

Implementasi Pohon Keputusan dan Metode Quine-McCluskey dalam Deteksi Gejala ADHD pada Anak Usia Dini

Adinda Putri - 13523071¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

adindaputri0517@gmail.com, 13523071@std.stei.itb.ac.id

Abstrak—*Attention Deficit Hyperactivity Disorder (ADHD)* adalah gangguan perkembangan yang ditandai dengan kesulitan dalam konsentrasi, perilaku hiperaktif, dan impulsivitas. Gangguan ini dapat terjadi pada anak usia dini dan memengaruhi kemampuan anak dalam berinteraksi dan belajar. ADHD diklasifikasikan menjadi tiga jenis utama, yaitu tipe dominan inatentif, tipe dominan hiperaktif-impulsif, dan tipe gabungan. Pada makalah ini, dilakukan analisis penerapan pohon keputusan dan metode penyederhanaan Quine-McCluskey untuk mendeteksi apakah seorang anak memiliki gejala ADHD. Pendekatan ini bertujuan untuk menyediakan metode berbasis data yang dapat membantu orang tua atau pendidik dalam memahami gejala awal ADHD sehingga mendukung proses deteksi dini.

Kata Kunci—Tipe ADHD, Pohon Keputusan, Quine-McCluskey, Deteksi ADHD

I. PENDAHULUAN

Attention Deficit Hyperactivity Disorder (ADHD) merupakan salah satu gangguan mental pada anak usia dini yang paling umum ditemukan [1]. ADHD didefinisikan sebagai gangguan perkembangan saraf yang ditandai dengan perilaku inatentif, hiperaktivitas, dan/atau impulsivitas. Perilaku inatentif mengacu pada ketidakmampuan penderitanya untuk mempertahankan kefokusannya dalam waktu yang lama, sehingga mudah teralih dan terlihat seperti tidak mendengarkan. Sementara itu, hiperaktivitas dan impulsivitas ditandai dengan perilaku yang berlebihan, seperti ketidakmampuan anak untuk diam dalam waktu yang lama. [2]

ADHD sering berlanjut hingga dewasa dan dapat menimbulkan dampak negatif terhadap kehidupan penderitanya. Gangguan tersebut tidak hanya memengaruhi prestasi akademik, tetapi juga berdampak pada kemampuan sosial penderita dalam interaksi sehari-hari. Penderita ADHD memiliki risiko yang lebih tinggi untuk mengalami kegagalan akademik dan konflik dalam hubungan intrapersonal, baik di lingkungan keluarga maupun masyarakat. [2] Untuk mengurangi dampak negatif tersebut, diperlukan pemahaman yang menyeluruh mengenai ADHD, termasuk pengenalan terhadap gejala dan klasifikasinya.

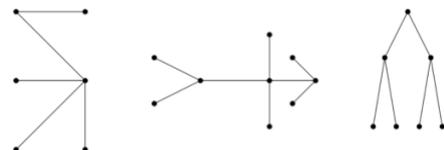
Berdasarkan *Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition (DSM-5)*, ADHD dikategorisasikan menjadi tiga tipe utama, yaitu tipe dominan inatentif, tipe

dominan hiperaktif-impulsif, dan tipe gabungan. Setiap tipe memiliki gejala yang berbeda-beda. [2] Klasifikasi tipe ADHD dapat dijabarkan dengan konsep pohon keputusan. Setiap simpul input merepresentasikan gejala ADHD dan digunakan untuk mengarahkan proses klasifikasi menuju daun yang menentukan salah satu dari tiga tipe ADHD. Untuk meningkatkan efisiensi, hasil klasifikasi yang dihasilkan oleh pohon keputusan dapat disederhanakan menggunakan metode Quine-McCluskey, sehingga menghasilkan proses yang lebih sederhana tanpa mengurangi ketepatan analisis.

II. LANDASAN TEORI

A. Definisi Pohon

Pohon adalah graf tak-berarah yang setiap simpulnya saling terhubung dan tidak mengandung sirkuit. Pohon pertama kali diperkenalkan pada tahun 1857 oleh matematikawan Inggris, Arthur Cayley, yang menggunakannya untuk menghitung jenis-jenis senyawa kimia tertentu. Sejak saat itu, konsep pohon berkembang dan diaplikasikan secara luas untuk menyelesaikan permasalahan dalam berbagai bidang ilmu pengetahuan. [3]



Gambar 1. Tiga ilustrasi pohon

Sumber: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf)

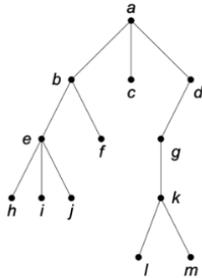
Suatu graf tak-berarah $G = (V, E)$ dengan jumlah simpul n dapat didefinisikan sebagai pohon apabila ekuivalen dengan pernyataan-pernyataan di bawah ini: [5]

1. Setiap pasangan simpul dalam G terhubung melalui tepat satu lintasan tunggal.
2. G terhubung dan memiliki $m = n - 1$ sisi.
3. G tidak mengandung sirkuit dan memiliki $m = n - 1$ sisi.
4. Penambahan satu sisi pada graf G yang tidak mengandung sirkuit akan menghasilkan tepat satu sirkuit.

5. Graf G terhubung dan setiap sisinya merupakan jembatan.

B. Pohon Berakar

Pada struktur graf yang lebih kompleks, pohon berakar sering digunakan untuk memberikan representasi hierarkis yang lebih mudah dipahami. Pohon berakar (rooted tree) adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah. Dalam pohon berakar, setiap simpul dapat dipilih sebagai akar, tergantung pada kebutuhan analisis.

