

BAB IV PEMBAHASAN

4.1 Gambaran Umum Obyek Penelitian

Perpustakaan STIKI Malang menyediakan beberapa layanan seperti: sirkulasi buku, pemesanan buku, pengusulan pengadaan buku, peminjaman scanner, hingga penjiilidan tugas akhir / PKL. Berikut ini merupakan pemaparan visi dan misi perpustakaan STIKI Malang.

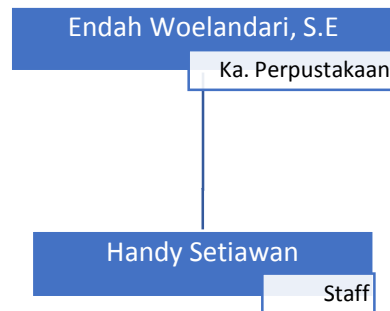
Visi:

1. Menjadi pusat literasi informasi yang berkualitas dan diakui di Indonesia dan dunia internasional.

Misi:

1. Menyediakan kebutuhan pustaka yang relevan sesuai kebutuhan pemustaka
2. Mengembangkan repository lokal konten (deposit) yang *open access*
3. Menyelenggarakan pelayanan prima yang memenuhi standar
4. Mengembangkan sistem otomasi perpustakaan
5. Mengembangkan kompetensi pustakawan
6. Mengembangkan *total quality management* dan perpustakaan yang terakreditasi

Struktur Organisasi



Tugas dan Wewenang:

1. Ka. Perpustakaan
 - a. Menyusun program pengembangan perpustakaan
 - b. Mengorganisasikan sumber daya dan dana
 - c. Menyusun tata tertib perpustakaan
 - d. Menyusun laporan berkala
2. Staff
 - a. Layanan sirkulasi atau peminjaman buku
 - b. Mengklasifikasi buku perpustakaan
 - c. Memberikan pelayanan kepada pengunjung

4.2 Implementasi

4.2.1 Spesifikasi Produk

Sistem ini membutuhkan spesifikasi *hardware* maupun *software* berikut agar dapat berjalan baik. Spesifikasi berikut ini adalah sebuah spesifikasi rekomendasi agar dapat menjalankan sistem secara lancar.

a. Rekomendasi hardware

Tabel 4.1 Daftar Spesifikasi Rekomendasi Perangkat Keras

No.	Komponen	Spesifikasi
1.	CPU	Intel Core I5
2.	RAM	4 GB
3.	VGA	-
4.	GPU	-
5.	Hard Disk	32 GB
6.	Monitor	Resolusi 1980 x 1024

b. Rekomendasi software

Tabel 4.2 Daftar Spesifikasi Rekomendasi Perangkat Lunak

No.	Nama	Deskripsi	Versi
1.	OS	Sebagai sistem operasi	Windows 11 / Ubuntu 22.10
2.	PHP	Sebagai kebutuhan <i>backend</i> sistem	7.4.8
3.	MariaDB	Sebagai <i>database</i>	10.4.13
4.	Apache	Sebagai <i>web server</i>	2.4.43

Spesifikasi diatas merupakan spesifikasi yang disarankan untuk menjalankan sistem ini. Namun spesifikasi tersebut tidak selalu bersifat mutlak. Sistem ini dapat berjalan dengan minimum spesifikasi berikut ini:

a. Kebutuhan minimum hardware

Tabel 4.3 Daftar Spesifikasi Minimum Perangkat Keras

No.	Komponen	Spesifikasi
1.	CPU	Pentium IV
2.	RAM	512 MB
3.	VGA	-
4.	GPU	-
5.	Hard Disk	2 GB ruang kosong untuk penyimpanan program dan file pendukung
6.	Monitor	Resolusi 1024 x 768

b. Kebutuhan minimum software

Tabel 4.4 Daftar Spesifikasi Minimum Perangkat Lunak

No.	Nama	Deskripsi	Versi
1.	OS	Sebagai sistem operasi	Windows 8 / Ubuntu 18.04
2.	PHP	Sebagai kebutuhan <i>backend</i> sistem	7.4.8
3.	MariaDB	Sebagai <i>database</i>	10.4.13
4.	Apache	Sebagai <i>web server</i>	2.4.43

infrastruktur sistem dan analisis kebutuhan. Untuk membangun tampilan depan (*front-end*), *framework* Vue.JS 3 digunakan sebagai *framework* utama. Selain itu, beberapa *library* juga digunakan untuk mendukung fungsionalitas sistem informasi yang akan dijelaskan pada Segmen Program 4.1. Setelah itu, diperlukannya inisialisasi *library* secara global agar *library* dapat digunakan seutuhnya yang akan dijelaskan pada Segmen Program 4.2.

Loc	Code
1	{
2	"name": "STIKI Malang - Digilib Gamifikasi",
3	"version": "1.0.0",
4	"private": true,
5	"type": "module",
6	"scripts": {
7	"dev": "vite",
8	"build": "vite build",
9	"preview": "vite preview",
10	"lint": "eslint . --ext .vue,.js,.jsx,.cjs,.mjs --fix --ignore-path
11	.gitignore",
12	"format": "prettier --write src/"
13	},
14	"dependencies": {
15	"@fortawesome/fontawesome-svg-core": "^6.5.1",
16	"@fortawesome/free-brands-svg-icons": "^6.5.1",
17	"@fortawesome/free-regular-svg-icons": "^6.5.1",
18	"@fortawesome/free-solid-svg-icons": "^6.5.1",
19	"@fortawesome/vue-fontawesome": "^3.0.6",
20	"@tato30/vue-pdf": "^1.9.6",
21	"@tweenjs/tween.js": "^23.1.1",
22	"axios": "^1.6.8",
23	"bootstrap": "^5.3.3",
24	"buffer": "^6.0.3",
25	"dayjs": "^1.11.10",
26	"multi-range-slider-vue": "^1.1.4",
27	"pinia": "^2.1.7",
28	"vee-validate": "^4.12.6",
29	"vue": "^3.4.21",
30	"vue-router": "^4.3.0",
	"vue3-radial-progress": "^1.1.1"

```

31   },
32   "devDependencies": {
33     "@rushstack/eslint-patch": "^1.10.1",
34     "@vitejs/plugin-vue": "^5.0.4",
35     "@vue/eslint-config-prettier": "^9.0.0",
36     "eslint": "^8.57.0",
37     "eslint-plugin-vue": "^9.24.0",
38     "postcss-rename": "^0.6.1",
39     "prettier": "^3.2.5",
40     "sass": "^1.72.0",
41     "terser": "^5.30.1",
42     "vite": "^5.2.7"
43   }
44 }

