

Pengembangan Sistem Informasi Manajemen *Barbershop* dengan Pendekatan *Gamification* dan *Forward Chaining* (Studi Kasus : *Goodfellas Barbershop*)

M. Hamzah¹, Adam Hendra Brata², Eriq Muhammad Adams Jonemaro³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹cupite_lupyma@student.ub.ac.id, ²adam@ub.ac.id, ³eriq.adams@ub.ac.id

Abstrak

Salah satu *barbershop favorite* di malang berdasarkan Radar Malang Award 2019 adalah *GoodFellas Barbershop*. Dari proses bisnis yang berlangsung diperoleh beberapa permasalahan yang dialami oleh pihak *GoodFellas* seperti kurangnya manajemen data pelanggan, pengantrian *customer* yang masih memanfaatkan *note paper*, susahny *memberikan* hadiah atau promo, *customer* tidak memiliki gambaran akan potong rambut yang diinginkan, susahny mengelola *product*, dan perlunya suatu metode yang membikin pelanggan untuk rutin memotong rambut. Sistem manajemen informasi *barbershop* ini dibangun untuk membenahi permasalahan tersebut. Sistem ini dibangun dengan pendekatan metode *gamification* untuk membikin pelanggan rutin memotong rambut dan *forward chaining* untuk rekomendasi potongan rambut. Proses pengembangan sistem ini dilakukan dengan memanfaatkan metode *prototyping* dengan dua kali iterasi. Pada tahap analisis kebutuhan diperoleh 54 kebutuhan fungsional setelah 2 kali iterasi dan 2 kebutuhan non fungsional. Pada tahap implementasi dilakukan dengan memanfaatkan *framework codeigniter*. Pengujian dilakukan dengan 2 pendekatan yaitu pendekatan *white-box* dan pendekatan *black-box*. Pendekatan *white-box* dimanfaatkan untuk pengujian unit dengan hasil 100% valid sedangkan pendekatan *black-box* dimanfaatkan untuk pengujian validasi dengan hasil 100 % valid. Untuk pengujian non fungsional diperoleh Pengujian *compatibility* dengan hasil dapat berlangsung di berbagai macam browser dan pengujian usability memanfaatkan metode *Task Scenario* dengan hasil 100% dan *System Usability Scale* (SUS) dengan hasil 86.5 yang berarti termasuk kategori *acceptable* sehingga aplikasi sudah serasi dengan kebutuhan pengguna.

Kata kunci: *Sistem Informasi Manajemen, Barbershop, Gamification, Forward Chaining*

Abstract

One of the favorite barbershop in Malang based on Radar Malang Award 2019 is GoodFellas Barbershop. From the running business processes, there are several problems experienced by GoodFellas such as lack of customer management, the queue of customers who still use note paper, it's hard to give promos, The customer has no idea of hair style, difficult to manage products owned, and they need methods that make customers cut their hair regularly. This barbershop information management system was built to overcome these problems. This system is build by using gamification method to make customers using this system continuously and forward chaining for recomentation hairstyle. The system development process is carried out using the prototyping method. In the needs analysis phase, 54 functional needs and 2 non-functional requirements are found with 2 iterations At the implementation stage, it is carried out using a codeigniter framework. Testing is done with 2 approaches, namely the white-box approach and the black-box approach. The white-box approach is used for unit testing with 100% valid results while the black-box approach is used for validation testing with 100% valid results. Compatibility testing has been done that this system can run on many browser. Usability testing is done with 2 approaches, namely Task scenario and System Usability Scale (SUS), for task scenario get 100% valid result, and for system usability scale (SUS) get score 86.5 which means included in the acceptable category so that the application is according to user requirements

Keywords: *Management System Information, Barbershop, Gamification, Forward Chaining*

1. PENDAHULUAN

Untuk mendapatkan rambut yang baik biasanya laki-laki akan pergi ketempat cukur langganannya baik itu ditempat cukur tradisional, salon ataupun *barbershop*. *Barbershop* sendiri sudah banyak diperoleh di berbagai kota di Indonesia. salah satu *barbershop* yang terkenal dimalang adalah GoodFellas *Barbershop*

Berdasarkan wawancara terhadap pihak *owner* GoodFellas, proses bisnis yang berlangsung diperoleh beberapa kendala yang dialami oleh pihak GoodFellas seperti kurangnya manajemen pelanggan, pengantrian *customer* yang masih memanfaatkan *note paper*. susah nya *memberikan* promo, *customer* tidak memiliki gambaran akan potong rambut yang diinginkan, susah nya manajemen stok product yang ada. Dan Perlunya suatu sistem yang membikin pelanggan rutin memotong rambut.

