

**PENGEMBANGAN FRONT-END APLIKASI *SCREENING SPEECH DELAY* BERBASIS WEB MENGGUNAKAN METODE SCRUM**

**(Skripsi)**

**Oleh**

**SELVIA ELDINA  
NPM 1915061007**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2023**

**PENGEMBANGAN FRONT-END APLIKASI SCREENING SPEECH  
DELAY BERBASIS WEB MENGGUNAKAN METODE SCRUM**

Oleh  
**SELVIA ELDINA**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA TEKNIK**

**Pada**

**Program Studi Teknik Informatika  
Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2023**

## ABSTRAK

# PENGEMBANGAN FRONT-END APLIKASI *SCREENING SPEECH DELAY* BERBASIS WEB MENGGUNAKAN METODE SCRUM

Oleh

**SELVIA ELDINA**

Screening speech delay sangat penting dilakukan terhadap anak pada usia 6 bulan hingga 5 tahun guna mendeteksi adanya indikasi potensi masalah dalam perkembangan bicara (*speech delay*). Namun, kebanyakan rumah terapi atau rumah sakit masih menerapkan metode screening secara mengisi *form screening speech delay* manual dengan datang ke tempat dokter tumbuh kembang anak atau rumah terapi, tidak semua daerah di seluruh Indonesia memiliki tempat untuk melakukan *screening speech delay*. Penelitian ini bertujuan mengembangkan aplikasi berbasis web untuk melakukan *screening speech delay* dengan menerapkan metode Scrum. Aplikasi ini mencakup fitur utama *screening speech delay*, menampilkan hasil dan analisa *screening*, serta grafik pemantauan perkembangan hasil *screening*. Proses pengembangan aplikasi menggunakan Scrum dilakukan selama 5 sprint dengan 21 item dalam *product backlog*. Tahapan penelitian dengan metode Scrum yaitu *Sprint Planning*, *Daily Scrum (Daily Meeting dan Development)*, *Sprint Review* dan *Sprint Retrospective*. Uji coba aplikasi dilakukan melalui pengujian *Black-box testing* yang difokuskan pada fungsionalitas fitur-fitur aplikasi. Sebanyak 5 fitur diuji melalui 21 skenario tes, hasilnya menunjukkan bahwa seluruh fitur yang diuji berjalan dengan sesuai dan sesuai dengan kebutuhan fungsional yang telah ditentukan. Hasil evaluasi *User Experience Questionnaire (UEQ)* yang melibatkan 25 responden, aplikasi web *Screening Speech Delay* mendapatkan nilai *mean excellent* atau sangat baik pada kategori *attractiveness*, *efficiency*, *dependability*, *stimulation* dan *novelty*, serta nilai *mean good* untuk kategori *perspicuity*.

Kata kunci: *Screening Speech Delay*, Web, Scrum, UEQ

## **ABSTRACTION**

### **DEVELOPMENT OF WEB-BASED FRONT-END APPLICATION FOR SCREENING SPEECH DELAY USING SCRUM METHOD**

*By*

**SELVIA ELDINA**

*Screening for speech delay is crucial for children aged 6 months to 5 years to detect potential issues in speech development. However, most therapy centers or hospitals still employ manual screening methods, where individuals fill out speech delay screening forms by visiting a child development doctor or therapy center. Not all regions in Indonesia have facilities for conducting speech delay screenings. This research aims to develop a web-based application for speech delay screening using the Scrum methodology. The application includes key features for speech delay screening, displaying screening results and analysis, as well as monitoring the development progress of screening outcomes. The application development process using Scrum took place over 5 sprints, involving 21 items in the product backlog. The Scrum method research phases included Sprint Planning, Daily Scrum (Daily Meetings and Development), Sprint Review, and Sprint Retrospective. Application testing was conducted through Black-box testing, focusing on the functionality of application features. A total of 5 features were tested through 21 test scenarios, and the results showed that all tested features functioned correctly and met the predefined functional requirements. The evaluation results from the User Experience Questionnaire (UEQ), involving 25 respondents, indicated that the Screening Speech Delay web application received an average rating of "excellent" for attractiveness, efficiency, dependability, stimulation, and novelty categories, as well as an average rating of "good" for the perspicuity category.*

*Keywords: Screening Speech Delay, Web, Scrum, UEQ*



Judul Skripsi : **PENGEMBANGAN FRONT-END APLIKASI  
SCREENING SPEECH DELAY BERBASIS  
WEB MENGGUNAKAN METODE SCRUM**

Nama Mahasiswa : **Selvia Eldina**

Nomor Pokok Mahasiswa : 1915061007

Program Studi : Teknik Informatika

Fakultas : Teknik

**MENYETUJUI**

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

  
**Ing. Hery Dian Septama, S.T.**

NIP 19850915 200812 1 001

  
**Puput Budi Wintoro, S.Kom., M.T.I.**

NIP 19841031 201903 1 004

2. Mengetahui

Ketua Jurusan  
Teknik Elektro

Ketua Program Studi  
Teknik Informatika

  
**Herlinawati, S.T., M.T.**

NIP 19710314 199903 2 001

  
**Mona Arif Muda, S.T., M.T.**

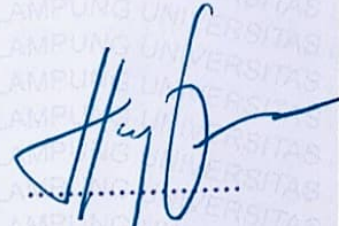
NIP 19711112 200003 1 002



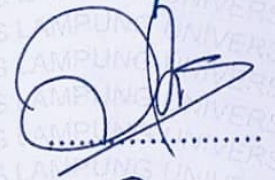
## MENGESAHKAN

### 1. Tim Penguji

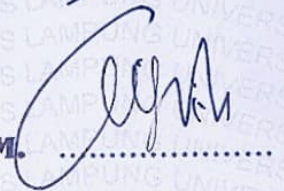
Ketua : **Ing. Hery Dian Septama, S.T.** .....



Sekretaris : **Puput Budi Wintoro, S.Kom., M.T.I.** .....



Penguji : **Ir. Gigih Forda Nama, S.T., M.T.I., IPM.** .....



### 2. Dekan Fakultas Teknik



**Dr. Eng. R. Helmy Fitriawan, S.T., M.Sc.)**

NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **7 Agustus 2023**

## SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya dengan judul “Pengembangan Front-End Aplikasi *Screening Speech Delay* Berbasis Web Menggunakan Metode Scrum” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 07 Agustus 2023

Pembuat pernyataan,



Selvia Eldina

NPM. 1915061007

## RIWAYAT HIDUP



Penulis dilahirkan di provinsi yang disebut Bumi *Rafflesia*, pada tanggal 6 September. Penulis merupakan anak pertama dari dua bersaudara dari pasangan Bapak Eldi Ariyanto dan Ibu Nuti Herawati.

