

Penerapan Struktur Data Tree menggunakan Array Pada Bahasa JAVA menunjang Logika Cerdas untuk menampilkan Garis Keturunan

Didik Nugroho, masDidikhoho@sinus.ac.id, STMIK Sinar Nusantara
Sri Tomo, szrie@sinus.ac.id, STMIK Sinar Nusantara
Arie Febrianto, ariefebrianto @sinus.ac.id, STMIK Sinar Nusantara

ABSTRAKSI

Struktur data merupakan proses bagaimana data akan tersimpan dan terkelola didalam memori komputer, beberapa permasalahan dalam kehidupan nyata dapat terkomputerisasi dengan baik dilakukan dengan bagaimana penanganan struktur datanya. Penggunaan stuktur data yang tepat dapat dengan mudah dilakukan penentuan algoritma pelacakannya. Salah satu model struktur data yang dapat diterapkan pada silsilah keluarga berupa struktur data berbentuk model tree. Model tree ini yang bisanya dalam pelajaran dicontohkan dalam bentuk tree biner, dalam penelitian ini dibuat untuk mengelola silsilah keluarga jawayang jarang dibahas dalam pembelajaran di perkuliahan. Metode penelitian dilakukan dengan membuat sampel-sampel silsilah keluarga, kemudian menyusun algoritma cerdas untuk mencari anggota sesuai istilah kekeluargaan suku Jawa. Penerapan struktur data tree yang dibuat dalam program Java dengan menyimpan dalam array, program silsilah keluarga ini dapat digunakan dan dikembangkan untuk membuat sistem pengelolaan silsilah keluarga serta tujuan edukasi bagi tim pengembang sistem informasi, khususnya edukasi terkait ilmu struktur data tree serta logika cerdas.

Kata kunci : Struktur Data Tree, Bahasa Java, Array, silsilah keluarga Logika Cerdas

1. LATAR BELAKANG MASALAH

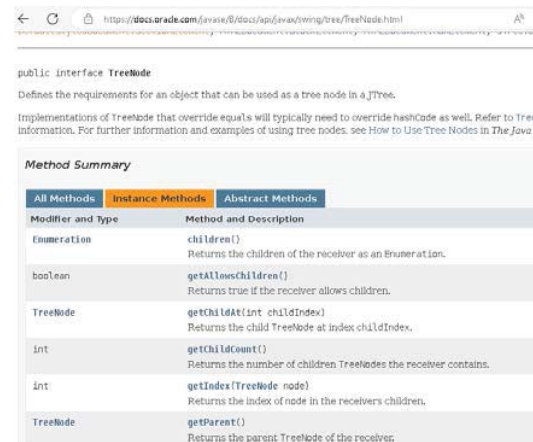
Struktur data merupakan proses bagaimana data akan tersimpan dan terkelola didalam memori komputer. Permasalahan struktur data diantaranya *List*, *stack*, *array*, *graph*, dan *Tree*. Masing masing materi bahasan struktur data dapat dikombinasikan untuk dapat di manfaatkan untuk persoalan-persoalan tertentu yang ada di masyarakat.

Materi struktur data diatas dalam implementasinya juga akan tergantung dengan bahasa pemrograman yang digunakan. Misalnya sintak bahasa Pascal akan berbeda dengan bahasa C, C# atau bahasa Java tiap bahasa program akan berbeda sintaknya. Namun secara alur logika materi stuktur data tersebut tidak akan berbeda antar bahasa program.

Salah satu materi struktur data yang menarik diajarkan ke mahasiswa Teknik Informatika yaitu struktur data *tree*, menggunakan bahasa Java.

Struktur data *tree* dalam bahasa Java telah disediakan interface nya. Interface tree yang ada di bahasa Java, berdasarkan oracle berbentuk Class,

objek jadi menerapkan model *OOP* (*object oriented programming*).



