

## **UTS 2 SDNL**

**“Aplikasi Binary Search Tree dalam Penyusunan dan Pencarian Kata pada Kamus Bahasa Indonesia”**



**DISUSUN OLEH :**

205314006 Josephine Diva

205314013 Lorenzo Ananda

205314020 FX. Bima YP.

205314024 Glorious Aldo P.J

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA**

**2022**

## A. LATAR BELAKANG

Bahasa adalah salah satu komponen penting dalam kehidupan manusia, baik dalam faktor sosial maupun pengembangan diri. Dengan adanya bahasa, kita dapat membangun komunikasi dengan orang lain, mengutarakan pendapat serta berkarya. Sebagai alat komunikasi, bahasa meliputi kata, kumpulan kata, klausa dan kalimat yang diungkapkan secara lisan maupun tulisan (tirto.id). Kata adalah unsur paling dasar dalam penggunaan bahasa. Tanpa adanya kata, bahasa tidak dapat disusun dan dikembangkan. Begitu juga ketika kata disalahartikan, maka akan terjadi ketidaksinambungan antara hal yang ingin disampaikan dengan apa yang tertulis atau dikatakan. Dari permasalahan tersebut, kumpulan kata beserta artinya sangatlah dibutuhkan agar suatu bahasa dapat digunakan dengan baik dan benar. Maka, kamus, buku yang memuat kumpulan istilah atau nama yang disusun menurut abjad beserta penjelasan tentang makna dan pemakaiannya, dapat menjadi salah satu solusi (kbbi.web.id).

Akhir-akhir ini, kamus telah berkembang, baik secara bentuk maupun fungsinya. Dari segi bentuk, kamus telah banyak didigitalisasi menjadi sebuah aplikasi yang lebih mudah digunakan dari berbagai golongan usia. Untuk fungsi dari kamus sendiri, kecepatan pencarian arti sangat dibutuhkan untuk menunjang kebutuhan pengguna. Selain itu, kini kamus dapat terus diperbarui baik oleh pengguna maupun admin asli, yang di mana aplikasi kamus ini merupakan aplikasi open source. Hal tersebut dapat dilakukan dengan mengimplementasikan konsep *binary search tree* dalam penyusunan tiap kata di dalam kamus. Di mana penyusunan tersebut tetap dapat dilakukan secara urut dan pencarian kata dapat dilakukan dengan lebih cepat dibandingkan dengan kamus cetak yang membutuhkan waktu lebih lama.

Untuk dapat terus memperluas penggunaan kata dalam bahasa Indonesia yang baik dan benar, maka penulis merancang sebuah aplikasi yang berjudul “Aplikasi Binary Search Tree dalam Penyusunan dan Pencarian Kata pada Kamus Bahasa Indonesia” yang diharapkan dapat mempermudah dalam pencarian kata serta arti bahasa Indonesia.

## B. TUJUAN

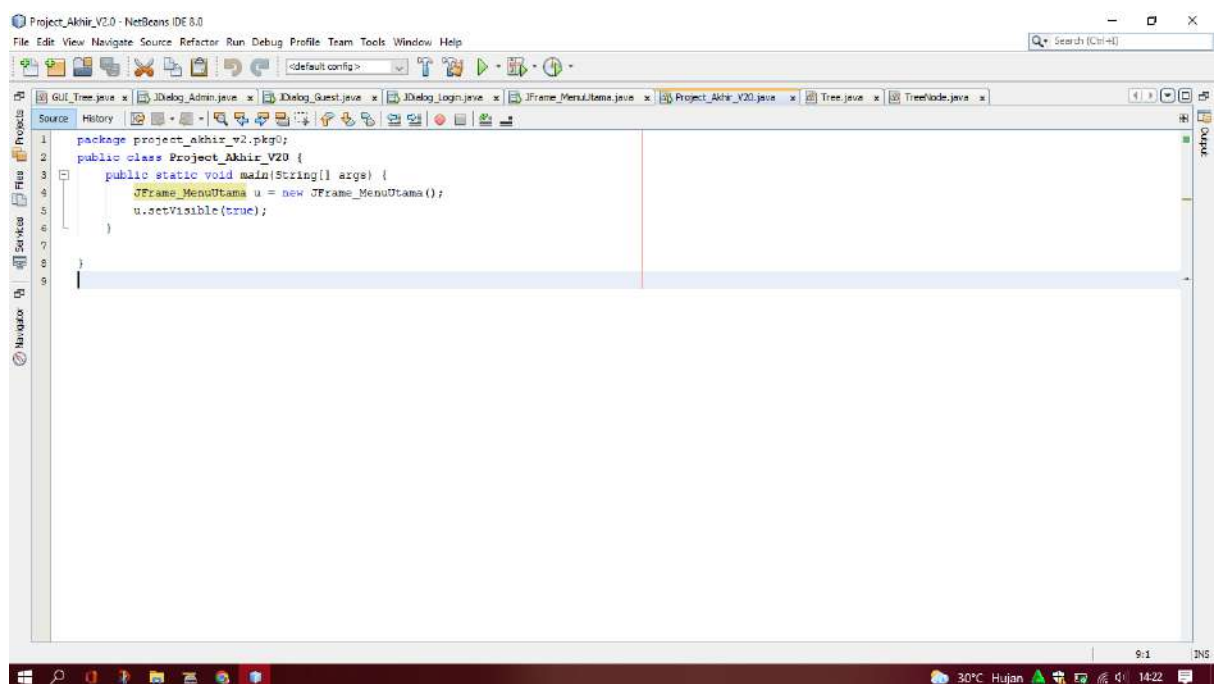
- a. Penyimpanan data berupa kata beserta artinya berbasis komputerisasi.
- b. Mempermudah dalam pencarian kata beserta artinya untuk pengguna.
- c. Mengurutkan kata dengan menggunakan algoritma binary search tree berdasarkan kata yang dimasukkan.
- d. Menambah pengetahuan terkait arti dalam kata bahasa Indonesia bagi pengguna.

## C. MANFAAT

- a. Membantu pengguna menyelesaikan kendala terkait pencarian kata beserta artinya untuk kebutuhan akademis maupun sosial.
- b. Dapat menjadi referensi bagi pihak-pihak yang berkepentingan tentang pengertian kata dalam bahasa Indonesia.
- c. Sarana untuk menambah pengetahuan dan wawasan untuk penggunaan teori teori yang dipelajari dalam bangku perkuliahan.

## D. SCREENSHOT CODE

### 1. Class Project\_Akhir\_V20



### 2. Class JFrame\_MenuUtama



```

1 package project_akhir_v2.pkg0;
2 import javax.swing.JOptionPane;
3 public class JDIALOG_Guest extends javax.swing.JDialog {
4     GUI_Tree phn;
5     TreeNode node;
6     public JDIALOG_Guest(java.awt.Frame parent, boolean modal, GUI_Tree data) {
7         super(parent, modal);
8         initComponents();
9         this.phn = data;
10    }
11    @SuppressWarnings("unchecked")
12    Generated Code
13
14    private void btn_cariActionPerformed(java.awt.event.ActionEvent evt) {
15        String temp = txtKata.getText().toUpperCase();
16        try {
17            if (temp == null) {
18                throw new Exception("Kolom Kosakata Tidak Boleh Kosong");
19            }
20            node = this.phn.Search_Node(temp);
21            if (node == null) {
22                throw new Exception("Tidak ada data silahkan menambah data terlebih dahulu " + "\n"
23                    + "atau kembali ke menu utama untuk menambahkan data secara otomatis");
24            } else {
25                txtKata.setText(node.getData());
26                txtArti.setText(node.getArti());
27                JOptionPane.showMessageDialog(this, "Kata ditemukan!", "INFORMATION",
28                    JOptionPane.INFORMATION_MESSAGE);
29            }
30        }
31    }
32 }

```

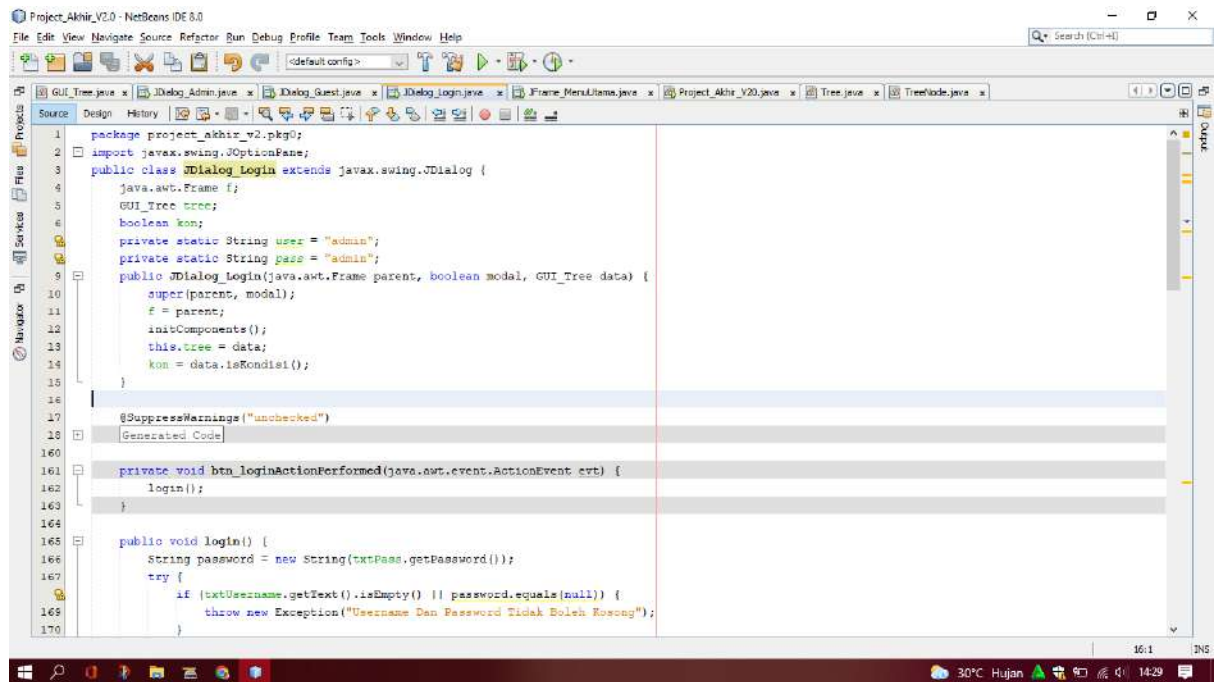
```

157 txtKata.setText(node.getData());
158 txtArti.setText(node.getArti());
159 JOptionPane.showMessageDialog(this, "Kata ditemukan!", "INFORMATION",
160     JOptionPane.INFORMATION_MESSAGE);
161 }
162 catch (Exception e) {
163     JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
164         JOptionPane.WARNING_MESSAGE);
165 }
166 }
167
168 public static void main(String args[]) {
169     // Variables declaration - do not modify
170     private javax.swing.JButton btn_cari;
171     private javax.swing.JLabel jLabel1;
172     private javax.swing.JPanel jPanel1;
173     private javax.swing.JPanel jPanel2;
174     private javax.swing.JPanel jPanel3;
175     private javax.swing.JPanel jPanel4;
176     private javax.swing.JScrollPane jScrollPane1;
177     private javax.swing.JTextArea txtArti;
178     private javax.swing.JTextField txtKata;
179     // End of variables declaration
180 }
181

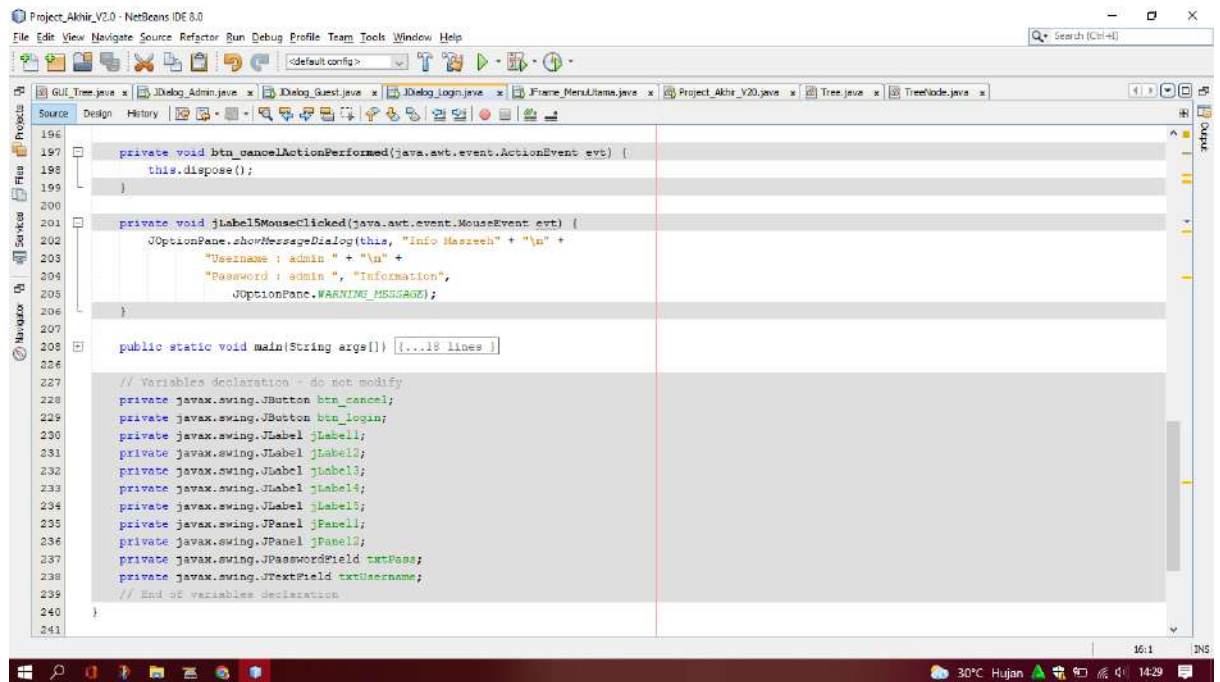
```

#### 4. Class JDIALOG\_Login

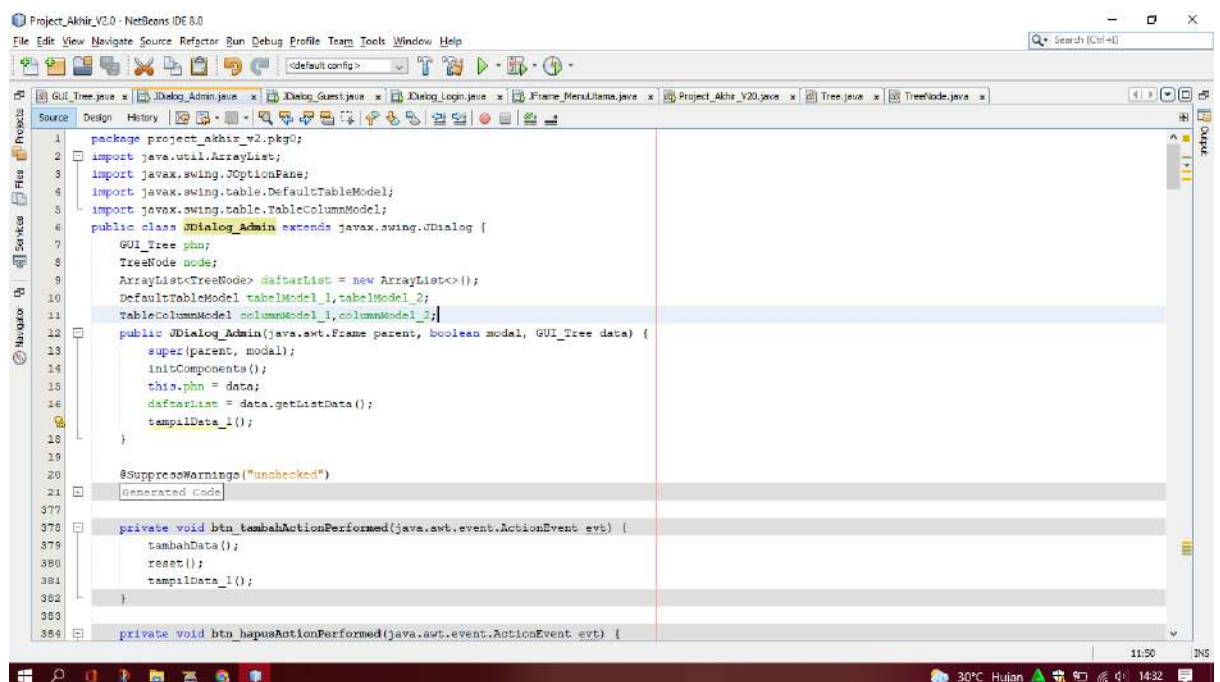




```
1 package project_akhir_v2.pkg0;
2 import javax.swing.JOptionPane;
3 public class JDialog_Login extends javax.swing.JDialog {
4     java.awt.Frame f;
5     GUI_Tree tree;
6     boolean kon;
7     private static String user = "admin";
8     private static String pass = "admin";
9     public JDialog_Login(java.awt.Frame parent, boolean modal, GUI_Tree data) {
10         super(parent, modal);
11         f = parent;
12         initComponents();
13         this.tree = data;
14         kon = data.isKondisi();
15     }
16
17     @SuppressWarnings("unchecked")
18     Generated Code
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
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
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
```



