Sistem Pendukung Keputusan Pemilihan Jenis Usaha Menggunakan Metode AHP Berbasis Web

Arwansyah, Asrul Syam
STMIK Dipanegara Makassar
Jalan Perintis Kemerdekaan KM.9 Makassar, Telp.(0411)587194 – Fax (0411)588284
e-mail: arwansyah@dipanegara.ac.id, assyams03@gmail.com

Abstrak

Pembuatan sebuah bisnis atau usaha belum memperhatikan aspek –aspek pendukung keberhasilan sebuah usaha atau bisnis sehingga beberapa pemilik usaha atau bisnis tidak dapat melanjutkan usahanya dan bahkan mengalami kerugian. Pemanfaatan teknologi komputer dalam pemilihan jenis usaha memiliki beberapa kelebihan dibanding cara yang selama ini digunakan oleh masyarakat, salah satunya yaitu memberikan informasi yang lebih akurat karena menggunakan metode AHP dalam pemilihan keputusannya, serta memudahkan masyarakat dalam mengakses karena menggunakan flatform web. Informasi mengenai jenis usaha yang dihasilkan akan berbeda setiap user karena data-data yang diproses disesuaikan dengan user sehingga jenis usaha yang dibuat nantinya akan berbeda-beda dan membuat arus transaksi menjadi lebih baik

***Kata kunci**:* SPK, Jenis usaha, AHP

Abstract

Making a business or not pay attention to aspects of supporting the success of a business or a business that some business owners or businesses can not continue its business and even a loss. Use of computer technology in the selection of the type of business has several advantages over the way that has been used by the community, one of which is to provide information that is more accurate because it uses AHP in the selection decision, as well as facilitate the public in accessing the web because it uses a flat form. Information on the type of business that is generated will be different every user because the data are processed tailored to the user so that kind of effort made will be different and make the flow of transactions to be better.

***Keywords:** SPK, Type of Business, AHP*

1. Pendahuluan

Salah satu faktor penting dalam pertumbuhan ekonomi adalah menciptakan lapangan pekerjaan sehingga tingkat pengangguran menjadi lebih berkurang.Sumber utama dalam menciptakan lapangan pekerjaan adalah dengan membuka usaha atau bisnis yang nantinya dapat menyerap tenaga kerja.Pembukaan kegiatan usaha atau bisnis tidak hanya dapat dilakukan oleh perusahaan skala besar namun juga dapat dilakukan oleh individu pada suatu kelompok masyarakat yang memiliki tekad, motivasi, serta inovasi baik dalam bidang produk atau jasa yang dapat diterima oleh masyarakat.

Dalam membangun sebuah bisnis atau usaha perlu diperhatikan berbagai hal sehingga tingkat atau resiko kegagalan dapat diminimalisir sementara realita yang ada saat ini adalah pembuatan sebuah bisnis atau usaha belum memperhatikan aspek-aspek pendukung keberhasilan sebuah usaha atau bisnis sehingga beberapa pemilik usaha atau bisnis tidak dapat melanjutkan usahanya atau bahkan mengalami kerugian dalam waktu singkat. Salah satu solusi yang dapat dilakukan adalah memanfaatkan teknologi komputer dalam menghasilkan sebuah informasi yang dapat dijadikan sebagai acuan dalam pengambilan keputusan yang berkaitan dengan jenis usaha yang sebaiknya dibangun oleh masyarakat.

Pemanfaatan teknologi komputer dalam pemilihan jenis usaha memiliki beberapa keunggulan dibandingkancara yang selama ini digunakan oleh masyarakat dalam membuat sebuah usaha atau bisnis. Salah satu keunggulannya yakni informasi yang dihasilkan lebih akurat karena menggunakan metode tertentu, memudahkan setiap masyarakat dalam mengaksesnya karena dibuat menggunakan *flatform* web, dan informasi mengenai jenis usaha yang sebaiknya dibuat akan berbeda untuk setiap user, karena data-

data yang diproses sesuai dengan tipe user sehingga jenis usaha yang dibuat nantinya akan berbeda dan membuat arus transaksi menjadi lebih baik.

