

# Perancangan Aplikasi Tes IQ Siswa untuk Pertimbangan Pemilihan Jurusan dengan Metode Forward Chaining

## *IQ Test Design Student Application for Consideration Election Department of Using Forward Chaining*

Ahyuna\*<sup>1</sup>, Irmawati<sup>2</sup>

<sup>1,2</sup>STMIK Dipanegara Makassar

E-mail: \*<sup>1</sup> [sakuraabadi2013@gmail.com](mailto:sakuraabadi2013@gmail.com), <sup>2</sup> [faizirmawati@gmail.com](mailto:faizirmawati@gmail.com)

### **Abstrak**

*IQ merupakan pengelompokan kecerdasan manusia, untuk mengetahui kecerdasan seseorang berbagai macam alat ukur yang digunakan diantaranya adalah tes inteligensi, tes emosional, tes spiritual, psikotes dan alat ukur lainnya. Semakin tinggi semangat orang untuk meraih sukses, semakin tinggi pula kebutuhan akan modal intelektual, emosional, psikologis, minat dan bakat. Segala kemampuan tersebut bisa jadi merupakan bakat terpendam dalam diri yang belum tergal, ataupun bakat yang harus dimunculkan dengan berbagai latihan. Tes IQ dalam penentuan dan pertimbangan pemilihan jurusan pada sekolah biasanya dilakukan secara manual dan membutuhkan waktu yang cukup banyak untuk mengetahui hasil IQ dalam penentuan pemilihan jurusan, selain itu banyak peserta tes sulit memahami untuk menghitung hasil tes IQ karena cukup rumit sehingga untuk mempermudah pihak sekolah dalam melaksanakan tes IQ penentuan jurusan maka diperlukan adanya suatu aplikasi yang berbasis komputer dan untuk membangun aplikasi yang akan digunakan bahasa pemrograman serta metode pengujian white box. Hasil evaluasi terhadap aplikasi yang dibangun sangat membantu serta mempercepat pihak sekolah dalam penentuan jurusan dan tepat dalam penentuan jurusan siswa yang disesuaikan kemampuan mereka serta membudayakan pemakaian komputer di kalangan siswa.*

**Kata Kunci** — Tes IQ, Penentuan Jurusan

### **Abstract**

*IQ is a grouping of human intelligence, to determine a person's intellect wide range of measuring instruments used include intelligence tests, tests of emotional, spiritual tests, psychological and other measuring devices. The higher the spirit of people to achieve success, the higher the need for intellectual capital, emotional, psychological, interests and talents. All these capabilities can be a hidden talent within untapped, or talent that should be raised with the various exercises. IQ tests in the determination and consideration of election department at school is usually done manually and requires considerable time to know the results of IQ in determining the selection of majors, besides many test-takers is difficult to understand to calculate the results of an IQ test because it is quite complicated so as to facilitate the school in IQ tests determining the department implement the necessary existence of a computer-based applications and to build applications that will use the programming language as well as white-box testing methods. Results of evaluation of the applications that are built very helpful as well as accelerate the schools in the determination of the exact majors and students majoring in determining adjusted their abilities and cultivate the use of computers among students.*

**Keywords** — IQ tests, Determination Department

## 1. PENDAHULUAN

Inteligensi merupakan salah satu modal penting untuk meraih kesuksesan, sedangkan tes inteligensi dimaksudkan untuk mengetahui tingkat kecerdasan seseorang. Kecerdasan sangat penting bagi manusia karena langsung berkaitan dengan ketepatan pemahaman terhadap tugas, kewajiban, hak, wewenang, dan pengambilan keputusan. Inteligensi adalah kemampuan untuk bertindak secara terarah, berpikir secara rasional, dan menghadapi lingkungan secara efektif.

Tes IQ banyak digunakan oleh berbagai kalangan, misalnya pihak sekolah di SMA dalam menentukan jurusan. Hal ini dilakukan karena kemampuan IQ siswa berbeda-beda pada tingkatannya. Tes IQ yang biasa dilakukan adalah secara manual yang biasanya membutuhkan waktu yang cukup banyak untuk mengetahui hasil IQ, selain itu banyak peserta tes sulit memahami untuk menghitung hasil tes IQ karena cukup rumit.

