

Penerapan Antarmuka Adaptif Berbasis Perilaku Pemain pada *E-Learning* Bidang Pemrograman

(Implementation of Player Behavior-Based Adaptive Interfaces in *E-Learning Programming*)

Fajar Pradana¹, Fitra A. Bachtiar², Retno Indah Rokhmawati³

Abstract— The pandemic has caused a significant impact on the educational sector's implementation. The teaching and learning process that was previously carried out face-to-face now must be conducted online. Online learning utilizing e-learning is very beneficial for learners because it can be accessed online anytime, anywhere. Not only is it expected to be a medium to share material files that serve as learning supports, but e-learning is also required to replace the teachers' or lecturers' role in the classroom. In the teaching and learning process, it is essential to understand the students' conditions and behaviors. This knowledge of students' conditions and circumstances during the learning process can be used as resources for improving the quality and students' learning process. Most of the existing e-learning has not been equipped with features to detect the students' state while using the system. In this research, an adaptive interface was applied as a reflection of e-learning that could adapt to the characteristics of user behavior. The performance testing results showed that the time required for the log data with 950 active users was 23,35 ms. Meanwhile, based on the functionality test, the system succeeded in displaying the interface according to the cluster member.

Intisari— Pada masa pandemi seperti sekarang ini, bidang pendidikan mengalami dampak yang signifikan dalam pelaksanaannya. Proses belajar mengajar yang sebelumnya dilakukan secara tatap muka, saat ini harus dilaksanakan secara daring. Pembelajaran daring dengan memanfaatkan *e-learning* sangat bermanfaat bagi pembelajar karena dapat diakses secara daring kapan pun dan dari mana pun. *E-learning* diharapkan tidak hanya menjadi tempat berbagi *file* materi sebagai pendukung pembelajaran, tetapi perannya dituntut dapat menggantikan peran guru atau dosen di kelas. Dalam proses belajar mengajar, kondisi dan perilaku siswa merupakan hal yang penting diketahui. Pengetahuan tentang kondisi dan keadaan siswa ketika belajar dapat digunakan oleh pengajar sebagai bahan untuk meningkatkan kualitas maupun proses pembelajaran bagi siswa. Kebanyakan *e-learning* yang ada sekarang belum dilengkapi dengan fitur untuk mendeteksi keadaan siswa selama menggunakan sistem. Pada makalah ini dilakukan penerapan antarmuka yang adaptif sebagai cerminan *e-learning* yang mampu menyesuaikan dengan karakteristik perilaku pengguna. Pengujian kinerja menunjukkan hasil bahwa dengan jumlah data *log* mencapai 950 pengguna aktif, dibutuhkan waktu 23,35 ms. Sementara itu, pada uji fungsionalitas, sistem berhasil menampilkan antarmuka sesuai dengan *cluster member*.

Kata Kunci— *E-learning*, Adaptif, Perilaku, Karakter, Antarmuka.

^{1,2,3} Universitas Brawijaya, Jalan Veteran, Malang, Indonesia, 65145 (telp: 0341-551611 ext: 126; fax: 0341-565420; e-mail: fajar.p@ub.ac.id; fitra.bachtiar@ub.ac.id; retnoindah@ub.ac.id)

I. PENDAHULUAN

Pada masa pandemi seperti sekarang ini, bidang pendidikan mengalami dampak yang signifikan dalam pelaksanaannya. Proses belajar mengajar yang sebelumnya dilakukan secara tatap muka, saat ini harus dilaksanakan secara daring. Pembelajaran daring adalah proses belajar mengajar yang dilakukan dalam dan dengan bantuan jaringan internet [1]. Saat ini, siswa yang sedang belajar tidak harus berada di kelas, tetapi dapat belajar melalui media internet atau layanan pembelajaran daring yang biasa disebut *e-learning*. Proses belajar yang didukung dengan teknologi diharapkan dapat menjadi solusi pembelajaran pada masa pandemi dan pada akhirnya akan menggantikan pembelajaran konvensional di masa yang akan datang. Pembelajaran daring dengan memanfaatkan *e-learning* sangat bermanfaat bagi pembelajar karena dapat diakses secara daring kapan pun dan dari mana pun [2].

E-learning diharapkan tidak hanya menjadi tempat berbagi *file* materi untuk mendukung pembelajaran, tetapi perannya dituntut dapat menggantikan peran guru atau dosen di kelas. *E-learning* mengacu pada proses pembelajaran yang terpusat pada siswa atau disebut dengan *Student Centered Learning* (SCL) [3]. Pembelajaran yang berbasis pada siswa adalah pembelajaran yang keseluruhan aktivitasnya terpusat pada siswa. Model pembelajaran ini menekankan penguasaan pada kegiatan yang memungkinkan siswa untuk dapat melatih kemampuan agar memiliki kapasitas dalam penyelesaian masalah.

Dalam proses belajar mengajar, kondisi dan perilaku dari siswa merupakan hal yang penting untuk diketahui. Dengan mengetahui kondisi dan keadaan siswa ketika belajar, pengajar dapat meningkatkan kualitas maupun proses pembelajaran bagi siswa. Misalnya, jika ada siswa yang terlihat bosan dan mengantuk, guru dapat mengganti model pembelajaran ceramah dengan studi kasus atau permainan simulasi. Pada pembelajaran tatap muka secara konvensional, kondisi perilaku anak didik dapat diketahui langsung oleh pengajar. Kondisi yang dilihat dapat berupa motivasi, emosi, maupun tingkah laku peserta didik. Namun, proses identifikasi perilaku pengguna (*user*) ini sulit dilakukan pada pembelajaran yang berbasis teknologi dengan *e-learning*, tidak seperti pembelajaran konvensional yang melibatkan interaksi antara siswa dan guru [4]. Kebanyakan *e-learning* yang ada sekarang belum dilengkapi dengan fitur untuk mendeteksi keadaan siswa selama menggunakan sistem. Banyak *e-learning* yang berkembang saat ini merupakan *e-learning* konvensional yang digunakan untuk unggah/unduh materi dan tugas, mengerjakan

kuis, mengumpulkan tugas, dan sejenisnya. Fitur yang dapat merekam kondisi dan perilaku pengguna belum ditemukan, padahal pada pembelajaran adaptif yang berdasarkan identifikasi, gaya belajar individu merupakan fitur yang sangat penting untuk meningkatkan pembelajaran individu tersebut, terutama dalam pembelajaran daring.