2. Bahan dan Metode

2.1. Pengertian Perancangan

Perancangan sistem adalah sebuah teknik pemecahan masalah yang saling melengkapi (dengan analisis sistem) yang merangkai kembali bagian-bagian komponen yang menjadi sistem yang lengkap. Hal ini melibatkan penambahan, penghapusan dan perubahan-perubahan bagian relatif pada sistem awal (aslinya)[2].

2.2. Perangkat lunak (*Software*)

Aplikasi merupakan suatu kelompok file (form, class, report) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, dimana ruang lingkup dari suatu aplikasi berbeda-beda dari satu perusahaan ke perusahaan lainnya.[3].

2.3. Pengertian Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu dan perangkat[4].

2.4. Sistem Penunjang Keputusan (SPK)

Sistem pendukung Keputusan diterjemahkan dari istilah DSS *(decision support system).*Istilah DSS diciptakan pada tahun 1971 oleh G. Anthony Gorry dan Michael S. Scott Morton untuk mengarahkan aplikasi komputer pada pengambilan keputusan manajemen. Keduanya adalah profesor dari MIT, yang kemudian bersama-sama menulis artikel dalam jumal yang berjudul *"A Framework for Management Information System"* mereka merasakan perlunya ada kerangka untuk menyalurkan aplikasi computer terhadap pembuatan keputusan manajemen. Secara harafiah, DSS *(decision support system)* diterjemahkan dalam bahasa Indonesia sebagai Sistem Pendukung Keputusan, dan dianggap berkaitan erat dengan pengertian sebagai Sistem infonnasi atau model analisis yang dirancang untuk membantu para pengambil keputusan dan para profesional agar mendapatkan data yang akurat berdasarkan data yang ada.

Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. Istilah SPK mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan. Sistem penunjang keputusan merupakan sistem berbasis komputer yang diharapkan dapat membantu menyelesaikan masalah-masalah yang komplek yang tidak terstruktur maupun yang semi terstruktur.Sistem Penunjang Keputusan merupakan perpaduan antara keahlian manusia dan juga komputer.Dengan kemampuan yang dimiliki, sistem penunjang keputusan diharapkan dapat membantu dalam pengambilan keputusan baik untuk masalah semi terstruktur maupun tidak terstruktur. [6]

2.5. Metode AHP

Metode AHP dikembangkan oleh Thomas L. Saaty, seorang ahli matematika. Metode ini adalah sebuah kerangka untuk mengambil keputusan dengan efektif atas persoalan yang kompleks dengan menyederhanakan dan mempercepat proses pengambilan keputusan dengan memecahkan persoalan tersebut kedalam bagian-bagiannya, menata bagian atau variabel ini dalam suatu susunan hirarki, member nilai numerik pada pertimbangan subjektif tentang pentingnya tiap variabel dan mensintesis berbagai pertimbangan ini untuk menetapkan variabel mana yang memiliki prioritas paling tinggi dan bertindak untuk mempengaruhi hasil pada situasi tersebut.

Metode AHP ini membantu memecahkan persoalan yang kompleks dengan menstruktur suatu hirarki kriteria, pihak yang berkepentingan, hasil dan dengan menarik berbagai pertimbangan guna mengembangkan bobot atau prioritas. Proses hierarki adalah suatu model yang memberikan kesempatan bagi perorangan atau kelompok untuk membangun gagasan-gagasan dan mendefinisikan persoalan dengan cara membuat asumsi mereka masing-masing dan memperoleh pemecahan yang diinginkan darinya. [6]

Metode "*pairwise comparison*" AHP mempunyai kemampuan untuk memecahkan masalah yang diteliti multi obyek dan multi kriteria yang berdasar pada perbandingan preferensi dari tiap elemen dalam

hierarki.Jadi model ini merupakan model yang komperehensif. Pembuat keputusan menetukan pilihan atas pasangan perbandingan yang sederhana, membengun semua prioritas untuk urutan alternatif. " *Pairwaise comparison*" AHP mwenggunakan data yang ada bersifat kualitatif berdasarkan pada persepsi, pengalaman, intuisi sehigga dirasakan dan diamati, namun kelengkapan data numerik tidak menunjang untuk memodelkan secara kuantitatif [10].

