

# SISTEM PAKAR DETEKSI KUALITAS AIR DENGAN METODE *FORWARD CHAINING* PADA LABORATORIUM “AQUARYAN” MARISA

Bahrin, S.Kom.,MT.  
Universitas Pohuwato

Jl.Trans Sulawesi No. 147 Marisa Kaupaten Pohuwato kode pos 96266  
bahrindahlan@gmail.com

*Air merupakan kebutuhan utama bagi proses kehidupan di bumi, sehingga tidak ada kehidupan seandainya di bumi tidak ada air. Namun demikian, air dapat menjadi malapetaka bilamana tidak tersedia dalam kondisi yang benar, baik kualitas maupun kuantitasnya. Air yang relatif bersih sangat didambakan oleh manusia, baik untuk keperluan hidup sehari-hari, untuk keperluan industri, keperluan pertanian dan lain sebagainya. Sehingga secara kualitas, sumber daya air telah mengalami penurunan. Demikian pula secara kuantitas, yang sudah tidak mampu memenuhi kebutuhan yang terus meningkat. Oleh sebab itu dibutuhkan seorang pakar yang dapat menangani masalah kualitas air. Karena sistem ini adalah milik instansi maka dari itu sistem pakar ini merupakan sistem pakar yang diperuntukkan untuk masyarakat Pohuwato sebagai tempat kami melakukan penelitian. Hasil pengujian yang dilakukan dengan metode white box testing dan bases path testing yang menghasilkan nilai  $V(G) = CC$ , dimana  $V(G) = 4$  dan  $CC = 4$ , sehingga didapat bahwa logika flowchart perhitungan vektor  $S$  dan Vektor  $V$  serta perankingan adalah benar dan berdasarkan pengujian black box yang meliputi uji input proses dan output dengan mengacu pada rancangan perangkat lunak yang sudah dibuat telah terpenuhi dengan hasil sesuai dengan rancangan.*

Kata kunci: *Sistem Pakar, Kualitas Air, Logika, rancangan, White Box*

## I. PENDAHULUAN

Air merupakan komponen lingkungan yang penting bagi kehidupan. Makhluk hidup di muka bumi ini tak dapat terlepas dari kebutuhan akan air. Air merupakan kebutuhan utama bagi proses kehidupan di bumi, sehingga tidak ada kehidupan seandainya di bumi tidak ada air. Namun demikian, air dapat menjadi malapetaka bilamana tidak tersedia dalam kondisi yang benar, baik kualitas maupun kuantitasnya. Air yang relatif bersih sangat didambakan oleh manusia, baik untuk keperluan hidup sehari-hari, untuk keperluan industri, keperluan pertanian dan lain sebagainya.

Dewasa ini air menjadi masalah yang perlu mendapat perhatian yang seksama dan cermat. Air yang baik sesuai standar tertentu, saat ini menjadi barang yang mahal, karena air sudah banyak tercemar oleh bermacam-macam limbah dari hasil kegiatan manusia. Sehingga secara kualitas, sumber daya air telah mengalami penurunan. Demikian pula secara kuantitas, yang sudah tidak mampu memenuhi kebutuhan yang terus meningkat. Oleh sebab itu dibutuhkan seorang pakar yang dapat menangani masalah kualitas air. Akan tetapi pada saat sekarang ini belum tentu dapat memakai seorang pakar karena dipandang dari segi keuangan maupun waktu pakar tersebut. Dengan adanya masalah diatas, maka peranan komputer akan sangat diperlukan dalam membantu perusahaan khususnya bagian personalia dan masyarakat dalam memilih kualitas air yang baik untuk dikonsumsi. Oleh karena itu, maka dibuatlah suatu sistem pakar untuk memudahkan perusahaan dan masyarakat (*User*) dalam memilih air yang akan dikonsumsi baik berdasarkan kriteria-kriteria yang sesuai dengan kebutuhan perlukan. Solusi dari permasalahan ini adalah penggunaan sistem pakar yang berbasis komputer.

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan dan kini manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar[4]. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangn. Sistem pakar ini juga akan dapat membantu aktivitas pakar sebagai asisten yang berpengalaman dan mempunyai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan. Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu.

II. STUDI PUSTAKA

A. Sistem Pakar

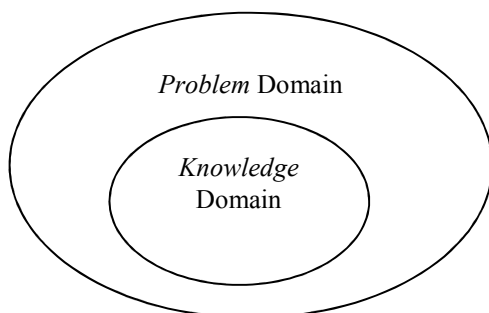
Sistem pakar merupakan penyelesaian pendekatan yang sangat berhasil/bagus untuk permasalahan *Artificial intelligence* klasik dari pemrograman *intelligent* (cerdas). Profesor Edward Feigenbaum dari Universitas Stanford yang merupakan seorang pelopor awal dari teknologi sistem pakar, yang mendefinisikan sistem pakar sebagai "suatu program komputer cerdas yang menggunakan *knowledge* (pengetahuan) dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang yang ahli untuk menyelesaikannya." [5]. Suatu sistem pakar adalah suatu sistem komputer yang menyamai (*emulates*) kemampuan pengambilan keputusan dari seorang pakar. Istilah *emulates* berarti bahwa sistem pakar di harapkan dapat bekerja dalam semua hal seperti seorang pakar. Suatu emulasi jauh lebih kuat daripada suatu simulasi yang hanya membutuhkan sesuatu yang bersifat nyata dalam beberapa bidang atau hal.

