Jurnal Teknik Ilmu dan Aplikasi
ISSN: 2460-5549 | E-ISSN: 2797-0272

APLIKASI PEMBELAJARAN REKONSTRUKSI ALGORITMA PSEUDOCODE DENGAN PENDEKATAN *ELEMENT FILL-IN-BLANK PROBLEMS* DI PEMROGRAMAN JAVA

Vivin Ayu Lestari [1)], Rakhmat Arianto[2)], Baskoro Singgih Anindito[3)], Yuni Kurnia Taramita[4)], Eka Larasati Amalia[5)], Odhitya Desta Triswidrananta[6)]
[1),2),4),5),6)] Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang
[3)] Program Studi Teknik Komputer dan Jaringan, Jurusan Teknik Komputer dan Informatika, SMKN 9 Malang
[1)]*vivin@polinema.ac.id*
[2)]*arianto@polinema.ac.id*
[3)]*baskorosinggih@gmail.com*
[4)]*kurniataramita186@gmail.com*
[5)]*eka.larasati@polinema.ac.id*
[6)]*odhitya.desta@gmail.com*

Abstrak

Algoritma dasar pemrograman merupakan sebuah ilmu yang mempelajari dasar-dasar dalam memecahkan suatu masalah dalam membuat suatu program di komputer dengan urutan yang logis, serta tahapan dan kaidah yang sesuai. Namun, saat ini mahasiswa masih mengalami kesulitan dalam memahami konsep algoritma dan melakukan analisis studi kasus. Oleh karena itu, perlu dikembangkan suatu media pembelajaran rekonstruksi algoritma pseudocode dengan pendekatan Element Fill-in-Blank Problems di pemrograman java. Pada media pembelajaran yang dikembangkan ini berisikan studi kasus dan elemen kosong pada langkah-langkah algoritma pseudocode, dimana mahasiswa mengurutkan langkah-langkah penyelesaian dan mengisi dengan melakukan drag and drop ke elemen kosong tersebut. Berdasarkan pengujian fungsional menggunakan *blackbox*, didapatkan hasil dimana semua kasus uji mendapatkan hasil sesuai dengan hasil yang diharapkan yang berarti bahwa telah valid dengan tingkat validitas mencapai 100%. Selain pengujian fungsional system juga dilakukan pengujian usability. Hasil pengujian usability menggunakan SUS (*System Usability Scale*) diperoleh hasil 81,25 yang artinya sistem bisa diterima. Dalam hal skala penilaian, aplikasi ini memiliki nilai "B" dan termasuk kategori "Baik". Berdasarkan hasil uji usability ini, aplikasi rekonstruksi algoritma pseudocode layak untuk digunakan dan diimplementasikan sebagai aplikasi pembelajaran.

Kata Kunci: Algoritma, *Element Fill-in-Blank Problem* (EFP), Java, Pemrograman, *Pseudocode*.

1. PENDAHULUAN

Algoritma pemrograman dasar adalah sebuah ilmu yang mempelajari dasar-dasar dalam memecahkan suatu masalah dalam membuat suatu program di komputer dengan urutan yang logis, serta tahapan dan kaidah yang sesuai [1]. Dasar pemrograman juga merupakan salah satu mata kuliah yang wajib dikuasai oleh mahasiswa jurusan Teknik Informatika. Tujuannya mata kuliah ini adalah agar mahasiswa memiliki kemampuan dalam memecahkan masalah ke dalam suatu program untuk mendukung mata kuliah lain yang membutuhkan kemampuan pemrograman [2].

Pemrograman java merupakan salah satu bahasa pemrograman yang banyak dipelajari oleh masyarakat khususnya mahasiswa Teknik Informatika. Pada tahun 2016 berdasarkan TIOBE Index, Bahasa Pemrograman Java menduduki posisi pertama dan menduduki posisi kedua untuk jumlah repository di Github dibawahnya Javascript [3]. Pada tahun 2018 bahasa pemrograman java terpilih sebagai program yang memiliki banyak pengguna, karena telah banyak digunakan di berbagai bidang seperti aplikasi clouds, machine learning dan IOT karena keandalannya yang tinggi [4]. Selain itu, dalam dunia industri saat ini banyak membutuhkan programer dengan keahlian pemrograman java [5].

Sehingga sangat penting bagi mahasiswa untuk mempersiapkan kemampuan dan skill pada pemrograman java.

Dalam mempelajari dasar-dasar algoritma dan pemrograman, saat ini mahasiswa masih bingung dan kesulitan dalam menerjemahkan studi kasus kedalam sebuah program, salah satu diantaranya adalah masih kurangnya media pembelajaran tambahan sebagai media belajar mandiri di luar pembelajaran di sekolah [6]. Model pembelajaran yang ada saat ini masih berupa materi dan latihan pemrograman di komputer, sehingga membuat mahasiswa kurang paham untuk menerjemahkan studi kasus kedalam sebuah program [4]. Selain itu permasalahan yang lain adalah kurangnya kemampuan mahasiswa dalam menyusun kode program dan melakukan analisis studi kasus [7].