Pada makalah ini dilakukan penerapan antarmuka yang adaptif sebagai cerminan *e-learning* yang mampu menyesuaikan dengan karakteristik perilaku pengguna. Pembelajaran yang dirancang pada saat ini bersifat personal yang mampu beradaptasi dengan masing-masing pengguna dan dapat memprediksi preferensi pengguna yang mengaksesnya [5]. Antarmuka dibedakan berdasarkan hasil *clustering* yang dieksekusi oleh *middleware* berdasarkan data *log* pengguna. Makalah ini merupakan pengembangan lanjut dari *e-learning* bidang pemrograman Code Mania (CoMa) [4], dan menerapkan hasil perhitungan *clustering* pada penelitian sebelumnya, yang di dalamnya terdapat tiga *cluster* karakteristik pengguna [6]. Penentuan jumlah kelompok dilakukan dengan cara melihat karakteristik data menggunakan metode *elbow* (*elbow method*). Proses perhitungan dengan metode *elbow* dilakukan secara iteratif. CoMa merupakan sebuah sistem *e-learning* yang dikembangkan dengan memanfaatkan *gamification* [4]. *E-learning* ini nantinya dapat menangkap perilaku pemain berdasarkan data *log* tersebut untuk kemudian dilakukan *clustering* menggunakan algoritme *fuzzy C-Means*. Hasil *clustering* digunakan untuk menentukan antarmuka yang tepat sesuai dengan karakter pemain. Personalisasi pengguna diperlukan karena setiap siswa mempunyai kemampuan yang berbeda-beda dalam memahami sebuah materi yang disampaikan [7].

II. E-LEARNING ADAPTIF

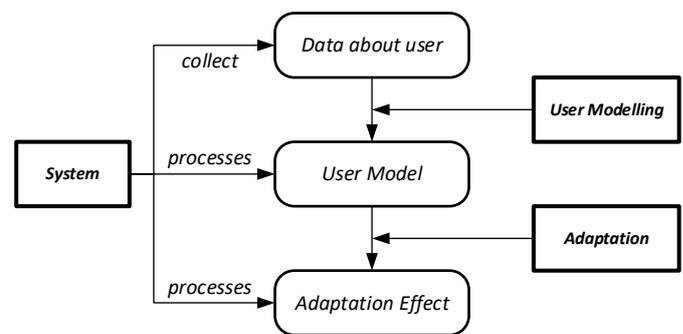
Perkembangan *e-learning* saat ini mengarah pada, salah satunya, pengembangan model pembelajaran yang lebih personal, bahkan pelajar dapat memilih cara belajar yang paling sesuai untuk diri mereka, sehingga pelajar tersebut merasa cocok dengan sistem pembelajaran yang diberikan. Model sistem *e-learning* adaptif diperlihatkan pada Gbr. 1 [8].

Proses model sistem *e-learning* adaptif pada Gbr. 1 terdiri atas tiga tahap sebagai berikut.

1) *Tahap Pengumpulan Data Mengenai Profil Pengguna (User Profile)*: Tahap ini bertujuan untuk mendapatkan informasi awal mengenai pengguna. Informasi ini disimpan sebagai model pengguna (*user model*).

2) *Tahap Pembangunan Model Pengguna*: Tahap ini membangun model pengguna yang didapatkan dari profil pengguna yang telah dikelompokkan. Model pengguna ini dapat dikelompokkan berdasarkan perilaku pengguna, pengetahuan pengguna, pencapaian pengguna, dan preferensi pengguna.

3) *Tahap Adaptasi*: Tahap adaptasi berisi satu set aturan adaptasi yang diimplementasikan dalam bentuk kondisi tertentu pada sistem *e-learning*. Aturan adaptasi ini dibuat berdasarkan model pengguna pada tahapan sebelumnya. Model adaptasi sistem bertujuan antara lain untuk memberikan materi



Gbr. 1 Model *e-learning* adaptif.

yang relevan dengan topik yang dipelajari, struktur navigasi tertentu, serta menyediakan fasilitas tambahan dan antarmuka yang berbeda.

III. E-LEARNING CODE MANIA (CoMa)

CoMa adalah sistem *e-learning* khusus bidang pemrograman. Dalam siklus hidup pengembangan perangkat lunak, *programming* merupakan kemampuan dasar yang harus dimiliki oleh tim pengembang [8]. CoMa dikembangkan untuk membantu para calon pengembang belajar dan meningkatkan kemampuan memrogramnya, khusus untuk bahasa pemrograman Java. Gbr. 2 menunjukkan tampilan awal CoMa yang mengadaptasi model pembelajaran yang atraktif dengan penerapan *gamification* untuk mendukung konsep pembelajaran SCL.

Fitur yang disediakan CoMa memungkinkan pengguna mengakses materi dan mengerjakan tantangan berupa soal-soal bidang pemrograman. *Course* yang disediakan merupakan materi pembelajaran Java, mulai dari dasar sampai menengah, seperti ditunjukkan pada Gbr. 3.