Walaupun tujuan umum penyelesaian masalah masih jauh dari apa yang diharapkan, namun sistem pakar berfungsi sangat baik dalam batasan domainnya. Hal ini dapat di buktikan bahwa sistem pakar telah banyak diaplikasikan dalam berbagai bidang yang trend saat ini, seperti bisnis, kedokteran, ilmu pengetahuan, dan teknik. Hasil tersebut dengan mudah dapat ditemukan di dalam buku-buku, jurnal, konferensi-konferensi, dan produksi yang berkenaan dengan sistem pakar.

B. Konsep Dasar Sistem Pakar

Bagian dalam sistem pakar terdiri dari 2 komponen utama, yaitu *knowledge base* yang berisi *knowledge* dan mesin inferensi yang menggambarkan kesimpulan tersebut merupakan respons dari sistem pakar atas permintaan pengguna.

Penggunaan *system knowledge base* (basis pengetahuan) juga di rancang untuk aksi pemandu cerdas seorang ahli. Pemandu cerdas di rancang dengan teknologi sistem pakar karena banyak memberikan keuntungan terhadap pengembangannya.



Gambar 1. Hubungan Antara Domain Knowledge Dan Domain Masalah

*Knowledge* dari sistem pakar tentang penyelesaian masalah yang khusus disebut dengan domain *knowledge* dari suatu pakar. Sebagai contoh, sistem pakar kedokteran yang dirancang untuk mendiagnosis infeksi penyakit akan mempunyai suatu uraian *knowledge* tentang gejala-gejala

penyakit yang disebabkan oleh infeksi penyakit, gejala dan cara pengobatan. Hubungan antara domain masalah

Dan domain *knowledge*. Dapat dilihat juga bahwa domain *knowledge* secara keseluruhan merupakan bagian dari domain masalah

Seorang pakar dan sistem pakar mempunyai banyak perbedaan. [6] mengemukakan perbandingan kemampuan antara seorang pakar dengan sebuah sistem pakar.

III. METODE PENELITIAN

Adapun metode yang digunakan dalam penelitian ini adalah metode *deskriptif*, yaitu penelitian yang berusaha untuk memecahkan masalah yang ada sekarang secara sistematis berdasarkan data-data yang ada. Tujuan dari penelitian *deskriptif* ini adalah untuk membuat deskriptif, menggambarkan atau melukiskan secara sistematis, faktual dan akurat mengenai fakta-fakta, sifat-sifat serta hubungan antar fenomena yang diteliti.

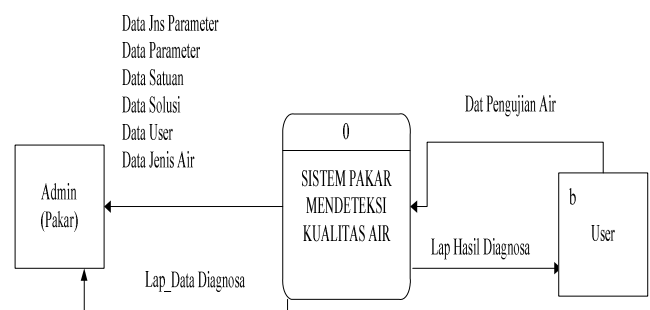
A. Tahap analisis

Gambaran sistem yang akan dibuat

Tahap analisis merupakan tahap penguraian dari suatu sistem informasi dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dalam kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Didalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut :

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisa sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

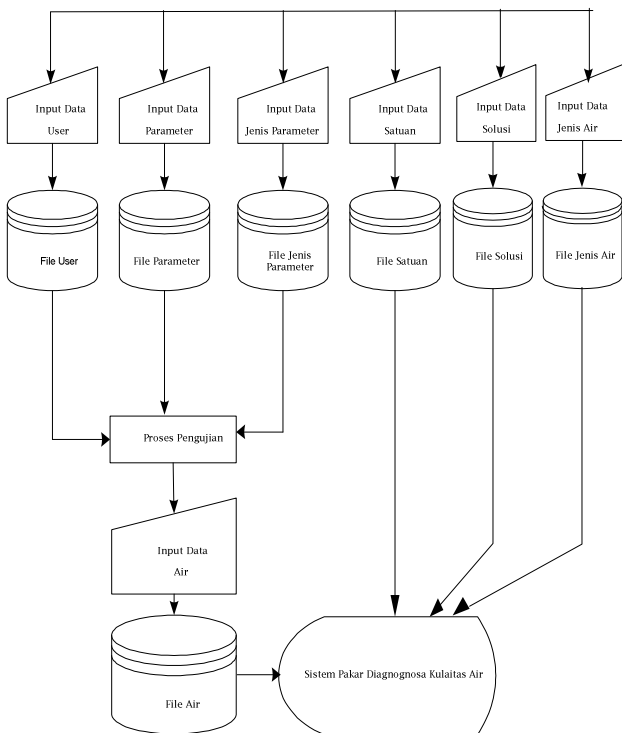
Sistem yang diusulkan  
 DFD ( Diagram Arus Data) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.