Aplikasi pembelajaran yang sudah dikembangkan dalam pemrograman diantaranya adalah JPLAS (Java Programming Learning Assistant) dengan pedekatan Element Fill-in-Blank Problem (EFP), yang mana mahasiswa dapat mengisi bagian yang kosong dalam sebuah kode program yang benar dan menerapkan aturan sintaks

untuk masing-masing pernyataan. Dengan pedekatan EFP dalam pembelajaran pemrograma java dapat meingkatkan pengetahuan dasar pemrograman java secara efektif. Aplikasi yang lainnya adalah PYPLAS (*Python Programming Learning Assistant System*) [8] dan CPLAS (*C Programming Learning Assistant System*) [9]. Kedua aplikasi tersebut menggunakan pendekatan EFP yang digunakan untuk meningkatkan keterampilan pemrograman mahasiswa, sekaligus untuk belajar tata bahasa dan dasar pemrograman [8]. Dengan memodelkan blank algoritma yang harus diisi dengan memperhatikan konsep tata bahasa pemrograman, agar elemen kosong tersebut terisi dengan jawaban yang benar, serta terdapat berbagai jenis soal sesuai dengan tingkat kesulitan dari soal [9]. Hasil dari penelitian ini adalah pembelajaran dengan pendekatan *Element Fill-in-Blank Problem* ini dinilai efektif untuk mengatasi persoalan dalam mempelajari pemrograman baik menggunakan pyhton, java maupun bahasa C [9]. Selain itu pembelajaran pemrograman dengan pendekatan *Element Fill-in-Blank Problems* ini juga memiliki manfaat untuk meningkatkan keterampilan pemrograman mahasiswa [8].

Aplikasi lainnya tentang pembelajaran yang telah dikembangkan adalah tentang konsep *drag and drop* pada media pembelajaran menyebutkan bahwa kurangnya minat belajar siswa saat ini karena media belajar berupa buku dan *ebook* masih kurang efektif dan efisien untuk menunjang pembelajaran yang interaktif dan meningkatkan pemahaman serta kemampuan terhadap materi yang diberikan [1]. Aplikasi drag and *drop* ini dikembangakan untuk pembelajaran matematika dengan hasil pengujian *User Acceptance Testing* adalah sangat baik.

Berdasarkan penjelasan diatas, maka dibutuhkan suatu media pembelajaran yang dapat membantu mahasiswa untuk belajar tentang algoritma dasar pemrograman dalam memecahkan studi kasus. Pada penelitian ini penulis membuat suatu aplikasi pembelajaran rekonruksi algoritma pseudocode dengan pendekatan *Element Fill-in-Blank Problem* Di Pemrograman Java. Aplikasi ini berisikan studi kasus dan penyelesaian algoritma dalam bentuk *pseudocode*, dimana mahasiswa diminta untuk mengurutkan langkah-langkah penyelesaian tersebut dengan mengisi langkah-langkah yang kosong serta menentukan tipe data sesuai dengan yang dibutuhkan. Dalam melakukan rekonstruksi algoritma tersebut dapat dilakukan dengan *drag* and *drop* ke elemen yang kosong. Dengan adanya aplikasi ini diharapkan dapat meningkatkan pemahaman mahasiswa dalam menyelesaikan studi kasus dalam bentuk algoritma *pseudocode* sebelum dituliskan ke dalam kode program.

2. ANALISIS DAN PERANCANGAN

2.1 Analisis Kebutuhan

Analisa kebutuhan fungsional berisikan proses atau fungsi yang harus dikerjakan oleh sistem untuk melayani kebutuhan pengguna. Analisis kebutuhan fungsional dalam pengembangan Rekonstruksi Algoritma Pseudocode sebagai berikut :

a. Aplikasi diharapkan dapat mengelola data soal dan data mahasiswa.
b. Aplikasi dapat menampilkan soal-soal berdasarkan kategori yang dipilih
c. Aplikasi dapat digunakan sebagai media belajar algoritma pseudocode dengan konsep Element Fill-in-Blank Problem dan drag and drop

2.2 Perancangan Database

Perancangan tabel database merupakan pengaturan tabel pada database yang akan digunakan untuk menyimpan data. Struktur tabel dapat mempermudah dalam memasukkan data sesuai dengan tipe data yang telah ditentukan. Perancangan struktur tabel dari *database* adalah sebagai berikut :

TABEL 2.1. PERANCANGAN TABEL MAHASISWA

Nama Atribut	Tipe Data
Id_mahasiswa	Int
Nama	Varchar (50)
Nim	Char
email	Varchar (254)
Jenis_kelamin	Enum (‘L’,’P’)

TABEL 2.2. PERANCANGAN TABEL USER

Nama Atribut	Tipe Data
Id_admin	Int
Username	Varchar (100)
Password	Varchar (255)
Email	Varchar (255)
Created_on	Int
Active	Tinyint
First_name	Varchar (50)
Last_name	Varchar (50)

TABEL 2.3. PERANCANGAN TABEL GRUP

Nama Atribut	Tipe Data
Id_grup	Int
Nama	Varchar (20)
Description	Varchar (100)

TABEL 2.4. PERANCANGAN TABEL USER GRUP

Nama Atribut	Tipe Data
Id	Int
Id_user	Int
Id_grup	Int

TABEL 2.5. PERANCANGAN TABEL SOAL

Nama Atribut	Tipe Data
Id_soal	Int
Id_level	Int
Bobot	Int
Soal	Longtext
judul	Varchar (255)
Opsi_	Longtext
Urut_	Varchar (255)

Clue_	Varchar (255)
Created_on	Int
Updated_on	Int
Variable_	Varchar (100)
Jenis_data_	Varchar (100)

TABEL 2.6. PERANCANGAN TABEL LEVEL

Nama Atribut	Tipe Data
Id_level	Int
Nama	Varchar (100)
Image	Varchar (255)
Bts_nilai	Int

