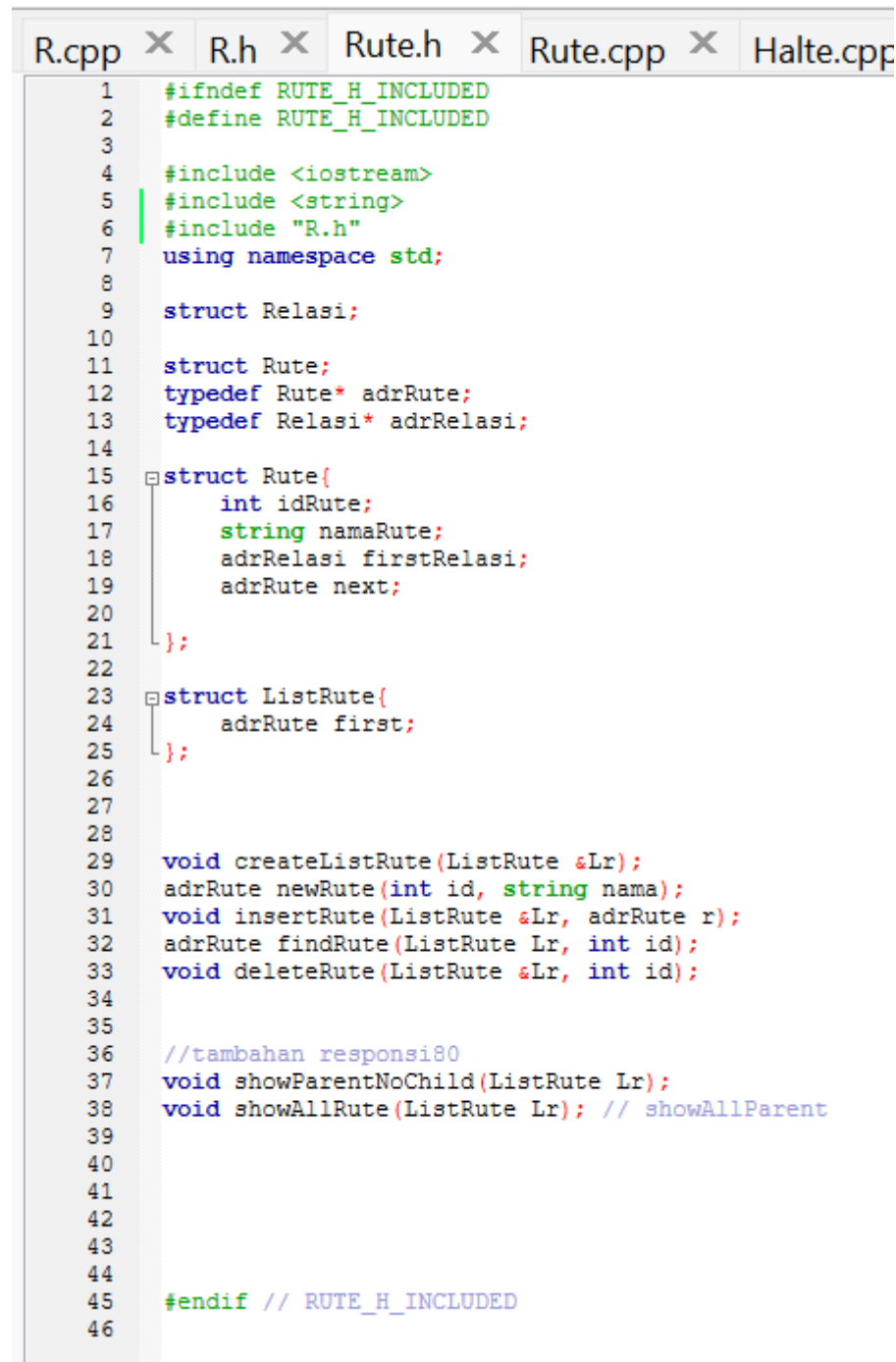


Kode 80%

Kelompok 15 , M.GHYMVANDYAR EL M. 103092400087 & LAURA DJASMIN TANDIAWAN

RUTE H DAN CPP



```
1  #ifndef RUTE_H_INCLUDED
2  #define RUTE_H_INCLUDED
3
4  #include <iostream>
5  #include <string>
6  #include "R.h"
7  using namespace std;
8
9  struct Relasi;
10
11 struct Rute;
12 typedef Rute* adrRute;
13 typedef Relasi* adrRelasi;
14
15 struct Rute{
16     int idRute;
17     string namaRute;
18     adrRelasi firstRelasi;
19     adrRute next;
20 };
21
22
23 struct ListRute{
24     adrRute first;
25 };
26
27
28
29 void createListRute(ListRute &Lr);
30 adrRute newRute(int id, string nama);
31 void insertRute(ListRute &Lr, adrRute r);
32 adrRute findRute(ListRute Lr, int id);
33 void deleteRute(ListRute &Lr, int id);
34
35
36 //tambahan responsi80
37 void showParentNoChild(ListRute Lr);
38 void showAllRute(ListRute Lr); // showAllParent
39
40
41
42
43
44
45 #endif // RUTE_H_INCLUDED
46
```

```

R.cpp X R.h X Rute.h X
1 #include "R.h"
2 #include "Rute.h"
3 #include "R.h"
4 using namespace std;
5
6
7
8 void main() {
9     int n;
10
11     // Input
12     cout << "Masukkan jumlah simpul: ";
13     cin >> n;
14
15     // Output
16     cout << "Masukkan jumlah sisi: ";
17     cin >> m;
18
19     // Create graph
20     Graph g(n, m);
21
22     // Add edges
23     for (int i = 1; i <= m; i++) {
24         int u, v;
25         cout << "Masukkan sisi ke-" << i << ": ";
26         cin >> u >> v;
27         g.addEdge(u, v);
28     }
29
30     // Display graph
31     g.display();
32
33     // Find shortest path
34     int s, t;
35     cout << "Masukkan simpul awal: ";
36     cin >> s;
37     cout << "Masukkan simpul tujuan: ";
38     cin >> t;
39
40     // Find shortest path
41     g.findShortestPath(s, t);
42
43     // End
44     return 0;
45 }

```

HALTE H DAN CPP

R.cpp x R.h x Rute.h x *Rute.cpp x Halte.cpp x Halte.h x main.cpp x

D:\Desktop\vandy bahasa c\TubesStrukdat02\TubesStrukdat\R.cpp

Project: TubesStrukdat.

```
4  #include <iostream>
5  #include <string>
6  using namespace std;
7
8
9  // struct deklarasi
10 struct Halte;
11 typedef Halte* adrHalte;