IV. FUZZY C-MEANS

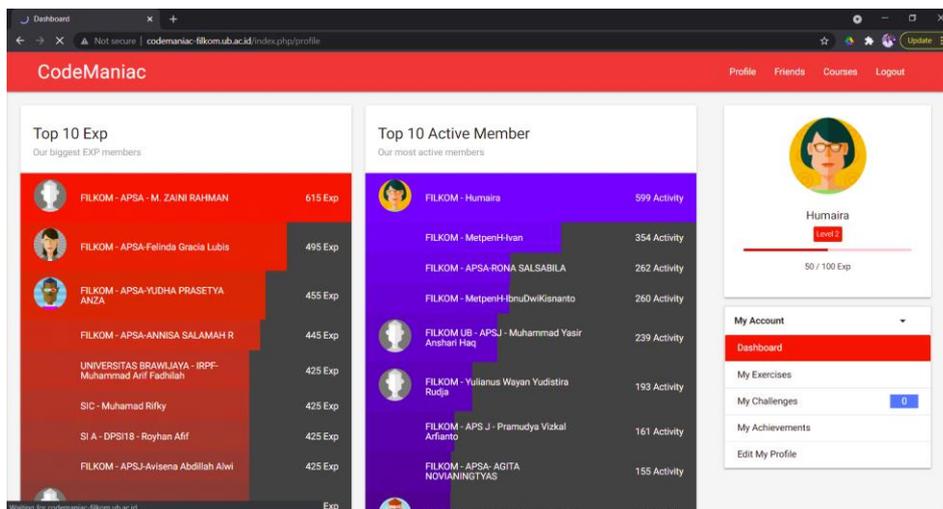
Algoritme *fuzzy C-Means* (FCM) dikembangkan oleh Dunn pada tahun 1973, lalu dikembangkan lagi oleh Bezdek di tahun 1981. FCM adalah metode pengelompokan dengan pendekatan *fuzzy*, yaitu setiap data yang dikelompokkan dimungkinkan menjadi anggota lebih dari satu kelompok. Pada makalah ini, FCM digunakan untuk mengelompokkan data *log* pengguna CoMa sesuai dengan karakteristik pengguna. Proses pengelompokan ini dilakukan secara terpisah dengan menggunakan *middleware* berbasis Python. Arsitektur sistem akan dijelaskan pada bagian metodologi.

V. METODOLOGI

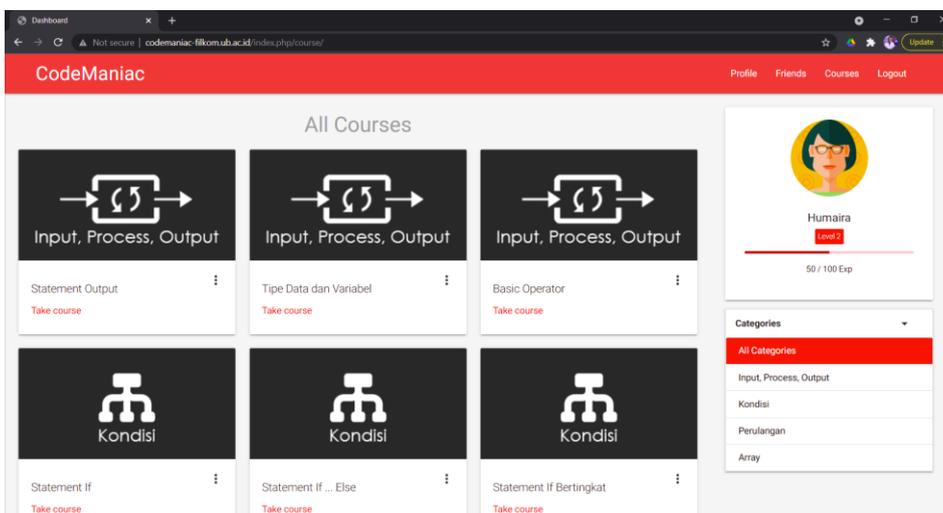
Pada makalah ini dilakukan beberapa langkah yang tertuang pada metodologi penelitian seperti pada Gbr. 4. Langkah-langkah tersebut meliputi analisis kode CoMa, perancangan arsitektur sistem, pengumpulan data *log*, *clustering*, penerapan antarmuka, dan pengujian.

A. Analisis Kode CoMa

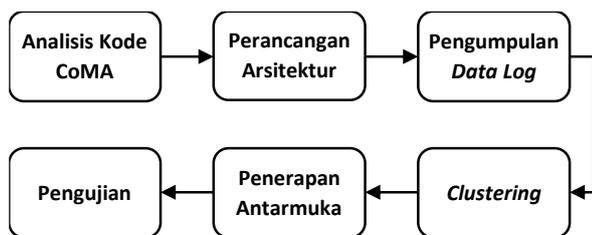
Pada tahapan ini dilakukan analisis kode dengan memahami dokumentasi perancangan sistem CoMa. Hal dilakukan untuk melihat kemungkinan penambahan modul tambahan serta menganalisis dampak adanya tambahan fitur terhadap kinerja



Gbr. 2 Halaman utama CoMa.



Gbr. 3 Halaman latihan.



Gbr. 4 Metode penelitian.

sistem secara keseluruhan. Tabel I menunjukkan gambaran singkat mengenai sistem CoMa. CoMa dikembangkan dengan menggunakan bahasa pemrograman PHP beserta framework Codeigniter versi 3.1.3. Basis data yang digunakan adalah MySQL, sedangkan untuk antarmuka halaman pengguna digunakan *framework* MaterializeCSS dan untuk antarmuka halaman admin digunakan Bootstrap.