```

Segmen Program 4.1 Segmen Program Pada File package.json

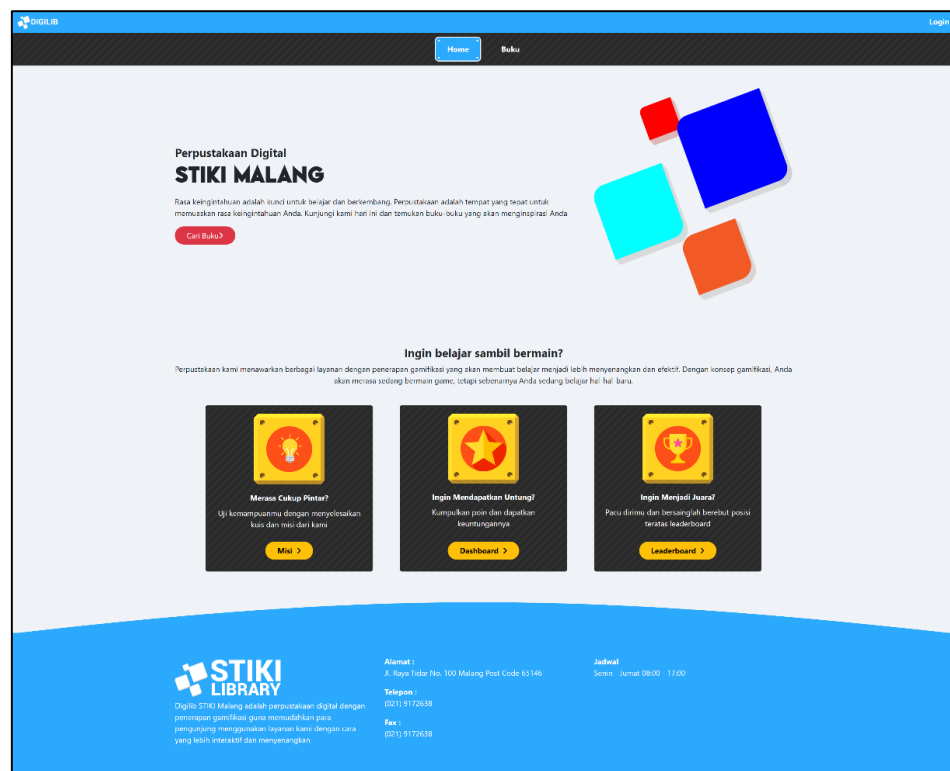
Loc	Code
1	import 'bootstrap/dist/css/bootstrap.min.css';
2	import 'bootstrap';
3	
4	import { createApp } from 'vue'
5	import { createPinia } from 'pinia'
6	
7	import App from './App.vue';
8	import router from './router';
9	import helper from './helpers';
10	
11	import EmptyLayout from './views/layouts/EmptyLayout.vue';
12	import HomepageLayout from './views/layouts/HomepageLayout.vue';
13	import AppLayout from './views/layouts/AppLayout.vue';
14	
15	const app = createApp(App)
16	.use(createPinia())
17	.use(router)
18	.use(helper)
19	.component('empty-layout', EmptyLayout)
20	.component('homepage-layout', HomepageLayout)
21	.component('app-layout', AppLayout)
22	
23	app.mount('#app')

Segmen Program 4.2 Segmen Program Pada File main.js

Setelah menyelesaikan persiapan pembuatan sistem informasi. Maka selanjutnya adalah menjelaskan implementasi setiap fitur yang telah didefinisikan. Seluruh kode sumber dapat diakses di *Github* melalui tautan: https://github.com/farispurnomo/skripsi_gamification_in_digilib.git

a) Halaman *homepage*

Halaman ini ditampilkan secara statis tanpa tanpa memerlukan API atau *webservice*. Listing kode program tercantum dalam Lampiran 5.1. Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.2.



Gambar 4.2 Implementasi Halaman Homepage

b) Halaman login

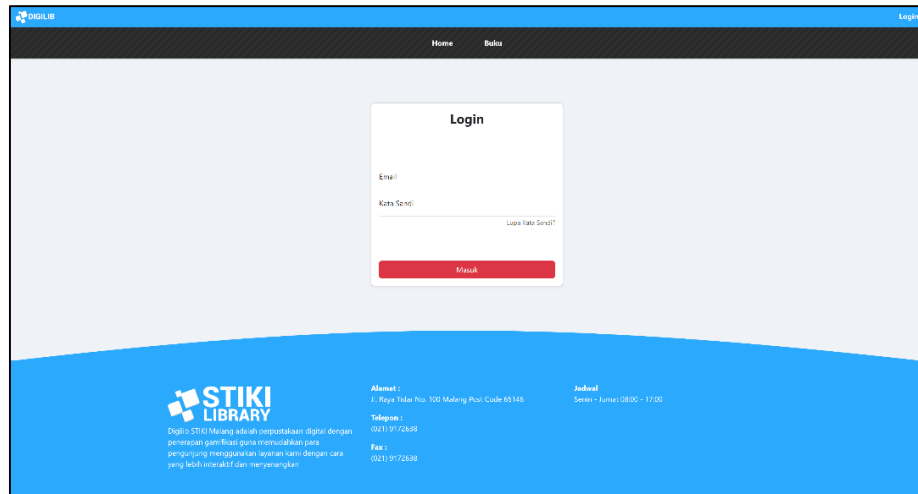
Halaman ini bertujuan untuk mengautentikasi pemain. Pemain memasukkan data yang kemudian dikirimkan ke server untuk verifikasi, jika data valid, *server* akan mengembalikan data profil pemain dan token yang kemudian

keduanya akan disimpan pada *LocalStorage* browser. Token tersebut digunakan untuk autentikasi pada setiap *request* selanjutnya. Cuplikan kode program tercantum dalam segmen program 4.3 dibawah ini.