TABEL 2.7. PERANCANGAN TABEL NILAI

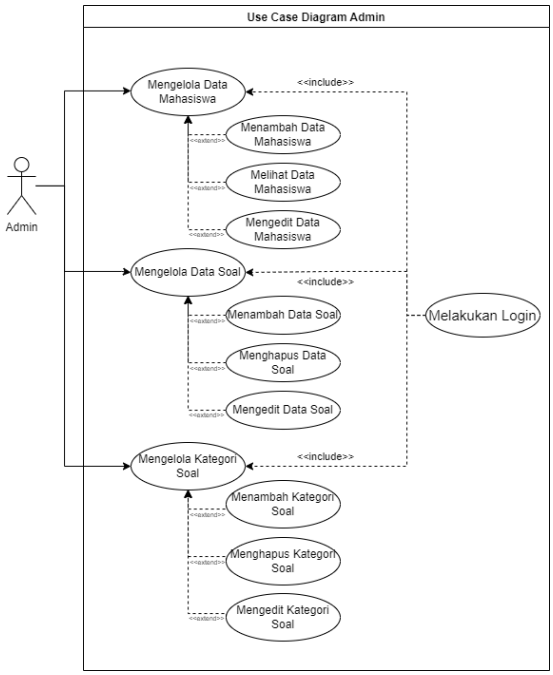
Nama Atribut	Tipe Data
Id_nilai	Int
Id_user	Int
Id_soal	Int
Nilai	Int

TABEL 2.7. PERANCANGAN TABEL HISTORY UJIAN

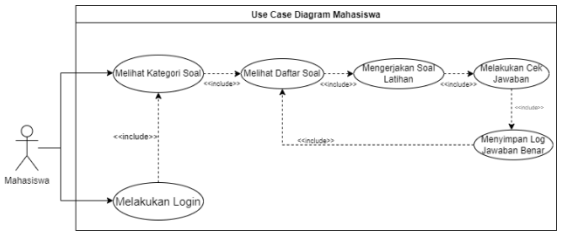
Nama Atribut	Tipe Data
Id	Int
Id_user	Int
Id_soal	Int

2.3 Perancangan Use Case Diagram

Perancangan Use Case Diagram merupakan tahap merancang atau mendesain suatu system atau aplikasi dimana akan dibahas seperti rancangan database dan juga rancangan interface untuk pengguna. Pada gambar dibawah dijelaskan bahswa terdapat 2 aktor untuk masing-masing use case diagram yaitu mahasiswa dan admin, yang mana admin bertugas untuk mengelola website, sedangkan mahasiswa sebagai pengakses atau pengguna. Perancangan usecase aplikasi ini dapat dilihat pada Gambar 1 dan 2.



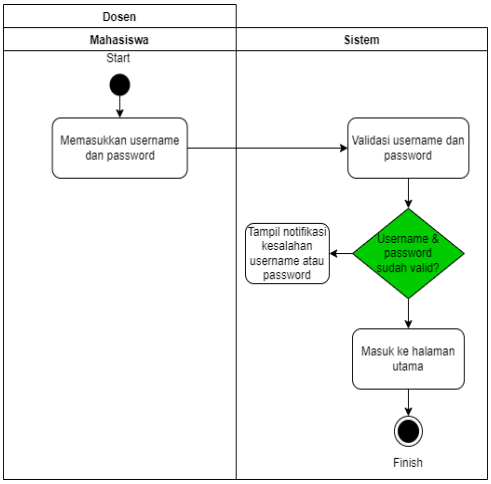
Gambar 1. Use Case Diagram Admin



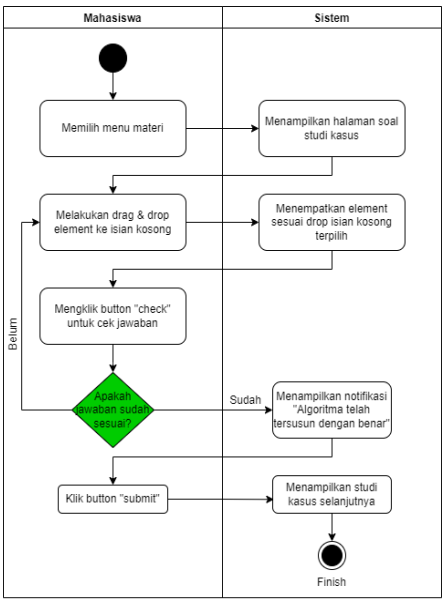
Gambar 2. Use Case Diagram Mahasiswa

2.4 Perancangan Activity Diagram

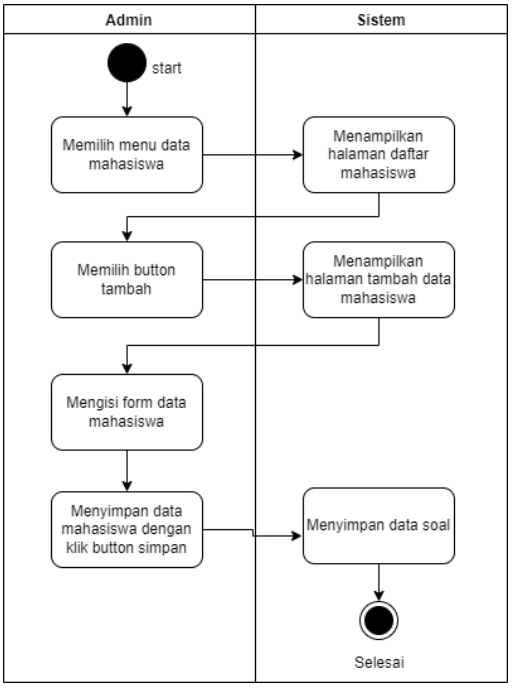
Perancangan activity diagram digunakan untuk menggambarkan semua aktifitas secara keseluruhan dalam sistem. Pada sistem terdapat 5 activity diagram yang akan dibuat, yaitu melakukan login, mengerjakan soal Latihan studi kasus, menambah data mahasiswa oleh admin, menambah soal, menambah kategori soal.



