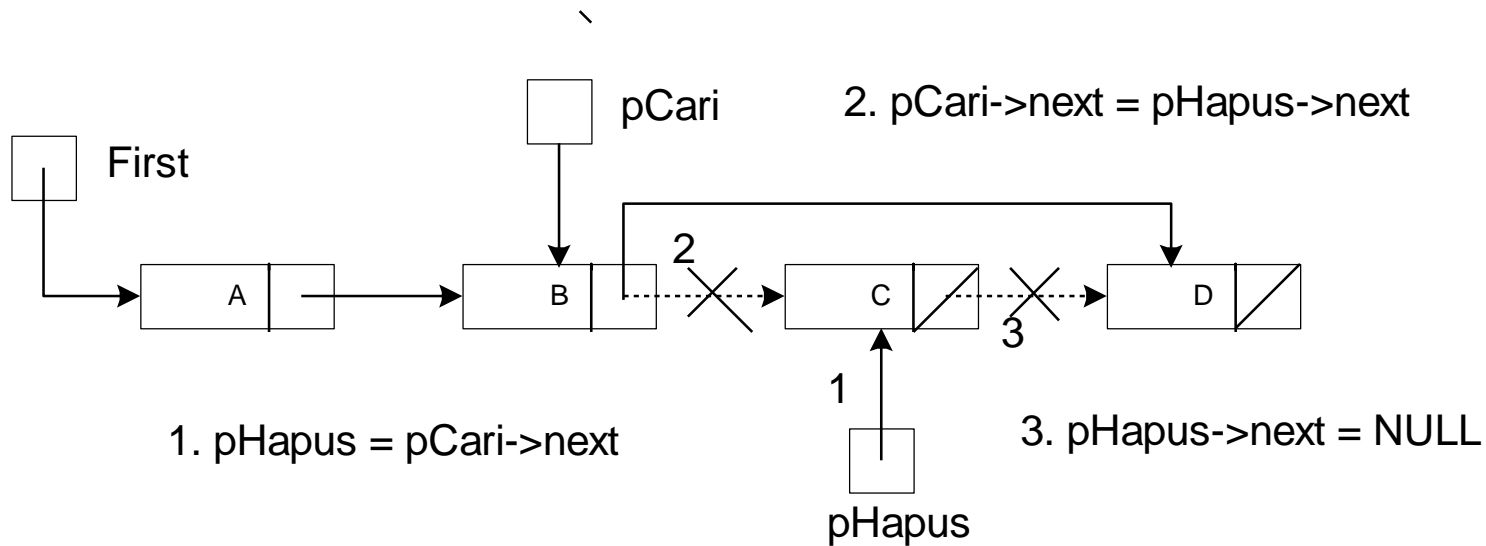
Delete After (Penghapusan Sesudah Elemen)

Penghapusan pada Delete After adalah proses penghapusan sebuah elemen sesudah suatu elemen tertentu. Alamat elemen dimana elemen sebelumnya berada sudah diketahui terlebih dahulu. Untuk mengetahui dan mendapatkan alamat elemen tertentu bisa menggunakan fungsi pencarian linear search.

Langkah-langkah penghapusan :

- ❑ Temukan suatu elemen pencarian , yang akan dihapus adalah elemen sesudah elemen pencarian tersebut.
- ❑ Jika ditemukan, maka ada 3 kasus yang mungkin terjadi yaitu
 - a. Elemen yang dicari (pCari) berada di belakang sehingga tidak ada yang akan dihapus sesudah pCari tersebut.
 - b. pCari menunjuk elemen sebelum yang terakhir maka akan dilakukan penghapusan di belakang (delete Last)
 - ❑ pHapus = pCari->next
 - ❑ pCari->next = NULL
 - c. pCari berada pada selain a dan b atau yang akan dihapus berada diantara 2 buah elemen.
 1. pHapus adalah elemen sesudah pCari
 - pHapus = pCari->next
 2. lepaskan kaitan antara yang ditunjuk oleh pCari dengan yang akan dihapus dengan cara menyambungkan pCari ke yang sesudah pHapus
 - pCari->next = pHapus->next
 3. putuskan kaitan antara yang ditunjuk oleh pHapus dengan elemen sesudahnya
 - pHapus->next = NULL

Delete After



- Antara kasus b dan kasus c bisa digabungkan dengan hanya menjadikannya mengikuti kasus no c saja.

Fungsi Delete After

```
//I.S : List tidak mungkin kosong  
//F.S : pHapus adalah elemen sesudah pCari  
  
void deleteAfter(List& First,pointer pCari,pointer& pHapus){  
    if (pCari->next==NULL){  
        pHapus = NULL;  
        cout<<"Tidak ada yang dihapus "<<endl;  
    }  
    else {  
        pHapus = pCari->next;  
        pCari->next = pHapus->next;        // NULL jika pHapus  
                                           // di belakang  
        pHapus->next = NULL;  
    }  
}
```

Latihan dan Tugas

1. Buatlah fungsi untuk melakukan penghapusan elemen yang dicari (DeletePCari). Dilakukan pencarian menggunakan Searching dan jika ditemukan maka elemen tersebut akan dihapus dari list berkait. Analisis semua kasus yang mungkin terjadi. Tambahkan pointer bantuan untuk membantu proses pemutusan elemen.
2. Buatlah program lengkap semua primitive list dengan pengelolaan menggunakan suatu menu. Kasus list yang ditangani adalah list pegawai dengan atribut NIP, nama, alamat, gol, dan gaji