Gambar 1. Materi interface dan method struktur *Tree* pada laman `javax.swing.tree` Class Hierarchy (Java Platform SE 8 oracle)

Model pemrograman *Tree* dalam bahasa Java, biasanya dilakukan dengan membuat class *Node*, serta memanfaatkan *List*, Terbentuk :

```
private static class Node {  
    String value;  
    List<Node> children;
```

```
Node(String value) {  
    this.value = value;  
    this.children = new ArrayList<>();  
}
```

Permasalahan di lapangan masih banyak mahasiswa yang belum mampu menggunakan model OOP, memanfaatkan interface tree yang telah disediakan oleh compiler Java, seperti Java Neatbean untuk mengelola permasalahan strukturnya berbentuk Tree. Berdasarkan kajian peneliti dari 36 mahasiswa yang belajar pemrograman OOP masih kesulitan penerapan interface Tree yang ada di Java Mahasiswa lebih cenderung paham pada struktur data berjenis larik seperti Array, List yang ada di bahasa java [2].

Hasil penelusuran di lakukan dengan wawancara terdapat urutan penyebutan keturunan didalam trah jawa(Jawa Tengah) sebagai berikut :

1. Keturunan ke-1. Anak
2. Keturunan ke-2. Putu/cucu
3. Keturunan ke-3. Buyut /cicit
4. Keturunan ke-4. Canggih
5. Keturunan ke-5. Wareng
6. Keturunan ke-6. Udhek-udhek
7. Keturunan ke-7. Gantungsiwur
8. Keturunan ke-8. Cicipmoning
9. Keturunan ke-9. Petaranganbobrok
10. Keturunan ke-10. Gropaksenthe
11. Keturunan ke-11. Gropakwaton
12. Keturunan ke-12. Cendheng
13. Keturunan ke-13. Giyeng
14. Keturunan ke-14. Cumpleng
15. Keturunan ke-15. Ampleng
16. Keturunan ke-16. Menyaman
17. Keturunan ke-17. Menya-menya
18. Keturunan ke-18. Trahtumerah.

Selain urutan juga ada penyebutan istilah kekeluargaan lain :

1. Mbah kakung /Mbah Putri
2. Bapak / Bopo Ibu /Simbok /make
3. Pakdhe /Budhe
4. Paklek / Bu lek

Hal ini yang menarik untuk dilakukan penelitian bagaimana menerapakan struktur data berbentuk larik, dalam hal ini array untuk menyimpan data – data yang memiliki struktur hirarkhi seperti kasus pada silsilah keluarga.

2. RUMUSAN MASALAH

Permasalahan yang telah diungkapkan pada sub bab pendahuluan yaitu bagaimana bisa menyimpan struktur silsilah keluarga menggunakan variabel bertipe array. Serta menemukan alur mencari orang sesuai hubungan kekerabatan di jawa yaitu Kekek dan Nenek, Ayah dan ibu, pak lek dan bu lek, pak dhe dan Bu dhe, dan anak.

3. METODE PENELITIAN

Metode penelitian yang dilakukan dengan tahapan penyiapan data, kemudian analisis terhadap data, kemudian melakukan rancangan , implementasi dan pengujian.

Data penelitian ini data primer data silsilah keluarga serta hubungan kekeluargaan diperoleh dari wawancara dan observasi dilingkungan masyarakat Solo dan Sekitarnya .

Dalam penelitian ini data sekunder diperoleh dari informasi tertulis atau hasil dokumentasi pada karya tulis sebelumnya serta mempelajari referensi mengenai tree, array dan cara pemrograman java, serta yang berkaitan dengan silsilah keluarga.

a) Analisis

Identifikasi dari kegiatan pengumpulan data dapat terlihat bahwa susunan keluarga / silsilah keluarga jika di gambarkan akan terlihat menyerupai struktur data dalam bentuk pohon / tree.

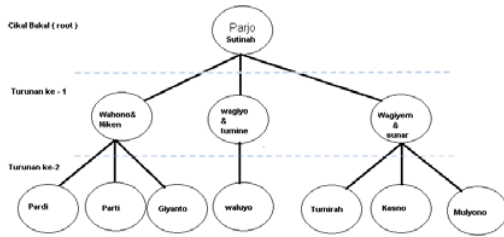
Didalam struktur data jenis struktur data berbentuk tree akan disimpan dalam simpul-simpul . Selain itu menurut penulis struktur data juga bisa disimpan dalam jenis tipe data larik, seperti array. Hal ini yang akan dikaji apakah nanti array dapat digunakan untuk menyimpan data model tree ini.

b) Perancangan

Proses perancangan dilakukan dengan membuat model dari array yang akan digunakan untuk menyimpan data silsilah keluarga. berikut merupakan gambaran dari desain rancangan struktur array.