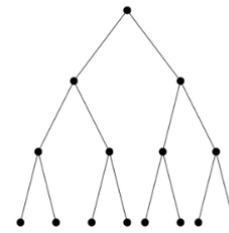


Gambar 2. Ilustrasi pohon berakar

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/24-Pohon-Bag2-2024.pdf>

Dalam pohon berakar, terdapat beberapa terminologi yang digunakan untuk mendeskripsikan dan memperjelas struktur pohon tersebut. Sebagai contoh, terminologi pohon berakar pada Gambar 2 adalah sebagai berikut: [6]

1. Anak (*child*) dan orangtua (*parent*)
Simpul b , c , dan d adalah anak dari a , sedangkan a adalah orang tua dari simpul-simpul tersebut.
2. Lintasan (*path*)
Lintasan dari a ke j adalah a, b, e, j dengan panjang lintasan sebesar 3.
3. Saudara kandung (*sibling*)
Simpul e dan f adalah saudara kandung karena memiliki orangtua yang sama, tetapi g dan e bukan saudara kandung.
4. Upapohon (*subtree*)
Upapohon dengan akar b mencakup b, e, f, h, i, j .
5. Derajat (*degree*)
Simpul b memiliki jumlah anak (derajat) 3, sedangkan c memiliki derajat 0.
6. Daun (*leaf*)
Simpul h, i, j, f, c, l , dan m adalah daun karena tidak memiliki anak.
7. Simpul dalam (*internal node*)
Simpul b, d, e, g , dan k adalah simpul dalam karena memiliki setidaknya satu anak.
8. Aras (*level*)
Simpul a berada di aras 0, b, c, d di aras 1.
9. Tinggi (*height*)
Tinggi pohon adalah 4, karena aras maksimumnya 4.

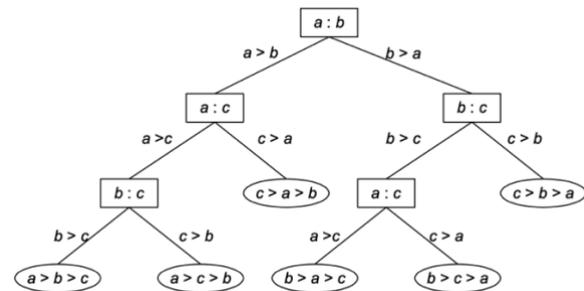


Gambar 3. Ilustrasi pohon biner

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/24-Pohon-Bag2-2024.pdf>

C. Pohon Biner

Bentuk khusus dari pohon berakar yang sering digunakan dalam berbagai aplikasi adalah pohon biner. Pohon biner merupakan jenis pohon berakar di mana setiap simpulnya memiliki paling banyak dua anak, yang biasanya disebut sebagai anak kiri (*left child*) dan anak kanan (*right child*). Salah satu penerapan dari pohon biner adalah dalam pohon keputusan, yang digunakan untuk menyelesaikan berbagai masalah pengambilan keputusan. [6]



Gambar 4. Ilustrasi pohon keputusan

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/24-Pohon-Bag2-2024.pdf>

D. Definisi Aljabar Boolean

Aljabar Boolean adalah konsep matematika yang menggunakan sistem aljabar dua nilai, yaitu nilai benar (*true/yes*) dan nilai salah (*false/no*). Konsep ini pertama kali diperkenalkan oleh George Boole pada tahun 1854 melalui karya tulisnya, *The Laws of Thought*. Karya tersebut menguraikan aturan dasar logika yang menjadi dasar dari desain sirkuit modern. Pendekatan tersebut membentuk dasar dari Aljabar Boolean yang hingga saat ini digunakan secara luas dalam berbagai bidang ilmu pengetahuan. [3]

Aljabar Boolean didefinisikan secara formal sebagai sebuah struktur matematika dengan himpunan B yang dilengkapi dua operator biner, AND yang direpresentasikan dengan simbol $+$ dan OR yang direpresentasikan dengan simbol \cdot , satu operator uner, NOT yang direpresentasikan dengan simbol $'$, serta dua elemen khusus, 0 dan 1 yang berbeda dari B . Struktur tersebut dinyatakan dalam tupel $\langle B, +, \cdot, ', 0, 1 \rangle$ dan disebut Aljabar Boolean jika memenuhi beberapa aksioma berikut: [4]

1. Identitas
 - (i) $a + 0 = a$
 - (ii) $a \cdot 1 = a$
2. Komutatif

- (i) $a + b = b + a$
- (ii) $a \cdot b = b \cdot a$
- 3. Distributif
 - (i) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
 - (ii) $a + (b \cdot c) = (a + b) \cdot (a + c)$
- 4. Komplemen

Untuk setiap $a \in B$ terdapat elemen unik $a' \in B$ sehingga

 - (i) $a + a' = 1$
 - (ii) $a \cdot a' = 0$

E. Operasi Aljabar Boolean Dua Nilai

Operasi aljabar Boolean dua nilai melibatkan dua operator biner, yaitu penjumlahan '+' dan perkalian '.', serta satu operator uner, yaitu komplemen '. Operator ini memiliki aturan tertentu yang didefinisikan melalui tabel kebenaran: [4]

Tabel 1. Aturan operator biner '.'

a	b	a · b
0	0	0
0	1	0
1	0	0
1	1	1

Sumber: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf)

Tabel 2. Aturan operator biner '+'

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

Sumber: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf)

Tabel 3. Aturan operator biner "'"

a	a'
0	1
1	0

Sumber: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf)

F. Fungsi Boolean

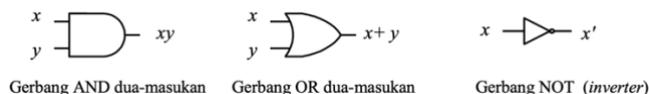
Fungsi Boolean adalah suatu fungsi matematis yang melibatkan variabel biner (nilai 0 dan 1) dan operator logika seperti penjumlahan (OR) '+', perkalian (AND) '.', dan komplemen. Nilai fungsi Boolean dapat ditentukan dengan mensubstitusi suatu nilai untuk setiap peubah. Fungsi Boolean dapat disajikan dalam dua bentuk standar (kanonik). Pertama, sebagai penjumlahan dari hasil kali (*sum of product* atau SOP). Kedua, sebagai perkalian dari hasil jumlah (*product of sum* atau POS).

