Aplikasi Pohon untuk Menentukan Langkah Pemain dalam Permainan Menara Hanoi

Mohammad Sheva Almeyda Sofjan 13519018
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519018@std.stei.itb.ac.id

***Abstrak*— Menara Hanoi (*Tower of Hanoi*) merupakan puzzle atau permainan teka-teki yang terdiri atas tiga menara dan beberapa piringan berbeda ukuran yang ditengahnya diberi lubang yang minimum seukuran luas penampang menara sehingga cakram dapat ditumpuk pada menara dan bertujuan untuk memindahkan seluruh cakram pada menara awal ke menara akhir, dengan aturan urutan cakram yang berukuran paling besar berada di paling bawah tumpukan dan cakram paling kecil berada di paling atas. Pada makalah ini akan dibahas aplikasi pohon dalam membantu menentukan langkah pemain dalam menyelesaikan permainan Menara Hanoi dengan memanfaatkan solusi iteratif dari permainan itu sendiri.**
***Kata kunci*—Hanoi, Menara, Permainan, Pohon**

I. PENDAHULUAN

Menara Hanoi (*Tower of Hanoi*), juga disebut *Tower of Brahma* atau *Lucas' Tower* merupakan puzzle atau permainan teka-teki yang diciptakan oleh matematikawan Prancis bernama François Édouard Anatole Lucas pada tahun 1883. Permainan ini (biasanya) terdiri atas 3 (tiga) menara/*tower* dan beberapa cakram atau piringan berbeda ukuran yang ditengahnya diberi lubang yang minimum seukuran luas penampang menara sehingga cakram dapat ditumpuk pada menara.

Tujuan dari permainan ini adalah untuk memindahkan seluruh cakram pada menara awal ke menara akhir, dengan aturan urutan cakram yang berukuran paling besar berada di paling bawah tumpukan dan cakram paling kecil berada di paling atas. Kondisi awal tumpukan pada menara awal biasanya sudah sesuai dengan kondisi tumpukan seharusnya pada akhir permainan (terurut bawah-besar hingga atas-kecil).

Gambar 1. Permainan Menara Hanoi
Diambil dari
https://commons.wikimedia.org/wiki/File:Tower_of_Hanoi.jpeg
pada 6 Desember 2020

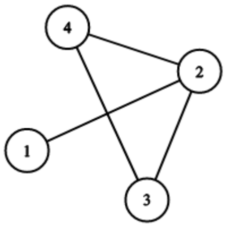
Nama "Hanoi" dari Menara Hanoi berasal dari sebuah legenda yang menyatakan bahwa dahulu, (yang dipercayai berada) di kota Hanoi, Vietnam, terdapat tiga buah tiang tegak (*tower)* berukuran ± 5 meter dan 64 buah cakram berbeda ukuran yang tengahnya diberi lubang. Awalnya cakram tersebut tersusun sedemikian rupa sehingga cakram yang di bawah memiliki ukuran yang lebih besar daripada cakram di atasnya. Pendeta Brahma, mengikuti arahan dari ramalan, telah mencoba menggerakkan cakram-cakram tersebut sedemikian rupa sehingga tumpukan cakram pada menara pertama kosong dan pada menara ketiga penuh dengan tumpukan cakram sesuai konfigurasi awal. Konon, apabila langkah terakhir dari pemindahan cakram tersebut telah usai, dunia akan berakhir. [1]

Penggunaan pohon sebagai alat bantu penyelesaian permainan Menara Hanoi ini dititikberatkan pada pohon keputusan. Tujuan dari penggunaan pohon dalam menyelesaikan permasalahan dalam permainan ini adalah untuk membantu menyajikan visualisasi pemilihan keputusan sehingga dapat membantu pemain dalam memahami langkah - langkah yang diperlukan untuk menyelesaikan permainan secara optimal.

II. LANDASAN TEORI

A. Definisi Graf

Graf didefinisikan sebagai himpunan dari pasangan *(V,E)* dengan V (*vertex)* melambangkan himpunan tidak kosong dari simpul-simpul $\{v_1, v_2, \ldots, v_n\}$ dan E (*edge)* melambangkan himpunan sisi yang menghubungkan sepasang simpul $\{e_1, e_2, \ldots, e_n\}$. Sebagai contoh, terdapat graf G = *(V,E)* yang memiliki $V = \{1,2,3,4\}$ dan E = {(1,2),(2,3),(3,4),(2,4)} dapat direpresentasikan sebagai berikut.