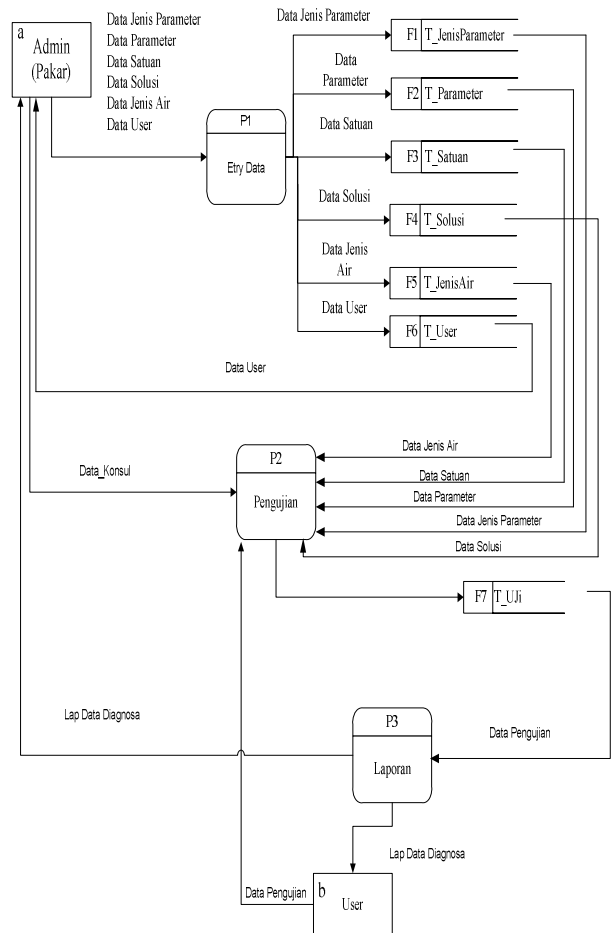
Gambar. 2. Diagram Konteks

Tabel 1. Perbandingan Seorang Pakar (*Human Expert*) dengan Sistem Pakar (*Expert Sistem*)

Faktor	Human Expert	Expert System
<i>Time Availability</i>	Hari Kerja	Setap saat
Geografis	Lokal/tertentu	Dimana saja
Keamanan	Tidak tergantung	Dapati diganti
Perishable/dapat habis	Ya	Tidak
Perfomansi	Variabel	Konsisten
Kecepatan	Variabel	Konsisten dan lebih cepat
Biaya	Tinggi	Tejangkau



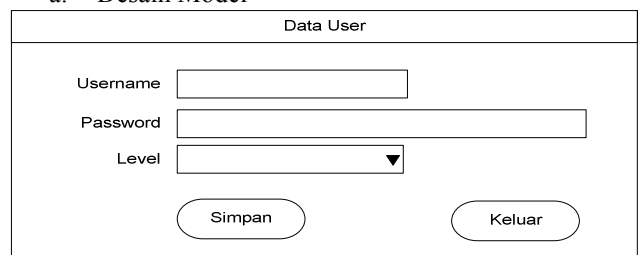
Gambar 3 Sistem yang di usulkan



Gambar 4. Daigram Arus Data (DAD)

B. Tahap Desain

a. Desain Model



Gambar 5. Desain Input Login

Merupakan tahapan yang lebih berfokus pada spesifikasi detail berbasis komputer. Sistem yang digunakan adalah *Model-Driven Design*, yaitu sebuah pendekatan desain sistem yang menekankan penggambaran model sistem untuk mendokumentasikan aspek teknis dan implementasi dari sebuah sistem. Dimana pada tahap ini kita melakukan pertimbangan-pertimbangan mengenai bagaimana suatu sistem akan diterapkan, baik dalam teknologi dan lingkungan implementasi. Pada tahap ini digunakan *Data Flow Diagram* (DFD), dimana kita memodelkan persyaratan bisnis logis dari suatu sistem informasi. DFD memodelkan keputusan-keputusan teknis dan desain manusia

untuk diimplementasikan sebagai bagian dari suatu sistem informasi.

b. Desain *Output*

Desain *output* dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk *output-output* dari sistem yang akan dibuat. Desain *output* terinci terbagi atas dua, yaitu desain *output* berbentuk laporan di media kertas dan desain *output* dalam bentuk dialog di layar terminal (*monitor*).

c. Desain *Input*

Masukan merupakan awal dimulainya proses pengolahan informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil dari transaksi tidak lepas dari data yang dimasukkan. Desain *input* terinci dimulai dari desain dokumen dasar sebagai penangkap *input* yang pertamakali. Jika dokumen dasar tidak didesain dengan baik, kemungkinan *input* yang tercatat dapat salah bahkan kurang.

d. Desain *Database*

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Penerapan *database* dalam aplikasi disebut *database sistem*.

e. Desain Teknologi

Pada tahap ini kita menentukan teknologi yang akan dipergunakan dalam menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

### C. Tahap pembuatan

Merupakan tahapan dimana kita melakukan pengembangan, melakukan tahap produksi sesuai dari hasil analisa dan desain sistem yang sebelumnya. Termasuk didalamnya membangun sebuah aplikasi, menulis *listing* program dan membangunnya dalam bentuk sebuah antarmuka dan integrasi sistem-sistem program yang terdiri dari *input*, *proses*, dan *output*, yang tersusun dalam sebuah sistem *menu* sehingga dapat dijalankan oleh pengguna sistem. Dalam tahapan ini, penulis menggunakan bahasa pemrograman Borland Delphi 7.0 dan *Database Mysql*.