Berdasarkan urutan silsilah keluarga jawa tersebut, maka pada makalah ini peneliti membuat

gambar 5 berupa struktur sisilah keluarga sebagai berikut :



Gambar 5. Tree stuktur Silsilah keluarga

Berdasarkan gambar diatas membentuk tree dengan tingkat turunan dari cikal bakal turunan sampai keturunan 2, maka silsilah keluarga yang tersajikan dari gambar 1, dari root sampai turunan ke 2, atau dari orang tua paling atas kebawah sampai "putu". Maka setiap node akan mempunyai identitas. Identitas tiap node tersebut dibuat dalam tabel 1.

Tabel 1. Isi node

Node	Induk	Turunan ke	sex	Pasangan	sek	Anak-ke
Parjo	-	0	L	Sutinah	P	0
Wahono	Parjo	1	L	Niken	P	1
Wagiyo	Parjo	1	L	tuminem	P	1
Wagiyem	Parjo	1	P	Sunar	L	1
Pardi	Wahono	2	L	-	-	1
Parti	Wahono	2	P	-	-	2
Giyanto	Wahono	2	L	-	-	3
Waluyo	Wagiyo	2	L	-	-	1
Tumirah	Wagiyem	2	P	-	-	1
Kasno	Wagiyem	2	L	-	-	2
Mulyono	Wagiyem	2	L	-	-	3

Tabel 1 menjelaskan hubungan antar node yang ada pada gambar 1. Pada tabel 1 baris wahono, menikah dengan Niken, anak ke 1, keturunan dari parjo. Pada baris Parti, jenis kelamin P (Perempuan), merupakan anak ke 2, turunan dari Node Wahono.

Struktur array diatas tersusun pada baris, dan kolom. pada tabel 2.

Tabel 2. Struktur kolom dari array

0	1	2	3	4	5	6
Node (anak)	parent	level	Sex parent	mantu	Sex mantu	Anak

Penjelasan :

Pada tabel 2 isi tiap kolom adalah kolom ke 0 menunjukan node (anak), sedangkan kolom ke 1 diisi parent / orang tua dari anak / node. kolom ke 2 menunjukan turunan ke berapa dari node. kolom

ke 3 menunjukan sex parent, kolom ke 4 node mantu / pasangan dari node ke 0. kolom ke 5 sex dari pasangan, kolom ke 6 merupakan kolom yang menunjukan akan ke berapa.

c) Pembuatan

Pada tahap setelah melakukan kegiatan pengumpulan data, peneliti membuat gambaran silsilah keluarga dari beberapa keluarga. Sebagai gambaran berikut contoh gambaran sebuah silsilah keluarga.

Berdasarkan gambar 5 tree diatas maka struktur data penyimpanan struktur menggunakan Array [3][4] peneliti buat dalam bentuk berikut :

```
static String[][] keluarga = {
    {"PARJO", " ", "0", "L", "Sutinah", "P", "0"},
    {"WAHONO", "PARJO", "1", "L", "Niken", "P", "1"},
    {"WAGIYO", "PARJO", "1", "L", "Tuminem", "P", "2"},
    {"WAGIYEM", "PARJO", "1", "L", "Sutinah", "P", "3"},
    {"PARDI", "WAHONO", "2", "L", "Sutinah", "P", "1"},
    {"PARTI", "WAHONO", "2", "L", "Sutinah", "P", "2"},
    {"GIYANTO", "WAHONO", "2", "P", "Sutinah", "P", "3"},
    {"WALUYO", "WAGIYO", "2", "P", "Sutinah", "P", "3"},
    {"TUMIRAH", "WAGIYEM", "2", "P", "Sunar", "L", "1"},
    {"KASNO", "WAGIYEM", "2", "L", "Sunar", "L", "2"},
    {"MULYONO", "WAGIYEM", "2", "L", "Sunar", "P", "3"}
};
```

Gambar 6. pelacakan anak

d) Implemetasi

Merupakan proses pembuatan algoritma mencari kakek dan nenek, bapak dan ibu, pak lek dan bu lek, pak dhe dan bu dhe, serta anak.

e) Pengujian

Tahap Pengujian dilakukan dengan mencoba desain penyimpanan silsilah keluarga dengan menggunakan array. Proses pengujian dilakukan pada fungsi pencarian anak, kakek /nenek, ayah/ ibu, pak lek / bu lek, pak dhe / bu dhe, dan anak.