Penulis menyelesaikan pendidikannya di SD Negeri 15 Bengkulu Selatan pada tahun 2013, SMP Negeri 2 Bengkulu Selatan pada tahun 2016 dan SMA Negeri 2 Bengkulu Selatan pada tahun 2019. Pada tahun 2019 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Fakultas Teknik Universitas Lampung melalui jalur SNMPTN. Selama aktif menjadi mahasiswa Teknik Informatika, penulis melakukan beberapa kegiatan, antara lain:

1. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Sosial dan Kewirausahaan, Divisi Kewirausahaan periode 2019/2020;
2. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Pendidikan dan Pengembangan Diri, Divisi Kerohanian periode 2020/2021;
3. Mengikuti program Magang dan Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil Kelas Pengembang *Front-End Web* dan *Back-End* di Dicoding Academy pada tahun 2021;
4. Mengikuti program Magang dan Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil Kelas *Front-End Engineer* di Generasi Gigih X GoTo pada tahun 2022;
5. Melaksanakan Kuliah Kerja Nyata pada bulan Januari sampai dengan Februari 2022 di Tanjung Agung, Kecamatan Seginim, Kabupaten Bengkulu Selatan, Provinsi Bengkulu;



6. Melakukan Kerja Praktik di PT Gawai Mutu Digital pada bulan Agustus sampai dengan September tahun 2022 dengan membuat sistem informasi Taman Kupu-kupu Gita Persada menggunakan *Wordpress*;
7. Mendapatkan Juara 3 pada Kompetisi *Mini Hackathon* dengan tema “*Unlocking HER Potentials*” yang diselenggarakan oleh Web Programming Unpas (WPU) pada bulan Mei 2023;

## MOTTO

“Dan Sungguh, kelak Tuhanmu pasti memberikan karunia-Nya kepadamu, sehingga  
(hati) engkau menjadi puas”

**(Q.S adh-Dhuha : 5)**

“Ingatlah, hanya dengan mengingat Allah hati menjadi tentram”

**(Q.S ar-Ra'du : 28)**

“Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu”

**(Ali bin Abi Thalib)**

“Tidak perlu mencari pengakuan orang lain dan mengejar standar orang lain.  
Mulailah mengenal diri sendiri, mencintai diri sendiri dengan mengapresiasi setiap  
pencapaian diri mulai dari hal-hal kecil”

**(sepertiga.malamku)**

“Semua ada waktunya, jangan membandingkan hidupmu dengan orang lain. Tidak  
ada perbandingan antara matahari dan bulan, mereka bersinar saat waktunya tiba”

**(B.J. Habibie)**

## PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur kepada Allah SWT. atas segala Rahmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. shalawat serta salam teriring kepada Nabi Muhammad SAW sebagai suri tauladan *akhlakul karimah*.

### **KUPERSEMBAHKAN KARYA ILMIAH INI KEPADA:**

“Kedua Orang Tuaku Tercinta

yang senantiasa selalu memberikan yang terbaik, mengorbankan banyak hal dan terus melantunkan doa yang tak henti-hentinya untukku. Kuucapkan terima kasih sebesar-besarnya karena telah mendidik dan membesarkanku menjadi pribadi yang baik dengan kasih sayang, kecukupan dan kebahagiaan yang akan selalu aku syukuri seumur hidupku. Semoga dengan ilmu yang kudapatkan dari hasil jerih payah kalian menyekolahkanku akan menjadi amal jariah bagi kalian”

“Adikku Dwi Putri Larasati yang berharga, terima kasih yang selalu memberikan keceriaan dan memberikan kebahagiaan dalam hidupku. Semoga kelak kita bisa menjadi saudara yang selalu bersama dan membahagiakan kedua orang tua kita”

“Diriku sendiri yang telah berjuang sampai akhir. Terima kasih diriku sudah berjuang karena, tetap bangkit dan melangkah di tengah keraguan banyak orang sungguh tidak mudah. Tapi kamu mampu. Kamu hebat. Maaf sering dengan sengaja menenggelamkan diri pada zona malas dan berujung menyia-nyiakan waktu.”

“Seluruh Keluarga Besar Teknik Elektro 2019 dan Teman-temanku, terima kasih telah menemani dan berjuang bersama di bangku perkuliahan. Begitu banyak cerita yang telah dilalui bersama baik itu susah, senang, tangis yang akan selalu kukenang. Semoga kita bisa menjadi orang-orang yang sukses di kemudian hari”

“Almamater tercinta, Universitas Lampung dan Jurusan Teknik Elektro”

## SANWACANA

Puji syukur penulis panjatkan kepada kehadirat Allah SWT. yang telah memberikan rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul “Pengembangan Front-End Aplikasi *Screening Speech Delay* Berbasis Web Menggunakan Metode Scrum”. Selama masa penelitian penulis mendapatkan banyak bantuan, bimbingan dan dukungan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Allah yang senantiasa memberikan nikmat kemudahan dan kelancaran kepada penulis serta Rasulullah Muhammad yang menjadi suri tauladan selama penelitian berlangsung;
2. Kedua orang tua tercinta dan keluarga penulis yang selalu memberikan dukungan, doa, kasih sayang tiada akhir dan mengingatkan penulis untuk bangkit dalam menyelesaikan penelitian ini;
3. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
4. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
5. Bapak Mona Arif Muda, S.T.,M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung telah membantu proses kelancaran perkuliahan;
6. Bapak Ing. Hery Dian Septama, S.T. selaku Pembimbing Akademik dan Pembimbing Utama yang selalu meluangkan waktunya untuk memberikan bimbingan akademik selama perkuliahan dan memudahkan penulis dalam menyelesaikan penelitian;
7. Bapak Puput Budi Wintoro, S.Kom., M.T.I., selaku Pembimbing Pendamping yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik;



8. Bapak Ir. Gigih Forda Nama, S.T., M.T.I., IPM., selaku Penguji yang telah memberikan banyak saran dan masukan terhadap penelitian ini;
9. Pihak Rumah Terapi Al-birru yang telah membantu penulis selama melakukan penelitian;
10. Mbak Rika selaku *Admin* Program Studi Teknik Informatika yang telah banyak membantu penulis dalam urusan administrasi selama perkuliahan dan penelitian;
11. Silvia Naim sebagai *partner* penulis yang telah membantu dan memberi dukungan selama melakukan penelitian;
12. Reistha Ramadhanty dan Silvia Naim sebagai teman dekat penulis semasa perkuliahan hingga saat ini yang menemani, memberikan dukungan, dan motivasi.
13. Alfiyah Widiyaningsih, Dwi Liliyawati, Meilika Dwi Putri, Nilam Cahya, Yovanta Anjelina dan segenap teman-teman TI 2019 yang telah menjadi rumah bagi penulis selama masa perkuliahan. Menemani, membantu dan memberikan pundak bagi penulis dikala susah dan sedih;
14. Istimewa Mega Haztuti dan Tasya Pitria Azmalina sebagai teman semasa SMP sampai sekarang yang selalu mendengarkan keluh kesah dari awal perkuliahan hingga di titik ini dengan memberikan semangat dan motivasi;
15. Keluarga besar Teknik Elektro Angkatan 2019 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terima kasih telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama;

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang Teknik Informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 7 Agustus 2023  
Penulis,

