

I. MAIN PROGRAM :

The image shows a Windows desktop with two code editors. The left editor, titled 'D:\Drive I\Archive\Study\Kampus\Semester - III\Struktur Data Kompter\praktek\TB\Program_operatorKAI...', displays C++ code for a menu-driven program. The code includes a main function with a menu loop, a switch statement for menu options, and a linked list structure. The right editor, titled 'C:\MINDOS\SYSTEM32\cmd.exe - wslw KAI', shows the output of the program. The output displays a menu with options 1 (Pendaftaran Kereta Api Rosie) and 2 (Exit). The user has selected option 1, and the program has displayed a list of names: Arvin Wini Putra, Gema Putranto Sudibyo, Renanda Dwi Yanto, and 12015 Hoiron. The program also displays a copyright notice: 'Copyright © Crack12015_REV12'.

II. MAIN MENU :

The image displays a C++ program in a code editor and its execution output in a separate window.

Code Editor (Left):

- File: D:\Drive I\Archie\Study\Kampus\Semester - III\Struktur Data\praktek\Program_operator KAI...
- Menu: File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, Help
- Open files: Program_operatorKAI.cpp, KAI.cpp, LinkedListWBinaryTree.cpp, dlist.cpp
- Code content (Program_operatorKAI.cpp):

```
87 //menu dari metode
88 void menu()
89 {
90
91     cout<<"|***** PILIHAN MENU *****|\n";
92     cout<<"|***** PILIHAN MENU *****|\n";
93     cout<<"|----- DAFTAR ANTRIAN KERETA API(KODE KERETA)-----|\n";
94     cout<<"| 1. Daftar Antrian Kereta Api(KODE KERETA) |\n";
95     cout<<"| 2. Keberangkatan |\n";
96     cout<<"| 3. Memberangkatkan Semua Antrian |\n";
97     cout<<"| 4. Lihat Daftar Antrian Kereta |\n";
98     cout<<"| 5. Exit |\n";
99     cout<<"|***** PILIHAN MENU *****|\n";
100     cout<<endl;
101 }
102
103 //metode linked list
104 void LinkedList()
105 {
106     char pilihMenu;
107     int ulang = 1;
108
109     do
110     {
111         menu();
112         printf("Input Menu : ");
113         pilihMenu = getch();
114         printf("\n");
115         switch(pilihMenu)
116         {
117             case '1':
118                 insertData();
119                 break;
120             case '2':
121                 removeData();
122                 break;
123             case '3':
124                 bersih();
125                 break;
126             case '4':
127                 tampil();
128                 break;
129             case '5':
130                 penutup();
131                 break;
132             default :
133                 break;
134         }
135         ulang = 1;
136     } while (ulang == 1);
137 }
```

Execution Output (Right):

- Command prompt: C:\WINDOWS\system32\cmd.exe - wkwk KAI
- Output:

```
***** PILIHAN MENU *****
----- DAFTAR ANTRIAN KERETA API(KODE KERETA)-----
1. Daftar Antrian Kereta Api(KODE KERETA)
2. Keberangkatan
3. Memberangkatkan Semua Antrian
4. Lihat Daftar Antrian Kereta
5. Exit
***** PILIHAN MENU *****

Input Menu : _
```

Taskbar (Bottom):

