

Rancang Bangun Aplikasi Sistem Pakar Diagnosis Gangguan Emosional Pada Anak Berbasis Aplikasi Website

Fanidia Nur Utami, Kodrat Iman Satoto, Kurniawan Teguh Martono
Program Studi Sistem Komputer Fakultas Teknik Universitas Diponegoro
Jalan Prof. Sudharto, Tembalang, Semarang, Indonesia
fanidianu@outlook.com

Abstrak – *Ketepatan diagnosis terhadap suatu penyakit dan kecepatan proses penentuan diagnosis sangat penting dalam dunia kesehatan. Proses diagnosis penyakit ditentukan dari pengetahuan tenaga medis yang menangani pasien. Kesalahan diagnosis penyakit dan keterlambatan menentukan solusi pengobatan dapat memberikan dampak buruk bagi kesehatan pasien. Namun, tenaga medis yang dapat melayani terkadang jumlahnya terbatas. Penelitian ini dikhususkan untuk merancang suatu aplikasi sistem pakar diagnosis gangguan emosional yang disesuaikan dengan Pedoman Panduan Diagnosis Gangguan Jiwa di Indonesia.*

Perancangan sistem pakar pada penelitian ini bertujuan untuk menerapkan ilmu kecerdasan buatan, metodologi waterfall dan aplikasi website. Perencanaan sistem menggunakan metode terstruktur dengan mendefinisikan DFD, ERD, serta diagram alir sistem. Sistem pakar dirancang berbasis aplikasi website sehingga program ditulis dengan bahasa pemrograman C# dan Razor untuk bagian view sedangkan basisdata program menggunakan SQL Server Local Database

Proses diagnosis penyakit pada penelitian ini menggunakan metode forward chaining. Pengujian aplikasi sistem pakar ini menggunakan metode pengujian white box. Berdasarkan hasil pengujian white box, tidak ditemukan kesalahan fungsi dan proses diagnosis pada aplikasi.

Kata Kunci : gangguan emosional, aplikasi website, sistem pakar, *forward chaining*, pengujian *WhiteBox*

I. PENDAHULUAN

PADA saat ini terjadi perkembangan yang pesat dalam bidang ilmu pengetahuan dan teknologi, khususnya teknologi komputer dan komunikasi atau sering disebut dengan era Information and Communication Technology (ICT). Jika pada mulanya komputer digunakan hanya sekedar alat penghitung, maka saat ini komputer telah mampu menggantikan peran atau tugas-tugas rumit yang dilakukan oleh manusia, bahkan sanggup menirukan proses biologis manusia dalam pengambilan keputusan yang disebut kecerdasan buatan.

Perkembangan teknologi yang menghasilkan produk dengan mengimplementasikan ilmu kecerdasan buatan dapat melalui pengembangan sebuah aplikasi untuk dapat menghasilkan suatu keputusan. Salah satunya adalah

penggunaan konsep sistem pakar. Sistem ini dirancang untuk menirukan keahlian seorang pakar dalam menjawab pertanyaan dan menyelesaikan suatu permasalahan baik di bidang kesehatan atau kedokteran, bisnis, ekonomi dan sebagainya.

Pengimplementasian sistem pakar pada dunia kedokteran atau kesehatan dapat berupa diagnosis penyakit, konsultasi penjagaan kesehatan hingga pemberian saran penentuan solusi dari hasil diagnosis yang ada dengan cepat dan tepat. Rancang bangun sistem pakar diagnosis gangguan emosional pada anak ini memiliki tujuan untuk merancang suatu aplikasi yang dapat melakukan diagnosis gangguan emosional pada anak dengan menerapkan salah satu teknik kecerdasan buatan yaitu teknik *reasoning* dengan menggunakan runut maju (*forward chaining*) dan teknik representasi pengetahuan berupa kaidah produksi.

Dalam pembuatan tugas akhir ini pembahasan masalah memiliki batasan pada permasalahan berikut :

- Aplikasi sistem pakar diagnosis Gangguan emosi (*emotional disorder*) dirancang dan dibuat berbasis aplikasi website dengan menggunakan *framework* ASP.NET MVC bahasa pemrograman C#.
- Fitur aplikasi sistem pakar diagnosis Gangguan emosional (*emotional disorder*) pada anak meliputi keterangan gejala-gejala, diagnosis dan cara penanganan dari gangguan emosional tersebut.
- Fitur tambahan yang ditawarkan dalam aplikasi sistem pakar diagnosis gangguan emosional (*emotional disorder*) pada anak berbasis website adalah tanya pakar.
- Aplikasi hanya dapat menyimpan 10 jenis gangguan emosional dan gejala sebanyak 40 buah.
- Implementasi aplikasi sistem pakar diagnosis gangguan emosional pada anak adalah sebatas hanya pada *host* lokal dan menggunakan basisdata sql server lokal.
- Lingkup kelainan psikologi hanya meliputi gangguan emosional (*emotional disorder*) pada anak maupun remaja yang terbagi menjadi dua, yaitu: *mood disorders* (gangguan suasana hati) dan *anxiety disorders* (gangguan kegelisahan) mengacu pada referensi buku PPDGJ-III dan berdasarkan pengetahuan dari seorang pakar (psikiater dan psikolog).

II. LANDASAN TEORI

A. Penelitian Terdahulu

Dalam melakukan penelitian mengenai rancang bangun aplikasi sistem pakar diagnosis gangguan emosional berbasis

website mengambil beberapa referensi dari jurnal terkait sistem pakar dan psikologis anak. Pada penelitian sebelumnya, beberapa point yang dapat disimpulkan diantaranya :

Pada jurnal “Rancang Bangun Aplikasi Sistem Pakar Untuk Menentukan Jenis Gangguan Perkembangan Pada Anak”, sistem sudah mampu menganalisis jenis gangguan perkembangan yang dialami pasien berdasarkan gejala-gejala yang dimasukkan oleh *user*. Aplikasi mampu menyimpan representasi pengetahuan pakar berdasarkan nilai kebenaran MB dan nilai ketidakbenaran MD. Aplikasi sistem pakar ini sudah dapat menjelaskan definisi jenis gangguan perkembangan, penyebab dan pengobatannya.^[1]

Perbedaannya dengan penelitian yang akan dilakukan oleh penulis adalah aplikasi yang dibangun berupa aplikasi website yang dibuat menggunakan ASP.NET MVC dan bahasa pemrograman C#. Untuk metode yang digunakan adalah sama-sama runut maju, hanya saja pada penelitian kali ini penulis tidak memperitungkan nilai kebenaran dan nilai ketidakbenaran. Teknik representasi yang dipilih dalam penelitian ini adalah kaidah produksi.

B. Kecerdasan Buatan

Kecerdasan buatan (*artificial intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan lebih baik daripada yang dilakukan manusia.^[2]

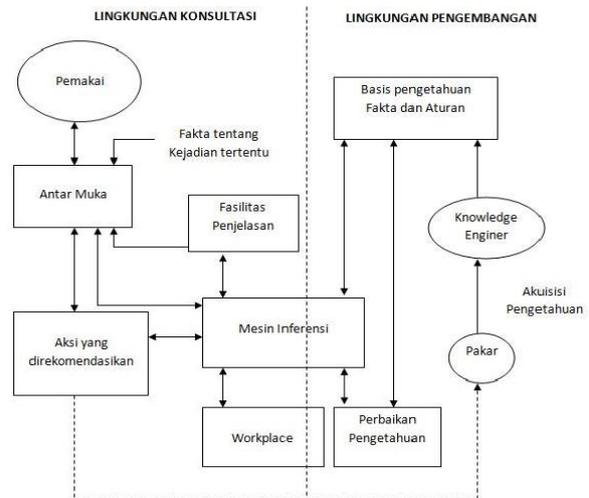
C. Sistem Pakar

Sistem pakar merupakan sebuah sistem yang menggunakan pengetahuan manusia di mana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia.

Sistem pakar dikembangkan dalam berbagai bidang termasuk dalam bidang medis. Saat ini kebutuhan manusia akan pelayanan medis yang lebih baik sangat mendesak, yang berarti dukungan instrumentasi dan informatika medis modern (telemedis) menjadi sangat dibutuhkan termasuk metode untuk membantu analisisnya sehingga dihasilkan diagnosis yang lebih optimal.^[3]

Sistem pakar memiliki ciri-ciri sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak pasti.
3. Dapat menyesuaikan rangkaian alasan-alasan yang diberikannya dengan cara yang dapat dipahami.
4. Dirancang untuk dapat dikembangkan secara bertahap.
5. Keluarannya bersifat anjuran



Gambar 1. Arsitektur Sistem Pakar^[4]

Beberapa istilah yang terdapat dalam sistem pakar diantaranya: mesin inferensi dan teknik representasi pengetahuan. Mesin inferensi merupakan program computer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan. Ada 2 (dua) metode inferensi yang umum dalam sistem pakar yaitu:

1. Forward Chaining (Runut Maju)

Forward chaining adalah pendekatan yang dimotori data (*data-driven*). Dalam pendekatan ini, pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan. *Forward chaining* mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN.^[5]

Kelebihan *forward chaining* adalah data baru dapat dimasukkan ke dalam basisdata inferensi yang membuat sistem menjadi lebih dinamis karena mengikuti perubahan fakta yang mendukung kesimpulan.