12
13
14 struct Halte{
15     string namaHalte;
16     adrHalte next;
17
18 };
19
20
21 struct ListHalte {
22     adrHalte first;
23
24 };
25
26
27 void createListHalte(ListHalte &Lh); // list kosong
28 adrHalte newHalte(string nama); // node halte baru
29 void insertHalte(ListHalte &Lh, adrHalte p); //
30 adrHalte findHalte(ListHalte &Lh, string nama); // serachir halte(child) be
31 void deleteHalte(ListHalte &Lh, ListRute &Lr, string nama); //hapus (by sea
32
33 //tambahan responsi80
34 void showAllHalte(ListHalte &Lh); //showAll halte
35
36
37
38
39
40
41
42 #endif // HALTE_H_INCLUDED
43
```


Edit View Search Project Build Debug Fortran wxSmith Tools
 ocks Settings Help

File Explorer

- Files
- Source Space
- TubesStrukdat.
- Sources
- Headers

Code Editor

R.cpp

```

1  #ifndef R_H
2  #define R_H
3
4  #include <iostream>
5  #include <string>
6
7  using namespace std;
8
9
10 #include "Rute.h"
11 #include "Rute.h"
12
13
14
15 struct Rute;
16 typedef Rute* adRute;
17
18
19
20 struct Rute{
21     adRute child; // ke node Rute
22     adRute next; // p ke node Rute selanjutnya
23 };
24
25
26
27
28 adRute newRute(adRute child);
29 void insertRute(adRute parent, adRute n);
30 void deleteRute(adRute parent, string namaRute);
31 adRute findRute(adRute parent, string namaRute);
32
33
34 //laporan
35
36 void showChildFromParent(adRute parent); //menampilkan child dari parent tertentu
37 void showAllParentWithChild(list<Rute> &r); //menampil setiap parent beserta childnya
38 void showParentOfChild(list<Rute> &r, string namaRute); //parent yang beresil dengan child tertentu
39
40 //count
41 int countRuteFromParent(adRute parent);
42 int countRuteChild(list<Rute> &r, string namaRute);
43 int countChildFromRute(list<Rute> &r, list<Rute> &r);
44
45
46 void editRute(adRute parent, list<Rute> &r, string oldRute, string newRute);
47
48
49 //tambahan response
50 void showChildFromParent(list<Rute> &r, list<Rute> &r);
51 void showAllParent(list<Rute> &r, list<Rute> &r);
52
53
54
55 //tambahan yang mungkin
56 void deleteRuteFromRute(list<Rute> &r, string namaRute);
57
58
59
60 #endif // R_H
61
62
  