Berdasarkan permasalahan yang telah dijelaskan, solusi dari permasalahan tersebut adalah dengan mengembangkan suatu aplikasi yang dapat memajemen informasi baik dari *customer* dan *barberman*, *memberikan* rekomendasi potongan rambut untuk *customer* berdasarkan tipe wajah, tipe rambut dan kondisi rambut, menyajikan proses pengantrian berbasis online, menyajikan manajemen stok barang yang diperoleh di *barbershop*, dan membikin suatu sistem *gamification* untuk menerapkan hadiah-hadiah atau promo tertentu kepada pelanggan. dengan dibuatnya aplikasi ini diharapkan dapat membantu proses bisnis dari GoodFellas *barbershop*, *memberikan* kemudahan kepada pelanggan dalam proses pengantrian, mendapatkan rekomendasi potongan rambut yang serasi dengan bentuk wajah, tipe rambut dan kondisi rambut atau mendapat gambaran ingin memotong rambut dengan model tertentu yang sebelumnya hanya dari *catalog* saja, dan dengan adanya metode *Gamification* diharapkan pelanggan lebih cenderung untuk terus memanfaatkan aplikasi ini karena tujuan dari *Gamification* sendiri yaitu untuk menaikkan kontribusi, dorongan dan kineja pengguna saat atau memanfaatkan sesuatu. karena dengan metode *Gamification* biasanya suatu tugas tertentu menjadi lebih memikat karena unsur-unsur yang ada didalamnya. Unsur *Gamification* yang dimanfaatkan pada aplikasi

ini yaitu *Point Sistem, Rankings, Awards, dan Level*.

Aplikasi ini dibangun dengan memanfaatkan pendekatan OO (pendekatan berorientasi objek) yaitu pemodelan yang dimanfaatkan untuk menspesifikasikan atau mendeskripsikan sebuah sistem perangkat lunak yang tergantung dengan objek, Teknologi yang dimanfaatkan adalah *PHP* dan *MySQL* karena aplikasi yang akan dibuat adalah berupa platform website. Dalam pengujian fungsional dilakukan pengujian unit untuk memahami unit terkecil dari sistem dapat dimanfaatkan atau tidak, lalu melakukan pengujian validasi untuk memahami apakah sistem yang dibangun serasi dengan daftar kebutuhan yang telah dideskripsikan. Lalu untuk pengujian non-fungsional dilakukan pengujian usability untuk menetapkan bahwa sistem yang dibangun gampang untuk dimanfaatkan, dipahami dan dioperasikan oleh *User* nantinya. Pengujian selanjutnya yaitu pengujian *compatibility* untuk menetapkan bahwa aplikasi dapat berlangsung pada berbagai perangkat *browser* seperti *Google Chrome, Brave, Mozilla Firefox* dan lain-lain.

2. LANDASAN KEPUSTAKAAN

2.1 Gamification

Gamification dimanfaatkan untuk menaikkan kontribusi, dorongan, dan kemampuan pengguna saat melakukan tugas khusus, dengan cara memadukan mekanika dan unsur permainan, sehingga membikin tugas itu lebih memikat. Tujuan dari penelitian ini adalah untuk memutuskan dan mengkarakterisasi keadaan seni *gamification* dalam rekayasa perangkat lunak, menganalisis penelitian yang ada dengan demikian mengidentifikasi potensi celah dan peluang untuk penelitian di masa yang akan datang (Pedreire, 2015).

Berikut daftar dari unsur *gamification*. Unsur *gamification* dan mekanika yang ditemukan dalam studi utama adalah:

1. Penghargaan (*achievement*): penghargaan dikasih kepada pemain saat menuntaskan sebuah perilaku.
2. Sistem berbasis poin (*score*): pemain mendapat hadiah dalam bentuk poin saat menuntaskan perilaku tertentu.
3. Lencana (*badge*): prestasi tertentu dari pengguna.
4. Tingkat: pengguna memiliki tingkat yang

meningkat saat mereka menggapai sejumlah titik tertentu.

5. *Quests*: tugas yang harus diselesaikan pemain, dengan unsur permainan tambahan (seperti cerita) yang membikinnya lebih menarik.

6. *Voting*: pemain dapat memberikan suara pada perilaku pemain lain. Suara itu sendiri mewakili penghargaan yang didapat masing-masing pemain.

7. Peringkat: peringkat dengan pemain terbaik disajikan ke semua pemain untuk menaikkan daya saing. Posisi di peringkat bisa ditentukan oleh poin, tingkat, atau voting.

8. Pertaruhan: pengguna bertaruh pada acara tertentu, seperti prediksi. Pemenang taruhan akan menerima beberapa imbalan.

Dalam pembuatan sistem ini tidak semua unsur dimanfaatkan, hanya sebagian unsur yang dimanfaatkan seperti *Point Sistem, Rankings, Awards, dan Level*.

2.2 Sistem Informasi Manajemen

Menurut Mcleod (2001) sistem informasi manajemen adalah suatu sistem pembuat informasi yang membantu sekelompok manajer suatu unit organisasi seperti suatu tingkat manajemen atau suatu bidang fungsional.

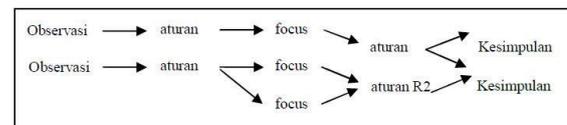
Sedangkan menurut O'Brien (2003) sistem informasi manajemen adalah sebuah sistem yang menyajikan informasi dalam bentuk laporan dan gambaran untuk diberikan kepada manajer dan para professional misalnya Analisa penjualan, kemampuan, produksi, sistem laporan biaya.

Lebih lanjut Gordon B. Davis juga menekankan bahwa sistem informasi manajemen (SIM) selalu berhubungan dengan pengolahan informasi yang berbasis pada *computer*. SIM merupakan suatu sistem yang melakukan fungsi-fungsi untuk menyajikan semua informasi yang berpengaruh pada operasi organisasi.