2. Backward Chaining (Runut Mundur)

Backward chaining adalah pendekatan yang dimotori tujuan (*goal driven*). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Lalu, proses pelacakan menggunakan premis aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya.^[5]

Teknik representasi pengetahuan merupakan suatu teknik untuk merepresentasikan basis pengetahuan yang diperoleh ke dalam suatu diagram tertentu sehingga dapat diketahui relasi/keterhubungan antar suatu data dengan data yang lain. Salah satu teknik yang digunakan dalam penelitian ini adalah kaidah produksi, yaitu dengan merepresentasikan pengetahuan dalam bentuk IF-THEN menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya.

Dalam model representasi pengetahuan kaidah produksi, terdapat beberapa tahapan yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah-langkah tersebut adalah menyajikan pengetahuan yang berhasil didapatkan dalam bentuk table keputusan (*decision table*) kemudian dari table keputusan dibuat pohon keputusan (*tree decision*).^[5]

D. Microsoft Visual Studio

Microsoft visual studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi windows ataupun aplikasi website. Dalam penelitian ini Microsoft Visual Studio yang digunakan adalah Microsoft Visual Studio Express for Web^[6].

E. ASP.NET MVC

ASP.NET MVC merupakan suatu *framework* untuk pengembangan website pada platform Microsoft .NET yang menyediakan jalan bagi para pengembang untuk membangun aplikasi website dengan struktur yang baik.^[7] ASP.NET MVC secara signifikan telah berkembang dan populer sejak pertama kali dipublikasikan yaitu pada tahun 2007. MVC merupakan kependekan dari *model-view-controller*^[8].

F. Entity Framework

Entity framework merupakan sekumpulan teknologi ADO.NET yang membantu memetakan antara pengembangan *object-oriented* dan basis data^[9]. Terdapat tiga cara untuk bekerja dengan entity framework, diantaranya adalah: *Database First*, *Model First* dan *Code First*.

G. SQL Server Database

SQL Server merupakan sebuah *host* dari runtime .NET . Beberapa versi sebelum CLR 2.0 sudah ada untuk menjalankan aplikasi .NET, misalnya *host* untuk ASP.NET. Setiap basis data yang menggunakan kode CLR menciptakan domain aplikasi sendiri. Hal ini menjamin bahwa CLR kode dari satu database tidak memiliki inflence pada database lain. Faktor yang paling penting dalam melakukan pemrograman SQL Server adalah tingkat keamanan yang memiliki level tinggi.^[11]

H. Bahasa Pemrograman C#

Bahasa pemrograman C# dirancang oleh Microsoft Corp. sebagai bahasa pemrograman yang berdaya-guna, aman serta mudah digunakan. Sebagai bagian dari platform .NET, bahasa pemrograman C# dirancang untuk bekerja dengan baik diatas framework .NET, yang mampu digunakan untuk menulis perangkat lunak yang handal demi layanan yang cepat^[11]. Bahasa pemrograman C# juga dapat digunakan untuk mengembangkan aplikasi-aplikasi sarana bergerak (*mobile application*), aplikasi berbasis website (*Web-based application*) serta aplikasi berskala besar (*Enterprise*)^[12]

I. Razor

Razor merupakan salah satu dari *view engine* yang menggunakan pendekatan dengan focus pada *template code*^[13] Razor memungkinkan pengguna untuk memulai dengan tag HTML statis (atau konten teks) dan kemudian membuatnya dinamis dengan menambahkan kode server didalamnya. Salah satu tujuan inti di balik Razor adalah untuk membuat proses coding menjadi mudah, dan dapat memungkinkan anda dengan

cepat mengintegrasikan kode server ke markup HTML dengan meminimalkan penekanan tombol pada keyboard.

J. Model Pengembangan Perangkat Lunak Waterfall

Pengembangan perangkat lunak merupakan sebuah pembelajaran iteratif atau berulang dan produk yang dihasilkan merupakan perwujudan dari pengetahuan yang dikumpulkan, disaring dan ditata selama proses berlangsung. Indikator keberhasilan perangkat lunak adalah kualitas, ketepatan waktu dan keberlangsungan jangka panjang dari sebuah produk perangkat lunak yang dihasilkan. Salah satu model pengembangan perangkat lunak adalah model *waterfall*.^[14]

Waterfall atau air terjun adalah model yang dikembangkan untuk pengembangan perangkat lunak, membuat perangkat lunak. model berkembang secara sistematis dari satu tahap ke tahap lain dalam mode seperti air terjun. Model ini mengusulkan sebuah pendekatan kepada pengembangan software yang sistematis dan sekuensial yang mulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Model ini melingkupi aktivitas-aktivitas sebagai berikut : rekayasa dan pemodelan sistem informasi, analisis kebutuhan, desain, penulisan kode program, pengujian dan pemeliharaan. Model pengembangan ini bersifat linear dari tahap awal pengembangan system yaitu tahap perencanaan sampai tahap akhir pengembangan system yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya.

K. Psikologi Abnormal Anak

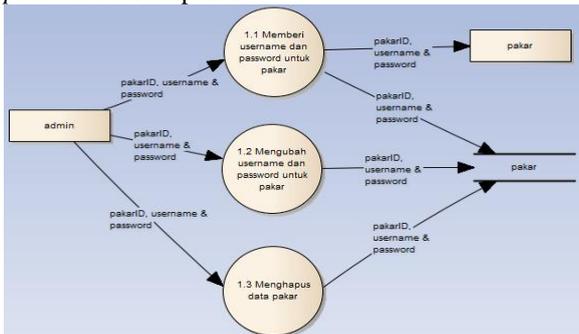
Studi mengenai perilaku abnormal sering membuat seseorang lebih sensitif dan waspada terhadap cara yang digunakan untuk menggambarkan perilaku orang lain. Gangguan psikologis didefinisikan sebagai pola perilaku, kognitif, emosional, atau gejala fisik yang ditunjukkan oleh seorang individu.^[15]

Pengelompokkan gangguan emosional terbagi menjadi dua kelompok yaitu gangguan rasa cemas (*anxiety disorder*) dan gangguan suasana hati (*mood disorder*). Anak-anak yang mengalami rasa cemas yang berlebihan ataupun justru tidak memiliki rasa gelisah dapat dikatakan mengalami gangguan rasa cemas (*anxiety disorder*).^[16]

Gangguan suasana hati adalah suatu kondisi dimana seorang anak atau seorang remaja yang secara konstan dapat menjadi sangat tidak senang, menunjukkan sedikit rasa antusias, suasana hati yang berubah sangat cepat atau bahkan menjadi sangat buruk dan memiliki pemikiran bahwa kehidupan bukan hanya bernilai sekedar kehidupan. Anak-anak yang mengalami *mood disorders* (gangguan suasana hati) menderita kondisi emosionalnya yang dapat berubah secara tiba-tiba dari kondisi emosi yang buruk, stabil hingga dalam keadaan paling ekstrem. Contohnya kondisi perasaan yang tidak senang secara berlebihan atau suasana hati dapat dengan mudah berubah dari kesedihan yang dalam ke kegembiraan yang tinggi

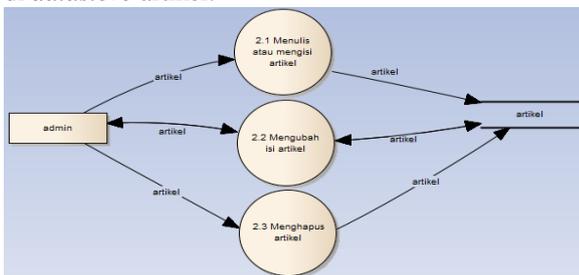
Gambar 3 menunjukkan DFD Level 0 yang menggambarkan proses apa saja yang terdapat dalam sistem pakar. Dalam gambar tersebut, terdapat sembilan proses utama yang menghubungkan entitas maupun proses ke penyimpanan data. Sembilan proses tersebut diantaranya adalah:

1. Proses 1: mengelola data pakar. Beberapa proses yang dapat dilakukan dalam mengelola data pakar diantaranya adalah memberi *username* dan *password* untuk pakar, menghapus data pakar dan mengubah *username* dan *password* untuk pakar.



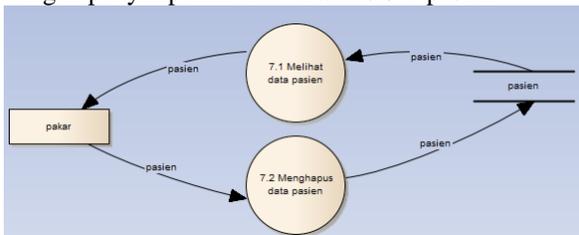
Gambar 4. DFD Level 1 Proses 1 (Mengelola Data Pakar)

2. Proses 2: mengelola data artikel. Beberapa proses yang dapat dilakukan diantaranya adalah mengisi atau menulis artikel, menghapus artikel dan mengubah isi artikel. Entitas yang terlibat hanya admin saja dengan penyimpanan data di *datastore* artikel.



