

# **LAPORAN PRAKTIKUM**

## **“Pertemuan ke-7:Post Test - Simulasi Queue dengan Array Circular”**

Diajukan untuk memenuhi salah satu praktikum Mata Struktur Data Informatika yang diampu oleh:

Dr., Ardiansyah, S.T., M.Cs.



Disusun Oleh:

Mohammad Farid Hendianto 2200018401

A / Rabu 10.30 – 13.30 Lab. Jaringan

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS AHMAD DAHLAN**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**TAHUN 2023**

Gunakan postes Pertemuan 5 untuk membuat model antrian menggunakan link list :

a. no ganjil : double link list

Sebelum merubah menjadi double link list (versi queue), mari kita lihat kodingan sebelumnya pada pertemuan 5.

```

1 #include <iostream>
2 using namespace std;
3
4 void errorInput() {
5     cout << "Input tidak valid" << endl;
6 }
7
8 struct Data {
9     string province;
10    string city;
11    string nkk;
12    string nama;
13    struct born {
14        string place;
15        int date;
16        int month;
17        int year;
18    } born;
19    bool gender;
20    string bloodtype;
21
22    struct Address {
23        string address;
24        int neighborhoodAssociation;
25        int communityAssociation;
26        string district;
27        string subdistrict;
28    } address;
29    string religion;
30    string isMarried;
31    string job;
32    string citizenship;
33    string validUntil;
34    struct Created {
35        int date;
36        int month;
37        int year;
38    } created;
39 };
40
41 class Node {
42 public:
43     Data data;
44     Node* next;
45     Node* prev;
46
47     Node(Data data, Node* next, Node* prev) {
48         this->data = data;
49         this->next = next;
50         this->prev = prev;
51     }
52     Node() {
53         this->next = nullptr;
54         this->prev = nullptr;
55     }
56
57     bool isEmpty() {
58         return this->next == nullptr && this->prev == nullptr ? true : false;
59     }
60     bool isEmpty() {
61         return this->next == nullptr && this->prev == nullptr ? true : false;
62     }
63
64     void enqueue(Data data) {
65         Node* newNode = new Node(data, nullptr, nullptr);
66         if (isEmpty()) {
67             this->next = newNode;
68             this->prev = newNode;
69         } else {
70             this->prev->next = newNode;
71             newNode->prev = this->prev;
72             this->prev = newNode;
73         }
74     }
75     void dequeue() {
76         if (isEmpty()) {
77             cout << "linked list kosong" << endl;
78             return;
79         }
80         Node* temp = this->next;
81         this->next = this->next->next;
82         if (this->next != nullptr) {
83             this->next->prev = nullptr;
84         }
85         delete temp;
86     }
87 };
88
89 class KRP {
90 private:
91     Node* head = nullptr;
92     Node* tail = nullptr;
93     int count = 0;
94
95 public:
96     bool validateProvince(string province) {
97         return province.length() <= 50;
98     }
99
100     bool validateCity(string city) {
101         return city.length() <= 50;
102     }
103
104     bool validateNkk(string nkk) {
105         return nkk.length() <= 16;
106     }
107     bool validateNama(string nama) {
108         return nama.length() <= 50;
109     }
110     bool validateBornPlace(string bornPlace) {
111         return bornPlace.length() <= 50;
112     }
113

```

```

124 }
125
126 bool validateDate(int date) {
127     return date >= 1 && date <= 31;
128 }
129
130 bool validateMonth(int month) {
131     return month >= 1 && month <= 12;
132 }
133
134 bool validateYear(int year) {
135     return year >= 1900 && year <= 2023;
136 }
137
138 bool validateGender(int gender) {
139     return gender == 0 || gender == 1;
140 }
141
142 bool validateBloodType(string bloodType) {
143     return bloodType == "A" || bloodType == "B" || bloodType == "AB" || bloodType == "O";
144 }
145
146 bool validateAddress(string address) {
147     return address.length() <= 200;
148 }
149
150 bool validateNeighborhoodAssociation(int neighborhoodAssociation) {
151     return neighborhoodAssociation >= 0 && neighborhoodAssociation <= 999;
152 }
153
154 bool validateCommunityAssociation(int communityAssociation) {
155     return communityAssociation >= 0 && communityAssociation <= 999;
156 }
157
158 bool validateSubDistrict(string subDistrict) {
159     return subDistrict.length() <= 20;
160 }
161
162 bool validateDistrict(string district) {
163     return district.length() <= 20;
164 }
165
166 bool validateReligion(string religion) {
167     return religion == "Islam" || religion == "Kristen" || religion == "Katolik" || religion == "Hindu" || religion == "Buddha" || religion == "Konhucu";
168 }
169
170 bool validateIsMarried(string isMarried) {
171     return isMarried == "Kawin" || isMarried == "Belum Kawin";
172 }
173
174 bool validateIDN(string idn) {
175     return idn.length() <= 20;
176 }
177
178 void input() {
179     Node* newNode = new Node;
180     do {
181         cout << "Provinsi: ";
182         getline(cin, newNode->data.province);
183         if (!validateProvince(newNode->data.province)) {
184             errorInput();
185         }
186     } while (!validateProvince(newNode->data.province));
187     do {
188         cout << "Kota: ";
189         getline(cin, newNode->data.city);
190         if (!validateCity(newNode->data.city)) {
191             errorInput();
192         }
193     } while (!validateCity(newNode->data.city));
194     do {
195         cout << "RT: ";
196         getline(cin, newNode->data.rtk);
197         if (!validateRTK(newNode->data.rtk)) {
198             errorInput();
199         }
200     } while (!validateRTK(newNode->data.rtk));
201     do {
202         cout << "Nama: ";
203         getline(cin, newNode->data.name);
204         if (!validateName(newNode->data.name)) {
205             errorInput();
206         }
207     } while (!validateName(newNode->data.name));
208     do {
209         cout << "Tempat lahir: ";
210         getline(cin, newNode->data.born.place);
211         if (!validateBornPlace(newNode->data.born.place)) {
212             errorInput();
213         }
214     } while (!validateBornPlace(newNode->data.born.place));
215     do {
216         cout << "Tanggal lahir: ";
217         cin >> newNode->data.born.date;
218         if (!validateDate(newNode->data.born.date)) {
219             errorInput();
220         }
221     } while (!validateDate(newNode->data.born.date));
222     do {
223         cout << "Bulan lahir: ";
224         cin >> newNode->data.born.month;
225         if (!validateMonth(newNode->data.born.month)) {
226             errorInput();
227         }
228     } while (!validateMonth(newNode->data.born.month));
229     do {
230         cout << "Tahun lahir: ";
231         cin >> newNode->data.born.year;
232         if (!validateYear(newNode->data.born.year)) {
233             errorInput();
234         }
235     } while (!validateYear(newNode->data.born.year));
236     do {
237         cout << "Jenis kelamin (Masuk laki, isperempuan): ";
238         cin >> newNode->data.gender;
239         if (!validateGender(newNode->data.gender)) {
240             errorInput();
241         }
242     } while (!validateGender(newNode->data.gender));
243     cin.ignore();
244     do {
245         cout << "Golongan darah: ";
246         getline(cin, newNode->data.bloodType);
247         if (!validateBloodType(newNode->data.bloodType)) {
248             errorInput();
249         }
250     } while (!validateBloodType(newNode->data.bloodType));
251     do {
252         cout << "Alamat: ";
253         getline(cin, newNode->data.address.address);
254         if (!validateAddress(newNode->data.address.address)) {
255             errorInput();
256         }
257     } while (!validateAddress(newNode->data.address.address));
258     do {
259         cout << "RT: ";
260         cin >> newNode->data.address.neighborhoodAssociation;
261         if (!validateNeighborhoodAssociation(newNode->data.address.neighborhoodAssociation)) {
262             errorInput();
263         }
264     } while (!validateNeighborhoodAssociation(newNode->data.address.neighborhoodAssociation));
265     do {
266         cout << "Desa: ";
267         cin >> newNode->data.address.communityAssociation;
268         if (!validateCommunityAssociation(newNode->data.address.communityAssociation)) {
269             errorInput();
270         }
271     } while (!validateCommunityAssociation(newNode->data.address.communityAssociation));
272     cin.ignore();
273     do {
274         cout << "Kecamatan: ";
275         getline(cin, newNode->data.address.subDistrict);
276         if (!validateSubDistrict(newNode->data.address.subDistrict)) {
277             errorInput();
278         }
279     } while (!validateSubDistrict(newNode->data.address.subDistrict));
280     do {
281         cout << "Kabupaten: ";
282         getline(cin, newNode->data.address.district);
283         if (!validateDistrict(newNode->data.address.district)) {
284             errorInput();
285         }
286     } while (!validateDistrict(newNode->data.address.district));
287     do {
288         cout << "Agama (Islam, Kristen, Katolik, Hindu, Buddha, Konhucu): ";
289         getline(cin, newNode->data.religion);
290         if (!validateReligion(newNode->data.religion)) {
291             errorInput();
292         }
293     } while (!validateReligion(newNode->data.religion));
294 }