Terdapat dua jenis suku dalam penyusunan fungsi Boolean, yaitu *minterm* dan *maxterm*. *Minterm* adalah suku dalam ekspresi SOP yang mencakup semua literal dalam bentuk hasil kali, sedangkan *maxterm* adalah suku dalam ekspresi POS yang mencakup semua literal dalam bentuk hasil jumlah. Setiap

peubah yang bernilai 0 dalam konteks *minterm* dinyatakan dalam bentuk komplemen, sedangkan yang bernilai 1 dinyatakan tanpa komplemen, berlaku sebaliknya dalam konteks *maxterm*. Dari tabel kebenaran, fungsi Boolean dapat dibangun dalam bentuk SOP dengan mengambil semua *minterm* untuk nilai fungsi yang bernilai 1, atau semua *maxterm* untuk nilai fungsi yang bernilai 0 dalam bentuk POS.

G. Rangkaian Logika

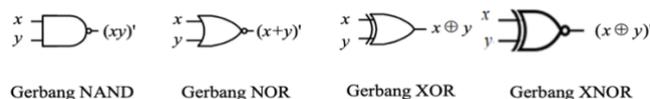
Fungsi Boolean dapat direpresentasikan dalam bentuk rangkaian logika, yang terdiri atas gerbang logika dasar dan gerbang logika turunan. Gerbang logika merupakan elemen utama dalam sistem digital yang digunakan untuk mengolah variabel biner, yaitu nilai 0 dan 1. Gerbang logika dasar meliputi gerbang AND, OR, dan NOT. Gerbang AND menghasilkan nilai 1 hanya jika semua input bernilai 1, sesuai dengan operasi logika perkalian ($x \cdot y$). Gerbang OR menghasilkan nilai 1 jika salah satu atau lebih input bernilai 1, sesuai dengan operasi penjumlahan logika ($x + y$). Sementara itu, gerbang NOT berfungsi untuk membalikkan nilai input, sehingga apabila x bernilai 1, maka x' akan bernilai 0, berlaku sebaliknya.



Gambar 5. Gerbang logika dasar

Sumber: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf)

Selain gerbang dasar, terdapat gerbang logika turunan yang disusun dari kombinasi gerbang dasar, yaitu gerbang NAND, NOR, XOR, dan XNOR. Gerbang NAND, atau NOT AND, menghasilkan kebalikan dari output gerbang AND ($(xy)'$), sementara gerbang NOR, atau NOT OR, menghasilkan kebalikan dari output gerbang OR ($(x + y)'$). Gerbang XOR, atau Exclusive OR, menghasilkan nilai 1 apabila jumlah *input* bernilai 1 adalah ganjil. Sebaliknya, gerbang XNOR, atau Exclusive NOR, memberikan nilai kebalikan dari XOR. [4]



Gambar 6. Gerbang logika turunan

Sumber: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf)

H. Penyederhanaan Fungsi Boolean dengan Metode Quine-McCluskey

Metode Quine-McCluskey adalah salah satu teknik yang digunakan untuk menyederhanakan fungsi Boolean secara sistematis. Metode ini sangat efektif untuk menyederhanakan fungsi Boolean dengan variabel dalam jumlah besar yang sulit divisualisasikan dengan peta Karnaugh. Metode ini dikembangkan oleh W. V. Quine dan E. J. McCluskey pada tahun 1950-an. Proses penyederhanaan ini melibatkan fungsi logika dalam bentuk *sum-of-products* (SOP) dengan langkah-

langkah sebagai berikut:

1. Representasi bit string
Setiap *minterm* dalam fungsi Boolean direpresentasikan sebagai string biner sepanjang jumlah variabel (n). Nilai “1” menunjukkan variabel dalam bentuk positif (x_i), sedangkan “0” menunjukkan variabel dalam bentuk komplemen (\bar{x}_i).
2. Pengelompokan berdasarkan jumlah bit ‘1’
String biner dikelompokkan berdasarkan jumlah bit “1” yang dimiliki. Misalnya, string dengan satu bit “1” dimasukkan ke dalam kelompok terpisah dari string dengan dua bit “1”. Pengelompokan ini bertujuan untuk mempermudah identifikasi kombinasi.
3. Kombinasi variabel
String dari kelompok yang berdekatan (berbeda di satu posisi bit) dikombinasikan untuk mengurangi jumlah variabel menjadi $n - 1$. Posisi bit yang berbeda digantikan dengan tanda “-”, misalnya 101 dan 111 menjadi 1-1. Kombinasi dilanjutkan hingga tidak ada lagi produk yang dapat dikombinasikan.
4. Pemilihan *essential prime implicants*
Prime implicants adalah produk yang tidak dapat dikombinasikan lebih lanjut. Setelah tidak ada produk yang dapat dikombinasikan, dilakukan pembentukan tabel untuk menunjukkan cakupan *prime implicants* terhadap *minterm*. *Prime implicants* yang hanya mencakup satu *minterm* disebut *essential* dan wajib dimasukkan dalam solusi akhir.
5. Penghapusan redundansi dan penyelesaian
Tabel disederhanakan dengan mengeliminasi *prime implicants* yang redundan. Proses ini dilakukan secara berulang hingga solusi minimal tercapai dan menghasilkan fungsi Boolean paling sederhana yang tetap mencakup semua *minterm*.

J. ADHD

Attention Deficit Hyperactivity Disorder (ADHD) merupakan gangguan perkembangan saraf yang umum terjadi pada anak usia dini dan dapat berlanjut hingga dewasa [1]. Gangguan ini ditandai dengan tiga karakteristik utama, yaitu inatentif, hiperaktivitas, dan impulsivitas. Perilaku inatentif mengacu pada kesulitan penderitanya untuk mempertahankan kefokusannya terhadap sesuatu dalam waktu yang lama, sehingga mereka cenderung mudah teralihkannya. Sementara itu, hiperaktivitas dan impulsivitas menggambarkan perilaku yang berlebihan, seperti ketidakmampuan untuk berdiam diri, serta kecenderungan untuk mengambil tindakan dengan tergesa-gesa tanpa mempertimbangkan konsekuensinya [2].

ADHD dapat memberikan dampak negatif yang signifikan terhadap kehidupan penderitanya. Gangguan ini tidak hanya menghambat kemampuan akademik, tetapi juga memengaruhi

interaksi sosial serta hubungan interpersonal [2]. Penderita ADHD memiliki risiko yang lebih tinggi untuk mengalami kegagalan akademik dan kesulitan dalam bersosialisasi di masyarakat [1]. Maka dari itu, pemahaman secara menyeluruh mengenai ADHD sangat penting untuk mendiagnosis gejalanya sejak usia dini serta mendukung proses penanganan yang tepat.