Gambar 5. DFD Level 1 Proses 2

3. Proses 3: mengelola data pertanyaan. Beberapa proses yang dapat dilakukan dalam mengelola data pertanyaan adalah melihat daftar pertanyaan yang diberikan oleh pengguna untuk pakar, menjawab pertanyaan serta menghapus pertanyaan.
4. Proses 4: mengelola data pasien. Beberapa proses yang dapat dilakukan dalam mengelola data pasien diantaranya adalah melihat data pasien dan menghapus data pasien. Entitas yang terlibat diantaranya adalah pakar dan pasien dengan penyimpanan data di *datastore* pasien.



Gambar 6. DFD Level 1 Proses 4

Setelah melakukan perancangan alur kerja sistem, tahapan selanjutnya adalah melakukan perancangan basisdata. Dalam melakukan perancangan basisdata, beberapa tahapan yang dilakukan adalah menentukan entitas apa saja yang akan terlibat. Entitas yang terlibat dan dibuat sendiri dalam penelitian ini sebanyak delapan buah, diantaranya: user, pakar, pasien, gejala, diagnosis, gangguan, artikel dan tanya pakar. Setelah menentukan entitas apa saja yang terlibat, tahapan selanjutnya adalah menentukan atribut dari tiap entitas tersebut. Tabel 1 hingga tabel 8 adalah perincian dari atribut pada tiap-tiap entitas.

Tabel 1. Tabel User

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	userID	VARCHAR(3)	Primary Key	Kolom ID untuk admin
2.	Email	VARCHAR(30)	-	Kolom untuk email admin
3.	Username	VARCHAR(20)	-	Kolom untuk username admin
4.	Password	VARCHAR(32)	-	Kolom untuk password admin

Tabel 2. Tabel Pakar

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	pakarID	INT(3)	Primary Key	Kolom ID untuk pakar
2.	nama_pakar	VARCHAR(30)	-	Kolom untuk nama pakar
3.	tgl_lahir	DATE	-	Kolom untuk tanggal lahir pakar
4.	alamat_pakar	VARCHAR(100)	-	Kolom untuk alamat pakar
5.	email	VARCHAR(50)	-	Kolom untuk email pakar
6.	no_hp	VARCHAR(15)	-	Kolom untuk no hp pakar
7.	alamat_kantor	VARCHAR(100)	-	Kolom untuk alamat kantor
8.	notes	VARCHAR(500)	-	Kolom untuk username pakar
9.	foto	VARBINARY	-	Kolom untuk password pakar
10.	User	VARCHAR(3)	Navigation Property	Menentukan relasi ke tabel user

Tabel 3. Tabel Pasien

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	pasienID	INT	Primary Key	Kolom ID untuk pasien
2.	nama_anak	VARCHAR(25)	-	Kolom untuk nama anak
3.	Usia	INT	-	Kolom untuk usia anak
4.	Jenkel	VARCHAR(10)	-	Kolom untuk jenis kelamin anak
5.	gol_darah	VARCHAR(2)	-	Kolom untuk golongan darah anak
6.	alamat	VARCHAR(50)	-	Kolom untuk alamat anak
7.	nama_ayah	VARCHAR(25)	-	Kolom untuk nama ayah anak
8.	nama_ibu	VARCHAR(25)	-	Kolom untuk nama ibu anak
9.	email_ortu	VARCHAR(25)	-	Kolom untuk alamat email orang tua
10.	waktuTes	DateTime	-	Kolom untuk waktu pengguna melakukan tes
11.	Gejala	VARCHAR(128)	Navigation Property	Kolom untuk menunjukkan relasi dengan tabel gejala
12.	diagnosisID	VARCHAR(3)	Navigation Property	Kolom untuk menunjukkan relasi dengan tabel diagnosis

Tabel 4. Tabel Gejala

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	gejalaID	VARCHAR(5)	Primary Key	Kolom ID untuk gejala
2.	nama_gejala	VARCHAR(100)	-	Kolom untuk nama gejala
3.	pertanyaan_sistem	VARCHAR(500)	-	Kolom untuk pertanyaan sistem
4.	gejala_gejalaID	VARCHAR(128)	Navigation Properties	Kolom untuk menunjukkan relasi ke tabel gejala
5.	gejala_gejalaID1	VARCHAR(128)	Navigation Properties	Kolom untuk menunjukkan relasi ke tabel gejala
6.	diagnosisSakit_gangguanID	VARCHAR(128)	Navigation Properties	Kolom untuk menunjukkan relasi ke tabel diagnosis

Tabel 5. Tabel Diagnosis

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	diagnosisID	VARCHAR(3)	Primary Key	Kolom ID untuk diagnosis
2.	gangguanID	VARCHAR(3)	-	Kolom ID untuk gangguan
3.	gejalaID	VARCHAR(1000)	-	Kolom ID untuk gejala

Tabel 6. Tabel Gangguan

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	gangguanID	VARCHAR(3)	Primary Key	Kolom ID untuk admin
2.	nama_gangguan	VARCHAR(50)	-	Kolom untuk nama gangguan
3.	Definisi	VARCHAR(300)	-	Kolom untuk definisi gangguan
4.	Penanganan	VARCHAR(2000)	-	Kolom untuk penanganan gangguan

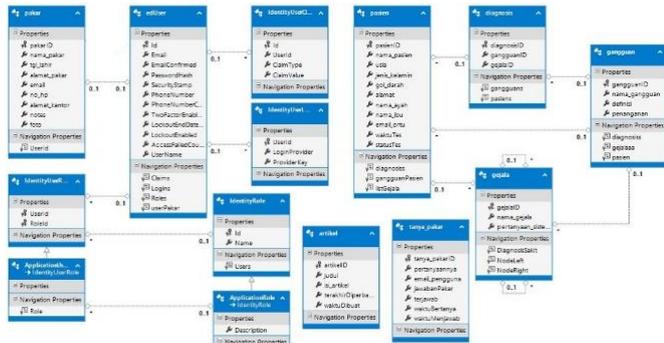
Tabel 7. Tabel Artikel

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	artikelID	INT	Primary Key	Kolom ID untuk artikel
2.	Judul	VARCHAR(80)	-	Kolom untuk judul artikel
3.	isi_artikel	VARCHAR(2000)	-	Kolom untuk isi artikel
4.	terakhirDiperbaharui	DateTime	-	Kolom untuk waktu kapan terakhir kali artikel diperbaharui
5.	waktuDibuat	DateTime	-	Kolom untuk waktu kapan artikel pertama kali ditulis admin

Tabel 8. Tabel tanya_pakar

No	Kolom	Tipe Data	Keterangan	Deskripsi
1.	pertanyaanID	INT	Primary Key	Kolom ID untuk pertanyaan tanya pakar
2.	email_user	VARCHAR(50)	-	Kolom untuk email user (pengguna)
3.	Pertanyaan	VARCHAR(250)	-	Kolom untuk pertanyaan
4.	jawabanPakar	VARCHAR(500)	-	Kolom untuk jawaban pakar
5.	Terjawab	BIT	-	Kolom untuk status pertanyaan
6.	waktuBertanya	Datetime	-	Kolom untuk waktu kapan pertanyaan diajukan pengguna
7.	waktuMenjawab	Datetime	-	Kolom untuk waktu kapan pertanyaan dijawab oleh pakar

Setelah menentukan entitas dan atribut yang dimiliki dari tiap entitas tersebut, tahapan selanjutnya adalah menggambarkan keterkaitan hubungan antar entitas tersebut (Diagram ER). Diagram ER yang digunakan untuk memperlihatkan hubungan antar entitas dalam sistem pakar diagnosis gangguan emosional pada anak adalah sebagai berikut ini



Gambar 7. ERD Aplikasi Sistem Pakar

Dalam melakukan rancang bangun sistem pakar, terdapat suatu tahapan yang disebut dengan perancangan basis pengetahuan. Desain basis pengetahuan berguna untuk memberikan gambaran dari aturan-aturan yang digunakan sistem untuk melakukan diagnosis penyakit. Aturan-aturan tersebut dapat terbentuk setelah mengumpulkan data-data dari pakar. Tabel 9 dijabarkan kode gejala dan data gejala yang berhubungan dengan penyakit yang menjadi bahan diagnosis sistem dan isi tabel basisdata sistem.