LoC	Code
1	<script setup>
2	import \$axios from '@api/api';
3	import useAuthStore from '@stores/auth';
4	import { ref } from 'vue';
5	import { useRouter } from 'vue-router';
6	
7	const router = useRouter();
8	const authStore = useAuthStore();
9	const is_loading = ref(false);
10	const error = ref(null);
11	
12	const onSubmit = async function (values) {
13	is_loading.value = true;
14	error.value = null;
15	\$axios.post('auth/login', values)
16	.then((response) => {
17	if (response.data.is_success) {
18	authStore.setToken(response.data.data.token);
19	authStore.setUser(response.data.data.user);
20	router.push({ name: 'app.dashboard.index' });
21	} else {
22	error.value = response.data.msg;
23	}
24	is_loading.value = false;
25	}).catch((response) => {
26	error.value = response?.response?.data?.msg response.message;
27	is_loading.value = false;
28	});
29	};
30	</script>

Segmen Program 4.3 Segmen Program Halaman Login

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.3.



Gambar 4.3 Implementasi Menu Login

c) Halaman Beranda

Halaman ini menyajikan informasi detail pemain yang diambil dari *webservice*. Cuplikan kode program tercantum dalam segmen program 4.4 dibawah ini.

LoC	Code
1	<code><script setup></code>
2	<code>import { onMounted, reactive, ref } from 'vue';</code>
3	<code>import useAuthStore from '@stores/auth';</code>
4	<code>import \$axios from '@api/api';</code>
5	
6	<code>const authStore = useAuthStore();</code>
7	<code>const point_current = ref(0);</code>
8	<code>const point_max = ref(0);</code>
9	
10	<code>const dashboard = reactive({</code>
11	<code> is_loading: true,</code>
13	<code> leaderboards: [],</code>
14	<code> activities: [],</code>
15	<code> summary: {</code>
16	<code> finished_books: 0,</code>
17	<code> completed_quests: 0,</code>
18	<code> }</code>
19	<code>});</code>
20	
21	<code>const requestDashboard = function () {</code>

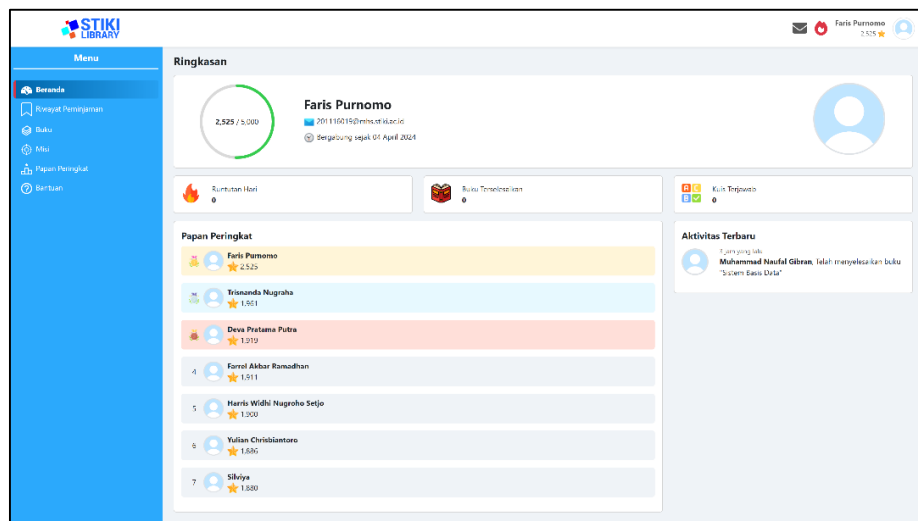
```

22     dashboard.is_loading = true;
23     $axios.get('dashboard')
24     .then(async response => {
25         dashboard.is_loading = false;
26         const data = response.data.data;
27         if (response.data.is_success) {
28             authStore.setUser(data.user);
29             dashboard.leaderboards = data.leaderboards;
30             dashboard.activities = data.activities;
31             point_current.value = data.point.current;
32             point_max.value = data.point.total;
33             dashboard.summary.finished_books =
34             data.summary.finished_books;
35             dashboard.summary.completed Quests =
36             data.summary.completed_quests;
37         }
38     })
39     .catch(() => dashboard.is_loading = false );
40 };
41 onMounted(() => requestDashboard() );
42 </script>

```

Segmen Program 4.4 Segmen Program Halaman Beranda

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.4.



Gambar 4.4 Implementasi Menu Beranda

d) Halaman riwayat peminjaman

Halaman ini menampilkan informasi riwayat peminjaman buku fisik pemain yang diperoleh dari *webservice*. Data ditampilkan dalam bentuk

tabel dan telah terpaginasi. Pemain dapat menekan baris peminjaman buku untuk diarahkan ke menu detail peminjaman. Cuplikan kode program tercantum dalam segmen program 4.5 dibawah ini.

Loc	Code
1	<script setup>
2	import \$axios from '@api/api';
3	import { onMounted, ref } from 'vue';
4	
5	const is_loading = ref(true);
6	const current_page = ref(1);
7	const total_items = ref(0);
8	const item_per_page = ref(1);
9	const offset = ref(1);
10	const items = ref([]);
11	
12	const requestData = function () {
13	is_loading.value = true;
14	\$axios
15	.post('loan/list', {
16	page: current_page.value
17	})
18	.then(async response => {
19	is_loading.value = false;
20	if (response.data.is_success) {
21	offset.value = response.data.data.from;
22	total_items.value = response.data.data.total;
23	item_per_page.value = response.data.data.per_page;
24	items.value = response.data.data.data;
25	}
26	})
27	.catch(() => is_loading.value = false);
28	};
29	onMounted(() => {
30	requestData();
31	});
31	</script>

Segmen Program 4.5 Segmen Program Halaman Riwayat Peminjaman

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.5.

