BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

Pada bab ini penulis menjelaskan mengenai hasil dan pembahasan dari penelitian dan eksperimen yang telah dilakukan.

4.1.1 Pengumpulan Data

Pada tahap ini, peneliti melakukan studi literatur untuk mencari fokus masalah yang akan di modelkan ke aplikasi. Studi literatur ini dilakukan melalui buku, jurnal, surat kabar, dan penelitian lain.

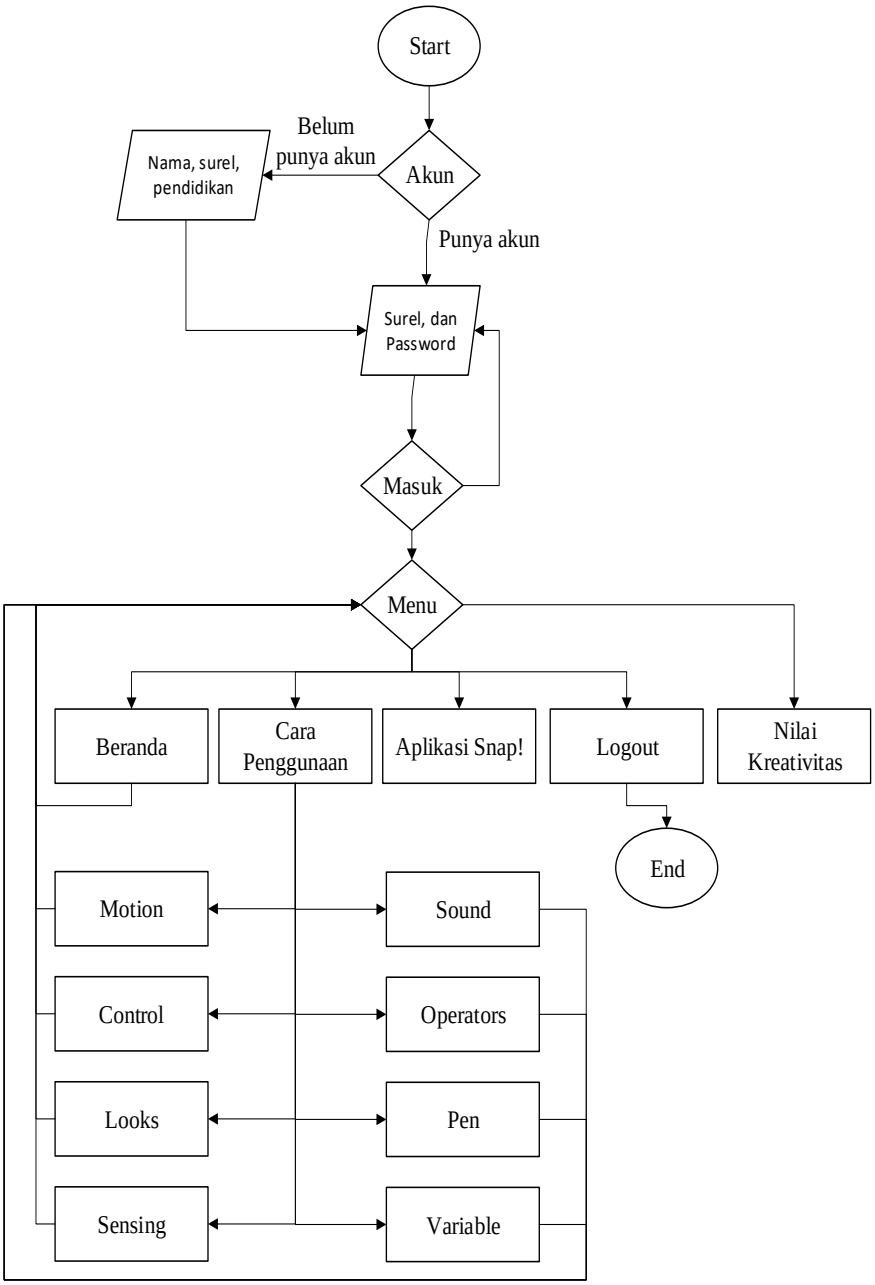
Informasi yang dicari tentang kreativitas, pengukuran kreativitas, keterkaitan antara kreativitas dengan kognitif.

4.1.2 Perencanaan dan Analisis

1) Membuat *Flowchart*

Rancangan pada tahap ini menggunakan simbolisasi yang telah lazim digunakan. Sehingga apabila ada pengembangan selanjutnya oleh orang yang berbeda, maka secara mudah *flowchart* ini dapat dibaca.

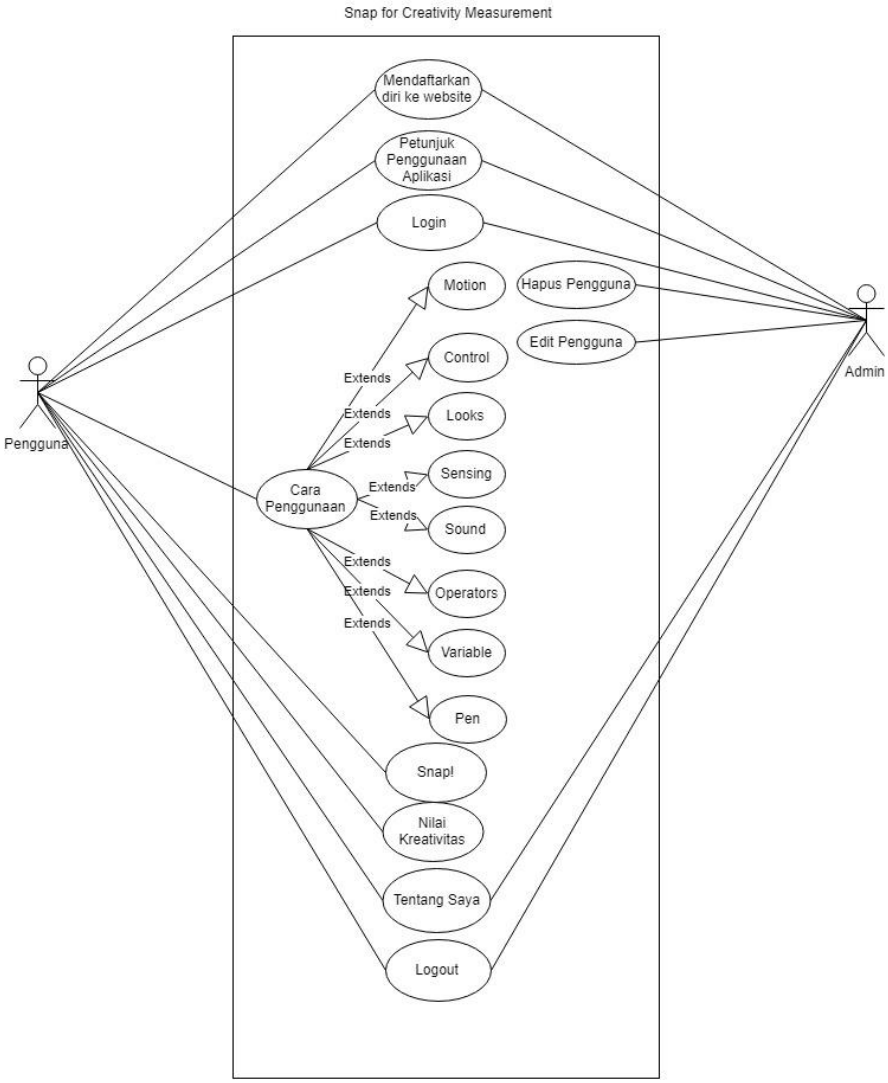
Makhrus Lu'ay Cholis, 2019
RANCANG BANGUN APLIKASI PENGUKURAN KREATIVITAS MEMPROGRAM MENGGUNAKAN BLOK PEMROGRAMAN
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu



Gambar 4. 1 Flowchart Aplikasi

Pada Gambar 4.1, menggambarkan alur utama dari aplikasi yang dimulai dari jika pengguna sudah mempunyai akun maka bisa ke halaman masuk jika tidak ke halaman daftar dan mengisi data diri. Setelah pengguna masuk atau mendaftar maka dilanjutkan ke halaman utama. Dalam menu utama pengguna dianjurkan untuk memilih petunjuk agar paham selama penggunaan aplikasi.