2.3 Forward Chaining

Sasikumar *et al.* (2007), menjelaskan *forward chaining* adalah metode pelacakan yang dimulai dengan fakta yang diketahui lalu memadankan fakta-fakta tersebut dengan fakta yang ada, lalu memadankan fakta-fakta tersebut dengan IF dari rules IF-THEN. Bila ada fakta yang cocok dengan IF, maka aturan tersebut dieksekusi. Bila sebuah aturan dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan kedalam database. Setiap kali pencocokkan,

dimulai dari rule teratas. Metode pelacakan yang dimanfaatkan bisa *Depth-first Search, Breadth-First Search* atau *Best First Search*. pendekatan dalam penelusuran dimulai dari informasi masukan dan selanjutnya mencoba memaparkan kesimpulan, penelusuran kedepan mencari fakta yang serasi dengan IF dari aturan IF-THEN. Metode ini cocok dimanfaatkan untuk membenahi masalah pengendalian (*controlling*) dan peramalan (*prognosis*) (Giarattano dan Riley 1994). Alur Inferensi forward chaining ditampilkan pada gambar 1.



Gambar 1 Alur Inferensi Forward Chaining

2.4 Best First Search

Best First Search Menurut T. Sutojo, dkk, merupakan gabungan dari metode *Breadth First Search* dan *Dept First Search* dimana pelacakan dibolehkan mengunjungi *node* level lebih rendah jika *node* level lebih tinggi memiliki nilai *heuristic* lebih buruk. Algoritma pada *best first search* yaitu :

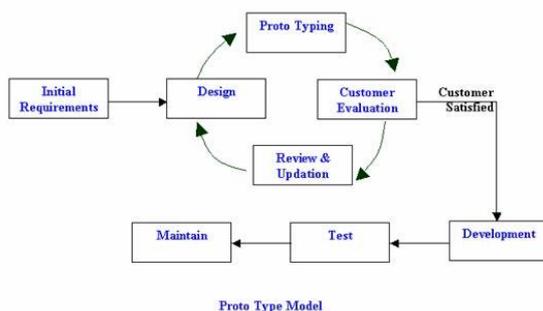
1. Buat sebuah *stack* lalu inisialisasikan *node* akar sebagai *node* pertama.
2. Bila *node* pertama \neq *GOAL*, *node* diganti dan dihapus dengan anak-anaknya.
3. Lalu, keseluruhan *node* pada *stack* diurutkan secara *ascending* berdasarkan fungsi *heuristic* yang dimanfaatkan.
4. Ulangi langkah 2 Sampai *node* pertama = *GOAL*, cari solusi dengan menelusuri jalur dari *GOAL* ke *node* akar.

Metode ini akan mengunjungi dimulai dari *node* terdalam selama *node* tersebut merupakan *node* yang terbaik. Jika *node* yang sedang dikunjungi ternyata tidak menuju kepada solusi yang diinginkan, maka akan dilakukan runut balik kearah *node* akar untuk mencari *node* anak lainnya yang lebih menjamin daripada *node* terakhir yang dikunjungi. Bila tidak ada, maka akan terus mengulang mencari kearah *node* akar sampai ditemukan *node* yang lebih baik untuk dijalankan suksesornya. Dalam pembuatan sistem ini diperoleh fungsi rekomendasi potongan rambut berdasarkan tipe wajah, tipe rambut dan kondisi rambut yang akan memanfaatkan metode *Best First Search* ini.

2.5 Prototyping Model

Pemilihan *Model* pengembangan diserasikan dengan hasil analisis yang diperoleh dari masalah yang dihadapi. *Model prototyping* dimanfaatkan karena keinginan pengguna yang dapat berubah-ubah, pengguna juga dapat berpartisipasi dalam proses pengembangan agar kesalahan dapat terlihat lebih awal seiring proses untuk membikin sistem informasi manajemen dengan pendekatan *gamification* dan *forward chaining*.

Model *prototyping* terdiri dari proses *Requirement, Design, Coding, Testing, Review, Implementasi dan Maintenance* (Pressman,2010). Pada pembangunan aplikasi ini dimanfaatkan model *prototyping* dengan iterasi sebanyak minimal 2 kali dan dalam pengembangan ini tidak dimanfaatkan semua tahap pada model *prototyping*. Model *prototyping* dijelaskan pada gambar 2.



Gambar 2 Model Pengembangan Prototyping

Berikut Tahapan–tahapan Proses Pengembangan dalam Model Prototype, yaitu :

a. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendeskripsikan seluruh perangkat lunak, mengidentifikasi semua keinginan, dan garis besar sistem yang akan dibikin.

b. Membuat *prototype*

Membuat *prototype* dengan membikin perancangan sementara yang berfokus pada pengutaraan kepada pelanggan (misalnya dengan membikin input dan utput).

c. Evaluasi *prototyping*

Evaluasi ini dilakukan oleh pelanggan, apakah *prototype* yang sudah dibangun sudah serasi dengan keinginan atau belum. Jika sudah serasi, maka langkah selanjutnya akan dijalankan. Namun jika tidak, *prototyping* direvisi dengan mengulang langkah-langkah sebelumnya.

d. Implementasi sistem

Dalam tahap ini *prototype* yang sudah di sepakati diubah ke dalam bahasa pemrograman yang serasi.

e. Pengujian sistem

Setelah sistem sudah menjadi suatu perangkat lunak yang siap digunakan, lalu dilakukan proses Pengujian. Pengujian ini dilakukan dengan metode *White Box, Black Box*, dan lain-lain.

f. Evaluasi Sistem

Pelanggan mengevaluasi apakah perangkat lunak yang sudah jadi sudah serasi dengan yang diinginkan. Jika ya, maka proses akan dilanjutkan ke tahap selanjutnya, namun jika tidak serasi dengan apa yang diinginkan, maka tahapan sebelumnya akan diulang.

g. Memanfaatkan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk dimanfaatkan.