Dengan melakukan diskusi secara langsung bersama calon mitra yaitu para kepala Sekolah dan guru-guru, maka ditemukanlah prioritas permasalahan yang sangat mendasar dimana guru-guru mendapat kesulitan dalam pemilihan jurusan sesuai kemampuan siswa. Guru hanya cenderung melihat siswa dari kesehariannya saja sehingga dalam pemilihan jurusannya terkadang kurang tepat, akibatnya sering terjadisiswa salah pilih jurusan sesuai dengan keinginan dan IQ nya. Olehkarena itu, dengan semakin berkembangnya teknologi maka mitra, dalam hal ini kami ingin mengembangkan metode pemilihan jurusan berbasis komputer yang diuji berdasarkan tes IQ mereka dan bagaimana merancang sebuah sistem aplikasi yang memindahkan tes IQ secara manual ke dalam sistem digital atau berbasis komputer.

Penelitian yang relevan dengan penelitian ini adalah dari kurnianingsih (2009) yang berjudul Aplikasi Tes IQ menggunakan Teknologi Mobile menggunakan Waterfall Model sebagai metode pengembangan sistemnya. Hasil dari penelitian ini adalah menghasilkan kuro IQ yang dapat berperan pada emulator Eclipse dengan target AVD Android 2. Éclair dan pada smartphone Android dengan operating sistem android 2.3 Gingerbrad. Kuro IG memudahkan penggunaan untuk mengetahui tingkat kecerdasan tanpa harus melakukan tes tertulis.

Penelitian kedua adalah penelitian dari wawan wardina dan viska veronica tobing (2012) yang berjudul aplikasi system pakar tes kepribadian berbasis web. Penelitian ini menghasilkan aplikasi yang menggunakan metodologi berorientasi objek dengan pemodelan visual Unified Modelling Language (UML). Perangkat ini dibuat untuk mengetahui kepribadian individu sehingga dapat membantu dalam mengembangkan kepribadianya.

Dari keseluruhan penelitian di atas masing-masing memberikan kontribusi dalam pengembangan tes IQ baik dalam bidang akademisi siswa SMA, pemanfaatan smartphone android, dan pemanfaatan tes IQ dalam pengembangan tes kepribadian. Ketiga topik tersebut menggunakan metode yang berbeda tetapi tujuannya searah ke pengoptimalan pelaksanaan tes IQ dari yang dulunya lebih lambat karena bersifat konvensional menjadi lebih terkomputerisasi.

### 1.1. Konsep jenis Tes IQ

#### 1.1.1. Berdasarkan Kelompok

Jenis tes IQ berdasarkan kelompok terdiri dari: tes informasi, tes pengertian, tes hitungan, tes kemiripan, tes rentan angka, tes perbendaharaan kata.

#### 1.1.2. Berdasarkan Sifat Pertanyaan

Pertanyaan yang bisa di sajikan dalam suatu tes IQ terdiri dari:

1. Tes dengan pertanyaan terbuka, maksudnya pertanyaan yang mempunyai kemungkinan jawaban lebih dari satu.
  2. Tes dengan pertanyaan tertutup, maksudnya pertanyaan yang memberi jumlah kemungkinan jawaban yang terbatas (biasanya hanya satu jawaban pasti).
-

### 1.1.3. Perbedaan Tes IQ dengan Tes TPA

Tes IQ merupakan bentuk pemeriksaan psikologis yang tujuannya hanya untuk mengukur tingkat kecerdasan seseorang dan mengetahui pada tingkatan mana seseorang itu berada apakah superior, cerdas atau biasa saja. Sedangkan Tes Potensi Akademik (TPA) adalah sebuah tes yang bertujuan untuk mengetahui bakat dan kemampuan seseorang di bidang keilmuan (akademis). Saat ini, TPA telah menjadi tes standar penyaringan Calon Pegawai Negeri Sipil (CPNS), rekrutmen karyawan swasta, serta karyawan BUMN. Tes Potensi Akademik juga umum dipakai sebagai tes penerimaan siswa untuk jenjang S2 dan S3.

### 1.1.4. Klasifikasi Tingkat IQ

Klasifikasi tingkat IQ yang di gunakan dalam penelitian ini adalah sebagai berikut:

Tabel 1. Klasifikasi Tingkat IQ

IQ	Klasifikasi
$\geq 140$	Sangat Superior
120 – 139	Superior
110 – 119	Di atas rata-rata
90 – 109	Rata-rata
80 – 89	Di bawah rata-rata
60 – 79	Lambat belajar
$\leq 59$	Keterbelakangan mental

## 1.2. Unified Modeling Language

Menurut Prabowo Pudjo Widodo dan Herlawati (2011:6) UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

Bagian-Bagian UML:

#### 1. View

*View* digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain: *use case view*, *logical view*, *component view*, *concurrency view*, dan *deployment view*.