```

```

200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

403 void delAtHead()
404 {
405     if(head == nullptr){
406         cout << "Data kosong" << endl;
407         return;
408     }
409     Node* temp = head;
410     head = head->next;
411     if(head != nullptr){
412         head->prev = nullptr;
413     }
414     delete temp;
415     cout << "Data berhasil dihapus" << endl;
416     count--;
417 }
418
419 ifstream file;
420 file.open("data.txt");
421 ofstream tempFile;
422 tempFile.open("temp.txt");
423 string line;
424 int linecount = 0;
425 while (getline(file, line)) {
426     if (line.find("X") != string::npos) {
427         linecount++;
428     }
429     if (linecount != 1) {
430         tempFile << line << endl;
431     }
432 }
433 file.close();
434 tempFile.close();
435 remove("data.txt");
436 rename("temp.txt", "data.txt");
437 }
438
439 void delAtTail()
440 {
441     if(head == nullptr){
442         cout << "Linked list kosong" << endl;
443         return;
444     }
445     Node* current = head;
446     while(current->next != nullptr){
447         current = current->next;
448     }
449     if(current->prev == nullptr){
450         head = nullptr;
451     }
452     else {
453         current->prev->next = nullptr;
454     }
455     delete current;
456     cout << "Data berhasil dihapus" << endl;
457     count--;
458 }
459
460 ifstream file;
461 file.open("data.txt");
462 ofstream tempFile;
463 tempFile.open("temp.txt");
464 string line;
465 int linecount = 0;
466 while (getline(file, line)) {
467     if (line.find("X") != string::npos) {
468         linecount++;
469     }
470     if (linecount != count + 1) {
471         tempFile << line << endl;
472     }
473 }
474 file.close();
475 tempFile.close();
476 remove("data.txt");
477 rename("temp.txt", "data.txt");
478 }
479 }
480
481 int main() {
482     Knp knp;
483     int choice;
484     while(true){
485         knp.initialize();
486         cout << "Menu : " << endl;
487         cout << "1. Input Data" << endl;
488         cout << "2. Tampilkan Data" << endl;
489         cout << "3. Delete Data" << endl;
490         cout << "4. Tampilkan " << endl;
491         cin >> choice;
492         system("cls");
493         cin.ignore();
494         switch (choice) {
495             case 1:
496                 knp.input();
497                 cout << "Data berhasil diinput" << endl;
498                 break;
499             case 2:
500                 knp.output();
501                 break;
502             case 3:
503                 while(true){
504                     system("cls");
505                     cout << "Menu : " << endl;
506                     cout << "1. Delete data di depan" << endl;
507                     cout << "2. Delete data di belakang" << endl;
508                     cout << "3. Tampilkan" << endl;
509                     cout << "4. Ulangi" << endl;
510                     cin >> choice;
511                     switch(choice){
512                         case 1:
513                             knp.delAtHead();
514                             break;
515                         case 2:
516                             knp.delAtTail();
517                             break;
518                         case 3:
519                             break;
520                         default:
521                             cout << "Pilihan tidak valid" << endl;
522                             system("pause");
523                             break;
524                     }
525                 }
526                 break;
527             case 4:
528                 return EXIT_SUCCESS;
529             default:
530                 cout << "Pilihan tidak valid" << endl;
531                 break;
532         }
533         system("pause");
534         system("cls");
535     }
536 }