Berdasarkan pengamatan terhadap dokumentasi perancangan sistem yang mengacu pada *class diagram* dan *sequence diagram* CoMa, disimpulkan bahwa sistem ini memiliki tingkat kompleksitas yang tinggi (> 5) dan *coupling*

TABEL I
PROFIL KODE CoMA

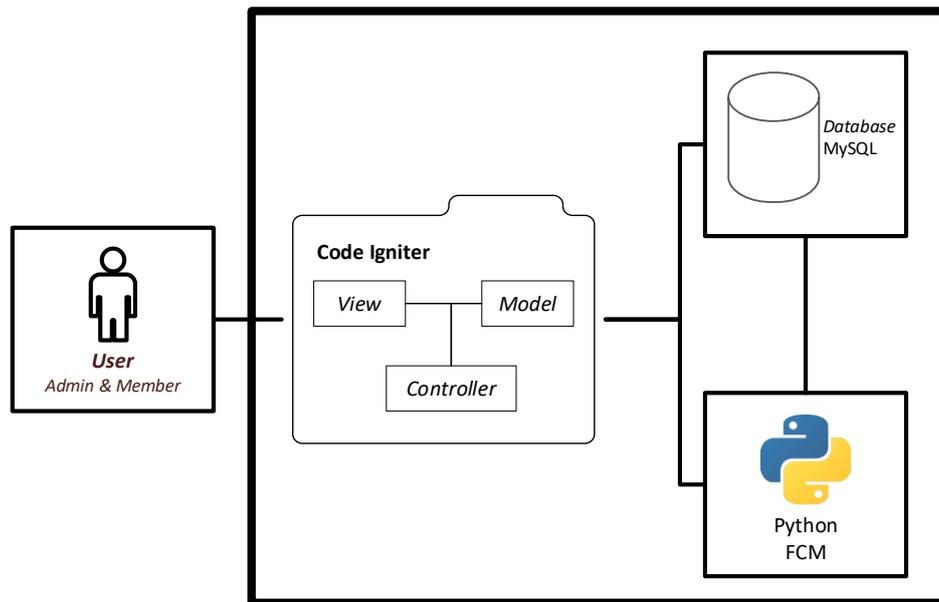
Nama Komponen	Spesifikasi
Bahasa pemrograman	PHP
Framework	Codeigniter
User interface	MaterializeCSS dan Bootstrap
Model perancangan	Object oriented analysis and design
Basis data	MySQL

TABEL II
NILAI KUALITAS KODE CoMA

Parameter	Nilai
Kekohesifan	0,95
Coupling	0,89
Kompleksitas	8

antar komponen yang juga tinggi (skala 0-1). Tabel II menunjukkan rata-rata nilai kekohesifan, *coupling*, dan kompleksitas pada diagram perancangan CoMa.

Tingkat kompleksitas dan *coupling* antar komponen menjadi indikasi kualitas perancangan perangkat lunak [9]. Tingkat *coupling* yang tinggi ini menjadikan proses penambahan fitur



Gbr. 5 Arsitektur yang diusulkan.

pada sistem akan memiliki dampak yang signifikan terhadap fitur yang telah berjalan sebelumnya. Maka, dengan pertimbangan tersebut, pada makalah ini dibangun aplikasi penengah (*middleware*) yang bertugas mengambil data *log* serta melakukan komputasi *clustering* untuk menentukan karakteristik dari pengguna.

B. Perancangan Arsitektur Sistem

Untuk menjadikan CoMa sebagai sistem yang adaptif, diperlukan sebuah mekanisme untuk merekam perilaku pengguna selama menggunakan sistem dalam periode waktu tertentu. CoMa telah memiliki mekanisme perekaman data *log* yang dapat diakses melalui halaman admin. Pada Gbr. 5 terlihat mekanisme *middleware* yang bertugas untuk mengambil rekaman pengguna untuk kemudian dibagi menjadi *cluster* menggunakan algoritme *fuzzy C-Means*. *Middleware* ini dikembangkan dengan bahasa pemrograman Python yang akan berjalan secara otomatis pada *background* tanpa mengganggu kinerja sistem CoMa. Setiap *cluster* akan memiliki tampilan yang berbeda sesuai dengan pengaturan pada sistem.

C. Pengumpulan Data Log

Data *log* yang dipertimbangkan pada makalah ini adalah data perilaku pengguna berupa aktivitas-aktivitas yang telah dilakukan pengguna dalam sistem pembelajaran CoMa. Untuk dapat mengembangkan *e-learning* adaptif, sistem perlu dapat menggali informasi yang menunjukkan tipe belajar pengguna, dengan melihat perilaku pengguna, tujuan pembelajaran pengguna, pola pembelajaran, tingkat pengetahuan, dan bakat pengguna. Adapun informasi perilaku pengguna yang dipertimbangkan mengacu pada Tabel III. *Middleware* akan mencatat setiap kali pengguna mengklik salah satu perilaku tersebut. Jumlah klik pada setiap perilaku akan dijumlahkan dan menjadi nilai untuk dilakukan proses pengelompokan menggunakan algoritme *fuzzy C-Means*.