#### 2. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external *actor*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya.

#### 3. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.

#### 4. Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administratif lainnya.

### 5. *Concurrency view*

Membagi sistem ke dalam proses dan prosesor. View ini digambarkan dalam diagram dinamis (*state, sequence, collaboration, dan activity diagrams*) dan diagram implementasi (*component dan deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasian (*integrator*), dan pengujian (*tester*).

### 6. *Deployment view*

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya.

### 7. *Diagram*

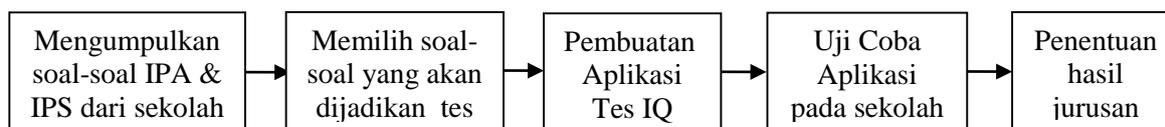
Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu.

## 1.3. *Lingkup Penelitian*

Penelitian ini dapat dilakukan di sekolah menengah atas (SMA) dengan pertimbangan setelah tahun pertama akan dilanjutkan dengan suatu jurusan untuk tahun berikutnya. Objek dalam penelitian ini adalah mengumpulkan pertanyaan-pertanyaan dari beberapa literatur untuk dijadikan soal dalam latihan tes IQ.

## 2. METODE PENELITIAN

### 2.1. *Alur penelitian*



Gambar 1. Alur Penelitian

Berdasarkan alur penelitian di atas dapat dilihat kontribusi penelitian ini yaitu dengan menggunakan metode *Forward chaining* dimana prinsip dasar *Forward Chaining* (Runut Maju) Menurut John Fredrik Ulysses (2012:5), metode ini melakukan pemrosesan berawal dari sekumpulan data untuk kemudian dilakukan inferensi sesuai dengan aturan yang diterapkan hingga diketemukan kesimpulan yang optimal dan selain itu menggunakan teknik pengujian *whitebox* sehingga dapat diketahui dan diperoleh keakuratan hasil tes siswa dalam penentuan jurusan [4].

Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan ke depan adalah *data-driven* yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Berikut adalah diagram *Forward Chaining* secara umum untuk menghasilkan sebuah *goal*. *Forward chaining* merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai TRUE), maka proses akan menyatakan konklusi. *Forward chaining* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan tree yang lebar dan tidak dalam, maka gunakan *forward chaining*.

Tipe sistem yang dapat dicari dengan *Forward Chaining*:

1. Sistem yang dipersentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari rule-rule dalam knowledge base untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian IF

3. Setiap rule dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian THEN. Kondisi baru ini ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi baru dari konklusi yang diminta, sistem akan kembali ke langkah 2 dan mencari rule-rule dalam knowledge base kembali. Jika tidak ada konklusi baru, sesi ini berakhir.

Contoh :

Terdapat 10 aturan yang tersimpan dalam basis pengetahuan yaitu :

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

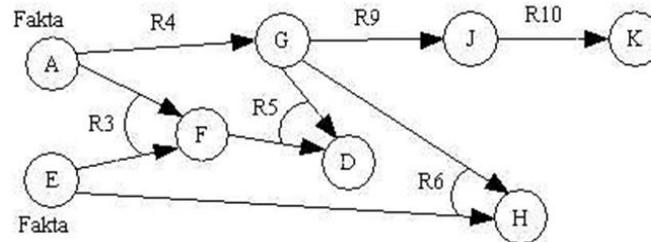
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran forward chaining terlihat pada gambar di bawah:

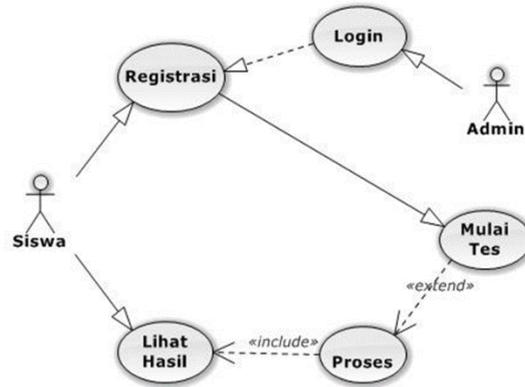


Gambar 2. Proses Penalaran Forward Chaining

2.1.1. Analisa Sistem yang Diajukan

Pada analisa yang diajukan disini seorang aktor (siswa) melakukan tes secara komputerisasi, caranya dengan melakukan registrasi sebagai peserta setelah itu maka siswa baru dapat login dan memulai tesnya secara online, hasil dari tes tersebut dapat diproses dan diperlihatkan secara langsung. (Prabowo Pudjo Widodo dan Herlawati. 2011) [5].