## 5. Class JDialog\_Admin



```
Project_Akhir_V2.0 - NetBeans IDE 8.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

GUI_Tree.java x JDialog_Admin.java x JDialog_Guest.java x JDialog_Login.java x JFrame_MenuUtama.java x Project_Akhir_V20.java x Tree.java x TreeNode.java x

Source Design History
private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
    int jawab = JOptionPane.showOptionDialog(this,
        "Anda Yakin Ingin Menghapus Kata Ini Dari Tree?",
        "Batal",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, null, null);

    String temp = txtKata.getText().toUpperCase();

    if (jawab == JOptionPane.YES_OPTION) {
        try {
            if (txtKata.getText().isEmpty()) {
                throw new Exception("Kolom Kosokata Masih Kosong");
            }
            node = this.phn.Search_Node(temp);
            if (node == this.phn.getRoot()) {
                throw new Exception("Root Tidak Dapat Di Hapus \n "
                    + "Silahkan Tambah satu Data Jika Ingin Menghapus "
                    + "Data yang ada di \"Root\"");
            } else {
                this.phn.hapus(node);
            }
            reset();
            jTable1.clearSelection();
            tampilData_1();
            JOptionPane.showMessageDialog(this, "Kata terhapus!", "INFORMATION",
                JOptionPane.INFORMATION_MESSAGE);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
                JOptionPane.WARNING_MESSAGE);
        }
    }
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    reset();
    int selectedRow = jTable1.getSelectedRow();
    node = this.phn.Search_Node(jTableModel_1.getValueAt(selectedRow, 1).toString());
    txtKata.setText(node.getData());
    txtArt1.setText(node.getArt1());

    btn_hapus.setEnabled(true);
    btn_edit.setEnabled(true);
}

private void btn_clearActionPerformed(java.awt.event.ActionEvent evt) {
    reset();
    jTable1.clearSelection();
}

private void btn_cariActionPerformed(java.awt.event.ActionEvent evt) {
    String temp = txtKata.getText().toUpperCase();
    try {
        if (temp == null) {
            JOptionPane.showMessageDialog(this, "Kata terhapus!", "INFORMATION",
                JOptionPane.INFORMATION_MESSAGE);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
            JOptionPane.WARNING_MESSAGE);
    }
}
```

```
Project_Akhir_V2.0 - NetBeans IDE 8.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

GUI_Tree.java x JDialog_Admin.java x JDialog_Guest.java x JDialog_Login.java x JFrame_MenuUtama.java x Project_Akhir_V20.java x Tree.java x TreeNode.java x

Source Design History
JOptionPane.showMessageDialog(this, "Kata terhapus!", "INFORMATION",
    JOptionPane.INFORMATION_MESSAGE);
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
        JOptionPane.WARNING_MESSAGE);
}
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    reset();
    int selectedRow = jTable1.getSelectedRow();
    node = this.phn.Search_Node(jTableModel_1.getValueAt(selectedRow, 1).toString());
    txtKata.setText(node.getData());
    txtArt1.setText(node.getArt1());

    btn_hapus.setEnabled(true);
    btn_edit.setEnabled(true);
}

private void btn_clearActionPerformed(java.awt.event.ActionEvent evt) {
    reset();
    jTable1.clearSelection();
}

private void btn_cariActionPerformed(java.awt.event.ActionEvent evt) {
    String temp = txtKata.getText().toUpperCase();
    try {
        if (temp == null) {
            JOptionPane.showMessageDialog(this, "Kata terhapus!", "INFORMATION",
                JOptionPane.INFORMATION_MESSAGE);
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
            JOptionPane.WARNING_MESSAGE);
    }
}
```



```
Project_Akhir_V2.0 - NetBeans IDE 8.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

GUI_Tree.java x JDialog_Admin.java x JDialog_Guest.java x JDialog_Login.java x JFrame_MenuUtama.java x Project_Akhir_V20.java x Tree.java x TreeNode.java x

Source Design History
437 try {
438     if (temp == null) {
439         throw new Exception("Kolom Kosakata Tidak Boleh Kosong");
440     }
441     node = this.phn.Search_Node(temp);
442     if (node == null) {
443         throw new Exception("Tidak ada data silahkan menambah data terlebih dahulu " + "\n"
444             + "Atau kembali ke menu utama untuk menambahkan data secara otomatis");
445     } else {
446         txtKata.setText(node.getData());
447         txtArti.setText(node.getArti());
448         btn_hapus.setEnabled(true);
449         btn_edit.setEnabled(true);
450         JOptionPane.showMessageDialog(this, "Data ditemukan!", "INFORMATION",
451             JOptionPane.INFORMATION_MESSAGE);
452     }
453 } catch (Exception e) {
454     JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
455         JOptionPane.WARNING_MESSAGE);
456 }
457
458
459 private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
460     int jawab = JOptionPane.showOptionDialog(this,
461         "Anda yakin ingin mengupdate arti dari kata ini?",
462         "Batal",
463         JOptionPane.YES_NO_OPTION,
464         JOptionPane.QUESTION_MESSAGE, null, null, null);
465 }
```

```
Project_Akhir_V2.0 - NetBeans IDE 8.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

GUI_Tree.java x JDialog_Admin.java x JDialog_Guest.java x JDialog_Login.java x JFrame_MenuUtama.java x Project_Akhir_V20.java x Tree.java x TreeNode.java x

Source Design History
JOptionPane.QUESTION_MESSAGE, null, null, null);
464
465 String temp = txtKata.getText().toUpperCase();
466 String tempArti = txtArti.getText();
467
468 if (jawab == JOptionPane.YES_OPTION) {
469     try {
470         if (txtKata.getText().isEmpty()) {
471             throw new Exception("Kolom Kosakata Masih Kosong");
472         }
473         this.node = this.phn.Search_Node(temp);
474         if (node == null) {
475             throw new Exception("Tidak ada kosakata yang sama dengan field kosakata "
476                 + "\n"
477                 + "Data arti dari kosakata tidak dapat di Update");
478         } else {
479             node.setData(temp);
480             node.setArti(tempArti);
481         }
482         reset();
483         jTable1.clearSelection();
484         tempIdata_1();
485         JOptionPane.showMessageDialog(this, "Data arti kosakata terupdate!", "INFORMATION",
486             JOptionPane.INFORMATION_MESSAGE);
487     } catch (Exception e) {
488         JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
489             JOptionPane.WARNING_MESSAGE);
490     }
491 }
492 }
```

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Source Design History

GUI\_Tree.java x JDialog\_Admin.java x JDialog\_Guest.java x JDialog\_Login.java x JFrame\_MenuUtama.java x Project\_Akhir\_V20.java x Tree.java x TreeNode.java x

```
491     }
492 }
493
494
495 private void btn_inActionPerformed(java.awt.event.ActionEvent evt) {
496     this.daftarList.clear();
497     this.phn.inOrderTraversal();
498     this.daftarList = this.phn.getListData();
499     tampilData_2();
500 }
501
502 private void btn_preActionPerformed(java.awt.event.ActionEvent evt) {
503     this.daftarList.clear();
504     this.phn.preorderTraversal();
505     this.daftarList = this.phn.getListData();
506     tampilData_2();
507 }
508
509 private void btn_postActionPerformed(java.awt.event.ActionEvent evt) {
510     this.daftarList.clear();
511     this.phn.postOrderTraversal();
512     this.daftarList = this.phn.getListData();
513     tampilData_2();
514 }
515
516 public void reset() {
517     txtKata.setText("");
518     txtArti.setText("");
519 }
```

11:50 JNS

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Source Design History

GUI\_Tree.java x JDialog\_Admin.java x JDialog\_Guest.java x JDialog\_Login.java x JFrame\_MenuUtama.java x Project\_Akhir\_V20.java x Tree.java x TreeNode.java x

```
518     txtArti.setText("");
519 }
520
521 public void tambahData() {
522     try {
523         if (txtKata.getText().equalsIgnoreCase("")
524             && txtArti.getText().equals("")) {
525             throw new Exception("Kolom kata dan kolom arti tidak boleh kosong");
526         } else {
527             String temp = txtKata.getText().toUpperCase();
528             node = this.phn.Search_Node(temp);
529             if (node == null) {
530                 if (temp == null) {
531                     throw new Exception("Kolom Kosakata Tidak Boleh Kosong");
532                 } else {
533                     String kosakata = temp;
534                     String pengertian = txtArti.getText();
535                     phn.insert(kosakata, pengertian);
536                     JOptionPane.showMessageDialog(this, "Data Berhasil Di Tambahkan", "Information", JOptionPane.INFORMATION_MESSAGE);
537                 }
538             } else {
539                 throw new Exception("Data sudah tersedia dalam daftar");
540             }
541         }
542     } catch (Exception e) {
543         JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
544             JOptionPane.WARNING_MESSAGE);
545     }
546 }
```

11:50 JNS

```
Project_Akhir_V2.0 - NetBeans IDE 8.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

GUI_Tree.java x JDialog_Admin.java x JDialog_Guest.java x JDialog_Login.java x JFrame_MenuUtama.java x Project_Akhir_V20.java x Tree.java x TreeNode.java x

Source Design History
545 }
546
547
548 public void tampilData_1() {
549     this.daftarList.clear();
550     this.phn.listData.clear();
551     this.phn.inOrderTraversal();
552     this.daftarList = this.phn.getListData();
553
554     String[] header = {"No", "Kosakata"};
555     Object[][] objKata = new Object[daftarList.size()][2];
556     int i = 0;
557     int j = 1;
558     for (TreeNode n : daftarList) {
559         if (n instanceof TreeNode) {
560             String[] arrData = {String.valueOf(j++), n.getData()};
561             objKata[i] = arrData;
562             i++;
563         }
564     }
565     tabelModel_1 = new DefaultTableModel(objKata, header) {
566         //membuat tabel tidak bisa diedit saat diklik
567         @Override
568         public boolean isCellEditable(int rowIndex, int colIndex) {
569             return false;
570         }
571     };
572     jTable1.setModel(tabelModel_1);
573     columnModel_1 = jTable1.getColumnModel();
```

```
Project_Akhir_V2.0 - NetBeans IDE 8.0
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl-F)

GUI_Tree.java x JDialog_Admin.java x JDialog_Guest.java x JDialog_Login.java x JFrame_MenuUtama.java x Project_Akhir_V20.java x Tree.java x TreeNode.java x

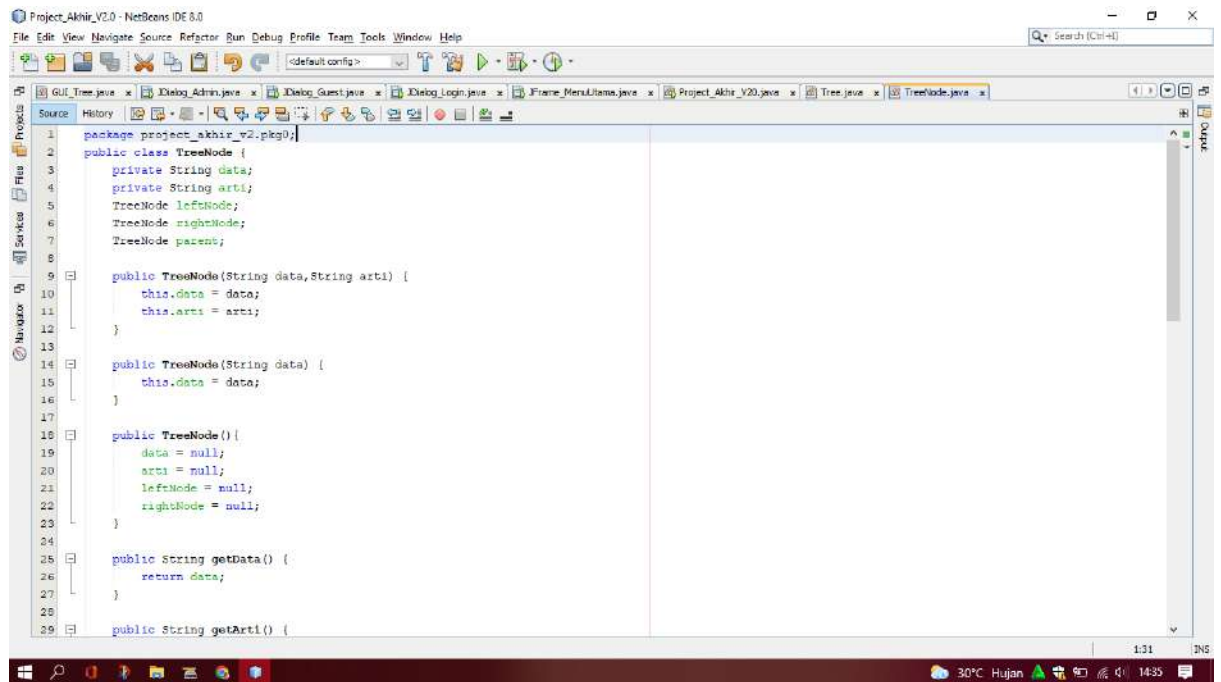
Source Design History
572 jTable1.setModel(tabelModel_1);
573 columnModel_1 = jTable1.getColumnModel();
574 columnModel_1.getColumn(0).setMinWidth(40);
575 columnModel_1.getColumn(1).setMinWidth(200);
576
577
578 public void tampilData_2() {
579     String[] header = {"No", "Kosakata", "Arti"};
580     Object[][] objKata = new Object[daftarList.size()][3];
581     int i = 0;
582     int j = 1;
583     for (TreeNode n : daftarList) {
584         if (n instanceof TreeNode) {
585             String[] arrData = {String.valueOf(j++), n.getData(), n.getArti()};
586             objKata[i] = arrData;
587             i++;
588         }
589     }
590     tabelModel_2 = new DefaultTableModel(objKata, header) {
591         //membuat tabel tidak bisa diedit saat diklik
592         @Override
593         public boolean isCellEditable(int rowIndex, int colIndex) {
594             return false;
595         }
596     };
597     jTable2.setModel(tabelModel_2);
598     columnModel_2 = jTable2.getColumnModel();
599     columnModel_2.getColumn(0).setMinWidth(3);
600     columnModel_2.getColumn(1).setMinWidth(90);
```

```
599 columnModel_2.getColumnModel().setMinWidth(2);
600 columnModel_2.getColumnModel().setMinWidth(90);
601 columnModel_2.getColumnModel().setMinWidth(805);
602
603 }
604
605 public static void main(String args[]) { ...18 lines ... }
606
607 // Variables declaration - do not modify
608 private javax.swing.JButton btn_cari;
609 private javax.swing.JButton btn_clear;
610 private javax.swing.JButton btn_edit;
611 private javax.swing.JButton btn_hapus;
612 private javax.swing.JButton btn_in;
613 private javax.swing.JButton btn_post;
614 private javax.swing.JButton btn_pre;
615 private javax.swing.JButton btn_tambah;
616 private javax.swing.JFrame jInternalFrame1;
617 private javax.swing.JFrame jInternalFrame2;
618 private javax.swing.JLabel jLabel1;
619 private javax.swing.JPanel jPanel1;
620 private javax.swing.JPanel jPanel2;
621 private javax.swing.JPanel jPanel3;
622 private javax.swing.JPanel jPanel4;
623 private javax.swing.JPanel jPanel5;
624 private javax.swing.JPanel jPanel6;
625 private javax.swing.JPanel jPanel7;
626 private javax.swing.JScrollPane jScrollPane1;
627 private javax.swing.JScrollPane jScrollPane2;
```

```
628 private javax.swing.JButton btn_cari;
629 private javax.swing.JButton btn_clear;
630 private javax.swing.JButton btn_edit;
631 private javax.swing.JButton btn_hapus;
632 private javax.swing.JButton btn_in;
633 private javax.swing.JButton btn_post;
634 private javax.swing.JButton btn_pre;
635 private javax.swing.JButton btn_tambah;
636 private javax.swing.JFrame jInternalFrame1;
637 private javax.swing.JFrame jInternalFrame2;
638 private javax.swing.JLabel jLabel1;
639 private javax.swing.JPanel jPanel1;
640 private javax.swing.JPanel jPanel2;
641 private javax.swing.JPanel jPanel3;
642 private javax.swing.JPanel jPanel4;
643 private javax.swing.JPanel jPanel5;
644 private javax.swing.JPanel jPanel6;
645 private javax.swing.JPanel jPanel7;
646 private javax.swing.JScrollPane jScrollPane1;
647 private javax.swing.JScrollPane jScrollPane2;
648 private javax.swing.JScrollPane jScrollPane3;
649 private javax.swing.JTabbedPane jTabbedPane1;
650 private javax.swing.JTable jTable1;
651 private javax.swing.JTable jTable2;
652 private javax.swing.JTextArea txtArtis;
653 private javax.swing.JTextField txtKata;
```