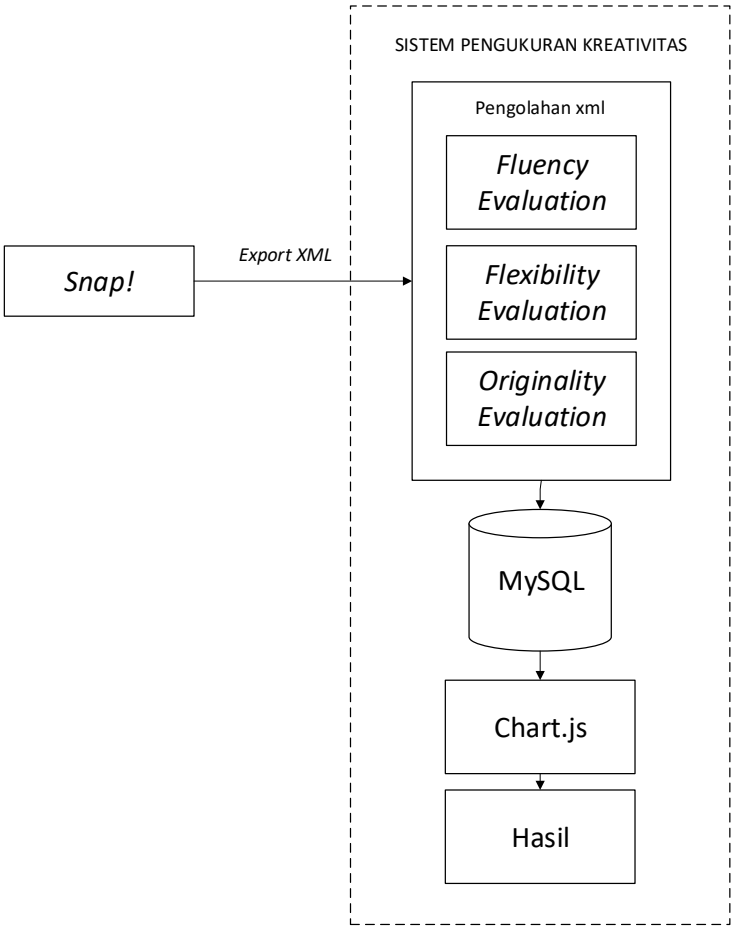
2) Membuat *Unified Modeling Language*(UML)



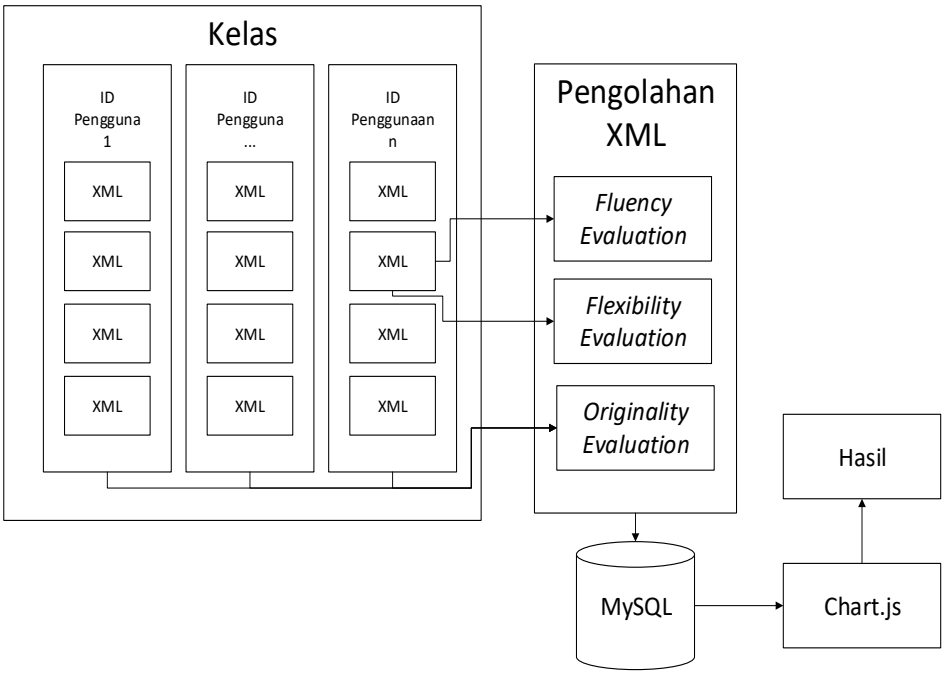
Gambar 4. 2 Unified Modeling Language Aplikasi Pengukuran Kreativitas

3) Desain Pengukuran Kreativitas

Snap dikemas dalam *framework* CodeIgniter. Pada dasarnya, S*nap!* dapat berdiri sendiri, tetapi karena peneliti membutuhkan data hasil penggunaan *snap!*, *Snap!* dimodifikasi agar dapat tersimpan kedalam *database* dan diolah dengan bahasa pemrograman php. Modifikasi *Snap!* dilakukan pada *file* gui.js. gui.js mengatur keseluruhan tampilan pada *snap!*.



Gambar 4. 3 Desain alur sistem pengukuran kreativitas



Gambar 4. 4 Alur data yang diolah pada pengukuran kreativitas

Hasil proyek *Snap!* diekspor kedalam *database* dibedakan berdasarkan id pengguna. Berdasarkan aspek nilai kreativitas yang telah disepakati, yaitu penilaian *fluency* berdasarkan blok yang sama dan digunakan berulang, penilaian *flexibility* dilakukan berdasarkan penggunaan blok yang berbeda, dan penilaian *originality* berdasarkan kesamaan blok yang digunakan dibandingkan dengan proyek pengguna yang lain.

Proyek yang dihasilkan *Snap!* berformat *.xml, selanjutnya konten dari xml tersebut diambil dan dimasukan kedalam table t_xml. Tabel tersebut memiliki kolom id, nik(id pengguna), xml, keterangan, waktu, *fluency*, dan *flexibility*.

Pengecekan *originality* dilakukan dengan cara membandingkan file 1 xml dengan file keseluruhan dari yang pengguna unggah.

Pada saat pengguna mengunggah proyek *snap!*, secara otomatis php mengolah xml tersebut untuk mendapatkan nilai *fluency*, dan *flexibility*.

Penilaian *fluency*, dan *flexibility* dapat dihitung per 1 proyek xml, sedangkan untuk *originality* dihitung per blok dan dibandingkan dengan keseluruhan proyek pengguna lain.

Tabel 4. 1 Tabel Penilaian *originality*

Nama Blok	Persentase penggunaan blok secara keseluruhan	Pengguna
Start	80%	Pengguna 1, pengguna 2
If	80%	Pengguna 1, pengguna 3
For	80%	Pengguna 2, pengguna 3

Snap! Telah dimodifikasi agar dapat mengekspor data kedalam *database*, modifikasi dilakukan agar data yang didapatkan dari Snap! dapat diolah. Dengan penambahan tombol *export to database*, php dapat mengolah data xml yang diekspor. Penambahan tombol *export to database* ditambahkan pada file gui.js di dalam *snap!*

```
        menu.addItem(
            'Export to database...',
            function () {
                if (myself.projectName) {
                    myself.saveToDatabase(myself.projectName,
shiftClicked);
                } else {
                    myself.prompt('Export    Project    As...',
function (name) {
                        myself.saveToDatabase(name,
shiftClicked);
                    }, null, 'saveToDatabase');
                }
            }
        );
```

Proses menambahkan tombol *export to database* pada *Snap!*

Setelah menambahkan tombol *export to database*, data akan mengeksekusi fungsi saveToDatabase

```
    IDE_Morph.prototype.saveToDatabase = function (name,plain)
{
        var menu, str, dataPrefix;

        var currentDate = new Date();
        var currentDate = currentDate.toYMD();
        console.log(currentDate);
```

```
        var base_url = window.location.origin;
        if (name) {
            this.setProjectName(name);
            dataPrefix = 'data:text/' + plain ? 'plain,' :
'xml,';
            try {
                    menu = this.showMessage('Exporting');
                    str =
this.serializer.serialize(this.stage);
                    $.ajax({
                        url: base_url + "/smartai/get_xml/get",
                        type: 'POST',
                        data: { "xml" : str, "keterangan" :
name, "waktu" : currentDate },
                        dataType : 'json',
                        success:function() {
                            alert("Berhasil dikirim!");
                        },
                }});
            } catch (err) {
                if (Process.prototype.isCatchingErrors) {
                    this.showMessage('Export failed: ' + err);
                } else {
                    throw err;
                }
            }
        }

        menu.destroy();
        this.showMessage('Exported!', 1);
    }
```