Gambar 2. Representasi Graf G

B. Jenis Graf

Graf dapat digolongkan berdasarkan ada tidaknya gelang dan ada tidaknya arah pada graf.

Berdasarkan ada tidaknya gelang/*loop* (sisi yang menghubungkan dua simpul yang sama) atau sisi ganda pada graf, graf digolongkan menjadi 2 (dua) jenis yaitu :

1. Graf sederhana (*simple graph)*
   Graf sederhana adalah suatu graf yang tidak mengandung gelang maupun sisi ganda.
2. Graf tak sederhana (*unsimple graph)*
   Graf tak sederhana adalah suatu graf yang mengandung sisi ganda atau gelang. Graf tak sederhana dapat dikelompokkan lagi menjadi graf ganda (*multi-graph)* dan graf semu (*pseudo-graph)*. Sesuai dengan namanya, graf ganda merupakan graf yang mengandung sisi ganda, sedangkan graf semu merupakan graf yang mengandung sisi gelang.

Sedangkan berdasarkan ada tidaknya arah pada sisi graf, graf digolongkan menjadi 2 (dua) jenis yaitu :

1. Graf berarah (*directed graph / digraph)*
   Graf berarah merupakan suatu graf yang setiap sisinya diberikan orientasi arah.
2. Graf tak berarah (*undirected graph)*
   Graf tak berarah merupakan suatu graf yang sisinya tidak mempunyai orientasi arah.

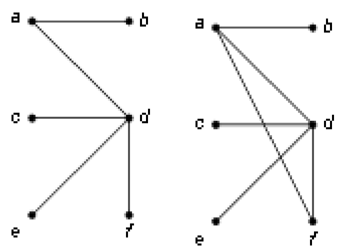
C. Terminologi Graf

Misal terdapat graf G = *(V,E)* yang memiliki *V* = {1,2,3,4,5} dan E = {(1,2),(2,3),(3,4),(2,4)}, dapat didefinisikan beberapa terminologi pada graf, sebagai berikut.

1. Ketetanggaan (*adjacent)*
   Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung (terdapat sisi yang menghubungkan kedua simpul tersebut). Pada graf G yang merupakan graf tak berarah, dapat dikatakan bahwa simpul 1 bertetanggaan dengan simpul 2 , simpul 2 bertetanggaan dengan simpul 1,3, 4, dan seterusnya.
2. Bersisian (*incidency)*
   Untuk sembarang sisi e = *(u,v)* dapat dikatakan bahwa e bersisian dengan simpul *u* atau e bersisian dengan simpul *v.* Pada graf G dapat dikatakan bahwa sisi (1,2) bersisian dengan simpul 1 dan 2.
3. Derajat (*degree)*
   Derajat merupakan suatu besaran pada simpul suatu graf yang menyatakan banyaknya sisi yang bersisian dengan simpul tersebut. Derajat suatu graf dinyatakan dengan notasi *d(v)*. Pada graf G, $d(1) = 1$, $d(2) = 3$, $d(3) = 2$, $d(4) = 2$, dan $d(5) = 0$.
4. Lintasan (*path)*
   Lintasan denhan panjang *n* dari simpul awal *vo* ke simpul tujuan *vn* pada suatu graf merupakan barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk v_0, e_1, v_1, v_2, ……, v_{n-1}, e_n, v_n sedemikian sehingga $e_1 = \{v_0, v_1\}$, $e_2 = \{v_1, v_2\}$,…, $e_n = \{v_{n-1}, v_n\}$ adalah sisi-sisi dari graf tersebut. Pada graf G, lintasan 2,3,4 merupakan lintasan dengan sisi (2,3) dan (3,4). Panjang lintasan merupakan jumlah sisi pada suatu lintasan. Lintasan 2,3,4 pada graf G memiliki panjang 2.
5. Siklus / Sirkuit (*cycle / circuit)*
   Siklus / sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama. Pada graf G, lintasan 2,3,4,2 merupakan sebuah sirkuit karena berawal pada simpul 2 dan berakhir pada simpul 2 juga. Sirkuit 2,3,4,2 pada graf G memiliki panjang 3.
6. Keterhubungan (*connected)*
   Dua buah simpul *u* dan *v* dikatakan terhubung apabila terdapat lintasan dari *u* ke *v*. Suatu graf tak berarah dikatakan graf terhubung jika untuk setiap pasang simpul v_i dan v_j pada himpunan *V* terdapat lintasan dari v_i ke v_j. Graf G bukan merupakan graf terhubung.
7. Upagraf (*subgraph)*
   Misalkan terdapat $G_1 = (V_1,E_1)$, G_1 merupakan upagraf dari G apabila $V_1 \subseteq V$ dan $E_1 \subseteq E$.
8. Graf Berbobot (*weighted graph)*
   Graf berbobot adalah suatu graf yang setiap sisinya diberi sebuah harga (bobot).