Dengan metode ini pelanggan dan pengembang dapat saling berinteraksi selama proses pembuatan sistem. *Prototyping* dapat diartikan sebagai proses yang dimanfaatkan untuk membantu pengembang perangkat lunak dalam membentuk model dari perangkat lunak yang ingin dibuat.

Terdapat tiga bentuk model yaitu :

1. Bentuk *prototype* di atas kertas/model berbasis komputer yang memaparkan interaksi manusia yang mungkin terjadi.
2. *Working prototype*, yang membuat sebagian dari fungsi yang ingin di perangkat lunak.
3. Program jadi yang membuat sebagian atau seluruh fungsi yang akan dilakukan, tapi ada fitur yang masih dikembangkan.

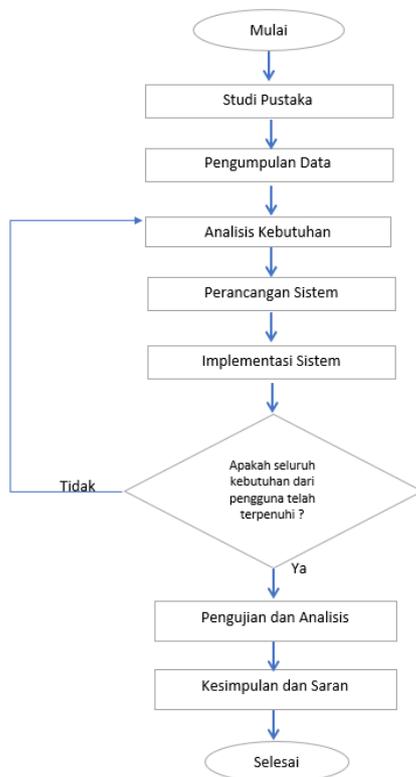
Dalam *prototyping* ada 3 pendekatan utama, yaitu:

- a. *Throw-Away, prototype* dibuat dan dites. hasil yang diperoleh dari pembuatan *prototype* dimanfaatkan untuk membikin produk akhir, lalu *prototype* tersebut tidak digunakan lagi.
- b. *Incremental*, Produk akhir dibuat sebagai komponen-komponen yang terpisah. Desain produk akhirnya secara keseluruhan hanya ada satu tetapi terbagi kedalam komponen-komponen kecil yang terpisah (*independent*).
- c. *Evolutionary*, Pada metode ini *prototype* tidak dibuang tetapi dimanfaatkan untuk

iterasi berikutnya. Dalam hal ini, sistem atau produk dipandang sebagai evolusi dari versi sebelumnya menuju produk akhir.

Pada pembangunan aplikasi ini pemodelan *prototype* memanfaatkan pemodelan program jadi yang melakukan seluruh fungsi yang akan dilakukan lalu dengan memanfaatkan pendekatan *evolutionary* maka program jadi tersebut dievaluasi apakah serasi dengan keinginan pengguna atau tidak, jika iya maka akan dilakukan pengujian terhadap sistem tersebut, dan jika tidak maka dilakukan perbaikan pada iterasi berikutnya.

3. METODOLOGI PENELITIAN



Gambar 3 Alur Metodologi Penelitian

Tahapan pertama yang dilakukan yaitu studi kepustakaan. tahap ini dilakukan dengan menghimpunkan dasar teori yang akan dimanfaatkan dalam membangun sistem yang diperoleh dari ebook, jurnal, dan penelitian sebelumnya.

Tahapan selanjutnya adalah analisis kebutuhan. Pada tahap ini, analisis kebutuhan dilakukan untuk memahami keinginan dari pelanggan yang diperlukan dalam melakukan pengembangan sistem. Baik kebutuhan fungsional maupun kebutuhan non fungsional. Metode yang dimanfaatkan dalam analisis

kebutuhan adalah pendekatan *object oriented analysis*. Proses ini dimulai dari elisitasi dan analisis keinginan pelanggan. Elisitasi dilakukan dengan cara observasi dan wawancara.

Tahapan selanjutnya adalah perancangan. Perancangan dilakukan berdasarkan hasil dari tahapan analisis kebutuhan. Perancangan sistem penelitian ini memanfaatkan *Object Oriented Design* sehingga pemodelan menggunakan *Unified Modelling Language*. Hasil dari perancangan yaitu perancangan arsitektur, komponen dan perancangan antarmuka.

Tahapan selanjutnya adalah implementasi. Pada tahap ini proses rancangan sistem menjadi kode program. Implementasi ini memanfaatkan metode *Object Oriented Programming*. Sebelum ketahap selanjutnya dilakukan evaluasi terhadap sistem apabila diperoleh perubahan maka kembali ketahap analisis kebutuhan, jika tidak maka lanjut ke tahap selanjutnya.