```

Kodingan sesudah diubah ke double link list versi queue

```

1 #include <iostream>
2 using namespace std;
3
4 void errorInput() {
5     cout << "Input tidak valid" << endl;
6 }
7
8 struct Data {
9     string province;
10    string city;
11    string mki;
12    string name;
13    struct born {
14        string place;
15        int date;
16        int month;
17        int year;
18    } born;
19    bool gender;
20    string bloodType;
21
22    struct Address {
23        string address;
24        int neighborhoodAssociation;
25        int communityAssociation;
26        string district;
27        string subdistrict;
28    } address;
29    string religion;
30    string subvillage;
31    string job;
32    string citizenship;
33    string validUntil;
34    struct Created {
35        int date;
36        int month;
37        int year;
38    } created;
39 };
40
41 struct Node {
42     Data data;
43     Node* next;
44     Node* prev;
45 };
46
47 class KMP {
48 private:
49     Node* head = nullptr;
50     Node* tail = nullptr;
51     int count = 0;
52 public:
53     void writeFile(Node* newnode, const string mode) {
54         ofstream file;
55         file.open("data.txt", mode == "front" ? ios::app : ios::trunc);
56         while(newnode != nullptr) {
57             file << "Province" << " " << newnode->data.province << endl;
58             file << "Kota/Kabupaten" << " " << newnode->data.city << endl;
59             file << "Mki" << " " << newnode->data.mki << endl;
60             file << "Nama" << " " << newnode->data.name << endl;
61             file << "Tempat/Tgl Lahir" << " " << newnode->data.born.place << " / " << setfill('0') << setw(2) << newnode->data.born.date << " / " << setfill('0') << setw(2) << newnode->data.born.month << " / " << newnode->data.born.year << endl;
62             file << "Jenis Kelamin" << " " << newnode->data.gender << " / " << setfill('0') << setw(2) << newnode->data.address.district << endl;
63             file << "Jalan" << " " << newnode->data.address.subdistrict << endl;
64             file << "Religion" << " " << setfill('0') << setw(2) << newnode->data.address.neighborhoodAssociation << " / " << setfill('0') << setw(2) << newnode->data.address.communityAssociation << endl;
65             file << "Subvillage" << " " << newnode->data.address.subvillage << endl;
66             file << "Occupation" << " " << newnode->data.job << endl;
67             file << "Agama" << " " << newnode->data.citizenship << endl;
68             file << "Status Perkawinan" << " " << newnode->data.validUntil << endl;
69             file << "Pekerjaan" << " " << newnode->data.job << endl;
70             file << "Kewarganegaraan" << " " << newnode->data.citizenship << endl;
71             file << "Berlaku Hingga" << " " << newnode->data.validUntil << endl;
72             file << "Tanggal" << " " << setfill('0') << setw(2) << newnode->data.created.date << " / " << setfill('0') << setw(2) << newnode->data.created.month << " / " << newnode->data.created.year << endl;
73             file << "-----" << endl;
74             newnode = newnode->next;
75         }
76         file.close();
77     }
78
79     void enqueueFront(Data readData) {
80         Node* newnode = new Node(readData, nullptr, nullptr);
81         if (head == nullptr) {
82             head = newnode;
83             tail = newnode;
84         }
85         else {
86             newnode->next = head;
87             head->prev = newnode;
88             head = newnode;
89         }
90         writeFile(newnode, "front");
91         count++;
92     }
93
94     void enqueueRear(Data readData) {
95         Node* head = nullptr;
96         Node* tail = nullptr;
97         Node* newnode = new Node(readData, nullptr, nullptr);
98         if (head == nullptr) {
99             head = newnode;
100            tail = newnode;
101        }
102        else {
103            newnode->prev = tail;
104            tail->next = newnode;
105            tail = newnode;
106        }
107        writeFile(newnode, "rear");
108        count++;
109    }
110
111     void dequeueFront() {
112         if (head == nullptr) {
113             cout << "Antrian kosong" << endl;
114             return;
115         }
116         Node* current = head;
117         head = head->next;
118         while(current != nullptr) {
119             current = current->next;
120         }
121         if (current == tail) {
122             tail = nullptr;
123         }
124         cout << "Data berhasil dihapus" << endl;
125         count--;
126         writeFile(head, "");
127     }
128
129     void dequeueRear() {
130         if (head == nullptr) {
131             cout << "Antrian kosong" << endl;
132             return;
133         }
134         Node* current = head;
135         while(current->next != nullptr) {
136             current = current->next;
137         }
138         if (current->prev == nullptr) {
139             head = nullptr;
140         }
141         else {
142             current->prev->next = nullptr;
143         }
144         delete current;
145         cout << "Data berhasil dihapus" << endl;
146         count--;
147     }
148
149     ifstream file;
150     file.open("data.txt");
151     ofstream tempFile;
152     tempFile.open("temp.txt");
153     string line;
154     int lineCount = 0;
155     while (getline(file, line)) {
156         if (line.find("-----") != string::npos) {
157             lineCount++;
158         }
159         if (lineCount % 5 == 0) {
160             tempFile << line << endl;
161         }
162     }
163 }

```

```

104     }
105     file.close();
106     tempFile.close();
107     rename("data.txt");
108     rename("temp.txt", "data.txt");
109 }
110
111 bool validateProvince(string province) {
112     return province.length() <= 50;
113 }
114
115 bool validateCity(string city) {
116     return city.length() <= 50;
117 }
118
119 bool validateNIK(string nik) {
120     return nik.length() == 16;
121 }
122
123 bool validateName(string name) {
124     return name.length() <= 50;
125 }
126
127 bool validateBornePlace(string bornePlace) {
128     return bornePlace.length() <= 20;
129 }
130
131 bool validateDate(int date) {
132     return date >= 1 && date <= 31;
133 }
134
135 bool validateMonth(int month) {
136     return month >= 1 && month <= 12;
137 }
138
139 bool validateYear(int year) {
140     return year >= 1900 && year <= 2023;
141 }
142
143 bool validateGender(int gender) {
144     return gender == 0 || gender == 1;
145 }
146
147 bool validateBloodType(string bloodType) {
148     return bloodType == "A" || bloodType == "B" || bloodType == "AB" || bloodType == "O";
149 }
150
151 bool validateAddress(string address) {
152     return address.length() <= 100;
153 }
154
155 bool validateNeighborhoodAssociation(int neighborhoodAssociation) {
156     return neighborhoodAssociation >= 0 && neighborhoodAssociation <= 999;
157 }
158
159 bool validateCommunityAssociation(int communityAssociation) {
160     return communityAssociation >= 0 && communityAssociation <= 999;
161 }
162
163 bool validateSubDistrict(string subDistrict) {
164     return subDistrict.length() <= 20;
165 }
166
167 bool validateDistrict(string district) {
168     return district.length() <= 20;
169 }
170
171 bool validateReligion(string religion) {
172     return religion == "Islam" || religion == "Kristen" || religion == "Katolik" || religion == "Hindu" || religion == "Buddha" || religion == "Konghucu";
173 }
174
175 bool validateMarried(string isMarried) {
176     return isMarried == "Kawin" || isMarried == "Belum Kawin";
177 }
178
179 bool validateJob(string job) {
180     return job.length() <= 50;
181 }
182
183 void input() {
184     Node* newNode = new Node;
185     do {
186         cout << "Provisi: ";
187         getline(cin, newNode->data.province);
188         if (!validateProvince(newNode->data.province)) {
189             errorInput();
190         }
191     } while (!validateProvince(newNode->data.province));
192     do {
193         cout << "Kota: ";
194         getline(cin, newNode->data.city);
195         if (!validateCity(newNode->data.city)) {
196             errorInput();
197         }
198     } while (!validateCity(newNode->data.city));
199     do {
200         cout << "NIK: ";
201         getline(cin, newNode->data.nik);
202         if (!validateNIK(newNode->data.nik)) {
203             errorInput();
204         }
205     } while (!validateNIK(newNode->data.nik));
206     do {
207         cout << "Nama: ";
208         getline(cin, newNode->data.name);
209         if (!validateName(newNode->data.name)) {
210             errorInput();
211         }
212     } while (!validateName(newNode->data.name));
213     do {
214         cout << "Tempat lahir: ";
215         getline(cin, newNode->data.bornePlace);
216         if (!validateBornePlace(newNode->data.bornePlace)) {
217             errorInput();
218         }
219     } while (!validateBornePlace(newNode->data.bornePlace));
220     do {
221         cout << "Tanggal lahir: ";
222         cin >> newNode->data.born.date;
223         if (!validateDate(newNode->data.born.date)) {
224             errorInput();
225         }
226     } while (!validateDate(newNode->data.born.date));
227     do {
228         cout << "Bulan lahir: ";
229         cin >> newNode->data.born.month;
230         if (!validateMonth(newNode->data.born.month)) {
231             errorInput();
232         }
233     } while (!validateMonth(newNode->data.born.month));
234     do {
235         cout << "Tahun lahir: ";
236         cin >> newNode->data.born.year;
237         if (!validateYear(newNode->data.born.year)) {
238             errorInput();
239         }
240     } while (!validateYear(newNode->data.born.year));
241     do {
242         cout << "Jenis kelamin (0=laki-laki, 1=perempuan): ";
243         cin >> newNode->data.gender;
244         if (!validateGender(newNode->data.gender)) {
245             errorInput();
246         }
247     } while (!validateGender(newNode->data.gender));
248     cin.ignore();
249
250     do {
251         cout << "Jalan raya: ";
252         getline(cin, newNode->data.address.address);
253         if (!validateAddress(newNode->data.address.address)) {
254             errorInput();
255         }
256     } while (!validateAddress(newNode->data.address.address));
257     do {
258         cout << "RT: ";
259         cin >> newNode->data.address.neighborhoodAssociation;
260         if (!validateNeighborhoodAssociation(newNode->data.address.neighborhoodAssociation)) {
261             errorInput();
262         }
263     } while (!validateNeighborhoodAssociation(newNode->data.address.neighborhoodAssociation));
264 }