TABEL III
PERILAKU PENGGUNA

No	Perilaku Pengguna
1	Accept request friend
2	Check exercise
3	Compile exercise
4	Confirm challenge
5	Detail exercise
6	Login
7	Logout
8	Search friends
9	Send request
10	Show profile
11	Show challenge
12	Show exercise
13	Show friends
14	Submit exercise
15	Take challenge
16	Take course
17	Take exercise
18	Test exercise

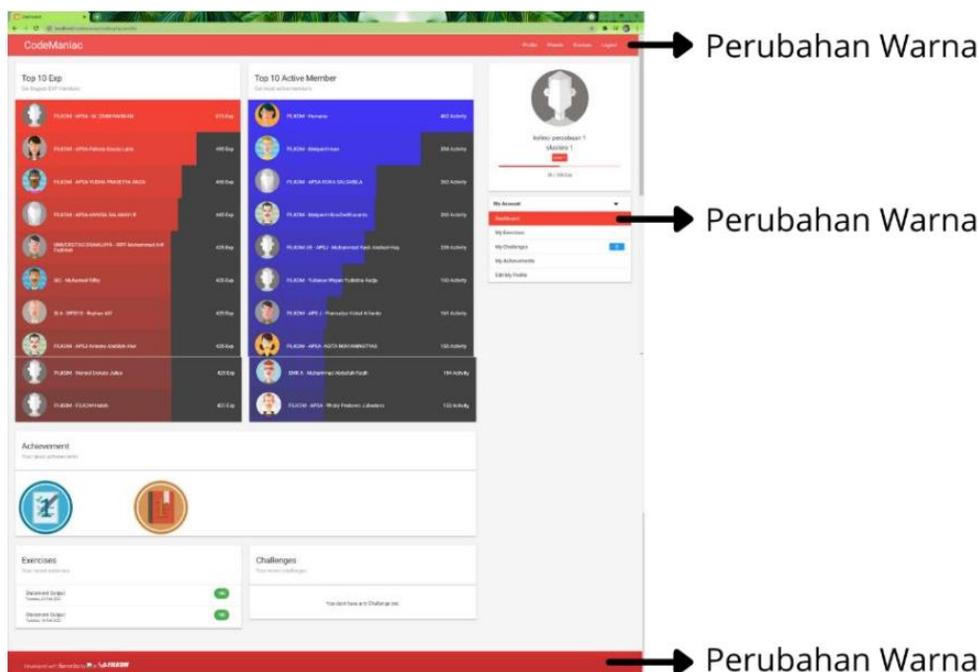
D. Clustering

Setelah mendapatkan data *log*, tahap berikutnya adalah melakukan proses *clustering*. *Pseudocode* metode *clustering* adalah sebagai berikut.

```

Begin Deklarasi
variable command = berisi fungsi escapeshellcmd
untuk menjalankan python serta direktori script
clustering;
Deklarasi variable output = menjalankan fungsi
shell_exac dengan parameter command;
deklarasi variable 'data' = menjalankan fungsi get dari
model cluster;
mengarahkan kehalaman view 'admin/header';
mengarahkan kehalaman view 'admin-cluster-list', dan
menampilkan data;
mengarahkan kehalaman view 'admin-footer';
end

```



Gbr. 6 Perancangan antarmuka.

TABEL IV
PERUBAHAN ANTARMUKA

Cluster	Warna/Tema Antarmuka
1	Merah
2	Biru
3	Hijau

E. Penerapan Antarmuka

Berdasarkan hasil *clustering*, terdapat tiga kelompok karakteristik pengguna. Dari pembagian kelompok tersebut, dirancang antarmuka yang berbeda untuk setiap kelompok. Nantinya setiap pengguna akan mendapatkan antarmuka yang berbeda tergantung pada kelompok yang telah diperhitungkan oleh sistem, seperti ditunjukkan pada Gbr. 6. Pada makalah ini, dirancang tiga usulan antarmuka dengan pembeda warna seperti diperlihatkan pada Tabel IV. Perubahan antarmuka dibatasi hanya sampai perubahan warna. Perubahan ini hanya sebagai penanda/respons ketika proses integrasi berhasil mengenali perilaku pemain. Untuk penelitian berikutnya, antarmuka perlu dibuat lebih detail agar memberikan respons yang lebih kompleks daripada sekadar mengubah warna.

F. Pengujian

Setelah perancangan dan implementasi dilakukan, sistem masuk pada tahap pengujian. Tahap ini meliputi pengujian akurasi *middleware* dalam melakukan pengelompokan dan pengujian validasi sistem secara keseluruhan