### D. Tahap pengujian

Tahap ini dilakukan setelah semua modul selesai dibuat, dan program dapat berjalan, dimana seluruh perangkat lunak, program tambahan, dan semua program yang terlibat dalam pembangunan sistem diuji untuk memastikan sistem dapat berjalan sesuai dengan rancangan atau belum. Pengujian yang dilakukan dengan dua teknik pengujian, yaitu;

a. *White Box*

Dalam pengujian *White Box* ini dengan membuat bagan alir program, *listing* program, grafik alir, pengujian *basis path* serta perhitungan *Cyclomatic Complexity*.

b. *Black Box*

Pengujian *Black Box* yang termasuk dalam tahap ini yaitu menguji antarmuka sistem, apakah sebuah sistem setelah diberikan ke pengguna dapat dioperasikan atau tidak.

### E. Tahap implementasi

Tahap implementasi sistem (*Sistem Implementation*) merupakan tahap meletakkan sistem supaya siap untuk dioperasikan.

### F. Operasi dan Pemeliharaan

Setelah masa sistem berjalan sepenuhnya menggantikan sistem lama, sistem memasuki pada tahapan operasi dan pemeliharaan. [8] membagi pemeliharaan perangkat lunak menjadi tiga macam, yaitu :

a. Pemeliharaan Perfektif

Pemeliharaan Perfektif ditujukan untuk memperbarui sistem lama sebagai tanggapan atas perubahan kebutuhan pemakai dan kebutuhan organisasi, meningkatkan efisiensi sistem, dan memperbaiki dokumentasi.

b. Pemeliharaan Adaptif

Pemeliharaan Adaptif berupa perubahan aplikasi untuk menyesuaikan diri terhadap lingkungan perangkat keras dan perangkat lunak baru. Sebagai contoh pemeliharaan ini dapat berupa perubahan aplikasi dari *mainframe* ke lingkungan *client/ server* atau mengkonversi dari sistem berbasis berkas ke lingkungan basis data.

c. Pemeliharaan Korektif

Pemeliharaan korektif berupa pembetulan atas kesalahan-kesalahan yang ditemukan pada saat sistem berjalan.

Sebelum program diterapkan, maka program harus bebas terlebih dahulu dari kesalahan-kesalahan. Oleh sebab itu program harus diuji untuk menemukan kesalahan-kesalahan yang mungkin dapat terjadi. Program ditest untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai. Kesalahan dari program yang mungkin terjadi dapat diklasifikasikan dalam tiga bentuk kesalahan, yaitu sebagai berikut:

1. Kesalahan bahasa (*language errors*) atau disebut juga dengan kesalahan penulisan (*syntax errors*) atau kesalahan tata bahasa (*grammatical errors*) adalah kesalahan didalam penulisan *source program* yang tidak sesuai dengan yang telah disyaratkan. Kesalahan ini relatif mudah ditemukan dan diperbaiki, karena kompiler akan memberitahukan letak dan sebab kesalahannya sewaktu program dikompilasi.
2. Kesalahan sewaktu proses (*run time errors*), adalah kesalahan yang terjadi sewaktu *executable program* dijalankan. Kesalahan ini akan

menyebabkan proses program berhenti sebelum selesai pada saatnya, karena kompiler menemukan kondisi-kondisi yang belum terpenuhi yang tidak bisa dikerjakan. Kesalahan ini juga relatif mudah ditemukan, karena juga ditunjukkan letak serta sebab kesalahannya.

Kesalahan Logika (*logical errors*), adalah kesalahan dari logika program yang dibuat. Kesalahan seperti ini sulit ditemukan, karena tidak ada pemberitahuan mengenai kesalahannya dan tetap akan didapatkan hasil dari proses program, tetapi hasilnya salah. Kesalahan seperti ini merupakan kesalahan yang berbahaya karena bila tidak disadari dan tidak ditemukan, hasil yang salah dapat menyesatkan bagi yang menggunakannya. Cara mencari kesalahan logika dapat dilakukan dengan *test data*, yaitu dengan menjalankan program dengan menggunakan data tertentu dan membandingkan hasil pengolahannya dengan hasil yang sudah diketahui. Bila hasilnya berbeda, berarti mengalami kesalahan dan harus dilacak serta ditemukan sebab-sebab kesalahannya. Proses melacak kesalahan ini dikenal dengan istilah mencari kutu (*debugging*). Hasil pelacakannya adalah didapatkan kutu tersebut (*bug* yang berarti penyebab kesalahannya).