Menurut *Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition* (DSM-5), ADHD dikategorisasikan menjadi tiga tipe utama, yaitu tipe dominan inatentif, tipe dominan hiperaktif-impulsif, dan tipe gabungan. [3]

1. Tipe dominan inatentif

Tipe ini ditandai dengan gejala berupa kesulitan mempertahankan fokus dan perhatian pada tugas tertentu. Anak usia dini yang menderita ADHD dengan tipe ini mudah lupa terhadap aktivitas sehari-hari, atau sering kehilangan barang yang dibutuhkan untuk tugas atau kegiatan. Hal tersebut membuat anak kesulitan mengikuti instruksi atau menyelesaikan tugas yang memerlukan konsentrasi jangka panjang. Gejala lainnya yang sering adalah ketidakmampuan anak untuk memperhatikan detail suatu instruksi serta membuat kesalahan ceroboh. [2][3]

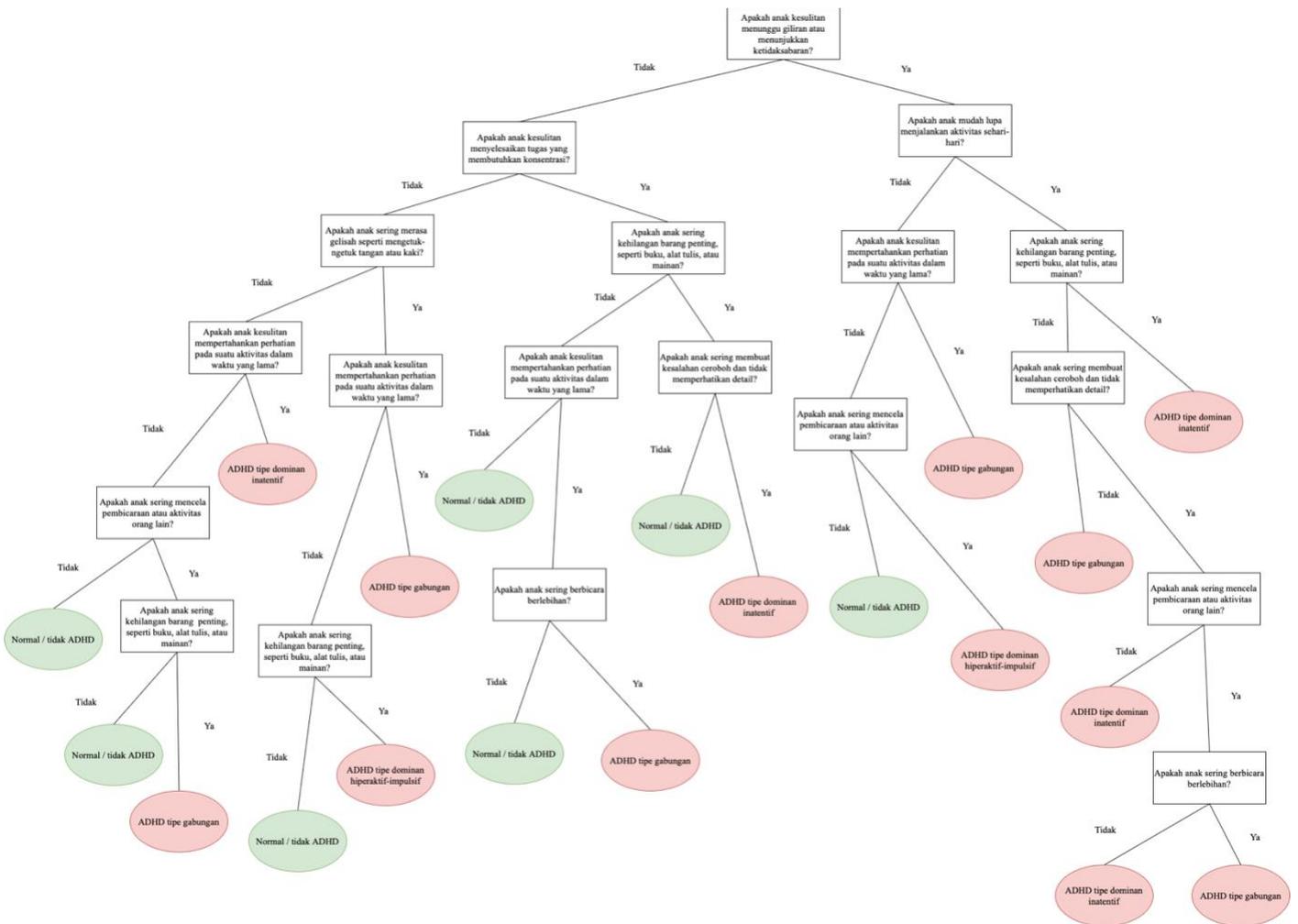
2. Tipe dominan hiperaktif-impulsif

Tipe ini ditandai dengan perilaku yang sangat aktif dan impulsif. Anak usia dini dengan tipe ini sulit untuk duduk diam, berbicara berlebihan, dan sering mengganggu aktivitas teman sebayanya. Perilaku ini dapat menyebabkan anak kesulitan dalam menjaga hubungan sosial serta menghambat proses belajar. Gejala umum pada tipe ini meliputi kegelisahan yang berlebihan, sulit menunggu giliran, serta sering mencela percakapan atau aktivitas orang lain. [3]

3. Tipe gabungan

Tipe ini merupakan kombinasi dari gejala inatentif dan hiperaktif-impulsif. Anak usia dini dengan tipe ini menunjukkan gejala dari kedua tipe secara bersamaan, seperti kesulitan memusatkan perhatian sekaligus perilaku hiperaktif yang berlebihan. Tipe gabungan adalah yang paling umum ditemukan pada individu dengan ADHD. [3]

Dengan memahami *Attention Deficit Hyperactivity Disorder* (ADHD) secara komprehensif, diagnosis dan penanganan ADHD dapat dilakukan lebih awal untuk mendukung tumbuh kembang anak usia dini. Setiap anak dengan ADHD memiliki gejala yang berbeda, bahkan untuk tipe yang sama. Misalnya, pada tipe inatentif, ada anak yang sering lupa detail kecil dan ada juga yang sulit menyelesaikan tugas. Berlaku juga pada tipe hiperaktif-impulsif, beberapa anak bisa sering mengalami kegelisahan, sementara yang lain lebih impulsif. Dengan pemahaman mengenai keragaman gejala ADHD, orang tua dan pendidik dapat memberikan dukungan yang tepat, sehingga anak-anak dengan ADHD dapat tumbuh optimal dan menjalani kehidupan yang lebih produktif.