Tabel 9. Tabel Daftar Gejala

gejalaID	nama_gejala
G01	Gejala fisik
G02	Gejala psikologis
G03	Kecemasan saat harus berada di situasi yang ramai
G04	Kecemasan ketika pergi ke tempat umum
G05	Kecemasan ketika berpergian sendiri
G06	Berusaha menghindari dari situasi ramai atau di tempat umum atau berpergian sendiri
G07	Rasa cemas ketika berada diluar lingkungan keluarga
G08	Memiliki ketakutan sendiri pada situasi tertentu
G09	Pengaruh obat-obatan
G10	Kecemasan dalam keadaan yang biasa saja (cemas tanpa sebab) yang tidak perlu dikhawatirkan
G11	Sulit berkonsentrasi
G12	Memiliki kekhawatiran akan nasib buruk

G13	Tampak depresi (sifatnya sementara)
G14	Ketegangan motoric
G15	Overaktivitas otonomik
G16	Kebutuhan berlebihan untuk ditenangkan
G17	Stress akan kehidupan yang penuh masalah
G18	Ketidakstabilan menetap dari afek perasaan (suasana perasaan)
G19	Afek depresif
G20	Afek yang meninggi atau berubah
G21	Peningkatan aktivitas
G22	Gejala dimulai sejak dini
G23	Kehilangan minat dan kegembiraan
G24	Berkurangnya energy yang menuju meningkatnya keadaan mudah lelah
G25	Konsentrasi dan perhatian berkurang
G26	Harga diri dan kepercayaan diri berkurang
G27	Selalu merasa bersalah atau tidak berguna
G28	Memiliki pandangan masa depan yang suram
G29	Mencoba melakukan bunuh diri
G30	Tidur terganggu
G31	Nafsu makan berkurang
G32	Gangguan tersebut dialami sekurang-kurangnya 2 minggu
G33	Gejala tampak selama 1 bulan
G34	Gejala tampak setiap hari
G35	Gejala tampak selama 1 tahun atau lebih
G36	Memiliki dampak kelancaran aktivitas
G37	Overaktivitas otonomik
G38	Gejala tampak setiap hari
G39	Stress seolah merasa kehidupan yang penuh masalah
G40	Afek Depresif

Dalam model representasi pengetahuan kaidah produksi, terdapat beberapa tahapan yang harus dilalui dari pengetahuan yang didapatkan. Tahapan tersebut diantaranya adalah menyajikan pengetahuan yang berhasil didapatkan dalam bentuk tabel keputusan kemudian dari tabel keputusan dibuat menjadi pohon keputusan.

Tabel 10. Tabel Keputusan

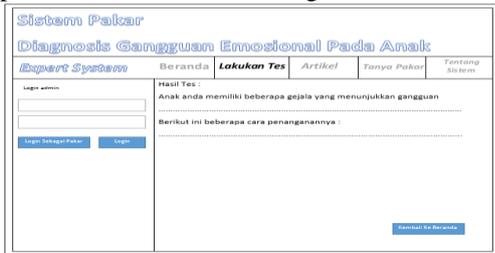
Gejala	Gangguan									
	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
G01										
G02										
G03										
G04										
G05										
G06										
G07										
G08										
G09										
G10										
G11										
G12										
G13										
G14										
G15										
G16										
G17										
G18										
G19										
G20										
G21										
G22										
G23										
G24										
G25										
G26										
G27										
G28										
G29										
G30										
G31										
G32										
G33										
G34										
G35										

3. Tampilan untuk menjawab pertanyaan



Gambar 12. Tampilan Menjawab Pertanyaan

4. Tampilan untuk melihat hasil diagnosis



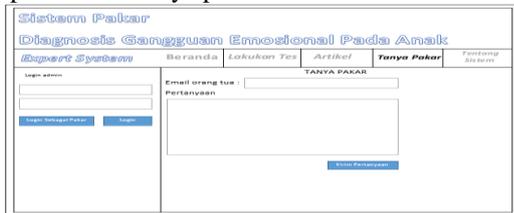
Gambar 13. Tampilan Melihat Hasil Diagnosis

5. Tampilan untuk artikel



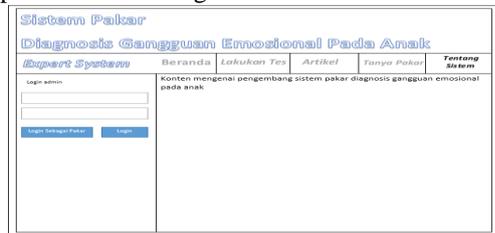
Gambar 14. Tampilan untuk Artikel

6. Tampilan untuk tanya pakar



Gambar 15. Tampilan untuk Tanya Pakar

7. Tampilan untuk tentang sistem



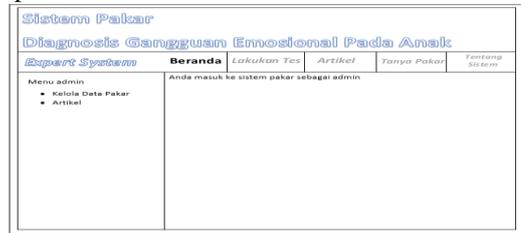
Gambar 16. Tampilan untuk Tentang Sistem

8. Tampilan untuk login admin



Gambar 17. Tampilan untuk Login

9. Tampilan beranda admin



Gambar 18. Tampilan Beranda Admin

10. Tampilan beranda pakar



Gambar 19. Tampilan Beranda Pakar

IV. IMPLEMENTASI DAN PENGUJIAN

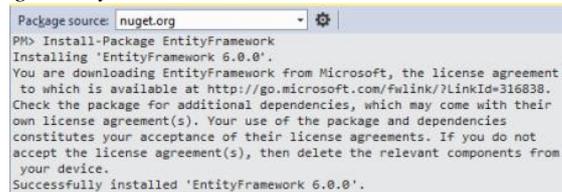
A. Tahap Pembuatan Sistem

Tahapan awal yang dilakukan dalam melakukan implementasi sistem adalah pembuatan basisdata, pemodelan basisdata yang dipilih adalah *codefirst*, yaitu membangun basis data dengan menggunakan baris kode yang nantinya diintegrasikan dengan Entity Framework untuk menerjemahkannya ke dalam basisdata SQL Server Local Database.

Dalam membuat basisdata beberapa tahapan yang dilakukan diantaranya:

1. Instalasi Entity-Framework

Instalasi Entity-Framework memiliki tujuan agar kode program dengan basisdata dapat terkoordinasi sesuai dengan perancangan sebelumnya. Instalasi Entity-Framework dilakukan pada *Package Manager Console* yang terdapat pada menu *Tools*. Pada *console* tersebut berikan perintah *Install-Package EntityFramework*.



Gambar 20. Instalasi Entity-Framework

2. Pemrograman Model

Pemrograman model dilakukan untuk membentuk tabel-tabel yang terdapat pada basis data rancang bangun sistem pakar diagnosis gangguan emosional pada anak. Dalam melakukan pemrograman model, setiap model diberikan referensi *library*

`System.ComponentModel.DataAnnotations.Schema` yang berfungsi untuk menggunakan keterangan seperti *class* tersebut akan terbentuk menjadi tabel, variabel tidak boleh null, serta keterangan lainnya sesuai dengan kebutuhan kolom-kolom untuk tabel pada basisdata yang akan dibentuk.

Beberapa *class* yang nantinya dibentuk menjadi tabel, diantaranya adalah: pasien, gangguan, gejala, diagnosis, pakar, tanya_pakar, artikel dan edUser. Salah satu contoh kode program dalam pemrograman model adalah sebagai berikut (*class* gejala).

```
[Table("gejala")]
public class gejala
{
    [Required]
    public String gejalaID { get; set; }
    [Required, MaxLength(100)]
    public String nama_gejala { get; set; }
    [Required, MaxLength(500)]
    public String pertanyaan_sistem { get; set; }

    public List<gejala> NodeLeft { get; set; }
    public List<gejala> NodeRight { get; set; }
    public gangguan DiagnosisSakit { get; set; }
}
```

3. DAL (Data Access Layer)

Data Access Layer merupakan salah satu dari bagian program yang menangani konteks basisdata. Class konteks basisdata merupakan class utama yang mengkoordinasikan antara fungsionalitas entity framework terhadap model yang sebelumnya telah dibuat. Dalam melakukan pemrograman rancang bangun sistem pakar diagnosis gangguan emosional berbasis aplikasi website, class konteks basisdata diberikan nama *edContext.cs* yang diturunkan dari class *System.Data.Entity.DbContext*.

4. Migrasi

Codefirst-migration merupakan salah satu fitur dari entity framework yang berguna untuk memecahkan masalah ketika melakukan perubahan pada bagian model. Dengan menggunakan metode migrasi, database akan tetap sinkron dengan perubahan yang terjadi di model sampai aplikasi benar-benar *deploy* pada tahap produksi.

Tahapan dalam melakukan migrasi adalah meniadakan database initializer pada *webconfig* dan memberikan nama pada basisdata untuk aplikasi seperti pada kode program berikut ini:

```
<connectionStrings>
<add name="edContext" connectionString="Data
Source=(LocalDb)\v11.0;AttachDbFilename=|DataDire
ctory|\aspnet-bismillah_ta-fix9.mdf;Initial
Catalog=bismillah_ta;Integrated Security=SSPI"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

Pada penelitian ini nama basis data yang digunakan adalah *aspnet-bismillah_ta-fix9.mdf*. Tahapan selanjutnya adalah mengaktifkan migrasi melalui *Package Manager Console* yang terdapat pada menu *Tools*. Berikan perintah pada console :

PM > Enable-Migrations -ContextTypeName : (nama class konteks). Proses mengaktifkan migrasi ditunjukkan pada gambar 21.