Kelebihan dalam Metode AHP

1. Struktur yang berhierarki sebagai konskwensi dari kriteria yang dipilih sampai pada sub-sub kriteria yang paling dalam.
2. Memperhitungkan validitas sampai batas toleransi inkonsentrasi sebagai kriteria dan alternatif yang dipilih oleh para pengambil keputusan.
3. Memperhitungkan daya tahan atau ketahanan output analisis sensitivitas pengambilan keputusan.

Kekurangan dalam Metode AHP

1. Ketergantungan model AHP pada input utamanya.
   Input utama ini berupa persepsi seorang ahli sehingga dalam hal ini melibatkan subyektifitas sang ahli selain itu juga model menjadi tidak berarti jika ahli tersebut memberikan penilaian yang keliru.
2. Metode AHP ini hanya metode matematis tanpa ada pengujian secara statistik sehingga tidak ada batas kepercayaan dari kebenaran model yang terbentuk

Langkah – langkah dan proses Analisis Hierarki Proses (AHP) adalah sebagai berikut :

1. Mendefinisikan permasalahan dan penentuan tujuan.
2. Menyusun masalah kedalam hirarki
3. Penyusunan prioritas untuk tiap elemen masalah pada hierarki..
4. Melakukan pengujian konsitensi terhadap perbandingan antar elemen

Setelah dilakukan pengolahan tersebut, maka dapat disimpulkan adanya konsitensi atau tidak ada konsistensi, bila data tidak konsisten maka diulangi lagi dengan pengambilan data seperti semula, namun bila sebaliknya maka digolongkan data terbobot yang selanjutnya dapat dicari nilai beta.

2.6. UML (Unified Modelling Language)

UML *(Unifield Modelling Language)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak.UML menawarkan sebuah standar untuk merancang model sebuah sistem. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram perangkat lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefenisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya yakni Grady Booch OOD *(Object-Oriented Design)*, Jim Rubaugh OMT *(Object Modelling Technique)*, dan Ivan Jacobson OOSE (*Object-Oriented Software Engineering*). Diagram UML terdiri dari:*Use Case* Diagram, *Class* Diagram, *Activity* Diagram, dan *Sequence* Diagram.[1]

2.7. White box testing

Pengujian *white box* adalah metode perancangan *test case* yang menggunakan struktur kontrol dari perancangan *procedural* untuk mendapatkan test case. Dengan menggunakan metode *white box*, analisis system akan dapat memperoleh *test case*yaitu :

1. Menjamin seluruh *independent path* didalam modul yang dikerjakan sekurang-kurang
2. Mengerjakan seluruh keputusan logical
3. Mengerjakan seluruh loop yang sesuai dengan batasannya
4. Mengerjakan seluruh struktur data internal yang menjamin *validitas.*

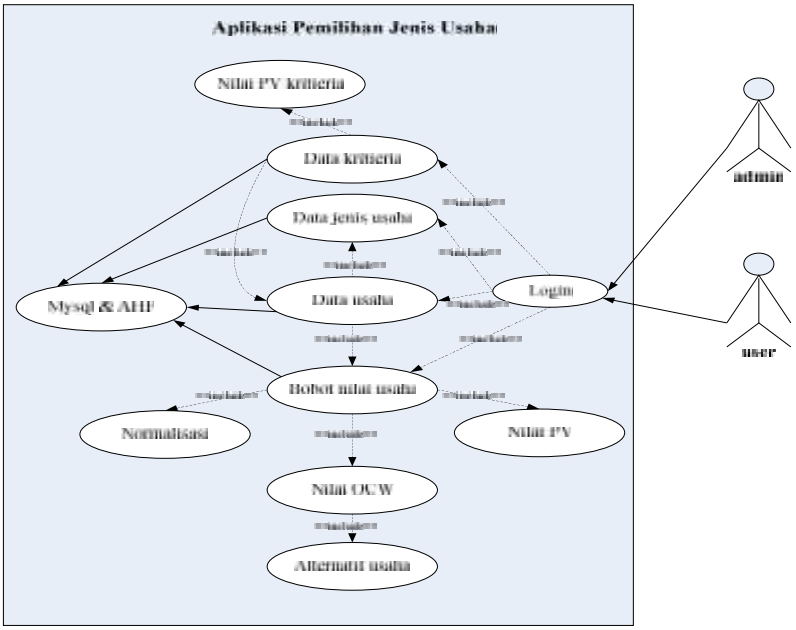
Pengujian white box dilakukan untuk menguji prosedur-prosedur yang ada. Lintasan lojik yang dilalui oleh setiap bagian prosedur diuji dengan memberikan kondisi/loop spesifik. Pengujian *white box* menjamin pengujian terhadap semua lintasan yang tidak bergantungan minimal satu kali, mencoba semua keputusan lojik dari sisi 'true' dan 'false', eksekusi semua loop dalam batasan kondisi dan batasan operasionalnya dan pengujian validasi data internal. [5]