- Line 78, Column 48
- Tab Size: 4
- C++
- System clock: 21:10, 11/01/2018

III. Push

```
150 }
151 //fungsi untuk memasukkan data
152 void insertData()
153 {
154     int angka;
155     string nama, jam;
156     Node *nodeBaru;
157     nodeBaru = new Node;
158
159     cout<<"Masukkan kode Kereta (Angka) : ";
160     cin>>angka;
161     cout<<"Masukkan Nama Kereta      : ";
162     cin>>nama;
163     cout<<"Masukkan jam keberangkatan Kereta (HH:MM:SS:pm/am) : ";
164     cin>>jam;
165
166     nodeBaru->data = angka;
167     nodeBaru->kereta = nama;
168     nodeBaru->waktu = jam;
169     nodeBaru->next = NULL;
170
171     if(tail == NULL)
172     {
173         head = tail = nodeBaru;
174         tail->next = NULL;
175     }
176     else
177     {
178         tail->next = nodeBaru;
179         tail = nodeBaru;
180     }
181     cout<<"\n|+++++||\n";
182     cout<<"Kode Kereta "<<angka<<" masuk antrian! "<<endl;
183     cout<<"Antrian Kereta " << nama << endl;
184     cout<<"Waktu antrian! " << jam << endl;
185     cout<<"|+++++||\n";
186     puts("Press any key for back to the menu");
187     getch();
188     system("cls");
189 }
```

```
***** PILIHAN MENU *****
-----
1. Daftar Antrian Kereta Api<KODE KERETA>
2. Keberangkatan
3. Memberangkatkan Semua Antrian
4. Lihat Daftar Antrian Kereta
5. Exit
-----

Input Menu : 1
Masukkan kode Kereta (Angka) : 12015
Masukkan Nama Kereta      : Lodaya
Masukkan jam keberangkatan Kereta (HH:MM:SS:pm/am) : 11:45:00pm

||+++++||
Kode Kereta 12015 masuk antrian!
Antrian Kereta! Lodaya
Waktu antrian! 11:45:00pm
||+++++||
Press any key for back to the menu
```

IV. Showing All Element

```
256 puts("Data Masih Kosong!\n");
257
258 puts("Press any key for back to the menu");
259 getch();
260 system("cls");
261 }
262
263 //untuk menampilkan list data
264 void tampil()
265 {
266     Node *see;
267
268     see = head;
269     int no = 1;
270     if(tail != NULL)
271     {
272         puts("Daftar Data Antrian Kereta Api :");
273         puts("");
274         while(see != NULL)
275         {
276             cout<<"no<<". "<<see->data<<" "<<see->waktu<<" "<<see->kereta<<endl;
277             puts("");
278             no++;
279             see = see->next;
280         }
281         puts("\n");
282     }
283     else
284     {
285         puts("Data masih dalam keadaan kosong!\n");
286     }
287     puts("Press any key for back to the menu");
288     getch();
289     system("cls");
290 }
291
292 //fungsi penutup / keluar
293 void penutup()
294 {
295     cout<<"\n\tPROGRAM EXIT<<endl;
296     cout<<"Matur suwun geh pak dosen wess bimbing kita : "<<endl;
297     cout<<" Amin Widi Duta      : 5160411134"<<endl;
298 }
```