Fungsi saveToDatabase

Pada fungsi ini, javascript dan ajax akan mengirim data xml, keterangan, dan waktu ke url: {base_url}/get_xml/get. Pada url tersebut, data akan dikirim ke *database*

```
        public function get() {
            $dom = new DOMDocument;
            $dom->loadXml($this->input->post('xml'));
            $fluency = $dom->getElementsByTagName('block')-
>length;

            $baris = array(
                'id' => null,
```

```
                'nik'            =>            $this->session-
>userdata('admin_user'),
                'xml' => $this->input->post('xml'),
                'keterangan'          =>          $this->input-
>post('keterangan'),
                'waktu' => $this->input->post('waktu'),
                'fuency' => $fluency
            );

            $this->db->insert('t_xml', $baris);
            echo json_encode(array('status' => 'check' ));
            $this->output-
>set_content_type('application/json');
        }
```

Pengiriman data ke *database*

Setelah tersimpan, data kemudian ditampilkan dan diolah dengan chart.js dengan cara:

1. Data yang terdapat dalam *database* ditampung pada suatu array

```
            <?php
                $tgl=[];
                $fcy=[];
                $flx=[];
                foreach($data as $d){
                    array_push($tgl,$d->TANGGAL);
                    array_push($fcy,$d->fluency);
                    array_push($flx,$d->flexibility);
                };

            ?>
```

2. Setelah data yang dibutuhkan ditampung, selanjutnya data disajikan menggunakan chart.js

```
var ctx = document.getElementById('myChart').getContext('2d');
                var myChart = new Chart(ctx, {
                    type:'line',
                    options: {
                        scales: {
                            yAxes: [{
                                ticks: {
                                    beginAtZero: true
                                }
```

```
                            }]
                        },
                    },
                    data: {
                        labels: [<?php echo '"'.implode('","',
$tgl ).'"' ?>],
                        datasets: [{
                            label: "Fluency",
                            data:          [<?php          echo
'"'.implode('","',  $fcy ).'"' ?>],
                            pointHoverBorderColor:
settings.colors.accent[400],
                            pointHoverBackgroundColor:
settings.colors.white,
                            backgroundColor: [
                                'rgba(54, 162, 235, 0.2)'
                            ],
                            borderColor: [
                                'rgba(54, 162, 235, 1)'
                            ],
                            borderWidth: 1,
                            borderJoinStyle: 'bevel',
                            lineTension: .1
                        }]
                    }


                });
```

Data yang ditampung, disajikan dan diolah ke dalam chart.js, data yang ditampung, ditampilkan per proyek yang dihasilkan.

4) Analisis Kebutuhan

Aplikasi ini ditujukan untuk para pengguna yang ingin mengetahui seberapa besar tingkat kreativitasnya.

Setelah menemukan analisis calon pengguna, dibutuhkan pula analisis perangkat lunak. Kebutuhan perangkat lunak yang digunakan dalam perancangan aplikasi ini ialah sebagai berikut:

a. Visual Studio Code 1.36.1

Teks editor yang digunakan proses pengembangan adalah Visual Studio Code. Visual Studio Code dapat digunakan untuk melakukan pengembangan

aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web.

b. Windows 10

Sistem operasi pada proses pengembangan aplikasi ini menggunakan windows 10 Home Single Language versi 1903. Untuk spesifikasi lengkap komputer yang digunakan ialah:

1.Processor : 1.7 GHz Intel Core i3

2.Memory : 6 GB 1600 MHz

3.Graphic : Intel HD Graphic 4400

Kemudian *requirement* komputer yang bisa menjalan aplikasi ini adalah perangkat yang mempunyai web *browser*, karena aplikasi ini merupakan aplikasi web maka yang terpenting adalah adanya web *browser* disarankan untuk memakai Google Chrome.

c. Google Chrome 75

Google Chrome ialah sebuah web *browser* yang digunakan untuk mengakses aplikasi.

d. MySQL 5.7

Database yang dipakai dalam aplikasi adalah MySQL. MySQL merupakan basis data SQL (bahasa Inggris: database management system) atau DBMS yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia.

e. PHP 5.6.40

Aplikasi dibuat dengan bahasa pemrograman PHP karena mudah dipelajari dan banyak referensinya. bahasa pemrograman ini dapat ditanamkan atau disisipkan ke dalam HTML. PHP juga banyak dipakai untuk memprogram situs web dinamis.

f. CodeIgniter 3.1.10

Untuk kecepatan pengembangan peneliti memakai CodeIgniter sebagai kerangka kerjanya dan juga bersifat *open source*, menggunakan konsep *model–view–controller*. CodeIgniter berada dibawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode.

4.1.3 Pengembangan

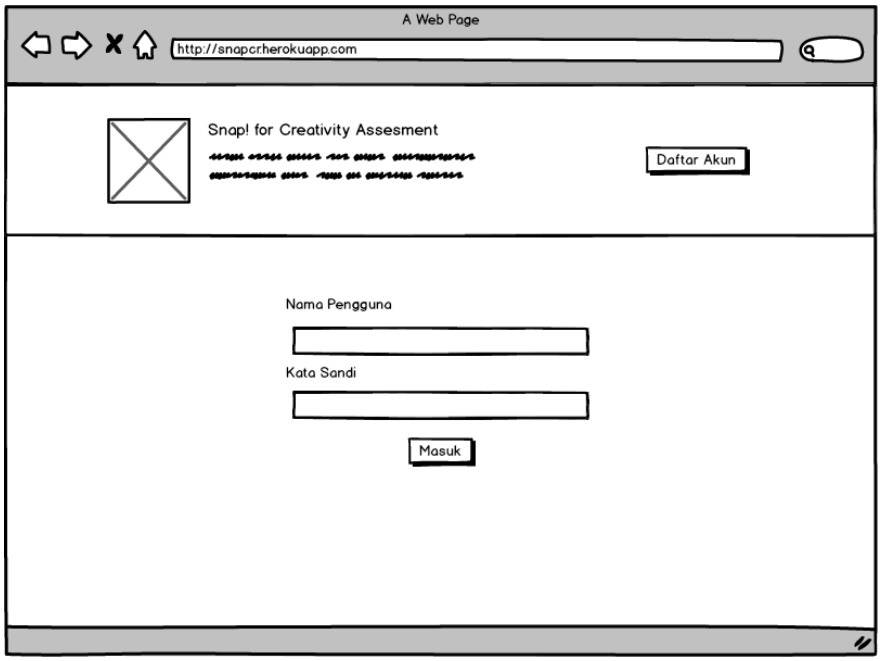
1) Membuat Perangkat Lunak

a. Wire Frame Mock-up

Pembuatan antarmuka aplikasi ini mengacu pada *flowchart* yang telah dirancang sebelumnya. Tampilan *Mock-up* sebagai berikut:

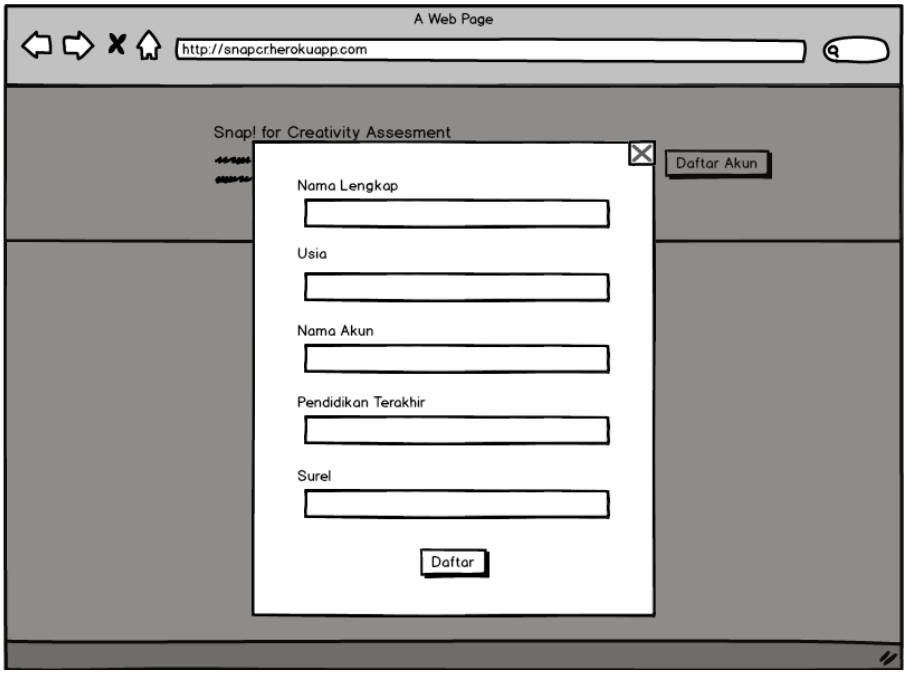
1. Halaman Awal/Masuk

Pada halaman awal pengguna langsung dihadapkan dengan halaman masuk yaitu form email dan password. Terdapat juga menu di kanan atas daftar dan masuk untuk navigasinya.