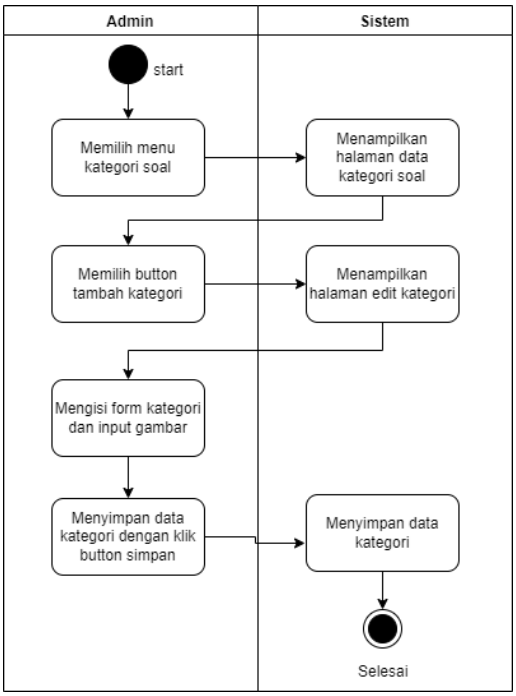
Gambar 3. Activity Diagram Login



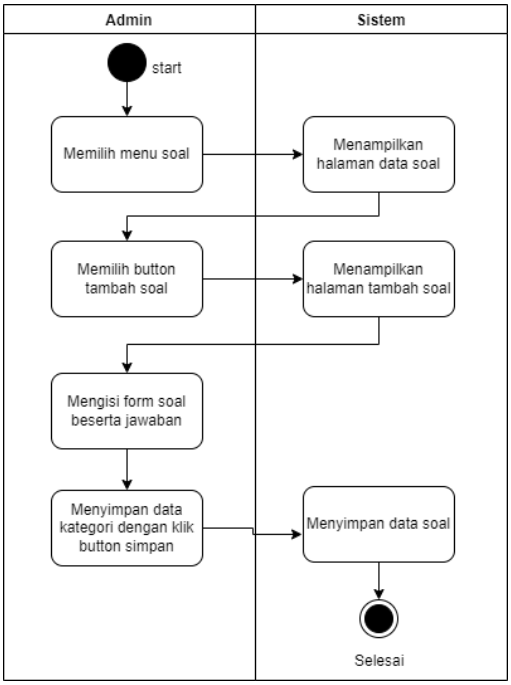
Gambar 4. Activity Diagram Belajar Algoritma Studi Kasus



Gambar 5. Menambahkan User Mahasiswa



Gambar 6. Activity Diagram Menambah Kategori Soal

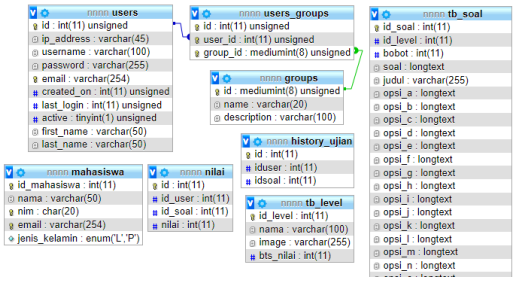


Gambar 7. Activity Diagram Menambah Soal

3. IMPLEMENTASI

3.1 Implementasi Database

Implementasi database adalah hasil dari Analisa dan perancangan yang telah dilakukan pada bab sebelumnya. Database yang digunakan terdiri dari 8 tabel, yaitu groups, mahasiswa, nilai, tb_level, tb_soal, users, dan users_groups. Hasil impelemntasi database dapat dilihat pada Gambar 8.

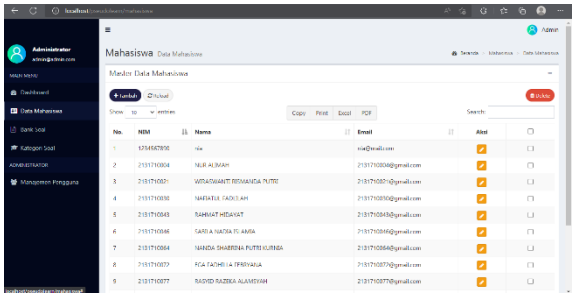


Gambar 8. Implementasi Database

3.2 Implementasi Interface

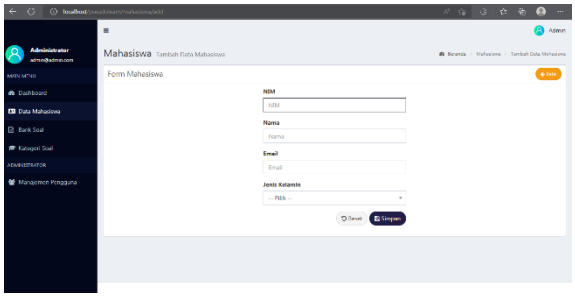
Pada tahap implementasi tampilan interface aplikasi rekontruksi algoritma pseudocode menggunakan pendekatan EFP ini sudah dirancang berdasarkan desain dan database sebelumnya. Pada user interface ini dibagi menjadi 2 tampilan yaitu admin dan mahasiswa. Gambar 9 merupakan tampilan halaman Login.