Program dapat diuji untuk tiap-tiap modulnya dan dilanjutkan dengan pengujian untuk semua modul yang telah dirangkai. Dengan demikian terdapat tiga tingkat pengujian yang dilakukan, yaitu :

1. Pengujian modul

Pengujian untuk tiap-tiap modul program (dapat berupa program utama, sub routine, sub program) disebut dengan *stub testing*. Pengetesan suatu modul dapat saja dilakukan walaupun modul lainnya yang berhubungan dengannya belum ditulis. Hal ini mudah saja dilakukan, yaitu dengan cara mensimulasi modul yang dipanggil yang belum ditulis. Modul dipanggil yang disimulasi ini disebut dengan *stub*. Modul *stub* dapat berupa sub routine atau sub program yang tidak berisi dengan logika-logika program. Mungkin juga modul *stub* ini diisi dengan instruksi-instruksi yang akan mencetak parameter yang diterimanya untuk menunjukkan bahwa modul ini sudah dapat dipanggil dengan benar.

2. Pengujian Unit atau Pengujian Program

Setelah semua modul program selesai ditulis dan diuji secara independent sampai bebas dari kesalahan dan telah dirangkai menjadi satu unit program ini perlu diuji kembali. Pengetesan untuk ini disebut dengan *unit testing* atau *program testing* yang dimaksudkan untuk meyakinkan bahwa semua modul telah bekerja terintegrasi tanpa mengalami kesalahan.

3. Pengujian Sistem

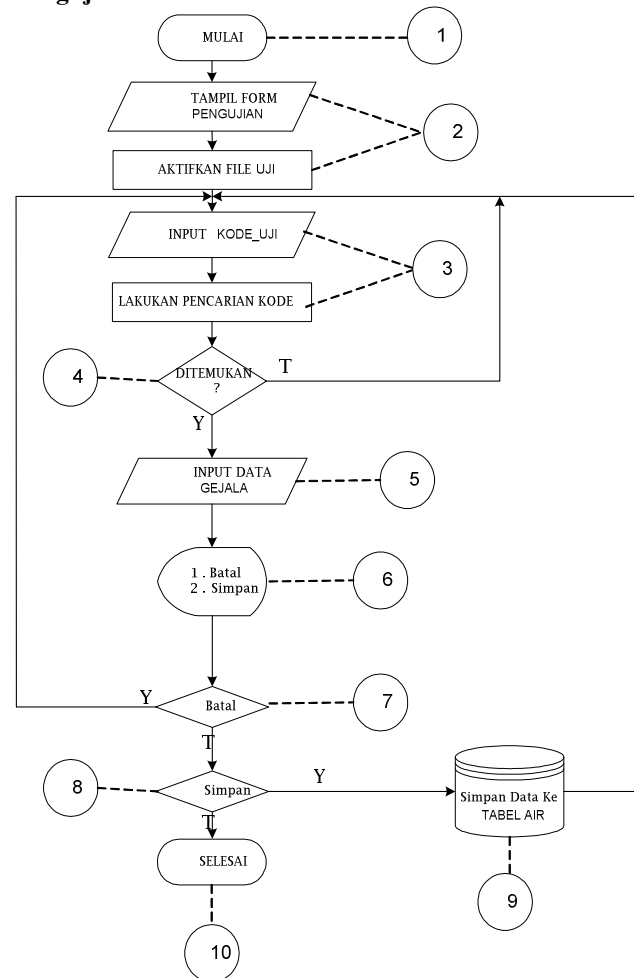
Pengujian sistem biasanya dilakukan setelah pengujian program. Pengujian sistem dilakukan untuk memeriksa kekompakan antar komponen sistem yang diimplementasikan. Tujuan utama dari pengujian sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengujian perlu dilakukan untuk mencari kesalahan-kesalahan atau kelemahan-kelemahan

yang mungkin terjadi. Pengujian sistem termasuk juga pengujian program secara menyeluruh. Pada pengujian program, masing-masing program yang telah berjalan dengan benar dan baik bukan berarti program tersebut juga akan dapat berjalan dengan program lainnya dalam sistem dengan baik. Kumpulan dari semua program yang telah diintegrasikan perlu diuji kembali untuk melihat apakah suatu program dapat menerima input data dengan baik, dapat memprosesnya dengan baik dan dapat memberikan output kepada program yang lainnya