```

1288



```

400 void initialize() {
401     ifstream file;
402     file.open("data.txt");
403     string line;
404     count = 0;
405     while (getline(file, line)) {
406         if (line.find("N/A") != string::npos) {
407             continue;
408         }
409     }
410     cout << "Input Nomor KTP : ";
411     file.close();
412     file.seekg(0, ios::beg);
413     this->head = nullptr;
414     this->tail = nullptr;
415     for (int i = 0; i < count; i++) {
416         data newData;
417         while (getline(file, line)) {
418             if (line.find("Provinsi") != string::npos) newData.province = line.substr(line.find(":") + 1);
419             else if (line.find("Kota/Kabupaten") != string::npos) newData.city = line.substr(line.find(":") + 1);
420             else if (line.find("No") != string::npos) newData.no = line.substr(line.find(":") + 1);
421             else if (line.find("Nama") != string::npos) newData.name = line.substr(line.find(":") + 1);
422             else if (line.find("Tempat/Tgl. lahir") != string::npos) {
423                 newData.born_place = line.substr(line.find(":") + 1, line.find(",") - line.find(":") - 1);
424                 newData.born_date = stoi(line.substr(line.find(":") + 1, 2));
425                 newData.born_month = stoi(line.substr(line.find(":") + 1, 2));
426                 newData.born_year = stoi(line.substr(line.find_last_of(",") + 1));
427             }
428             else if (line.find("Jenis Kelamin") != string::npos) {
429                 newData.gender = line.substr(line.find(":") + 1) == "Laki Laki" ? 0 : 1;
430             }
431             else if (line.find("Gol. Darah") != string::npos) newData.bloodtype = line.substr(line.find(":") + 1);
432             else if (line.find("Alamat") != string::npos) newData.address.address = line.substr(line.find(":") + 1);
433             else if (line.find("No") != string::npos) {
434                 newData.address.neighborhoodassociation = stoi(line.substr(line.find(":") + 1));
435                 newData.address.communityassociation = stoi(line.substr(line.find_last_of(",") + 1));
436             }
437             else if (line.find("Tipe Rumah") != string::npos) newData.address.subdistrict = line.substr(line.find(":") + 1);
438             else if (line.find("Kecamatan") != string::npos) newData.address.district = line.substr(line.find(":") + 1);
439             else if (line.find("Agama") != string::npos) newData.religion = line.substr(line.find(":") + 1);
440             else if (line.find("Tingkat Pendidikan") != string::npos) newData.levelofedu = line.substr(line.find(":") + 1);
441             else if (line.find("Pekerjaan") != string::npos) newData.job = line.substr(line.find(":") + 1);
442             else if (line.find("Nomor pengantar") != string::npos) newData.citizenid = line.substr(line.find(":") + 1);
443             else if (line.find("Berlaku Hingga") != string::npos) newData.validuntil = line.substr(line.find(":") + 1);
444             else if (line.find("Nama") != string::npos) {
445                 newData.created_date = stoi(line.substr(line.find(":") + 1, 2));
446                 newData.created_month = stoi(line.substr(line.find(":") + 1, 2));
447                 newData.created_year = stoi(line.substr(line.find_last_of(",") + 1));
448             }
449             else if (line.find("-----") != string::npos) {
450                 head->next = new Node(newData, nullptr, tail);
451                 if (head == nullptr) {
452                     head = head->next;
453                     tail = head->next;
454                 }
455                 else {
456                     tail->next = head->next;
457                     tail = head->next;
458                 }
459                 break;
460             }
461         }
462     }
463     file.close();
464 }
465
466 int main() {
467     Ktp ktp;
468     int choice;
469     while (true) {
470         ktp.initialize();
471         cout << "Menu : ";
472         cout << "1. Input Data" << endl;
473         cout << "2. Tampilkan Data" << endl;
474         cout << "3. Hapus Data" << endl;
475         cout << "4. Keluar" << endl;
476         cout << "Pilihan : ";
477         cin >> choice;
478         system("cls");
479         cin.ignore();
480         switch (choice) {
481             case 1:
482                 ktp.input();
483                 cout << "Data berhasil diinput" << endl;
484                 break;
485             case 2:
486                 ktp.output();
487                 break;
488             case 3:
489                 ktp.remove();
490                 system("cls");
491                 cout << "Menu : ";
492                 cout << "1. Dequeue Front" << endl;
493                 cout << "2. Dequeue Rear" << endl;
494                 cout << "3. Manual" << endl;
495                 cout << "Pilihan : ";
496                 cin >> choice;
497                 switch (choice) {
498                     case 1:
499                         ktp.dequeueFront();
500                         break;
501                     case 2:
502                         ktp.dequeueRear();
503                         break;
504                     case 3:
505                         break;
506                     default:
507                         cout << "Pilihan tidak valid" << endl;
508                         system("pause");
509                         break;
510                 }
511                 break;
512             case 4:
513                 return EXIT_SUCCESS;
514             default:
515                 cout << "Pilihan tidak valid" << endl;
516                 break;
517         }
518         system("pause");
519         system("cls");
520     }
521 }