3. Metode Perancangan

3.1.*Use Case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. [7]

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.*Use case* diagram dapat sangat membantu apabila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.*Use Case* untuk perangkat lunak yang akan dibangun dijelaskan pada gambar 3.1

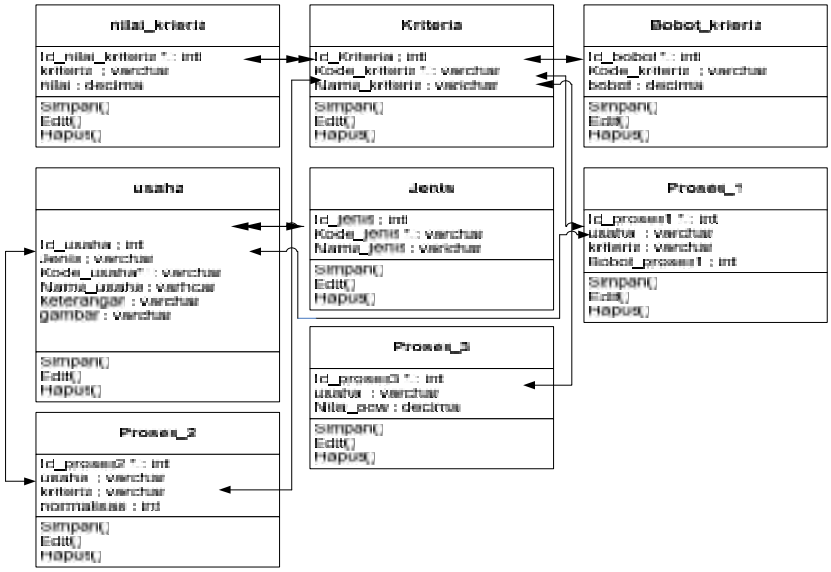


Gambar 3.1 *Use Case* Aplikasi Pemilihn Jenis Usaha

3.2. Diagram Kelas *(Class Diagram)*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. [1] Berikut penjelasan mengenai bentuk *class diagram* pada aplikasi yang dibangun dapat dilihat pada gambar 3.2

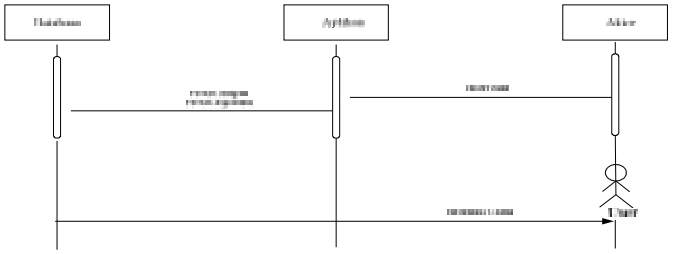
.