Use Case dari sistem yang diajukan:



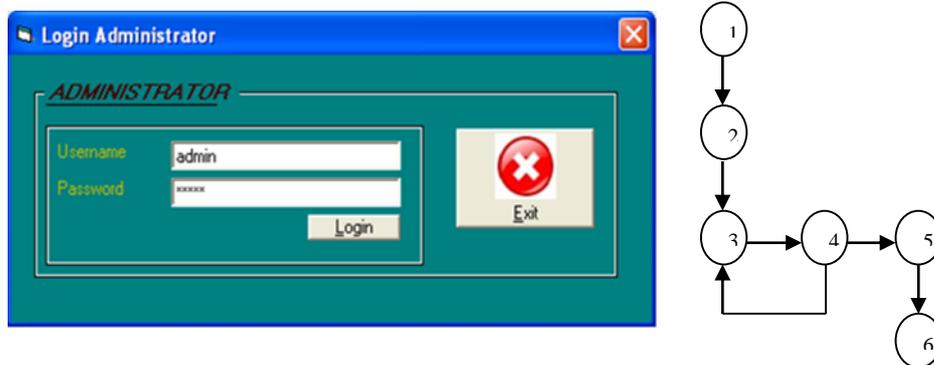
Gambar 3. Use CaseDiagram yang Diajukan

3. HASIL DAN PEMBAHASAN

3.1. Tampilan Hasil Input dan Output Aplikasi yang Telah Dirancang

3.1.1. Login Administrator

Digunakan untuk login admin agar dapat mengakses soal-soal dan hasil tes.



Gambar 4. Tampilan Login Admin

Perhitungan Cyclomatic Complexity dari Flowgraph di atas memiliki :

Region = 2

Menghitung Cyclomatic Complexity dari Egde dan Node

Dengan Rumus :  $V(G) = E - N + 2$

Dimana : E (jumlah edge pada flowgraph) = 6

N (jumlah node pada flowgarph) = 6

Penyelesaian :  $V(G) = 6 - 6 + 2$   $V(G) = 2$

Menghitung Cyclomatic Complexity dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus

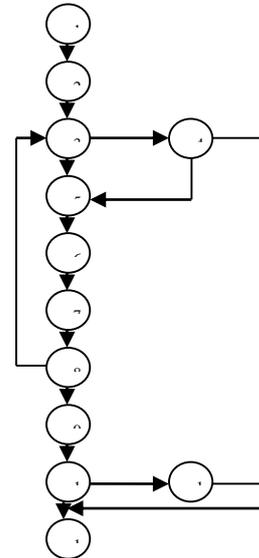
$V(G) = P + 1$  dimana  $P = 1$

Penyelesaian :  $V(G) = 1 + 1 \quad V(G) = 2$   
*Independent Path* pada *flowgarph* di atas adalah :  
 Path 1 = 1-2-3-4-3  
 Path 2 = 1-2-3-4-5-6

### 3.1.2. Registrasi Siswa

Registrasi siswa digunakan untuk meregistrasi data-data dari siswa yang akan ujian sebelum memulai tes.

Gambar 5. Tampilan Registrasi Siswa



Perhitungan *Cyclomatic Complexity* dari *Flowgraph* di atas memiliki :

Region = 4

Menghitung *Cyclomatic Complexity* dari *Egde* dan *Node*

Dengan Rumus :  $V(G) = E - N + 2$

Dimana :  $E$  (jumlah *edge* pada *flowgraph*) = 14

$N$  (jumlah *node* pada *flowgraph*) = 12

Penyelesaian :  $V(G) = 14 - 12 + 2$   
 $V(G) = 4$

Menghitung *Cyclomatic Complexity* dari *P*

*P* adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus  $V(G) = P + 1$  dimana  $P = 3$

Penyelesaian :  $V(G) = 3 + 1 \quad V(G) = 4$

*Independent Path* pada *flowgarph* di atas adalah :