```

**Program jalan (Screenshot input output)**

```
Banyak Antrian KTP: 12
Menu:
1. Input Data
2. Tampilkan Data
3. Delete Data
4. Keluar
Pilihan: 
```

Tampilan menu

Mencoba input menggunakan enqueueFront()



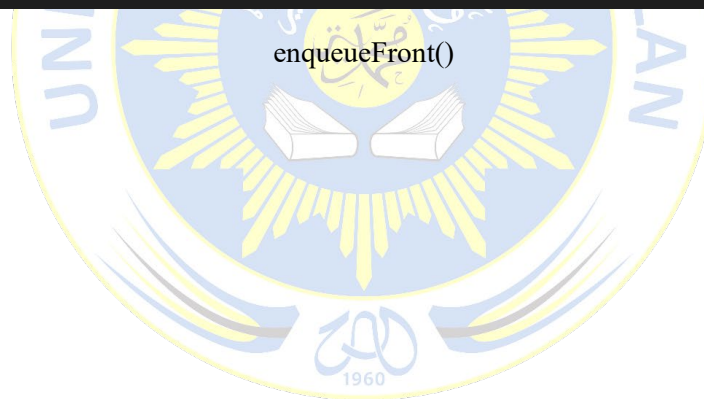
```

=====
KARTU TANDA PENDUDUK
=====
PROVINSI KALIMANTAN TENGAH
KABUPATEN KOTAWARINGIN BARAT
NIK          : 6201052907040001
Nama         : RENDIE ABDI SAPUTRA
Tempat/Tgl Lahir : PANGKALAN BUN, 29-07-2004
Jenis Kelamin  : PEREMPUAN          Gol.Darah : B
Alamat       : JL. A. YANI
RT/RW        : 006/003
Kel/Desa     : SUMBER AGUNG
Kecamatan    : PANGKALAN LADA
Agama       : ISLAM
Status Perkawinan : BELUM KAWIN          Foto.jpg
Pekerjaan    : PELAJAR/MAHASISWA
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP          02-08-2021
                                           TTD
=====
PROVINSI BANTEN
KOTA TANGERANG SELATAN
NIK          : 3674072703040003
Nama         : MOHAMMAD FARID HENDIANTO
Tempat/Tgl Lahir : TANGERANG, 27-03-2004
Jenis Kelamin  : PEREMPUAN          Gol.Darah : B
Alamat       : BATAN INDAH BLOK J-14
RT/RW        : 003/004
Kel/Desa     : KADEMANGAN
Kecamatan    : SETU
Agama       : Islam
Status Perkawinan : Belum Kawin          Foto.jpg
Pekerjaan    : PELAJAR/MAHASISWA
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP          27-05-2021
                                           TTD
=====
PROVINSI DAERAH ISTIMEWA YOGYAKARTA
KOTA YOGYAKARTA
NIK          : 3471135407040001
Nama         : AISYAH SYAFI'I NURJANNAH
Tempat/Tgl Lahir : YOGYAKARTA, 14-07-2004
Jenis Kelamin  : LAKI-LAKI          Gol.Darah : B
Alamat       : GLAGAH UH 4/67
RT/RW        : 012/003
Kel/Desa     : WARUNGBOTO
Kecamatan    : UMBULHARJO
Agama       : ISLAM
Status Perkawinan : BELUM KAWIN          Foto.jpg
Pekerjaan    : PELAJAR/MAHASISWA
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP          24-06-2021
                                           TTD
=====
PROVINSI DAERAH ISTIMEWA YOGYAKARTA
KOTA YOGYAKARTA
NIK          : 3203012503770011
Nama         : GUOHUI CHEN
Tempat/Tgl Lahir : FUJIAN, 25-03-1977
Jenis Kelamin  : PEREMPUAN          Gol.Darah : A
Alamat       : JL. SELAMET PERUMAHAN RANCA BALI NO.40
RT/RW        : 002/004
Kel/Desa     : MUKA
Kecamatan    : CIANJUR
Agama       : KRISTEN
Status Perkawinan : KAWIN          Foto.jpg
Pekerjaan    : OTHERS
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP          11-10-2023
                                           TTD
=====

```

Data sebelum diinputkan, seharusnya data sebelumnya ada Rendie Abdi.

```
Provinsi: PROVINSI JAWA TIMUR
Kota: KABUPATEN BANYUWANGI
NIK: 3510243006730004
Nama: TUHAN
Tempat Lahir: BANYUWANGI
Tanggal Lahir: 30
Bulan Lahir: 6
Tahun Lahir: 1973
Jenis kelamin (0=laki-laki, 1=perempuan): 0
Golongan Darah: AB
Alamat: DUSUN KRAJAN
RT: 1
RW: 2
Kel/Desa: KLUNCING
Kecamatan: LICN
Agama (Islam, Kristen, Katolik, Hindu, Buddha, Konghucu): Islam
Status Perkawinan (Kawin, Belum Kawin): Kawin
Pekerjaan: PNS
Pilih cara input data:
1. enqueue front
2. enqueue rear
Pilihan: 1
```



enqueueFront()

KARTU TANDA PENDUDUK			
PROVINSI JAWA TIMUR			
KABUPATEN BANYUMANGI			
NIK	: 3510243006730004		
Nama	: TUHAN		
Tempat/Tgl Lahir	: BANYUMANGI, 30-06-1973		
Jenis Kelamin	: PEREMPUAN	Gol.Darah : AB	
Alamat	: DUSUN KRAJAN		
RT/RW	: 001/002		
Kel/Desa	: KLUNCING		
Kecamatan	: LICIN		
Agama	: ISLAM		
Status Perkawinan	: KAWIN	Foto.jpg	
Pekerjaan	: PNS		
Kewarganegaraan	: WNI	28-11-2023	
Berlaku Hingga	: SEUMUR HIDUP	TTD	
PROVINSI KALIMANTAN TENGAH			
KABUPATEN KOTAWARINGIN BARAT			
NIK	: 6201052907040001		
Nama	: RENDIE ABDI SAPUTRA		
Tempat/Tgl Lahir	: PANGKALAN BUN, 29-07-2004		
Jenis Kelamin	: PEREMPUAN	Gol.Darah : B	
Alamat	: JL. A. YANI		
RT/RW	: 006/003		
Kel/Desa	: SUMBER AGUNG		
Kecamatan	: PANGKALAN LADA		
Agama	: ISLAM		
Status Perkawinan	: BELUM KAWIN	Foto.jpg	
Pekerjaan	: PELAJAR/MAHASISWA		
Kewarganegaraan	: WNI	02-08-2021	
Berlaku Hingga	: SEUMUR HIDUP	TTD	
PROVINSI BANTEN			
KOTA TANGERANG SELATAN			