#	No. Referensi	Buku	Tanggal Peminjaman	Tanggal Harus Kembali	Status
1.	P-2402559	Cathedral And The Bazaar: Musings On Linux And Open Source By An Accidental Revolutionary	2012-08-05 08:17:48	2022-03-03	DIBALIKKAN
2.	P-240269	The Magic Of Illustrator	2011-04-21 04:39:15	2014-12-26	DIBALIKKAN
3.	P-2402489	Peminjaman Deep Learning Dengan Python	2020-09-10 13:16:46	2016-12-03	DIBALIKKAN
4.	P-2402739	Metodologi Penelitian Pendidikan Karya Edisi Revisi	2006-06-29 16:00:21	2016-04-15	DIBALIKKAN
5.	P-2402779	Buku Dasar Nirmala: Asas dan Prinsip Dasar Seni Visual	2005-11-15 15:00:39	2013-12-14	DIBALIKKAN
6.	P-2402376	Laravel Uncover: Panduan Belajar Framework Laravel 10	2022-06-25 07:59:53	2013-05-30	DIBALIKKAN
7.	P-2402977	Aplikasi Analisis Data Sensor Dengan Instrumental, Potlato, Chart Pada Arduino Uno	2021-11-23 22:24:39	2021-02-08	DIBALIKKAN
8.	P-2402630	Prinsip Dasar Perencanaan Dan Pembuatan Film Animasi	1998-12-30 19:19:53	2007-01-11	DIBALIKKAN
9.	P-2402854	Cathedral And The Bazaar: Musings On Linux And Open Source By An Accidental Revolutionary	1998-04-29 00:41:10	2021-05-11	DIBALIKKAN
10.	P-2402496	And i-b: Mini Its balanced Processor Multimedia	1991-09-05 11:55:56	2022-01-10	DIBALIKKAN

Gambar 4.5 Implementasi Menu Riwayat Peminjaman

e) Halaman buku pemain

Halaman ini menyajikan daftar buku berdasarkan hasil filter. Data pada halaman ini diperoleh dari *webservice*. Cuplikan kode program tercantum dalam segmen program 4.6 dibawah ini.

Loc	Code
1	<script setup>
2	import \$axios from '@api/api';
3	import '@assets/css/multi-range-slider-custom.scss';
4	import { onMounted, reactive, ref } from 'vue';
5	
6	const barMinValue = ref(0);
7	const barMaxValue = ref(0);
8	const filter = reactive({
9	keyword: '',
10	sort_by: 'publish-year-newest',
11	tahun_publikasi_awal: null,
12	tahun_publikasi_akhir: null,
13	publishers: [],
14	publisher_keyword: '',
15	categories: [],
16	category_ids: [],
17	is_fisik: true,
18	is_digital: true
19	});

```

20 const mode = {
21   GRID: 'GRID',
22   LIST: 'LIST'
23 };
24 const book = reactive({
25   is_loading: true,
26   list: [],
27   mode: mode.LIST,
28   total: 0,
29   paging: {
30     current_page: 1,
31     total_items: 0,
32     items_per_page: 1,
33     offset: 1
34   }
35 });
36 const requestBook = function () {
37   book.is_loading = true;
38   $axios.post('book/search', {
39     keyword: filter.keyword,
40     category_ids: filter.category_ids,
41     publisher_ids: filter.publishers.filter((x) =>
42 x.checked).flatMap(x => x.id),
42     sort_by: filter.sort_by,
43     is_digital: filter.is_digital,
44     is_fisik: filter.is_fisik,
45     tahun_publikasi_awal: filter.tahun_publikasi_awal,
46     tahun_publikasi_akhir: filter.tahun_publikasi_akhir,
47     page: book.paging.current_page
48   }).then(response => {
49     book.is_loading = false;
50     if (response.data.is_success) {
51       const regex = new RegExp(filter.keyword, 'ig');
52       const replacement = `<mark>${filter.keyword}</mark>`;
53       book.list = response.data.data.data.map(function (item) {
54         let title = item.title;
55         if (filter.keyword !== '')
56           title = item.title.replaceAll(regex, replacement);
57         item.title = title;
58         return item;
59       });
60       book.paging.total_items = response.data.data.total;
61       book.paging.items_per_page = response.data.data.per_page;

```

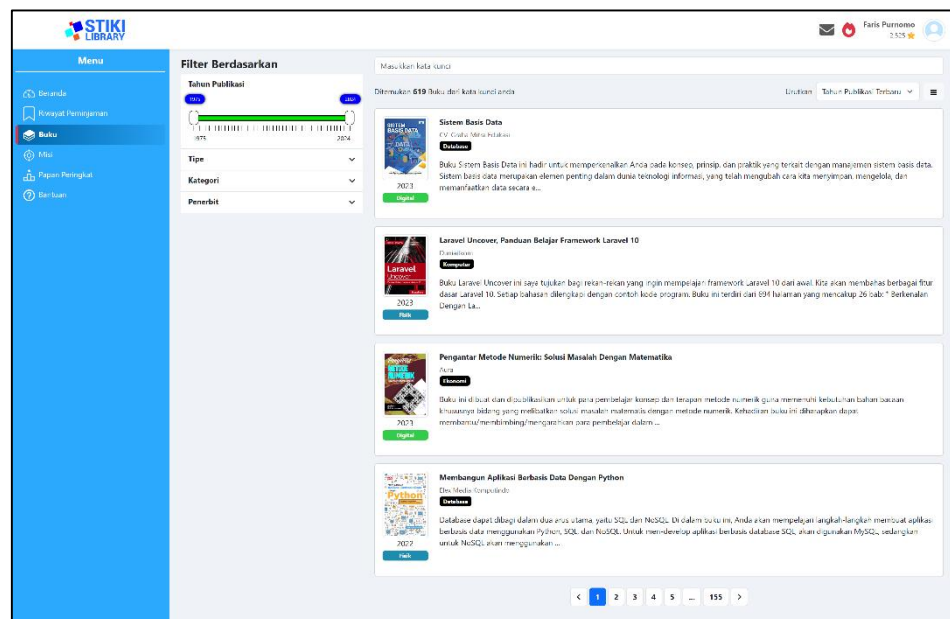
```

62     book.paging.offset = response.data.data.from;
63     book.total = response.data.data.total;
64 }
65 }).catch(() => book.is_loading = false );
66 };
67 onMounted(async () => {
68     const current_year = new Date().getFullYear()
69     const min_year = 1975;
70     barMaxValue.value = current_year;
71     barMinValue.value = min_year;
72     filter.tahun_publikasi_awal = min_year;
73     filter.tahun_publikasi_akhir = current_year;
74     requestBook();
75 });
76 </script>