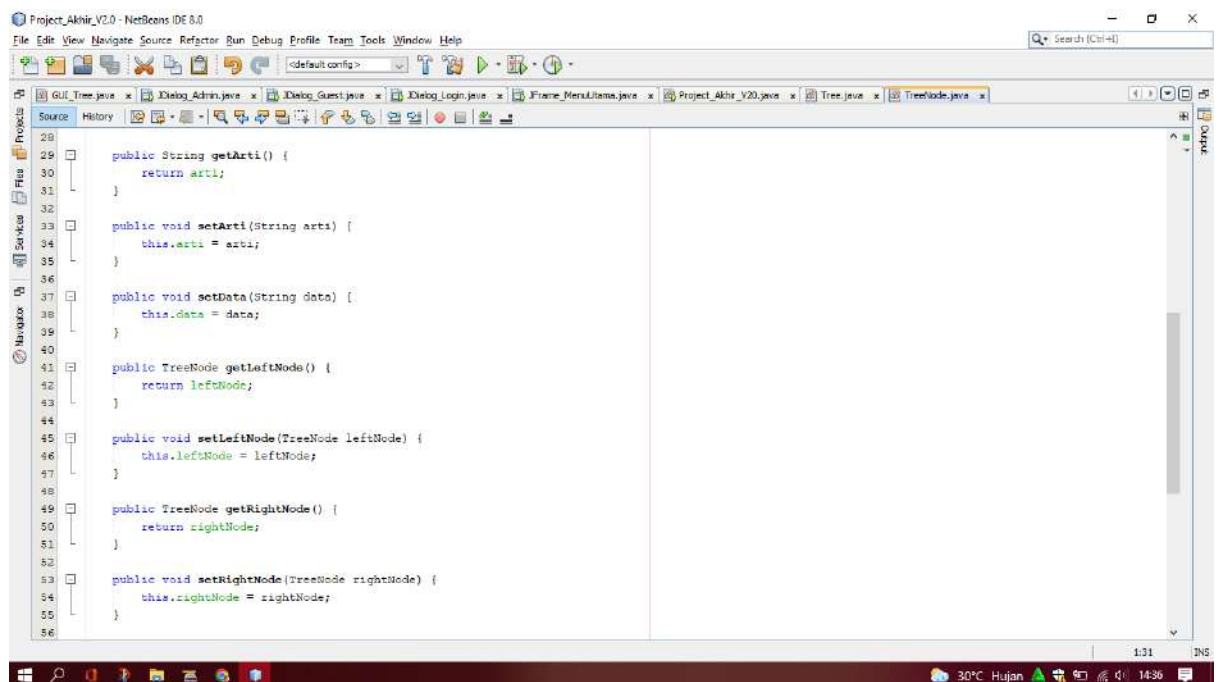
## 6. Class TreeNode





Project\_Akhir\_V2.0 - NetBeans IDE 8.0

```
1 package project_akhir_v2.pkg0;
2 public class TreeNode {
3     private String data;
4     private String arti;
5     TreeNode leftNode;
6     TreeNode rightNode;
7     TreeNode parent;
8
9     public TreeNode(String data,String arti) {
10         this.data = data;
11         this.arti = arti;
12     }
13
14     public TreeNode(String data) {
15         this.data = data;
16     }
17
18     public TreeNode() {
19         data = null;
20         arti = null;
21         leftNode = null;
22         rightNode = null;
23     }
24
25     public String getData() {
26         return data;
27     }
28
29     public String getArti() {
```



Project\_Akhir\_V2.0 - NetBeans IDE 8.0

```
28
29     public String getArti() {
30         return arti;
31     }
32
33     public void setArti(String arti) {
34         this.arti = arti;
35     }
36
37     public void setData(String data) {
38         this.data = data;
39     }
40
41     public TreeNode getLeftNode() {
42         return leftNode;
43     }
44
45     public void setLeftNode(TreeNode leftNode) {
46         this.leftNode = leftNode;
47     }
48
49     public TreeNode getRightNode() {
50         return rightNode;
51     }
52
53     public void setRightNode(TreeNode rightNode) {
54         this.rightNode = rightNode;
55     }
56
```



```
56
57     public TreeNode getParent() {
58         return parent;
59     }
60
61     public void setParent(TreeNode parent) {
62         this.parent = parent;
63     }
64
65 }
```

## 7. Class Tree



Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Source History

```
1 package project_akhir_v2.pkg0;
2 import java.util.ArrayList;
3 public class Tree {
4
5     ArrayList<TreeNode> listData = new ArrayList<>();
6     TreeNode root,temp;
7     TreeNode parent;
8     String hasilPencarian_kata, hasilPencarian_arti;
9
10
11     public Tree() {
12         this.root = null;
13     }
14
15     public Tree(TreeNode node) {
16         this.root = node;
17     }
18
19     public ArrayList<TreeNode> getListData() {
20         return listData;
21     }
22
23     public void insert(String dataNode, String arti) {
24         if (root == null) {
25             root = new TreeNode(dataNode, arti);
26         } else {
27             insertHelper(root, dataNode, arti);
28         }
29     }
```

18:1 2NS

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Source History

```
29 }
30
31 public void insertHelper(TreeNode pointer, String kata, String arti){
32     if(kata.compareTo(pointer.getData()) < 1){
33         if(pointer.getLeftNode() != null){
34             insertHelper(pointer.getLeftNode(), kata, arti);
35         }else{
36             TreeNode newNode = new TreeNode(kata, arti);
37             pointer.setLeftNode(newNode);
38             newNode.setParent(pointer);
39         }
40     }else{
41         if(pointer.getRightNode() != null)
42             insertHelper(pointer.getRightNode(), kata, arti);
43         else{
44             TreeNode newNode = new TreeNode(kata, arti);
45             pointer.setRightNode(newNode);
46             newNode.setParent(pointer);
47         }
48     }
49 }
50
51
52
53
54 public void inorderTraversal() {
55     listData.clear();
56     inorderHelper(this.getRoot());
57 }
58
59 public void inorderHelper(TreeNode localRoot) {
60     if (localRoot != null) {
61         inorderHelper(localRoot.getLeftNode());
62         listData.add(localRoot);
63         inorderHelper(localRoot.getRightNode());
64     }
65 }
66
67
68
69 public void inorderHelper(TreeNode localRoot) {
70     if (localRoot != null) {
71         inorderHelper(localRoot.getLeftNode());
72         listData.add(localRoot);
73         inorderHelper(localRoot.getRightNode());
74     }
75 }
76
77
78
79
80 public void inorderTraversal_Revese() {
81     listData.clear();
82     inorderHelper_Revese(this.getRoot());
83 }
84
85 public void inorderHelper_Revese(TreeNode localRoot) {
86     if (localRoot != null) {
87         inorderHelper_Revese(localRoot.getRightNode());
88         listData.add(localRoot);
89         inorderHelper_Revese(localRoot.getLeftNode());
90     }
91 }
```

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

GUI\_Tree.java x JDialog\_Admin.java x JDialog\_Guest.java x JDialog\_login.java x JFrame\_MenuItems.java x Project\_Akhir\_V2.0.java x Tree.java x TreeNode.java x

Source History

```
82 public void postOrderHelper(TreeNode localRoot) {
83     if (localRoot != null) {
84         postOrderHelper(localRoot.getLeftNode());
85         postOrderHelper(localRoot.getRightNode());
86         listData.add(localRoot);
87     }
88 }
89
90 public TreeNode getRoot() {
91     return root;
92 }
93
94 public void setRoot(TreeNode root) {
95     this.root = root;
96 }
97
98 public boolean isLeaf(TreeNode data) {
99     boolean cek = false;
100     TreeNode ptr = data;
101     if (ptr.getLeftNode() == null && ptr.getRightNode() == null) {
102         cek = true;
103     }
104     return cek;
105 }
106
107 public boolean delete(String data) {
108     boolean cek = false;
109     TreeNode node = Search_Node(data);
110     if (node != null) {
```

100:1 2NS

26°C Hujan rin... 21:43

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

GUI\_Tree.java x JDialog\_Admin.java x JDialog\_Guest.java x JDialog\_login.java x JFrame\_MenuItems.java x Project\_Akhir\_V2.0.java x Tree.java x TreeNode.java x

Source History

```
107 public boolean delete(String data) {
108     boolean cek = false;
109     TreeNode node = Search_Node(data);
110     if (node != null) {
111         if (!isLeaf(node))
112             && (node.getLeftNode() != null && node.getRightNode() != null) {
113                 cek = delete_2_anak(node);
114             } else if (!isLeaf(node)
115                 && (node.getLeftNode() != null || node.getRightNode() != null)) {
116                 cek = delete_1_anak(node);
117             } else {
118                 cek = delete_0_anak(node);
119             }
120     }
121     return cek;
122 }
123
124 public boolean delete_0_anak(TreeNode data) {
125     TreeNode parent = data.getParent();
126     if (parent.getLeftNode() == data) {
127         parent.setLeftNode(null);
128         data.setParent(null);
129     } else {
130         parent.setRightNode(null);
131         data.setParent(null);
132     }
133     return true;
134 }
135
```

100:1 2NS

26°C Hujan rin... 21:44

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Source History

```
135
136 public boolean delete_1_anak(TreeNode data) {
137     TreeNode parent = data.getParent();
138     TreeNode temp = data;
139
140     if (temp != root) {
141         if (parent.getLeftNode() == temp) {
142             if (temp.getLeftNode() != null) { //jika cabang kiri temp tidak null
143                 parent.setLeftNode(temp.getLeftNode());
144                 temp.getLeftNode().setParent(parent);
145             } else {
146                 parent.setLeftNode(temp.getRightNode());
147                 temp.getRightNode().setParent(parent);
148             }
149             temp.setParent(null);
150         } else {
151             if (temp.getRightNode() != null) { //jika cabang kanan temp tidak null
152                 parent.setRightNode(temp.getRightNode());
153                 temp.getRightNode().setParent(parent);
154             } else {
155                 parent.setRightNode(temp.getLeftNode());
156                 temp.getLeftNode().setParent(parent);
157             }
158             temp.setParent(null);
159         }
160     } else {
161         if (data.getRightNode() != null) { //cabang kanan ada ==> cabang kiri kosong
162             TreeNode n = GetSuccessor(temp);
163             if (n != null) {
```

163:1 JNS

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Source History

```
162     TreeNode n = GetSuccessor(temp);
163     if (n != null) {
164         root.setData(n.getData());
165         root.setArti(n.getArti());
166     } else { //temp = root
167         if (isLeaf(temp.getRightNode())) {
168             root.setData(temp.getRightNode().getData());
169             root.setArti(temp.getRightNode().getArti());
170             root.setRightNode(null);
171         } else {
172             root.setData(temp.getRightNode().getData());
173             root.setArti(temp.getRightNode().getArti());
174             root.setRightNode(temp.getRightNode().getRightNode());
175             temp.getRightNode().setParent(root);
176         }
177     }
178 } else { //cabang kiri ada ==> cabang kanan kosong
179     TreeNode n = GetPredecessor(temp);
180     if (n != null) {
181         root.setData(n.getData());
182         root.setArti(n.getArti());
183     } else { // temp = root
184         if (isLeaf(temp.getLeftNode())) {
185             root.setData(temp.getLeftNode().getData());
186             root.setArti(temp.getLeftNode().getArti());
187             root.setLeftNode(null);
188         } else {
189             root.setData(temp.getLeftNode().getData());
190             root.setArti(temp.getLeftNode().getArti());
```

190:1 JNS

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

GUI\_Tree.java x IDiag\_Admin.java x IDiag\_Guest.java x IDiag\_login.java x JFrame\_MenuItems.java x Project\_Akhir\_V2.0.java x Tree.java x TreeNode.java x

```
190         root.setArti(temp.getLeftNode().getArti());
191         root.setLeftNode(temp.getLeftNode().getLeftNode());
192         temp.getLeftNode().setParent(root);
193     }
194 }
195 }
196 }
197 return true;
198 }
199 }
200
201 public boolean delete_2_anak(TreeNode node) {
202     boolean cek = false;
203     TreeNode parent = node.getParent();
204     TreeNode temp = node;
205     if (node != root) {
206         if (parent.getRightNode() == temp) {
207             if (temp.getRightNode() != null) {
208                 parent.setRightNode(temp.getRightNode());
209                 temp.getRightNode().setParent(parent);
210                 temp.getLeftNode().setLeftNode(temp.getRightNode());
211                 temp.getLeftNode().setParent(temp.getRightNode());
212             } else {
213                 parent.setLeftNode(temp.getLeftNode());
214                 temp.getLeftNode().setParent(parent);
215                 temp.getRightNode().setRightNode(temp.getRightNode());
216                 temp.getRightNode().setParent(temp.getLeftNode());
217             }
218         }
219         temp.setParent(null);
220     }
```

218:1 JNS

Project\_Akhir\_V2.0 - NetBeans IDE 8.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