Gambar 9. Pohon Keputusan Tipe ADHD
Sumber: Penulis

III. ANALISIS DAN PEMBAHASAN

A. Penerapan Pohon Keputusan

Pohon keputusan merupakan salah satu metode analisis yang efektif untuk mendeteksi gejala ADHD pada anak usia dini. Metode ini dirancang berdasarkan kajian literatur dari berbagai sumber, sehingga setiap simpul dan sisi didalamnya disusun berdasarkan data dan penelitian yang relevan. Gambar 9 memperlihatkan bagaimana pohon keputusan dapat digunakan untuk memetakan gejala ADHD ke dalam kategori diagnosis tertentu, yaitu normal (tidak ADHD), ADHD tipe inatentif, ADHD tipe hiperaktif-impulsif, atau ADHD tipe gabungan.

Pada pohon keputusan di atas, proses analisis mencakup berbagai gejala yang berkaitan dengan tipe-tipe ADHD yang bersumber dari buku yang diterbitkan American Psychiatric Association dan jurnal publikasi *Research on Child and Adolescent Psychopathology*. Pertanyaan mengenai gejala-gejala tersebut dipetakan secara sistematis dalam simpul pohon untuk mengarahkan pengguna menuju hasil diagnosis yang akurat berdasarkan jawaban “ya” atau “tidak” dari pengguna. Setiap simpul menggambarkan gejala yang mungkin dialami anak, sehingga pohon keputusan tidak hanya memberikan hasil yang akurat, tetapi juga mengarahkan pengguna pada langkah penanganan yang sesuai dengan kebutuhan anak.

Sebagai contoh, pada tipe dominan inatentif, seorang anak mungkin menunjukkan gejala seperti sering lupa terhadap detail tugas atau kesulitan menyelesaikan pekerjaan. Sementara itu, pada tipe hiperaktif-impulsif, anak cenderung kesulitan duduk diam atau berbicara tanpa berpikir terlebih dahulu. Apabila seorang anak menunjukkan kombinasi kedua jenis gejala tersebut, pohon akan mengarahkan pengguna pada daun berisi diagnosis ADHD tipe gabungan.

Anak dengan ADHD memiliki gejala yang beragam, bahkan pada tipe yang sama. Perbedaan ini menunjukkan pentingnya diagnosis yang terstruktur untuk memahami kebutuhan setiap anak. Pohon keputusan dirancang untuk menganalisis gejala ADHD secara terstruktur, sehingga memudahkan tenaga kesehatan, pendidik, maupun orang tua dalam memahami hasil diagnosis dan menentukan langkah penanganan yang sesuai.

B. Representasi dalam Aljabar Boolean

Gejala ADHD dapat direpresentasikan dalam bentuk variabel aljabar Boolean. Representasi tersebut bertujuan untuk mempermudah proses analisis dan penyederhanaan logika dalam pohon keputusan. Pendekatan ini mendukung penyederhanaan menggunakan metode Quine-McCluskey dalam aljabar Boolean, sehingga menghasilkan proses analisis yang lebih terstruktur dan efisien.

Tabel 4. Gejala ADHD beserta variabel Boolean

Sumber: Penulis

Gejala	Variabel Boolean
Kesulitan menunggu giliran atau menunjukkan ketidaksabaran	<i>a</i>
Kesulitan menyelesaikan tugas yang membutuhkan konsentrasi	<i>b</i>
Sering merasa gelisah seperti mengetuk-ngetuk tangan atau kaki	<i>c</i>
Kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama	<i>d</i>
Sering mencela pembicaraan atau aktivitas orang lain	<i>e</i>
Mudah lupa dalam menjalankan aktivitas sehari-hari	<i>f</i>
Sering kehilangan barang penting seperti buku, alat tulis, atau mainan	<i>g</i>
Sering membuat kesalahan ceroboh dan tidak memperhatikan detail	<i>h</i>
Sering berbicara berlebihan	<i>i</i>

C. Penyederhanaan dengan Metode Quine-McCluskey

Hasil diagnosis gejala ADHD dari pohon keputusan pada Gambar 9 dapat disederhanakan menggunakan Quine-McCluskey dalam aljabar Boolean. Untuk memudahkan proses penyederhanaan, hasil diagnosis ditinjau pada kategori normal atau tidak ADHD, dengan ekspresi Boolean dan deskripsi yang terlampir pada Tabel 5. Pendekatan ini membantu dalam mengurangi kompleksitas data yang dipetakan dalam pohon keputusan tanpa mengurangi tingkat akurasi dari hasil diagnosis.

Tabel 5. Ekspresi Boolean untuk gejala dengan kategori normal atau tidak ADHD

Sumber: Penulis

Ekspresi Boolean	Deskripsi
<i>a'b'c'd'e'</i>	1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak tidak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak tidak sering merasa gelisah seperti mengetuk-ngetuk tangan atau kaki. 4. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama. 5. Anak tidak sering mencela pembicaraan atau aktivitas orang lain.
<i>a'b'c'd'eg'</i>	1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak tidak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak tidak sering merasa gelisah seperti mengetuk-ngetuk tangan atau kaki.