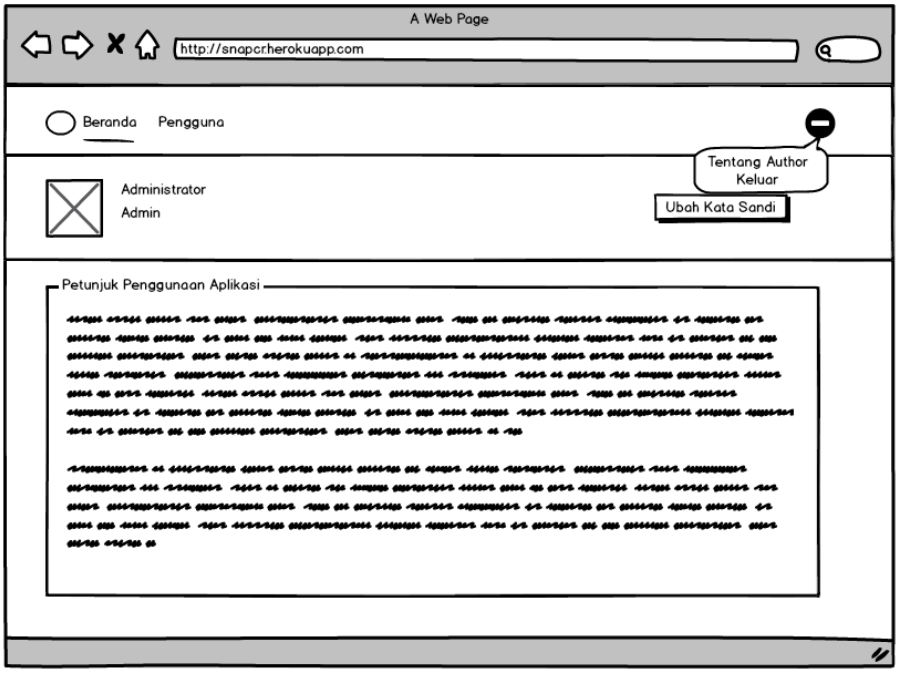
2. Halaman Daftar

Pada halaman daftar aplikasi berisi form yang harus diisi oleh pengguna yang belum mempunyai akun. Data yang harus di masukan adalah nama lengkap, usia, surel, pendidikan, dan pekerjaan.



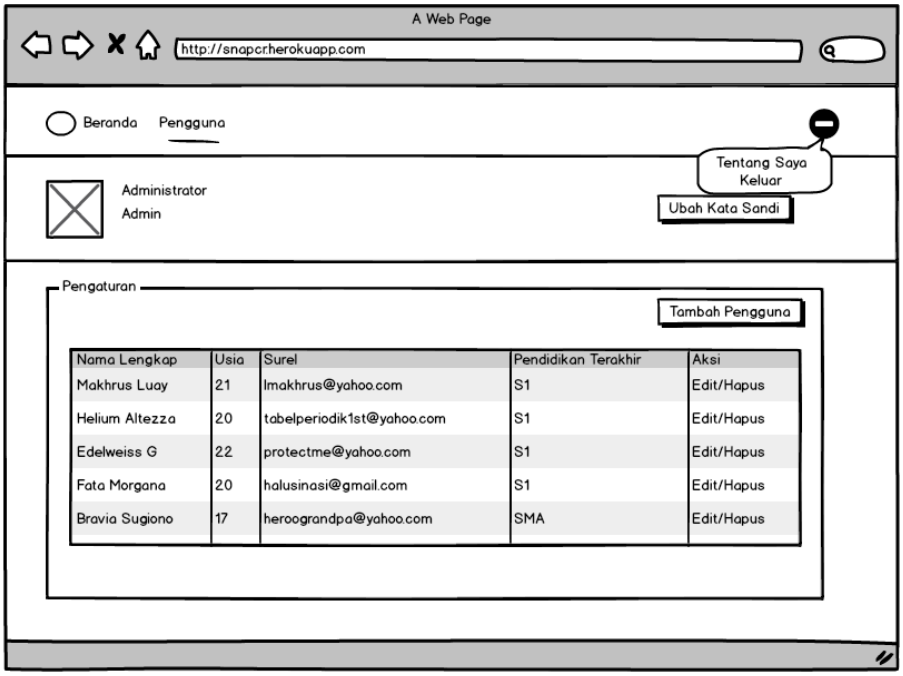
3. Halaman Beranda Admin

Di dalam halaman beranda admin, admin disuguhkan bagaimana aplikasi bekerja



4. Halaman Pengguna Admin

Admin dapat mengatur pengguna yang ingin menggunakan aplikasi ini.

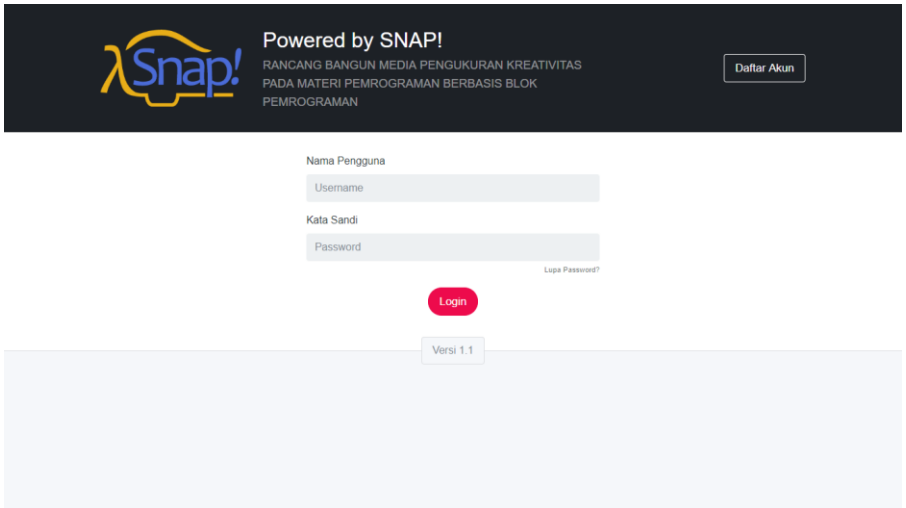


b. Antarmuka

Pembuatan antarmuka aplikasi ini mengacu pada *flowchart* yang telah dirancang sebelumnya. Tampilan antarmuka sebagai berikut:

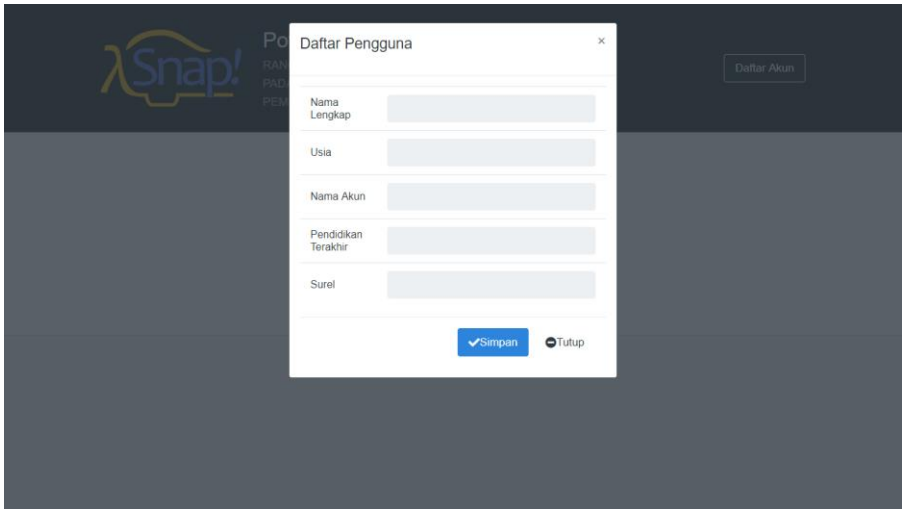
1. Halaman Awal / Masuk

Pada halaman awal pengguna langsung dihadapkan dengan halaman masuk yaitu form email dan password. Terdapat juga menu di kanan atas daftar dan masuk untuk navigasinya.