```
PM> Enable-Migrations -ContextTypeName:edContext
Checking if the context targets an existing database...
Code First Migrations enabled for project bismillah_ta.
```

Gambar 21. Mengaktifkan Migrasi

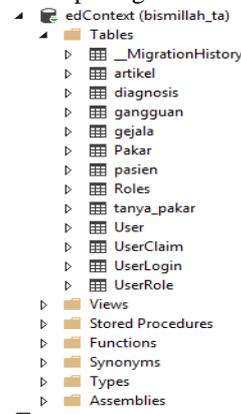
Setelah berhasil mengaktifkan migrasi, maka akan membentuk suatu folder baru yang bernama *Migrations*. Tahapan selanjutnya adalah dengan menambahkan migrasi

yang bertujuan memberi gambaran kepada pengembang sistem tentang perubahan apa saja yang dilakukan dalam model, jika penambahan migrasi baru dilakukan pertama kali maka barisan kode yang akan ditunjukkan kepada pengguna adalah gambaran mengenai basisdata yang akan terbentuk dengan menyesuaikan pada model yang sudah diprogram sebelumnya. Cara menambahkan migrasi ditunjukkan pada gambar 22.

```
PM> Add-Migration LastMigrations
Scaffolding migration 'LastMigrations'.
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is used to calculate the changes to your model when you scaffold the next migration. If you make additional changes to your model that you want to include in this migration, then you can re-scaffold it by running 'Add-Migration LastMigrations' again.
```

Gambar 22. Menambahkan Migrasi

Setelah melakukan migrasi pada *SQL Server Local Database* sudah terbentuk tabel-tabel basisdata sesuai dengan apa yang sebelumnya sudah dikodekan di model maupun di *data access layer*. Susunan tabel-tabel basisdata untuk sistem pakar diagnosis gangguan emosional pada anak berbasis aplikasi website ditunjukkan pada gambar 23.



Gambar 23. Struktur Tabel Basisdata Aplikasi Sistem Pakar

5. Update-Database

Tabel-tabel yang sudah terbentuk sebelumnya masih memiliki nilai null pada tiap-tiap atribut yang terdapat di dalam tabel. Sesuai dengan perancangan sebelumnya, beberapa tabel yang akan mendapatkan inisialisasi nilai adalah tabel gejala, tabel gangguan, tabel artikel dan tabel diagnosis. Untuk dapat melakukan inisialisasi nilai-nilai atribut pada tabel di basisdata, maka hal yang dilakukan adalah memberikan kode program di class *configuration* pada method *Seed*. Kode program yang terdapat pada method *Seed* tersebut dijalankan dengan memberikan perintah *update-database* pada *Package Console Manager* hingga muncul keterangan *Running Seed Method* yang berarti method *seed* berhasil dijalankan dan data sudah masuk ke basisdata.

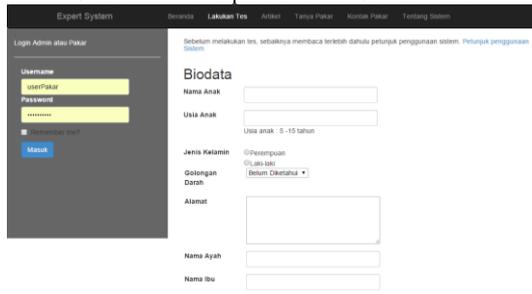
Setelah kelima tahapan tersebut dilalui, maka sudah terbentuk suatu basisdata untuk aplikasi sistem pakar. Tahapan selanjutnya yang dilakukan adalah melakukan pemrograman antarmuka pengguna dengan sistem (pemrograman *view*).

Dalam melakukan pemrograman dibagian *view*, tampilan dikelompokkan sesuai dengan bagiannya masing-masing seperti: tampilan khusus untuk Home (beranda), tampilan ketika pengguna masuk ke sistem sebagai admin, tampilan ketika pengguna masuk ke sistem sebagai pakar dan tampilan

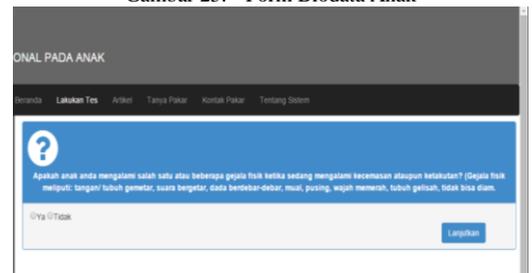
yang menangani pengguna. Beberapa tampilan dari aplikasi sistem pakar dapat terlihat pada gambar berikut ini.



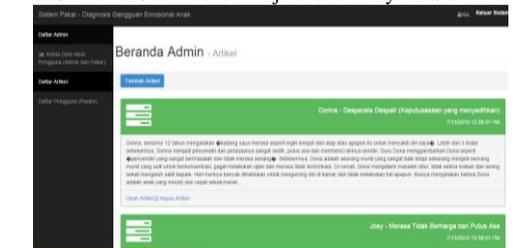
Gambar 24. Tampilan Beranda Sistem Pakar



Gambar 25. Form Biodata Anak



Gambar 26. Proses Menjawab Pertanyaan Sistem



Gambar 27. Tampilan Beranda Admin Menu Artikel



Gambar 28. Tampilan Beranda Pakar Menu List Gejala

B. Pemrograman Pohon Keputusan

Dalam melakukan diagnosis gangguan emosional, aplikasi sistem pakar ini menggunakan metode *forward chaining* berdasarkan dari pohon keputusan yang sebelumnya sudah digambarkan dalam perancangan sebelumnya. Pemrograman pohon keputusan tersebut dilakukan pada suatu class tersendiri

yang terpisah, dalam penelitian ini pemrograman pohon keputusan dilakukan pada class yang dinamakan *decisionclass.cs* dengan menempatkan class tersebut pada folder *Helper*. Beberapa method yang diberikan pada class *decisionclass.cs* ditunjukkan pada gambar

```
namespace bismillah_ta.Helper
{
    public class DecisionClass
    {
        edContext context = new edContext();
        public gejala root, current;

        public gejala setRootNode()...

        public String DisplayFirstQuestion()...

        public String NextQuestion(String YesNo, String pertanyaan)...)

        public bool HasDiagnosis(gejala NodeSekarang)...)

        public gangguan GetGangguan(gejala NodeSekarang)...)

        public gejala GetParentNodeLeft(gejala NodeSekarang)...)

        public gejala GetParentNodeRight(gejala NodeSekarang)...)