Selvia Eldina

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI.....</b>	<b>i</b>
<b>DAFTAR GAMBAR.....</b>	<b>iii</b>
<b>DAFTAR TABEL .....</b>	<b>v</b>
<b>I. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	5
<b>II. TINJAUAN PUSTAKA.....</b>	<b>6</b>
2.1 Bicara dan Bahasa .....	6
2.2 Website.....	7
2.3 <i>Speech Delay</i> .....	7
2.4 Metode <i>Screening Speech Delay</i> .....	8
2.5 UML ( <i>Unified Modeling Language</i> ).....	8
2.6 <i>Mock-up</i> .....	12
2.7 Metode Scrum .....	12
2.8 <i>Front-End</i> .....	15
2.9 <i>Tech Stack</i> .....	16
2.9.1 React Js.....	17
2.9.2 Tailwind CSS .....	18
2.10 Pengujian Sistem .....	19
2.10.1 <i>Blackbox</i> .....	19
2.10.2 <i>User Experience Questionnaire (UEQ)</i> .....	19
2.11 Penelitian Terkait .....	20
<b>III. METODOLOGI PENELITIAN.....</b>	<b>26</b>

3.1	Waktu dan Tempat .....	26
3.1.1	Jadwal Penelitian.....	26
3.2	Alat Penelitian .....	26
3.2.1	Perangkat Keras ( <i>Hardware</i> ) .....	27
3.2.2	Perangkat Lunak ( <i>Software</i> ).....	27
3.3	Pengumpulan Data .....	28
3.3.1	Wawancara.....	28
3.3.2	Perumusan Masalah .....	29
3.3.3	Studi Literatur .....	30
3.4	Tahapan Penelitian .....	30
3.4.1	<i>User Story</i> .....	32
3.4.2	<i>Product Backlog</i> .....	34
3.4.3	<i>Sprint Planning</i> .....	56
3.4.4	<i>Daily Scrum</i> .....	56
3.4.5	<i>Sprint Review</i> .....	59
3.4.6	<i>Sprint Retrospective</i> .....	59
3.5	Penyusunan Laporan .....	59
<b>IV.</b>	<b>HASIL DAN PEMBAHASAN.....</b>	<b>60</b>
4.1	Hasil.....	60
4.1.1	<i>Sprint 1</i> .....	60
4.1.2	<i>Sprint 2</i> .....	68
4.1.3	<i>Sprint 3</i> .....	77
4.1.4	<i>Sprint 4</i> .....	85
4.1.5	<i>Sprint 5</i> .....	103
4.2	Pembahasan .....	122
4.2.1	Capaian Penelitian Aplikasi Web <i>Screening Speech Delay</i> .....	122
4.2.2	Kendala Dalam Pengembangan Aplikasi.....	122
<b>V.</b>	<b>KESIMPULAN .....</b>	<b>124</b>
5.1	Kesimpulan.....	124
5.2	Saran .....	125
	<b>DAFTAR PUSTAKA .....</b>	<b>126</b>
	<b>LAMPIRAN.....</b>	<b>130</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Scrum (Scrum.org).....	12
Gambar 3.1 Tahapan Scrum (Scrum.org).....	31
Gambar 3.2 <i>Use Case Diagram</i> Aplikasi <i>Screening Speech Delay</i> (Birrussment).....	39
Gambar 3.3 <i>Activity Diagram</i> <i>Home Page</i> .....	41
Gambar 3.4 <i>Activity Diagram</i> <i>Home Page</i> .....	41
Gambar 3.5 <i>Activity Diagram</i> <i>Registrasi</i> .....	42
Gambar 3.6 <i>Activity Diagram</i> <i>Login</i> .....	43
Gambar 3.7 <i>Activity Diagram</i> <i>Update Password</i> .....	44
Gambar 3.8 <i>Activity Diagram</i> <i>Profile User</i> .....	45
Gambar 3.9 <i>Activity Diagram</i> <i>Screening Test</i> .....	46
Gambar 3.10 <i>Activity Diagram</i> <i>Riwayat Screening</i> .....	47
Gambar 3.11 <i>Activity Diagram</i> <i>Subscription</i> .....	48
Gambar 3.12 <i>Mockup</i> <i>Home Page</i> <i>Section Hero</i> .....	49
Gambar 3.13 <i>Mockup</i> <i>Home Page</i> <i>Content Section</i> .....	49
Gambar 3.14 <i>Mockup</i> <i>Home Page</i> <i>Section Testimoni</i> .....	50
Gambar 3.15 <i>Mockup</i> <i>Home Page</i> <i>Section Clients</i> .....	50
Gambar 3.16 <i>Mockup</i> <i>Sitemap</i> dan <i>Footer</i> .....	50
Gambar 3.17 <i>Mockup</i> <i>Register</i> .....	51
Gambar 3.18 <i>Mockup</i> <i>Login</i> .....	51
Gambar 3.19 <i>Mockup</i> <i>About Page</i> .....	52
Gambar 3.20 <i>Mockup</i> <i>FAQ Section</i> .....	52
Gambar 3.21 <i>Mockup</i> <i>Halaman Pre-screening</i> dan <i>screening</i> .....	53
Gambar 3.22 <i>Mockup</i> <i>Hasil Screening</i> .....	53
Gambar 3.23 <i>Mockup</i> <i>Update Profile</i> .....	54
Gambar 3.24 <i>Mockup</i> <i>Checkout</i> .....	54
Gambar 3.25 <i>Mockup</i> <i>Payment</i> .....	55
Gambar 3.26 <i>Mockup</i> <i>Riwayat Screening</i> .....	55
Gambar 3.27 <i>Mockup</i> <i>Form Update Password</i> .....	56
Gambar 3.28 <i>Kuesioner User Experience Questionnaire (UEQ)</i> .....	58
Gambar 4.1 <i>Kode Program</i> <i>Halaman Registrasi</i> .....	61
Gambar 4.2 <i>Kode Program</i> <i>Halaman Login</i> .....	62
Gambar 4.3 <i>Kode Program</i> <i>Halaman About</i> .....	63
Gambar 4.4 <i>Kode Program</i> <i>Homepage</i> .....	63
Gambar 4.5 <i>Tampilan</i> <i>Halaman Registrasi</i> .....	64
Gambar 4.6 <i>Tampilan</i> <i>Halaman Login</i> .....	65
Gambar 4.7 <i>Tampilan</i> <i>Homepage</i> .....	65