Gambar3.2 *Class Diagram* Aplikasi Pemilihn Jenis Usaha

3.3. Sequence Diagram

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu.Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message* [8]. *Sequence Diagram* untuk perangkat lunak yang akan dibangun dijelaskan pada gambar 3.3

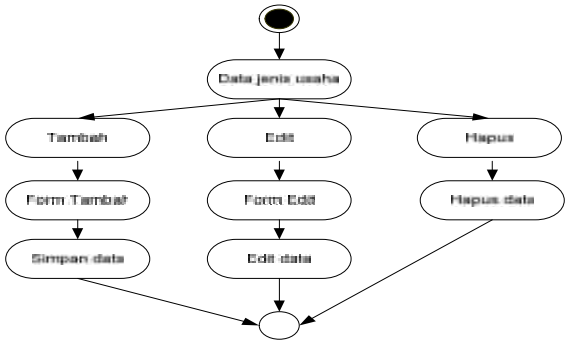


Gambar 3.3 Sequence Diagram Aplikasi

3.4. Diagram Aktivitas *(Activity* Diagram*)*

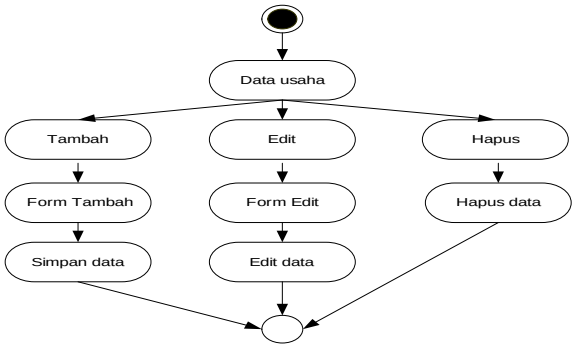
Activity diagram menggambarkan berbagai alir aktivitas dalam aplikasi yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi [9]. Berikut penjelasan mengenai beberapa bentuk *activity diagram* pada aplikasi yang dibangun

1. Activity diagram data jenis usaha



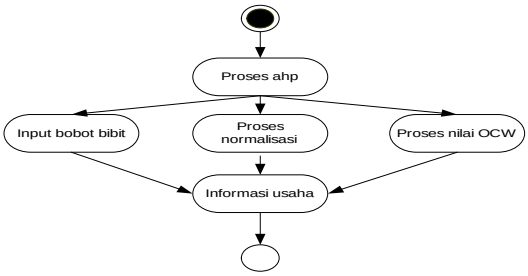
Gambar 3.6 Activity diagram data jenis usaha

2. Activity diagram data usaha



Gambar 3.7 Activity diagram data usaha

3. Activity diagram proses AHP



Gambar 3.8 Activity diagram proses AHP

4. Hasil dan Pembahasan

4.1. Rancangan Sistem

4.1.1 Rancangan Input

Input merupakan data yang masuk ke dalam sistem informasi, ini diperlukan ada karena bahan dasar dalam pengolahan informasi, Berikut ini adalah *interface* rancangan input ditunjukan pada gambar 4.1 sampai gambar 4.4

Gambar 4.1 Rancangan input data kriteria

Gambar 4.2Rancangan input data bobot

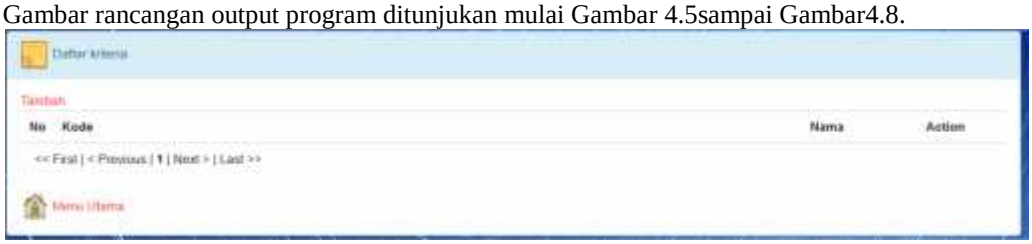
Gambar 4.3Rancangan input data jenis usaha