VI. HASIL DAN PEMBAHASAN

A. Penerapan FCM pada Middleware

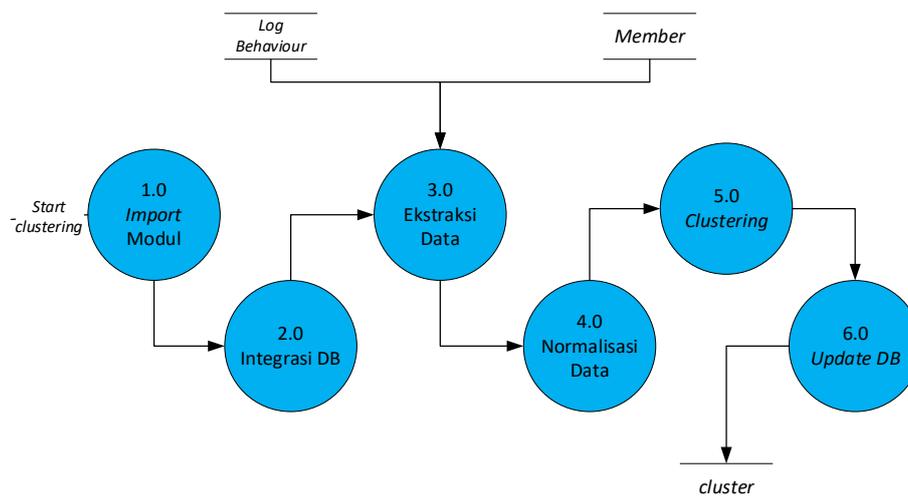
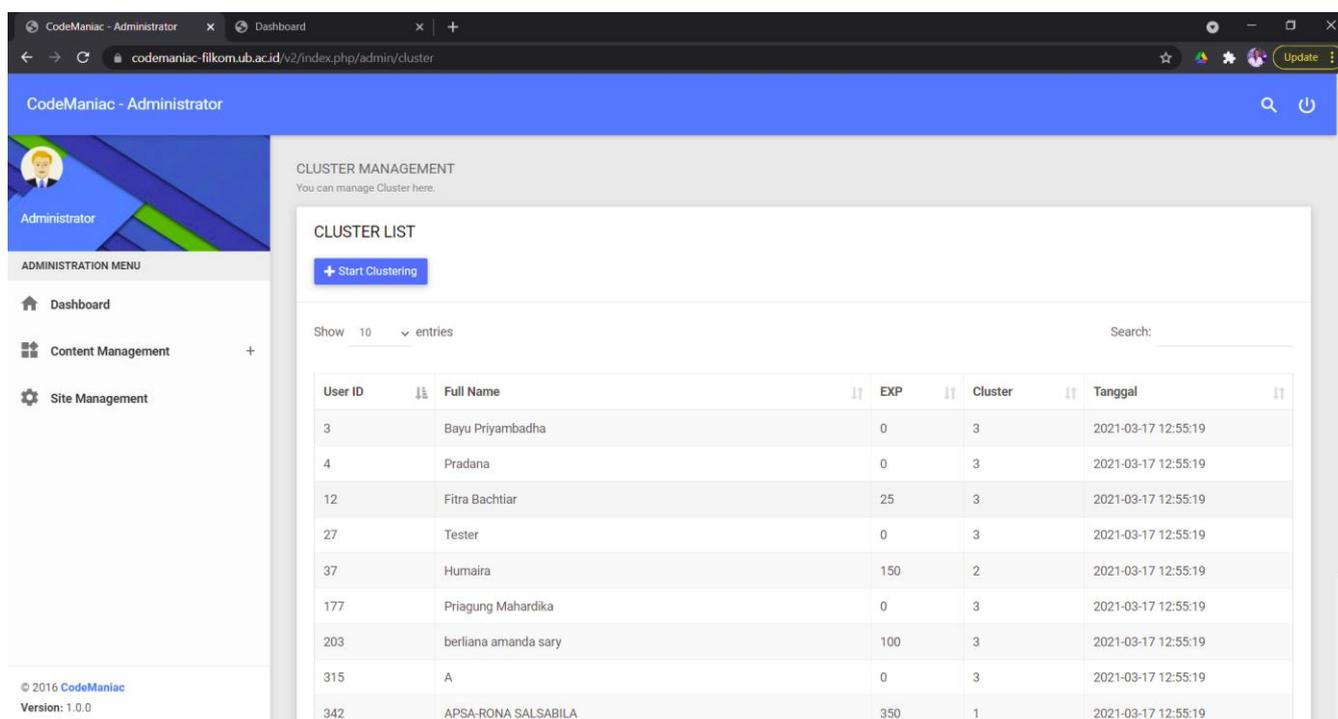
Sebagaimana perancangan pada Gbr. 5, proses integrasi melibatkan aplikasi *middleware* untuk melakukan pengolahan

data *log* sekaligus melakukan pengelompokan menjadi kelompok-kelompok yang memiliki kedekatan. Pada makalah ini *middleware*, dikembangkan dengan bahasa Python. Algoritme yang digunakan untuk mengelompokkan data *log* adalah *Fuzzy C-Means*. Algoritme diawali dengan deklarasi variabel *command* yang digunakan untuk menyimpan perintah dalam menjalankan Python dan direktori *script* yang akan dieksekusi. Kemudian, terdapat variabel data yang berbentuk *array*, yang digunakan untuk menyimpan data dari fungsi *Get*, lalu digunakan untuk menampilkan halaman, mulai dari *header*, *cluster-list*, sampai dengan *footer*. Implementasi algoritme ini mengacu pada Gbr. 7.

Setelah CoMa menerima perintah untuk melakukan *clustering*, *middleware* akan mulai mengimpor modul-modul yang diperlukan. Kemudian, proses berikutnya adalah melakukan koneksi dengan basis data sehingga dapat memulai ekstraksi data *log* yang berasal dari tabel *log behaviour* dan tabel *member* milik sistem warisan. Proses normalisasi data terlebih dahulu dilakukan sebelum pengolahan menggunakan *fuzzy C-Means*. Selanjutnya, data hasil pengelompokan dikembalikan lagi ke dalam basis data, sehingga dapat ditampilkan di halaman admin. Gbr. 8 merupakan hasil implementasi halaman admin yang menunjukkan hasil *clustering* data.

B. Implementasi Antarmuka Adaptif

Pada makalah ini diterapkan antarmuka yang mampu beradaptasi dan berubah sesuai dengan *cluster* dari pemain yang menggunakan sistem. Merujuk pada desain antarmuka pada Gbr. 6 dan Tabel III, diperlukan mekanisme untuk menjalankan fungsi-fungsi pergantian antarmuka tersebut. *Sequence diagram* yang menjadi acuan dalam implementasi antarmuka adaptif diperlihatkan pada Gbr. 9. Gbr 10 di

Gbr. 7 Fungsi *clustering*.Gbr. 8 Implementasi *clustering* pada halaman admin.

halaman terakhir makalah menunjukkan hasil implementasi antarmuka apabila seorang *member* termasuk dalam *cluster 2*. Implementasi antarmuka ini berpengaruh pada warna *header*, *sidebar*, dan *footer* seluruh halaman yang dapat diakses *member* [10].