Tahapan selanjutnya adalah pengujian. Pada tahap ini dilakukan 2 pendekatan pengujian yaitu *White-Box* dan *Black-Box* pada pendekatan *White-Box* melakukan pengujian unit sedangkan untuk *Black-Box* melakukan pengujian validasi, lalu unon-fungsional terdiri dari pengujian *compatibility* dan pengujian *usability*.

Tahapan terakhir adalah kesimpulan dan saran. Pada tahap ini dilakukan pengambilan kesimpulan dari proses analisis sampai dengan pengujian untuk menjawab rumusan masalah.

4. ANALISIS KEBUTUHAN

4.1 Elisitasi Kebutuhan dan Analisis Kebutuhan

Elisitasi kebutuhan dilakukan dengan memanfaatkan metode wawancara, observasi dan pengamatan dokumen proses bisnis. observasi dilakukan langsung di *barbershop GoodFellas*. Observasi ini dilakukan dengan mengamati proses bisnis yang berlangsung pada *barbershop goofellas*. wawancara memanfaatkan metode *unstructured interviews*. Wawancara dimanfaatkan untuk mencari permasalahan yang ada pada *goodfellas barbershop*. Narasumber pada wawancara ini yaitu admin dan owner *goodfellas barbershop*. Pengamatan dokumen dilakukan untuk memahami proses bisnis secara keseluruhan berdasarkan data yang ada

4.2 Identifikasi Aktor

Tabel 1 Identifikasi Aktor

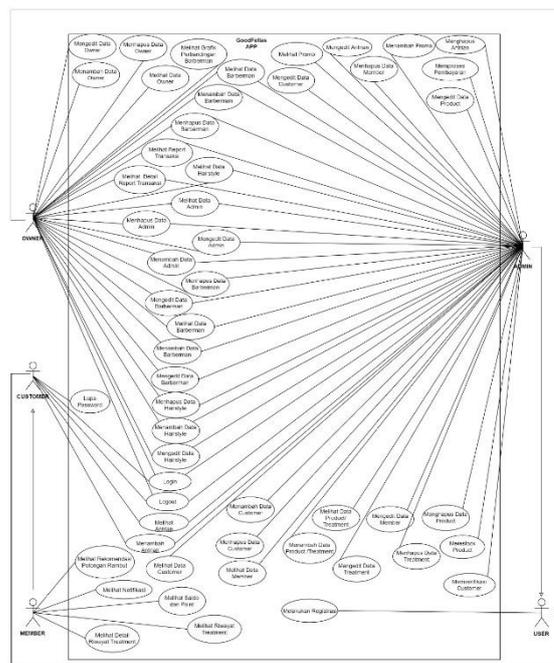
Aktor	Deskripsi
<i>user</i>	<i>User</i> merupakan aktor yang hanya dapat melakukan fungsi login untuk masuk ke dalam sistem
<i>Owner</i>	Merupakan pemilik dari <i>Barbershop GoodFellas</i> , dimana dalam hal ini dapat melihat laporan harian <i>barbershop</i> , laporan bulanan <i>barbershop</i> , melihat stok barang dan pemberitahuan tergantung stok barang masuk dan keluar, melihat statistic <i>barberman</i> yang sering mendapatkan pelanggan dan dapat melihat seluruh data pelanggan.
<i>Customer</i>	Merupakan seorang pelanggan dari <i>Barbershop GoodFellas</i> , dimana dalam sistem sebagai pelanggan <i>non-member</i> , dalam hal ini <i>customer</i> hanya dapat <i>login</i> , dan melakukan fungsi antrian saja.
<i>Admin</i>	Merupakan seseorang Administrative dari <i>Barbershop GoodFellas</i> , dimana dalam hal ini adalah seseorang yang melakukan pendataan <i>member</i> , <i>memberikan</i> laporan harian & bulanan, dan juga sebagai seorang kasir.
<i>Member</i>	Merupakan seorang pelanggan dari <i>Barbershop GoodFellas</i> , dimana dalam sistem sebagai pelanggan <i>member</i> , dalam hal ini <i>member</i> tidak hanya dapat melakukan fungsi selayaknya <i>customer</i> , namun juga mendapatkan promo-promo atau hadiah memikat yang tersedia dari <i>Barbershop GoodFellas</i> .

4.3 Kebutuhan Sistem

Daftar kebutuhan sistem terdiri dari kebutuhan fungsional dan kebutuhan non fungsional. kebutuhan fungsional merupakan kemampuan atau layanan yang harus dimiliki oleh sistem sehingga sistem mampu berjalan dengan baik. Sistem informasi manajemen *barbershop* dengan pendekatan *gamification* dan *forward chaining* mempunyai 54 kebutuhan fungsional dan 2 kebutuhan non fungsional.

4.4 Use Case Diagram

Use Case Diagram adalah sebuah *diagram* yang dimanfaatkan untuk memodelkan perilaku sebuah user terhadap sistem. *Use Case Diagram* ini merupakan Gambaran kebutuhan fungsional sistem yang dibangun. Tujuan dibuat *Use Case Diagram* agar yang melihat diagram ini mengerti dan paham sistem yang akan dibuat. *Use case diagram* dapat dilihat pada Gambar 4.