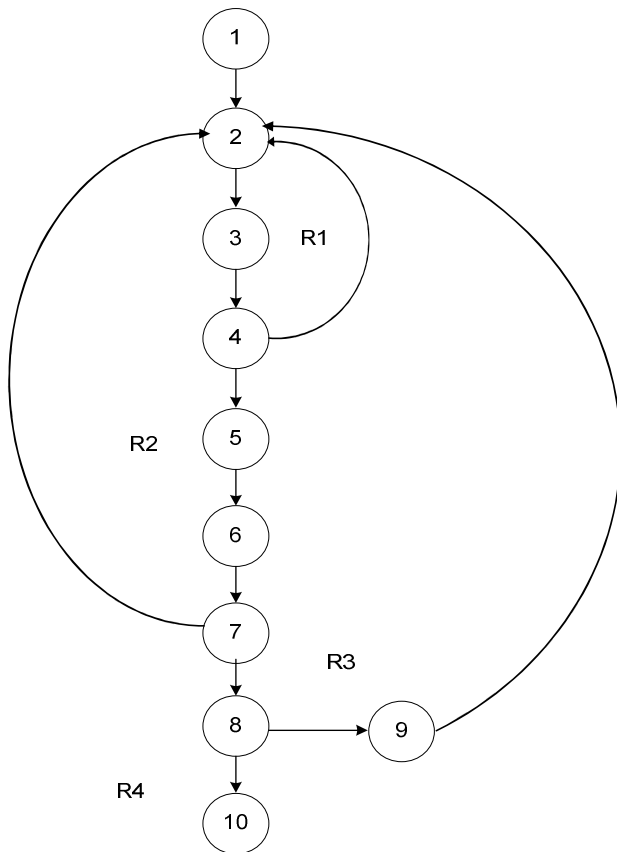
IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Sistem

Pengujian White Box



Gambar 7. Flowchart Form Pengujian Air



Gambar 6. Flowgraph Pengujian

Dari flowgraph diatas, maka diperoleh :

- Region (R) = 4
- Node (N) = 10
- Edge (E) = 12
- Predicate Node (P) = 3

Maka,  $V(G) = E - N + 2$   
 $= 12 - 10 + 2$   
 $= 4$

Maka,  $V(G) = P + 1$   
 $= 3 + 1$   
 $= 4$

Dan, Cyclomatic Complexity = R1, R2, R3, R4 = 4

Kemudian hasil diatas kita bandingkan dengan hasil pengujian *bases path*

Pengujian Jalur Logika pada flowgraph di atas adalah sebagai berikut :

Jadi Cyclomatic Complexity (CC) = 4, V(G) = 4

- Jalur 1 = 1 - 2 - 3 - 4 - 2 ...
- Jalur 2 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 2...
- Jalur 3 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 2...
- Jalur 4 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10

**B. Pembahasan**

**Deskripsi Kebutuhan Hardware /Software**

Hardware yang dibutuhkan oleh sistem informasi ini adalah cukup dengan menggunakan Processor Setara Pentium Dual Core, Memory 1 GB, Hardisk 80 GB atau

lebih, Monitor SVGA dengan Resolusi 1024 x 768 (disarankan), Peralatan Input / Output lainnya.

Adapun software yang di gunakan untuk membangun Sistem Pakar yaitu :

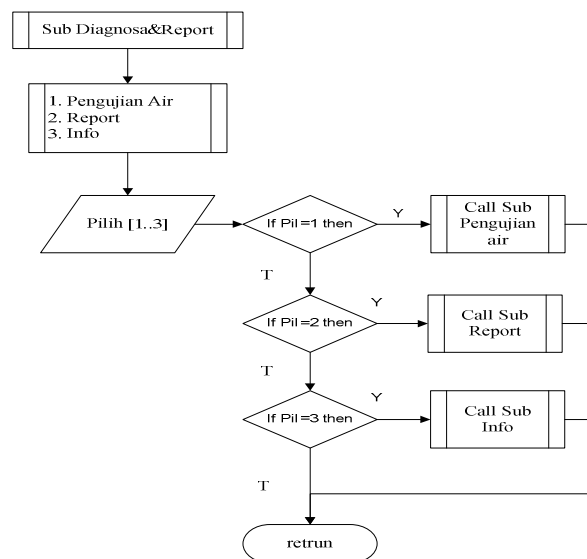
1. Sistem Operasi Windows /XP /windows 7.
2. Program aplikasi

Program aplikasi yang di gunakan pengetikan kode sumber (*source code*) dan untuk merancang *Sistem Pakar* adalah Borland Delphi 7.0 yang menyediakan kemampuan profesional untuk tiap-tiap aspek pengarah pengembangan Aplikasi Sistem Pakar, menggabungkan semua dalam satu paket membangun sistem dengan menciptakan isi yang interaktif.

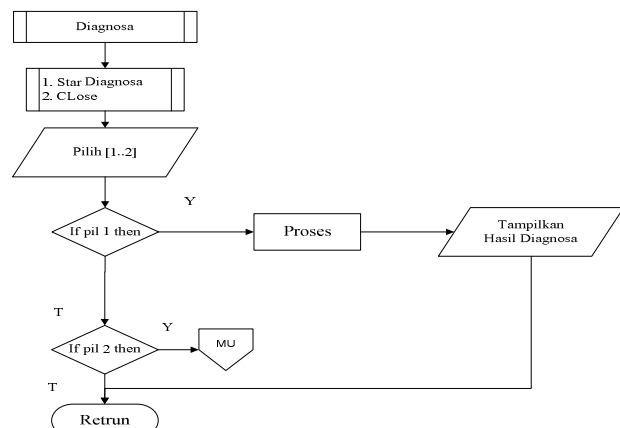
**3. Database**

Merupakan penampung data pada saat penginputan data produk-produk dan data pesanan barang. Dalam hal ini di gunakan Access 2007 dan MySQL merupakan *database server* yang paling populer saat ini dalam membangun Aplikasi Database.