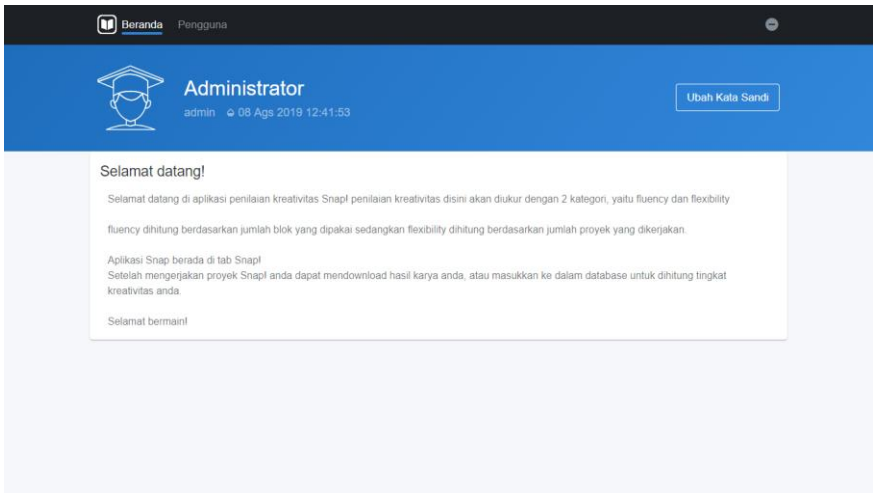
2. Halaman Daftar

Pada halaman daftar aplikasi berisi form yang harus diisi oleh pengguna yang belum mempunyai akun. Data yang harus di masukan adalah nama lengkap, usia, nama akun, pendidikan terakhir, dan surel.



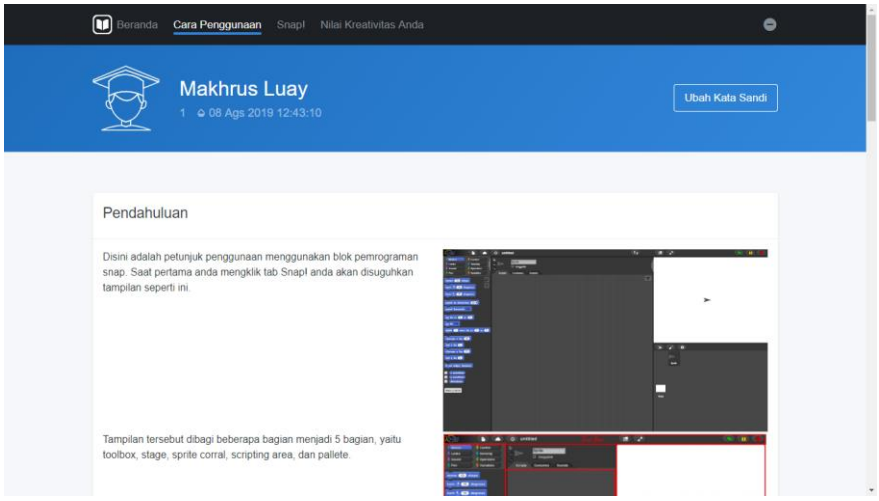
3. Halaman Beranda Admin

Pada halaman beranda menampilkan deskripsi dan fitur aplikasi.



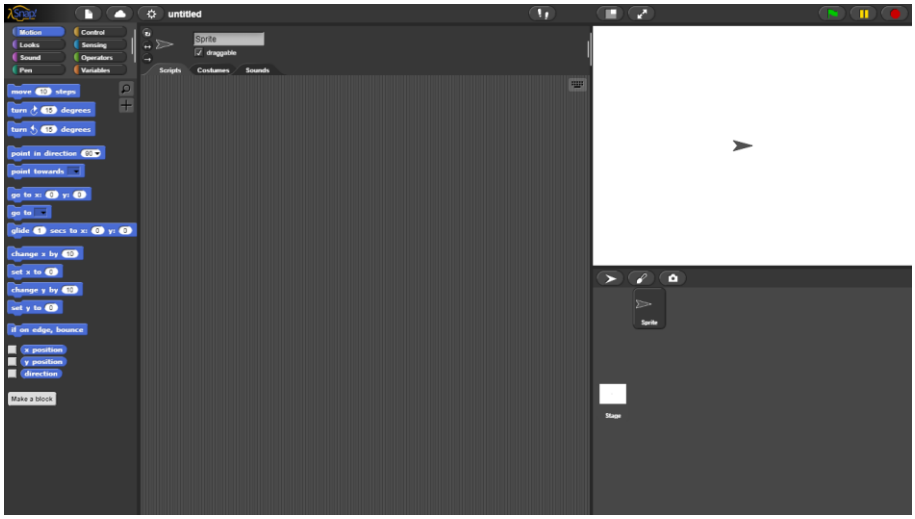
4. Halaman Cara Penggunaan

Pada halaman pemahaman atau bisa disebut juga materi merupakan halaman yang menampilkan materi untuk pengenalan tentang cara penggunaan Snap!



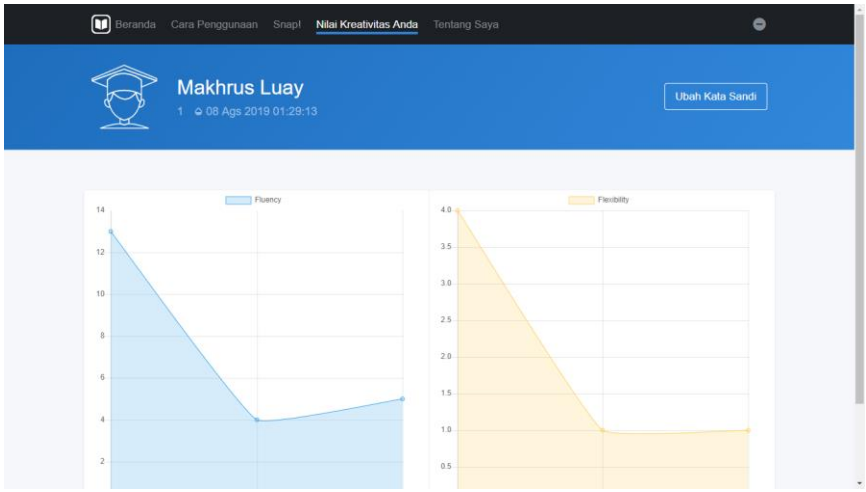
5. Halaman Snap!

Pada halaman Snap! pengguna dapat memulai pembuatan aplikasi atau game, proyek yang dihasilkan dapat dimasukan ke dalam *database* untuk selanjutnya diolah di halaman penilaian kreativitas.



6. Halaman Nilai Kreativitas Anda

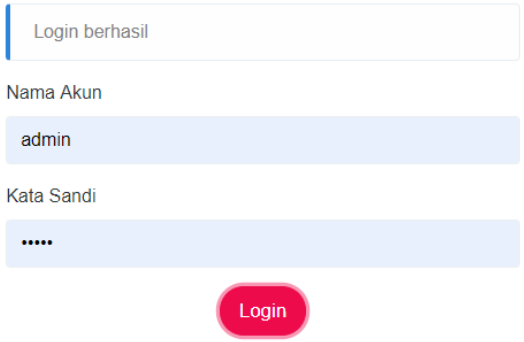
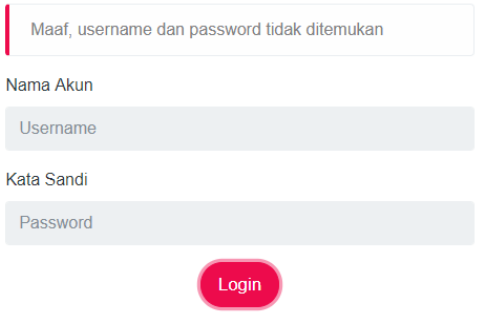
Pada halaman ini dapat dilihat hasil kreativitas yang didapatkan dari aplikasi Snap!