Gambar 9. Halaman Login

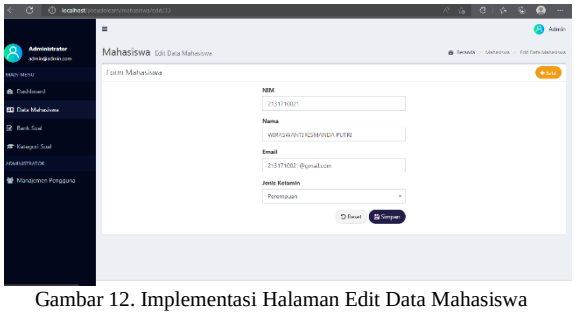


Gambar 10. Halaman Daftar Mahasiswa

Gambar 10 merupakan implementasi dari halaman daftar data mahasiswa. Pada halaman ini menampilkan nama mahasiswa, email dan username NIM yang digunakan sebagai password login. Admin dapat menginputkan data, mengedit dan menghapus mulai dari NIM, nama, email dan jenis kelamin yang dapat dilihat pada Gambar 11 dan 12.

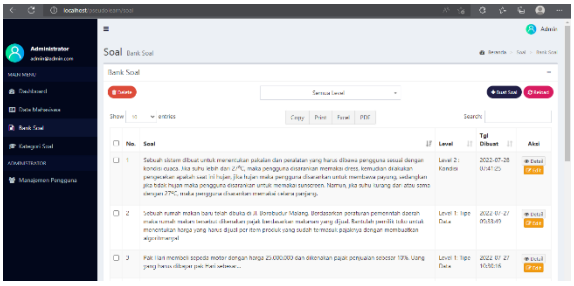


Gambar 11. Halaman Tambah Data Mahasiswa

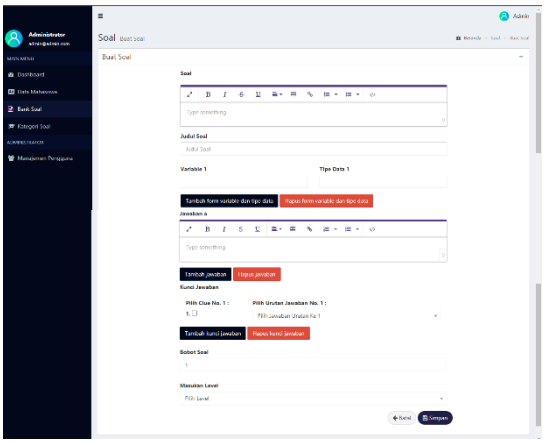


Gambar 12. Implementasi Halaman Edit Data Mahasiswa

Gambar 13 dan 14 merupakan implementasi dari halaman daftar soal pada halaman admin, yang menampilkan soal dan level soal. Pada halaman ini admin dapat melakukan tambah, edit, lihat detail dan hapus soal

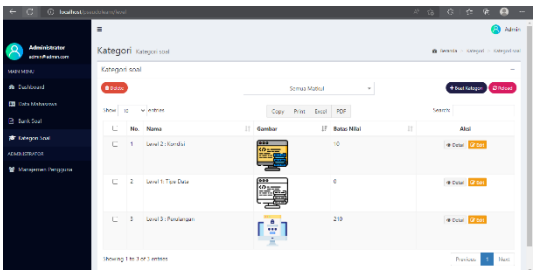


Gambar 13. Implementasi Dari Halaman Daftar Soal

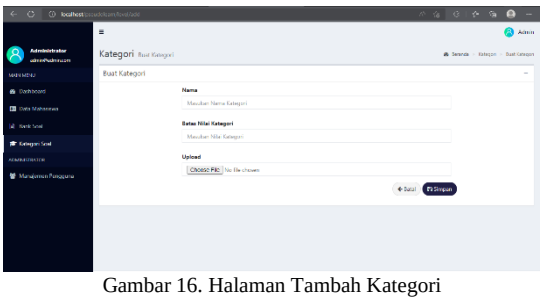


Gambar 14. Halaman Tambah Soal

Gambar 15 merupakan implementasi dari halaman daftar kategori. Terdapat beberapa informasi yang ditampilkan, antara lain nama kategori, gambar dan batas nilai. Pada Gambar 16 merupakan tampilan tambah, hapus dan edit kategori yang dapat dilakukan oleh admin.

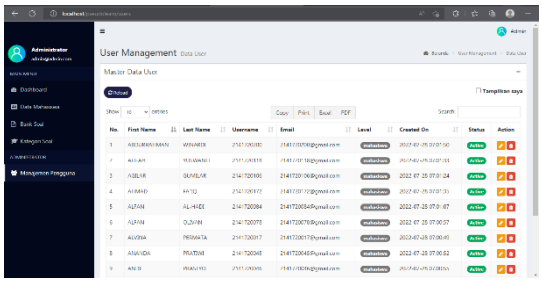


Gambar 15. Halaman Daftar Kategori

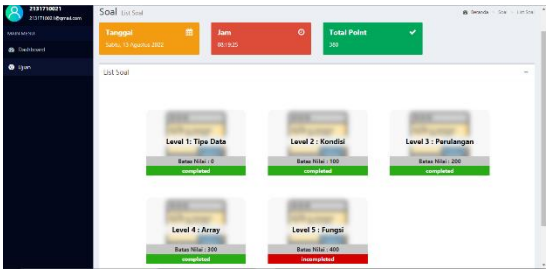


Gambar 16. Halaman Tambah Kategori

Gambar 17 merupakan implementasi dari halaman manajemen pengguna mahasiswa pada halaman admin. Pada halaman ini menampilkan nama mahasiswa, email dan username NIM yang digunakan sebagai password login