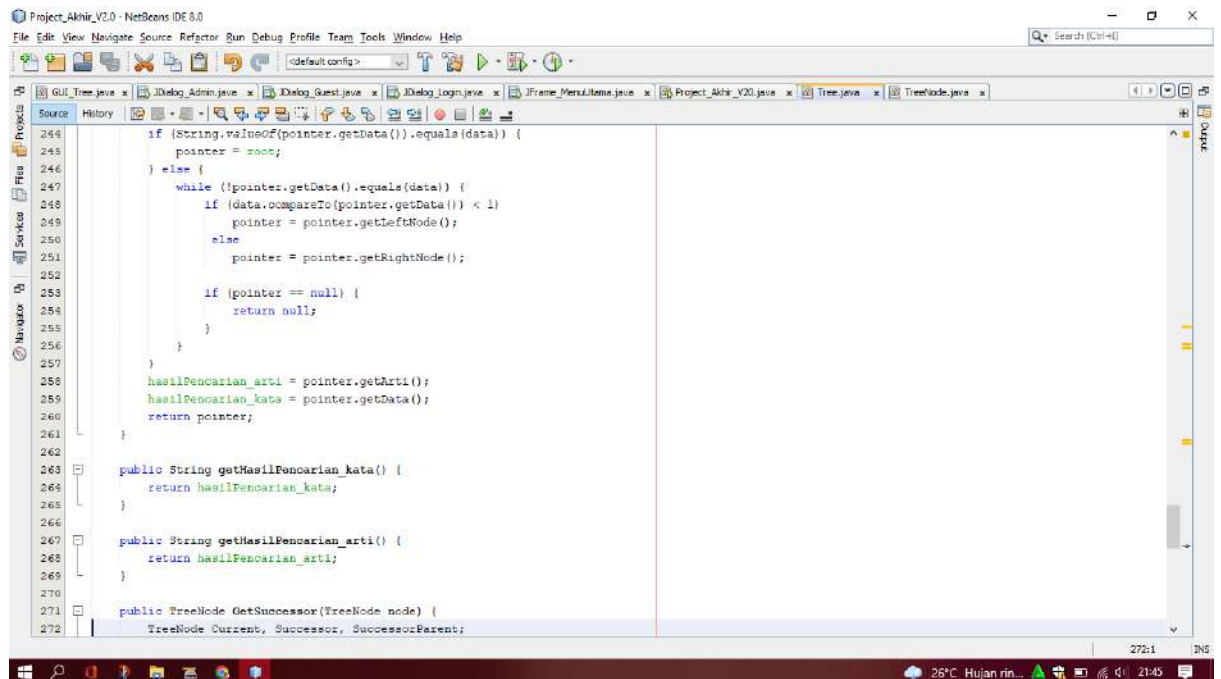
Search (Ctrl-F)

GUI\_Tree.java x IDiag\_Admin.java x IDiag\_Guest.java x IDiag\_login.java x JFrame\_MenuItems.java x Project\_Akhir\_V2.0.java x Tree.java x TreeNode.java x

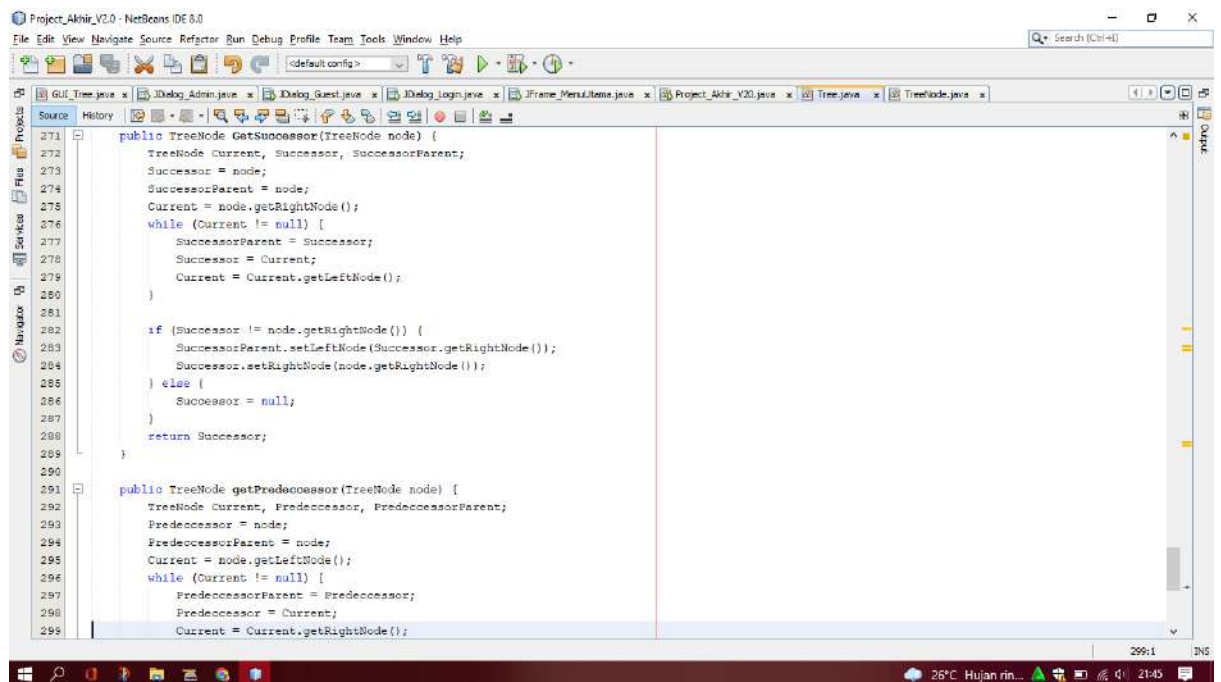
```
217         temp.setParent(null);
218         cek = true;
219     } else if (parent.getLeftNode() == temp) {
220         if (!isLeaf(temp.getRightNode())) {
221             TreeNode GetSuccessor = GetSuccessor(temp);
222             temp.setData(GetSuccessor.getData());
223             temp.setArti(GetSuccessor.getArti());
224             temp.setRightNode(null);
225         } else {
226             TreeNode GetSuccessor = GetSuccessor(temp);
227             temp.setData(GetSuccessor.getData());
228             temp.setArti(GetSuccessor.getArti());
229         }
230         cek = true;
231     }
232 }
233 } else {
234     TreeNode getPredecessor = getPredecessor(temp);
235     root.setData(getPredecessor.getData());
236     root.setArti(getPredecessor.getArti());
237     cek = true;
238 }
239 return cek;
240 }
241
242 public TreeNode Search_Node(String data) {
243     TreeNode pointer = root;
244     if (String.valueOf(pointer.getData()).equals(data)) {
245         pointer = root;
```

245:1 JNS

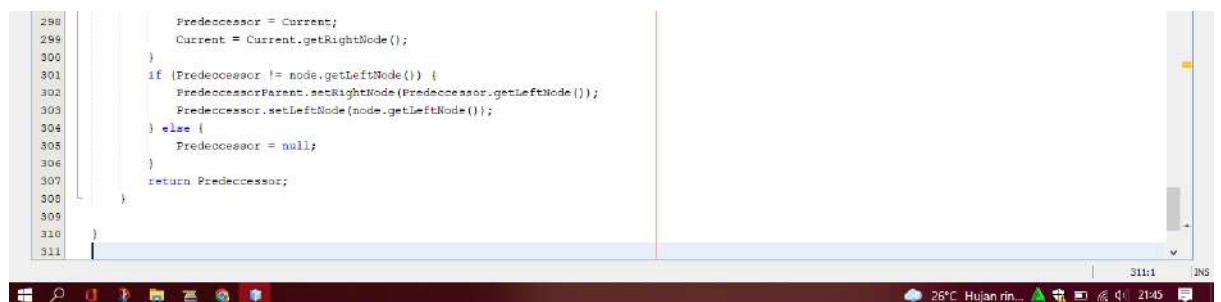




```
244     if (String.valueOf(pointer.getData()).equals(data)) {
245         pointer = root;
246     } else {
247         while (!pointer.getData().equals(data)) {
248             if (data.compareTo(pointer.getData()) < 1)
249                 pointer = pointer.getLeftNode();
250             else
251                 pointer = pointer.getRightNode();
252         }
253         if (pointer == null) {
254             return null;
255         }
256     }
257     hasilPencarian_arti = pointer.getArti();
258     hasilPencarian_kata = pointer.getData();
259     return pointer;
260 }
261
262
263 public String getHasilPencarian_kata() {
264     return hasilPencarian_kata;
265 }
266
267 public String getHasilPencarian_arti() {
268     return hasilPencarian_arti;
269 }
270
271 public TreeNode GetSuccessor(TreeNode node) {
272     TreeNode Current, Successor, SuccessorParent;
```



```
271 public TreeNode GetSuccessor(TreeNode node) {
272     TreeNode Current, Successor, SuccessorParent;
273     Successor = node;
274     SuccessorParent = node;
275     Current = node.getRightNode();
276     while (Current != null) {
277         SuccessorParent = Successor;
278         Successor = Current;
279         Current = Current.getLeftNode();
280     }
281
282     if (Successor != node.getRightNode()) {
283         SuccessorParent.setLeftNode(Successor.getRightNode());
284         Successor.setRightNode(node.getRightNode());
285     } else {
286         Successor = null;
287     }
288     return Successor;
289 }
290
291 public TreeNode getPreddecessor(TreeNode node) {
292     TreeNode Current, Preddecessor, PreddecessorParent;
293     Preddecessor = node;
294     PreddecessorParent = node;
295     Current = node.getLeftNode();
296     while (Current != null) {
297         PreddecessorParent = Preddecessor;
298         Preddecessor = Current;
299         Current = Current.getRightNode();
```



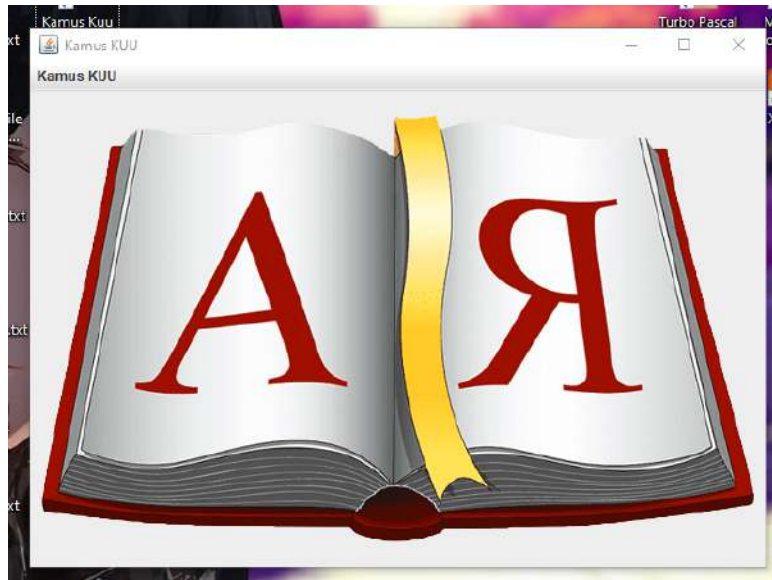
```
298         Preddecessor = Current;
299         Current = Current.getRightNode();
300     }
301     if (Preddecessor != node.getLeftNode()) {
302         PreddecessorParent.setRightNode(Preddecessor.getLeftNode());
303         Preddecessor.setLeftNode(node.getLeftNode());
304     } else {
305         Preddecessor = null;
306     }
307     return Preddecessor;
308 }
309
310 }
311 }
```

## E. PENJELASAN UMUM



## 1. JFrame\_Utama

Pada aplikasi ini menggunakan menu utama pada saat user menjalankan aplikasi maka frame utama ini yang akan ditampilkan pertama dan terus ditampilkan saat user menggunakan aplikasi ini. Frame utama yang dimaksud adalah seperti gambar di bawah ini.



Yang dimana pada menu ini menampung menu lainnya untuk navigasi ke dalam fitur admin dan fitur guest. Sehingga code yang ada di dalam frame ini adalah sebagai berikut.

```
public GUI_Tree phn = new GUI_Tree();
```

```
public JFrame_MenuUtama() {  
    initComponents();  
    phn = new GUI_Tree();  
    help();  
}
```

```
@SuppressWarnings("unchecked")
```

```
Generated Code
```

```
private void menu_adminActionPerformed(java.awt.event.ActionEvent evt) {  
    if (this.phn.isKondisi() == false) {  
        JDialog_Login login = new JDialog_Login(this, true, phn);  
        login.setVisible(true);  
    }else{  
        JDialog_Admin admin = new JDialog_Admin(this,true,phn);  
        admin.setVisible(true);  
    }  
}
```

Pada bagian konstruktor dari class frame ini membuat object dengan nama variabel phn dari class GUI\_Tree yang nantinya akan digunakan oleh frame lainnya dengan melempar data object yang sudah di buat pada class frame ini. Dan juga pada konstruktor memanggil method help.

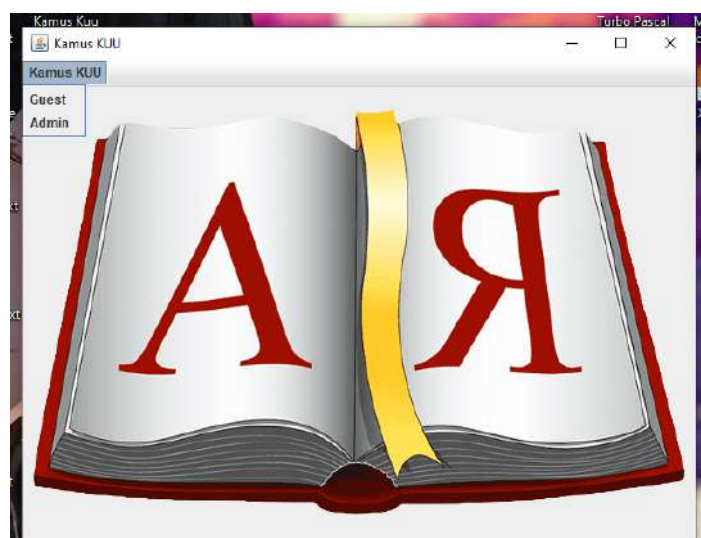
```
private void menu_guestActionPerformed(java.awt.event.ActionEvent evt) {  
    JDialog_Guest g = new JDialog_Guest(this, true, phn);  
    g.setVisible(true);  
}  
  
public void help() {  
    String[] kata = {"AKAR", "BAJU", "BANDAR", "CERITA", "CENDAWAN", "ABAD",  
        "ALUR", "BERDALIH", "BERSUA"};  
    String[] arti = {"Bagian tumbuhan yang tertanam di dalam tanah", "Pakaian"  
        + " penutup badan bagian atas", "Orang yang bermodal dalam "  
        + "perdagangan", "Omong kosong", "Jamur yang besar", "Masa "  
        + "Seratur Tahun", "Rangkaian peristiwa yang direka dan di "  
        + "jalin dengan seksama dan menggerakkan jalan cerita melalui "  
        + "kerumitan ke arah klimaks dan penyelesaian", "Mengemukakan "  
        + "alasan (yang dicari-cari) untuk menghindari tugas atau menutupi "  
        + "perbuatan yang salah atau tercela", "Data saling mendekati "  
        + "(berdekat-dekatan)"};  
    for (int i = 0; i < kata.length; i++) {  
        phn.insert(kata[i], arti[i]);  
        System.out.println("Data Berhasil Di Tambahkan");  
    }  
}
```

Method help ini di gunakan untuk menambahkan data secara otomatis ketika aplikasi di jalankan maka di dalam aplikasi sudah terdapat data yang sudah secara otomatis di tambahkan.

## 2. Form Login

Pada aplikasi ini menggunakan fitur login sederhana yang dimana ketika user ingin mengganti arti dari kosakata yang sudah ada ataupun ingin menambah kosakata baru dan juga jika ingin menghapus kosakata yang diinginkan oleh user.

Dan form ini akan muncul ketika user mengklik menu button admin pada menu utama aplikasi kamus seperti gambar di bawah ini.



Pada kondisi gambar di atas merupakan saat keadaan awal membuka aplikasi kamus dengan kode seperti gambar di bawah ini.

```
private void menu_adminActionPerformed(java.awt.event.ActionEvent evt) {  
    if (this.phn.isKondisi() == false) {  
        JDialog_Login login = new JDialog_Login(this, true, phn);  
        login.setVisible(true);  
    }else{  
        JDialog_Admin admin = new JDialog_Admin(this,true,phn);  
        admin.setVisible(true);  
    }  
}
```



Gambar tersebut terletak pada menu navigasi admin pada form awal aplikasi yang dimana dalam menu terdapat pengecekan kondisi dimana mengambil data dari atribut kondisi dari object tree dimana jika bernilai false maka akan memunculkan form login seperti gambar di bawah ini dan jika bernilai true maka form login tidak akan dimunculkan kembali sehingga user langsung dihadapkan dengan form admin.

