Aplikasi Pohon Keputusan dan Aljabar Boolean Dalam Diagnosis Autisme Anak

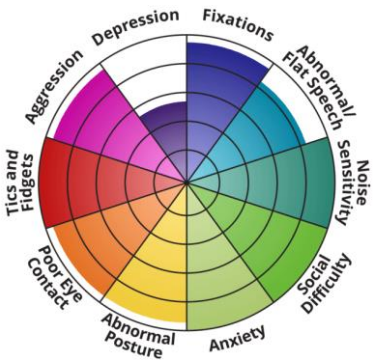
Thea Josephine Halim - 13522012[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522012@mahasiswa.itb.ac.id

***Abstract*— Autisme adalah kelainan gangguan syaraf yang menyebabkan penderitanya mengalami kesusahan dalam berkomunikasi ataupun berkonsentrasi pada satu hal. Dari berbagai tanda-tanda gejala, autisme dapat dibedakan menjadi 3 level. Pembagian level ini berguna untuk menentukan penanganan profesional yang diperlukan, dan bukan menjadi patokan klasifikasi autisme. Pada makalah ini akan dilakukan analisis mengenai penggunaan pohon keputusan untuk menentukan apakah seorang anak mengalami autisme atau tidak, dan jika ya, pada level berapakah.**

***Keywords*—Pohon Keputusan, Aljabar Boolean, Autisme, Autism Spectrum Disorder.**

I. PENDAHULUAN

Salah satu gangguan kelainan pada anak yang sering menjadi topik bahasan adalah autisme pada anak. Autisme atau *Autism Spectrum Disorders* (ASD) dapat menyebabkan anak kesulitan berinteraksi dengan orang lain. Anak penderita autis juga kerap melakukan perilaku yang tidak lazim, seperti menggaruk-garuk, mencubit, berputar-putar, atau kelakuan lain yang membahayakan diri sendiri. Perilaku autisme dapat berbeda-beda pada setiap individu dan kemungkinan berganti-ganti seiring berjalannya waktu. Akan tetapi, kita dapat memetakan perilaku autis dalam spektrum, seperti pada gambar 1.1. Perilaku setiap penderita autis memang bisa berbeda-beda, sehingga untuk mempermudah penentuan penanganan yang diambil, autisme akan dibagi menjadi 3 level berdasarkan tingkat keparahannya, yang akan dibahas pada bab berikutnya.

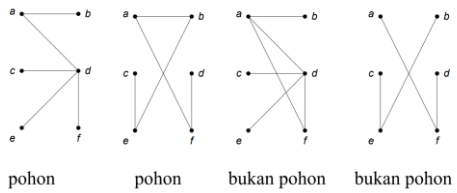


Gambar 1.1 Spektrum autisme
Sumber:https://childresidentialtreatment.com/autism-spectrum/

Pengklasifikasian autisme menjadi 3 level ini dapat kita jabarkan dengan konsep pohon keputusan. Dengan setiap *trait* autisme sebagai input simpul pohon tersebut. Perbedaan tingkat *traits* ini akan menjadi penentu dalam level autisme.

II. LANDASAN TEORI

A. Pohon Keputusan



Gambar 2.1 Ilustrasi diagram pohon
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/22-Pohon-Bag1-2023.pdf

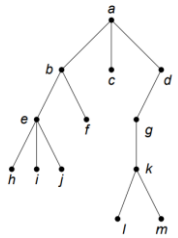
Dilansir dari Homepage Rinaldi Munir, pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Sifat-sifat dari pohon adalah sebagai berikut. Misal $G = (V, E)$ adalah graf tak-berarah sederhana dan memiliki jumlah simpul n. Maka akan memenuhi:

1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki $m = n - 1$ buah sisi.
4. G tidak mengandung sirkuit dan memiliki dan memiliki $m = n - 1$ buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

Pohon berakar (*rooted tree*) adalah salah satu jenis pohon yang simpulnya diperlakukan sebagai akar dan memiliki sisi dengan arah. Contoh pada gambar 2.2, sebuah pohon berakar memiliki terminologi sebagai berikut:

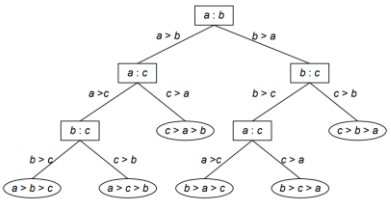
1. Anak dan orangtua. Simpul b, c, dan d sebagai anak simpul a, sedangkan a adalah orangtua dari anak-anak itu.
2. Lintasan (*path*). Lintasan dari *a* ke *j* adalah a, b, e, j.
3. Saudara kandung (*sibling*). Simpul f saudara kandung e.