```

Segmen Program 4.6 Segmen Program Halaman Buku Pemain

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.6.



Gambar 4.6 Implementasi Menu Buku

f) Halaman detail buku

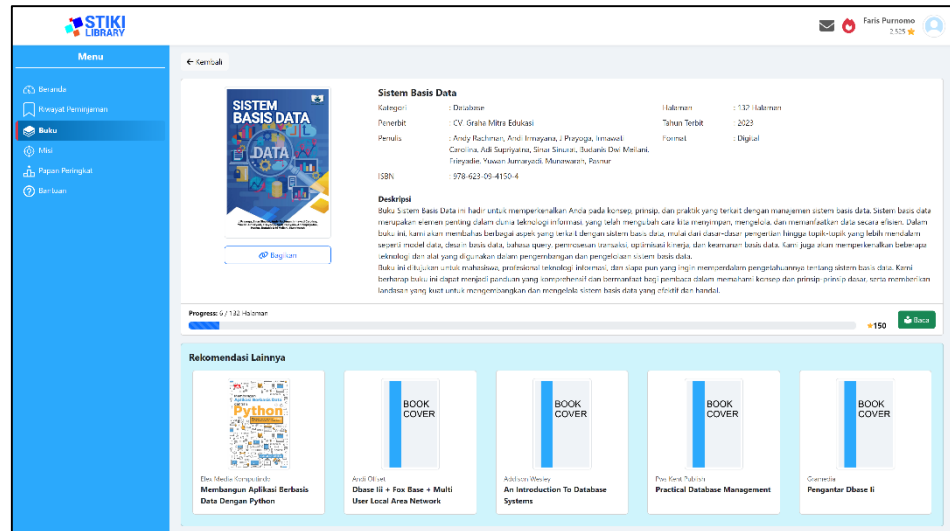
Halaman ini menyajikan informasi lengkap mengenai sebuah buku yang diperoleh dari *webservice*. Informasi yang ditampilkan meliputi judul,

penulis, penerbit, ISBN dan lain sebagainya. Cuplikan kode program tercantum dalam segmen program 4.7 dibawah ini.

Loc	Code
1	<script setup>
2	import \$axios from '@api/api';
3	import { onMounted, reactive, watch } from 'vue';
4	import { useRoute, useRouter } from 'vue-router';
5	
6	const route = useRoute();
7	const router = useRouter();
8	const book = reactive({
9	is_loading: false,
10	detail: {}
11	});
12	
13	const requestDetail = function (id) {
14	book.is_loading = true;
15	\$axios.get(`book/\${id}/detail`)
16	.then(response => {
17	if (response.data.is_success) {
18	book.detail = response.data.data;
19	} else {
20	alert(response.data.msg);
21	router.back();
22	return;
23	}
24	book.is_loading = false;
25	})
26	.catch(() => book.is_loading = false);
27	};
28	watch(route, (new_value, _) => {
29	requestDetail(new_value.params.id);
30	});
31	onMounted(() => requestDetail(route.params.id));
35	</script>

Segmen Program 4.7 Segmen Program Halaman Detail Buku

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.7.



Gambar 4.7 Implementasi Menu Detail Buku

g) Halaman misi

Halaman ini memuat daftar misi dan kuis yang sudah diterbitkan dan bisa dikerjakan oleh para pemain. Pada halaman ini, pemain juga dapat mengklaim *reward* atas penyelesaian misi. Cuplikan kode program tercantum dalam segmen program 4.8 dibawah ini.

Loc	Code
1	<code><script setup></code>
2	<code>import \$axios from '@api/api';</code>
3	<code>import { onMounted, ref } from 'vue';</code>
4	<code>import useAuthStore from '@stores/auth';</code>
5	
6	<code>const authStore = useAuthStore();</code>
7	<code>const missions = ref([]);</code>
8	<code>const quest_publishes = ref([]);</code>
9	<code>const is_loading = ref(true);</code>
10	<code>const nav_active = ref(0);</code>
11	
12	<code>const requestMissions = function () {</code>
13	<code> is_loading.value = true;</code>
14	<code> \$axios.get('mission/list')</code>
15	<code> .then(async response => {</code>
16	<code> is_loading.value = false;</code>
17	<code> if (response.data.is_success) {</code>