4. IMPLEMENTASI

Implementasi ini dilakukan untuk membuat fungsi / logika pencarian silsilah kekeluargaan di masyarakat Jawa Tengah[5][6].

a Anak

alur logika proses pencari anak dari suatu node, misal mencari anak "wagiyo" , gambar 6 berikut :



Gambar 7. pelacakan anak

Proses pencarian anak (wagiyo) dilakukan dengan tahapan:

- a. Cari posisi c (orang dicari anaknya)

```
static String[][] keluarga = {
    {"PARJO", " ", "0", "L", "Sutinah", "P", "0"},
    {"WAHONO", "PARJO", "1", "L", "Niken", "P", "1"},
    {"WAGIYO", "PARJO", "1", "L", "Tuminem", "P", "2"},
    {"WAGIYEM", "PARJO", "1", "P", "Sunar", "L", "3"},
};
```

Gambar 8. pelacakan anak

- b. cari pada baris array yang kolom 1, isinya sama dengan c. Cetak kolom ke 0.

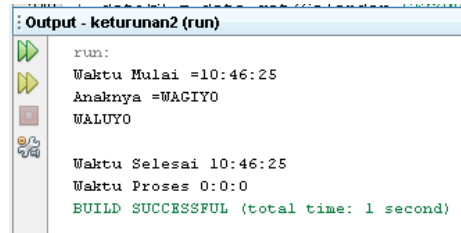
```
{ "WAGIYEM", "PARJO", "1", "P", "Sunar", "L", "3"},
{"PARDI", "WAHONO", "2", "L", "Sutinah", "P", "1"},
{"PARTI", "WAHONO", "2", "L", "Sutinah", "P", "2"},
{"GIYANTO", "WAHONO", "2", "P", "Sutinah", "P", "3"},
{"WALUYO", "WAGIYO", "2", "P", "Sutinah", "P", "3"},
{"TUNIRAH", "WAGIYEM", "2", "P", " ", " ", "1"},
```

Gambar 9. pelacakan anak

alur logika tersebut dibuat prosedur dengan bahasa java :

```
//FUNGSI
public static void anak(String c) {
    int i,j;
    System.out.println("Anaknya =" + c);
    for(i=0;i<keluarga.length;i++)
    {
        if (c.compareTo(keluarga[i][1]) == 0) {
            // cek data orang tua laki /perempuan
            System.out.println(keluarga[i][0]);
        } //if
    } //i
}
```

Luaran dari fungsi anak, tersebut adalah :



Gambar 10. pelacakan anak

b Orang Ayah

proses pencarian ayah suatu node ("wagiyem"), dilakukan dengan memasukkan

1. Cari node dicari pada baris array pada kolom 0,
2. baris yang ditemukan cek kolom 3 jika isi "L", maka cetak kolom 1, jika kolom 3 berisi "P" maka cetak kolom 4.

```
static String[][] keluarga = {
    {"PARJO", " ", "0", "L", "Sutinah", "P", "0"},
    {"WAHONO", "PARJO", "1", "L", "Niken", "P", "1"},
    {"WAGIYO", "PARJO", "1", "L", "Tuminem", "P", "2"},
    {"WAGIYEM", "PARJO", "1", "L", "Sunar", "L", "3"},
    {"PARDI", "WAHONO", "2", "L", "Sutinah", "P", "1"},
};
```