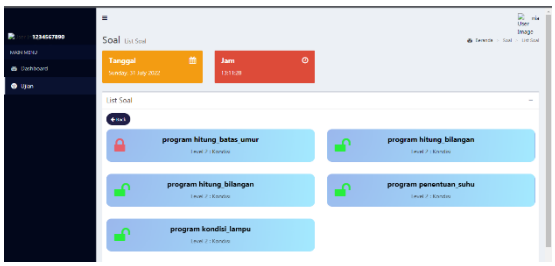


Gambar 17. Halaman Manajemen Pengguna.Mahasiswa



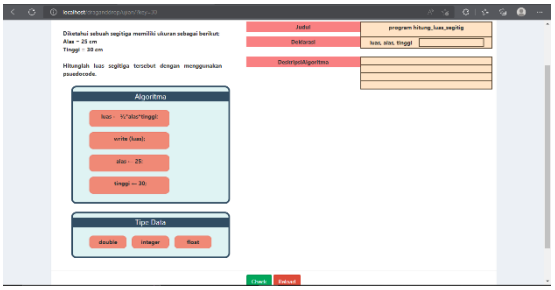
Gambar 18. Halaman Utama Mahasiswa

Pada Gambar 18 di atas merupakan halaman utama mahasiswa yang menampilkan beberapa materi. Dalam halaman tersebut materi yang dapat diakses hanya jika sudah muncul tulisan completed, yang mana apabila kita ingin mengakses materi-materi lanjutan lainnya, maka mahasiswa harus menyelesaikan studi kasus yang terdapat pada materi sebelumnya. Mahasiswa dapat memilih soal mana yang akan dikerjakan. Apabila telah berhasil menyelesaikan soal latihan studi kasus, maka soal tersebut akan otomatid terkunci dan tidak dapat dikerjakan kembali. Gambar 19 merupakan implementasi dari halaman daftar soal, pada halaman ini soal-soal sesuai dengan kategori yang dipilih sebelumnya.

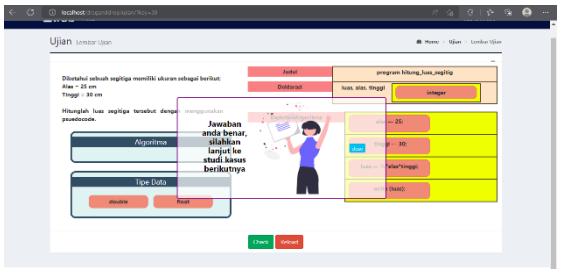


Gambar 19. Halaman Daftar Soal

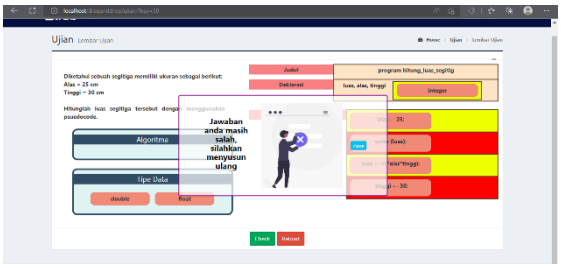
Pada Gambar 20 diatas merupakan halaman soal yang menampilkan studi kasus serta element kosong dari deklarasi dan skripsi/algoritma, yang nantinya mahasiswa diharuskan untuk mengisi dengan cara drag pilihan jawaban yang ada di tabel algoritma dan tipe data, kemudian drop jawaban ke element kosong tersebut. Selain itu terdapat button “Check” yang berfungsi untuk melihat apakah struktur algoritma sudah sesuai atau belum. Apabila studi kasus telah selesai dikerjakan maka klik button “submit” untuk melanjutkan ke soal selanjutnya. Gambar 21 merupakan implementasi tampilan pengecekan jawaban, apabila jawaban yang terisi sudah benar semua maka akan muncul notifikasi bahwa jawaban sudah benar dan elemen akan berwarna kuning. Sedangkan Gambar 22 merupakan implementasi tampilan pengecekan jawaban, apabila masih terdapat jawaban yang belum tepat maka elemen akan berwarna merah dan muncul notifikasi bahwa jawaban masih salah.



Gambar 20. Halaman Latihan Studi Kasus



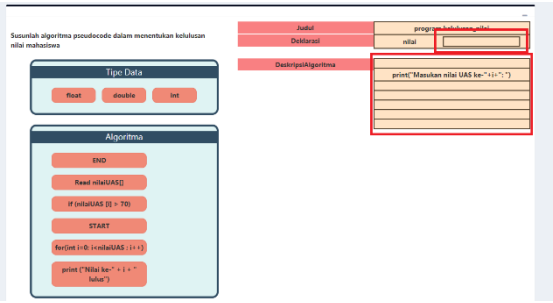
Gambar 21. Tampilan Cek Jawaban Benar



Gambar 22. Tampilan Cek Jawaban Salah

3.3 Pendekatan *Element Fill in the Blank Problem (EFP)*

Pada aplikasi rekontruksi algoritma *pseudocode* ini dibangun dengan pendekatan *Element Fill in the Blank Problem (EFP)*. EFP yang diterapkan di aplikasi ini dengan menyajikan urutan susunan algoritma *pseudocode* dari sebuah studi kasus yang berisis beberapa elemen kosong untuk menguji kemampuan pengguna dalam memahami penyelesaian studi kasus. Gambar 23 merupakan tampilan halaman studi kasus yang berisikan soal studi kasus, pilihan jawaban (sebelah kiri) yang harus disusun menjadi sebuah algoritma *pseudocode* yang benar (sebelah kanan) yang menerapkan pendekatan EFP.



Gambar 23. Tampilan Halaman Studi Kasus yang Menerapkan Pendekatan EFP

4. UJI COBA DAN ANALISIS

4.1 Pengujian Fungsional

Berdasarkan hasil dari pengujian sistem yang dilakukan di semua fitur menggunakan *blackbox* pada aplikasi rekontruksi algoritma pseudocode, didapatkan hasil seperti pada Tabel 4.1 dan 4.2. Setelah dilakukan pengujian pada tiap proses didapatkan hasil dimana semua kasus uji mendapatkan hasil sesuai dengan hasil yang diharapkan yang berarti bahwa telah valid dengan tingkat validitas mencapai 100%.