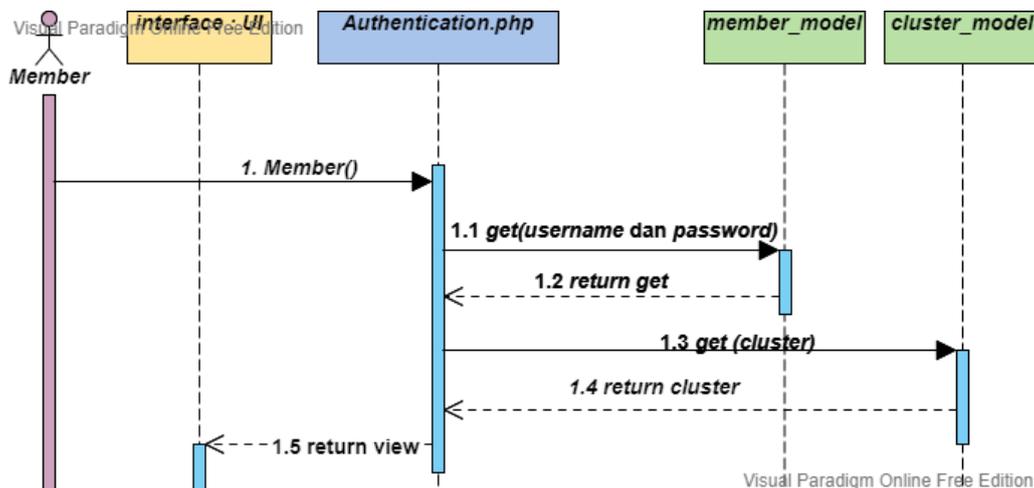
C. Pengujian Kinerja dan Fungsionalitas

Proses komputasi dilakukan secara terjadwal setiap hari pada pukul 23:59. Hal ini dilakukan agar jalannya sistem utama tidak terganggu. Hasil pengujian kinerja menunjukkan bahwa dengan jumlah data *log* mencapai 950 pengguna aktif, dibutuhkan waktu 23,35 ms. Dari hasil uji fungsionalitas, sistem berhasil menampilkan antarmuka sesuai dengan *cluster member*: jika termasuk *cluster 1*, antarmuka berwarna merah;

cluster 2 menjadi warna biru; dan *cluster 3* atau belum mendapatkan *cluster* akan berwarna hijau. Selain itu, untuk memastikan sistem mampu berjalan pada berbagai jenis *browser*, telah dilakukan pengujian kompatibilitas dengan menggunakan aplikasi SortSite.

VII. KESIMPULAN

Penerapan *e-learning* adaptif berdasarkan perilaku pengguna dilakukan dengan menerapkan antarmuka yang berbeda untuk setiap karakter pengguna. Untuk penelitian selanjutnya, diperlukan pengembangan agar sistem dapat adaptif mengarahkan kepada model soal dan tantangan yang dihadapi. Proses komputasi dalam mengolah data *log* dilakukan oleh *middleware* yang berjalan pada *background*. Hasil pengujian



Gbr. 9 Perancangan perubahan antarmuka.

mencapai waktu 23,35 ms untuk setiap proses *clustering*. Secara fungsional, sistem mampu bekerja normal tanpa mengganggu fungsi inti CoMa. *E-learning* adaptif diharapkan dapat memberikan pengalaman yang berbeda kepada pengguna, sehingga dapat meningkatkan motivasi dalam pembelajaran.