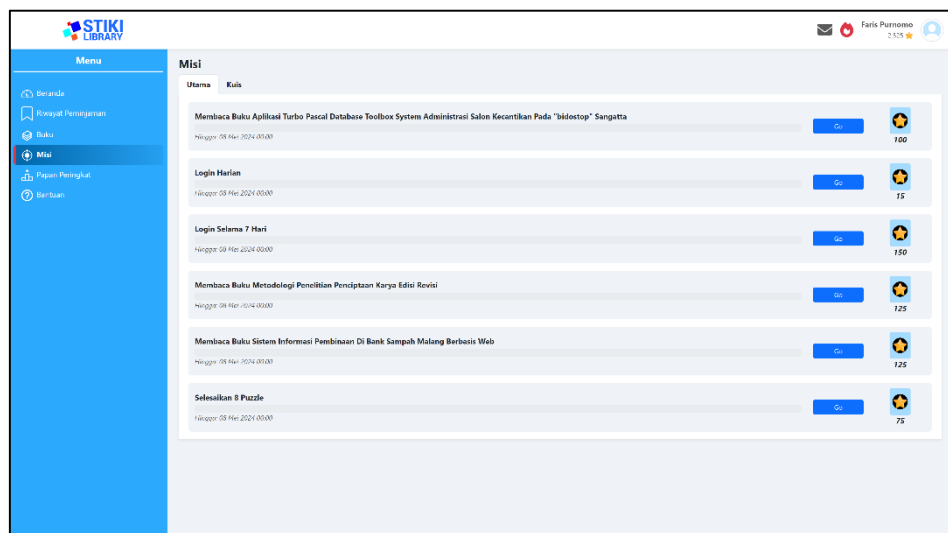

```

18         missions.value = response.data.data.missions;
19         quest_publishes.value = response.data.data.quests;
20     }
21     }).catch(() => is_loading.value = false );
22 };
23 onMounted(() => {
24     nav_active.value = 0;
25     requestMissions();
26 });
27 </script>

```

Segmen Program 4.8 Segmen Program Halaman Misi

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.8.



Gambar 4.8 Implementasi Menu Misi

h) Halaman menu detail kuis

Halaman ini menampilkan pertanyaan-pertanyaan yang terdapat pada kuis.

Setelah menyelesaikan semua pertanyaan, pemain akan mendapatkan *reward*. Cuplikan kode program tercantum dalam segmen program 4.9 dibawah ini.

Loc	Code
1	<script setup>
2	import { onMounted, reactive } from 'vue';
3	import { useRoute } from 'vue-router';
4	import \$axios from '@api/api';

```

5  import shuffle from 'lodash/shuffle';
6
7  const route = useRoute();
8  const state = {
9    QUEST: 'QUEST',
10   EVALUATE: 'EVALUATE',
11  };
12  const quest = reactive({
13    id: null,
14    is_loading: true,
15    title: '',
16    answer_id: null,
17    right_answer_id: null,
18    is_done: false,
19    state: state.QUEST
20  });
21  const requestQuest = async function () {
22    let formdata = undefined;
23    if (quest.answer_id) {
24      formdata = {
25        answer_id: quest.answer_id,
26        question_id: quest.question.id
27      };
28    }
29    await $axios.post(`mission/quest/${quest.id}/questions`, formdata)
30      .then(response => {
31        quest.is_loading = false;
32        const data = response.data.data;
33        if (response.data.is_success) {
34          quest.title = data.question.title;
35          quest.page.current = data.current_page;
36          quest.page.total = data.total_page;
37          quest.summary.reward = data.summary.reward;
38          quest.summary.wrong = data.summary.wrong;
39          quest.summary.right = data.summary.right;
40          if (data.question) {
41            if (quest.state == state.EVALUATE) {
42              quest.question.id = data.question.id;
43              quest.question.title = data.question.title;
44              quest.question.options = shuffle(data.question.options);
45            } else {
46              quest.next_question.id = data.question.id;

```

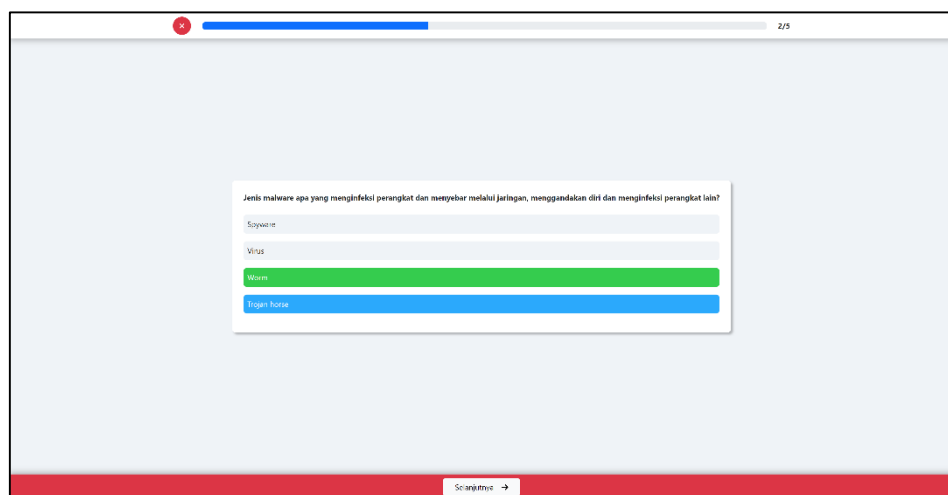
```

47         quest.next_question.title = data.question.title;
48         quest.next_question.options = shuffle(data.question.options);
49     }
50 }
51 quest.right_answer_id = data.right_answer_id;
52 }
53 quest.is_done = data.is_done;
54 });
55 };
56 onMounted(() => requestQuest(route.params.id) );
57 </script>

```

Segmen Program 0.9 Segmen Program Halaman Detail Kuis

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.9.



Gambar 0.9 Implementasi Menu Detail Kuis

i) Halaman menu papan peringkat

Halaman ini menampilkan daftar pemain peringkat tertinggi hingga terendah, yang datanya diperoleh dari *webservice* dan disajikan dalam bentuk tabel. Ketika salah satu baris pemain ditekan, sistem akan mengambil data detail pemain dari *webservice* dan menampilkannya dalam bentuk *popup*. Cuplikan kode program tercantum dalam segmen program 4.10 berikut ini.