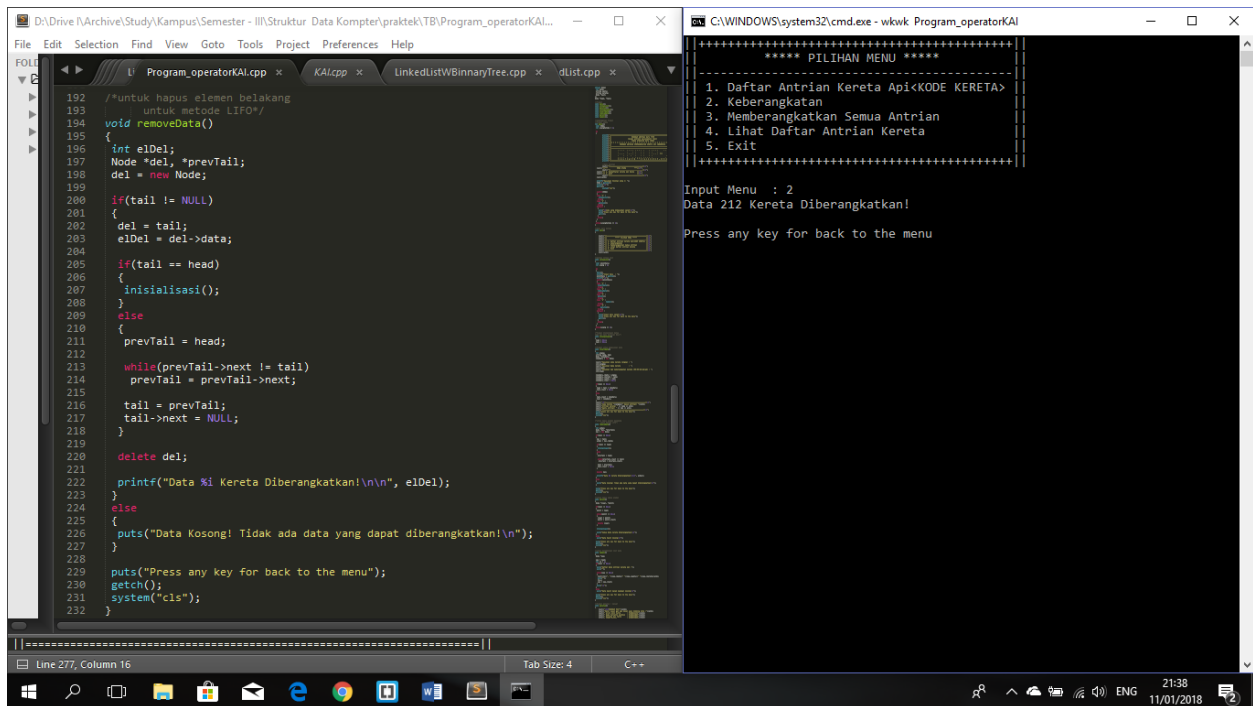
```
***** PILIHAN MENU *****
-----
1. Daftar Antrian Kereta Api<KODE KERETA>
2. Keberangkatan
3. Memberangkatkan Semua Antrian
4. Lihat Daftar Antrian Kereta
5. Exit
-----

Input Menu : 4
Daftar Data Antrian Kereta Api :

1. 12015 11:45:00pm Lodaya
2. 212 12:00:00pm Serayu

Press any key for back to the menu
```

V. Pop Stack Linkedlist one-by-one



The screenshot shows a C++ IDE with the file `Program_operatorKAL.cpp` open. The code implements a `removeData()` function that removes the last element from a linked list. The function uses a `tail` pointer to traverse to the last node, then updates the `prevTail` pointer to point to the previous node and sets the `tail->next` to `NULL`. The terminal window on the right shows the program's output, including a menu, the selection of option 2, and the successful removal of data 212.