```

R.cpp X R.h X Rute.h X *Rute.cpp

```

1  include<iostream>
2  include "R.h"
3  using namespace std;
4
5
6
7  dirnolasi newnolasi(adrmolte child){
8      adrmolasi n = new nolasi;
9      n->child = child;
10     n->next = null;
11     return n;
12
13
14
15  void insertnolasi(adrmolte parent, adrmolasi n){
16      if(parent->firstnolasi == null){
17          parent->firstnolasi = n;
18      }else{
19          adrmolasi p = parent->firstnolasi;
20          while(p->next != null){
21              p = p->next;
22          }
23          p->next = n;
24      }
25
26
27
28
29  dirnolasi findnolasi(adrmolte parent, string namanalce){
30      adrmolasi p = parent->firstnolasi;
31      while(p != null){
32          if(p->child->namanalce == namanalce){
33              return p;
34          }
35          p = p->next;
36      }
37      return null;
38
39
40
41
42  void deletenolasi(adrmolte parent, string namanalce){
43      if(parent->firstnolasi == null) return;
44
45      adrmolasi cur = parent->firstnolasi;
46      adrmolasi prev = null;
47
48      while(cur != null && cur->child->namanalce != namanalce){
49          prev = cur;
50          cur = cur->next;
51      }
52
53      if(cur == null) return;
54
55      if(prev == null){
56          parent->firstnolasi = cur->next;
57      } else {
58          prev->next = cur->next;
59      }
60
61      delete cur;
62
63

```

```

void showChildFromParent(adrmute parent){
    //q rute tidak temu
    if(parent == null){
        cout << "rute tidak ditemukan\n";
        return;
    }

    cout << "rute " << parent->namaRute << " memiliki halte:\n";
    adrute p = parent->firstRute;

    //q rute ada punya rute
    if(p != null){
        cout << "[tidak ada halte]\n";
        return;
    }

    while(p != null){
        cout << " " << p->child->namaRute << endl;
        p = p->next;
    }

void showAllParentWithChild(adrmute ir){
    adrute r = ir->first;
    while(r != null){
        showChildFromParent(r);
        r = r->next;
    }

void showParentOfChild(adrmute ir, string namaRute){
    adrute r = ir->first;
    bool found = false;

    while(r != null){
        if(findRute(r, namaRute) != null){
            cout << "rute: " << r->namaRute << endl;
            found = true;
        }
        r = r->next;
    }

    if(!found){
        cout << "tidak memiliki rute(parent)\n";
    }

//count
int countRuteParent(adrmute parent){
    int count = 0;
    adrute p = parent->firstRute;
    while(p != null){
        count++;
        p = p->next;
    }

    return count;
}

```

```

131
132
133 int countRelasiChild(ListRute Lr, string namaHalte){
134     int count = 0;
135     adrRute r = Lr.first;
136
137     while(r != NULL){
138         if(findRelasi(r, namaHalte) != NULL){
139             count++;
140         }
141         r = r->next;
142     }
143     return count;
144 }
145
146
147 int countChildNoRelasi(ListHalte Lh, ListRute Lr){
148     int count = 0;
149     adrHalte h = Lh.first;
150
151     while(h != NULL){
152         if(countRelasiChild(Lr, h->namaHalte) == 0){
153             count++;
154         }
155         h = h->next;
156     }
157     return count;
158 }
159
160
161
162 void editRelasi(adrRute parent, ListHalte Lh, string oldHalte, string newHalte){
163     if(parent == NULL) return;
164
165     //cek relasi lama
166     adrRelasi R = findRelasi(parent, oldHalte);
167     if(R == NULL){
168         cout << "Relasi lama tidak di temukan \n";
169         return;
170     }
171
172     //carilahaltebaru
173     adrHalte hNew = findHalte(Lh, newHalte);
174     if(hNew == NULL){
175         cout << "Halte baru tidak ditemukan\n";
176         return;
177     }
178
179     //cek dup
180     if(findRelasi(parent, newHalte) != NULL){
181         cout << "Halte sudah ada di Rute ini\n";
182         return;
183     }
184
185     //gantiR
186     R->child = hNew;
187
188     cout << "Relasi berhasil dirubah\n";
189
190 }
191