Berhasil terinput sebelum data Rendie Abdi

enqueueRear()

Sebelum itu, akan melihat data yang paling bawah

PROVINSI JAWA TENGAH  
KABUPATEN KUDUS

NIK : 3319042612640001  
 Nama : BUDI SANTOSO  
 Tempat/Tgl Lahir : KUDUS, 29-12-1984  
 Jenis Kelamin : PEREMPUAN Gol.Darah : B  
 Alamat : LARIREJO  
 RT/RW : 002/001  
 Kel/Desa : LARIREJO  
 Kecamatan : UNDAAN  
 Agama : ISLAM  
 Status Perkawinan : BELUM KAWIN Foto.jpg  
 Pekerjaan : PETANI/PEKEBUN  
 Kewarganegaraan : WNI 17-10-2023  
 Berlaku Hingga : SEUMUR HIDUP TTD

=====

PROVINSI JAWA TENGAH  
KOTA SURAKARTA

NIK : 3372011210770001  
 Nama : SELAMAT IDUL ADHA  
 Tempat/Tgl Lahir : SURAKARTA, 12-10-1977  
 Jenis Kelamin : PEREMPUAN Gol.Darah : AB  
 Alamat : BARON CILIK  
 RT/RW : 004/005  
 Kel/Desa : BUMI  
 Kecamatan : LAWEYAN  
 Agama : ISLAM  
 Status Perkawinan : KAWIN Foto.jpg  
 Pekerjaan : PERDAGANGAN  
 Kewarganegaraan : WNI 17-10-2023  
 Berlaku Hingga : SEUMUR HIDUP TTD

=====

PROVINSI SULAWESI SELATAN  
KABUPATEN SOPPENG

NIK : 7312042510720002  
 Nama : ABDURRAUF, S.Pd, M.Pd  
 Tempat/Tgl Lahir : CELLENGENGE, 25-10-1972  
 Jenis Kelamin : PEREMPUAN Gol.Darah : O  
 Alamat : JL. MERDEKA NO.43  
 RT/RW : 001/004  
 Kel/Desa : BILA  
 Kecamatan : LALABATA  
 Agama : ISLAM  
 Status Perkawinan : KAWIN Foto.jpg  
 Pekerjaan : PEGAWAI NEGERI SIPIL  
 Kewarganegaraan : WNI 17-10-2023  
 Berlaku Hingga : SEUMUR HIDUP TTD

=====

Press any key to continue . . .

1900

```

Provinsi: PROVINSI JAWA TIMUR
Kota: KABUPATEN BANYUWANGI
NIK: 3510243006730004
Nama: TUHAN
Tempat Lahir: BANYUWANGI
Tanggal Lahir: 30
Bulan Lahir: 6
Tahun Lahir: 1973
Jenis kelamin (0=laki-laki, 1=perempuan): 0
Golongan Darah: AB
Alamat: DUSUN KRAJAN
RT: 1
RW: 2
Kel/Desa: KLUNCING
Kecamatan: LICN
Agama (Islam, Kristen, Katolik, Hindu, Buddha, Konghucu): Islam
Status Perkawinan (Kawin, Belum Kawin): Kawin
Pekerjaan: PNS
Pilih cara input data:
1. enqueue front
2. enqueue rear
Pilihan: 2
    
```

Input data untuk persiapan input enqueue rear

Hasil setelah di input antrian di rear



```

Berlaku Hingga : SEUMUR HIDUP
=====
PROVINSI SULAWESI SELATAN
KABUPATEN SOPPENG
NIK          : 7312042510720002
Nama         : ABDURRAUF, S.Pd, M.Pd
Tempat/Tgl Lahir : CELLENGENGE, 25-10-1972
Jenis Kelamin  : PEREMPUAN           Gol.Darah : O
Alamat        : JL. MERDEKA NO.43
RT/RW         : 001/004
Kel/Desa      : BILA
Kecamatan     : LALABATA
Agama         : ISLAM
Status Perkawinan : KAWIN           Foto.jpg
Pekerjaan     : PEGAWAI NEGERI SIPIL
Kewarganegaraan : WNI              17-10-2023
Berlaku Hingga : SEUMUR HIDUP      TTD
=====
PROVINSI JAWA TIMUR
KABUPATEN BANYUWANGI
NIK          : 3510243006730004
Nama         : TUHAN
Tempat/Tgl Lahir : BANYUWANGI, 30-06-1973
Jenis Kelamin  : PEREMPUAN           Gol.Darah : AB
Alamat        : DUSUN KRAJAN
RT/RW         : 001/002
Kel/Desa      : KLUNCING
Kecamatan     : LICN
Agama         : ISLAM
Status Perkawinan : KAWIN           Foto.jpg
Pekerjaan     : PNS
Kewarganegaraan : WNI              28-11-2023
Berlaku Hingga : SEUMUR HIDUP      TTD
=====
Press any key to continue . . .

```

dequeueFront()





KARTU TANDA PENDUDUK			
=====			
PROVINSI KALIMANTAN TENGAH			
KABUPATEN KOTAWARINGIN BARAT			
NIK	: 6201052907040001		
Nama	: RENDIE ABDI SAPUTRA		
Tempat/Tgl Lahir	: PANGKALAN BUN, 29-07-2004		
Jenis Kelamin	: PEREMPUAN	Gol.Darah : B	
Alamat	: JL. A. YANI		
RT/RW	: 006/003		
Kel/Desa	: SUMBER AGUNG		
Kecamatan	: PANGKALAN LADA		
Agama	: ISLAM		
Status Perkawinan	: BELUM KAWIN	Foto.jpg	
Pekerjaan	: PELAJAR/MAHASISWA		
Kewarganegaraan	: WNI	02-08-2021	
Berlaku Hingga	: SEUMUR HIDUP	TTD	
=====			
PROVINSI BANTEN			
KOTA TANGERANG SELATAN			
NIK	: 3674072703040003		
Nama	: MOHAMMAD FARID HENDIANTO		
Tempat/Tgl Lahir	: TANGERANG, 27-03-2004		
Jenis Kelamin	: PEREMPUAN	Gol.Darah : B	
Alamat	: BATAN INDAH BLOK J-14		
RT/RW	: 003/004		
Kel/Desa	: KADEMANGAN		
Kecamatan	: SETU		
Agama	: Islam		
Status Perkawinan	: Belum Kawin	Foto.jpg	
Pekerjaan	: PELAJAR/MAHASISWA		
Kewarganegaraan	: WNI	27-05-2021	
Berlaku Hingga	: SEUMUR HIDUP	TTD	
=====			
PROVINSI DAERAH ISTIMEWA YOGYAKARTA			
KOTA YOGYAKARTA			
NIK	: 3471135407040001		
Nama	: AISYAH SYAFI'I NURJANNAH		
Tempat/Tgl Lahir	: YOGYAKARTA, 14-07-2004		
Jenis Kelamin	: LAKI-LAKI	Gol.Darah : B	
Alamat	: GLAGAH UH 4/67		
RT/RW	: 012/003		
Kel/Desa	: WARUNGBOTO		