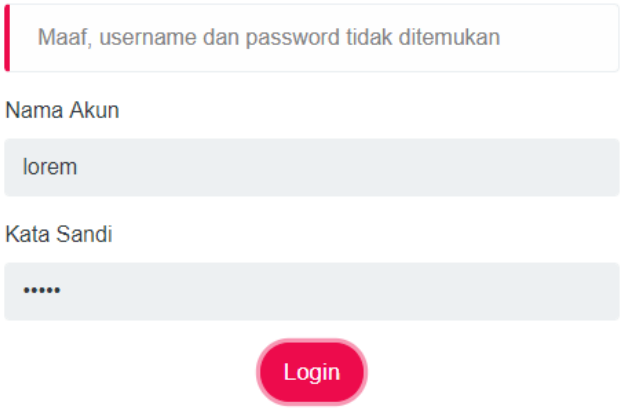
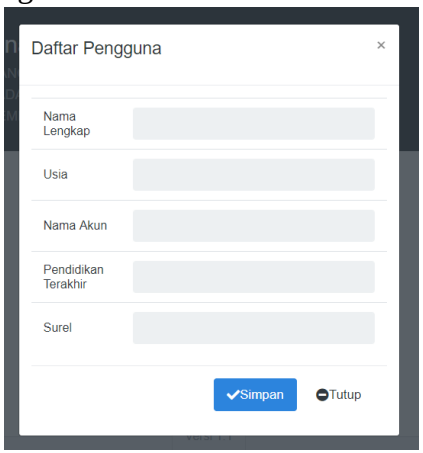


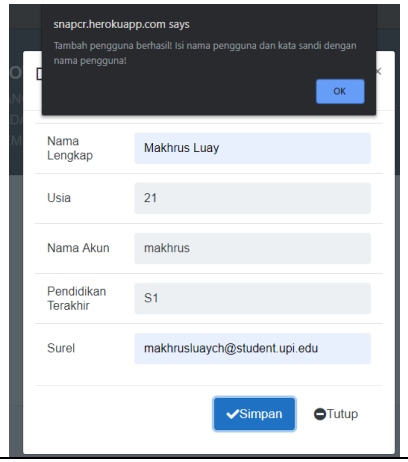
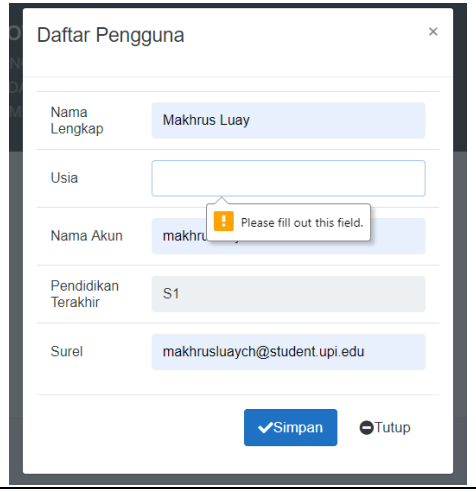
2) Melakukan *Black Box Testing*

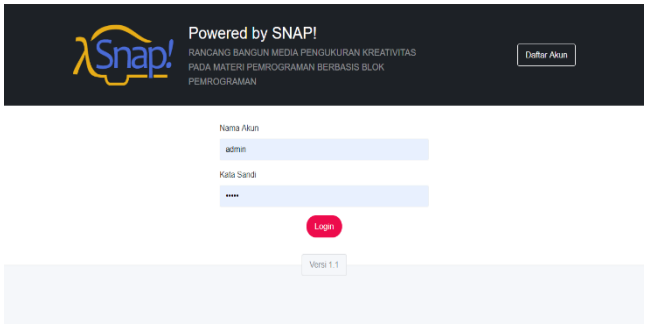
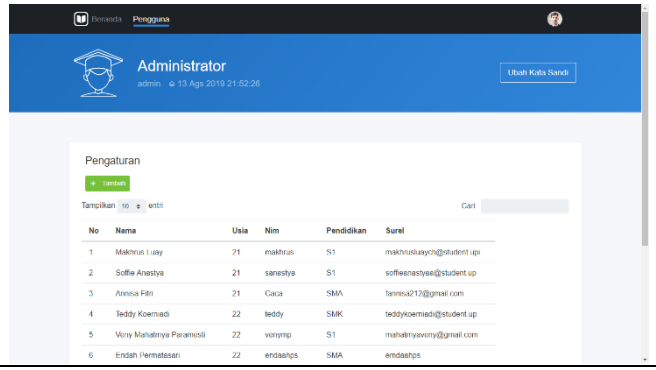
Pada tahap ini digunakan metode Black Box Testing, yang dilakukan untuk mengetahui masukan dan melihat hasil respon atau keluarannya yang ditampilkan oleh aplikasi, apakah sesuai dengan yang diharapkan, kesesuaian fungsi, tombol pada menu yang ada, serta pengecekan kesalahan input dan sebagainya. Hasil dari black box testing yang dilakukan dapat dilihat pada tabel 4.1 berikut:

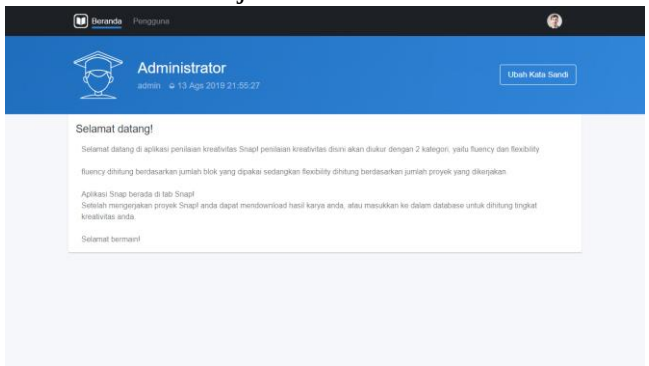
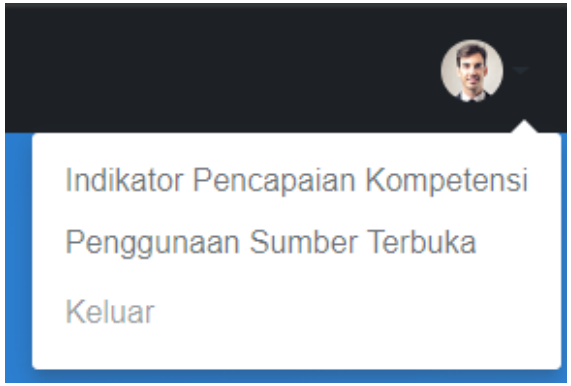
Tabel 4. 2 *Black Box Testing*

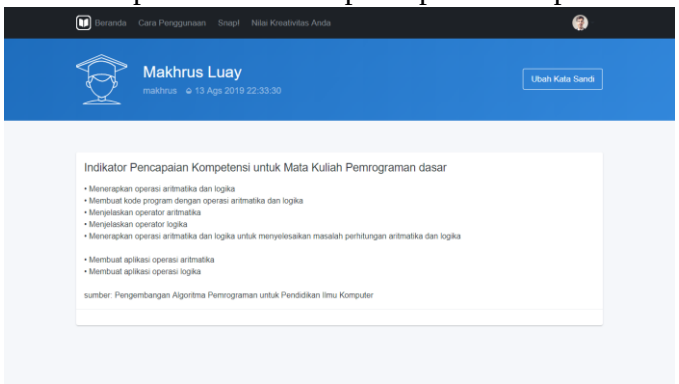
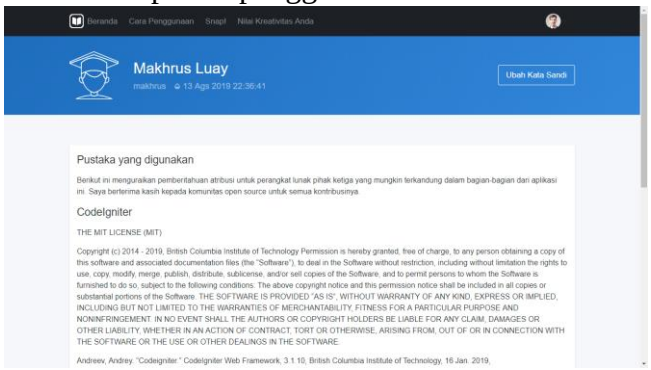
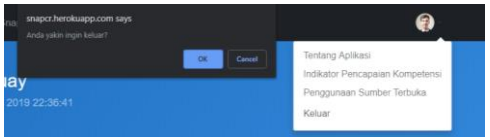
No	Skenario Pengujian	Data	Hasil Yang Diharapkan	Hasil Keluaran
1	Data yang dimasukkan lengkap dan sesuai	Nama Akun, Kata Sandi	Menuju halaman beranda  	Sukses
2	Data yang dimasukkan tidak lengkap	NULL	Tetap dihalaman masuk  	Sukses