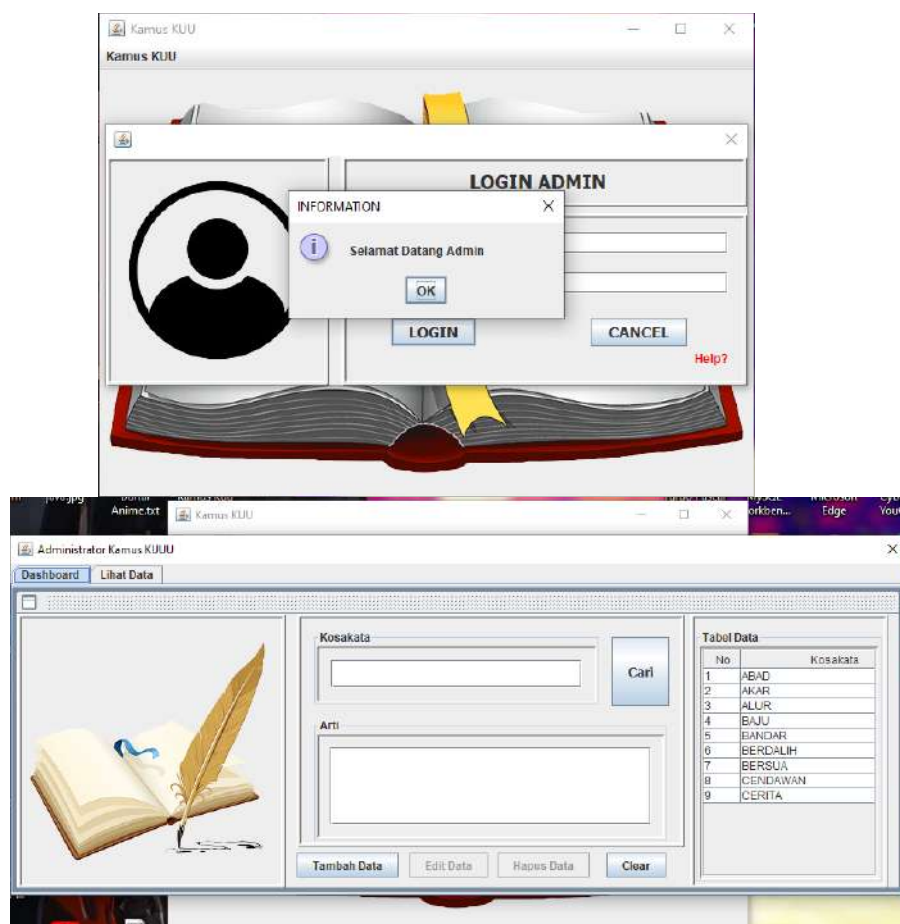
Pada form ini terdapat beberapa button yang pertama yaitu login, cancel dan help atau petunjuk jika user tidak bisa login.

Yang pertama button cancel yaitu button untuk menutup form ini jika user tidak jadi untuk melakukan login, dan kembali akan di arahkan ke form menu utama dari aplikasi kamus ini. Yang code nya sebagai berikut :

```
private void btn_cancelActionPerformed(java.awt.event.ActionEvent evt) {  
    this.dispose();  
}
```

Yang dimana berfungsi seperti menyembunyikan form login sehingga dalam pandangan user nampak form ini tertutup kembali.

Pada saat user berhasil login akan muncul pop up seperti gambar di bawah ini.



Ketika user berhasil login dan masuk ke dalam form admin maka form login tersebut tidak akan ada lagi selama aplikasi berjalan, namun ketika aplikasi di

tutup oleh user dan membuka kembali aplikasi kamus ini maka akan muncul

```
private void btn_loginActionPerformed(java.awt.event.ActionEvent evt) {  
    login();  
}
```

kembali form login ini.

```
152 public void login() {  
153     String password = new String(txtPass.getPassword());  
154     try {  
155         if (txtUsername.getText().isEmpty() || password.equals(null)) {  
156             throw new Exception("Username Dan Password Tidak Boleh Kosong");  
157         }  
158         if (!txtUsername.getText().equals(user)) {  
159             throw new Exception("Username yang anda masukkan salah");  
160         }  
161         if (!password.equals(pass)) {  
162             throw new Exception("Password yang anda masukkan salah");  
163         }  
164         if (txtUsername.getText().equals(user) && password.equals(pass)) {  
165             reset();  
166             JOptionPane.showMessageDialog(this, "Selamat Datang Admin", "INFORMATION",  
167                 JOptionPane.INFORMATION_MESSAGE);  
168             this.dispose();  
169             JDialog_Admin a = new JDialog_Admin(this.f, true, this.tree);  
170             a.setVisible(true);  
171         }  
172     } catch (Exception e) {  
173         JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",  
174             JOptionPane.WARNING_MESSAGE);  
175     }  
176 }
```

Pada gambar di atas merupakan code pada button login yang dimana button login akan memanggil method login yang ada. Kemudian method login di eksekusi akan mengambil data dari atribut txtPass yang dimana di casting menjadi string lalu di simpan ke variabel password. Kemudian mengecek apakah atribut txtUsername atau txtPass bernilai kosong atau null maka akan memunculkan dialog error "Username Dan Password Tidak Boleh Kosong", kemudian jika txtUsername tidak bernilai sama dengan data yang ada pada atribut user yang dimana data nya di tampilkan pada button help maka akan memunculkan dialog error "Username yang anda masukkan salah", kemudian jika txtPass juga tidak bernilai sama dengan attribut pass maka akan memunculkan dialoh error "Password yang anda masukkan salah" dan jika username dan password sama dengan atribut user dan pass maka akan memanggil method reset yang code nya seperti gambar di bawah ini.

```
public void reset() {  
    txtUsername.setText("");  
    txtPass.setText("");  
}
```

Method ini mengosongkan txtUsername dan txtPass, kemudian akan merubah data pada atribut kondisi menjadi true, selanjutnya akan memunculkan dialog



informasi success "Selamat Datang Admin" dan kemudian menutup form login dengan method .dispose(), baru setelah itu memanggil form admin.

Kemudian pada form login ini juga terdapat error trapping yang dimana ketika user langsung mengklik button login tanpa mengisi username dan password maka akan muncul pop-up informasi error seperti gambar di bawah ini.



Dan ketika user salah memasukkan username dan password maka akan muncul pop-up informasi error login seperti gambar di bawah ini.



### 3. GUI Guest JDialog

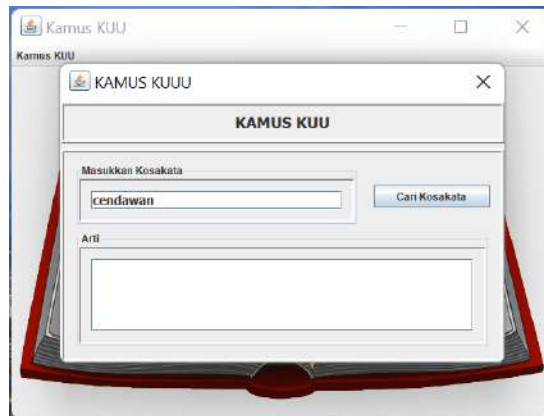
Pada form JDialog gui untuk guest ini akan menerima data yang dipassingkan dari Jpanel atau dari form sebelumnya yaitu form menu utama yang dimana hard code nya seperti gambar di bawah ini.

```
5      public class JDialog_Guest extends javax.swing.JDialog {  
6  
7          GUI_Tree phn;  
8          TreeNode node;  
9      public JDialog_Guest(java.awt.Frame parent, boolean modal, GUI_Tree data) {  
10         super(parent, modal);  
11         initComponents();  
12         this.phn = data;  
13     }
```

Yang dimana memasukkan data yang dimasukan ke dalam variabel yang sudah dideklarasikan secara local pada class JDialog\_Guest. Sehingga pada saat form ini dijalankan sudah memiliki data di dalam nya dan user dapat menggunakan form ini untuk mencari kata yang di inginkan. Yang dimana seperti gambar di bawah ini.



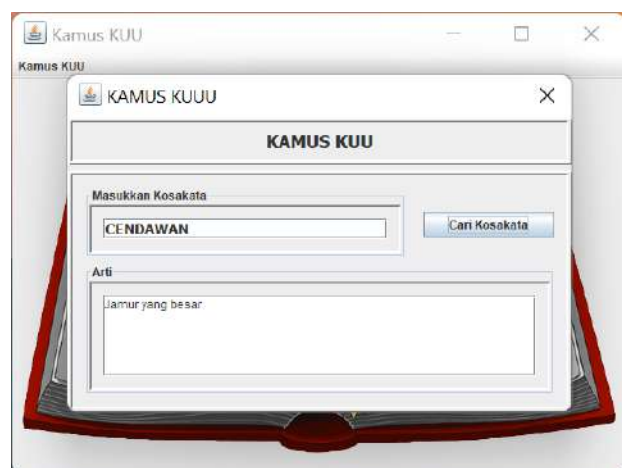
Selanjutnya bisa memasukkan kosakata yang akan dicari pada bagian text field "Masukkan Kosakata", misalnya mencari arti kata 'cendawan'.



Setelah memasukkan kosakata bisa meng-klik bagian “Cari Kosakata”, maka akan muncul pesan pop-up bahwa kata berhasil ditemukan.

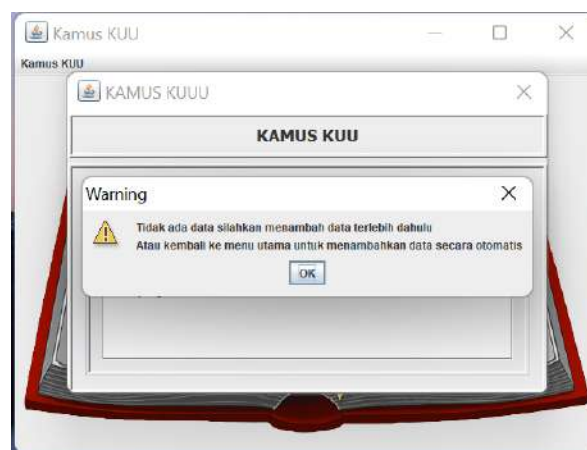


Pengguna dapat mengklik ‘Ok’ untuk melihat arti kata yang dicari, seperti di bawah ini.



Jika kata tidak ada pada kamus (tidak ada node pada tree dengan data yang sama dengan kata yang dicari), maka akan muncul pesan peringatan

bahwa kata tidak ada pada kamus serta solusi yang dapat dilakukan. Misalnya mencari kata ‘Akbar’, seperti di bawah ini.



Pencarian kosa kata pada JDialog Guest ini dapat dilakukan dengan memanggil event handling ketika button “Cari Kosakata” diklik.

```
149 private void btn_cariActionPerformed(java.awt.event.ActionEvent evt) {  
150     String temp = txtKata.getText().toUpperCase();  
151     try {  
152         if (temp == null) {  
153             throw new Exception("Kolom Kosakata Tidak Boleh Kosong");  
154         }  
155         node = this.phn.Search_Node(temp);  
156         if (node == null) {  
157             throw new Exception("Tidak ada data silahkan menambah data terlebih dahulu " + "\n"  
158                 + "Atau kembali ke menu utama untuk menambahkan data secara otomatis");  
159         } else {  
160             txtKata.setText(node.getData());  
161             txtArti.setText(node.getArti());  
162             JOptionPane.showMessageDialog(this, "Kata ditemukan!", "INFORMATION",  
163                 JOptionPane.INFORMATION_MESSAGE);  
164         }  
165     } catch (Exception e) {  
166         JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",  
167             JOptionPane.WARNING_MESSAGE);  
168     }  
169 }
```

Yang dimana data dari text field yang nama variabel nya adalah txtKata akan diambil data nya kemudian data tersebut diubah bentuk string nya menjadi

Uppercase atau menjadi huruf besar semua yang kemudian dimasukkan ke dalam variabel yang bernama temp dengan tipe data String.

Dan kemudian aplikasi akan mencoba untuk mengecek apakah data pada variabel temp bernilai null, dan jika true maka akan memunculkan error “Kolom Kosakata Tidak Boleh Kosong”. Error ini berfungsi ketika user langsung mengklik button Cari Kosakata tanpa mengisi field Masukkan Kosakata.

Kemudian jika lolos maka program akan menginisialisasi variabel node dengan data nya yang berasal dari method Search\_Node dengan memasingkan data dari variabel temp.

Kemudian di cek kembali apakah data pada variabel node yang didapat dari method Search\_Node bernilai null atau tidak. Jika true maka akan memunculkan error kembali yang dimana pesan error nya adalah “Tidak ada data silahkan menambah data terlebih dahulu atau kembali ke menu utama untuk menambahkan data secara otomatis” yang dimana ini adalah bentuk error ini muncul ketika user memasukkan kosakata pada field Cari Kosakata kemudian kosakata tersebut tidak ditemukan atau tidak ada dalam penyimpanan di dalam aplikasi ini.

Dan jika data pada variabel node tidak bernilai null maka akan menampilkan data secara lengkap dari kosakata sampai artinya. Yang dimana lokasi penampilan kosakata diletakkan pada field Cari Kosakata, dan penampilan arti dari kosakatanya diletakkan pada field Arti, lalu kemudian aplikasi menampilkan pesan success yaitu “Kata Ditemukan.”. Yang dimana kata dari user masukkan di field Cari Kosakata ditemukan di dalam penyimpanan aplikasi dan user dapat mengetahui kosakata dan artinya sekaligus.

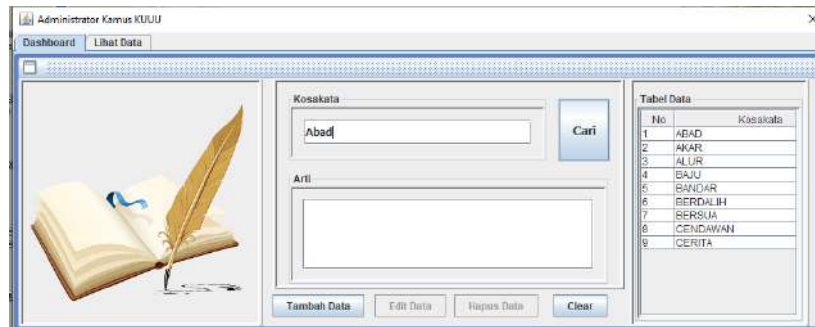
#### **4. JDialog\_Admin**

Pada jdialog ini, gui yang ditampilkan memiliki dua bagian yang dibagi dengan tab. Pada tab utama, admin dapat melakukan pencarian kata, penambahan kata, pengeditan pada kata tertentu serta penghapusan kata. Fitur dari bagian admin dilengkapi dengan pilihan untuk menampilkan data kata secara pre-order, in-order atau post-order dan ditampilkan pada bagian tab kedua.

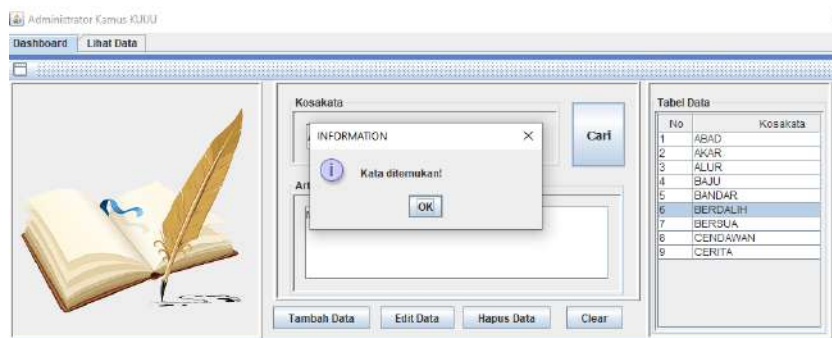
##### **a. Pencarian arti kata**

Berikut adalah langkah-langkah dari pencarian kata “Abad” yang sudah tersimpan di dalam kamus. Pertama, tuliskan “Abad” pada kolom kosakata dan setelahnya klik button “Cari”.