Gambar 11. pelacakan anak

Alur logika diatas kemudian penulis buat fungsi sebagai berikut :

```
public static void ayah(String c) {
    int i,j,k;
    String sementara,ayahnya = null;
    System.out.print("Ayah Nya "+ c +" =");
    // cari baris
    for(i=0;i<keluarga.length;i++)
    {
        if (c.compareTo(keluarga[i][0]) == 0) {
            sementara=keluarga[i][1];
            // cek data baris calon ortu
            for(k=0;k<keluarga.length;k++)
            {
                if (sementara.compareTo(keluarga[k][0]) == 0) {
                    if(keluarga[k][3].compareTo("P")==0){
                        ayahnya= keluarga[k][4];
                    }else{
                        ayahnya= keluarga[k][0];
                    }
                    System.out.println(ayahnya);
                } //k
            } // if compare
        } // cari ayah
    }
}
```

hasil luaran fungsi ayah pencari ayah "wagiyem", sebagai berikut :

```

Output - keturunan2 (run)
run:
Waktu Mulai =11:1:23
Ayah Nya WAGIYEM =PARJO
Waktu Selesai 11:1:23
Waktu Proses 0:0:0
BUILD SUCCESSFUL (total time: 0 seconds)
    
```

Gambar 12. pelacakan anak

```

Output - keturunan2 (run)
run:
Waktu Mulai =11:24:31
Ibu Nya WAGIYEM =Sutinah
Waktu Selesai 11:24:31
Waktu Proses 0:0:0
BUILD SUCCESSFUL (total time: 1 second)
    
```

Gambar 14. pelacakan anak

c Ibu

proses pencarian ibu suatu node ("wagiyem"), dilakukan dengan memasukkan

1. Cari node dicari pada baris array pada kolom 0,
2. baris yang ditemukan cek kolom 3 jika isi "P", maka cetak kolom 1, jika kolom 3 berisi "P" maka cetak kolom 4.

Alur proses tersebut digambarkan pada gambar berikut :

```

static String[][] keluarga = {
    {"PARJO","", "0","L","Sutinah","P","0"},
    {"WAHONO","PARJO","1","L","Niken","P","1"},
    {"WAGIYO","PARJO","1","L","Tumilem","P","2"},
    {"WAGIYEM","PARJO","1","L","Sutinah","P","3"},
    {"PARTI","WAHONO","2","L","Sutinah","P","2"},
    {"GIYANTO","WAHONO","2","P","Sutinah","P","3"},
    {"WALUYO","WAGIYO","2","P","Sutinah","P","3"},
    {"TUMIRAH","WAGIYEM","2","P","","P","1"},
    {"RASNO","WAGIYEM","2","L","","P","2"},
    {"HUSONO","WAGIYEM","2","L","Suni","P","3"}
}
    
```

Gambar 13. pelacakan anak

Bentuk code dari fungsi pencarian ibu sebagai berikut :

```

// fungsi ayah / bapak
public static void ibu(String c) {
    int i,j,k;
    String sementara,Ibunya = null;
    System.out.print("Ibu Nya "+ c +" =");
    // cari baris
    for(i=0;i<keluarga.length;i++)
    {
        if (c.compareTo(keluarga[i][0]) == 0) {
            sementara=keluarga[i][1];
            // cek data baris calon ortu
            for (k=0;k<keluarga.length;k++)
            {
                if (sementara.compareTo(keluarga[k][0]) == 0) {
                    if (keluarga[k][3].compareTo("P")==0) {
                        Ibunya= keluarga[k][0];
                    }else{
                        Ibunya= keluarga[k][4];
                    }
                    System.out.println(Ibunya);
                }
            }
        }
    }
}
    
```

output pencarian ibu wagiyem , luaranya sebagai berikut :

d Kakek / Nenek

Algoritma proses pencarian kakek dan nenek dilakukan dengan tahapan sebagai berikut:

1. cari baris dari node yang akan dicari kakek dan neneknya,
2. baca kolom 1(kolom parentnya)
3. cari baris node parent
4. pada bari node parent, cek kolom 3 jika berisi "L" , maka kakek kolom 1, nenek kolom 4.