Gambar 4.8 Tampilan Halaman <i>About</i> .....	66
Gambar 4.9 Tampilan Hasil <i>Sprint Retrospective</i> .....	67
Gambar 4.10 Kode Program <i>Slicing</i> Halaman <i>Profile</i> Pengguna.....	69
Gambar 4.11 Kode Program <i>Slicing</i> Halaman <i>Pra Screening</i> .....	70
Gambar 4.12 Kode Program <i>Slicing</i> Halaman <i>Screening</i> .....	71
Gambar 4.13 Kode Program <i>Slicing</i> Halaman Hasil <i>Screening</i> .....	72
Gambar 4.14 Tampilan Hasil <i>Slicing</i> Halaman <i>Profile</i> Pengguna.....	73
Gambar 4.15 Tampilan Hasil <i>Slicing</i> Halaman <i>Pra Screening</i> .....	74
Gambar 4.16 Tampilan Hasil <i>Slicing</i> Halaman <i>Screening</i> .....	74
Gambar 4.17 Tampilan <i>Slicing</i> Halaman Hasil <i>Screening</i> .....	75
Gambar 4.18 Tampilan Hasil <i>Sprint</i> Kedua <i>Retrospective</i> .....	76
Gambar 4.19 Kode Program <i>Slicing</i> Halaman Riwayat <i>Screening</i> .....	78
Gambar 4.20 Kode Program <i>Slicing</i> Halaman <i>Change Password</i> .....	79
Gambar 4.21 Kode Program <i>Slicing</i> Halaman <i>Subscription</i> .....	80
Gambar 4.22 Kode Program <i>Slicing</i> Halaman <i>Checkout</i> .....	81
Gambar 4.23 Tampilan <i>Slicing</i> Halaman Riwayat <i>Screening</i> .....	82
Gambar 4.24 Tampilan <i>Slicing</i> Halaman <i>Change Password</i> .....	82
Gambar 4.25 Tampilan <i>Slicing</i> Halaman <i>Subscription</i> .....	83
Gambar 4.26 Tampilan <i>Slicing</i> Halaman <i>Checkout</i> .....	84
Gambar 4.27 Tampilan Hasil <i>Sprint</i> Ketiga <i>Retrospective</i> .....	85
Gambar 4.28 Kode Program Integrasi API <i>Sign Up</i> .....	88
Gambar 4.29 Kode Program Integrasi API <i>Login</i> .....	89
Gambar 4.30 Kode Program Integrasi API <i>Logout</i> .....	90
Gambar 4.31 Kode Program Integrasi API <i>Header</i> Utama .....	90
Gambar 4.32 Kode Program Integrasi API <i>Profile</i> Pengguna .....	91
Gambar 4.33 Kode Program Integrasi API Mulai <i>Screening</i> .....	92
Gambar 4.34 Kode Program Integrasi API Mengirimkan Jawaban <i>Screening</i> ....	93
Gambar 4.35 Kode Program Integrasi API Riwayat <i>Screening</i> .....	94
Gambar 4.36 Kode Program Integrasi API <i>Change Password</i> .....	94
Gambar 4.37 Tampilan Setelah dilakukan Integrasi API <i>Sign Up</i> .....	96
Gambar 4.38 Tampilan Setelah dilakukan Integrasi API <i>Login</i> .....	96
Gambar 4.39 Tampilan Hasil Integrasi API <i>showMe Navbar</i> .....	97
Gambar 4.40 Tampilan Hasil Integrasi API <i>Profile</i> Pengguna.....	97
Gambar 4.41 Tampilan Hasil Integrasi API <i>Screening</i> .....	98
Gambar 4.42 Tampilan Hasil Integrasi API Hasil <i>Screening</i> .....	99
Gambar 4.43 Tampilan Hasil Integrasi API Riwayat <i>Screening</i> .....	100
Gambar 4.44 Tampilan Hasil Integrasi API <i>Change Password</i> .....	101
Gambar 4.45 <i>Activity Diagram Dashboard</i> Grafik.....	106
Gambar 4.46 <i>Usecase Dashboard</i> Grafik .....	107
Gambar 4.47 <i>Mockup Dashboard</i> Grafik.....	108
Gambar 4.48 Tampilan Hasil Integrasi API <i>Change Password</i> .....	110
Gambar 4.49 Diagram Responden .....	114
Gambar 4.50 Grafik Nilai <i>Mean</i> .....	117
Gambar 4.51 Standar Set Data <i>Benchmark</i> UEQ.....	118
Gambar 4.52 Tampilan <i>Dashboard</i> Grafik <i>Screening</i> .....	119
Gambar 4.53 Tampilan Pengujian <i>User Experience Questionnaire</i> (UEQ) .....	120
Gambar 4.54 Tampilan Hasil <i>Sprint</i> Kelima <i>Retrospective</i> .....	121

## DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol <i>Use Case Diagram</i> .....	9
Tabel 2.2 Simbol <i>activity diagram</i> .....	11
Tabel 3.1 Jadwal Penelitian Desember 2022 hingga Juni 2023.....	26
Tabel 3.2 Perangkat lunak.....	27
Tabel 3.3 Pertanyaan wawancara.....	29
Tabel 3.4 Scrum <i>Team</i> Aplikasi Birrussment .....	31
Tabel 3.5 <i>User Story</i> Aplikasi Birrussment .....	32
Tabel 3.6 <i>Product Backlog</i> Birrussment (Fitur Inti) .....	34
Tabel 3.7 <i>Product Backlog</i> Birrussment (Fitur Tambahan).....	37
Tabel 3.8 Definisi <i>Use Case</i> .....	40
Tabel 4.1 <i>Sprint Backlog</i> dari <i>Sprint 1</i> Birrussment.....	60
Tabel 4.2 Hasil <i>Sprint Review</i> <i>Sprint Backlog</i> dari <i>Sprint 1</i> Birrussment .....	66
Tabel 4.3 <i>Sprint Backlog</i> dari <i>Sprint 2</i> Birrussment .....	68
Tabel 4.4 Hasil <i>Sprint Review</i> <i>Sprint Backlog</i> dari <i>Sprint 2</i> Birrussment .....	75
Tabel 4.5 <i>Sprint Backlog</i> dari <i>Sprint 3</i> Birrussment .....	77
Tabel 4.6 Hasil <i>Sprint Review</i> <i>Sprint Backlog</i> dari <i>Sprint 3</i> Birrussment .....	84
Tabel 4.7 <i>Sprint Backlog</i> dari <i>Sprint 4</i> Birrussment .....	86
Tabel 4.8 Hasil <i>Sprint Review</i> <i>Sprint Backlog</i> dari <i>Sprint 4</i> Birrussment .....	101
Tabel 4.9 Hasil Perubahan <i>Product Backlog</i> Birrussment.....	103
Tabel 4.10 Definisi <i>usecase Dashboard</i> Pengguna.....	107
Tabel 4.11 <i>Sprint Backlog</i> dari <i>Sprint 5</i> Birrussment .....	108
Tabel 4.12 <i>Black-box Testing</i> Pengguna Melakukan Registrasi.....	111
Tabel 4.13 <i>Black-box Testing</i> Pengguna Melakukan <i>Login</i> .....	111
Tabel 4.14 <i>Black-box Testing</i> Pengguna Melakukan <i>Screening Test</i> .....	112
Tabel 4.15 <i>Black-box Testing</i> Pengguna Melakukan <i>Edit Profile</i> .....	113
Tabel 4.16 <i>Black-box Testing</i> Pengguna Melakukan <i>Change Password</i> .....	113
Tabel 4.17 Distribusi Jawaban Responden .....	115
Tabel 4.18 Distribusi Jawaban Responden .....	116
Tabel 4.19 Hasil <i>Mean</i> , <i>Variance</i> , dan Simpangan Baku .....	116
Tabel 4.20 Hasil Rata-rata Berdasarkan Skala UEQ ( <i>Mean</i> dan <i>Variance</i> ) .....	117