Gambar 4.4Rancangan input data usaha

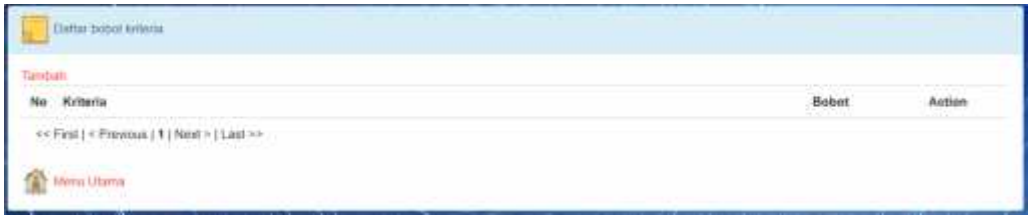
4.1.2 Rancangan Output

Output merupakan produk dari sistem informasi yang dapat dilihat. Output inidapat berupa hasil yang dikeluarkan di media keras (kertas dan lain-lain) dan output yang berupa hasil dikeluarkan ke media lunak (tampilan dilayar

Gambar rancangan output program ditunjukan mulai Gambar 4.5sampai Gambar4.8.



Gambar 4.5 Rancangan output data kriteria



Gambar 4.6 Rancangan output data bobot

Gambar 4.7 Rancangan output data jenis usaha

Gambar 4.8 Rancangan output data usaha

1.2. Pengujian White Box

Pengujian perangkat lunak yang dilakukan dengan menggunakan teknik *white box*, terlebih dahulu memetakan *flowchart* ke dalam *flowgraph* kemudian menghitungbesarnya jumlah *edge* dan *node* dimana jumlah *node* dan *edge* ini akan menentukanbesarnya *cyclomatic complexcity*. Adapun *flowgraph* dari *flowchart* sistem yang telahdibangun :

1. Flowgraph data kriteria

Dari *flowchart* kriteria yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut :

Notasi *flowgraph :*

1. Mulai
2. Tampilkan data kriteria
3. Masukkan pilihan
4. Jika pilihan 3, tambah data
5. Jika pilihan 10, edit data
6. Jika pilihan 15, hapus data
7. Jika pilihan tidak, proses batal dan looping
8. Kembali

Gambar 4.10 Flowgraph data kriteria

Keterangan :

Node (N) = 23

Edge (E) = 28

Predikat (P) = 6

a. Region (R) = 7

b. Cyclomatic complexcity

1. V (G) = (E – N) + 2
= (28 – 23) + 2
= 7

2. V (G) = P + 1
= 6 + 1
= 7

c. Independent Path

Path 1 : 1-2-3-2-3-4-5-2

Path 2 : 1-2-3-4-6-7-2

Path 3 : 1-2-3-4-6-8-9-21

Path 4 : 1-2-3-10-9-11-12-13-21

Path 5 : 1-2-3-10-14-15-16-17-18-21
Path 6 : 1-2-3-10-14-15-16-20-19
Path 7 : 1-2-3-10-14-19-21-22

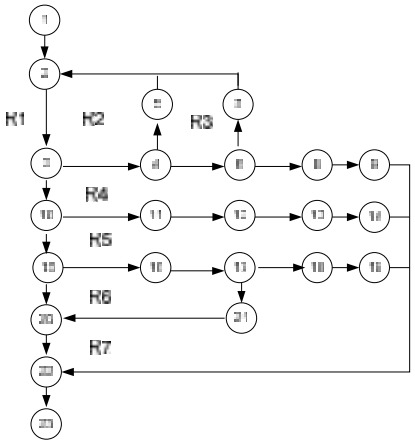
Berdasarkan hasil yang didapatkan dimana Region, Independentt Path dan Cyclomatic Complexcity bernilai sama. Maka dapat disimpulkan bahwa aplikasi yang dirancang dapat dikatakan bebas dari kesalahan logika

3. Flowgraph bobot

Dari *flowchart* bobot yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut :

Notasi *flowgraph :*

1. Mulai
2. Tampilkan data bobot
3. Masukkan pilihan
4. Jika pilihan 3, tambah data
5. Jika pilihan 10, edit data
6. Jika pilihan 15, hapus data
7. Jika pilihan tidak, proses batal dan looping
8. Kembali



Gambar 4.11 Flowgraph bobot

Keterangan :
Node (N) = 23
Edge (E) = 28
Predikat (P) = 6
a. Region (R) = 7
b. Cyclomatic complexcity