**Flowchart Sub Deteksi**



Gambar 8. Flowchart Form Menu Utama (Sub Diagnosa & Report)



Gambar 9. Flowchart Deteksi

C. Implementasi

Form Menu Utama



Gambar.10 Form Menu Utama

Pada gambar form menu Utama di atas terdapat dua sub menu yakni file dan Diagnosa & Reoport, pada sub file terdapat :

1. Button Jenis Parameter (*Pemanggilan Form Jenis Parameter*).
2. Button Satuan (*Pemanggilan Form Satuan*).
3. Button Parameter (*Pemanggilan Form Parameter*).
4. Button Solusi (*Pemanggilan Form Solusi*).
5. Button Jenis Air (*Pemanggilan Form Jenis Air*).
6. Button User (*Pemanggilan Form User*).
7. Button Login (*Pemanggilan Form Login*).

Pada Sub Diagnosa digunakan untuk pemanggilan form pengujian dan report program.

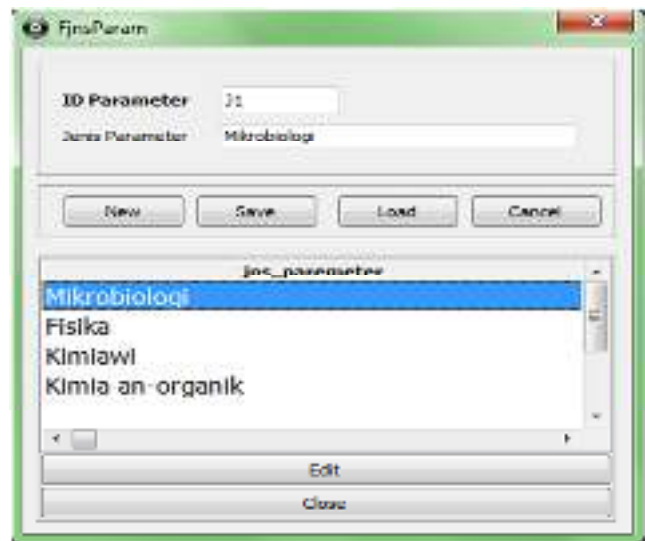
Form Login



Gambar 11. Form Login

Dapat kita lihat pada gambar form di atas Terdapat dua buah radio yang pertama untuk login sebagai admin yang kedua untuk login sebagai user biasa.

Form Jenis Parameter



Gambar 12. Form Jenis Parameter

Implementasi form jenis parameter adalah interface yang berfungsi untuk mengolah data jenis parameter pada gamabr di atas terlihat beberapa tombol.

Tombol New berfungsi untuk penginputan jenis parameter baru, sedangkan button edit digunakan untuk mengubah jenis parameter, button simpan berfungsi untuk melakukan penyimpanan baru ataupun penyimpanan dari hasil pengeditan, tombol cancel digunakan untuk melakukan pembatasan eksekusi sedangkan tombol load digunakan untuk melihat data-data yang terdapat pada tabel jenis parameter.

Form Parameter

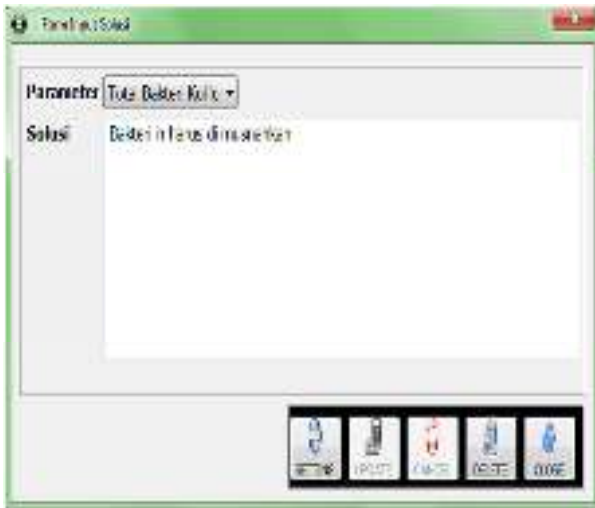


Gambar 13. Form Parameter

Gambar di atas adalah implementasi dari form input parameter. Pda gambar di atas terdapat button yang mempunyai fungsi sama pada form sebelumnya. Tapi hal yang baru di sini adalah sebuah pemilihan golongan lewat radio button. Ini berfungsi untuk memberikan golongan

kadar apakah dia berupa angka atau string. yang dalam hal ini menggunakan parameter huruf Y (ya) atau T(tidak). Kemudian pada pemilihan kategori ada dua yakni kategori satu dan kategori dua. Pada kategori satu kadar maksimum yang digunakan untuk hanya salah satu dari kadar maksimum, tetapi ketika kategori dua yang terpilih maka kadar maksimum yang berfungsi adalah kadar maksimum awal dan akhir.

**Form Solusi**



Gambar 14. Form Solusi

Untuk memberikan penjelasan dan penanganan pada kadar air yang tidak memenuhi syarat air yang dapat dikonsumsi maka form inilah yang berfungsi, pada form ini akan memberikan penjelasan penanganan atau solusi pada tiap parameter. Tombol yang terdapat adalah button Setting dan Update, pada saat button Setting yang dipilih maka akan diarahkan kepada penginputan pada memo solusi.