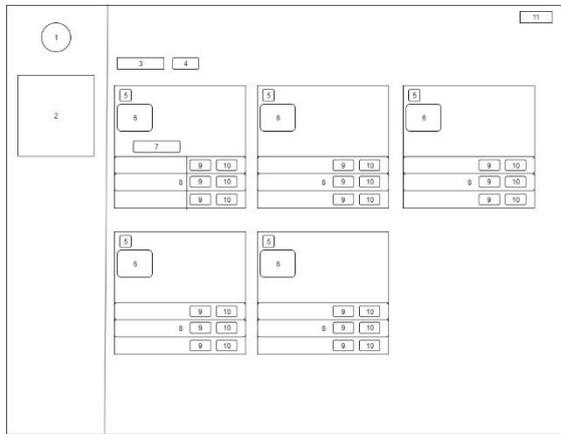
Gambar 4 Use Case Diagram iterasi kedua

5. PERANCANGAN DAN IMPLEMENTASI

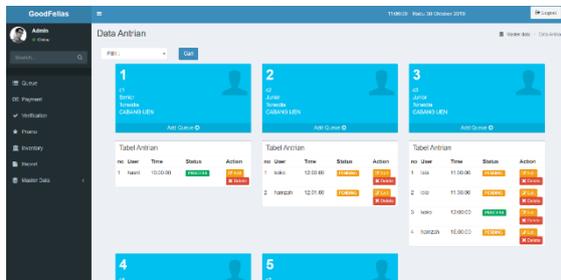
Tahapan perancangan mengubah hasil dari analisis kebutuhan menjadi struktur sistem. Perancangan dibagi menjadi 4 bagian yaitu perancangan arsitektur, perancangan data, perancangan komponen dan perancangan antarmuka. Perancangan arsitektur adalah proses mengidentifikasi unsur – unsur dari struktur sistem dan relasi antar unsur. Pada tahap ini dimodelkan menjadi 2 yaitu pemodelan *sequence diagram* dan pemodelan *class diagram*. Pemodelan *sequence diagram*. Pada pemodelan *sequence diagram* pada use case memproses pembayaran. Perancangan data adalah proses mentransformasi hasil analisis menjadi struktur data. Hasil perancangan data dimodelkan dalam bentuk *Entity Relationship Diagram (ERD)* lalu diubah menjadi *Conceptual Data Model (CDM)* lalu menjadi *Physical Data Model (PDM)*. Perancangan antarmuka adalah proses memaparkan tampilan sistem yang akan dibuat. Untuk sampel perancangan antarmuka antrian dapat dilihat pada Gambar 5.

Tahapan Implementasi memanfaatkan hasil dari proses perancangan sehingga menjadi kumpulan kode program untuk membangun sistem. Implementasi memiliki tiga bagian yaitu implementasi kode program, implementasi data dan implementasi antarmuka. Implementasi antarmuka antrian admin dapat dilihat pada Gambar 6. Pada implementasi kode program memanfaatkan *framework codeigniter* dan PHP

sebagai bahasa pemrograman. Untuk basis data memanfaatkan MySQL dan untuk tampilan memanfaatkan CSS dan Bootstrap.



Gambar 5 Perancangan Antarmuka Antrian



Gambar 6 Antarmuka Antrian

6. PENGUJIAN

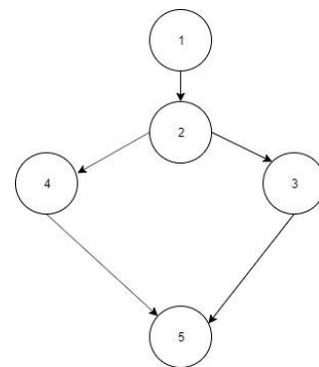
6.1 Pengujian Unit

Pengujian Unit digunakan untuk menguji alur program sistem di unit terkecil. Pengujian dilakukan pada method sampel insert_antrian() dari klas c_antrian.

Tabel 2 Pseudocode method insert_antrian()

1	Start	
2	Inisialisasi variable id_barberman =	1
3	Inputan id barberman yang user pilih	
4	Inisialisasi variable no_kursi = inputan	
5	no kursi barberman yang user pilih	
6	inisialisasi variable time = inputan waktu	
7	yang user masukkan	
8	inisialisasi variable id_barberman =	
9	inputan nama cabang yang user pilih	
10	inisialisasi variable where = variable	
11	nomor kursi, cabang dan waktu di dalam	
12	array	
13		
14	inisialisasi variable data = variable	
15	antrian status sebagai "pending", variable	
16	id barberman, nomor kursi, pelanggan,	
17	cabang dan waktu didalam array	
18		
19		
20	memanggil method cek_data pada	
21	model = antrian dengan parameter	

22	antrian dan variable where yang	
23	disimpan pada variable	
24		
25	if (cek > 0)	2
26		
27	menampilkan "waktu sudah ada yang	3
28	pesan, silahkan pilih waktu yang lain"	
29	memanggil fungsi index dari controller	
30	C_antrian	
31		
32	Else	4
33	Menampilkan "sukses input data"	
34	Memanggil fungsi index dari controller	
35	C_antrian	
36		
37	End if	5
38	end	



Gambar 7 Flow Graph Method insert_data()

Jalur 1 : 1 – 2 – 3 – 5

Jalur 2 : 1 – 2 – 4 – 5

Perhitungan cyclomatic complexity :

$$V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

$$V(G) = R$$

$$= 2$$

Tabel 3 Hasil Pengujian Unit method c_antrian.