D. Definisi Pohon

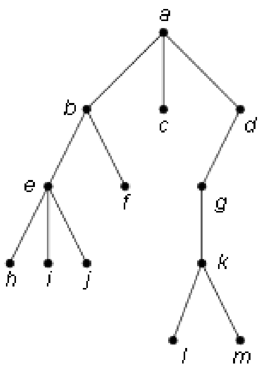
Pohon merupakan graf tak berarah terhubung yang tak mengandung sirkuit. Graf G = *(V,E)* dikatakan sebagai pohon apabila merupakan graf sederhana dan tidak mengandung sirkuit.



Gambar 3. Sebuah pohon (kiri) dan bukan pohon (kanan) Diambil dari http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf pada 9 Desember 2020

E. Definisi Pohon Berakar

Pohon berakar adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah menjauhi akar sehingga menjadi graf berarah. Dalam penggambarannya, pohon berakar tidak perlu diberikan penanda arah pada sisinya karena setiap sisinya sudah memiliki arah yang tetap (menjauhi akar).



Gambar 3. Sebuah pohon berakar, diambil dari http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf pada 9 Desember 2020

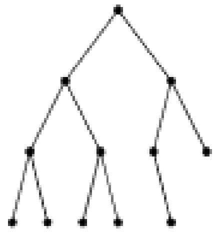
F. Terminologi Pohon Berakar

Misal terdapat pohon berakar seperti pada gambar 3. Dapat didefinisikan beberapa terminologi pada pohon berakar sebagai berikut.

1. Anak (*child* atau *children)* dan orangtua (*parent)*
   Pada pohon di gambar 3, *b, c, dan d* adalah anak-anak simpul *a*, dan *a* adalah orangtua dari anak-anak itu.
2. Lintasan (*path)*
   Pada pohon di gambar 3, lintasan dari simpul *a* ke *h* adalah *a,b,e,h* dan lintasan tersebut memiliki panjang 3.
3. Saudara Kandung (*sibling)*
   Pada pohon di gambar 3, simpul *f* merupakan saudara kandung *e,* namun *g* bukan saudara kandung *e,* karena memiliki orangtua yang berbeda.
4. Upapohon (*subtree)*
   Pohon berakar yang memiliki akar yang merupakan bagian dari pohon lainnya merupakan upapohon dari pohon lain tersebut. Pada gambar 3, pohon berakar dengan akar simpul b dengan himpunan simpul *V = (b, e, f, h, i, j)* merupakan upapohon dari pohon pada gambar 3.
5. Derajat (*degree)*
   Derajat sebuah simpul adalah jumlah upapohon atau jumlah anak dari simpul tersebut.Sebagai contoh, pada gambar 3 derajat dari *a* adalah 3 dan derajat dari *c* adalah 0.
6. Daun (*leaf)*
   Simpul yang tidak memiliki anak atau berderajat nol disebut dengan daun. Pada gambar 3, simpul *h, i , j, f, c, l, m* merupakan daun.
7. Simpul Dalam (*internal node)*
   Simpul yang memiliki anak disebut simpul dalam. Pada gambar 3, simpul *b,d,e,g,k* disebut simpul dalam.
8. Aras / Tingkat (*level)*
   Aras sebuah simpul pada pohon berakar adalah jarak dari akar suatu pohon ke simpul tersebut. Pada gambar 3, aras dari simpul *a* adalah 0.
9. Tinggi (*height)* atau Kedalaman (*depth)*
   Tinggi atau kedalaman suatu pohon berakar merupakan aras maksimum dari pohon tersebut. Pada gambar 3, tinggi dari pohon adalah 4, karena aras maksimum dari pohon tersebut (aras simpul *l* dan *m)* adalah 4.