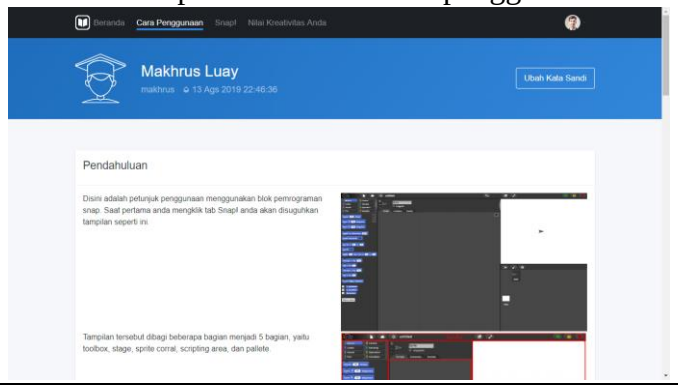
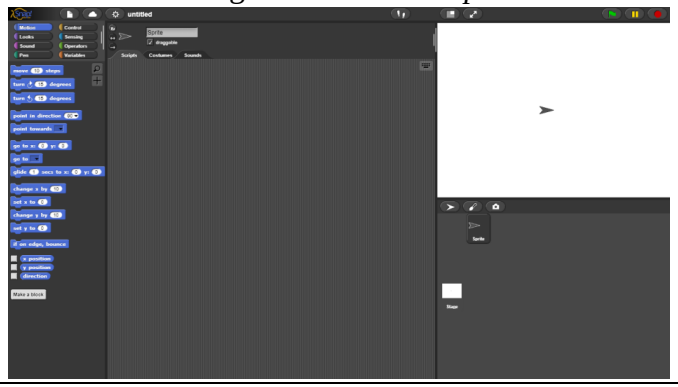
3	Kesalahan input sandi	Salah nama akun, dan/atau password	Tetap dihalaman masuk  	Sukses
4	Mendaftar akun	NULL	Mengeluarkan modal untuk daftar akun  	Sukses

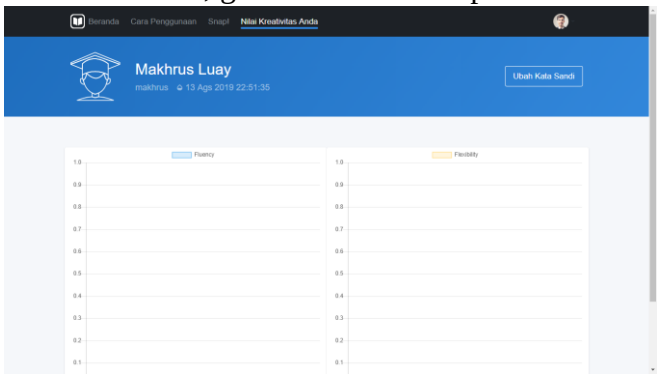
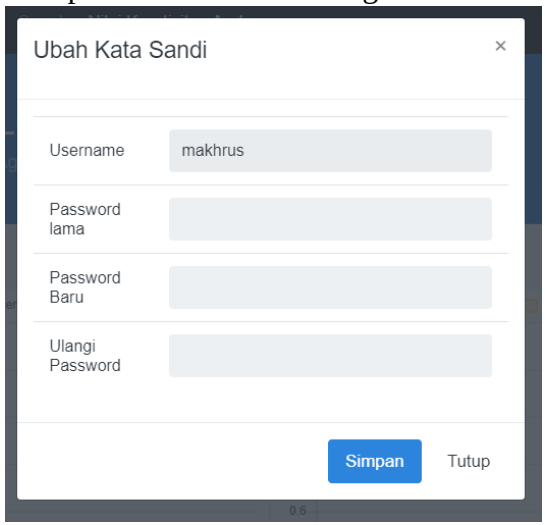
5	Daftar Akun	Nama lengkap, Usia, Nama Akun, Pendidikan terakhir, surel	Pengguna terdaftar 	Sukses
6	Data yang dimasukkan tidak lengkap	Nama lengkap, Nama Akun, Pendidikan terakhir, surel	Pengguna tidak terdaftar 	Sukses

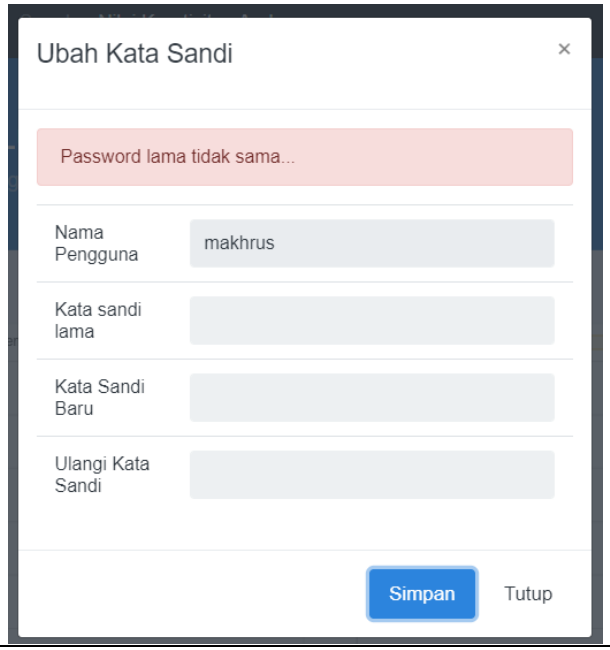
7	Batal daftar akun	NULL	Modal tertutup 	Sukses
8	Tekan tombol pengguna di navigasi halaman admin	NULL	Menuju halaman pengguna, tabel pengguna ditampilkan 	Sukses

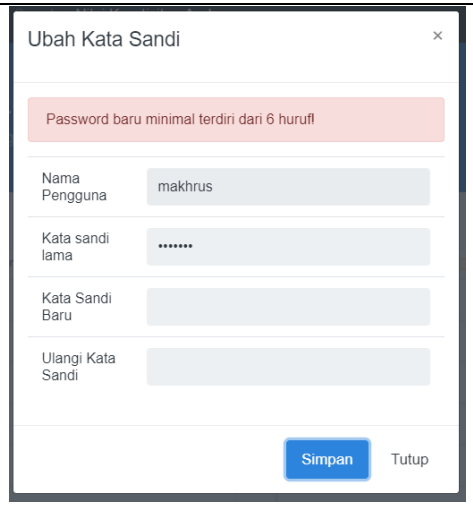
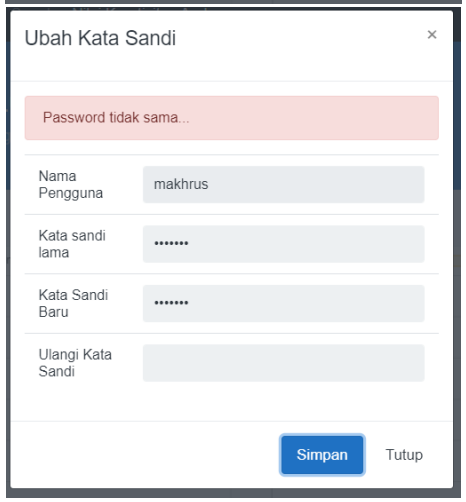
9	Tekan tombol beranda di navigasi halaman admin	NULL	Menuju halaman beranda 	Sukses
10	Tekan menu *dropdown* navigasi	NULL	Menampilkan *dropdown* 	Sukses

11	Tekan tombol Indikator Pencapaian Kompetensi di dalam menu *dropdown*	NULL	Menampilkan indikator pencapaian kompetensi 	Sukses
12	Tekan tombol penggunaan sumber terbuka	NULL	Menampilkan penggunaan sumber terbuka 	Sukses
14	Tekan tombol keluar	NULL	Menampilkan konfirmasi keluar, lalu keluar dari akun 	Sukses

15	Tekan tombol cara penggunaan	NULL	Menampilkan halaman cara penggunaan 	Sukses
16	Tekan tombol *Snap!*	NULL	Pergi ke halaman *Snap!* 	Sukses

17	Nilai kreativitas	NULL	Jika pengguna belum pernah mengunggah proyek ke *database*, grafik tidak menampilkan data 	Sukses
18	Ubah kata sandi	NULL	Menampilkan form untuk mengubah kata sandi 	Sukses

19	Tekan tombol simpan pada data yang tidak lengkap	Bervariasi	Tidak dapat menyimpan 	Sukses

			 Ubah Kata Sandi Password baru minimal terdiri dari 6 hurufl Nama Pengguna makhrus Kata sandi lama Kata Sandi Baru Ulangi Kata Sandi Simpan Tutup  Ubah Kata Sandi Password tidak sama... Nama Pengguna makhrus Kata sandi lama Kata Sandi Baru Ulangi Kata Sandi Simpan Tutup	

