

UNGUIDED LAPORAN PRAKTIKUM STUKTUR DATA MODUL 5

1. Unguided

4.1. Kode Program :

```
singlelist.h X singlelist.cpp X main.cpp X
1 #ifndef SINGLELIST_H
2 #define SINGLELIST_H
3
4 // Deklarasi tipe data
5 typedef int infotype;
6 typedef struct ElmList *address;
7
8 struct ElmList {
9     infotype info;
10    address next;
11 }
12
13 struct List {
14    address First;
15 };
16
17 // Deklarasi prosedur dan fungsi
18 void createList(List &L);
19 address alokasi(infotype x);
20 void dealokasi(address &P);
21 void printInfo(List L);
22 void insertFirst(List &L, address P);
23 address findElm(List L, infotype x);
24
25 #endif
26
```

```
singlelist.h X singlelist.cpp X main.cpp X
1 #include <iostream>
2 #include "singlelist.h"
3
4 using namespace std;
5
6 void createList(List &L) {
7     L.First = NULL;
8 }
9
10 address alokasi(infotype x) {
11     address P = new ElmList;
12     P->info = x;
13     P->next = NULL;
14     return P;
15 }
16
17 void dealokasi(address &P) {
18     delete P;
19     P = NULL;
20 }
21
22 void printInfo(List L) {
23     address P = L.First;
24     while (P != NULL) {
25         cout << P->info << " ";
26         P = P->next;
27     }
28     cout << endl;
29 }
30
31 void insertFirst(List &L, address P) {
32     P->next = L.First;
33     L.First = P;
34 }
35
```

```
singlelist.h X singlelist.cpp X main.cpp X
1 #include <iostream>
2 #include "singlelist.h"
3
4 using namespace std;
5
6 int main() {
7     List L;
8     address P1, P2, P3, P4, P5;
9
10    createList(L);
11    P1 = alokasi(2);
12    insertFirst(L, P1);
13
14    P2 = alokasi(0);
15    insertFirst(L, P2);
16
17    P3 = alokasi(8);
18    insertFirst(L, P3);
19
20    P4 = alokasi(12);
21    insertFirst(L, P4);
22
23    P5 = alokasi(9);
24    insertFirst(L, P5);
25
26    printInfo(L);
27
28    return 0;
29 }
30
31
```

Output dari Kode Program :

```
12 0 0 2
Process returned 0 (0x0)   execution time : 0.208 s
Press any key to continue.
```

4.2. Code Program :

```
singlelisth X singlelist.cpp X main.cpp X
1  #ifndef SINGLELIST_H
2  #define SINGLELIST_H
3
4  // Deklarasi Data
5  typedef int infotype;
6  typedef struct ElmList *address;
7
8  struct ElmList {
9      infotype info;
10     address next;
11 };
12
13 struct List {
14     address First;
15 };
16
17 // Deklarasi prosedur dan fungsi
18 void createList(List &L);
19 address alokasi(infotype x);
20 void dealokasi(address &P);
21 void printInfo(List L);
22 void insertFirst(List &L, address P);
23 address findElm(List L, infotype x);
24
25 #endif
26
```

```
singlelisth X singlelist.cpp X main.cpp X
4  using namespace std;
5
6  void createList(List &L) {
7      L.First = NULL;
8  }
9
10 address alokasi(infotype x) {
11     address P = new ElmList;
12     P->info = x;
13     P->next = NULL;
14     return P;
15 }
16
17 void dealokasi(address &P) {
18     delete P;
19     P = NULL;
20 }
21
22 void printInfo(List L) {
23     address P = L.First;
24     while (P != NULL) {
25         cout << P->info << " ";
26         P = P->next;
27     }
28     cout << endl;
29 }
30
31 void insertFirst(List &L, address P) {
32     P->next = L.First;
33     L.First = P;
34 }
35
36 address findElm(List L, infotype x) {
37     address P = L.First;
38     while (P != NULL) {
39         if (P->info == x) {
40             return P;
41         }
42         P = P->next;
43     }
44     return NULL;
45 }
46
```

```
singlelisth X singlelist.cpp X main.cpp X
1  #include <iostream>
2  #include "singlelist.h"
3
4  using namespace std;
5
6  int main() {
7      List L;
8      address P1, P2, P3, P4, P5;
9
10     createList(L);
11
12     P1 = alokasi(1);
13     insertFirst(L, P1);
14
15     P2 = alokasi(0);
16     insertFirst(L, P2);
17
18     P3 = alokasi(9);
19     insertFirst(L, P3);
20
21     P4 = alokasi(10);
22     insertFirst(L, P4);
23
24     P5 = alokasi(9);
25     insertFirst(L, P5);
26
27     printInfo(L);
28
29     infotype x = 0;
30     address found = findElm(L, x);
31     if (found != NULL) {
32         cout << x << " ditemukan dalam list" << endl;
33     } else {
34         cout << x << " tidak ditemukan dalam list" << endl;
35     }
36
37     return 0;
38 }
39
```