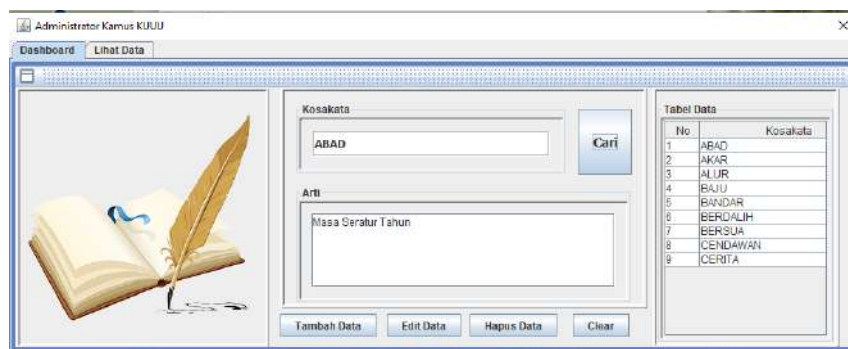




Jika kata yang dicari tersimpan dalam kamus, maka akan muncul pop up pemberitahuan yang mengatakan “Kata ditemukan”.



Setelah mengklik “OK”, arti dari kata akan ditampilkan pada kolom arti.



Pencarian kata pada kamus dapat dilakukan dengan memanggil event handling button “Cari” pada kelas `JDialog_Admin`. Dalam method event ini, text pada text field bernama `txtKata` akan disimpan pada variabel `temp` yang bertipe `String`. Maka, `temp` akan memiliki nilai yang sama dengan kata yang dicari admin. Setelah itu, proses akan berlanjut pada trapping exception.

```
private void btn_cariActionPerformed(java.awt.event.ActionEvent evt) {
    String temp = txtKata.getText().toUpperCase();
    try {
        if (temp == null) {
            throw new Exception("Kolom Kosakata Tidak Boleh Kosong");
        }
        node = this.phn.Search_Node(temp);
    }
}
```

Jika temp bernilai null, dimana kolom masih kosong, maka exception dengan kalimat “Kolom Kosakata Tidak Boleh Kosong” akan diberikan. Pencarian pada tree akan dilakukan jika temp tidak null dengan memanggil method Search\_Node( ) yang berargumen temp. Nilai yang dikembalikan dari method tersebut akan disimpan pada variabel node.

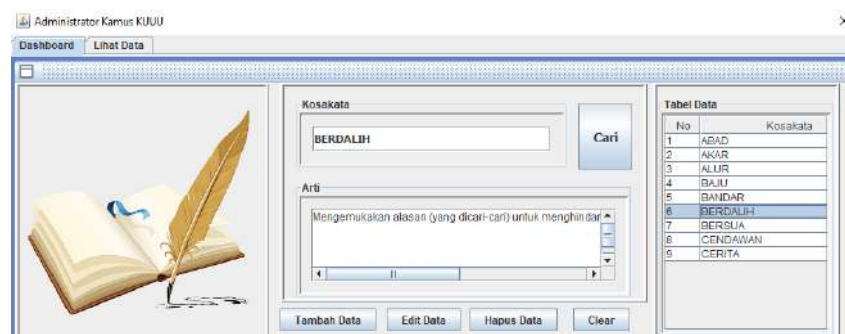
```
if (node == null) {
    throw new Exception("Tidak ada data silahkan menambah data terlebih dahulu " + "\n"
        + "Atau kembali ke menu utama untuk menambahkan data secara otomatis");
} else {
    txtKata.setText(node.getData());
    txtArti.setText(node.getArti());
    btn_hapus.setEnabled(true);
    btn_edit.setEnabled(true);
    JOptionPane.showMessageDialog(this, "Kata ditemukan!", "INFORMATION",
        JOptionPane.INFORMATION_MESSAGE);
}
```

Nilai node yang dikembalikan dapat bernilai null ketika kata yang dicari tidak ada pada kamus atau belum ditambahkan. Maka, sebuah exception akan di throw kembali. Untuk node yang tidak bernilai null, maka kolom kosa kata (txtKata) akan diubah menjadi atribut data dari objek node. Kolom arti (txtArti) juga akan diubah menjadi atribut arti dari objek node. Selanjutnya, button untuk menghapus (btn\_hapus) serta button untuk mengedit kata (btn\_edit) akan diaktifkan. Di atas disebutkan bahwa pesan pop akan muncul ketika kata ditemukan, ini dapat dilakukan dengan memanggil method JOptionPane berupa pesan informasi.

```
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
        JOptionPane.WARNING_MESSAGE);
}
```

Selanjutnya, exception yang di throw akan diproses dengan memanggil method JOptionPane berupa pesan peringatan dengan isi kalimat yang diberikan.

Pencarian kata juga dapat dilakukan dengan melakukan *double* klik kata yang terdapat pada tabel. Maka, kata yang ditampilkan di tabel akan muncul di kolom kosakata dan artinya akan mengisi kolom arti.



Untuk dapat melakukan pengaturan seperti di atas, diperlukan event handling pada tabel, yaitu `JTableMouseClicked`. Pada method ini, kolom kosakata dan kolom arti akan dikosongkan dengan memanggil method `reset()`. Selanjutnya, baris dimana admin mengklik di tabel akan disimpan ke sebuah variabel bertipe integer yaitu `selectedRow`. Untuk dapat menampilkan kata dan arti yang dipilih, cari terlebih dahulu node dengan data yang sama dengan memanggil method `Search_Node` dari kelas `Tree`. Hasil pengembalian akan disimpan pada sebuah variabel bernama `node`. Untuk node yang tidak bernilai null, maka kolom kosa kata (`txtKata`) akan diubah menjadi atribut data dari objek `node`. Kolom arti (`txtArti`) juga akan diubah menjadi atribut arti dari objek `node`. Selanjutnya, button untuk menghapus (`btn_hapus`) serta button untuk mengedit kata (`btn_edit`) akan diaktifkan.

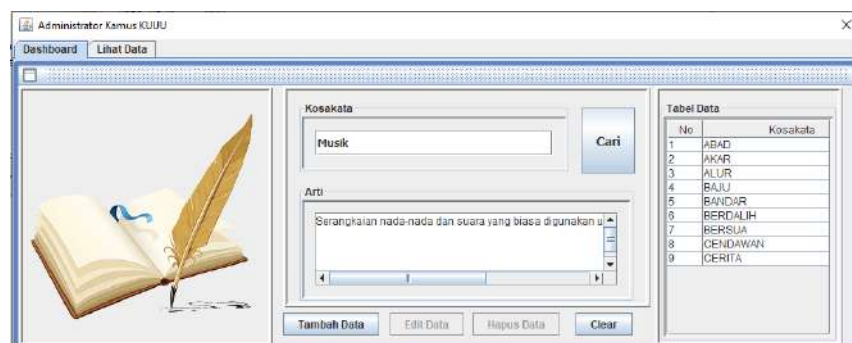
```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    reset();
    int selectedRow = jTable1.getSelectedRow();
    node = this.phn.Search_Node(tabelModel_1.getValueAt(selectedRow, 1).toString());
    txtKata.setText(node.getData());
    txtArti.setText(node.getArti());

    btn_hapus.setEnabled(true);
    btn_edit.setEnabled(true);
}

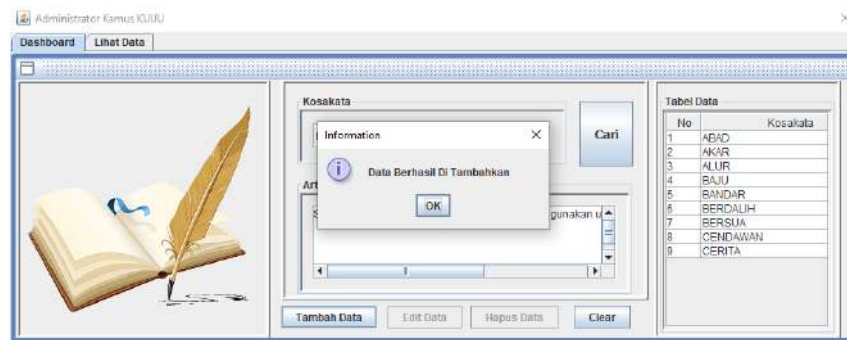
private void btn_clearActionPerformed(java.awt.event.ActionEvent evt) {
    reset();
    jTable1.clearSelection();
}
```

#### b. Penambahan kata baru

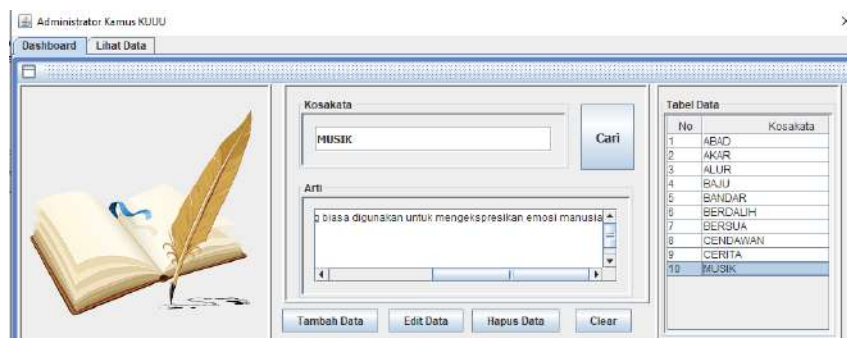
Berikut adalah langkah-langkah untuk menambahkan kata serta arti baru ke dalam kamus. Pertama, isi kolom kosa kata dengan kata baru, misalkan “Musik” beserta artinya pada kolom arti.



Selanjutnya klik button “Tambah Data” untuk melanjutkan proses. Akan muncul pesan pop yang menginformasikan ke admin bahwa kata telah berhasil ditambahkan ke kamus.



Kata baru akan ditampilkan pada tabel sebelah kanan dan ketika dilakukan *double* klik, kolom kosakata akan diisi dengan kata “Musik” dan kolom arti diisi dengan arti dari kata tersebut.



Penambahan kata ke kamus dapat dilakukan dengan memanggil event handling ketika button “Tambah Data” diklik. Pada method tersebut, terdapat pemanggilan method `tambahData()`, dilanjutkan dengan `reset()` untuk mengosongkan kolom kosa kata dan kolom arti lalu terakhir method `tampilData_1()`.

```
private void btn_tambahActionPerformed(java.awt.event.ActionEvent evt) {
    tambahData();
    reset();
    tampilData_1();
}
```

Pada method `tambahData()`, kolom kosa kata (`txtKata`) dan kolom arti (`txtArti`) akan dilakukan pengecekan. Jika keduanya kosong (bernilai “”), maka sebuah exception akan di throw dengan pesan bahwa kolom kata dan kolom arti tidak boleh kosong. Jika pengecekan pertama tidak terpenuhi, maka kedua kolom telah diisi dan kata pada kolom kosakata akan disimpan ke variabel temp yang bertipe `String`. Selanjutnya, akan dilakukan pencarian pada tree dengan memanggil method `Search_Node()` yang berargumen temp. Hasil pengembalian dari method tersebut akan disimpan pada variabel bernama `node` yang bertipe `TreeNode`.

```

public void tambahData() {
    try {
        if (txtKata.getText().equalsIgnoreCase("")
            && txtArti.getText().equalsIgnoreCase("")) {
            throw new Exception("Kolom kata dan kolom arti tidak boleh kosong");
        } else {
            String temp = txtKata.getText().toUpperCase();
            node = this.phn.Search_Node(temp);
        }
    }
}

```

Langkah berikutnya, yaitu melakukan pengecekan pada node, dimana jika node bernilai null, kata belum ada pada kamus. Di sini, temp akan diperiksa ulang, apakah bernilai null, jika tidak, maka variabel temp akan disimpan ke variabel kosakata yang bertipe String. Untuk arti dari kata, panggil method `getText` dari variabel `txtArti` dan simpan hasil pengembalian ke variabel pengertian yang bertipe String. Untuk memasukkan kata dan arti ke tree, panggil method `insert` dengan argumen berupa kosakata dan pengertian. Jika proses insert berhasil, maka panggil method `JOptionPane` untuk menampilkan pesan bahwa kata baru telah ditambahkan. Sementara itu, untuk exception yang telah di throw, akan dilakukan pemanggilan `JOptionPane` untuk menampilkan peringatan dengan isi pesan disesuaikan pada tiap exception.

```

        if (node == null) {
            if (temp == null) {
                throw new Exception("Kolom Kosakata Tidak Boleh Kosong");
            } else {
                String kosakata = temp;
                String pengertian = txtArti.getText();
                phn.insert(kosakata, pengertian);
                JOptionPane.showMessageDialog(this, "Data Berhasil Di Tambahkan",
                    "Information", JOptionPane.INFORMATION_MESSAGE);
            }
        } else {
            throw new Exception("Kata sudah tersedia dalam daftar");
        }
    }

    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
            JOptionPane.WARNING_MESSAGE);
    }
}

```

Proses penambahan data baru atau kata baru akan dilanjutkan dengan melakukan pembaharuan pada tabel dengan memanggil method `tampilData_1()`. Pada method ini, objek `DaftarList` akan dihapus dengan memanggil method `clear`, begitu pula objek `daftarList`. Untuk menampilkan data dilakukan kunjungan pada tiap node dengan memanggil method `inOrder` dari objek `phn` yang bertipe `Tree`. Kemudian inisiasi `daftarList` dengan hasil return dari method `getListData` objek `phn`.

```

public void tampilData_1() {
    this.daftarList.clear();
    this.phn.listData.clear();
    this.phn.inOrderTraversal();
    this.daftarList = this.phn.getListData();
}

```



Lanjutkan proses dengan membuat header dari tabel serta inisiasi objKata bertipe Object yang akan menjadi atribut pada tabel. Lakukan perulangan untuk menginputkan data pada daftarList ke objKata.

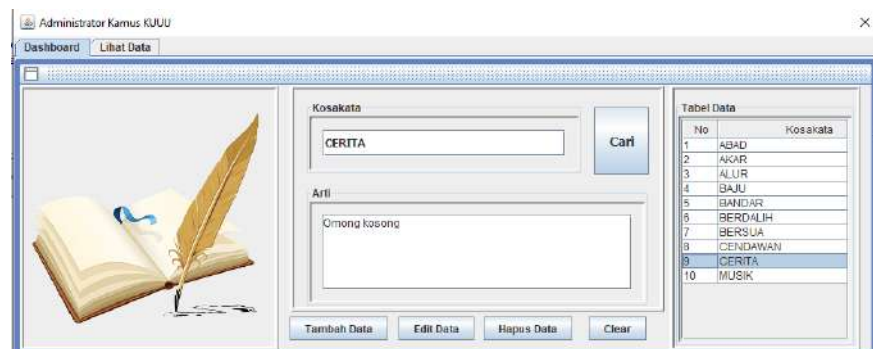
```
String[] header = {"No", "Kosakata"};
Object[][] objKata = new Object[daftarList.size()][2];
int i = 0;
int j = 1;
for (TreeNode n : daftarList) {
    if (n instanceof TreeNode) {
        String[] arrData = {String.valueOf(j++), n.getData()};
        objKata[i] = arrData;
    }
    i++;
}
```

Pada proses di bawah ini, atur agar.JTable berisikan data dari objKata dengan header yang telah didefinisikan.