1. V (G) = (E – N) + 2
= (28 – 23) + 2
= 7

2. V (G) = P + 1
= 6 + 1
= 7

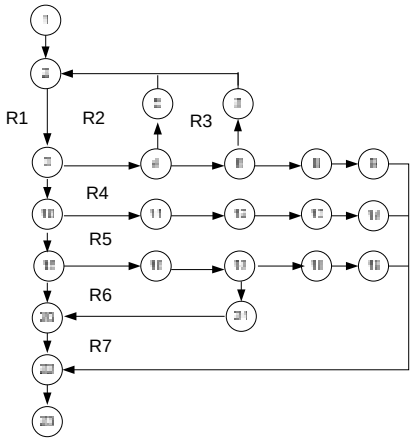
c. Independent Path
Path 1 : 1-2-3-2-3-4-5-2
Path 2 : 1-2-3-4-6-7-2
Path 3 : 1-2-3-4-6-8-9-21
Path 4 : 1-2-3-10-9-11-12-13-21
Path 5 : 1-2-3-10-14-15-16-17-18-21
Path 6 : 1-2-3-10-14-15-16-20-19
Path 7 : 1-2-3-10-14-19-21-22

Berdasarkan hasil yang didapatkan dimana Region, Independentt Path dan Cyclomatic Complexcity bernilai sama. Maka dapat disimpulkan bahwa aplikasi yang dirancang dapat dikatakan bebas dari kesalahan logika

4. Flowgraph data usaha

Dari *flowchart* usaha yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut :

Notasi *flowgraph :*
1. Mulai
2. Tampilkan data usaha
3. Masukkan pilihan
4. Jika pilihan 3, tambah data
5. Jika pilihan 10, edit data
6. Jika pilihan 15, hapus data
7. Jika pilihan tidak, proses batal dan looping
8. Kembali



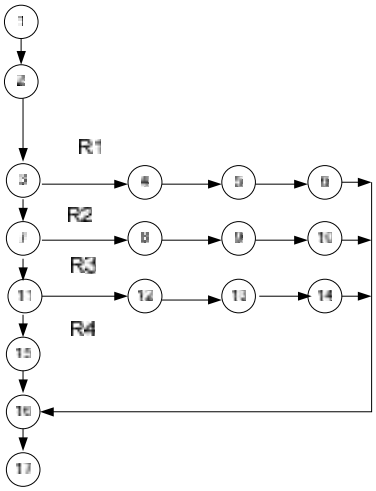
Gambar 4.12Flowgraph data usaha

Keterangan :
Node (N) = 23
Edge (E) = 28
Predikat (P) = 6
a. Region (R) = 7
b. Cyclomatic complexcity
1. V (G) = (E – N) + 2
= (28 – 23) + 2
= 7
2. V (G) = P + 1
= 6 + 1
= 7
c. Independent Path
Path 1 : 1-2-3-2-3-4-5-2
Path 2 : 1-2-3-4-6-7-2
Path 3 : 1-2-3-4-6-8-9-21
Path 4 : 1-2-3-10-9-11-12-13-21
Path 5 : 1-2-3-10-14-15-16-17-18-21
Path 6 : 1-2-3-10-14-15-16-20-19
Path 7 : 1-2-3-10-14-19-21-22

Berdasarkan hasil yang didapatkan dimana Region, Independentt Path dan Cyclomatic Complexcity bernilai sama. Maka dapat disimpulkan bahwa aplikasi yang dirancang dapat dikatakan bebas dari kesalahan logika

5. Flowgraph proses ahp

Dari *flowchart* laporan yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut :
Notasi *flowgraph :*
1. Mulai
2. Tampilkan pilihan
3. Masukkan pilihan
4. Jika pilihan 3, proses ahp
5. Jika pilihan 7, view data hasil proses ahp
6. Jika pilihan 11, hapus data hasil proses ahp
7. Jika pilihan tidak, proses batal dan looping
8. Kembali