Loc	Code
1	<script setup>
2	import { onMounted, reactive } from 'vue';
3	import { useRoute } from 'vue-router';
4	import \$axios from '@api/api';
5	import shuffle from 'lodash/shuffle';
6	
7	const route = useRoute();
8	const state = {
9	QUEST: 'QUEST',
10	EVALUATE: 'EVALUATE',
11	};
12	const quest = reactive({
13	id: null,
14	is_loading: true,
15	title: '',
16	answer_id: null,
17	right_answer_id: null,
18	is_done: false,
19	state: state.QUEST,
20	page: {
21	current: 0,
22	total: 0
23	},
24	question: {
25	id: null,
26	title: '',
27	options: []
28	},
29	next_question: {
30	id: null,
31	title: '',
32	options: []
33	},
34	summary: {
35	reward: 0,
36	right: 0,
37	wrong: 0
38	}
39	});
40	const requestQuest = async function () {
41	let formdata = undefined;

```

42   if (quest.answer_id) {
43     formdata = {
44       answer_id: quest.answer_id,
45       question_id: quest.question.id
46     };
47   }
48   await $axios.post(`mission/quest/${quest.id}/questions`, formdata)
49     .then(response => {
50       quest.is_loading = false;
51       const data = response.data.data;
52       if (response.data.is_success) {
53         quest.title = data.question.title;
54         quest.page.current = data.current_page;
55         quest.page.total = data.total_page;
56         quest.summary.reward = data.summary.reward;
57         quest.summary.wrong = data.summary.wrong;
58         quest.summary.right = data.summary.right;
59         if (data.question) {
60           if (quest.state == state.EVALUATE || quest.question.id ==
61 null) {
62             quest.question.id = data.question.id;
63             quest.question.title = data.question.title;
64             quest.question.options = shuffle(data.question.options);
65           } else {
66             quest.next_question.id = data.question.id;
67             quest.next_question.title = data.question.title;
68             quest.next_question.options =
69 shuffle(data.question.options);
70           }
71           quest.right_answer_id = data.right_answer_id;
72           quest.is_done = data.is_done;
73         }).catch(() => quest.is_loading = false );
74       };
75       onMounted(() => {
76         quest.id = route.params.id;
77         requestQuest();
78       });
79     </script>

```

Segmen Program 4.10 Segmen Program Halaman Papan Peringkat

Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.10.

#	Pemain	Poin
1	Faris Purnomo	2,525
2	Devi Putrismi Putra	1,919
3	Karel Adhar Ramadhani	1,911
4	Andreas Nugra	1,741
5	Stephanus Dimas Hubo	1,734
6	Kevin Andrian Felle Nurwanto	1,698
7	Pessy Adi Arbanth	1,559
8	Dana Sugan Erlangga	1,506
9	Hilamah	1,443
10	Devi Agnirio	1,429
11	Wahyu Bagus Wicaksono	1,095

Gambar 4.10 Implementasi Menu Papan Peringkat

j) Halaman profil

Halaman ini menampilkan *badge* yang dikumpulkan oleh pemain. Selain itu pemain juga bisa mengganti kata sandi melalui halaman ini. Cuplikan kode program tercantum dalam segmen program 4.11 dibawah ini.

Loc	Code
1	<code><script setup></code>
2	<code>import { onMounted, reactive, watch } from 'vue';</code>
3	<code>import \$axios from '@/api/api.js';</code>
4	<code>import useAuthStore from '@/stores/auth';</code>
5	
6	<code>const authStore = useAuthStore();</code>
7	<code>const user = reactive({</code>
8	<code> is_loading: true,</code>
9	<code> is_submitting: false,</code>
10	<code> detail: {},</code>
11	<code> skills: []</code>
12	<code>});</code>
13	
14	<code>const requestUser = function () {</code>
15	<code> user.is_loading = true;</code>
16	<code> \$axios.get('profile')</code>
17	<code> .then(response => {</code>
18	<code> user.is_loading = false;</code>
19	<code> if (response.data.is_success) {</code>

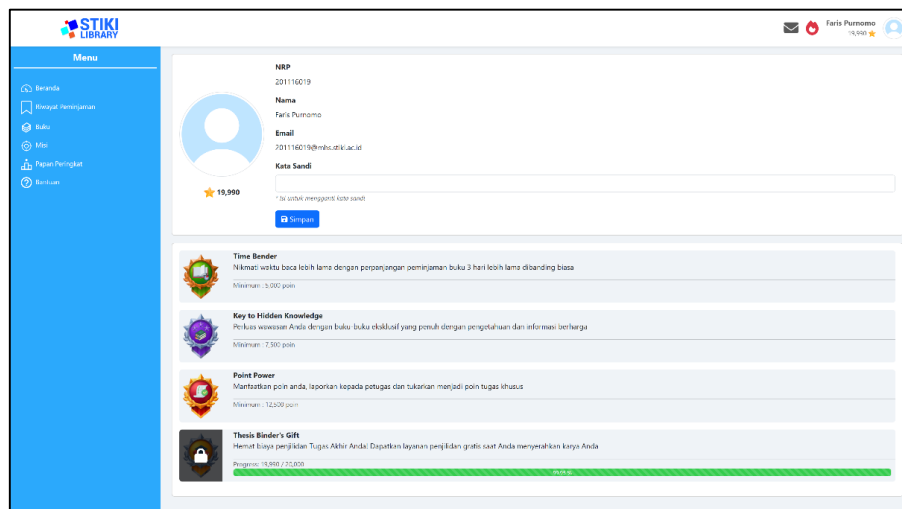
```

20     user.detail = response.data.data.user;
21     user.skills = response.data.data.skills;
22     authStore.setUser(response.data.data.user);
23   }
24 })
25 };
26 onMounted(() => requestUser() );
27 </script>

```

Segmen Program 4.11 Segmen Program Halaman Profil

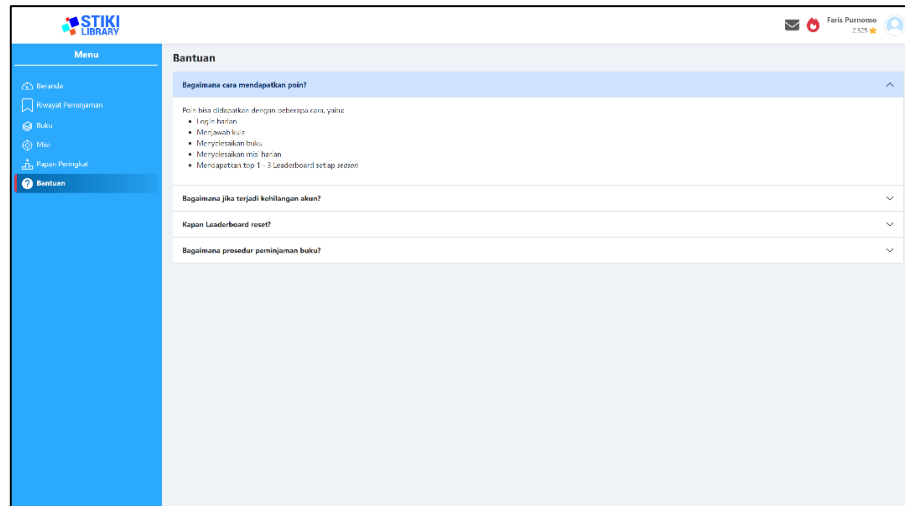
Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.9.