proses pembuktian algoritma pencarian kekek dan nenek penulis buat dengan list menggunakan bahasa java sebagai berikut:

```

static String[][] keluarga = {
    {"PARJO","", "0","L","Sutinah","P","0"},
    {"WAHONO","PARJO","1","L","Niken","P","1"},
    {"WAGIYO","PARJO","1","L","Tumilem","P","2"},
    {"WAGIYEM","PARJO","1","L","Sutinah","P","3"},
    {"PARTI","WAHONO","2","L","Sutinah","P","2"},
    {"GIYANTO","WAHONO","2","P","Sutinah","P","3"},
    {"WALUYO","WAGIYO","2","P","Sutinah","P","3"},
    {"TUMIRAH","WAGIYEM","2","P","","P","1"},
    {"RASNO","WAGIYEM","2","L","","P","2"},
    {"HUSONO","WAGIYEM","2","L","Suni","P","3"}
}
    
```

Gambar 15. pelacakan anak

Gambar 17. pelacakan anak

```
public static void mbahaya(String c) {
    int i,j,k,urutan;
    int baris_ortu;
    String saya,ortu,sementara = null;
    // cari ortu
    System.out.println("Kekek& Nenek "+ c + " ==");
    // cari baris ortu
    for(i=0;i<keluarga.length;i++)
    {
        if (c.compareTo(keluarga[i][0]) == 0)
        {
            ortu=keluarga[i][1];
            for(j=0;j<keluarga.length;j++)
            {
                if (ortu.compareTo(keluarga[j][0]) == 0)
                {
                    sementara=keluarga[j][1];
                    // cari data simbah
                    for(k=0;k<keluarga.length;k++)
                    {
                        if (sementara.compareTo(keluarga[k][0]) == 0)
                        {
                            System.out.println(" "+keluarga[k][0] );
                            System.out.println(" "+keluarga[k][4] );
                        }
                    }
                }
            }
        }
    }
}
}
```

```
public static void pakbulek(String c) {
    // cari level ayah
    int i,j,k,urutan;
    int baris_ortu;
    String sementara,ayahnya, ortu = null;
    String anakde,level=null;
    System.out.println("Pak Lek & Bu Lek "+ c + " ==");
    // cari baris ortu
    for(i=0;i<keluarga.length;i++)
    {
        if (c.compareTo(keluarga[i][0]) == 0) {
            ortu=keluarga[i][1];
            for(j=0;j<keluarga.length;j++)
            {
                if (ortu.compareTo(keluarga[j][0])!=0){
                    level=keluarga[j][2];
                    //System.out.println("level "+keluarga[j][0]);
                    urutan=Integer.parseInt(keluarga[j][6]);
                    ortu=keluarga[j][0];
                    for(k=0;k<keluarga.length;k++)
                    {
                        if (level.compareTo(keluarga[k][2]) == 0)
                        {
                            //if (Integer.parseInt(keluarga[k][6])< urutan)
                            // && keluarga[k][2]=="L" )
                            if (Integer.parseInt(keluarga[k][6])< urutan)
                            {
                                System.out.println("Pak lek = "+keluarga[k][0]+", Buk lek "+keluarga[k][4]);
                            }
                        }
                    }
                }
            }
        }
    }
}
}
```

Hasil dari logika diatas terlihat pada luaran berikut :

```
Output - keturunan2 (run)
run:
Waktu Mulai =11:28:35
Kekek& Nenek KASNO =
PARJO
Sutinah

Waktu Selesai 11:28:35
Waktu Proses 0:0:0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 16. pelacakan anak

Berikut hasil luaran pencarian paman dan bulekasno :

```
Output - keturunan2 (run)
run:
Waktu Mulai =11:41:0
Pak Lek & Bu Lek KASNO =
Pak lek = WAHONO, Buk lek Niken
Pak lek = WAGIYO, Buk lek Tuminem

Waktu Selesai 11:41:0
Waktu Proses 0:0:0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 18. pelacakan anak

e Paman

Proses pencarian paman atau dalam bahasa jawa disebut “paklek”, serta “bulek”. Alur logika pencarian paklek dari node “Kasno” diatas digambarkan pada alur berikut :

1. Cari baris node Kasno, baca ayah kasno
2. Cari node ayah Kasno, baca kolom 6, baca nilainya(3),
3. Cari baris yang kolom 1 berisi sama dengan nama ayah kasno, serta kolom ke 6 dibawah 3.