Tabel 4.21 Set Data <i>Benchmark</i> .....	118
Tabel 4.22 Hasil <i>Sprint Review Sprint Backlog</i> dari <i>Sprint 5 Birrussment</i> .....	120

## I. PENDAHULUAN

### 1.1 Latar Belakang

*Screening* merupakan kegiatan medis yang dilakukan untuk mendeteksi dan menganalisis potensi masalah pada tubuh manusia. *Screening* biasanya dilakukan dengan pengisian pada formulir untuk melengkapi data fakta gejala yang dapat mendiagnosis masalah pada tubuh manusia. Internet memegang peran penting dalam melakukan pembaruan teknologi guna untuk mempermudah aktivitas manusia. Teknologi akan sangat membantu tim medis dalam menyimpan data *screening* guna melakukan penelitian lebih lanjut terhadap permasalahan yang dianalisis.

Internet merupakan bentuk kemajuan teknologi informasi dan komunikasi yang memberikan peluang untuk mendapatkan informasi dengan cepat, akurat, dan terjangkau[1]. Internet memfasilitasi pertukaran informasi antara tenaga medis dan pasien. Dalam hal pengumpulan data, internet juga dapat digunakan untuk survei kesehatan secara *online*. Dengan menggunakan formulir daring, data *screening* medis dapat dikumpulkan dengan efisien. Bentuk umum informasi yang dipublikasikan di internet, seperti website, media sosial, aplikasi, *podcast*, e-jurnal, dan forum *online*.

Website menjadi layanan yang disediakan internet dan paling banyak digunakan. Website dapat menjadi wadah untuk melakukan *screening* berbasis *online* dengan memanfaatkan perkembangan internet yang sangat cepat berkembang. Kelebihan website dibanding dengan fasilitas internet yang lainnya adalah website mampu menampilkan informasi berupa teks, gambar, audio ataupun video yang interaktif.



Website dapat dijadikan sarana untuk mengenalkan produk kepada publik secara detail [2].

Menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), tahun 2022 pengguna internet di Indonesia mencapai sekitar 210 juta. Perkembangan pengguna internet diimbangi dengan perkembangan aplikasi berbasis web. Aplikasi berbasis web dapat berjalan diberbagai sistem operasi dengan bantuan perambah web. Aplikasi berbasis web termasuk aplikasi yang ringan digunakan, oleh karena itu aplikasi web menjadi pilihan bagi sebagian orang [3]. Sebelum membangun aplikasi berbasis web pemilihan metode pengembangan perangkat lunak sangat penting, agar saat pengerjaan proyek terstruktur pengerjaannya. Metode Scrum adalah metode pengelolaan proyek yang berfokus pada kolaborasi tim dan fleksibel terhadap perubahan *task* aplikasi. Scrum dalam pengerjaan proyek dipecah menjadi serangkaian iterasi (*sprint*).

*Speech delay* merupakan kondisi dimana kemampuan bicara dan bahasa pada anak dibawah rata-rata anak yang sebaya dengannya [4]. *Speech delay* pada anak adalah masalah gangguan perkembangan anak yang selalu dipermasalahkan di seluruh dunia [5]. Pada tahun 2022, masalah *speech delay* menyentuh angka 10% usia prasekolah. Di Indonesia, angka pasti untuk jumlah anak yang mengalami *speech delay*, data yang tersedia sangat terbatas dan tidak *update* terkini. Provinsi Lampung menurut Komdat Kesmas (Seksi Kesehatan Keluarga dan Gizi) Dinas Kesehatan Provinsi Lampung pada tahun 2020 tercatat anak melakukan *speech therapy* untuk usia 28 hari hingga satu tahun terdapat 15 anak berjenis kelamin laki-laki dan 2 anak berjenis kelamin perempuan, pada usia 1 tahun hingga 4 tahun tercatat 37 anak berjenis kelamin laki-laki dan 18 orang berjenis kelamin perempuan. Menurut data penelitian dari Rumah Terapi Al-birru Bandar Lampung tahun 2022 tercatat 96 anak terdiagnosis *speech delay* dan melakukan *speech therapy*. Masalah *speech delay* dapat diketahui dengan cara melakukan *screening* pada anak yang menunjukkan gejala *speech delay*.

*American Academy of Pediatrics* (AAP) membuat panduan klinis yang merekomendasikan agar anak pada saat berusia kurang dari 36 bulan melakukan

*screening speech delay*. Dalam panduan tersebut, pada usia 9, 18, dan 24 atau 30 bulan dianggap tepat untuk melakukan *screening speech delay*, *screening* perkembangan anak, dan mengukur perkembangan bahasa anak [6]. Namun, untuk melakukan *screening speech delay* pada rumah terapi Al-birru masih secara manual, kurang terorganisir untuk penyimpanan hasil *screening* dan mengharuskan orang tua untuk datang ke tempat terapi untuk melakukan *screening* sehingga membutuhkan *effort* waktu dan transportasi.

Berdasarkan uraian masalah di atas, dalam penelitian ini akan membangun sebuah aplikasi berbasis web yang digunakan untuk melakukan *screening speech delay* pada anak yang dilakukan oleh para orang tua. Aplikasi *screening speech delay* “Al-birru Assessment (Birrusment)”, pada aplikasi ini terdapat beberapa pertanyaan *screening* yang dikelompokkan berdasarkan kategori usia anak. Sebelum menggunakan aplikasi ini pengguna harus melakukan autentikasi akun. Setelah itu, pengguna wajib untuk melengkapi *profile* agar dapat memulai tes *screening*. Kuesioner *screening* diisi oleh orang tua atau orang terdekat yang melihat proses tumbuh kembang anak, setelah *submit* diakhir pertanyaan maka pengguna akan memperoleh hasil tesnya. Hasil dari *screening* berupa persentase dan analisis terindikasi atau tidak. Pada sistem ini juga pengguna dapat melihat perkembangan hasil pertama hingga hasil tes terakhir dilakukan bentuk hasilnya berupa grafik. Aplikasi *screening* berbasis web ini diharapkan dapat menjadi solusi agar memudahkan para orang tua dapat *screening speech delay* secara digital. Dengan adanya aplikasi ini diharapkan para orang tua lebih memperhatikan masalah *screening speech delay* pada anak, sehingga tidak terjadi keterlambatan dalam melakukan stimulasi dan intervensi *speech delay*.

## **1.2 Perumusan Masalah**

Rumusan masalah dalam penelitian ini yaitu bagaimana mengembangkan aplikasi *screening speech delay* berbasis website menggunakan metode scrum, menampilkan analisa diagnosa deteksi awal berdasarkan *screening* yang

dilakukan, dan dapat menampilkan *output* berupa grafik untuk melihat atau memantau perubahan *screening* awal dengan akhir.