        public gejala GetNodeSekarang(String pertanyaan)...)
    }
}
```

Gambar 29. Method Pada Class decisionclass.cs

Proses kerja pada method yang terdapat di class *decisionclass* tersebut akan dijalankan bersamaan dengan method yang berada pada *controller* Home yang akan dibahas lebih rinci dalam proses pengujian sistem nanti. Penjelasan dari beberapa method yang berada di *decisionclass* adalah sebagai berikut ini:

- `public gejala setRootNode()`
Method tersebut berfungsi untuk mengatur node awal yang dijadikan sebagai *root*, sesuai dengan gambaran pohon keputusan yang sudah dirancang node yang dijadikan *root* adalah satu objek gejala dengan gejalaID = G01.
- `public String DisplayFirstQuestion()`
Method tersebut akan *return* pertanyaan sistem dari node *root*.
- `public gejala GetNodeSekarang(String pertanyaan)`
Method tersebut akan mendapatkan node yang sekarang dengan menggunakan parameter dari pertanyaan gejala yang sedang ditampilkan untuk pengguna.
- `public gejala GetParentNodeLeft(gejala NodeSekarang)`
Method tersebut mengecek apakah *NodeSekarang* merupakan parent dari *Node* lainnya, jika ya maka akan mengembalikan nilai *nodeLeft* node tersebut berupa satu objek gejala.
- `public gejala GetParentNodeRight(gejala NodeSekarang)`
Method tersebut mengecek apakah *NodeSekarang* merupakan parent dari *Node* lainnya, jika ya maka akan mengembalikan nilai *nodeRight* node tersebut berupa satu objek gejala.
- `public String NextQuestion(String YesNo, String pertanyaan)`
Method tersebut akan *return* pertanyaan sistem dari node sekarang.
- `public bool HasDiagnosis(gejala NodeSekarang)`
Method tersebut mengecek apakah *NodeSekarang* mempunyai *NodeLeft* yang berupa gangguan, jika ya maka

akan mengembalikan nilai *true* jika tidak maka akan mengembalikan nilai *false*.

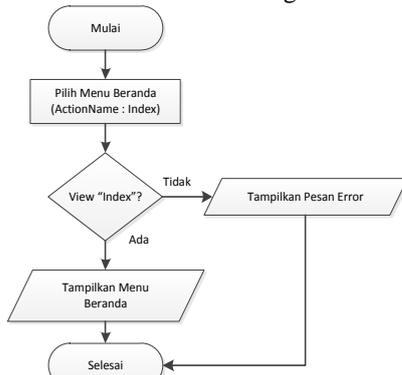
- public gangguan GetGangguan(gejala NodeSekarang)
Method tersebut akan mengembalikan nilai berupa satu objek gangguan apabila node sekarang mempunyai NodeLeft berupa gangguan.

C. Tahap Pengujian Sistem

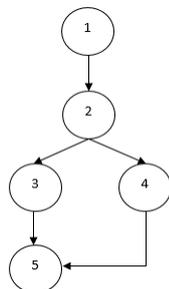
Pengujian aplikasi sistem pakar diagnosis gangguan emosional pada anak menggunakan metode *whitebox*. Dalam pengujian *whitebox* setiap fungsi yang ada akan diuji secara rinci jalur-jalur logika yang ada. Metode pengujiannya menggunakan Struktur Kontrol Program untuk memperoleh kasus uji. Pengujian akan dilakukan untuk masing-masing jalur, cabang dan perintah.

Fungsi-fungsi dari aplikasi sistem pakar yang akan dilakukan pengujian sistem diantaranya adalah pada halaman beranda beserta setiap navigasi menu beserta fitur-fitur yang terdapat pada menu tersebut, fungsi login dan logout serta pengujian terhadap metode *forward chaining* untuk melakukan diagnosis gangguan.

1. Pengujian menu beranda aplikasi sistem pakar. Beranda merupakan halaman awal yang akan ditampilkan ketika mengakses aplikasi sistem pakar. Diagram alir proses pengujian menu beranda adalah sebagai berikut.



Gambar 30. Diagram Alir Menu Beranda



Gambar 31. Flowgraph Menu Beranda

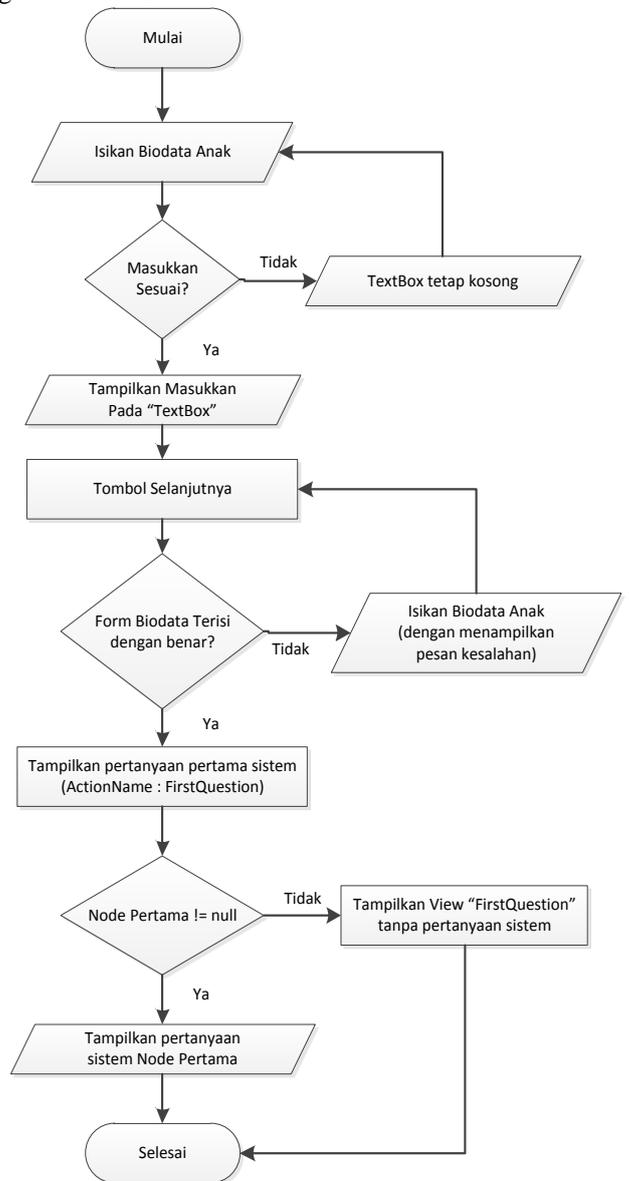
Tabel 12. Jalur Proses Menu Beranda

Jalur	Tahapan	Keterangan
1	1-2-3-5	Menampilkan Menu Beranda
2.	1-2-4-5	View "Index" tidak ada dan menampilkan pesan kesalahan

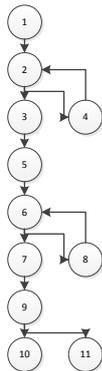
Dari *flowgraph* pada gambar 31 didapatkan nilai *cyclomatic complexity* $V(G)=E-N+2$ atau $V(G)=P+1$, dimana E merupakan jumlah Edge, N merupakan jumlah Node,

sedangkan P merupakan jumlah cabang, sehingga $V(G)=1+1=2$.

2. Pengujian Lakukan Tes dalam mengisi biodata pasien. Pada halaman ini, pengguna diminta untuk mengisi biodata anak sesuai dengan kebutuhan sistem, seperti masukkan untuk nama hanya diizinkan abjad saja tidak dapat diketikkan angka. Sebaliknya untuk umur hanya meminta angka saja, tidak mengizinkan abjad. Pengecekan dapat dilakukan dengan mengetikkan masukkan pada *textbox* yang disediakan, sedangkan tampilan pesan kesalahan dapat ditampilkan setelah menekan tombol selanjutnya. Diagram alir untuk pengujian navigasi menu Lakukan Tes (mengisi form biodata) ditunjukkan pada gambar 32 berikut.



Gambar 32. Diagram Alir Isi Biodata dan Tombol Selanjutnya



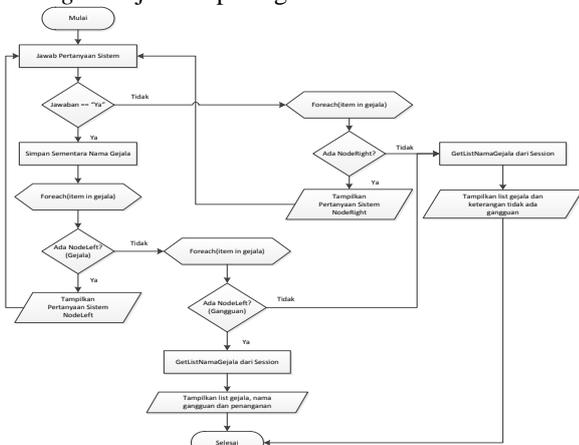
Gambar 33. Flowgraph Isi Biodata dan Tombol Selanjutnya

Tabel 13. Jalur Proses Isi Biodata dan Tombol Selanjutnya di Halaman Menu "Lakukan Tes"

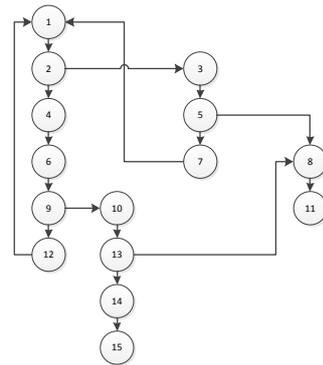
Jalur	Tahapan	Keterangan
1.	1-2-3-5-6-7-9-10	Menampilkan pertanyaan sistem node pertama, data anak yang dimasukkan benar
2.	1-2-3-5-6-7-9-11	Masukkan yang diberikan benar tetapi tidak ada view FirstQuestion
3.	1-2-3-4-1-2-3-5-6-8-7-9-10	Pengecekan terhadap kesesuaian input lalu diproses menampilkan FirstQuestion
4.	1-2-3-4-1-2-3-5-6-8-7-9-11	Pengecekan terhadap kesesuaian input lalu diproses, tidak ada view FirstQuestion

Dari flowgraph pada gambar 31 didapatkan nilai cyclomatic complexity $V(G)=E-N+2$ atau $V(G)=P+1$, dimana E merupakan jumlah Edge, N merupakan jumlah Node, sedangkan P merupakan jumlah cabang, sehingga $V(G)=3+1=4$.

- Pengujian metode *forward chaining* yang diterapkan pada aplikasi sistem pakar ini. Penerapan metode runut maju terdapat ketika pengguna sistem memberikan jawaban terhadap pertanyaan yang diajukan oleh sistem. Dalam melakukan proses diagnosis, sistem diharapkan dapat melakukan 3 (tiga) kemungkinan hasil diantaranya: anak memiliki gejala yang menunjukkan suatu gangguan emosional, anak memiliki gejala-gejala tetapi belum dapat dikatakan anak tersebut mengalami gangguan emosional karena tidak semua gejala dari suatu gangguan dialami, anak tidak mengalami gejala apapun dan tidak mengalami gangguan emosional. Diagram alir untuk kasus uji *forward chaining* ditunjukkan pada gambar berikut.