berikut gambaran alur diatas :

```
static String[][] keluarga = (
    {"PARJO", "1", "0", "L", "Sutinah", "P", "0"},
    {"WAHONO", "PARJO", "1", "L", "Niken", "P", "1"},
    {"WAGIYO", "PARJO", "1", "L", "Tuminem", "P", "2"},
    {"WAGIYEH", "PARJO", "1", "L", "Sutinah", "P", "3"},
    {"PANEL", "WAHONO", "2", "L", "Sutinah", "P", "1"},
    {"PARTI", "WAHONO", "2", "L", "Sutinah", "P", "2"},
    {"GIYANTO", "WAHONO", "2", "p", "Sutinah", "P", "3"},
    {"WALUYO", "WAGIYO", "2", "p", "Sutinah", "P", "3"},
    {"TUMIRAH", "WAGIYEH", "2", "p", "Sutinah", "P", "1"},
    {"KASNO", "WAGIYEH", "2", "L", "Sutinah", "P", "2"},
    {"WAGIYEH", "WAGIYEH", "2", "L", "Sutinah", "P", "3"}
);
```

f Pakdhe / Bu dhe

Algoritma proses pencarian pak dhe dan budhe dilakukan dengan tahapan sebagai berikut:

1. cari baris dari node yang akan dicari pak dhe dan budhenya,
2. baca kolom 1(kolom parentnya)
3. cari baris node parent
4. baca kolom 1sebagai ,node parent atasnya.sebut M,, dan kolom 6 (anak ke) sebut N
5. baca baris yang kolom ke 1 berisi M dan kolom ke 6 isinya diatas N
6. pada bari node parent, cek kolom 3 jika berisi “L” , maka dia pak dhe kolom 1, budhenya kolom 4.

2. Penyimpanan data menggunakan array , maka proses pelacakan kekerabatan dilakukan secara sequential melalui baris 0.

DAFTAR PUSTAKA

1.Penerapan Algoritma Breadth First Search dan Depth First Search pada Game Angka Ari Muhardono1)* , 1)Universitas Pekalongan, Indonesia 1)arimuhardono.unikal@ac.id, Jurnal Minfo Polgan Volume 12, Nomor1, Maret 2023, e-ISSN : 2797-3298,hal 171-182

2. [HYPERLINK "" \l "IGe20" BO], I Gede Wira Kusuma Jaya, Ida Bagus Nyoman Wijana Manuaba, Kadek Ryan Wijaya,I Putu Surya Pratama Wardhana,I Made Andika Saputra ,I Gede Aris Gunadi,Jurnal Ilmu Komputer Indonesia(JIK) Vol : 5, No. 2, November 2020 ISSN (Print): 2615-2703, ISSN (Online): 2615-2711

3.IMPLEMENTASI STRUKTUR DATA TREE PADA GAME SNAKE DENGAN C Dena Djayusman,Ida Suhartini,Fadhlan Naufal Gunawan,Naufal Taufik Alhakim, Thalissa Gunawan A,Politeknik Negeri Bandung, JURNAL DIGIT Vol. 11, No.1 Mei 2021, pp.20~27, ISSN : 2088-589X

4 Algoritma dan Struktur data, Afrizal Zein, Emi Sita Eriana, umpampres, ISBN: 978-623-6352-65-6, 14 Januari 2022

5.Mukaddimah: Jurnal Studi Islam Kopertais Wilayah III D.I Yogyakarta , Vol. 8, No. 1, Januari-Juni 2023 – ISSN 2338-6924 (online) – ISSN 2579-4957 (cetak) . Judul Hikmah Pendidikan Islam Pada Silsilah Keluarga Upin Dan Ipin, penulis Robingun Suyud El Syam (1), Sofan Rizqi (2)Universitas Sains Al-Qur'an Wonosobo (1), (2)

6.Fitri Latifah, PENERAPAN ALGORITHMA POHON UNTUK OPERASI PENGOLAHAN DAN PENYIMPANAN DATA DALAM TEKNIK PEMROGRAMAN (kajian algoritma pohon pada teknik pemrograman)