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Kelas antrian memanggil fungsi cek_data dengan nilai cek > 0 lalu memanggil fungsi index pada	Sistem akan menampilkan pesan "waktu sudah ada yang pesan, silahkan pilih waktu yang lain"	Sistem menampilkan pesan "waktu sudah ada yang pesan, silahkan pilih waktu yang lain"	passed

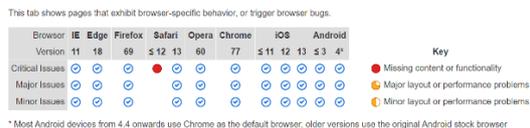
controller				
2	Kelas antrian memanggil fungsi cek_data dengan nilai cek <= 0 lalu memanggil fungsi index pada pada controller	Sistem akan menampilkan pesan "Sukses" Input Data"	Sistem akan menampilkan pesan "Sukses" Input Data"	passe d

6.2 Pengujian Validasi

Pengujian Validasi digunakan untuk memastikan semua kebutuhan telah diimplementasi dengan baik dan benar. Hasil pengujian dari 65 kasus uji mendapatkan hasil 100% valid. Berdasarkan hal tersebut maka sistem informasi manajemen *barbershop* dengan pendekatan *gamification* dan *forward chaining* telah dijalankan dengan baik serasi dengan hasil yang diinginkan.

6.3 Pengujian Compatibility

Pengujian *Compatibility* dilakukan dengan memanfaatkan bantuan aplikasi *Sortsite*. Hasil pengujian menunjukkan bahwa aplikasi dapat berlangsung pada beberapa *browser* dapat dilihat pada Gambar 8.



Gambar 8 Hasil Pengujian *Compatibility*

6.4 Pengujian Usability

Pengujian *Usability* memanfaatkan dua metode yaitu *task scenario* dan *system usability scale (SUS)*. Untuk hasil *task scenario* mendapatkan hasil 100% pada seluruh task yang diberikan dan hasil *system usability scale* dapat dilihat pada tabel 4.

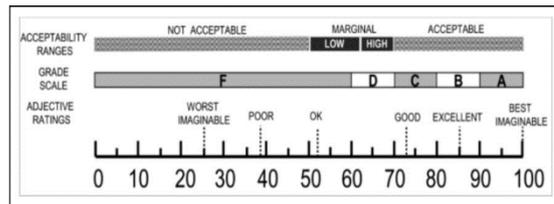
Tabel 4 Hasil Pengujian *System Usability Scale*

	No	Nama	Nilai Jawaban										Total Skor	Skor Akhir
			1	2	3	4	5	6	7	8	9	10		
1	Febrian Seti	3	4	2	5	3	4	3	5	3	5	37	37	

a	2.												
Putr	5												
a	=												
	9												
	2.												
	5												
2	Putr	3	5	3	4	2	4	3	5	2	5	36	3
.	i												6
	Rizki												x
	ki												2.
	Audya												5
													=
													9
													0
3	Alp hab arka h Azi z Ifrani	2	4	3	4	3	5	2	4	3	4	34	3
.													4
													x
													2.
													5
													=
													8
													5
4	Ani f Fathul Amin	3	4	2	5	3	4	2	3	3	4	33	3
.													3
													x
													2.
													5
													=
													8
													2.
													5
5	Muhamad Abbulu Adam Rahman	3	5	2	4	2	5	3	4	1	4	33	3
.													3
													x
													2.
													5
													=
													8
													2.
													5

Rata-Rata Skor Akhir 86.5

skor *likert* yang diperoleh akan dilakukan analisis Hasil dari pengujian kali ini usability diukur memanfaatkan pengujian post study dengan metode System Usability Scale adalah 86.5 yang berarti masuk kedalam acceptability rating yaitu diperoleh acceptable dan adjective ratings yaitu excelent sehingga aplikasi sudah serasi dengan keinginan pengguna. Seperti pada gambar 10.



Gambar 10 Rating dan skala konversi rata-rata *System Usability Scale*

7. PENUTUP

7.1 Kesimpulan

1. Hasil analisis kebutuhan sistem manajemen informasi *goodfellas* menghasilkan 54 kebutuhan fungsional dan 2 kebutuhan non-fungsional dari 2 kali iterasi. Aktor yang terlibat pada sistem ada 5 yaitu *user*, *Admin*, *owner*, *customer*, dan *member*. Hasil analisis kebutuhan diperoleh dari studi literatur, wawancara dan observasi.

2. Hasil perancangan sistem informasi manajemen *barbershop* menghasilkan beberapa diagram seperti *sequence diagram*, *class diagram* untuk masing – masing *model* dan *controller*, lalu perancangan komponen pada empat fungsi dan desain antarmuka sistem. Perancangan dibuat serasi dengan analisis kebutuhan yang telah dilakukan dengan memanfaatkan metode *prototyping* untuk mendapatkan daftar keinginan dan desain sistem.

3. Hasil implementasi sistem dilakukan berdasarkan hasil perancangan sistem . Sistem informasi manajemen *barbershop* dibangun dengan memanfaatkan *framework Codeigniter* untuk Bahasa pemrograman *PHP*, *MySQL* untuk basis data dan untuk *front-end* memanfaatkan *framework bootstrap*, *HTML* dan *CSS*. Proses implementasi menghasilkan empat fungsi *algoritme* dan Sembilan belas desain tampilan sistem.