Gambar 34. Diagram Alir Kasus Uji Runut Maju



Gambar 35. Flowgraph Runut Maju

Tabel 14. Jalur Proses Fungsi Runut Maju

Jalur	Tahapan	Keterangan
1.	1-2-3-5-8-11	Anak tidak mengalami gangguan dan gejala apapun
2.	1-2-3-5-7-1-2-4-6-9-10-13-8-11	Anak mengalami gejala tetapi belum menunjukkan suatu gangguan
3.	1-2-3-5-7-1-2-4-6-9-10-13-14-15	Anak memiliki gejala dan sebuah gangguan
4.	1-2-4-6-9-10-11-13-14-15	Anak memiliki gejala dan sebuah gangguan
5.	1-2-4-6-9-12-1-2-4-6-9-10-11-13-14-15	Anak memiliki gejala dan sebuah gangguan

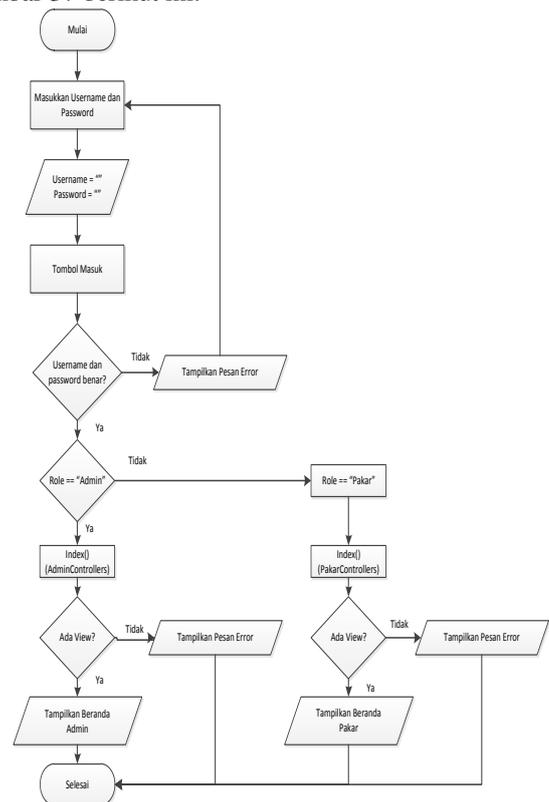
Dari Flowgraph pada Gambar 4.46 didapatkan nilai cyclomatic complexity $V(G)=E-N+2$ atau $V(G)=P+1$, dimana E merupakan jumlah Edge, N merupakan jumlah Node, sedangkan P merupakan jumlah cabang, sehingga $V(G)=7+1=8$. Pengujian yang dilakukan untuk mengecek proses runut maju dapat digambarkan pada tabel 4.12 berikut:

Tabel 15. Tabel Skenario Pengujian Dignosis

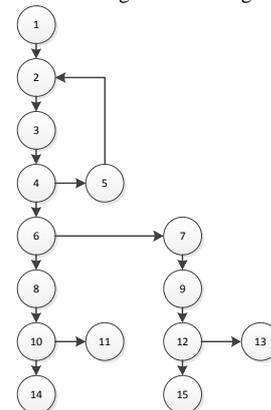
No	Nama Diagnosis	Runut Jawaban dari Pengguna	Hasil Diagnosis (Setelah Pengguna Memberikan Jawaban)	Hasil Sesuai
1.	Agorafobia (A01)	G09 = "Tidak", G02 = "Ya", G01 = "Ya", G05 = "Ya", G06 = "Ya", G04 = "Ya", G03 = "Ya"	Agorafobia (A01)	Sesuai
2.	Fobia Sosial (A02)	G09 = "Tidak", G02 = "Ya", G01 = "Ya", G05 = "Ya", G06 = "Ya", G04 = "Tidak", G07 = "Ya"	Fobia Sosial (A02)	Sesuai
3.	Fobia Khas (A03)	G09 = "Tidak", G02 = "Ya", G01 = "Ya", G05 = "Ya", G06 = "Ya", G04 = "Tidak", G07 = "Tidak", G08 = "Ya"	Fobia Khas (A03)	Sesuai
4.	Gangguan Panik (A04)	G09 = "Tidak", G01 = "Tidak", G11 = "Tidak", G10 = "Ya"	Gangguan Panik (A04)	Sesuai
5.	Gangguan Cemas	G09 = "Tidak", G01 = "Tidak", G11 =	Gangguan Cemas	Sesuai

	Menyeluruh (A05)	"Ya", G12 = "Ya", G13 = "Ya", G14 = "Ya", G15 = "Ya", G16 = "Ya", G33 = "Ya", G38 = "Ya"	Menyeluruh (A05)	
6.	Gangguan Campuran Anxiety dan Depresi (A06)	G09 = "Tidak", G01 = "Ya", G02 = "Ya", G05 = "Tidak", G37 = "Ya", G39 = "Ya"	Gangguan Campuran Anxiety dan Depresi (A06)	Sesuai
7.	Siklotimia (A07)	G09 = "Tidak", G01 = "Tidak", G11 = "Tidak", G10 = "Tidak", G40 = "Ya", G23 = "Ya", G24 = "Ya", G25 = "Ya", G26 = "Ya", G27 = "Ya", G28 = "Ya", G29 = "Ya", G30 = "Ya", G31 = "Tidak", G18 = "Ya"	Siklotimia (A07)	Sesuai
8.	Distimia (A08)	G09 = "Tidak", G01 = "Tidak", G11 = "Ya", G12 = "Tidak", G17 = "Ya", G19 = "Ya", G22 = "Ya", G35 = "Ya"	Distimia (A08)	Sesuai
9.	Manik (A09)	G09 = "Tidak", G01 = "Tidak", G11 = "Tidak", G10 = "Tidak", G40 = "Tidak", G20 = "Ya", G21 = "Ya", G34 = "Ya", G36 = "Ya"	Manik (A09)	Sesuai
10.	Depresi (A10)	G09 = "Tidak", G01 = "Tidak", G11 = "Tidak", G10 = "Tidak", G23 = "Ya", G24 = "Ya", G25 = "Ya", G36 = "Ya", G37 = "Ya", G38 = "Ya", G39 = "Ya", G30 = "Ya", G31 = "Ya", G32 = "Ya"	Depresi (A10)	Sesuai
11.	Normal (Tanpa Gejala)	G09 = "Tidak", G01 = "Tidak", G11 = "Tidak", G10 = "Tidak", G40 = "Tidak", G20 = "Tidak"	Normal (Tanpa Gejala)	Sesuai
12.	Normal (Terdapat Beberapa Gejala)	G09 = "Tidak", G01 = "Ya", G02 = "Tidak"	Normal (Terdapat Beberapa Gejala)	Sesuai
13.	- (Gejala tidak dipengaruhi obat-obatan tertentu)	G09 = "Ya"	-	Sesuai

4. Pengujian fungsi login sistem. Pengujian yang dilakukan adalah berupa validitas *username* dan *password* serta pengujian apakah sistem sudah dapat membedakan pengguna yang melakukan login adalah seorang pakar atau seorang admin. Selain itu pengujian juga dilakukan untuk menguji apakah menu-menu yang disediakan untuk pakar maupun admin sudah dapat diakses sesuai dengan *role* yang diberikan. Diagram alir untuk kasus uji tersebut ditunjukkan pada gambar 36 sedangkan *flowgraph* ditunjukkan pada gambar 37 berikut ini.



Gambar 36. Diagram Alir Fungsi Login



Gambar 37. Flowgraph Fungsi Login

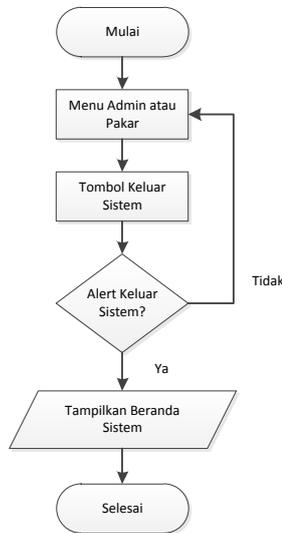
Tabel 16. Jalur Proses Fungsi Login

Jalur	Tahapan	Keterangan
1.	1-2-3-4-5-2-3-4-6-8-10-14	Login sebagai admin dan menampilkan beranda admin

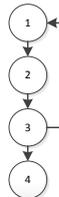
2.	1-2-3-4-5-2-3-4-6-8-10-11	Login sebagai admin tetapi beranda admin tidak tersedia
3.	1-2-3-4-5-2-3-4-6-7-9-12-15	Login sebagai pakar dan menampilkan beranda admin
4.	1-2-3-4-5-2-3-4-6-7-9-12-15	Login sebagai pakar tetapi beranda admin tidak tersedia

Dari *Flowgraph* pada Gambar 37 didapatkan nilai *cyclomatic complexity* $V(G)=E-N+2$ atau $V(G)=P+1$, dimana E merupakan jumlah Edge, N merupakan jumlah Node, sedangkan P merupakan jumlah cabang, sehingga $V(G)=3+1=4$.

5. Pengujian terakhir adalah bagaimana pengguna (admin atau pakar) yang sudah dapat masuk ke dalam sistem dapat keluar dari sistem dengan memilih menu “Keluar Sistem”. Diagram alir untuk kasus uji tersebut ditunjukkan pada gambar 38 sedangkan *flowgraph* ditunjukkan pada gambar 39 berikut ini.



Gambar 38. Diagram Alir Fungsi Keluar Sistem



Gambar 39. *Flowgraph* Keluar Dari Sistem

Tabel 17. Jalur Proses Keluar Sistem

Jalur	Tahapan	Keterangan
1	1-2-3-1-2-3-4	Tidak jadi keluar dari sistem, lalu kembali ke menu awal
2.	1-2-3-4	Keluar dari sistem

Dari *Flowgraph* pada Gambar 39 didapatkan nilai *cyclomatic complexity* $V(G)=E-N+2$ atau $V(G)=P+1$, dimana E merupakan jumlah Edge, N merupakan jumlah Node, sedangkan P merupakan jumlah cabang, sehingga $V(G)=1+1=2$.

Pengujian selanjutnya adalah pengujian sistem untuk pengguna dengan tujuan mengetahui manfaat aplikasi. Proses pengujian dilakukan dengan cara meminta orang tua yang memiliki anak dengan rentang usia 5-15 tahun sebagai responden untuk menjawab 10 soal dengan pilihan jawaban berupa ‘Ya’ atau ‘Tidak’.

Proses pengujian dilakukan dengan cara menjelaskan konsep dari sistem pakar serta mendemonstrasikan aplikasi sistem pakar diagnosis gangguan emosional pada anak kepada responden, selanjutnya meminta responden untuk menjawab 10 soal yang tertera pada lembar kuisioner.