### 1.3 Tujuan Penelitian

Tujuan penelitian ini adalah, sebagai berikut.

1. Mengembangkan aplikasi *screening speech delay* berbasis web dengan menerapkan metode Scrum.
2. Mengembangkan aplikasi pada bidang *front-end (clients side)* menggunakan *library* React JS
3. Mengevaluasi fungsionalitas aplikasi menggunakan *Black-box testing* dan *user experience* dari aplikasi menggunakan *User Experience Questionnaire (UEQ)*.

### 1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat mempermudah para orang tua dalam melakukan *screening speech delay*, melihat hasil *screening*, dan melihat grafik perkembangan anak dari hasil *screening*, lalu *screening* dilakukan secara *online* sehingga mengurangi resiko keterlambatan dalam stimulasi dan intervensi jika anak terindikasi *speech delay*, serta penelitian ini dapat menghasilkan data yang berharga tentang perkembangan bicara pada anak.

### 1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Membahas mengenai pengembangan *Aplikasi Screening Speech Delay* berbasis web dengan kuesioner pada buku *ASSESSMENT in Speech-Language Pathology*.
2. Hasil analisa *screening* bukan hasil akhir yang dapat menyatakan bahwa anak tersebut mengalami *speech delay*. Keputusan yang dapat menyatakan bahwa

anak mengalami *speech delay* adalah ahli terapi wicara atau dokter tumbuh kembang anak.

3. Aplikasi *screening speech delay* yang dikembangkan menggunakan metode *screening* dan perhitungan *score screening* hanya menggunakan perhitungan dengan rumus matematis yang diarahkan oleh terapis Rumah Terapi Al-birru.
4. Implementasi aplikasi hanya terbatas pada pengembangan *front-end* dan penerapan integrasi API pada aplikasi yang disediakan oleh pengembang *back-end*, serta tidak melibatkan pengembangan dari sisi *back-end*.

## 1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini terdiri dari 5 (lima) bab sebagai berikut:

### BAB I : PENDAHULUAN

Bab ini menguraikan latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, dan sistematika penulisan laporan.

### BAB II : TINJAUAN PUSTAKA

Bab ini berisi tentang teori dasar, prinsip, dan penelitian terkait yang digunakan dalam Pengembangan *Aplikasi Screening Speech Delay Berbasis Web*.

### BAB III : METODE PENELITIAN

Bab ini membahas mengenai metode penelitian yang digunakan dalam pengembangan *Aplikasi Screening Speech Delay Berbasis Web*.

### BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan yang diperoleh dalam penelitian.

### BAB V : KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan berdasarkan hasil pembahasan penelitian yang diharapkan dapat meningkatkan wawasan dan dijadikan sebagai bahan perbaikan untuk kedepannya.



## II. TINJAUAN PUSTAKA

### 2.1 Bicara dan Bahasa

Bahasa merupakan sebuah sarana untuk berkomunikasi agar seseorang dapat menyampaikan perasaan, pemikiran, keinginan, dan pendapat kepada orang lain yang diajak berkomunikasi. Sehingga, bahasa memiliki peran sosial penting dalam berkomunikasi dengan sesama manusia. Saat menggunakan bahasa untuk berkomunikasi memiliki tujuan agar dapat menarik perhatian pendengar. Manusia memiliki dua cara saat berkomunikasi, yaitu verbal dan non-verbal. Berkomunikasi secara verbal dilakukan menggunakan media lisan ataupun tulisan, sedangkan non-verbal dilakukan menggunakan media simbol, seperti tanda lalu lintas yang dapat diterjemahkan kedalam bahasa manusia [7]. Perkembangan kemampuan berbahasa memegang peranan penting dalam masa pertumbuhan dan perkembangan anak. Sehingga, para orang tua memiliki peran penting dalam mengawasi perkembangan berbahasa anak.

Menurut Hurlock, bicara merupakan bentuk bahasa melalui kata-kata untuk mengungkapkan keinginan kepada orang lain. Bicara membutuhkan keterampilan yang saling menyeimbangkan antara mental dan otot sehingga mengeluarkan bunyi yang memiliki arti. Fase perkembangan bicara yang dilalui oleh anak merupakan hal yang penting karena fase tersebut anak dapat berbicara dengan intonasi yang baik dan mampu menyampaikan maksud dengan tujuan yang jelas [8]. Seorang anak dinilai memiliki keterlambatan bicara dan bahasa jika perkembangan bicara dan bahasanya secara signifikan berada di bawah kemampuan anak seusianya. Perkembangan bicara dan bahasa anak tersebut masih dalam sekuen yang benar, namun lebih lambat dari yang diharapkan. Sedangkan anak-anak dengan gangguan bicara dan bahasa, perkembangan bicara dan bahasanya secara kualitatif berbeda dari anak pada umumnya [6].

## 2.2 Website

Website merupakan kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau bergerak, animasi, suara, ataupun gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk serangkaian bangunan yang saling terikat yang masing-masing dihubungkan dengan jaringan halaman. Hubungan antara halaman web dengan halaman web lainnya disebut dengan *Hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *Hypertext*. Jenis website ada dua, yaitu website dinamis dan statis. Website dinamis merupakan sebuah website yang menyediakan konten atau isi yang selalu berubah setiap saat (*real-time*). Bahasa pemrograman yang digunakan antara lain PHP, JavaScript, ReactJs dan memanfaatkan *database* My SQL, MongoDB, PostgreSQL, dan Oracle. Contoh umum seperti media sosial, situs *e-commerce*, blog, forum, atau situs berita yang menampilkan konten yang berubah secara teratur. Sedangkan, website statis merupakan website yang kontennya sangat jarang berubah atau bahkan tidak berubah. Bahasa pemrograman yang digunakan HTML, CSS, dan belum memanfaatkan *database*. Contoh umumnya seperti website profil perusahaan atau blog yang tidak diperbarui secara teratur [9].

## 2.3 Speech Delay

*Speech delay* merupakan keadaan perkembangan bicara yang memiliki kualitas di bawah rata-rata [8]. *Speech delay* berhubungan dengan gangguan tumbuh kembang anak. Gangguan pendengaran menjadi salah satu faktor anak memiliki *speech delay*. Perbedaan *speech delay* dan *speech disorder* yaitu *speech delay* kemampuan bicara anak masih dapat berkembang namun waktunya lebih lambat dari anak seusianya. Sedangkan, *speech disorder* kemampuan bicara anak tidak berkembang seperti anak pada umumnya. *Speech delay* dikategorikan menjadi dua sesuai dengan penyebabnya, *primary speech delay* dan *secondary speech delay*. *Primary speech delay* merupakan kondisi dimana penyebabnya dari *speech delay* tidak diketahui, sedangkan *secondary speech delay* adalah kondisi *speech delay* yang disebabkan oleh kondisi lain seperti gangguan atau kecacatan pendengaran,

autisme, dan kecacatan syaraf. Keterlambatan berbicara tidak boleh dibiarkan karena dapat berpengaruh terhadap kemampuan belajar anak sehingga dengan keadaan yang sulit berkomunikasi tidak jarang anak mengalami keterlambatan berbicara memiliki masalah kesulitan belajar (*learning disabilities*).