4. Upapohon (*subtree*).
5. Derajat (*degree*). Derajat a adalah 3, derajat c adalah 0.
6. Daun (*leaf*). Simpul yang berderajat 0.
7. Simpul dalam (*internal nodes*). Simpul yang punya anak, contoh simpul b, d, dan e.
8. Aras (level)
9. Tinggi (*height*) atau kedalaman (*depth*)



Gambar 2.2 Contoh pohon berakar
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/23-Pohon-Bag2-2023.pdf

Pohon biner adalah contoh pohon berakar, tetapi memiliki maksimal 2 anak di setiap simpulnya. Anak kanan dan kiri berbeda, dan merupakan pohon terurut. Salah satu penerapan konsep pohon biner adalah pohon keputusan. Sebuah pohon keputusan dapat menentukan alur *outcome* beberapa input. Setiap simpul merupakan sebuah atribut yang diuji, dan berdasarkan input dari pengguna, pohon akan membawa pengguna ke cabang berikutnya, untuk diuji dengan atribut yang lain lagi. Proses akan berlangsung terus-menerus hingga tercapai daun pohon, yang melambangkan output.



Gambar 2.3 Contoh pohon keputusan
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/23-Pohon-Bag2-2023.pdf

B. Aljabar Boolean

Aljabar Boolean adalah struktur matematika yang menggunakan sistem aljabar dua nilai, nilai benar (truth/yes) dan nilai salah (false/no). Kedua nilai ini juga kerap kali dilambangkan oleh angka 1 dan 0. Konsep ini pertama kali diperkenalkan oleh George Boole pada 1854. George Boole adalah matematisi asal Inggris yang berhasil menghubungkan antara konsep aljabar logika dan himpunan. Aljabar boolean Aljabar boolean sering digunakan dalam penyederhanaan sirkuit rangkaian mikroprosesor, memungkinkan melakukan penghematan biaya, bahan, dan peningkatan efisiensi.

Dilansir dari Homepage Rinaldi Munir, aljabar boolean dapat didefinisikan sebagai berikut. Misalkan B adalah himpunan yang didefinisikan pada dua operator biner + dan ·, dan sebuah operator under, ’. Misalkan angka 0 dan 1 adalah elemen yang berbeda dari B. Dapat dituliskan tupel yang terbentuk adalah <B,+, ·,’,0,1>, yang kemudian dapat kita sebut sebagai Aljabar Boolean, jika untuk setiap a, b, c ∈ B berlaku aksioma sebagai berikut:

1. Indentitas
   a. a + 0 = a
   b. a · 1 = a
2. Komutatif
   a. a + b = b + a
   b. a · b = b · a
3. Distributif
   a. a · (b + c) = (a · b) + (a · c)
   b. a + (b · c) = (a + b) · (a + c)
4. Komplemen
   Untuk setiap a ∈ B terdapat elemen unik a‘ ∈ B sehingga
   a. a + a’ = 1
   b. a · a’ = 0

Aljabar boolean dua nilai dapat kita definisikan dengan masukan elemen {0, 1}, operator biner + dan ·, serta operator uner ‘. Kaidah penerapan operator biner dan uner tersebut dapat dituliskan seperti berikut:

a	b	a · b
0	0	0
0	1	0
1	0	0
1	1	1

Tabel 2.1 Kaidah Perkalian Operator Biner dan Uner
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/11-Aljabar-Boolean-(2023)-bagian1.pdf

a	b	a + b
0	0	0
0	1	1
1	0	1
1	1	1

Tabel 2.2 Penjumlahan Operator Biner dan Uner
Sumber:https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/11-Aljabar-Boolean-(2023)-bagian1.pdf

a	a’
0	1
1	0

Tabel 2.3 Komplemen Operator Biner dan Uner
https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/11-Aljabar-Boolean-(2023)-bagian1.pdf

Adapun beberapa hukum-hukum aljabar boolean yang dapat digunakan untuk pembuktian aljabar boolean ataupun mengoperasikan literal-literal suatu fungsi boolean.

1. Hukum identitas: (i) a + 0 = a (ii) a · 1 = a	2. Hukum idempoten: (i) a + a = a (ii) a · a = a
3. Hukum komplemen: (i) a + a’ = 1 (ii) aa’ = 0	4. Hukum dominansi: (i) a · 0 = 0 (ii) a + 1 = 1

5. Hukum involusi: (i) (a')' = a	6. Hukum penyerapan: (i) a + ab = a (ii) a · (a + b) = a
7. Hukum komutatif: (i) a + b = b + a (ii) ab = ba	8. Hukum asosiatif: (i) a + (b + c) = (a + b) + c (ii) a (b c) = (a b) c
9. Hukum distributif: (i) a + (b c) = (a + b) (a + c) (ii) a · (b + c) = a b + a c	10. Hukum De Morgan: (i) (a + b)' = a' · b' (ii) (ab)' = a' + b'
11. Hukum 0/1: (i) 0' = 1 (ii) 1' = 0	