Path 1 = 1-2-3-4-12

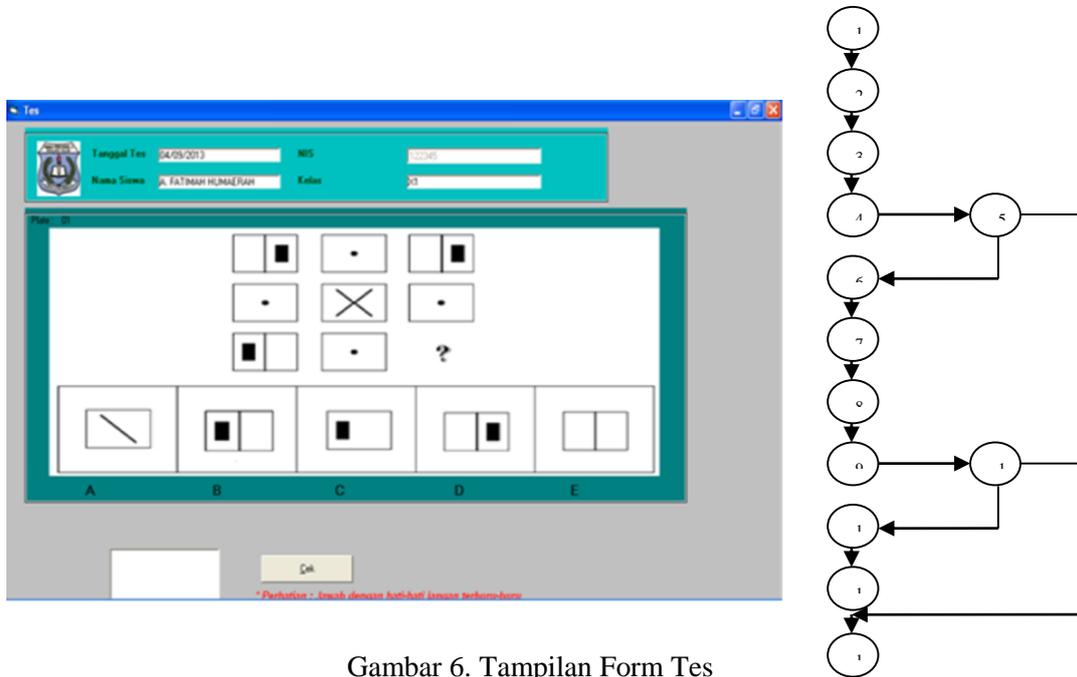
Path 2 = 1-2-3-4-5-6-7-8

Path 3 = 1-2-3-4-5-6-7-8-9-10-11-12

Path4 = 1-2-3-5-6-7-8-9-10-12

3.1.3. Form Tes

Pada form tes dibawah berupa tes-tes yang akan diujikan dimana nantinya akan mengasah kemampuan siswa.



Gambar 6. Tampilan Form Tes

Perhitungan *Cyclomatic Complexity* dari *Flowgraph* di atas memiliki:

Region = 3

Menghitung *Cyclomatic Complexity* dari *Egde* dan *Node*

Dengan Rumus :  $V(G) = E - N + 2$

Dimana : E (jumlah *edge* pada *flowgraph*) = 14

N (jumlah *node* pada *flowgarph*) = 13

Penyelesaian :  $V(G) = 14 - 13 + 2$   $V(G) = 3$

Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan

rumus  $V(G) = P + 1$  dimana P = 2

Penyelesaian :  $V(G) = 2 + 1$

$V(G) = 3$

*Independent Path* pada *flowgarph* di atas adalah :

Path 1 = 1-2-3-4-5-13

Path 2 = 1-2-3-4-5-6-7-8-9-10-11-12-13

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

### 3.1.4. Form Hasil Tes

Form hasil tes ini akan muncul jika sudah dilakukan tes dan memberikan nilai atau skor serta jurusan yang disarankan buat siswa. Untuk hasilnya tidak dimasukkan dalam flowgraph karena sudah memberikan hasil.

Tanggal Tes	04/09/2013	NIS	22345	
Nama Siswa	A. FaTmahan H.M.A.M. Fauzi		Kelas	X
Nilai IQ Anda	117			
Jurusan yang Disarankan	IPA			
Keterangan	Normal Ting			

Gambar 7. Tampilan Form Hasil Tes

## 3.2. Evaluasi Terhadap Aplikasi IQ yang Telah Terbangun Dalam Penelitian

### 3.2.3. Aspek Pengujian

Pengujian ini meliputi aspek Pengujian *white box* (*glass box*) adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian (Febriyanno Suryana:2011) [1].