```



```

193 void showAllData(ListRute Lr, ListHalte Lh){
194
195     cout << "----- Data Rute & Halte ----- \n\n";
196
197     // 1. Parent + Child
198     if(Lr.first == NULL){
199         cout << "[TIDAK ADA RUTE]\n\n";
200     } else {
201         adrRute r = Lr.first;
202         while(r != NULL){
203             cout << "Rute: " << r->namaRute << endl;
204
205             if(r->firstRelasi == NULL){
206                 cout << " [TIDAK MEMILIKI HALTE]\n";
207             } else {
208                 adrRelasi rel = r->firstRelasi;
209                 while(rel != NULL){
210                     if(rel->child != NULL){
211                         cout << " - " << rel->child->namaHalte << endl;
212                     } else {
213                         cout << " - [RELASI RUSAK]\n";
214                     }
215                     rel = rel->next;
216                 }
217             }
218
219             cout << endl;
220             r = r->next;
221         }
222     }
223
224     // 2. Child tanpa Parent
225     cout << "Halte tanpa Rute:\n";
226
227     if(Lh.first == NULL){
228         cout << "[TIDAK ADA HALTE]\n";
229     } else {
230         adrHalte h = Lh.first;
231         bool ada = false;
232
233         while(h != NULL){
234             if(countRelasiChild(Lr, h->namaHalte) == 0){
235                 cout << "- " << h->namaHalte << endl;
236                 ada = true;
237             }
238             h = h->next;
239         }
240
241         if(!ada){
242             cout << "[SEMUA HALTE SUDAH TERHUBUNG]\n";
243         }
244     }
245
246     cout << "\n-----\n";
247 }
248
249 // tambahan
250
251 void deleteRelasiByHalte(ListRute &Lr, string namaHalte){
252     adrRute r = Lr.first;
253     while(r != NULL){
254         deleteRelasi(r, namaHalte);
255         r = r->next;
256     }
257 }
258
259
260
261

```

main.cpp tampilan

```
"D:\Desktop\vandy bahasa c\" X + v

===== SISTEM MANAJEMEN RUTE BUS =====
1. Tambah Rute
2. Tambah Halte
3. Tambah Relasi Rute - Halte
4. Tampilkan Semua Rute
5. Tampilkan Semua Halte
6. Tampilkan Halte dari Rute
7. Tampilkan Rute dari Halte
8. Edit Relasi
9. Hapus Rute
10. Hapus Halte
11. Tampilkan Semua Data
0. Keluar
Pilih: 1
Masukkan ID Rute : 09
Masukkan Nama Rute : sepanjang

===== Tampilkan Semua Data =====
0. Keluar
Pilih: 1
Masukkan ID Rute : 09
Masukkan Nama Rute : sepanjang
Rute berhasil ditambahkan
```

0. Keluar

Pilih: 2

Masukkan Nama Halte : its

Halte berhasil ditambahkan

0. Keluar

Pilih: 3

Masukkan ID Rute : 09

Masukkan Nama Halte : its

Relasi berhasil ditambahkan

0. Keluar

Pilih: 4

Daftar Rute:

ID:9 , Nama:sepanjang

===== SISTEM MANAJEMEN RUTE BUS =====

11. Tampilkan Semua Data

0. Keluar

Pilih: 5

Daftar Halte:

- its

0. Keluar

Pilih: 6

Masukkan ID Rute : 09

Rute sepanjang memiliki halte:

- its

0. Keluar

Pilih: 7

Masukkan Nama Halte : its

Rute: sepanjang

Pilih: 8

Masukkan ID Rute : 08

Masukkan Nama Halte Lama : unesa

Masukkan Nama Halte Baru : its

Relasi berhasil dirubah

```
0. Keluar
Pilih: 9
Masukkan ID Rute : 09
Rute dihapus
```

```
0. Keluar
Pilih: 10
Masukkan Nama Halte : its
Halte dihapus
```

```
Pilih: 11
===== Data Rute & Halte =====

Rute: kediri
      [TIDAK MEMILIKI HALTE]

Halte tanpa Rute:
- unesa
```

```
0. Keluar
Pilih: 1
Masukkan ID Rute : 08
Masukkan Nama Rute : kediri
Rute berhasil ditambahkan
```