Gambar 4.13Flowgraph proses ahp

Keterangan :
Node (N) = 17
Edge (E) = 20
Predikat (P) = 3
a. Region (R) = 4
b. Cyclomatic complexcity
1. V (G) = (E – N) + 2
= (19 – 17) + 2
= 4
2. V (G) = P + 1
= 3 + 1
= 4
c. Independent Path
Path 1 : 1-2-3-4-5-6-16-17
Path 2 : 1-2-3-7-8-9-10-16-17
Path 3 : 1-2-3-7-11-12-13-14-16-17
Path 4 : 1-2-3-7-11-15-16-17

Berdasarkan hasil yang didapatkan dimana Region, Independentt Path dan Cyclomatic Complexcity bernilai sama. Maka dapat disimpulkan bahwa aplikasi yang dirancang dapat dikatakan bebas dari kesalahan logika

4.3 Hasil Pengujian

Adapun hasil pengujian yang dilakukan dengan metode white box, dapat dilihat pada tabel berikut:

Tabel 4.1 Hasil Pengujian Perangkat Lunak

No	Flowgraph	Independent Path	Region	Cyclomatic Compexcity
1	Kriteria	7	7	7
2	Bobot	7	7	7
3	Usaha	7	7	7
4	Proses ahp	4	4	4
Total		13	13	13

Berdasarkan Rekapitulasi perhitungan di atas jumlah Region, *Cyclomatic Complexcity*, Indepnden Path yang bernilai sama maka dapat disimpulkan bahwaaplikasi yang dirancang dapat dikatakan bebas dari kesalahan logika.

5. Kesimpulan

Dari hasil pengujian dan pembahasan yang dibuat, maka telah dihasilkan suatu sistem aplikasi trending topik yang dapat disimpulkan dengan uraian seperti di bawah ini :

1. Aplikasi pemilihan usaha dapat memberikan kemudahan kepada user dalam mengetahui jenis usaha yang sebaiknya dikembangkan
2. Metode ahp dapat diterapkan untuk mengetahui informasi mengenai usaha yang sebaiknya dikembangkan
3. Dari hasil pengujian program yang telah dilakukan, aplikasi yang telah dirancang sudah bebas dari kelemahan-kelemahan yang dapat menyebabkan program tidak dapat berjalan sesuai dengan apa yang diharapkan.

Daftar Pustaka

[1] Jogiyanto, HM, 2012, "Analisis dan Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis. Jogjakarta": Andi Offset. Yogyakarta

[2] Nugroho, Adi. 2012. "Rekayasa Perangkat Lunak Beorientasi Objek dengan Metode USDP". Jogjakarta: Andi.

[3] Santoso, Harip. 2000. "Membuat Multi Aplikasi Menggunakan Visual Basic 6." Jakarta: Alex Media Komputindo.

[4] Suhendar A & Gunadi Hariman 2008," Visual Modelling Menggunakan UML dan Rational Rose", Jakarta: Informatika.

[5] Iskandar Z .Nasibu.2009. Penerapan Metode AHP Dalam Sistem Pendukung Keputusan Penempatan Karyawan Menggunakan Aplikasi Expert Choice. Jurnal Pelangi Ilmu Volume 2 No.5, Mei 2009

[6] Nur Rochmah Dyah P.A, Armandira Maulana P. 2009. Sistem Pendukung Keputusan Perencanaan Strategis Kinerja Instansi Pemerintah Menggunakan Metode AHP. Jurnal Informatika Vol 3, No 2, Juli 2009

[7] Kusumadewi S, Hartati S, dan Wardoyo,R 2006, Fuzzy Multi-Attribute Decesion Making, Graha Ilmu. Yogyakarta.

[8]Wieta B. Komalasari.2007. Metode Pohon Regresi Untuk Eksploratori Data Dengan Peubah Yang Banyak Dan Kompleks. Jurnal Informatika Pertanian Vol 16 No.1, Juli 2007

[9] Azis, Anifuddin, Sunarminto, Hendro., Medhanita, Dewi Renanti (2006). Evaluasi Kesesuaian Lahan Untuk Budidaya Tanaman Pangan Menggunakan Jaringan Syaraf Tiruan

[10] Honggowibowo, Setiawan, Anto. (2007). "Sistem Pakar Diagnosa Penyakit Tanaman Padi Berbasis Web Dengan Forward dan Backward Chaining", ISSN : 1693-6930