G. Pohon Biner

Pohon biner merupakan pohon berakar yang setiap simpul cabangnya mempunyai paling banyak 2 buah anak. Anak dari suatu simpul dari pohon biner dibedakan menjadi anak kiri (*left child)* dan anak kana n (*right child)*. [2]

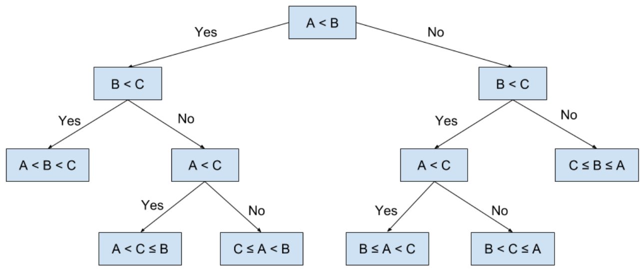


Gambar 4. Sebuah pohon biner, diambil dari http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag2.pdf pada 9 Desember 2020

H. Pohon Keputusan

Pohon keputusan (*decision tree)* merupakan penerapan dari pohon berakar yang digunakan sebagai model dari alur eksekusi langkah dari suatu masalah berdasarkan kemungkinan kondisi yang terjadi pada proses penyelesaian masalah tersebut.

Pada pohon keputusan, simpul-simpul pada pohon diisi dengan hal yang diputuskan, sisi-sisi berisi kemungkinan kondisi dari pilihan yang dipilih pada setiap simpulnya, dan daun-daun berisi solusi dari permasalahan pada akar.



Gambar 5. Sebuah pohon keputusan untuk menentukan urutan 3 bilangan , diambil dari https://elf11.github.io/images/decisionTree.png pada 9 Desember 2020

H. Rekursi

Rekursi merupakan suatu proses mendefinisikan objek dalam terminologi dirinya sendiri. Suatu objek dikatakan rekursif bila ia didefinisikan dalam terminologi dirinya sendiri.

Suatu fungsi rekursif didefinisikan oleh dua bagian, diantaranya adalah basis dan rekurens. Basis merupakan bagian yang berisi nilai fungsi yang sudah terdefinisi secara eksplisit, serta menghentikan rekursif untuk memberikan nilai yang terdefinisi pada suatu fungsi rekursif. Sedangkan rekurens merupakan bagian yang mendefinisikan fungsi dalam terminologi dirinya sendiri, berisi kaidah untuk menemukan nilai fungsi pada suatu input dari nilai-nilai lainnya pada input yang lebih kecil. *[2]*

Fungsi *F* berikut merupakan salah satu contoh dari sekian banyak fungsi rekursif.

$$F(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F(n-1) + F(n-2) & n \geq 2 \end{cases}$$

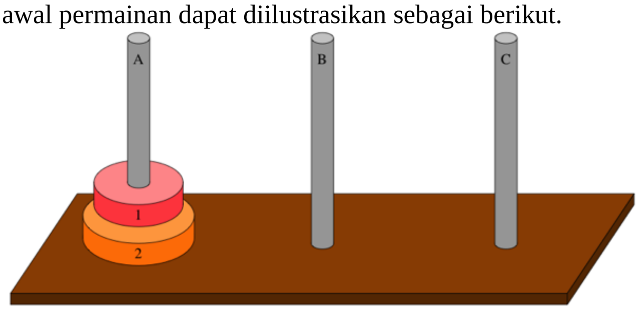
Gambar 6. Sebuah fungsi rekursif (Fibonacci), diambil dari https://miro.medium.com/max/2640/1*y9RU2g7i4tnrHO3VYPtqAQ.png pada *10 Desember 2020*

III. PEMBAHASAN

A. Aturan dan Tujuan Permainan

Dalam permainan Menara Hanoi yang dibahas pada makalah ini, akan dispesifikasikan banyaknya menara yang dipakai adalah yang umum dipakai, yaitu 3 buah menara. Setiap menara diberi indeks A, B, dan C.

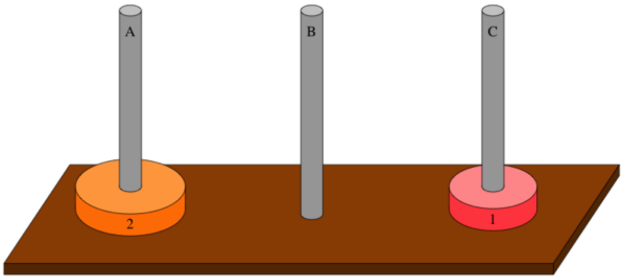
Menara awal di mana tumpukan piringan sudah tersusun disebut menara A. Menara B merupakan menara tujuan akhir permainan, di mana pemain harus memindahkan seluruh piringan pada menara A sesuai urutan yang valid (mulai dari piringan terbesar, semakin ke atas semakin kecil). Sedangkan menara C merupakan menara bantu (menara sementara) untuk membantu memindahkan piringan dari A ke B. Konfigurasi awal permainan dapat diilustrasikan sebagai berikut.