20	Simpan kata sandi lengkap	Nama, Kata sandi, kata sandi baru, ulang kata sandi baru	Kata sandi tersimpan	Sukses
21	Tutup modal kata sandi	NULL	Modal tertutup	Sukses
22	*Fluency* dan *flexibility* memproses 1 file xml	xml	Diproses	Sukses
23	Melihat perkembangan *progress fluency*, dan *flexibility*	xml	Diproses	Sukses
24	Pengukuran Orisinalitas	xml	Diproses	Sukses

4.1.4 Tahap Penilaian

1) Penerapan

Tahap-tahap sebelumnya sudah sudah selesai dilakukan, selanjutnya adalah tahap penerapan dengan aplikasi yang sudah selesai di kembangkan. Penerapan dilaksanakan secara online dengan subjeknya adalah mahasiswa yang telah mengontrak mata kuliah pemrograman dasar, hal ini dilakukan untuk mengefisienkan waktu pembelajaran. Tahap ini melalui langkah-langkah sebagai berikut:

a. Uji Coba Pengguna

Pengguna yang melakukan uji coba merupakan mahasiswa semester 8 dan sudah mengontrak mata kuliah pemrograman dasar. Pada tahap ini total sebanyak 30 pengguna yang telah berpartisi dalam uji coba ini.

b. Uji Coba Aplikasi

Uji coba aplikasi ke pengguna dilakukan untuk mengetahui sejauh mana kelayakan aplikasi ini untuk mengukur kreativitas. Setelah pengguna menggunakan aplikasi ini, pengguna akan mengisi kuesioner multimedia mania untuk siswa.

2) Hasil dan Pembahasan

Tahap terakhir dalam pengembangan aplikasi ini adalah hasil dan pembahasan. Tahap ini dilakukan untuk melihat hasil penerimaan yang di dasarkan pada pengisian instrumen multimedia mania oleh ahli dan pengguna.

1. Hasil validasi ahli

Kelayakan aplikasi dilakukan untuk mengetahui gambaran secara lebih jelas bahwa aplikasi ini layak untuk digunakan, dilihat dari kriteria mekanisme, elemen aplikasi, struktur informasi, dokumentasi, dan kualitas konten.

Tabel 4. 3 Hasil validasi aplikasi dari ahli 1

Aspek	Persentase (%)
Mekanisme	91
Elemen Aplikasi	89
Struktur Informasi	90
Dokumentasi	88.75
Kualitas Konten	86.04
Total Rata-rata	88.83

Tabel 4. 4 Hasil validasi aplikasi dari ahli 2

Aspek	Persentase (%)
Mekanisme	93
Elemen Aplikasi	90
Struktur Informasi	87.5
Dokumentasi	87.5
Kualitas Konten	87.5
Total Rata-rata	89.00

2. Hasil validasi pengguna

Penilaian dan tanggapan dari mahasiswa dilakukan untuk mengetahui secara pasti kelebihan dan kekurangan aplikasi, sehingga bisa menjadi penyesuaian dan gambaran jika dilakukan pengembangan selanjutnya. Penilaian aplikasi dilakukan oleh 30 mahasiswa pada tahap uji coba pertama sebagai pengguna yang berinteraksi langsung aplikasi. Pengujian dilakukan oleh mahasiswa dalam bentuk tanggapan mahasiswa menggunakan

Multimedia Mania 2003 – Student Checklist. Hasil tanggapan mahasiswa digambarkan dalam tabel berikut:

Tabel 4. 5 Penilaian pengguna

Aspek	Persentase (%)
Mekanisme	90
Elemen Aplikasi	100
Struktur Informasi	100
Dokumentasi	100
Kualitas Konten	100
Total Rata-rata	98,4

4.2 Pembahasan

4.2.1 Merancang Aplikasi Pengukuran Kreativitas

Perancangan dan pengembangan aplikasi dilakukan dengan menggunakan metode penelitian dan pengembangan yang telah disesuaikan oleh peneliti menjadi 4 tahap yaitu: pengumpulan data, perencanaan dan analisis, pengembangan, validasi dan uji coba.

Perancangan diawali dengan studi literatur dan pengumpulan data. Pengumpulan data didapat dari buku, jurnal, atau penelitian lain untuk mempertegas permasalah yang ada dan sebagai bahan acuan untuk mempertegas masalah.

Kemudian dilanjutkan dengan membuat desain rancangan aplikasi dengan membuat *flowchart*, *Mock-up*, *UML*, analisis kebutuhan.

Kemudian dilanjutkan dengan proses pengembangan aplikasi diawali dengan membuat database dengan acuan *flowchart* yang telah dibuat selanjutnya dilakukan pembuatan aplikasi menggunakan aplikasi Visual Studio Code untuk penulisan *source code* dengan bantuan aplikasi internet *browser* untuk menampilkan hasil *source code* yang telah melalui tahap pengkodean menjadi sebuah tampilan web.

Dilanjutkan dengan black box testing yang dilakukan untuk mengetahui masukan (*input*) dan keluaran (*output*) sesuai dengan yang diharapkan atau tidak.

4.2.3 Analisis Tanggapan Mahasiswa Terhadap Aplikasi Pengukuran Kreativitas

Tanggapan yang diberikan peserta didik terhadap pengukuran kreativitas ini sangat baik, antusiasme peserta didik dalam menggunakan aplikasi ini pun cukup tinggi, hal ini dapat dibuktikan dari hasil instrumen tanggapan mahasiswa mengenai aplikasi ini mendapatkan nilai rata-rata 98,4% dengan dikategorikan "Sangat Baik". Berikut ini hasil analisis detail pada 3 aspek utama (Mekanisme, Elemen Aplikasi, Struktur Informasi, Dokumentasi, Kualitas Konten) penilaian aplikasi oleh mahasiswa:

a. Aspek Mekanisme

Untuk aspek mekanisme responden memberikan skor rata-rata sebesar 90%. Dengan kata lain secara teknis, navigasi, ejaan, tata bahasa dan penyelesaiannya, aplikasi ini dapat dikategorikan sangat baik.

b. Aspek Elemen Aplikasi

Untuk aspek elemen aplikasi responden memberikan skor rata-rata sebesar 100%. Dengan kata lain desain antarmuka, konten, gambar, video, audio dan perangkat tambahan lainnya pada aplikasi ini digunakan secara efektif.

c. Aspek Struktur Informasi

Untuk aspek struktur informasi responden memberikan skor rata-rata sebesar 100%. Dengan kata lain rangkaian informasi logis dan intuitif serta aplikasi merupakan multimedia dengan desain menarik dan sesuai usia.

d. Aspek Dokumentasi

Untuk aspek dokumentasi responden memberikan skor rata-rata sebesar 100%. Dengan kata lain semua sumber dikutip dengan benar serta terdapat izin penggunaan semua asset dan hak cipta penggunaan asset tertera pada aplikasi ini.

e. Aspek Kualitas Konten

Untuk aspek dokumentasi responden memberikan skor rata-rata sebesar 100%. Dengan kata lain aplikasi menunjukan bukti signifikan dalam keaslian pengembangannya serta semua konten aplikasi mendukung tujuan pembelajaran yang diharapkan pada aplikasi ini.

4.2.4 Kelebihan, Kekurangan, dan Kendala

a. Kelebihan Aplikasi Pengukuran Kreativitas

1. Jika dilakukan secara konsisten, aplikasi ini dapat meningkatkan daya kreatif pengguna secara konsisten. Dikarenakan pengguna akan mengetahui pola penilaian kreativitas.
2. Aplikasi sangat cocok untuk pengguna yang ingin belajar pemikiran algoritmik.

b. Kekurangan Aplikasi Pengukuran Kreativitas

1. Pengguna harus secara konsisten menggunakan aplikasi ini, dan secara konsisten membuat proyek untuk mengetahui sejauh mana daya kreativitas pengguna meningkat.
2. Proyek yang diinput belum medukung validasi oleh mentor.
3. *Snap!* tidak dianjurkan untuk pengguna *smartphone*, dan tablet.

c. Kendala

1. Pengguna akan merasa bosan jika secara terus menerus menggunakan aplikasi.
2. Pembuatan 1 proyek di *Snap!* membutuhkan waktu yang cukup lama.
3. Dari masukan pengguna, sulitnya mencari inspirasi untuk ide pembuatan proyek, ditambah dengan keharusan dalam pembuatan proyek secara konsisten.