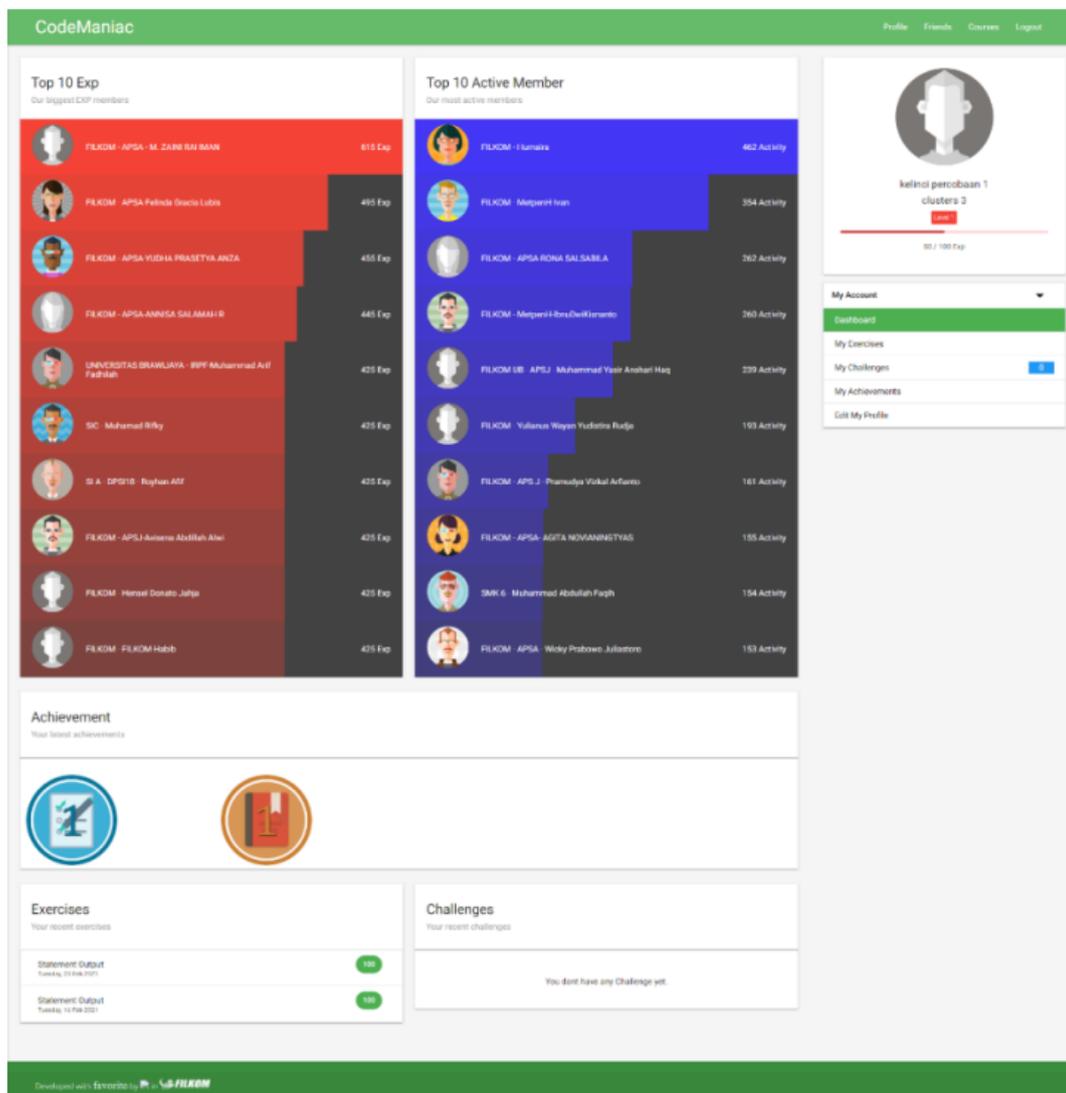
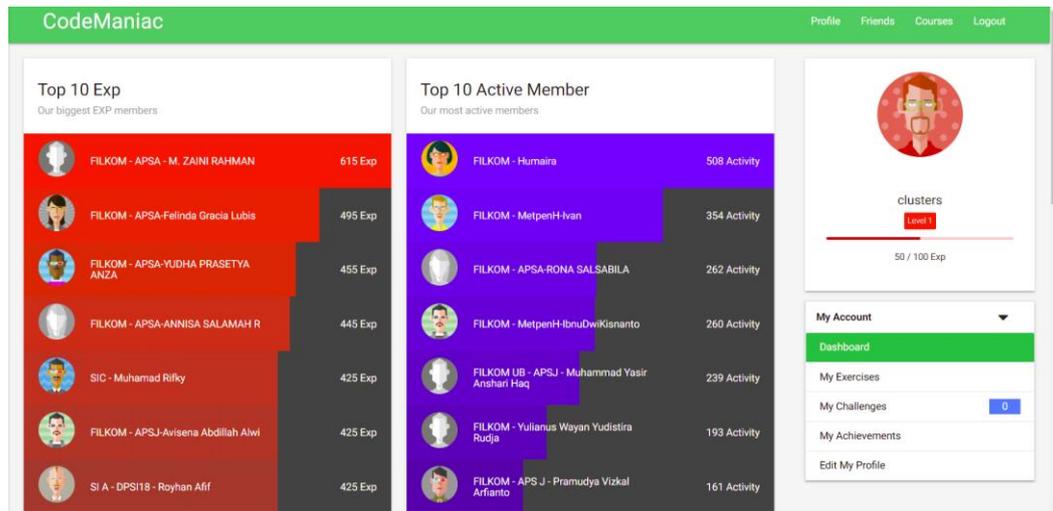
UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Achmad Hanim Nur Wahid yang telah membantu proses implementasi dan pengujian *e-learning* adaptif CoMa.

REFERENSI

[1] T. Belawati, *Pembelajaran Online*, Edisi kesatu, Tangerang, Indonesia: Universitas Terbuka, 2019.
 [2] B. Dhupia dan A. Alameen, "Adaptive eLearning System: Conceptual Framework for Personalized Study Environment," dalam *Adv. Inf. Comput. Res. (ICAICR 2019), Commun. in Comput. Inf. Sci.*, Vol 1075, A.K. Luhach, D.S. Jat, K.B.G. Hawari, X.-Z. Gao, P. Lingras, Eds., Cham, Swiss: Springer, 2019, hal. 334-342.
 [3] D. Hansen dan L.A. Imse, "Student-Centered Classrooms: Past Initiatives, Future Practices," *Music Edu. J*, Vol 103, No. 2, hal 20-26, 2016.

[4] F.A. Bachtiar, F. Pradana, B. Priyambadha, dan D.I. Bastari, "CoMa: Development of Gamification-based E-learning," *2018 10th Int. Conf. Inf. Technol. Elect. Eng. (ICITEE 2018)*, 2018, hal. 266-271.
 [5] Y. Kritikou, P. Demestichas, E. Adamopoulou, K. Demestichas, M. Theologou, dan M. Paradia, "User Profile Modeling in the Context of Web-based Learning Management Systems," *J. Netw. Comput. Appl.*, Vol 31, No. 4, hal. 603-627, 2008.
 [6] B. Priyambadha, F. Pradana, dan F.A. Bachtiar, "Penggalian Perilaku Pemain dalam Penentuan Tipe Permainan pada E-Learning Pemrograman Berbasis Gamification," *J. Teknol. Inf. Ilmu Komput.*, Vol. 7, No. 4, hal. 765-772, 2020.
 [7] R.M. Felder, dan R. Brent, "Understanding Student Difference," *J. Eng. Edu.*, Vol. 94, No. 1, hal. 57-72, 2005.
 [8] J. Simões, R.D. Redondo, dan A.F. Vilas, "A Social Gamification Framework for a K-6 Learning Platform," *Comput. Hum. Behav.*, Vol. 29, No. 2, hal. 345-353, 2013.
 [9] I. Chowdhury, dan M. Zulkernine, "Using Complexity, Coupling, and Cohesion Metrics as Early Indicators of Vulnerabilities," *J. Syst. Archit.*, Vol. 57, No. 3, hal. 294-313, 2011.
 [10] N. Abasi, N. Azad, dan K.F. Hafashjani, "Information Systems Success: The Quest for the Dependent Variable," *Uncertain Suppl. Chain Manag.*, Vol. 3. No. 2, hal. 181-188, 2015.



Gbr. 10 Perubahan antarmuka.