Gambar 7. Kondisi awal Menara Hanoi dengan 2 piringan, diambil dari https://www.khanacademy.org/computing/computer-science/algorithms/towers-of-hanoi/a/towers-of-hanoi pada 9 Desember 2020

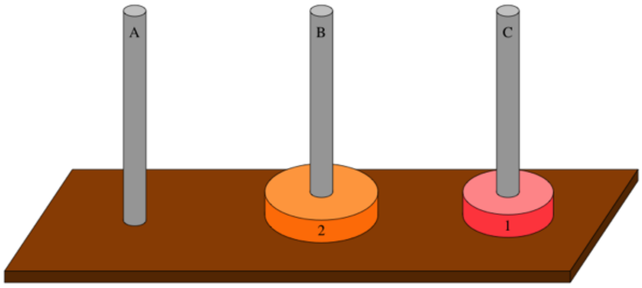
Untuk memindahkan tumpukan piringan pada suatu menara, pemain hanya boleh memindahkan piringan yang lebih atas terlebih dahulu pada suatu menara. Disini, piringan diurutkan sesuai dengan ukuran piringan (semakin besar indeks piringan, semakin besar piringan tersebut). Pemindahan piringan dari menara A(awal) ke B(tujuan) untuk 2 piringan dapat disimulasikan sebagai berikut.

- Langkah Pertama
  Untuk langkah pertama, piringan (1) dapat dipindahkan ke menara B secara langsung maupun C. Namun jika dipindahkan langsung ke B, maka piringan (2) harus ditempatkan di C terlebih dahulu, dan piringan (1) harus dipindahkan ke A lalu piringan (2) ke B lalu piringan (1) ke B. Pada kasus ini memakan total 5 langkah. Mungkin ada langkah yang lebih efektif. Akan dicoba untuk memindahkan piringan (1) ke menara C.



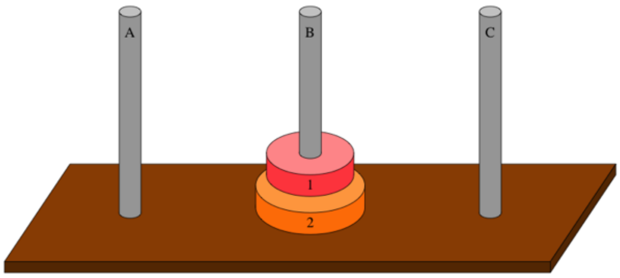
Gambar 8 . Langkah pertama, diambil dari https://www.khanacademy.org/computing/computer-science/algorithms/towers-of-hanoi/a/towers-of-hanoi pada 9 Desember 2020

- Langkah Kedua
  Pada langkah kedua, piringan (2) dapat dipindahkan ke B. Maka piringan (2) dipindahkan ke B.



Gambar 9. Langkah kedua, diambil dari https://www.khanacademy.org/computing/computer-science/algorithms/towers-of-hanoi/a/towers-of-hanoi pada 9 Desember 2020

- Langkah Ketiga
  Pada langkah ketiga, sudah jelas untuk mengakhiri permainan dan mencapai tujuan, piringan (1) perlu untuk dipindahkan ke B.



Gambar 10 . Langkah ketiga, diambil dari https://www.khanacademy.org/computing/computer-science/algorithms/towers-of-hanoi/a/towers-of-hanoi pada 9 Desember 2020

Dapat dilihat bahwa pada simulasi permainan di atas, terdapat 3 langkah untuk menyelesaikan permainan Menara Hanoi dengan 2 buah piringan. Untuk menyelesaikan permainan ini dengan *n* buah piringan, dapat digunakan metode rekursif dan metode iteratif, yang akan dibahas pada bagian-bagian berikutnya.

B. Metode Rekursif

Pada bagian ini akan dijelaskan langkah untuk menyelesaikan permainan Menara Hanoi menggunakan metode rekursif. Dari pemaparan langkah di simulasi permainan pada bagian *A* , untuk 2 buah piringan dapat diketahui alur penyelesaiannya adalah sebagai berikut.

1. Pindahkan 1 piringan dari A ke C
2. Pindahkan 1 piringan dari A ke B
3. Pindahkan 1 piringan dari C ke B

Dengan pola pikir yang sama, maka katakan, untuk 3 buah piringan dapat dilakukan langkah penyelesaian sebagai berikut.