	4. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama. 5. Anak sering mencela pembicaraan atau aktivitas orang lain. 6. Anak tidak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan.
<i>a'b'cd'g'</i>	1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak tidak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak sering merasa gelisah seperti mengetuk-ngetuk tangan atau kaki. 4. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama. 5. Anak tidak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan.
<i>a'bg'd'</i>	1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak tidak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan. 4. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama.
<i>a'bg'di'</i>	1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak tidak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan. 4. Anak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama. 5. Anak tidak sering berbicara berlebihan.
<i>a'bg'h'</i>	1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan. 4. Anak tidak sering membuat kesalahan ceroboh dan memperhatikan detail
<i>af'd'e'</i>	1. Anak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak tidak mudah lupa menjalankan aktivitas sehari-hari. 3. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama. 4. Anak tidak sering mencela pembicaraan atau aktivitas orang lain.

Dengan mengidentifikasi seluruh *minterm* dari tabel kebenaran fungsi Boolean yang memiliki *output* bernilai 1, proses penyederhanaan untuk gejala pada kategori normal atau tidak ADHD dilakukan dengan algoritma Quine-McCluskey dalam program berbahasa Python. Algoritma ini dimulai dengan mengelompokkan *minterm* berdasarkan jumlah bit 1 (*one's count*) dalam representasi binernya. Selanjutnya, pasangan *minterm* yang berbeda pada satu posisi bit digabungkan menjadi ekspresi yang lebih sederhana dengan menggunakan tanda '-' untuk menggantikan bit yang berbeda.

Hasil penggabungan ini menghasilkan Prime Implicants, yang kemudian dipetakan ke *minterm* dalam sebuah Prime Implicant Chart. Essential Prime Implicants kemudian diidentifikasi sebagai elemen utama yang mencakup seluruh *minterm*. Penyederhanaan dengan algoritma Quine-McCluskey menghasilkan ekspresi logika yang lebih sederhana dengan jumlah literal yang minimal. Hasil penyederhanaan dan kode program dapat diakses pada folder test di pranala Github berikut: <https://github.com/adndx/discrete-math.git>.

```
def main(n_var, minterms):
    a = minterms.split()
    a = list(map(int, a))

    group = [[] for x in range(n_var + 1)]

    for i in range(len(a)):
        a[i] = bin(a[i])[2:]
        if len(a[i]) < n_var:
            for j in range(n_var - len(a[i])):
                a[i] = '0' + a[i]
        elif len(a[i]) > n_var:
            print("\nError: Choose the correct number of variables(bits)\n")
            return
        index = a[i].count('1')
        group[index].append(a[i])

    all_group = []
    unchecked = []
    while verify_empty(group) == False:
        all_group.append(group)
        next_group, unchecked = pair_combination(group, unchecked)
        group = eliminate_redundant(next_group)

    s = "\nPrime Implicants :\n"
    for i in unchecked:
        s = s + change_to_letter(i) + " , "
    print(s[:len(s) - 3])

    Chart = [[0 for x in range(len(a))] for x in range(len(unchecked))]
    for i in range(len(a)):
        for j in range(len(unchecked)):
            if compare_binary_same(unchecked[j], a[i]):
                Chart[j][i] = 1

    primes = min_cost_calculation(Chart, unchecked)
    primes = eliminate_redundant(primes)
    print("\n-- Answers --\n")
    for prime in primes:
        s = ''
        for i in range(len(unchecked)):
            for j in prime:
                if j == i:
                    s = s + change_to_letter(unchecked[i]) + ' + '
        print(s[:len(s) - 3])

if __name__ == "__main__":
    try:
        with open("testfile.txt", "r") as file:
            content = file.readline().strip().split()
            n_var = int(content[0])
            minterms = " ".join(content[1:])
    except FileNotFoundError:
        print("File testfile.txt tidak ditemukan. Pastikan file ada di direktori yang benar.")
        exit()

    start_time = time.time()
    main(n_var, minterms)
    print("Time elapsed: {:.2f}ms".format((time.time() - start_time) * 1000))
```

Gambar 10. Fungsi main pada program penyederhanaan fungsi Boolean dengan algoritma Quine-McCluskey
Sumber: Penulis

```
adndx@Dindackermans-MacBook-Pro test % python3 quine-mccluskey.py
Prime Implicants :
a'b'c'd'e', b'c'd'e'f', a'c'd'e'h', c'd'e'f'h', bd'e'f'h', d'e'f'g',
a'bd'h', a'bg'i', a'bh'i', ad'e'f', a'bgh', a'd'g'

Essential Prime Implicants :
a'b'c'd'e', a'bg'i', ad'e'f', a'bgh', a'd'g'

-- Answers --

a'b'c'd'e' + a'bg'i' + ad'e'f' + a'bgh' + a'd'g'
Time elapsed: 51.12ms
```

Gambar 11. Hasil penyederhanaan fungsi Boolean dengan algoritma Quine-McCluskey
Sumber: Penulis

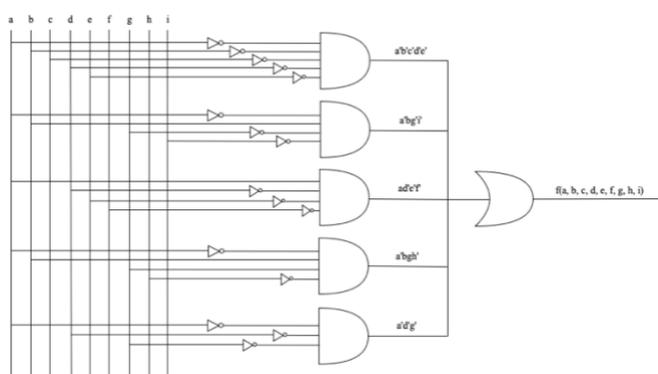
Tabel 6. Hasil penyederhanaan ekspresi Boolean untuk gejala dengan kategori normal atau tidak ADHD menggunakan algoritma Quine-McCluskey
Sumber: Penulis

Ekspresi Boolean	Deskripsi
a'b'c'd'e'	<ol style="list-style-type: none"> 1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak tidak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak tidak sering merasa gelisah seperti mengetuk-ngetuk tangan atau kaki. 4. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama. 5. Anak tidak sering mencela pembicaraan atau aktivitas orang lain.
a'bg'i'	<ol style="list-style-type: none"> 1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak tidak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan. 4. Anak tidak sering berbicara berlebihan.
ad'e'f'	<ol style="list-style-type: none"> 1. Anak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak tidak mudah lupa menjalankan aktivitas sehari-hari. 3. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama. 4. Anak tidak sering mencela pembicaraan atau aktivitas orang lain.
a'bgh'	<ol style="list-style-type: none"> 1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak kesulitan menyelesaikan tugas yang membutuhkan konsentrasi. 3. Anak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan. 4. Anak tidak sering membuat kesalahan ceroboh dan memperhatikan detail
a'd'g'	<ol style="list-style-type: none"> 1. Anak tidak kesulitan menunggu giliran atau menunjukkan ketidaksabaran. 2. Anak tidak sering kehilangan barang penting, seperti buku, alat tulis, atau mainan. 3. Anak tidak kesulitan mempertahankan perhatian pada suatu aktivitas dalam waktu yang lama.