Dari hasil jawaban responden yang didapat dari survey selanjutnya akan dimasukkan ke dalam rumus berikut untuk mencari koefisien reproduibilitas (Kr)

$$Kr = 1 - \frac{e}{n}$$

sehingga didapat :

$$Kr = 1 - \frac{26}{100}$$

$$Kr = 0.74$$

Setelah Kr diketahui, langkah selanjutnya ialah menghitung koefisien skalabilitas (Ks) dengan menggunakan rumus berikut :

$$Ks = 1 - \frac{e}{x}$$

sehingga didapat :

$$Ks = 1 - \frac{26}{0,5(100-26)}$$

$$Ks = |1 - 2|$$

$$Ks = 1$$

Karena Ks memiliki nilai $>0,60$ maka disimpulkan variabel yang ada baik untuk digunakan dalam survei dan hasil uji yang didapat berkaitan dengan penelitian ini adalah responden menunjukkan respon yang bagus pada aplikasi ini beserta konsep yang terdapat di dalamnya.

Aplikasi ini juga telah diuji dan diperiksa kebenaran data yang terdapat didalamnya dan dinyatakan data benar, tetapi masih terlalu dini untuk dapat diimplementasikan secara langsung oleh beberapa orang psikolog diantaranya Maharani, M.Psi dan Nuning Suhardiningsih, M.Psi. Selain oleh para psikolog, aplikasi juga diuji oleh konsultan kesehatan jiwa yaitu Witcha Rea A, S.Psi dan Margareta Dwina K, S.Psi. Keterangan wawancara dan hasil pengujian terlampir di dalam laporan ini.

KESIMPULAN

Kesimpulan dari penelitian tugas akhir ini adalah sebagai berikut:

1. Aplikasi sistem pakar diagnosis gangguan emosional pada anak merupakan sebuah aplikasi berbasis website yang memiliki fitur inti dapat melakukan diagnosis gangguan emosional melalui gejala yang diberikan sistem dalam bentuk pertanyaan kepada pengguna.
2. Metode yang dilakukan pada sistem pakar ini adalah dengan *forward chaining* (runut maju) dengan menggunakan model representasi kaidah produksi. Dalam menggunakan model tersebut, informasi yang diperoleh dari pakar diterjemahkan ke dalam tabel keputusan dan selanjutnya dibuat suatu pohon keputusan.
3. Pengujian yang dilakukan pada aplikasi ini adalah menggunakan metode *whitebox* dan memperoleh hasil

cyclomatic complexity yang diperoleh sudah sesuai dengan tabel jalur proses yang dibuat, sehingga dapat disimpulkan bahwa logika yang diberikan untuk sistem sudah sesuai dengan kebutuhan sistem.

4. Pengujian sistem dengan tujuan validitas data dilakukan oleh empat orang pakar dibidang kesehatan jiwa, memberikan kesimpulan bahwa gejala yang mengarah ke dalam suatu gangguan emosional sudah sesuai, tetapi masih terlalu dini jika aplikasi diimplementasikan secara langsung dan dipergunakan, karena batasan gangguan masih mempunyai cakupan yang terlalu luas.
5. Pengujian sistem dengan tujuan mengetahui manfaat aplikasi dilakukan oleh sepuluh orang pengguna (responden) memberikan hasil nilai Kr > 0.8 dan nilai Ks > 0.6 yang artinya responden menunjukkan respon yang bagus pada aplikasi ini beserta konsep yang terdapat di dalamnya.

SARAN

Saran yang diberikan dalam upaya pengembangan sistem yang lebih baik dikemudian hari diantaranya:

1. Sistem pakar diagnosis gangguan emosional anak ditujukan untuk anak normal (tanpa ada perhitungan faktor lain, seperti keterbelakangan mental), sehingga memungkinkan untuk pengembangan aplikasi selanjutnya dapat memperhitungkan faktor-faktor lain yang mungkin mempengaruhi seperti latar belakang lingkungan sosial maupun keluarga.
2. Penambahan fitur rekap data (termasuk cetak rekap data) yang dapat diurutkan berdasarkan waktu dari hasil pengguna yang sudah melakukan tes pada sistem pakar.

DAFTAR PUSTAKA

- [1.] F. F. Rohman and A. Fauziah, "Rancang Bangun Aplikasi Sistem Pakar Untuk Menentukan Jenis Gangguan Perkembangan Pada Anak," *Media Informatika*, vol. 6, pp. 1-23, 2008.
- [2.] E. Rich, K. Knight and S. B. Nair, *Artificial Intelligence Third Edition*, The McGraw-Hill Companies, 2009
- [3.] R. Hidayat dan S. Minarni, "Rancang Bangun Aplikasi Sistem Pakar untuk Kerusakan Komputer dengan Metode Backward Chaining," *Jurnal TEKNOIF*, pp. 26-35, 2013.
- [4.] T. Sutojo and dkk, *Kecerdasan Buatan*, Yogyakarta: Andi Publishing, 2011.
- [5.] S. Hartati and S. Iswanti, *Sistem Pakar dan Pengembangannya*, Yogyakarta: Graha Ilmu, 2008.
- [6.] wikipedia, "wikipedia," [Online]. Available: http://id.wikipedia.org/wiki/Microsoft_Visual_Studio. [Accessed 12 March 2015].
- [7.] D. Esposito, *Programming Microsoft ASP.NET 4*, USA: Microsoft, 2011.
- [8.] P. Jeffrey and dkk, *ASP.NET MVC 4 IN ACTION*, Shelter Island.
- [9.] M. Zainul, "Thinking Original Entity Framework," 1 July 2011. [Online]. Available:

<https://zainulmasadi.wordpress.com/2011/07/01/entity-framework>. [Accessed 12 march 2015].

- [10.] Nagel, Christian, dkk. *Professional C#4 and .NET 4*, Canada: Wiley Publishin, Inc. 2010.
- [11.] A. Nugroho, *Algoritma dan struktur data dengan C#*, Yogyakarta: ANDI, 2009.
- [12.] A. Troelsen, *Pro C# 2008 and the .NET 3.5 Platform (4th Edition)*, New York: Springer-Verlag New York, Inc., 2007.
- [13.] Aiskahendra, "Perbedaan Razor dan aspx," 17 December 2010. [Online]. Available: <https://aiskahendra.wordpress.com/2010/12/17/asp-net-mvc-3-mengenai-syntax-razor>. [Accessed 12 March 2015].
- [14.] kuliahs0, "Lapisan Perangkat Lunak," 13 March 2012. [Online]. Available: <https://kuliahs0.wordpress.com/2012/03/13/software-layer-lapisan-perangkat-lunak>. [Accessed 12 March 2015]
- [15.] E. J. Mash and D. A. Wolfe, *Abnormal Child Psychology, Fourth Edition*, USA: Wadsworth Cengage Learning, 2010.
- [16.] Maslim, Rusdi. *Buku Saku Diagnosis Gangguan Jiwa Rujukan Ringkas dari PPDGJ-III*
- [17.] *Pedoman Penggolongan Diagnosis Gangguan Jiwa di Indonesia, III, PPDGJ III*, Departemen Kesehatan RI, Direktorat Jendral Pelayanan Medik, 1993.