4. Hasil pengujian *fungisonal* dilakukan memanfaatkan pendekatan *white box* dan *black box*. Pendekatan *white box* dimanfaatkan untuk pengujian unit. Pengujian unit dilakukan memanfaatkan *basis path Testing* menghasilkan 100% valid dari 18 kasus uji. Sedangkan pendekatan *black box* dimanfaatkan untuk pengujian validasi. Pengujian validasi yang diperoleh dari *use Case scenario* menghasilkan 100% valid dari 65 kasus uji.

5. Hasil Pengujian *non-fungsional* diperoleh 2 pengujian yaitu pengujian *compatibility* dan pengujian *usability*. Untuk pengujian *compatibility* memanfaatkan aplikasi bantuan *sortside* dengan hasil menunjukkan bahwa sistem dapat dioperasikan pada berbagai macam *browser* terutama *Google chrome* ,*brave browser*, dan *Mozilla Firefox*. Untuk pengujian *usability* terbagi menjadi 2 yaitu *task scenario* dan *system usability scale (SUS)*. Hasil yang diperoleh dari *task Testing* yaitu diperoleh hasil 100% pada seluruh *task* yang diberikan. Dan

Hasil yang diperoleh dari *system usability scale (SUS)* yaitu sebesar 86.5 yang berarti masuk kedalam kategori *acceptable* sehingga aplikasi sudah serasi dengan keinginan pengguna.

7.2 Saran

1. Menambah fitur rating penilaian dari pelanggan terhadap *barberman* yang telah melakukan *treatment*.
2. Menambah fitur target *barberman* untuk penilaian dari sisi *owner*.
3. Menambah fitur sms untuk memahami lebih mudah pemberitahuan transaksi.
4. Menambah fitur untuk admin mengatur rekomendasi *hairstyle* yang ditampilkan berdasarkan potongan rambut yang lagi trend / viral.
5. Menambah fitur suka terhadap rekomendasi *hairstyle*.
6. Menambahkan fitur kirim promo ulang tahun secara otomatis berdasarkan bulan kelahiran.

8. DAFTAR REFERENSI

- Afuan. (2010). Pemanfaatan Framework Codeigniter dalam Pengembangan Sistem Informasi Pendataan Laporan Kerja Praktek Mahasiswa Program Studi Teknik Informatika Unsoed (Codeigniter Framework Used in Information System Development for Student's Report Data Collection .
- Andi Setiawan, A. I. (2006). Pengolahan Database MySQL dengan Script PHP.
- Anhar. (2010). *Panduan Menguasai PHP & MySQL Secara Otodidak*. Jakarta.:
- Anhar. (2010). Pemanfaatan Sistem Informasi Perpustakaan Digital Berbasis Website.
- Booch, G. J. (2007). *The Unified Modeling Language User Guide Second Edition*. United State: Addison Wesley Professional.
- Brooke, J., 1996. *SUS - A quick and dirty Usability Scale*. *Usability Evaluation in Industry* , pp. 189-194.
- Fatansyah, I. (2007). *Basis Data*. Bandung: Bandung : BI-OBSE.
- Fowler. (2003). *UML DISTILLED*
- Haverbeke. (2018). *ELOQUENT JAVASCRIPT*.

- James Rumbaugh, I. J. (2015). *The Unified Modeling Language Reference Manual*, Addison-Wesley.
- Laudon, K.C., & Laudon, J.P. (1998). *Managemen Information Systems New Approaches to Organization & Technology*. Upper Sadle River New Jersey: Prentice-Hall.
- Likert, R., 1932. Technique for the measurement of attitudes. *Archives of Psychology*, Volume 140, pp. 1-55.
- Marsic, I. (2012). *Software Engineering*. [e-book] New Jersey: Rutgers University. Tersedia di: <<https://www.ece.rutbrooke.edu/~marsic/>>
- Brooke, J., 1996. SUS - A quick and dirty usability scale. *Usability Evaluation in Industry* , pp. 189-194.
- McLeod, R., Jr. (1995). *Management Information Sistem*. Upper *Saddle River*, New Jersey: Prentice-Hall, Inc.
- Murdick, R. G., Ross, J. E., Clagget, J.R. (1997). *Sistem Informasi untuk Manajemen Modern edisi ketiga*(Terjemahan J. Djamil) Jakarta: Erlangga. (Buku asli diterbitkan tahun 1984).
- Nielsen, J., 2000. Why You Only Need to *Test with 5 Users*. [Online] Available at: [HTTPS://www.nngroup.com/articles/why-you-only-need-to-Testwith-5-Users/](https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/) [Accessed 6 June 2019].
- Onong Uchjana Effendi. (1989). *Sistem Informasi Manajemen*. Bandung: Mandar Maju.
- Pedreira, O., Garcia, F., dkk.2015. Gamification in software engineering – Asistematic mapping. Institute of Technology and Information Sistem, University of Castilla-La Mancha, Spain
- Pressman, R. (2010). *Software Engineering : a practitioner's approach*. New York.
- Sasikumar, M, S Raman, S Muthu Raman, KSR, Anjaneyulu, R Chandrasekar. 2007. *A Practical Introduction to Rule Base Expert Systems*. New Dehli : Narosa Publishing House.
- Sommerville, I. (2009). *Software Engineering Ninth Edition*. 9th penyunt.s.l.:s.n.