#### **2.4 Metode *Screening Speech Delay***

Metode *screening speech delay* merupakan kegiatan pemeriksaan awal *speech delay* yang bertujuan untuk mendiagnosa *speech delay* pada anak. Sehingga apabila terindikasi *speech delay* maka para orang dan dokter atau terapis dapat melakukan stimulasi dan intervensi *speech delay* sejak dini. *American Academy of Pediatrics* (AAP) membuat panduan klinis yang merekomendasikan kunjungan anak saat berusia kurang dari 36 bulan untuk dilakukan *screening* yang sebaiknya menggunakan perangkat perkembangan anak yang telah terstandarisasi. Dalam panduan tersebut, usia 9, 18, dan 24 atau 30 bulan dianggap tepat untuk *screening* perkembangan anak, termasuk perkembangan bahasa [6]. Intervensi harus dilakukan sejak dini, karena fase tumbuh kembang atau usia keemasan anak berada di 0-5 tahun. Setelah melakukan *screening* dan anak terindikasi *speech delay* maka intervensi dapat dilakukan pada fase keemasan anak (0-5 tahun) dan tingkat keberhasilan penanganan dapat lebih tinggi karena perkembangan kecerdasan, sosial emosi, dan keterampilan motorik sedang berkembang dengan pesat pada fase usia 0-5 tahun. Aplikasi *Birrusment* menggunakan *resource* pertanyaan *screening* berdasarkan Rumah Terapi Al-birru yang menerapkan metode *screening* menggunakan *form Assessment of Language Development* dari buku *ASSESSMENT in Speech-Language Pathology* [10].

#### **2.5 UML (*Unified Modeling Language*)**

UML (*Unified Modeling Language*) merupakan bahasa yang menampilkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian sebuah aplikasi *software development* berbasis OO (*Object-*

*Oriented*). Tujuan UML adalah untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membangun dan merancang model aplikasi. UML (*Unified Modeling Language*) memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep bisnis proses, penulisan kelas dalam bahasa pemrograman yang spesifik, skema database, dan komponen yang diperlukan dalam sistem *software* [11].


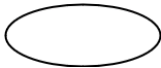
Berikut penjelasan jenis UML yang digunakan dalam penelitian ini, sebagai berikut.


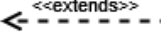
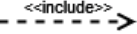

### 1. Use Case Diagram

Use case diagram merupakan model perilaku dari sistem atau aplikasi yang akan dibangun. Tujuan pembuatan *use case diagram* untuk mendeskripsikan interaksi antara aktor dengan sistem atau aplikasi yang dibangun. Tiga elemen penting dalam *use case diagram*, yaitu aktor, *use case*, dan *relationship*. Aktor diartikan sebagai siapapun atau apapun (mesin) yang melakukan interaksi dengan aplikasi untuk pertukaran informasi. *Use case* merupakan urutan tindakan yang terkait, baik manual ataupun otomatis, yang bertujuan menyelesaikan sebuah tugas bisnis. *Relationship* adalah gambaran hubungan antara dua simbol dalam sebuah *use case* [12].

Adapun simbol *use case* yang digunakan pada Tabel 2.1, sebagai berikut [13].

Tabel 2.1 Simbol *Use Case Diagram*




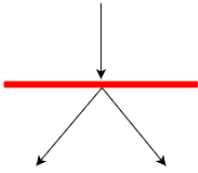
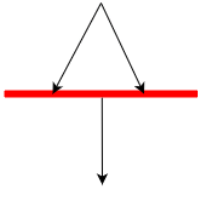
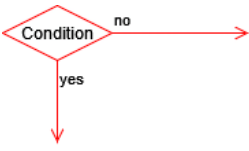
No	Gambar	Nama	Keterangan
1.		Aktor ( <i>actor</i> )	Menspesifikasikan orang, proses, mesin, atau sistem lain yang berinteraksi dengan <i>use case</i> tetapi tidak memiliki peran untuk mengontrol <i>use case</i> . Walaupun simbol dari aktor berupa gambar orang, tetapi tidak semua aktor adalah orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		<i>Use case</i>	Menspesifikasikan fungsionalitas yang disediakan oleh sistem/aplikasi sebagai unit

			yang melakukan sesuatu yang menghasilkan suatu hasil yang terukur, seperti melakukan pertukaran informasi antar unit atau aktor. Penamaan <i>use case</i> biasanya menggunakan kata kerja diawal frase.
3.		Asosiasi/ <i>association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Ekstensi/ <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal Arah panah mengarah pada <i>use case</i> yang ditambahkan. Misalnya, jika pada aplikasi ingin membuka halaman edit data profil, dapat langsung membuka tanpa syarat tertentu
5.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini (induk) untuk menjalankan fungsinya. Misalnya, jika ingin melakukan tes pada sebuah aplikasi, maka <i>user/actor</i> harus login terlebih dahulu
6.		Generalisasi/ <i>generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antar dua <i>use case</i> dimana fungsi satu lebih umum dari lainnya, Misalnya, arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).

## 2. Activity Diagram

*Activity diagram* merupakan diagram yang menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah aplikasi atau sistem pada perangkat lunak. *Activity diagram* berfokus pada gambaran proses-proses yang terjadi pada sebuah sistem. Simbol-simbol pada *activity diagram* terlihat pada Tabel 2.2, sebagai berikut [13].

Tabel 2.2 Simbol *activity diagram*

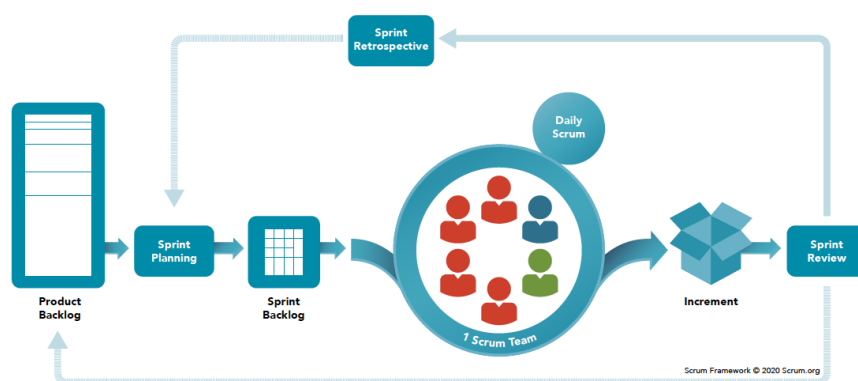
No	Gambar	Nama	Keterangan
1.		Mulai/ <i>Start point</i>	Simbol <i>start</i> digunakan sebagai tanda awal dimulainya aktivitas (status awal).
2.		<i>Endpoint</i>	Simbol <i>endpoint</i> mendeskripsikan akhir dari aktivitas (status akhir).
3.		<i>Activity</i>	Simbol <i>activity</i> digunakan untuk menggambarkan/memperlihatkan suatu proses/kegiatan.
4.		<i>Fork/</i> percabangan	<i>Fork</i> /percabangan, mendeskripsikan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5.		<i>Join/</i> penggabungan	<i>Join</i> /penggabungan, dimana lebih dari satu aktivitas digabungkan menjadi satu.
6.		<i>Decision</i>	<i>Decision</i> , dimana jika terdapat kondisi/pilihan lebih dari satu, yang menunjukkan pilihan aktivitas <i>yes</i> atau <i>no</i> .