Tabel 2.4 Hukum-Hukum Aljabar Boolean

Aljabar boolean dapat dituliskan dalam dua bentuk kanonik:

1. *Sum of Product* (SOP)

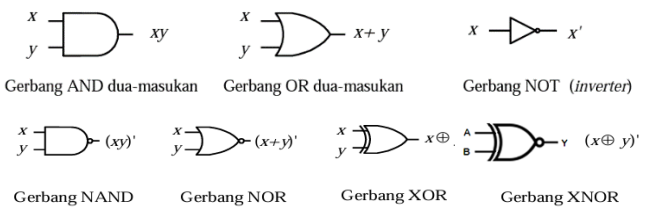
   Sum of product atau penjumlahan dari hasil kali beberapa literal. *Minterm* (dilambangkan dengan 'm' dan ∑) adalah suku (*term*) yang berisi produk perkalian literal, dengan masing-masing jenis literal harus muncul setidaknya sekali dan tidak lebih. Dalam *minterm*, bentuk komplemen literal dinyatakan sebagai 0, sedangkan 1 adalah dilambangkan tanpa komplemen. Contoh dari *minterm* dengan tiga literal adalah xyz, x'yz, dan x'y'z. SOP inilah bentuk kanonik yang terdiri dari penjumlahan satu atau beberapa *minterm.*

2. *Product of Sum* (POS)

   Product of Sum atau perkalian dari hasil penjumlahan beberapa literal. *Maxterm* (dilambangkan dengan 'M' dan ∏) adalah suku (*term*) yang berisi produk penjumlahan literal, dengan masing-masing jenis literal harus muncul setidaknya sekali dan tidak lebih. Dalam *maxterm*, komplemen literal bernilai 1, sedangkan literal tanpa komplemen melambangkan nilai 0. Contoh dari *maxterm* dengan tiga literal adalah x + y + z, x' + y + z, dan x' + y' + z. SOP inilah bentuk kanonik yang terdiri dari penjumlahan satu atau beberapa *maxterm.*

Kedua bentuk kanonik ini juga dapat dikonversi satu sama lain. SOP merupakan bentuk komplemen dari POS, begitu pula sebaliknya. Dengan mendapatkan fungsi salah satu kanonik, kita dapat mendapatkan bentuk kanonik yang lain.

Aljabar boolean juga dapat dipaparkan dalam bentuk rangkaian logika. Gerbang-gerbang logika utama yang digunakan adalah gerbang AND, gerbang OR, dan gerbang NOT. Sedangkan gerbang tambahan yang lain adalah gerbang NAND, NOR, XOR, dan XNOR.



Gambar 2.4 Gerbang-gerbang logika

Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/11-Aljabar-Boolean-(2023)-bagian1.pdf

Suatu rangkaian struktur aljabar boolean juga dapat kita sederhanakan menggunakan Peta Karnaugh (*Karnaugh Map*). Peta Karnaugh adalah salah satu metode penyederhanaan aljabar boolean di samping metode tabulasi dan penggunaan hukum-hukum aljabar boolean. Ditemukan oleh Maurice Karnaugh, metode ini menggunakan diagram semacam tabel bersisian yang masing-msaing berisi *minterm*. Setiap pergantian kolom dan baris hanya boleh terjadi perbedaan 1 literal, dan isi suatu sel tabel adalah 0 dan 1. Pengisian angka 0 dan 1 ini dapat dibantu dengan *truth table.*

Berikut adalah contoh Peta Karnaugh dari fungsi boolean $f(x,y) = x'y'z + xy'z' + xy'z + xyz$ dengan membuat tabel kebenaran terlebih dahulu

x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

		yz		
		00	01	11	10
x	0	0	1	0	0
	1	1	1	1	0

Gambar 2.5 Contoh Pembuatan Peta Karnaugh dengan Truth Table

https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/11-Aljabar-Boolean-(2023)-bagian1.pdf

Meminimalisasi aljabar boolean dapat dilakukan dengan melakukan pengelompokan nilai cell aljabar boolean yang bernilai 1 pada Peta Karnaugh. Pengelompokan ini dapat dilakukan dengan jumlah 8 (oktet), 4 (quad), dan 2 (pasangan), dengan diusahakan mencari pasangan oktet sebanyak mungkin, dan dilanjut dengan pasangan quad sebanyak mungkin.