Misalnya data sebelumnya seperti ini. Kita akan mencoba dequeue pada front sebanyak 2 x

```
Menu:
1. Dequeue Front
2. Dequeue Rear
3. Kembali
Pilihan: 1
Data berhasil dihapus
Press any key to continue . . .
```

1

```
Menu:
1. Dequeue Front
2. Dequeue Rear
3. Kembali
Pilihan: 1
Data berhasil dihapus
Press any key to continue . . .
```

2

Seharusnya berada di aisyah



KARTU TANDA PENDUDUK			
PROVINSI DAERAH ISTIMEWA YOGYAKARTA			
KOTA YOGYAKARTA			
NIK	: 3471135407040001		
Nama	: AISYAH SYAFI'I NURJANNAH		
Tempat/Tgl Lahir	: YOGYAKARTA, 14-07-2004		
Jenis Kelamin	: LAKI-LAKI	Gol.Darah : B	
Alamat	: GLAGAH UH 4/67		
RT/RW	: 012/003		
Kel/Desa	: WARUNGBOTO		
Kecamatan	: UMBULHARJO		
Agama	: ISLAM		
Status Perkawinan	: BELUM KAWIN	Foto.jpg	
Pekerjaan	: PELAJAR/MAHASISWA		
Kewarganegaraan	: WNI	24-06-2021	
Berlaku Hingga	: SEUMUR HIDUP	TTD	
PROVINSI DAERAH ISTIMEWA YOGYAKARTA			
KOTA YOGYAKARTA			
NIK	: 3203012503770011		
Nama	: GUOHUI CHEN		
Tempat/Tgl Lahir	: FUJIAN, 25-03-1977		
Jenis Kelamin	: PEREMPUAN	Gol.Darah : A	
Alamat	: JL. SELAMET PERUMAHAN RANCABALI NO.40		
RT/RW	: 002/004		
Kel/Desa	: MUKA		
Kecamatan	: CIANJUR		
Agama	: KRISTEN		
Status Perkawinan	: KAWIN	Foto.jpg	
Pekerjaan	: OTHERS		
Kewarganegaraan	: WNI	11-10-2023	
Berlaku Hingga	: SEUMUR HIDUP	TTD	
PROVINSI DKI JAKARTA			
JAKARTA BARAT			
NIK	: 3171234567890123		
Nama	: MIRA SETIAWAN		
Tempat/Tgl Lahir	: JAKARTA, 18-02-1986		
Jenis Kelamin	: LAKI-LAKI	Gol.Darah : B	
Alamat	: JL. PASTI CEPAT A7/66		
RT/RW	: 007/008		
Kel/Desa	: PEGADUNGAN		

Data yang sudah dequeue sebanyak 2x pada front

dequeueRear()

```

=====
PROVINSI JAWA TENGAH
KABUPATEN KUDUS
NIK          : 3319042612640001
Nama         : BUDI SANTOSO
Tempat/Tgl Lahir : KUDUS, 29-12-1984
Jenis Kelamin  : PEREMPUAN          Gol.Darah : B
Alamat       : LARIREJO
RT/RW        : 002/001
Kel/Desa     : LARIREJO
Kecamatan    : UNDAAN
Agama       : ISLAM
Status Perkawinan : BELUM KAWIN          Foto.jpg
Pekerjaan    : PETANI/PEKEBUN
Kewarganegaraan : WNI                  17-10-2023
Berlaku Hingga : SEUMUR HIDUP          TTD
=====
PROVINSI JAWA TENGAH
KOTA SURAKARTA
NIK          : 3372011210770001
Nama         : SELAMAT IDUL ADHA
Tempat/Tgl Lahir : SURAKARTA, 12-10-1977
Jenis Kelamin  : PEREMPUAN          Gol.Darah : AB
Alamat       : BARON CILIK
RT/RW        : 004/005
Kel/Desa     : BUMI
Kecamatan    : LAWEYAN
Agama       : ISLAM
Status Perkawinan : KAWIN          Foto.jpg
Pekerjaan    : PERDAGANGAN
Kewarganegaraan : WNI                  17-10-2023
Berlaku Hingga : SEUMUR HIDUP          TTD
=====
PROVINSI SULAWESI SELATAN
KABUPATEN SOPPENG
NIK          : 7312042510720002
Nama         : ABDURRAUF, S.Pd, M.Pd
Tempat/Tgl Lahir : CELLENGENGE, 25-10-1972
Jenis Kelamin  : PEREMPUAN          Gol.Darah : O
Alamat       : JL. MERDEKA NO.43
RT/RW        : 001/004
Kel/Desa     : BILA
Kecamatan    : LALABATA
Agama       : ISLAM
Status Perkawinan : KAWIN          Foto.jpg
Pekerjaan    : PEGAWAI NEGERI SIPIL
Kewarganegaraan : WNI                  17-10-2023
Berlaku Hingga : SEUMUR HIDUP          TTD
=====
Press any key to continue . . .

```

Mencoba dequeueRear() sebanyak 1 kali

```
Menu:
1. Dequeue Front
2. Dequeue Rear
3. Kembali
Pilihan: 2
Data berhasil dihapus
Press any key to continue . . .
```

```

Kewarganegaraan : WNI                                17-10-2023
Berlaku Hingga   : SEUMUR HIDUP                        TTD
=====
                        PROVINSI JAWA TENGAH
                        KABUPATEN KUDUS
NIK               : 3319042612640001
Nama              : BUDI SANTOSO
Tempat/Tgl Lahir  : KUDUS, 29-12-1984
Jenis Kelamin     : PEREMPUAN                        Gol.Darah : B
Alamat           : LARIREJO
RT/RW            : 002/001
Kel/Desa         : LARIREJO
Kecamatan        : UNDAAN
Agama            : ISLAM
Status Perkawinan: BELUM KAWIN                        Foto.jpg
Pekerjaan        : PETANI/PEKEBUN
Kewarganegaraan  : WNI                                17-10-2023
Berlaku Hingga   : SEUMUR HIDUP                        TTD
=====
                        PROVINSI JAWA TENGAH
                        KOTA SURAKARTA
NIK               : 3372011210770001
Nama              : SELAMAT IDUL ADHA
Tempat/Tgl Lahir  : SURAKARTA, 12-10-1977
Jenis Kelamin     : PEREMPUAN                        Gol.Darah : AB
Alamat           : BARON CILIK
RT/RW            : 004/005
Kel/Desa         : BUMI
Kecamatan        : LAWEYAN
Agama            : ISLAM
Status Perkawinan: KAWIN                        Foto.jpg
Pekerjaan        : PERDAGANGAN
Kewarganegaraan  : WNI                                17-10-2023
Berlaku Hingga   : SEUMUR HIDUP                        TTD
=====
Press any key to continue . . .