TABEL 4.1. HASIL UJI COBA SISTEM ADMIN

No	Fitur	Hasil	
		Berhasil	Tidak Berhasil
1	Masuk Akun Admin	√	
2	Lihat Tampilan Utama	√	
3	Lihat Daftar Mahasiswa	√	
4	TambahData Mahasiswa	√	
5	Edit Data Mahasiswa	√	
6	Hapus Data Mahasiswa	√	
7	Lihat Daftar Soal	√	
8	Tambah Data Soal	√	
9	Edit Data Soal	√	
10	Hapus Data Soal	√	
11	Lihat Daftar Kategori	√	
12	Tambah Data Kategori	√	
13	Edit Data Kategori	√	
14	Hapus Data Kategori	√	
15	Lihat Manajemen Pengguna	√	
16	Edit Data Pengguna	√	

TABEL 4.2. HASIL UJI COBA SISTEM MAHASISWA

No	Fitur	Hasil	
		Berhasil	Tidak Berhasil
1	MasukAkun Mahasiswa	√	
2	Lihat Tampilan Utama	√	
3	Locked/Unlocked Kategori	√	
4	Pilih Soal	√	
5	Lihat Halaman Latihan Soal	√	
6	Drag and drop elemen	√	
7	Cek Jawaban Salah	√	
8	Cek Jawaban Benar	√	
9	Soal Terkunci (Selesai Dikerjakan)	√	
10	Poin Nilai	√	
16	Edit Data Pengguna	√	

4.2 Pengujian Usability

Pengujian usability dilakukan dengan membagikan kuisioner kepada 28 pengguna SUS adalah kuesioner yang terdiri dari 10 item pertanyaan oleh Jhon Brooke [10], seperti yang ditunjukkan pada Tabel 4.4. Kuesioner SUS menggunakan skala Likert 5 poin. Responden diminta untuk memberikan penilaian “Sangat tidak setuju”, “Tidak setuju”, “Netral”, “Setuju”, dan “Sangat setuju”. Setiap item pernyataan memiliki skor kontribusi berkisar dari 1 sampai 5. Untuk item 1,3,5,7, dan 9 skor kontribusi adalah posisi skala dikurangi 1. Untuk item 2,4,6,8, dan 10, skor kontribusi adalah 5 dikurangi skala posisi. Kalikan total skor kontribusi dengan 2,5 untuk mendapatkan skor kegunaan sistem secara keseluruhan. Skor SUS berkisar dari 0 hingga 100. Skor SUS keseluruhan diperoleh dari rata-rata skor SUS individu. Berikut rumus untuk menghitung skor SUS:

$$Skor\ SUS = ((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1) + (5 - R8) + (R9 - 1) + (5 - R10)) * 2.5)$$

Hasil skor akhir SUS untuk masing-masing responden terlihat seperti pada Tabel 4.3. Tabel tersebut menunjukkan bahwa rata-rata skor kuesioner SUS adalah 81,25. Rata-rata skor inilah yang digunakan sebagai skor kriteria penerimaan pengguna terhadap sistem yang dikembangkan.

TABEL 4.3. KUISIONER *SYSTEM USABILITY SCALE* (SUS)

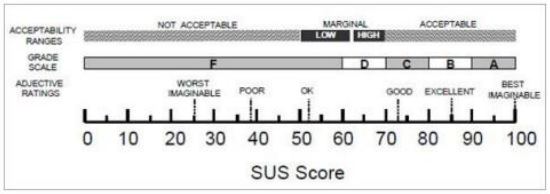
Kode	Pernyataan
S1	Saya berpikir akan menggunakan sistem ini lagi
S2	Saya merasa sistem ini rumit untuk digunakan
S3	Saya merasa sistem ini mudah digunakan
S4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
S5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
S6	Saya merasa ada banyak hal yang tidak konsisten
S7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
S8	Saya merasa sistem ini membingungkan
S9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
S10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

TABEL 4.4. HASIL PENGUJIAN USABILITY

R-i	**Skor-i**	**Skor SUS-i**	**R-i**	**Skor-i**	**Skor SUS-i**
R1	28	70,00	R15	33	82,50
R2	32	80,00	R16	36	90,00
R3	29	72.50	R17	32	80,00
R4	32	80,00	R18	35	87,50
R5	31	77,50	R19	30	75,00
R6	32	80,00	R20	31	77,50
R7	34	85,00	R21	30	75,00
R8	28	70,00	R22	30	75,00
R9	31	77,50	R23	35	87,50
R10	32	80,00	R24	34	85,00
R11	35	87,50	R25	36	90,00
R12	33	82,50	R26	36	90,00
R13	35	87,50	R27	32	80,00
R14	35	87,50	R28	33	82,50
Nilai Rata-Rata					81,25

Keterangan:
R-i: responden ke-i
Skor-i: total skor responden ke-i
Skor SUS -i: skor akhir ke-i

Langkah selanjutnya adalah mengubah nilai rata-rata SUS menjadi kategori tingkat penerimaan, skala penilaian, dan peringkat kata sifat. Kategori untuk skor SUS dikembangkan oleh Bangor. Kategori tingkat penerimaan diklasifikasikan menjadi “Tidak Dapat Diterima”, “Marjinal”, dan “Dapat Diterima”. Skala penilaian diklasifikasikan ke dalam kelas "A", "B", "C", "D", dan "F". Klasifikasi untuk peringkat kata sifat dibagi menjadi “Terburuk yang Dapat Dibayangkan”, “Buruk”, “OK”, “Baik”, “Luar Biasa”, dan “Best Imaginable”. Masing-masing didasarkan pada skor rata-rata SUS seperti yang ditunjukkan pada Gambar 24.