C. *Autisme dan Tingkatannya*

Autis (*Autism Spectrum Disorder)* adalah kelainan yang disebabkan orang penderitanya mengalami gangguan sistem saraf. Gangguan saraf ini menyebabkan penderitanya mengalami berbagai masalah sosial, kesulitan dalam berkomunikasi, hiperaktif, dan lain-lain. Dilansir dari psychologicaltoday.com, anak penderita autisme baik tingkat rendah ataupun tinggi akan sama-sama memiliki keinginan komunikasi yang sedikit hingga tidak ada, sensitif pada perubahan lingkungan, dan memiliki ketertarikan pada topik tertentu saja. Perlu diingat bahwa autisme memang tidak bisa dipetakan secara linear karena autisme dapat berbeda-beda di setiap individu. Akan tetapi, untuk mempermudah penentuan penanganannya autisme dibagi menjadi tiga level berdasarkan tingkat keparahannya:

1. Level 1: Butuh dukungan (*mild*).

   Merupakan level terendah autisme, paling sulit dideteksi karena gejala dan tanda yang tidak kentara. Anak kesulitan dalam menghadapi perubahan lingkungan, melakukan rutinitas yang sama yang menyebabkan mereka kesulitan dalam mengatur sebuah rencana. Akibatnya, penderita akan mengalami stress jika ada perubahan tiba-tiba. Penderita autisme level 1 juga tidak memiliki ketertarikan pada sosialisasi dan mengalami sedikit kesulitan dalam berkomunikasi, tetapi masih bisa mengkomunikasikan keinginan dan kebutuhannya dengan

baik.

Pada level ini, anak mungkin akan kesulitan mencari teman tanpa bantuan dan bimbingan orang lain. Akan tetapi, anak mungkin hanya perlu bantuan yang minimal dalam aktivitas sehari-harinya, seperti waktu tambahan ketika ujian sekolah.

2. Level 2: Butuh dukungan lebih (*mid-range*)

Merupakan level menengah autism, di level ini anak mungkin membutuhkan bantuan orang lain untuk menangani masalah-masalah yang seharusnya mudah bagi anak seumurnya. Penderita autisme level 2 mengalami gejala yang sama dengan level 1, tetapi jauh lebih sulit dalam berkomunikasi, seperti penggunaan kata-kata yang minim dan tidak ada ekspresi wajah atau pergi meninggalkan percakapan di tengah-tengah (perilaku tidak lazim). Anak juga mulai menunjukkan tanda-tanda *stress* jika ada peristiwa yang tiba-tiba.

Anak dengan autisme level 2 akan membutuhkan bantuan khusus sekolah, seperti *reading support* dan tugas yang disesuaikan dengan keadaan mereka.

3. Level 3: Butuh dukungan khusus (*severe*)

Merupakan level tertinggi dan terparah dari autisme. Pada level ini, anak akan kesulitan dalam kesehariannya, dan rata-rata tidak bisa berkomunikasi secara verbal. Penderita kerap berperilaku tidak lazim, seperti mengulang-ulang kata yang sama, berputar-putar, hingga melakukan tindakan yang menyakiti diri sendiri. Penderita autis level 3 sangatlah sensitif pada suara, rasa, dan penciumannya.

Anak dengan autisme level 3 akan membutuhkan bantuan yang sama seperti level 2, tetapi lebih sering dan didampingi dengan terapi-terapi. Anak akan membutuhkan perlakuan sekolah khusus seperti ruangan pengajaran yang terpisah.

III. PEMBAHASAN

A. Mapping dengan Pohon Keputusan

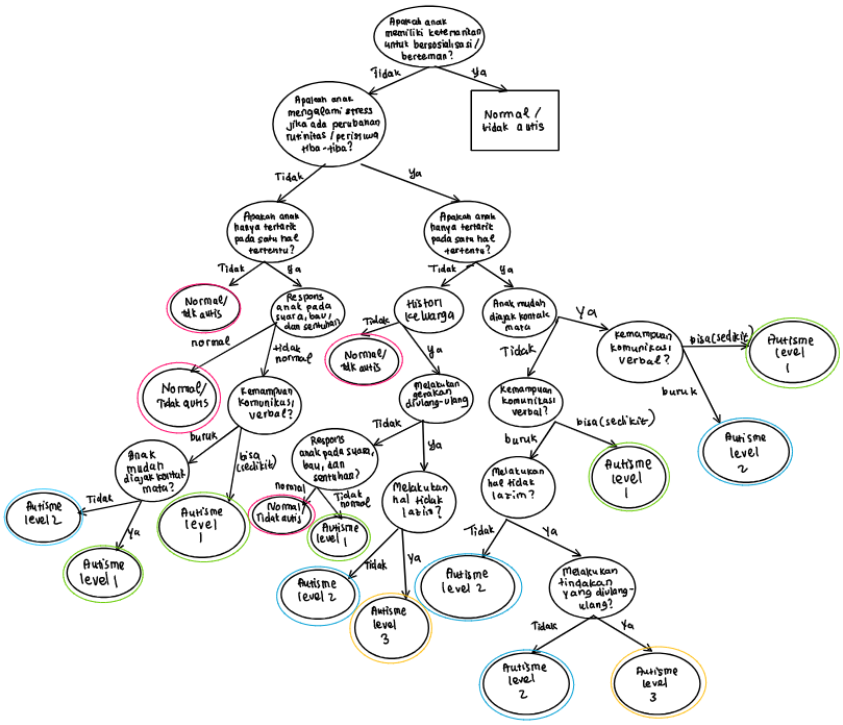
Masukan dari pengguna yang digunakan pada analisis pohon keputusan ini adalah gejala-gejala dari autisme. Pohon akan melakukan *tracing* ke output yang sesuai dengan rules alur yang telah dipetakan sebelumnya. Output yang muncul akan menampilkan apakah anak menderita autisme atau tidak, dan jika ya, akan menampilkan level autismenya.