Output dari Kode Program :

```
C:\Users\affa\Documents\C++\Uniguided Modul\Singlelist\Debug\Uniguided Modul.exe
0:12 0 0 0
0 ditemukan dalam list
Process returned 0 (0x0)   execution time : 0.230 s
Press any key to continue.
```

4.3. Kode Program :

```
singlelist.h X singlelist.cpp X main.cpp X
1 #ifndef SINGLELIST_H
2 #define SINGLELIST_H
3
4 // Deklarasi tipe data
5 typedef int infotype;
6 typedef struct ElmtList *address;
7
8 struct ElmtList {
9     infotype info;
10    address next;
11};
12
13 struct List {
14    address First;
15};
16
17 // Deklarasi prosedur dan fungsi
18 void createList(List &L);
19 address alokasi(infotype x);
20 void dealokasi(address &P);
21 void printInfo(List L);
22 void insertFirst(List &L, address P);
23 int sumInfo(List L);
24
25 #endif
26
```

```
singlelist.h X singlelist.cpp X main.cpp X
4 using namespace std;
5
6 void createList(List &L) {
7     L.First = NULL;
8 }
9
10 address alokasi(infotype x) {
11     address P = new ElmtList;
12     P->info = x;
13     P->next = NULL;
14     return P;
15 }
16
17 void dealokasi(address &P) {
18     delete P;
19     P = NULL;
20 }
21
22 void printInfo(List L) {
23     address P = L.First;
24     while (P != NULL) {
25         cout << P->info << " ";
26         P = P->next;
27     }
28     cout << endl;
29 }
30
31 void insertFirst(List &L, address P) {
32     P->next = L.First;
33     L.First = P;
34 }
35
36 int sumInfo(List L) { // Implementasi fungsi sumInfo
37     int sum = 0;
38     address P = L.First;
39     while (P != NULL) {
40         sum += P->info;
41         P = P->next;
42     }
43     return sum;
44 }
```

```
singlelist.h X singlelist.cpp X main.cpp X
1 #include <iostream>
2 #include "singlelist.h"
3
4 using namespace std;
5
6 int main() {
7     List L;
8     address P1, P2, P3, P4, P5;
9
10    createList(L);
11
12    P1 = alokasi(2);
13    insertFirst(L, P1);
14
15    P2 = alokasi(9);
16    insertFirst(L, P2);
17
18    P3 = alokasi(5);
19    insertFirst(L, P3);
20
21    P4 = alokasi(12);
22    insertFirst(L, P4);
23
24    P5 = alokasi(9);
25    insertFirst(L, P5);
26
27    printInfo(L);
28
29    int total = sumInfo(L); // Rumus fungsi sumInfo
30    cout << "Total info dari kelima elemen adalah " << total << endl;
31
32    return 0;
33 }
34
```

Output dari Kode Program :

```
C:\Users\paula\Documents\C++\Unguided Modul 5\Unguided Modul 5\Debug\Unguided Modul 5.exe
P1: 2
P2: 9
P3: 5
P4: 12
P5: 9
Total info dari kelima elemen adalah 31
Process returned 0 (0x0)   execution time : 0.211 s
Press any key to continue.
```