### 3.2.4. Mekanisme uji coba

Mekanisme uji coba menggunakan langkah-langkah dari teknik pengujian basis path sebagai berikut :

1. Buat *flowchart* ke *flowgraph*.
2. Transformasi *flowchart* ke *flowgraph*.
3. Tentukan *Cyclomatic Complexity*, *Region* dan *Independent Path* dari *Flowgraph*.
4. Akan disimpulkan bahwa modul sudah bebas dari kesalahan logika jika nilai  $CC = R = Independent Path$ .

Dengan menggunakan metode pengujian *White-Box* perancang sistem dapat melakukan *test case* yang dapat

1. Memberikan jaminan bahwa semua jalur *independent* pada suatu modul telah digunakan paling tidak satu kali.
2. Menggunakan semua keputusan logis pada sisi *true and false*.
3. Mengeksekusi semua *loop* pada batasan masalah dan pada batasan operasional masalah.
4. Menggunakan struktur data internet untuk menjamin validitasnya.

### 3.2.5. Skenario uji coba

Skenario uji coba jika hasil CC (*Cyclomatic Complexity*), IP (*Independent Path*), R (*Region*) = bernilai sama berarti pengujian sistemnya sudah benar atau sesuai dengan alur logika (bebas dari kesalahan logika).

### 3.2.6. Hasil Uji coba

Tabel 2. Hasil Pengujian *White Box*

No	Nama bagan Alir Program	<i>Independent Path</i>	<i>Region</i>	<i>Cyclomatic Complexity</i>
1	<i>Flowgraph Login</i>	2	2	2
2	<i>Flowgraph Menu Utama</i>	2	2	2
3	<i>Flowgraph Registrasi</i>	4	4	4
4	<i>Flowgraph Tes</i>	3	3	3
	<b>TOTAL</b>	<b>11</b>	<b>11</b>	<b>11</b>

Kesimpulan dari rekapitulasi hasil perhitungan diatas didapatkan jumlah *Cyclomatic Complexity* (CC) = 11, *Region* = 11 dan *IndependentPath* = 11, karena jumlah ketiga parameter ini sama, maka dapat disimpulkan bahwa Aplikasi yang dibuat telah bebas dari kesalahan logika.

### 3.2.5. Analisis uji cobanya

Setelah melihat table pada hasil pengujian *White box* dapat diberi kesimpulan bahwa keseluruhan form input dan output sudah sesuai alur logika.

## 4. KESIMPULAN

Berdasarkan hasil analisis dan desain serta pengujian sistem maka dapat ditarik kesimpulan bahwa form tes yang dibuat pada aplikasi ini adalah form yang berisi tes-tes yang telah mengarahkan pada siswa, apakah cenderung lebih menguasai soal-soal eksak ataukah soal-soal eksakta. Dari tes inilah dapat disimpulkan seorang siswa itu lebih cocok berada di salah satu jurusan yang ada di sekolah tersebut serta aplikasi yang telah dibuat dinyatakan bebas dari kesalahan logika.

## 5. SARAN

Dalam sistem yang kami kembangkan pasti tidak terlepas dari beberapa kesalahan dan kekurangan, oleh karenanya disarankan bagi peneliti yang ingin mengembangkan sistem ini lebih lanjut diharapkan ke depannya dapat lebih memperhatikan nilai-nilai kriteria yang digunakan serta efektifitas dan efisiensi penggunaan program. Serta variasi soal yang di berikan dapat diacak nomor soalnya sehingga siswa yang diuji tidak saling menukar jawaban.

## 6. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada kampus tercinta kami STMIK Diponegara dan teman-teman yang membantu kami dalam penyelesaian penelitian ini.

## DAFTAR PUSTAKA

- [1] Suryana, F., 2011, Metode Pengujian Perangkat Lunak (White Box), <http://suryagsc.files.wordpress.com/2012/04/meeting-9-whitebox.ppt>, diakses tanggal 23 Desember 2014.
  - [2] Misbach, I. H., 2008, Antara IQ, EQ, dan SQ Pelatihan Nasional Guru Se-Indonesia, *Tesis*, Fakultas Ilmu Pendidikan, Universitas Pendidikan Indonesia, Bandung.
  - [3] Ulysses, J. F., 2012, Sistem Pakar “Metode Inferensi II-Forward Chaining & Backward Chaining”, Bahan Ajar pada STMIK Palangkaraya.
  - [4] Widodo, P. P., Herlawati., 2011, “*Menggunakan UML*”, Informatika, Bandung.
-