Gambar 3.1 menunjukkan pohon keputusan dari berbagai fakta yang telah kita definisikan pada bab sebelumnya. Pohon keputusan akan dimulai dari pertanyaan paling sederhana dan menentukan, apakah seorang anak memiliki ketertarikan untuk bersosialisasi. Apabila pengguna menjawab ya, maka langsung muncul output anak tidak autis, sedangkan jika pengguna menjawab tidak, akan dilakukan pengujian berikutnya. Proses berlangsung terus-menerus dengan pohon mulai berakar ke input-input lain sebagai atribut pendukung hingga tercapai daun, atau simpul tanpa cabang yang menandakan output.

B. Penyederhanaan dengan Aljabar Boolean

Berdasarkan dasar teori yang telah dijabarkan dan referensi-referensi, dalam penentuan level autisme, gejala (*symptoms*) dari autisme dapat dituliskan dalam tabel ini. Pada tabel juga akan dituliskan simbol nama variabel yang akan digunakan di proses penyederhanaan aljabar boolean beserta artinya:

Input	Simbol/Variabel
Kebutuhan rutinitas	r = Anak mengalami stress jika ada perubahan rutinitas r'= Anak tidak mempermasalahkan perubahan rutinitas



Gambar 3.1 Pohon Keputusan
Sumber:Penulis

Keinginan bersosialisasi/berteman	s = Memiliki ketertarikan untuk interaksi sosial s'= Tidak memiliki ketertarikan untuk interaksi sosial
Histori keluarga (ada anggota keluarga dengan autisme)	t = Ada t'= Tidak ada
Respons dengan suara, sentuhan, dan bau (sensitivitas)	u = Tidak normal u'= Normal
Ketertarikan hanya pada topik/hal tertentu (*fixation)*	v = Anak hanya menunjukkan ketertarikan pada topik tertentu v'= Anak dapat menunjukkan ketertarikan pada banyak topik
Kemampuan komunikasi verbal	w = Komunikasi buruk dan susah dimengerti w'= Anak masih/mampu berkomunikasi dengan baik dan mudah dimengerti
Kontak mata	x = Anak tidak mampu kontak mata ketika diajak bicara x'= Anak mampu kontak mata ketika diajak bicara
Melakukan suatu tindakan yang diulang-ulang (berputar-putar, berkedip-kedip, dll)	y = Anak suka melakukan tindakan yang berulang-ulang y'= Anak tidak pernah/jarang melakukan tindakan yang berulang-ulang
Tindakan yang tidak lazim (agresi, menyakiti diri sendiri)	z = Sering/terkadang z'= Tidak pernah/jarang

Tabel 3.1 Tabel Input/simpul

Berdasarkan pohon keputusan yang telah dibuat (Gambar 3.1) kita dapat meminimalisasi ekspresi boolean yang menyebabkan munculnya suatu output. Akan tetapi, untuk mempermudah analisis dan pemahaman, kita ambil penyederhanaan ekpresi boolean yang menyebabkan anak tergolong normal/tidak autis. Pemetaan ekspresi boolean nonautis terlampirkan pada tabel berikut:

Ekspresi Boolean	**Penjelasan**
s	Memiliki ketertarikan untuk interaksi sosial
s' r' v'	Sedikit/tidak memiliki ketertarikan untuk interaksi sosial Anak tidak mempermasalahkan perubahan rutinitas Anak dapat menunjukkan ketertarikan pada semua topik
s' r' v' u'	Sedikit/tidak memiliki ketertarikan untuk interaksi sosial Anak tidak mempermasalahkan perubahan rutinitas Anak dapat menunjukkan ketertarikan pada banyak topik Respons normal pada suara, sentuhan, dan bau
s' r v' t'	Sedikit/tidak memiliki ketertarikan untuk interaksi sosial Anak mengalami stress jika ada perubahan rutinitas Anak dapat menunjukkan ketertarikan pada banyak topik Tidak ada histori pemilik autis di keluarga
s' r v' t y' u'	Sedikit/tidak memiliki ketertarikan untuk interaksi sosial Anak mengalami stress jika ada perubahan rutinitas Anak dapat menunjukkan ketertarikan pada banyak topik Ada histori pemilik autis di keluarga Anak tidak pernah/jarang melakukan tindakan yang berulang-ulang Respons normal pada suara, sentuhan, dan bau

Tabel 3.2 Ekpresi Boolean Penanda Anak Nonautis

Sesuai dengan deskripsi tabel 3.1 dan hasil kumpulan ekspresi boolean tabel 3.2, dapat kita rumuskan ekspresi boolean nontautis sebagai berikut:

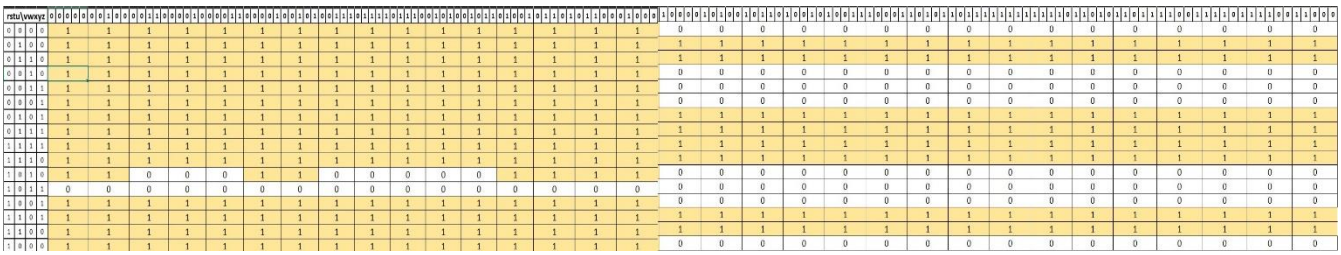
$$f(r,s,t,u,v,w,x,y,z) = s + s'r'v' + s'r'v'u' + s'rv't' + s'rv'ty'u'$$

Dengan menggunakan konsep Peta Karnaugh (Gambar 3.3), kita dapat meminimalisasi ekspresi boolean penanda seseorang normal/tidak autis seperti tabel berikut:

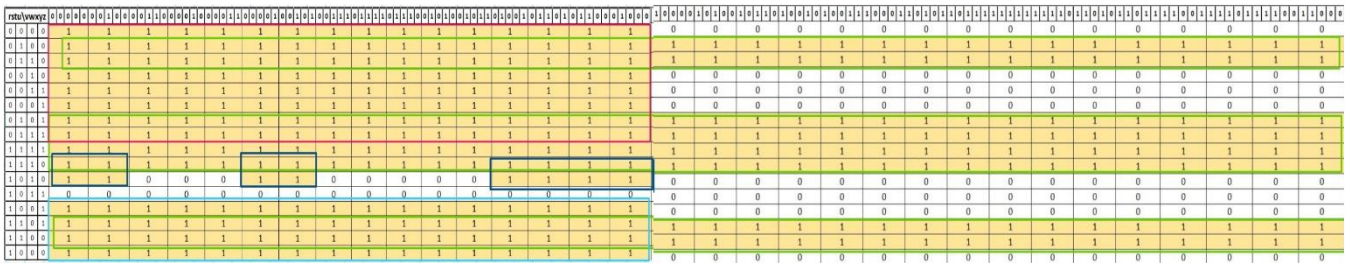
No	**Ekspresi Boolean**	**Penjelasan**
1.	s	Memiliki ketertarikan untuk interaksi sosial
2.	r u' v' y'	Anak mengalami stress jika ada perubahan rutinitas Respons normal pada suara, sentuhan, dan bau Anak dapat menunjukkan ketertarikan pada banyak topik Anak tidak pernah/jarang melakukan tindakan yang berulang-ulang
3.	r' v'	Anak tidak mengalami stress jika ada perubahan rutinitas Anak dapat menunjukkan ketertarikan pada banyak topik
4.	r t' v'	Anak mengalami stress jika ada perubahan rutinitas Tidak ada histori pemilik autis di keluarga Anak dapat menunjukkan ketertarikan pada banyak topik

Tabel 3.3 Ekpresi Boolean Penanda Anak Nonautis Setelah Penyederhanaan

Sebagai catatan tambahan, karena peta Karnaugh yang berukuran cukup besar, untuk akses lebih jelas dapat dilihat lewat link berikut ini. Hasil peta Karnaugh dapat kita lihat pada gambar berikut:



Gambar 3.2 Peta Karnaugh



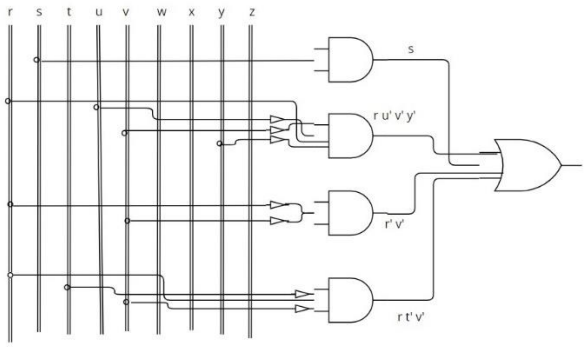
Gambar 3.3 Peta Karnaugh Setelah Grouping

Teknik penyederhanaan peta Karnaugh dimulai dengan menuliskan semua kemungkinan variabel yang ada. Pada analisis kali ini kita menggunakan 9 variabel, mulai dari alfabet r hingga z, maka kita bagi pada bagian kolom untuk v,w,x,y,z, dan bagian baris untuk r,s,t,u. Perlu dipastikan juga bahwa pengisian nama variabel pada bagian header kolom dan baris diwakilkan dengan 1 dan 0. Angka 1 mewakilkan variabel itu sendiri (contoh: r, s, dan v), sedangkan angka 0 mewakilkan komplemennya (contoh: r', s', dan v'), dengan setiap pergantian kolom dan barisnya mengikuti aturan perubahan satu variabel. Selanjutnya kita akan melakukan pengisian sebanyak 2^9 elemen. Kita akan mengisi elemen dengan angka 0 jika output yang dihasilkan adalah autis, dan angka 1 jika output normal/nonautis.

Setelah semuanya dipetakan pada masing-masing elemen sel, kita perlu melakukan grouping angka-angka 1. Didapatkan hasil ekspresi boolean sebagai berikut:

$$f'(r,s,t,u,v,w,x,y,z) = s + r\,u'\,v'\,y' + r'v' + rt'v'$$

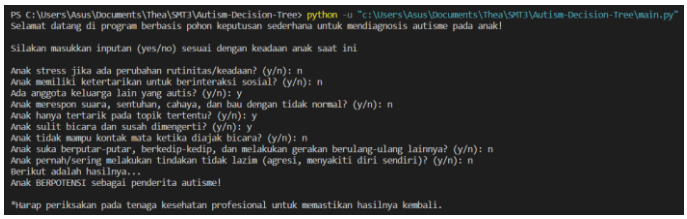
Selanjutnya dari hasil ekspresi boolean, dapat kita gambarkan lewat ilustrasi gerbang logika sebagai berikut:



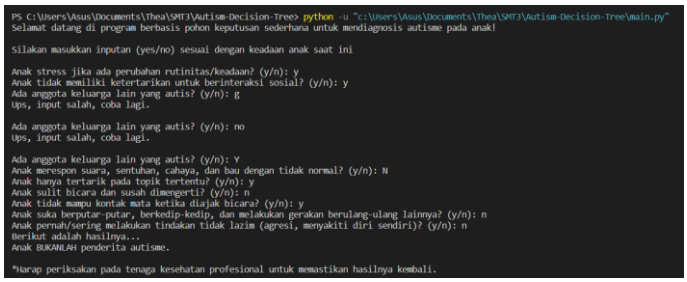
Gambar 3.4 Hasil Gerbang Logika

IV. Eksperimen Program

Untuk merealisasikan gambaran kita mengenai pohon keputusan dan hasil penyederhanaan aljabar boolean, telah dibuat sebuah program python sederhana. Implementasinya adalah rules case1 hingga case4 yang didapat dari fungsi boolean hasil penyederhanaan peta Karnaugh.



Gambar 4.1 Contoh Kasus Autis



Gambar 4.2 Contoh Kasus Nonautis

Berikut adalah kode program python yang digunakan, atau akses lewat link berikut:

File main.py

```
import function as fn
print("Selamat datang di program berbasis pohon keputusan sederhana untuk mendiagnosis autisme pada anak!\n")
print("Silakan masukkan inputan (yes/no) sesuai dengan keadaan anak saat ini\n")
question={
    "r":"Anak stress jika ada perubahan rutinitas/keadaan?",
    "s":"Anak memiliki ketertarikan untuk berinteraksi sosial?",
    "t":"Ada anggota keluarga lain yang autis?",
    "u":"Anak merespon suara, sentuhan, cahaya, dan bau dengan tidak normal?",
    "v":"Anak hanya tertarik pada topik tertentu?",
    "w":"Anak sulit bicara dan susah dimengerti?",
    "x":"Anak tidak mampu kontak mata ketika diajak bicara?",
    "y":"Anak suka berputar-putar, berkedip-kedip, dan melakukan gerakan berulang-ulang lainnya?",
```

```
    "z":"Anak pernah/sering melakukan tindakan tidak lazim (agresi,
menyakiti diri sendiri)?",
}

answer = {key: fn.getinput(value) for key, value in question.items()}
isR = answer["r"]
isS = answer["s"]
isT = answer["t"]
isU = answer["u"]
isV = answer["v"]
isW = answer["w"]
isX = answer["x"]
isY = answer["y"]
isZ = answer["z"]
print("Berikut adalah hasilnya... ")
case1 = isS==1
case2 = isR==1 and isU==0 and isV==0 and isY==0
case3 = isR==0 and isV==0
case4 = isR==1 and isT==0 and isV==0
if (case1 or case2 or case3 or case4):
    print("Anak BUKANLAH penderita autisme.\n")
else:
    print("Anak BERPOTENSI sebagai penderita autisme!\n")
print("*Harap periksakan pada tenaga kesehatan profesional untuk
memastikan hasilnya kembali.\n")
```

File function.py

```
def convert_bool(input_str):
    if input_str.lower() == "y":
        return 1
    elif input_str.lower() == "n":
        return 0
    else:
        print("Ups, input salah, coba lagi.\n")
        return False

def getinput(question):
    not_stop = True
    while not_stop:
        user_input = input(question + " (y/n): ")
        result = convert_bool(user_input)
        if result is not False:
            not_stop = False
    return result
```

V. Kesimpulan

Dari berbagai faka yang kita miliki, kita dapat memetakan sebuah pohon keputusan yang dapat membantu kita menentukan output dari berbagai fakta yang kita miliki. Setiap simpulnya mewakili sebuah pernyataan, dan bergantung pada input pengguna, pohon akan membawa pengguna menuju daun output. Pohon keputusan menentukan autisme ini juga dapat kita sederhanakan dengan menggunakan aljabar boolean dan teknik konsep peta Karnaugh, dengan representasi angka 0 sebagai autis dan 1 sebagai nonautis/normal. Dengan begitu, kita dapat menentukan faktor apa yang menyebabkan anak termasuk tidak autis.

VI. Ucapan Terima Kasih

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa atas penyertaan dan berkat yang telah diberikan selama ini sehingga makalah yang berjudul “Aplikasi Pohon Keputusan dan Aljabar Boolean Dalam Diagnosis Autisme Anak” ini dapat diselesaikan dengan baik. Saya juga ingin mengucapkan terima kasih pada pihak-pihak yang telah membantu dalam penyusunan makalah ini:

1. Ibu Dr. Nur Ulfa Maulidevi, selaku dosen pengajar mata kuliah IF2120 Matematika Diskrit K01, atas bimbingan dan pengajaran yang sangat membantu saya dalam memahami materi,
2. Orang tua saya yang selalu mendukung dan membantu saya selama ini,
3. Teman-teman dan sahabat-sahabat yang memberikan saya semangat,
4. Penulis-penulis jurnal, artikel, dan laman website yang tidak bisa saya sebutkan satu per satu atas sumber informasi yang sangat bermanfaat sebagai bahan referensi makalah ini.

Referensi

[1] Jack, C. (2016, August 22). *From Autistic Linear Spectrum to Pie Chart Spectrum*. Retrieved from psychologytoday: https://www.psychologytoday.com/gb/blog/women-autism-spectrum-disorder/202208/autistic-linear-spectrum-pie-chart-spectrum (diakses pada 8 Desember 2023)

[2] Lovering, N. (2022, November 10). *What Are the 3 Levels of Autism?*. Retrieved from psychcentral: https://psychcentral.com/autism/levels-of-autism#level-2 (diakses pada 8 Desember 2023)

[3] Andreasen, H. (2022, April 22). *Levels of Autism: Common Symptoms per Level*. Retrieved from songbirdcare: https://www.songbirdcare.com/articles/levels-of-autism (diakses pada 8 Desember 2023)

[4] Lively, J. (2022, September 13). *What Does the Autism Spectrum Mean?*. Retrieved from childresidentialtreatment: https://childresidentialtreatment.com/autism-spectrum/ (diakses pada 8 Desember 2023)

[5] Munir, R. (2023) Homepage Rinaldi Munir. Sekolah Teknik Elektro dan Informatika (STEI) ITB. Retrieved from childresidentialtreatment: https://informatika.stei.itb.ac.id/~rinaldi.munir/ (diakses pada 26 November 2023)

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.

Bandung, 9 Desember 2023

Thea Josephine Halim
(13522012)