Gambar 4.11 Implementasi Menu Papan Peringkat

k) Halaman menu bantuan

Halaman ini ditampilkan secara statis tanpa tanpa memerlukan API atau *webservice*. Listing kode program tercantum dalam Lampiran 5.2. Hasil dari kode program tersebut dapat ditunjukkan pada Gambar 4.12.



Gambar 4.12 Implementasi Menu Bantuan

4.3 Pengujian

Setelah kode program diimplementasikan, pengujian validasi dilakukan untuk memastikan sistem memenuhi spesifikasi kebutuhan yang telah ditetapkan. Hasil pengujian ini tertera pada Tabel 4.5.

Tabel 4.5 Hasil Pengujian Pada Sistem Informasi

Kasus Uji	Prosedur Pengujian	Hasil yang Diharapkan Dan Hasil yang Diperoleh	Status
Spesifikasi Kebutuhan: REQ_001 poin (a) Kasus Uji: Merepresentasikan aktivitas proses autentikasi akun.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (a).	Hasil yang diharapkan: Sistem akan mengarahkan pemain menuju halaman beranda. Hasil yang diperoleh: Pemain diarahkan menuju halaman beranda.	Sesuai
Spesifikasi Kebutuhan: REQ_001 poin (b) Kasus Uji: Merepresentasikan aktivitas proses reset kata sandi.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (b).	Hasil yang diharapkan: Sistem mengubah kata sandi lama dengan kata sandi yang baru. Hasil yang diperoleh: Kata sandi berhasil di <i>reset</i> .	Sesuai
Spesifikasi Kebutuhan: REQ_001 poin (c) Kasus Uji: Merepresentasikan aktivitas proses melihat detail buku.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (c).	Hasil yang diharapkan: Sistem menampilkan detail buku yang dipilih. Hasil yang diperoleh: Sistem berhasil menampilkan detail buku sesuai dengan yang dipilih.	Sesuai

Spesifikasi Kebutuhan: REQ_001 poin (d) Kasus Uji: Merepresentasikan aktivitas proses melihat detail riwayat detail peminjaman dan pengembalian buku	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (d).	Hasil yang diharapkan: Sistem menampilkan detail riwayat peminjaman dan pengembalian buku. Hasil yang diperoleh: Detail peminjaman berhasil ditampilkan.	Sesuai
Spesifikasi Kebutuhan: REQ_002 poin (a) Kasus Uji: Merepresentasikan aktivitas proses klaim poin dari login harian.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (e).	Hasil yang diharapkan: Pemain mendapat hadiah poin. Poin tersebut kemudian terakumulasi pada akun pemain. Hasil yang diperoleh: Poin pemain bertambah sebanyak 15 poin.	Sesuai
Spesifikasi Kebutuhan: REQ_002 poin (b) Kasus Uji: Merepresentasikan aktivitas proses menyelesaikan buku digital.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (f).	Hasil yang diharapkan: Pemain mendapatkan hadiah poin. Poin tersebut kemudian terakumulasi pada akun pemain. Hasil yang diperoleh: Poin pemain bertambah sebanyak 150 poin.	Sesuai
Spesifikasi Kebutuhan: REQ_002 poin (c) Kasus Uji: Merepresentasikan aktivitas proses mengerjakan misi.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.2.2 poin (g).	Hasil yang diharapkan: Pemain mendapatkan hadiah poin. Poin tersebut kemudian terakumulasi pada akun pemain. Hasil yang diperoleh: Poin pemain bertambah sebanyak 15 poin.	Sesuai
Spesifikasi Kebutuhan: REQ_002 poin (d) Kasus Uji: Merepresentasikan aktivitas proses mengerjakan kuis.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (h).	Hasil yang diharapkan: Pemain mendapatkan hadiah poin. Poin tersebut kemudian terakumulasi pada akun pemain. Hasil yang diperoleh: Poin pemain bertambah sebanyak 250 poin.	Sesuai
Spesifikasi Kebutuhan: REQ_002 poin (e) Kasus Uji: Merepresentasikan aktivitas proses melihat papan peringkat.	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (i).	Hasil yang diharapkan: Sistem menampilkan daftar papan peringkat kepada pemain. Hasil yang diperoleh: Sistem berhasil menampilkan daftar papan peringkat.	Sesuai
Spesifikasi Kebutuhan: REQ_002 poin (f) Kasus Uji: Merepresentasikan aktivitas proses melihat	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (j).	Hasil yang diharapkan: Sistem mampu menampilkan kemampuan khusus pemain. Hasil yang diperoleh:	Sesuai

kemampuan khusus dan badge yang didapatkan		Sistem berhasil menampilkan kemampuan khusus dan badge sesuai poin yang diperoleh	
Spesifikasi Kebutuhan: REQ_001 poin (e) Kasus Uji: Merepresentasikan aktivitas proses <i>logout</i> .	Prosedur Uji: Mengacu pada activity diagram yang digambarkan pada sub bab 3.2.1.2 poin (k).	Hasil yang diharapkan: Sistem mengarahkan pemain ke halaman login. Hasil yang diperoleh: Pemain diarahkan ke halaman login.	Sesuai