Ekspresi Boolean hasil penyederhanaan yang terdapat dalam Tabel 6 dapat direpresentasikan melalui fungsi Boolean, yaitu:

$$f(a, b, c, d, e, f, g, h, i) = a'b'c'd'e' + a'bg'i' + ad'e'f' + a'bg'h' + a'd'g'$$

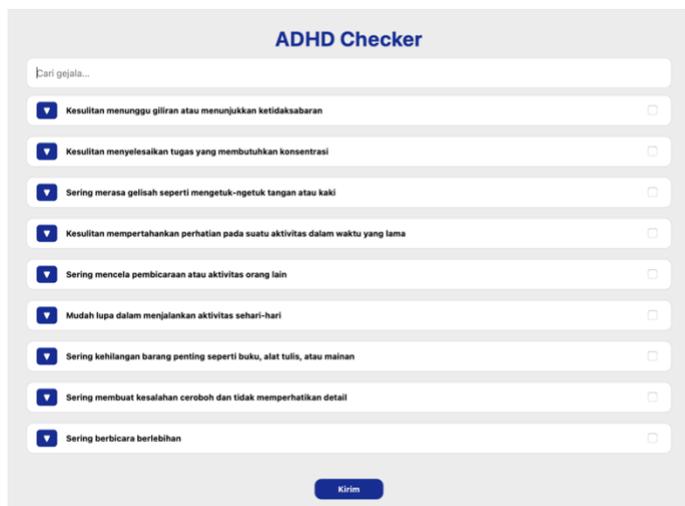
Ekspresi Boolean yang telah disederhanakan dapat direpresentasikan dalam bentuk diagram gerbang logika. Visualisasi ini menunjukkan bagaimana setiap variabel saling terhubung dan bekerja sesuai dengan fungsi logikanya, sehingga memudahkan penerapan dalam perancangan sistem eksperimen program.



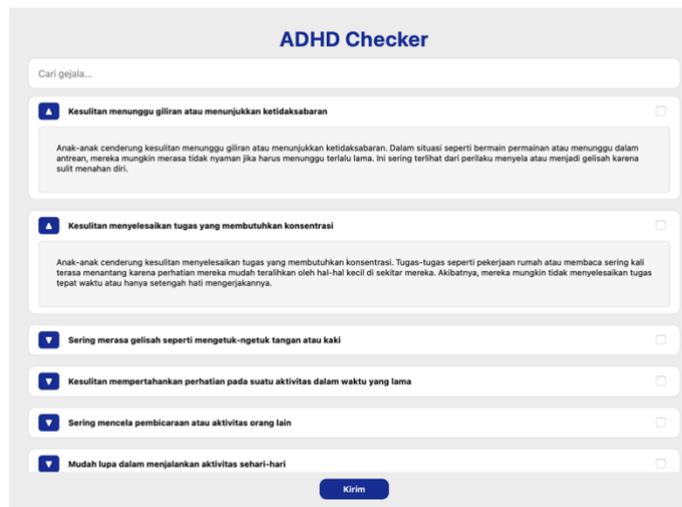
Gambar 12. Diagram gerbang logika untuk gejala dengan kategori normal atau tidak ADHD setelah penyederhanaan dengan algoritma Quine-McCluskey
Sumber: Penulis

IV. EKSPERIMEN PROGRAM

Sebagai bentuk penerapan konsep pohon keputusan yang disederhanakan menggunakan algoritma Quine-McCluskey, saya mengembangkan sebuah aplikasi sederhana dalam bahasa Python yang bernama ADHD Checker. Aplikasi ini dirancang untuk membantu proses awal diagnosis ADHD pada anak usia dini. Melalui aplikasi ini, pengguna dapat mengevaluasi gejala-gejala ADHD yang timbul dan mendapatkan gambaran awal terkait potensi gangguan tersebut.