1. Pindahkan piringan 1 dan 2 ke C , dengan melakukan langkah untuk dua piringan namun menara tujuan adalah menara C dan menara B adalah menara sementara.
2. Pindahkan piringan 3 ke menara B (menara tujuan)
3. Pindahkan 2 piringan sisa pada menara C ke menara B, dengan melakukan langkah untuk dua piringan dengan menara tujuan adalah menara B dan menara A adalah menara sementara (piringan 1 ke A lalu piringan 2 ke B lalu piringan 1 ke B).

Bila diperhatikan, langkah-langkah di atas dapat digeneralisasi menjadi prosedur rekursif untuk *n* buah piringan, sebagai berikut.

1. Pindahkan *n-1* piringan teratas dari A ke C
2. Pindahkan 1 piringan terbawah dari A ke B
3. Pindahkan *n-1* piringan pada C ke B
4. Lakukan hingga semua piringan berada pada B

Lebih jelasnya dalam bentuk *pseudocode* adalah sebagai berikut. Inisialisasi : (menara) awal = A, sementara = C, tujuan = B. [3]

```
MenaraHanoi(n,awal,sementara,tujuan)
     if(n = 1) then
       pindahkan satu piringan dari awal ke
       sementara
     else
       MenaraHanoi(n-1,awal,tujuan,sementara)
       pindahkan satu piringan dari awal ke tujuan
       MenaraHanoi(n-1,awal,tujuan,sementara)
```

C. Metode Iteratif

Pada bagian ini akan dijelaskan langkah untuk menyelesaikan permainan Menara Hanoi menggunakan metode iteratif. Dari paparan langkah-langkah pada metode rekursif, dapat diketahui total langkah yang diperlukan untuk menyelesaikan permainan dengan *n* piringan.

Asumsikan jumlah langkah yang diperlukan untuk menyelesaikan permainan dengan *n* piringan adalah H_n . Maka pada untuk menyelesaikan prosedur pertama (memindahkan *n-1* piringan teratas dari A ke C secara rekursif) terdapat H_{n-1} buah langkah, pada prosedur kedua (memindahkan 1 piringan terbawah dari A ke B) terdapat 1 buah langkah, dan pada prosedur ketiga (memindahkan *n-1* buah piringan dari C ke B) terdapat H_{n-1} buah langkah juga, maka total langkah yang diperlukan adalah

$$H_n = H_{n-1} + 1 + H_{n-1} = 2H_{n-1} + 1 \text{ ,dengan } H_1 = 1.$$

Persamaan di atas jika disederhanakan dapat menjadi sebagai berikut

$$\begin{aligned} H_n &= 2(2H_{n-2} + 1) + 1 = 2^2 H_{n-2} + 2 + 1 \\ &= 2^2(2H_{n-3} + 1) + 2 + 1 = 2^3 H_{n-3} + 2^2 + 2 + 1 \\ &= \ldots\ldots \\ &= 2^{n-1} H_1 + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1 \\ &= 2^n - 1 \end{aligned}$$

Jadi untuk menyelesaikan permainan Menara Hanoi dengan *n* buah piringan diperlukan $2^n - 1$ langkah.

Karena jumlah langkah untuk menyelesaikan permainan sudah diketahui, dapat dibuat solusi iteratif dalam menyelesaikan permainan.

Jika dilihat, untuk menyelesaikan permainan dengan hanya 1 piringan hanya diperlukan 1 langkah, yaitu memindahkan piringan tersebut dari menara A ke menara B.

Dari langkah penyelesaian untuk 2 piringan dan 3 piringan pada bagian *B*, diketahui juga bahwa untuk 2 piringan, langkah pertama yang diperlukan adalah memindahkan piringan 1 ke menara C (menara sementara), lalu memindahkan piringan 2 ke menara B, lalu memindahkan piringan 1 dari menara C ke menara B.

Untuk 3 piringan, langkah pertama yang diperlukan adalah memindahkan piringan 1 ke menara B, lalu memindahkan piringan 2 ke menara C, lalu memindahkan piringan 1 dari menara B ke menara C, lalu memindahkan piringan 3 ke menara B, lalu memindahkan piringan 1 dari menara C ke menara B, dan akhirnya memindahkan piringan 2 dari menara C ke menara B.

Dilihat bahwa untuk 1 piringan memiliki langkah awal yang sama dengan 3 piringan, dan 2 piringan memiliki langkah yang berbeda. Menurut metode rekursif, untuk 4 piringan maka langkah pertama yang dilakukan adalah memindahkan piringan pertama (piringan 1) ke menara C, sama seperti langkah awal untuk 2 piringan. Dapat disimpulkan bahwa untuk piringan berjumlah ganjil, langkah pertama yang dilakukan adalah memindahkan piringan pertama ke menara B, sedangkan untuk piringan berjumlah genap langkah pertama yang dilakukan adalah memindahkan piringan pertama ke menara C. Untuk *n* piringan, sebagai berikut.