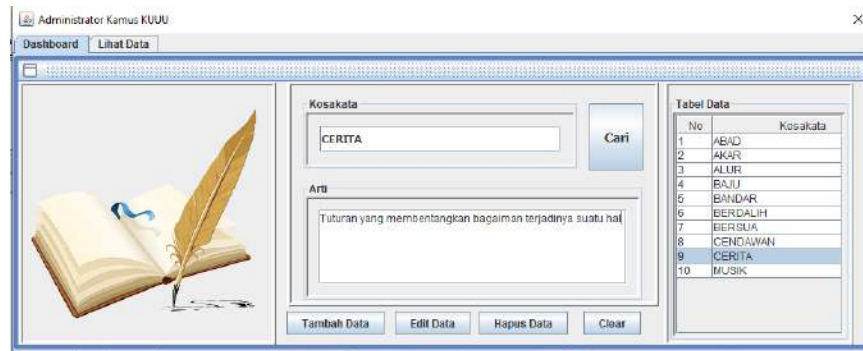
```
tabelModel_1 = new DefaultTableModel(objKata, header) {
    //membuat tabel tidak bisa diedit saat diklik
    @Override
    public boolean isCellEditable(int rowIndex, int colIndex) {
        return false;
    }
};
jTable1.setModel(tabelModel_1);
columnModel_1 = jTable1.getColumnModel();
columnModel_1.getColumn(0).setMinWidth(40);
columnModel_1.getColumn(1).setMinWidth(200);
}
```

### c. Pengeditan arti kata

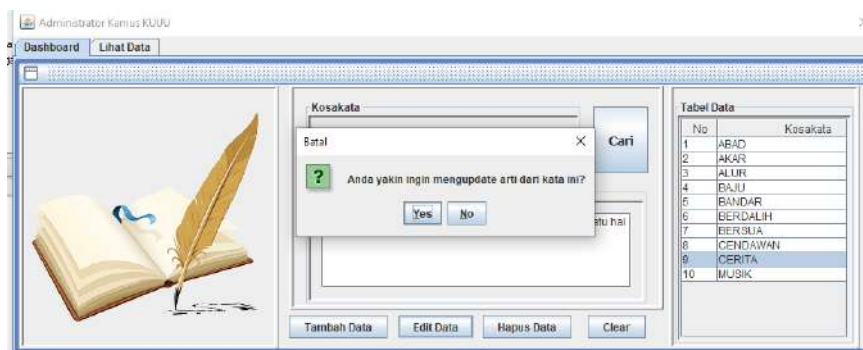
Berikut adalah langkah-langkah melakukan pengeditan pada kata “Cerita” di aplikasi kamus. Pertama, cari atau klik kata pada tabel, kata yang akan diedit.



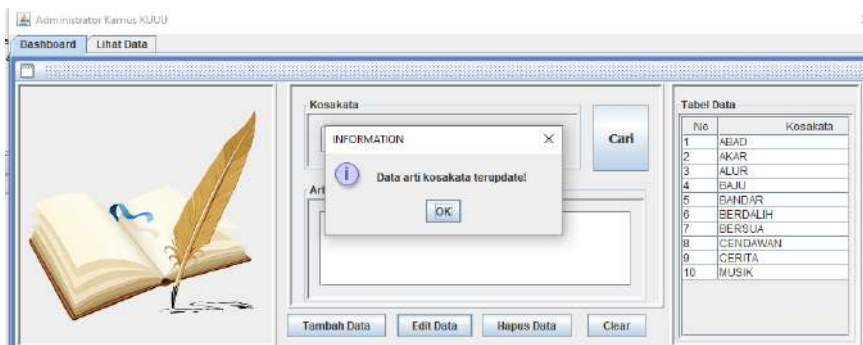
Selanjutnya, tuliskan arti baru pada kolom arti dan klik button “Edit Data”.



Akan muncul pesan *pop up* konfirmasi terkait pengeditan data. Jika yakin untuk mengedit data maka klik button “Yes”.



Jika data berhasil diedit, akan muncul pesan *pop up* pemberitahuan bahwa data telah di edit, user dapat mengklik “OK”.



Pengeditan kata pada kamus dapat dilakukan dengan memanggil event handling dari button ‘Edit Data’. Pertama akan dilakukan pemanggilan objek JOptionPane dengan memanggil dialog untuk memberikan konfirmasi pengeditan. Pilihan dari pengguna akan disimpan pada variabel jawab yang bertipe integer.

```
private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
    int jawab = JOptionPane.showOptionDialog(this,
        "Anda yakin ingin mengupdate arti dari kata ini?",
        "Batal",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, null, null);
}
```

Selanjutnya, definisikan variabel temp yang bernilai teks dari textfield txtKata serta variabel tempArti yang menyimpan isi text dari textfield txtArti. Variabel jawab akan dicek dan ketika jawaban yang dipilih 'Yes', akan dilanjutkan ke proses pengeditan. Sebelum arti kata diedit, akan dicek apakah textfield txtKata kosong atau tidak. Jika kosong akan dilempar exception yang menampilkan pesan bahwa kolom kosakata masih kosong.

```
String temp = txtKata.getText().toUpperCase();
String tempArti = txtArti.getText();

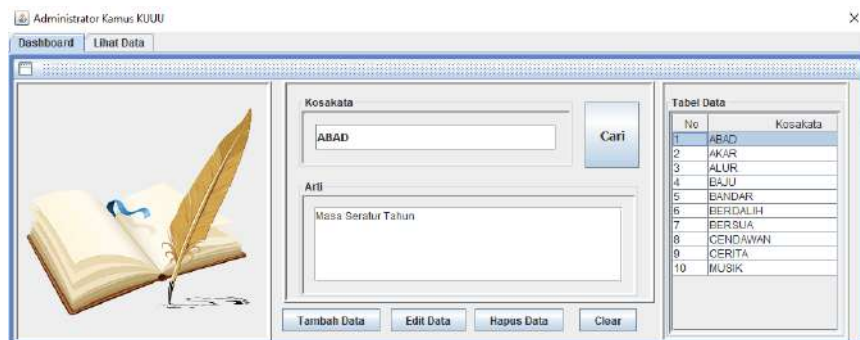
if (jawab == JOptionPane.YES_OPTION) {
    try {
        if (txtKata.getText().isEmpty()) {
            throw new Exception("Kolom Kosakata Masih Kosong");
        }
        this.node = this.pbm.Search_Node(temp);
        if (node == null) {
            throw new Exception("Tidak ada kosakata yang sama dengan field kosakata "
                + "\n"
                + "Data arti dari kosakata tidak dapat di Update");
        } else {
            node.setData(temp);
            node.setArti(tempArti);
        }
        reset();
        jTable1.clearSelection();
        tampilData_1();
        JOptionPane.showMessageDialog(this, "Data arti kosakata terupdate!", "INFORMATION",
            JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
            JOptionPane.WARNING_MESSAGE);
    }
}
```

Jika kolom text field tidak kosong, maka akan dilakukan pencarian pada tree dengan parameter berupa temp dan hasil pengembalian disimpan pada sebuah variabel bertipe TreeNode yaitu node. Pengecekan dengan try catch dilakukan kembali pada objek node jika mengembalikan null atau data tidak ada pada tree. Ubah data pada node menjadi temp dan arti menjadi tempArti untuk mengedit data. Hal tersebut dilakukan jika node tidak bernilai null dan data dapat diedit sesuai dengan input baru.

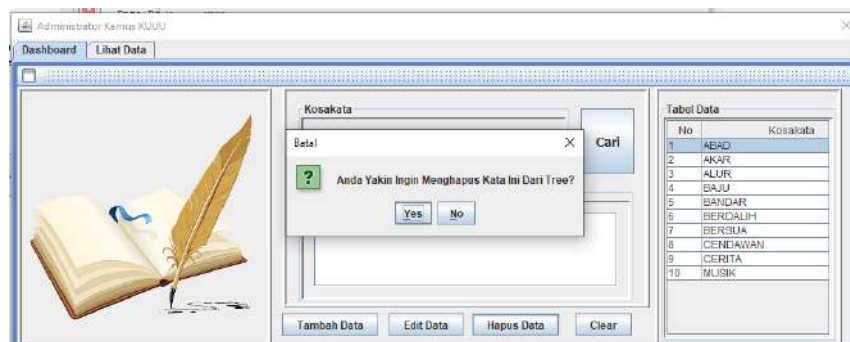
Langkah selanjutnya adalah memanggil method reset( ) untuk mengosongkan kolom textfield kata dan arti. Setelah itu, panggil method clearSelection dari objek.JTable1 untuk membersihkan tabel lalu tampilkan data baru yang telah diedit pada tabel juga. Proses pengeditan selesai dan akan muncul pesan pop yang menginformasikan bahwa pengeditan arti kata telah berhasil.

#### d. Penghapusan kata

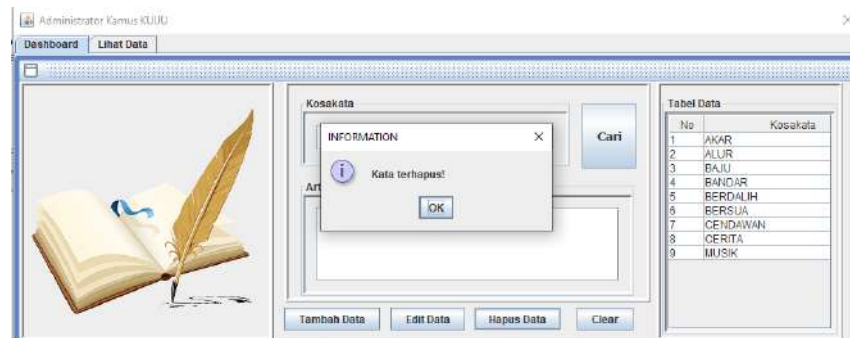
Berikut adalah langkah-langkah melakukan penghapusan pada kata “Abad” di aplikasi kamus. Pertama, cari atau klik kata pada tabel, kata yang akan diedit.



Jika data ditemukan, selanjutnya klik button 'Hapus Data' untuk menghapus kata yang telah dipilih dan selanjutnya akan muncul pesan pop up konfirmasi seperti di bawah ini.



Jika data berhasil terhapus, maka akan muncul pop up informasi bahwa data telah berhasil terhapus.



Penghapusan data dilakukan dengan memanggil event handling dari button 'Hapus Data' yang akan memanggil objek JOptionPane untuk menampilkan pesan konfirmasi. Hasil jawaban dari pesan konfirmasi akan disimpan pada variabel jawab yang bertipe integer.

```
private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
    int jawab = JOptionPane.showOptionDialog(this,
        "Anda Yakin Ingin Menghapus Kata Ini Dari Tree?",
        "Batal",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, null, null);
}
```

Jika jawaban yang diberikan yaitu 'Yes', maka proses penghapusan dapat dilanjutkan yaitu dengan menyimpan isi dari kolom textfield txtKata ke sebuah variabel temp yang bertipe String. Kolom textfield txtKata akan dicek kembali apakah kosong atau tidak, jika kosong akan mengembalikan exception bahwa kolom kosakata masih kosong. Jika tidak maka penghapusan akan dilakukan dengan memanggil method Search\_Node dengan parameter berupa temp dan hasil pengembalian ditaruh di variabel node.

Jika pengembalian berupa null, maka kata yang akan dihapus tidak ada di tree, maka dari itu, proses tidak dapat dilanjutkan. Namun, jika kata ditemukan (node tidak sama dengan null), hapus kata dengan memanggil method hapus dari objek phn dan parameternya berupa node.

```
private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
    int jawab = JOptionPane.showOptionDialog(this,
        "Anda Yakin Ingin Menghapus Kata Ini Dari Tree?",
        "Batal",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, null, null);

    String temp = txtKata.getText().toUpperCase();

    if (jawab == JOptionPane.YES_OPTION) {
        try {
            if (txtKata.getText().isEmpty()) {
                throw new Exception("Kolom Kosakata Masih Kosong");
            }
            node = this.phn.Search_Node(temp);
            if (node == this.phn.getRoot()) {
                throw new Exception("Root Tidak Dapat Di Hapus \n "
                    + "Silahkan Tambah satu Data Jika Ingin Menghapus "
                    + "Data yang ada di \"Root\"");
            } else {
                this.phn.hapus(node);
            }
        }
    }
}
```

Selanjutnya, pastikan tabel yang ada diperbarui dengan memanggil tampil\_data1() dan kosongkan kedua textfield dengan method reset().

```
System.out.println(txtKata.getText());
reset();
jTable1.clearSelection();
tampilData_1();
JOptionPane.showMessageDialog(this, "Kata terhapus!", "INFORMATION",
    JOptionPane.INFORMATION_MESSAGE);
} catch (Exception e) {
    JOptionPane.showMessageDialog(this, e.getMessage(), "Warning",
        JOptionPane.WARNING_MESSAGE);
}
}
```

## 5. Gui\_Tree

Gui\_tree merupakan suatu kelas yang meng-extends dari kelas tree sehingga kelas ini dapat memanfaatkan fungsi yang berada dalam kelas tree. Kelas ini dimanfaatkan sebagai tempat penyimpanan data, Gui\_Tree juga dapat melakukan penghapusan data.



Gui\_tree dalam admin digunakan sebagai suatu objek yang dapat menampung beberapa data dalam kamus sebagai penyimpanan dengan memanfaatkan arraylist dengan berbagai macam fungsinya seperti menghapus, add data dan juga mencari data. Selain menampung data kamus, objek ini juga digunakan sebagai menyimpan data pengguna admin seperti menyimpan password dan juga username. Pada submenu Guest Gui\_tree digunakan untuk mencari suatu data yang dicari oleh user dan menampilkannya.

```
import java.util.ArrayList;

public class GUI_Tree extends Tree{
    private boolean kondisi = false;
    ArrayList<TreeNode> daftarList = new ArrayList<>();

    public void hapus(TreeNode node){
        System.out.println("node \n" + node.getData() + ":" + node.getArti());
        boolean cek = false;
        if (node != null) {
            if (!isLeaf(node))
                && (node.getLeftNode() != null && node.getRightNode() != null) {
                    cek = delete_2_anak(node);
                } else if (!isLeaf(node)
                    && (node.getLeftNode() != null || node.getRightNode() != null)) {
                    cek = delete_1_anak(node);
                } else {
                    cek = delete_0_anak(node);
                }
        }
        System.out.println(cek);
    }

    public boolean isKondisi() {
        return kondisi;
    }

    public void setKondisi(boolean kondisi) {
        this.kondisi = kondisi;
    }
}
```

Untuk penghapusan data, terdapat method hapus( ) dengan parameter berupa TreeNode dan akan mengecek node yang menjadi argumen selama node tidak null. Pengecekan dilakukan dengan melihat apakah node merupakan leaf, memiliki cabang atau tidak. Dimana jika node bukan leaf dan memiliki cabang kiri serta cabang kanan, akan dipanggil method delete\_2\_anak( ). Jika node bukan leaf dan salah satu cabang tidak kosong akan dipanggil delete\_1\_anak( ). Terakhir, jika kedua syarat tidak terpenuhi, akan dipanggil method delete\_0\_anak( ).

## F. PENJELASAN KHUSUS

### 1. TreeNode

Proyek ini memilih konsep binary search tree sehingga dibutuhkan suatu kelas yang dapat membentuk sebuah Node. Kelas TreeNode pun dibuat dengan atribut berupa data dan arti yang keduanya bertipe String. Atribut data akan menyimpan informasi berupa kata sedangkan atribut arti akan menyimpan arti dari kata tersebut. Selain itu, terdapat atribut leftNode yang akan menyimpan informasi mengenai left child dari sebuah node. Terdapat rightNode juga untuk menyimpan informasi dari right child dari node. Sedangkan pada atribut parent, data yang akan disimpan adalah parent atau induk dari sebuah node.

```
public class TreeNode {
    private String data;
    private String arti;
    TreeNode leftNode;
    TreeNode rightNode;
    TreeNode parent;
}
```

Ketika sebuah object `TreeNode` dideklarasikan dan diinisialisasi, dibutuhkan parameter berupa data dan arti yang bertipe `String`. Dimana kedua argumen akan menjadi atribut dari node tersebut. Jika hanya diberikan satu argumen berupa `String`, argumen akan disimpan sebagai atribut kata. Namun, jika object diinisialisasi tanpa adanya parameter yang terisi, baik data, kata maupun childnya akan diset menjadi `null`.

```
public TreeNode(String data, String arti) {
    this.data = data;
    this.arti = arti;
}

public TreeNode(String data) {
    this.data = data;
}
```

Untuk dapat mengakses tiap atribut dalam object `TreeNode`, dibutuhkan setter dan getter. Dengan demikian, atribut penting dapat dipanggil serta diedit dengan lebih mudah.