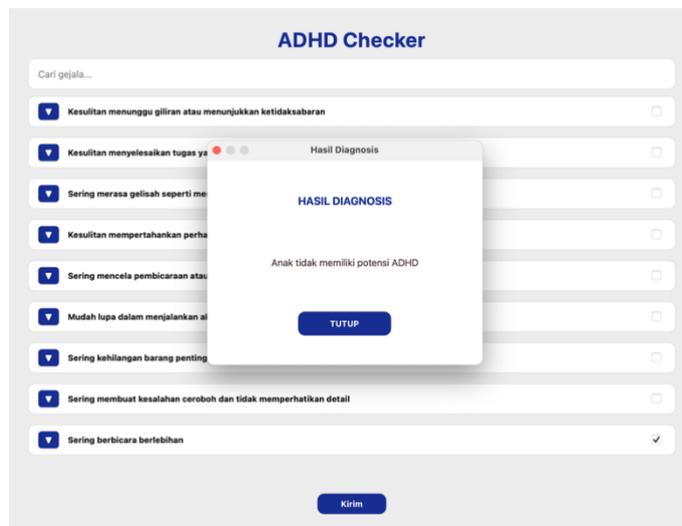


Gambar 13. Tampilan awal aplikasi
Sumber: Penulis

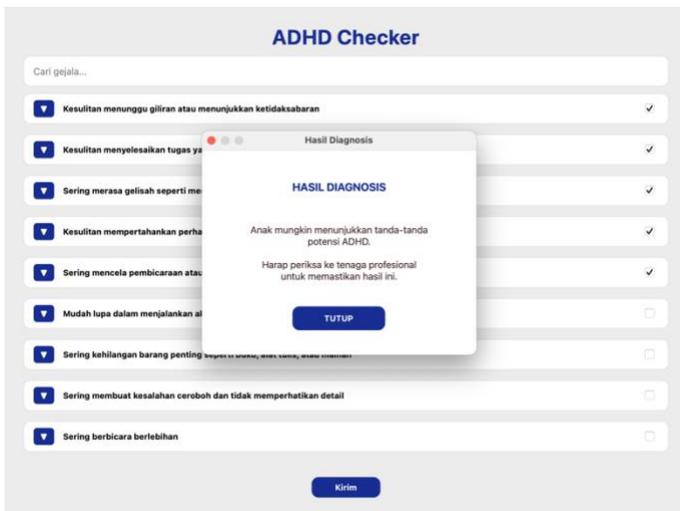


Gambar 14. Tampilan deskripsi gejala ADHD
Sumber: Penulis

Aplikasi ADHD Checker hadir dengan GUI berbasis PyQt, sehingga mudah untuk digunakan oleh orang tua atau pendidik. Gejala ADHD ditampilkan dalam bentuk daftar checkbox yang dapat dicentang sesuai kondisi yang diamati. Setelah pengguna memilih gejala yang relevan dan menekan tombol “Kirim,” aplikasi akan memproses data tersebut berdasarkan logika yang telah disederhanakan menggunakan metode Quine-McCluskey. Hasil diagnosis ditampilkan dalam bentuk pop-up dialog dengan dua kemungkinan: “Anak tidak menderita ADHD” atau “Anak mungkin menunjukkan tanda-tanda potensi ADHD.” Kode program aplikasi dapat diakses melalui pranala Github berikut: <https://github.com/adndax/discrete-math.git>.



Gambar 15. Tampilan ketika anak tidak memiliki potensi ADHD
Sumber: Penulis



Gambar 16. Tampilan ketika anak memiliki potensi ADHD
Sumber: Penulis

V. KESIMPULAN

Dari makalah ini, dapat disimpulkan bahwa penerapan pohon keputusan dan aljabar Boolean mampu memberikan kontribusi yang signifikan dalam mendeteksi gejala ADHD pada anak usia dini. Setiap simpul dalam pohon keputusan merepresentasikan pertanyaan yang berkaitan dengan gejala ADHD, dan setiap jalur membawa pengguna menuju hasil diagnosis ADHD.

Dengan menyederhanakan hasil analisis menggunakan metode Quine-McCluskey dalam konsep aljabar Boolean, proses pengambilan keputusan dapat dirancang secara lebih efisien dan sistematis. Representasi numerik berupa angka 0 untuk penderita ADHD dan angka 1 untuk individu normal membuat interpretasi data yang lebih terstruktur.

Makalah ini menunjukkan bahwa ilmu matematika diskrit, termasuk pohon keputusan dan aljabar Boolean, dapat diterapkan dalam memecahkan masalah kehidupan sehari-hari. Pendekatan ini diharapkan dapat memberikan pemahaman yang lebih mendalam terhadap pola gejala dan faktor pendukung ADHD, sehingga dapat membantu pendidik dan orang tua dalam mendeteksi dini serta memberikan penanganan yang tepat pada anak usia dini penderita ADHD.

VI. UCAPAN TERIMA KASIH

Sebagai penulis makalah ini, saya ingin mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga makalah yang berjudul *“Implementasi Pohon Keputusan dan Metode Quine-McCluskey dalam Deteksi Gejala ADHD pada Anak Usia Dini”* dapat diselesaikan dengan baik. Saya juga ingin menyampaikan rasa terima kasih kepada berbagai pihak yang telah mendukung dan membantu selama proses penulisan makalah ini, khususnya kepada:

1. Bapak Dr. Rinaldi Munir, Bapak Dr. Rila Mandala, serta Bapak Arrival Dwi Sentosa, M.T., selaku dosen pengajar mata kuliah IF2120 Matematika Diskrit, atas bimbingan, pengajaran, dan ilmu yang telah diberikan selama perkuliahan dan menjadi landasan penting dalam penyusunan makalah ini.
2. Kedua orang tua saya yang selalu memberikan doa restu

dan dukungan moral sehingga saya dapat menyelesaikan makalah ini dengan baik.

3. Pacar saya, Fajar Asyraf R., dan teman-teman saya yang telah memberikan dorongan, motivasi, serta masukan yang mendukung kelancaran penyusunan makalah ini.
4. Para penulis jurnal, artikel, dan referensi lainnya yang menjadi sumber inspirasi dan memberikan kontribusi penting melalui wawasan yang relevan untuk penulisan makalah ini.

Saya menyadari bahwa makalah ini tidak akan terwujud tanpa kontribusi dan dukungan dari seluruh pihak yang telah disebutkan di atas. Untuk itu, saya mengucapkan terima kasih yang sebesar-besarnya atas segala bantuan, inspirasi, dan ilmu yang telah diberikan. Saya berharap makalah ini dapat memberikan manfaat bagi para pembaca serta mendukung pengembangan ilmu pengetahuan, khususnya di bidang teknologi.

REFERENCES

- [1] V. A. Harpin, “The effect of ADHD on the life of an individual, their family, and community from preschool to adult life,” *Archives of Disease in Childhood*, vol. 90, Suppl. 1, pp. i2–i7, 2005.
- [2] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition (DSM-5)*. 2013.
- [3] K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th ed. New York, NY, USA: McGraw-Hill, 2012.
- [4] Munir, Rinaldi, “Aljabar Boolean (Bag. 1),” [Online]. Tersedia: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-\(2024\)-bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/12-Aljabar-Boolean-(2024)-bagian1.pdf). [Diakses: 24 Desember 2024].
- [5] Munir, Rinaldi, “Pohon (Bag. 1),” [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf>. [Diakses: 24 Desember 2024].
- [6] Munir, Rinaldi, “Pohon (Bag. 2),” [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/24-Pohon-Bag2-2024.pdf>. [Diakses: 24 Desember 2024].

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, 28 Desember 2024

Adinda Putri
13523071