Inisialisasi : total langkah = 0

- Untuk *n* ganjil
  1. Jika indeks piringan teratas pada menara B > indeks piringan teratas pada menara A atau menara B kosong, pindahkan dari A ke B.
     Jika indeks piringan teratas pada menara B < indeks piringan teratas pada menara A atau menara A kosong, pindahkan dari B ke A.
     Total langkah = total langkah + 1
  2. Jika indeks piringan teratas pada menara C > indeks piringan teratas pada menara A atau menara C kosong, pindahkan dari A ke C.
     Jika indeks piringan teratas pada menara C <

indeks piringan teratas pada menara A atau menara A kosong, pindahkan dari C ke A.
Total langkah = total langkah + 1

3. Jika indeks piringan teratas pada menara B > indeks piringan teratas pada menara C atau menara B kosong, pindahkan dari C ke B.
   Jika indeks piringan teratas pada menara B < indeks piringan teratas pada menara C atau menara C kosong, pindahkan dari B ke C.
   Total langkah = total langkah + 1
4. Ulangi langkah 1 – 3 hingga permainan selesai (total langkah = $2^n - 1$ langkah)

- Untuk *n* genap
  1. Jika indeks piringan teratas pada menara C > indeks piringan teratas pada menara A atau menara C kosong, pindahkan dari A ke C.
     Jika indeks piringan teratas pada menara C < indeks piringan teratas pada menara A atau menara A kosong, pindahkan dari C ke A.
     Total langkah = total langkah + 1
  2. Jika indeks piringan teratas pada menara B > indeks piringan teratas pada menara A atau menara B kosong, pindahkan dari A ke B.
     Jika indeks piringan teratas pada menara B < indeks piringan teratas pada menara A atau menara A kosong, pindahkan dari B ke A.
     Total langkah = total langkah + 1
  3. Jika indeks piringan teratas pada menara B > indeks piringan teratas pada menara C atau menara B kosong, pindahkan dari C ke B.
     Jika indeks piringan teratas pada menara B < indeks piringan teratas pada menara C atau menara C kosong, pindahkan dari B ke C.
     Total langkah = total langkah + 1
  4. Ulangi langkah 1 – 3 hingga permainan selesai (total langkah = $2^n - 1$ langkah)

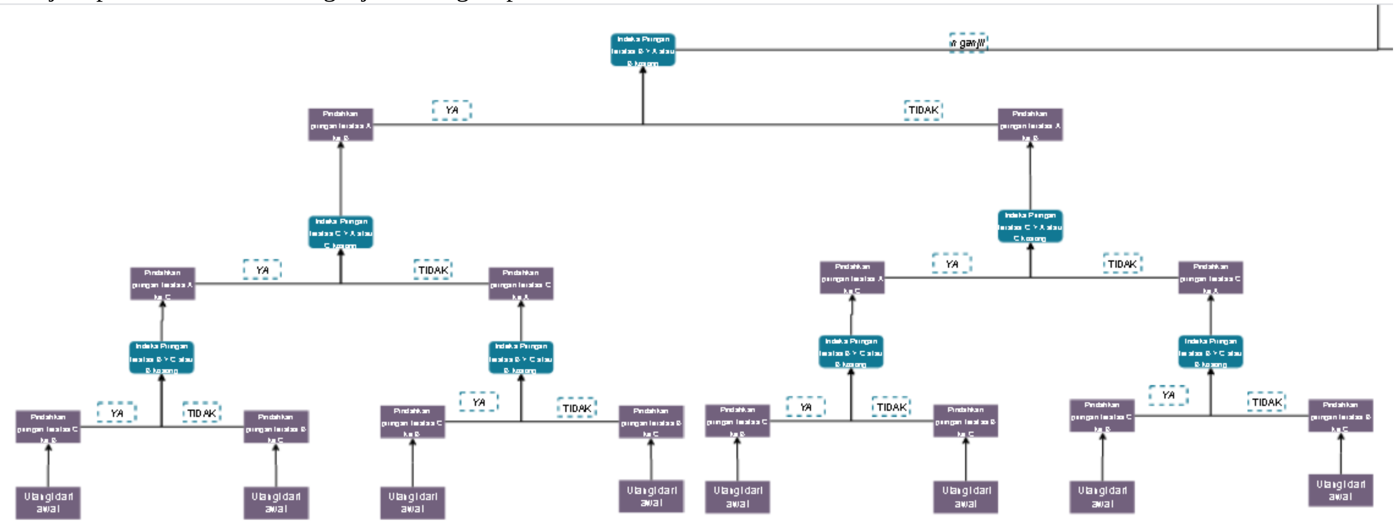
Berikut adalah pemodelan pohon keputusan dari langkah-langkah di atas. Karena keterbatasan ruang, pohon dibagi menjadi pohon untuk kasus *n* ganjil dan *n* genap