## 2. Tree

Selain membutuhkan `TreeNode` sebagai node nya proyek ini juga membutuhkan `Tree` yang dimana untuk membuat suatu objek tree yang memiliki beberapa atribut, terdapat list data yang bertipe `ArrayList` dari Object `TreeNode` yang digunakan untuk menyimpan data dari tree yang sudah dibuat kemudian ditampilkan ke dalam tabel data, kemudian terdapat atribut `root`, `parent` dan `temp` yang bertipe `TreeNode`, kemudian terdapat atribut `hasilPencarian_kata` dan `hasilPencarian_arti` yang bertipe `String`.

```
ArrayList<TreeNode> listData = new ArrayList<>();
TreeNode root, temp;
String dataPrint = "";
TreeNode parent;
static int jumlahTree = 0;
String hasilPencarian_kata, hasilPencarian_arti;
```

Ketika sebuah object `Tree` dideklarasikan dan diinisialisasikan, dibutuhkan satu parameter bertipe `TreeNode`. Argumen tersebut akan mengisi atribut `root` dari object `Tree` tersebut. Namun, jika object dibuat tanpa mengisi argumen, `root` akan diset nilai

nya menjadi null. Dan juga terdapat deklarasi saja pada class ini maka atribut lainnya bernilai null, karena hanya dideklarasikan saja tidak diinisialisasi.

Pada kelas Tree terdapat method insert yang digunakan untuk memasukkan node baru ke dalam object. Method ini memiliki parameter bertipe String yang berkaitan dengan dataNode atau kata serta arti. Method bekerja dengan mengecek terlebih dahulu apakah binary search tree kosong atau tidak, jika pengecekan menghasilkan true, maka node yang pertama masuk akan dijadikan root. Namun, jika pengecekan menghasilkan false, method insertHelper akan dipanggil dengan parameter berupa root dari BST, data kata serta data arti.

Pada method insertHelper ini, akan dilakukan pemanggilan secara rekursif dengan melakukan perbandingan. Perbandingan tersebut dilakukan antara kata pada parameter dengan kata pada node sementara menggunakan method compareTo. Jika hasil yang diberikan lebih kecil dari nol, maka method insertHelper dipanggil kembali untuk mengecek perbandingan yang sama pada left subtree sampai menemui leaf. Jika titik tersebut telah tercapai, maka set left child dari left subtree menjadi node yang baru masuk. Sementara itu, jika hasil pemanggilan method compareTo bernilai lebih besar dari nol, method insertHelper dipanggil kembali untuk mengecek perbandingan yang sama pada right subtree sampai menemui leaf. Jika titik tersebut telah tercapai, maka set right child dari right subtree menjadi node yang baru masuk.

Method lain dari kelas ini adalah InOrderTraversal( ), yang digunakan untuk mengunjungi seluruh node pada BST dengan cara in-order, method ini memiliki tujuan yang sama dengan method sebelumnya, yaitu mengunjungi seluruh node pada BST, tetapi dilakukan dengan cara in order. Sebelum melakukan kunjungan, arrayList dengan nama listData dikosongkan terlebih dahulu. Selanjutnya method ini akan memanggil inOrderHelper. Dibutuhkan parameter berupa root agar method ini dapat berjalan karena dilakukan trapping null untuk memastikan BST tidak kosong. Setelah itu, dilakukan kunjungan pada subtree kiri dengan memanggil method ini secara rekursif dimana argumen bernilai left Child dari root lalu root akan dimasukkan ke listData dengan memanggil method add data dari root akan diprint setelahnya. Subtree kanan akan dikunjungi selanjutnya dengan memanggil inOrderHelper dan argumen diisi dengan right Child dari root. Ini akan menghasilkan kunjungan yang terurut yang penting digunakan dalam proses menampilkan isi dari kamus.

Kemudian method InOrderTraversal\_reverse( ), yang digunakan untuk mengunjungi seluruh node pada BST dengan cara in-order\_reverse, method ini memiliki tujuan yang sama dengan method sebelumnya, yaitu mengunjungi seluruh node pada BST, tetapi dilakukan dengan cara in order reverse. Sebelum melakukan kunjungan, arrayList dengan nama listData dikosongkan terlebih dahulu. Selanjutnya method ini akan memanggil inOrderHelper\_Reverse. Dibutuhkan parameter berupa root agar method ini dapat berjalan karena dilakukan trapping null untuk memastikan BST tidak kosong. Setelah itu, dilakukan kunjungan pada subtree kanan dengan memanggil method ini secara rekursif dimana argumen bernilai right Child dari root

lalu root akan dimasukkan ke listData dengan memanggil method add data dari root akan diprint setelahnya. Subtree kanan akan dikunjungi selanjutnya dengan memanggil inOrderHelper\_Reverse dan argumen diisi dengan left Child dari root. Ini akan menghasilkan kunjungan yang terurut dari urutan Z-A yang penting digunakan dalam proses menampilkan isi dari kamus.

Method Search\_Node memiliki peran penting, terutama pada penghapusan sebuah node. Pertama, deklarasikan sebuah objek dengan tipe TreeNode bernama pointer yang nantinya digunakan sebagai penunjuk. Selanjut, buat pointer agar menunjuk ke node root. Jika node root memiliki data yang sama dengan apa yang menjadi argumen pada method ini, ubah nilai pointer menjadi root. Jika kondisi tersebut tidak terpenuhi, lakukan perulangan yang akan terus dilakukan selama data pointer tidak sama dengan data argumen. Pada tiap perulangan juga dilakukan pengecekan apakah data yang ditunjuk pointer memiliki nilai yang kurang dari data argumen, jika benar, ubah pointer menjadi right node dari node pointer tersebut. Sebaliknya, jika data pointer bernilai lebih besar dari data yang dicari, ubah pointer menjadi left child dari node pointer. Jika pointer yang ditemukan bernilai null, maka data yang dicari tidak ada pada tree.

Method selanjutnya digunakan untuk mengecek apakah sebuah node merupakan leaf atau bukan. Pengecekan dilakukan dengan mengecek apakah left child serta right child dari node bernilai null atau tidak. Jika benar, maka node tersebut adalah node leaf dan akan mengembalikan nilai true, begitu juga sebaliknya.

Salah satu method penting dari BST adalah delete( ). Argumen yang dimasukkan pada method adalah data dari node yang ingin dihapus. Pertama inisialisasi variabel check yang bertipe boolean lalu dilakukan pencarian node pada tree dengan method Search\_node. Jika node ditemukan, cek apakah node yang ingin dihapus merupakan node leaf atau bukan dengan memanggil method isLeaf. Jika node yang ingin dihapus adalah node leaf, panggil method delete\_0\_anak, sebaliknya jika isLeaf mengembalikan false, serta salah satu childnya bernilai null. panggil delete\_1\_anak. Jika kedua child tidak bernilai null dan node bukan leaf, panggil method delete\_2\_anak.

Berikut akan dijelaskan terkait penghapusan node 0 anak, dimana akan terlebih dahulu dicek, apakah node yang akan dihapus merupakan left child atau right child dari parentnya. Pengecekan dilakukan dengan method getLeftNode dan datanya akan dibandingkan dengan argumen. Jika keduanya memiliki nilai yang sama, set agar left child dari parent node yang akan dihapus menjadi null dan ubah agar node tersebut tidak lagi mengarah ke parent (set null). Pengubahan ini akan otomatis memutuskan hubungan antara node dengan parent dan menghapus node dari tree. Untuk node yang merupakan right child dari parentnya, lakukan hal yang sama, set parent menjadi null dan jadikan right child dari parent menjadi null. Jika salah satu dari pengecekan tersebut terjadi, kembalikan nilai true.

Untuk penghapusan 1 anak, pertama dicek apakah node yang akan dihapus merupakan root dari tree atau bukan. Jika node bukan root, cek kembali, apakah node yang akan dihapus merupakan left child atau right child. Untuk node yang merupakan left child, set agar child dari node menjadi child dari parent node yang dihapus (dilakukan pengecekan kembali node memiliki child kiri atau kanan). Hal serupa dilakukan jika node yang akan dihapus merupakan right child. Namun, jika node yang dihapus adalah root, cek terlebih dahulu, cabang sebelah mana yang tidak kosong dari node tersebut. Jika node memiliki right child, deklarasikan sebuah objek bertipe `TreeNode` yang diisi dengan hasil pemanggilan method `GetSuccessor` dengan argumen berupa node yang akan dihapus. `GetSuccessor` akan mengembalikan successor dari root dan selanjutnya root akan diset menjadi node successor. Namun, jika successor bernilai null, set root menjadi node dengan nilai paling tinggi dari subtree kanannya. Sebaliknya jika node memiliki left child, deklarasikan sebuah objek bertipe `TreeNode` yang isinya merupakan hasil pengambilan dari method `getPredecessor()`. Method tersebut akan mengembalikan node yang merupakan predecessor dari root. Node hasil pemanggilan predecessor dapat bernilai null dan jika demikian, ubah agar root menjadi left child atau node dengan nilai paling kecil pada subtree kiri.

Penghapusan dua anak dibagi menjadi dua, yaitu node yang akan dihapus merupakan root serta bukan root. Jika node yang dihapus bukan root, cek terlebih dahulu, node yang akan dihapus merupakan left child atau right child dari parentnya. Selanjutnya, jika node tersebut adalah right child dari parent, dilakukan pengecekan ketiga, node itu sendiri memiliki right child atau left child. Jika memiliki right child, set atau ubah right child dari parent node menjadi child node tersebut. Untuk node yang memiliki left child lakukan hal serupa tapi dengan left child dari node. Jika node yang dihapus merupakan left child, panggil method `GetSuccessor` jika right child dari node yang akan dihapus adalah node leaf. Selanjutnya ubah agar node successor menempati tempat node yang dihapus. Untuk node yang akan dihapus merupakan root, pertama cari terlebih dahulu predecessor dari root dan ubah root menjadi node predecessor.

Selanjutnya, akan dijelaskan method `GetSuccessor` yang digunakan pada saat penghapusan node dengan 1 anak dan node dengan 2 anak. Pada bagian awal method ini, inisiasi beberapa variabel yaitu `successor` yang akan menunjuk ke node sementara, `successor parent` dan `current` yaitu right child dari node sementara. `Current` bergeser terus menuju left child dari node sementara, `successor` akan menyimpan nilai dari `current` sebelumnya dan `successorParent` akan menyimpan nilai dari `successor` sebelumnya. Operasi ini dilakukan sampai `current` bernilai null atau left child dari node sementara tidak ada. Setelah keluar dari pengulangan, `successor` akan bernilai nilai terkecil pada subtree kanan node yang dihapus.

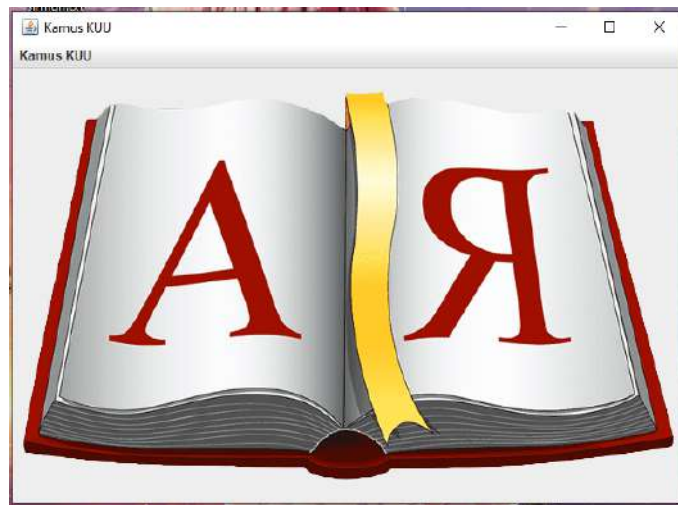
Terakhir, method `GetPredecessor` yang akan mengembalikan node dengan nilai tertinggi dari subtree kiri. Pada bagian awal method ini, inisiasi beberapa



variabel yaitu successor yang akan menunjuk ke node sementara, successor parent dan current yaitu left child dari node sementara. Current bergeser terus menuju right child dari node sementara, successor akan menyimpan nilai dari current sebelumnya dan successorParent akan menyimpan nilai dari successor sebelumnya. Operasi ini dilakukan sampai current bernilai null atau right child dari node sementara tidak ada. Setelah keluar dari pengulangan, predecessor akan bernilai nilai terbesar pada subtree kiri node yang dihapus.

## G. CAPTURE OUTPUT

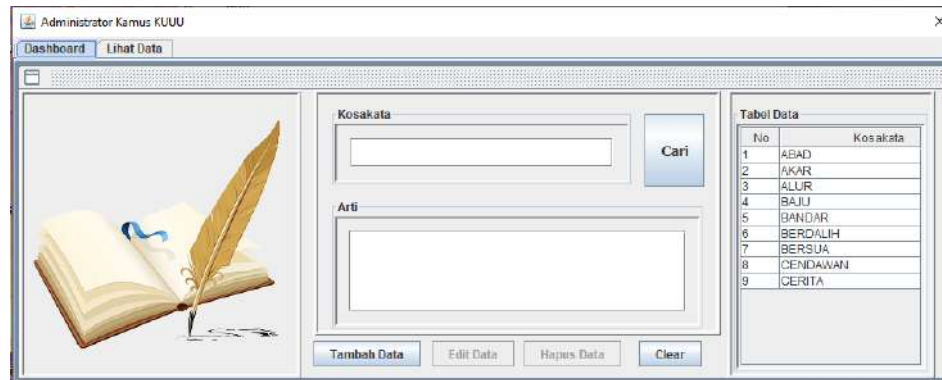
### - Menu Utama



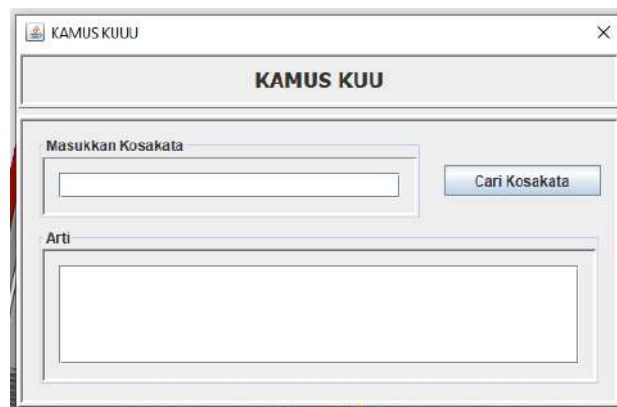
### - Login Admin



### - Menu Admin



## - Menu Guest



## H. KESIMPULAN

Kesimpulan dari pembahasan aplikasi di atas, bahwa konsep *binary search tree* dapat diterapkan untuk mencari, menyusun, menyimpan serta menampilkan kata secara terurut. Pencarian sebuah kata mengimplementasikan method search dari tree, sementara penyusunan dan penyimpanan kata menggunakan method insert. Penampilan data dengan method traversal secara in-order, dan in-order reverse. Selain itu, dengan memanfaatkan GUI Java, pengguna dapat mencari arti suatu kata dengan dengan lebih interaktif, efektif dan efisien. Fitur edit pada admin digunakan sebagai bentuk pengembangan lain dari sebuah kamus sehingga arti dari tiap kata dapat terus diperbarui.

## I. LAMPIRAN

Link Source Code : <https://github.com/boim212/Kamus-Kuuu>

[Pengertian Bahasa, Peran & Fungsi Bahasa secara Umum di Masyarakat \(tirto.id\)](https://tirto.id/Pengertian-Bahasa-Peran-&Fungsi-Bahasa-secara-Umum-di-Masyarakat/) diakses pada 19 April 2022

[Arti kata kamus - Kamus Besar Bahasa Indonesia \(KBBI\) Online](https://www.kbbi.kemdiknas.go.id/) diakses pada 19 April 2022