```

Yang di bawah menjadi Idul Adha

### Operasi menambah data (enqueue)

```

82 void enqueueFront(Data newData) {
83     Node* newNode = new Node{ newData, nullptr, nullptr };
84     if (head == nullptr) {
85         head = newNode;
86         tail = newNode;
87     }
88     else {
89         newNode->next = head;
90         head->prev = newNode;
91         head = newNode;
92     }
93     writeToFile(newNode, "front");
94     count++;
95 }
96 void enqueueRear(Data newData) {
97     Node* head = nullptr;
98     Node* tail = nullptr;
99
100     Node* newNode = new Node{ newData, nullptr, nullptr };
101     if (head == nullptr) {
102         head = newNode;
103         tail = newNode;
104         newNode->prev = nullptr;
105     } else {
106         tail->next = newNode;
107         newNode->prev = tail;
108         tail = newNode;
109     }
110     writeToFile(newNode, "rear");
111     count++;
112 }

```

Bisa menambahkan dari Front maupun Rear karena merupakan doubly linked list.

Fungsi-fungsi tersebut adalah untuk melakukan penambahan node baru pada sebuah linked list doubly.

`void enqueueFront(Data newData)` digunakan untuk menambahkan node baru di depan (head) linked list. Caranya:

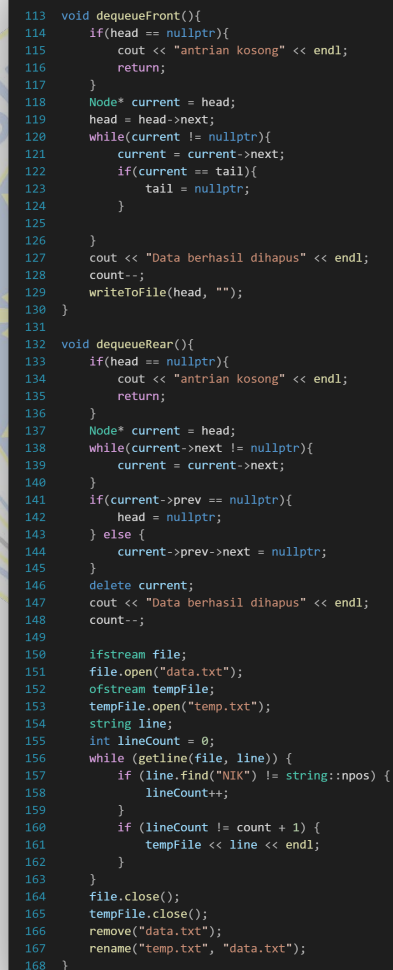
- Buat node baru dengan data `newData`
- Jika linked list masih kosong, jadikan node baru sebagai head sekaligus tail
- Jika tidak kosong, hubungkan node baru dengan head existing, dan update head dengan node baru
- Tulis data node baru ke file
- Tambah count

`void enqueueRear(Data newData)` digunakan untuk menambahkan node baru di belakang (tail) linked list. Caranya:

- Buat node baru dengan data newData
- Jika linked list masih kosong, jadikan node baru sebagai head sekaligus tail
- Jika tidak kosong, hubungkan tail existing dengan node baru, dan update tail dengan node baru
- Tulis data node baru ke file
- Tambah count

Jadi kedua fungsi tersebut sama-sama menambahkan node baru, hanya saja enqueueFront menambahkannya di depan sedangkan enqueueRear di belakang.

### Operasi menghapus data (dequeue)



```

113 void dequeueFront(){
114     if(head == nullptr){
115         cout << "antrian kosong" << endl;
116         return;
117     }
118     Node* current = head;
119     head = head->next;
120     while(current != nullptr){
121         current = current->next;
122         if(current == tail){
123             tail = nullptr;
124         }
125     }
126     cout << "Data berhasil dihapus" << endl;
127     count--;
128     writeToFile(head, "");
129 }
130
131 void dequeueRear(){
132     if(head == nullptr){
133         cout << "antrian kosong" << endl;
134         return;
135     }
136     Node* current = head;
137     while(current->next != nullptr){
138         current = current->next;
139     }
140     if(current->prev == nullptr){
141         head = nullptr;
142     } else {
143         current->prev->next = nullptr;
144     }
145     delete current;
146     cout << "Data berhasil dihapus" << endl;
147     count--;
148
149     ifstream file;
150     file.open("data.txt");
151     ofstream tempFile;
152     tempFile.open("temp.txt");
153     string line;
154     int lineCount = 0;
155     while (getline(file, line)) {
156         if (line.find("NIK") != string::npos) {
157             lineCount++;
158         }
159         if (lineCount != count + 1) {
160             tempFile << line << endl;
161         }
162     }
163     file.close();
164     tempFile.close();
165     remove("data.txt");
166     rename("temp.txt", "data.txt");
167 }
168

```

Bisa menghapus dari Front maupun Rear karena merupakan doubly linked list.

Fungsi `dequeueFront` digunakan untuk menghapus node pada bagian depan (head) linked list. Sedangkan fungsi `dequeueRear` digunakan untuk menghapus node pada bagian belakang (tail) linked list.

Cara kerjanya:

`dequeueFront`:

- Cek apakah head null, jika iya berarti list kosong
- Simpan head di variabel current
- Geser head ke node berikutnya
- Hapus current node
- Kurangi count

`dequeueRear`:

- Cek apakah head null, jika iya berarti list kosong
- Telusuri dari head sampai ketemu tail
- Lepaskan tail dari node sebelumnya
- Hapus tail node
- Kurangi count

Jadi intinya keduanya digunakan untuk menghapus node, hanya saja `dequeueFront` menghapus dari depan dan `dequeueRear` dari belakang.

## Class

Sudah terlihat jelas bahwa menggunakan class `KTP`

```
class KTP {
private:
    Node* head = nullptr;
    Node* tail = nullptr;
    int count = 0;

public:
    void writeToFile(Node* newNode, const string& mode) {
        ... (Isi kodingan lainnya, dapat dilihat di source code) ...
        .....
    };
};
```



Untuk mengakses kodingan dapat dilihat link github berikut pada folder praktikum pertemuan 7

<https://github.com/IRedDragonICY/Data-Structure>