IV. SIMPULAN

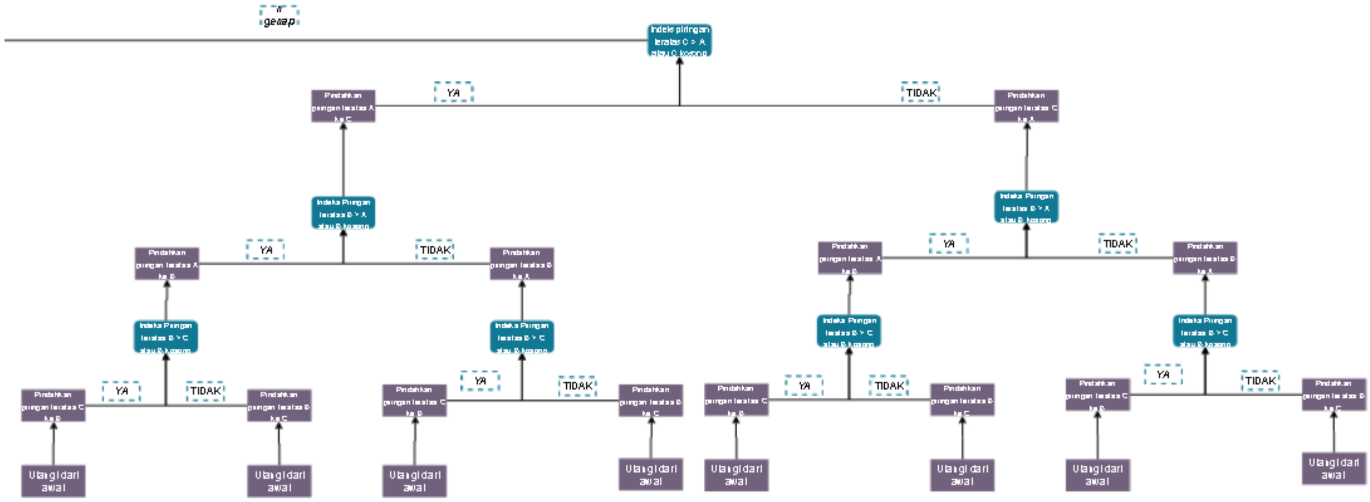
Teori matematika diskrit dapat dimanfaatkan dalam kehidupan. Salah satu pemanfaatan dari konsep pohon adalah untuk membantu menyelesaikan permasalahan dalam permainan Menara Hanoi. Permasalahan dalam permainan Menara Hanoi dapat diselesaikan menggunakan metode rekursif dan metode iteratif. Metode iteratif dari penyelesaian permainan Menara Hanoi dapat dimodelkan dalam bentuk pohon keputusan, sehingga langkah-langkah dalam menyelesaikan permainan dapat lebih mudah dipahami oleh pemain yang memainkan permainan.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan YME karena atas karunia-Nya sehingga penulis dapat menyelesaikan makalah ini tepat pada waktu yang ditetapkan. Juga terima kasih kepada seluruh pihak yang telah membantu pembuatan makalah ini, baik secara langsung maupun tidak langsung. Secara khusus, penulis mengucapkan terima kasih sebesar-besarnya atas bimbingan Bapak dan Ibu dosen mata kuliah IF2120 Matematika Diskrit selama semester ini.



Gambar 11. Subpohon ganjil dari proses iteratif



Gambar 12.. Subpohon genap dari proses iteratif

REFERENCES

[1] *Hinz, Andreas M.; Klavžar, Sandi; Milutinović, Uroš; Petr, Ciril (2013-01-31). The Tower of Hanoi – Myths and Maths.*

[2] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021. Diakses pada 6 Desember 2020.

[3] https://www.freecodecamp.org/news/analyzing-the-algorithm-to-solve-the-tower-of-hanoi-problem-686685f032e3/. Diakses pada 11 Desember 2020

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Tangerang, 5 Desember 2020

Mohammad Sheva Almeyda Sofjan
13519018