**Form User**



Gambar 15. Form User

Gambar di atas adalah interface dari form user yang berfungsi untuk mengolah data-data user, tombol-

tombol yang terdapa pada gambar di atas mempunyai fungsi sama pada form-form sebelumnya yang menggunakan button New, Save, Edit, Cancel, Delete dan Close.

**Form Jenis Air**



Gambar 16. Form Jenis Air

Implementasi dari form jenis air dapat kita lihat pada gambar di atas dengan langkah kerja sama pada form yang telah di bahas sebelumnya.

**Form Pengujian**



Gambar 17. Form Pengujian

Gambar di atas merupakan interface dari form Pengujian dengan menggunakan dua buah tombol yakni finish dan close.

Pada saat pemilihan jenis parameter maka tabel akan dikelompokkan sesuai jenis parameter yang dipilih, seperti yang terlihat pada gambar di atas jenis parameter yang dipilih adalah parameter fisika, maka



secara langsung sistem hanya akan menampilkan parameter yang tergolong pada jenis parameter fisika, kemudia user diharapkan mengisi kadar sesuai sampel air yang telah di ambil.setelah semua telah tersi dan set focus ke button finish, maka aka nada message yang memberiakn informasi apakan ingin mendiagnosa atau tidak.

### Form Hasil Deteksi



Gambar 18. Form Hasil Deteksi

Interface diatas adalah form yang akan meberikan informasi apakah kadar dari air memenuhi syarat ataupun tidak, ketika pada field memenuhi bersi Ya setelah klik button star diagnose maka aka nada informasi pada kotak samping kanan bahwa air tersebut layak untuk dikonsumsi. Begitupusn sebaliknya jika pada record memenuhi terdapat kata tidak maka akan diinformasikan mengenai parameter yang bermasalah kemudian cara penanganannya.

### V. KESIMPULAN

Setelah pengumpulan data, penganalisaan data kemudian perancangan sistem maka terbentuklah sistem pakar untuk memeriksa kualitas air apa layak dikonsumsi atau tidak:

Dapat dismpulkan bahwa layak tidaknya air dikonsumsi mempunyai beberapa parameter pengujian, dan dalam hal ini kami tuankan dalam bentuk sebuah sistem pakar dengan menyimpan kadar

maksimum.sebagai parameter ukur dari kadar sampel air yang akan di uji.

Hal ini dibuktikan dengan hasil pengujian yang dilakukan dengan metode *white box testing* dan *bases path testing* yang menghasilkan nilai  $V(G) = CC$ , dimana  $V(G) = 4$  dan  $CC = 4$ , sehingga didapat bahwa logika *flowchart* perhitungan vektor S dan Vektor V serta perankingan adalah benar dan berdasarkan pengujian *black box* yang meliputi uji *input* proses dan *output* dengan mengacu pada rancangan perangkat lunak yang sudah dibuat telah terpenuhi dengan hasil sesuai dengan rancangan.

Kami harapkan agar sekiranya tugas-tugas praktikum diarahkan kepada pembuatan sistem yang yang lebih besar dan dapat digunakan oleh masyarakat. Dengan Adanya sistem pakar ini maka kami sebagai penulis berharap agar sekiranya sistem ini dapat dikembangkan oleh mahasiswa yang nantinya mengajukan judul tentang sistem pakar.

### DAFTAR PUSTAKA

- [1]. Anonium, Pengolahan Air Gambut (Bandung:Puslitbang Pemukiman,DPU 1982) .
- [2]. Depertemen Kesehatan RI,Peraturan Menteri Kesehatan Republik Indonesia, Nomor : 01/Bir/Humas/1/1975 (Jakarta: 1975). EW Steelm,*Water Supply and Sewerage* (New York-Toronto-London: Mc.Garw Hill Book Co.Inc,1960).
- [3]. Jaja Jamaludin Malik, Mudah Belajar Membuat Aplikasi Pemrograman Delphi 2010. Andi Yogyakarta, 2010
- [4]. Kusrini, S.Kom. *Sistem Pakar Teori Dan Aplikasi*. Andi Yogyakarta, 2006.
- [5]. Muhammad Arhami. *Konsep dasar Sistem Pakar* . Andi Yogyakarta, 2005.
- [6]. Sri Kusumadewi. *Artificial Intelligenci (Teknik dan Aplikasinya)*. Yogyakarta; Graha Ilmu, 2003.
- [7]. Jaja Jamaludin Malik. *Mudah Belajar Membuat Aplikasi Pemrograman Delphi 2010*. Andi Yogyakarta, 2010.
- [8]. Kristianto, Harianto, *Konsep dan Perancangan Database*. Andi, Yogyakarta, 2002.
- [9]. Abdul Kadir, *Konsep & Tuntunan Praktis Basis Data*. Andi Yogyakarta, 2003.
- [10]. Fathul Wahid, *Dasar-dasar Algoritma & Pemrograman*. Andi Yogyakarta, 2004.
- [11]. Jaja Jamaludin Malik, *Mudah Belajar Membuat Aplikasi Pemrograman Delphi 2010*. Andi Yogyakarta, 2010.