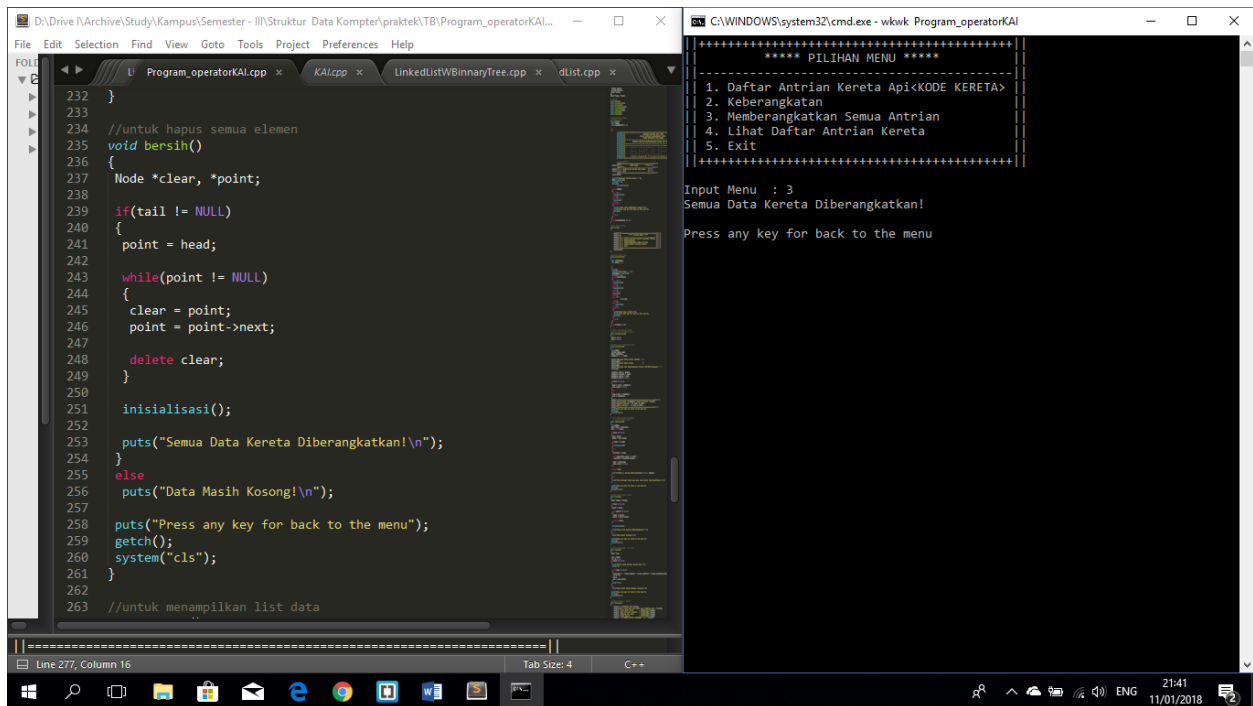
```
192  /*untuk hapus elemen belakang
193  untuk metode LIFO*/
194  void removeData()
195  {
196      int elDel;
197      Node *del, *prevTail;
198      del = new Node;
199
200      if(tail != NULL)
201      {
202          del = tail;
203          elDel = del->data;
204
205          if(tail == head)
206          {
207              inisialisasi();
208          }
209          else
210          {
211              prevTail = head;
212
213              while(prevTail->next != tail)
214                  prevTail = prevTail->next;
215
216              tail = prevTail;
217              tail->next = NULL;
218          }
219
220          delete del;
221
222          printf("Data %i Kereta Diberangkatkan!\n\n", elDel);
223      }
224      else
225      {
226          puts("Data Kosong! Tidak ada data yang dapat diberangkatkan!\n");
227      }
228
229      puts("Press any key for back to the menu");
230      getch();
231      system("cls");
232  }
```

```
***** PILIHAN MENU *****
1. Daftar Antrian Kereta Api<KODE KERETA>
2. Keberangkatan
3. Memberangkatkan Semua Antrian
4. Lihat Daftar Antrian Kereta
5. Exit
*****

Input Menu : 2
Data 212 Kereta Diberangkatkan!

Press any key for back to the menu
```

VI. Pop All Stack



The screenshot shows a C++ IDE with the file `Program_operatorKAL.cpp` open. The code implements a `bersih()` function that removes all elements from a linked list. It uses a `point` pointer to traverse the list, updating `clear` to point to the next node and deleting the current node. The terminal window on the right shows the program's output, including a menu, the selection of option 3, and the successful removal of all data.

```
232  }
233
234  //untuk hapus semua elemen
235  void bersih()
236  {
237      Node *clear, *point;
238
239      if(tail != NULL)
240      {
241          point = head;
242
243          while(point != NULL)
244          {
245              clear = point;
246              point = point->next;
247
248              delete clear;
249          }
250
251          inisialisasi();
252
253          puts("Semua Data Kereta Diberangkatkan!\n");
254      }
255      else
256      {
257          puts("Data Masih Kosong!\n");
258      }
259
260      puts("Press any key for back to the menu");
261      getch();
262      system("cls");
263  }
```

```
***** PILIHAN MENU *****
1. Daftar Antrian Kereta Api<KODE KERETA>
2. Keberangkatan
3. Memberangkatkan Semua Antrian
4. Lihat Daftar Antrian Kereta
5. Exit
*****

Input Menu : 3
Semua Data Kereta Diberangkatkan!

Press any key for back to the menu
```