Gambar 24. Skor SUS menurut Bagor

Berdasarkan Gambar 24, suatu sistem dapat diterima jika memiliki skor SUS minimal 70. Oleh karena itu berdasarkan hasil pengujian usability, aplikasi rekonstruksi algoritma pseudocode memiliki skor SUS 81,25 yang artinya sistem bisa diterima. Dalam hal skala penilaian, aplikasi ini memiliki nilai “B” dan memiliki klasifikasi “Baik”. Berdasarkan hasil uji usability ini, aplikasi rekonstruksi algoritma pseudocode layak untuk digunakan dan diimplementasikan sebagai aplikasi pembelajaran, sehingga aplikasi ini dapat mememberikan kontribusi baik bagi pengajar maupun mahasiswa dalam proses kegiatan belajar mengajar pemrograman.

5. KESIMPULAN

Dari hasil yang didapatkan saat melakukan pengujian dan analisis maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi dirancang dengan melakukan analisis kebutuhan dengan hasil yang berupa gambaran umum sistem, identifikasi aktor, kebutuhan fungsional dan kebutuhan non fungsional. Setelah dilakukan analisis kebutuhan kemudian di lakukan perancangan sistem yang terdiri dari perancangan UML (activity diagram dan usecase diagram), perancangan basis data, dan perancangan antarmuka. Pengembangan aplikasi ini diimplementasikan dengan Bahasa pemrograman PHP dan basis data mySQL.
2. Berdasarkan hasil pengujian validasi pada aplikasi menunjukan nilai dengan persentase 100% yang berarti sistem sudah memenuhi kebutuhan fungsional. Sedangkan untuk tingkat usability dari aplikasi. Hasil pengujian usability menggunakan kuesioner SUS menunjukkan bahwa skor rata-rata kuesioner SUS adalah 81,25. Berdasarkan skor tersebut, penerapan rekonstruksi algoritma pseudocode dapat diterima sebagai sistem yang dapat diimplementasikan lebih lanjut. Selanjutnya berdasarkan konversi skor SUS menurut Bangor, aplikasi ini memiliki nilai B dan termasuk dalam peringkat Baik.
3. Aplikasi rekontruksi algoritma pseudocode dengan pendekatan EFP dapat memberikan kontribusi sebagai media pembelajaran pada mata kuliah pemrograman.

6. Daftar Pustaka

[1] M. F. Rivaldi and Y. I. Kurniawan, "Game Edukasi Pengenalan dan Pembelajaran Berhitung untuk Siswa Kelas 1 Sekolah Dasar," *J. Manaj. Inform.*, vol. 11, no. 1, pp. 47–59, 2021, doi: 10.34010/jamika.v11i1.4354.

[2] M. Yasin, "Computational Thinking Untuk Pembelajaran Dasar-Dasar Pemrograman KOMPUTER," *Researchgate*, no. April, pp. 0–11, 2020, [Online]. Available: https://www.researchgate.net/publication/340637723_COMPUTATIONAL_THINKING_UNTUK_PEMBELAJARAN_DASAR-DASAR_PEMROGRAMAN_KOMPUTER

[3] F. F. Alghifari, Ko. C. Brata, and A. A. Supianto, "Pengembangan Aplikasi Pembelajaran Untuk Mendukung Mahasiswa Dalam Belajar Pemrograman," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 9, pp. 3021–3028, 2020.

[4] S. S. Wint and N. Funabiki, *A proposal of recommendation function for element fill-in-Blank problems in java programming learning assistant system*, vol. 17, no. 2. Springer International Publishing, 2021. doi: 10.1108/IJWIS-11-2020-0070.

[5] N. Funabiki, Tana, K. K. Zaw, N. Ishihara, and W. C. Kao, "A graph-based blank element selection algorithm for fill-in-blank problems in Java programming learning assistant system," *IAENG Int. J. Comput. Sci.*, vol. 44, no. 2, pp. 247–260, 2017.

[6] W. A. P. Kusumadewi, "Pengembangan Media Pembelajaran Berbasis Simulasi Pada Mata Pelajaran Perkaitan Komputer Untuk Siswa Kelas X di SMK Negeri 3 Surabaya," *It-Edu*, vol. 1, no. 01, 2016.

[7] L. S. Muchlis, K. Rukun, and K. Krismadinata, "Efektifitas Pengembangan Model Diva Learning Manajemen System Pada Matakuliah Algoritma Dan Pemrograman," *J. Pendidik. Teknol. Kejuru.*, vol. 3, no. 2, pp. 104–108, 2020, doi: 10.24036/jptk.v3i2.7123.

[8] H. W. Hnin and K. K. Zaw, "Element Fill-in-Blank Problems in Python Programming Learning Assistant System," *Proc. 4th Int. Conf. Adv. Inf. Technol. ICAIT 2020*, pp. 88–93, 2020, doi: 10.1109/ICAIT51105.2020.9261778.

[9] H. H. S. Kyaw, N. Funabiki, S. L. Aung, N. K. Dim, and W. C. Kao, "A Study of Element Fill-in-Blank Problems for C Programming Learning Assistant System," *Int. J. Inf. Educ. Technol.*, vol. 11, no. 6, pp. 255–261, 2021, doi: 10.18178/ijiet.2021.11.6.1520.

[10] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," *Usability Eval. Ind.*, no. July, pp. 207–212, 1996, doi: 10.1201/9781498710411-35.