## 2.6 Mock-up

*Mock-up* adalah gambaran lengkap dan menyediakan visual atau tampilan dengan simulasi alur perangkat lunak sehingga terlihat seperti perangkat lunak yang sudah jadi [14]. *Mock-up* mencakup elemen desain, estetika, dan detail visual dari antarmuka pengguna. *Mock-up* memberikan pemahaman tampilan *final* produk dan membantu mengkomunikasikan ide yang jelas kepada tim pengembang/*developer*. *Mock-up* sering digunakan dalam presentasi kepada *client* untuk mendapatkan persetujuan sebelum tahap implementasi. Perangkat lunak desain grafis yang digunakan untuk membuat *mock-up* seperti Adobe Photoshop, Sketch, atau Figma. Pada penelitian pengembangan *front-end* aplikasi *speech delay* desain perangkat yang digunakan yaitu Figma.

## 2.7 Metode Scrum

Scrum adalah kerangka metode *agile* untuk menyelesaikan tugas yang kompleks dan selalu berubah. Scrum digunakan untuk membantu orang, tim, dan organisasi menjawab permasalahan yang adaptif yang kompleks, sehingga menghasilkan kreatifitas dan inovasi. *Sprint* merupakan inti dari Scrum yang merupakan batasan waktu pengerjaan pengembangan produk dengan durasi 1-4 minggu sesuai dengan kesepakatan waktu yang ditentukan. *Sprint* memiliki durasi yang konsisten sepanjang siklus pengembangan produk. *Sprint* yang baru akan dimulai ketika *sprint* sebelumnya telah selesai [15].



Gambar 2.1 Scrum (Scrum.org)

Kerangka kerja Scrum adalah metode turunan dari *Agile*, sehingga metode scrum mengikuti prinsip-prinsip *Agile*. Metode scrum berfokus pada manajemen proyek yang mana keadaan dalam perencanaan awalnya menyusahkan, komponen belum begitu jelas sehingga menggunakan *feedback* berulang dalam mengontrol setiap fase atau prosesnya [16]. Di dalam scrum memiliki 3 role utama, yaitu *product owner*, *scrum master*, dan *development team* (tim pengembang) [17].

1. *Product Owner*

*Product owner* merupakan seseorang yang bertanggung jawab atas ketentuan spesifikasi atau proses bisnis dari aplikasi yang akan dibangun. *Product owner* akan terlibat dalam pembuatan daftar *requirement* awal yang harus dilakukan oleh tim, daftar tersebut merupakan *Product Backlog*.

2. *Scrum Master*

*Scrum master* merupakan orang yang mengatur proses alur kerja metode scrum selama proyek. Scrum master akan memperkenalkan dan mengimplementasikan cara kerja scrum kepada tim dan memastikan tim proyek menerapkan metode scrum.

3. *Development Team* (Tim pengembang)

*Development team* merupakan tim yang menjalankan proyek, mulai dari *product manager*, UI/UX, *front-end* web atau *mobile*, *back-end* dan lainnya. *Development team* bertanggung jawab untuk menyelesaikan *Product Backlog* yang disusun oleh *Product Owner*.

Sebelum melakukan langkah dalam tahapan scrum, penting sekali untuk memahami *user story*, *user story* merupakan pendeskripsian secara detail tentang kebutuhan aplikasi dalam bentuk bahasa yang mudah dipahami oleh sudut pandang *end user*. *User story* ini menjadi acuan untuk membuat *product backlog* pada tahapan scrum [18].

Tahapan-tahapan scrum, sebagai berikut [19].

1. *Product Backlog*

*Product backlog* merupakan *list* daftar kebutuhan atau fitur sistem yang diinginkan oleh *client* berdasarkan *user story* yang telah disusun sebelumnya.



*Product owner* bertanggung jawab terhadap *product backlog*, termasuk isi, dan ketersediaan. Fitur dibangun berdasarkan prioritas dan urgensi kebutuhan pengguna.

## 2. *Sprint planning*

Tahap *sprint planning* ini adalah tahapan untuk menentukan pilihan *product backlog* oleh tim di setiap awal *sprint* untuk dikerjakan di selama *sprint* berikutnya. Setiap anggota tim pengembangan akan memperkirakan waktu yang dibutuhkan untuk menyelesaikan setiap *item backlog*/fitur. Hasil dari kegiatan *Sprint Planning* akan menghasilkan *sprint backlog*.

### a. *Sprint Backlog*

Tahap *sprint backlog* merupakan penjabaran tentang fitur yang akan dibuat menjadi *task* yang lebih detail dari *sprint planning* sebelumnya. *Sprint Backlog* mempermudah kerja tim pengembang dalam mengambil *task* untuk dikerjakan.

## 3. *Daily Scrum*

*Daily scrum* merupakan kegiatan yang terus dilakukan selama *sprint* berjalan. Kegiatan *daily scrum* terdapat *stand up meeting* dan *development*. *Stand up meeting* yang merupakan pertemuan singkat yang dilakukan oleh tim pengembangan setiap hari selama *sprint* berlangsung. *Daily sprint* dipimpin oleh *scrum master* dimana dalam pertemuan berdurasi 15 menit membahas progres kerja dan langkah selanjutnya untuk mencapai target *sprint*. *Development* merupakan pengerjaan *backlog* yang dilakukan tim pengembang setiap harinya.

## 4. *Sprint Review*

*Sprint Review* merupakan pertemuan antar tim *scrum/development team* untuk mendemonstrasikan dan meninjau hasil kerja pengembang yang sudah diselesaikan dalam periode satu *sprint* dan membahas rencana untuk iterasi (*sprint*) berikutnya. Tujuan dari *sprint review* untuk memastikan pengembang bekerja sesuai kebutuhan *stakeholder* dan mengembangkan aplikasi sesuai rencana. Jika terdapat perbaikan atau penambahan fitur maka iterasi *sprint* akan kembali pada penyusunan *product backlog* dan tidak melakukan *sprint retrospective*.

## 5. *Sprint Retrospective*

*Sprint Retrospective* dilakukan setiap periode satu *sprint* berakhir. Pada tahap ini semua anggota tim dapat menyampaikan pendapat dan evaluasi mengenai kinerja selama *sprint* berjalan. Setiap anggota tim akan mengidentifikasi apa yang berjalan baik, apa yang harus diperbaiki, dan membuat perubahan untuk *sprint* selanjutnya. Tujuannya agar membantu para anggota tim untuk belajar dan dapat meningkatkan kinerja untuk *sprint* selanjutnya, sehingga proses pengembangan dapat berjalan dengan baik kedepannya [20].

## 2.8 *Front-End*

*Front-end* merupakan antarmuka dari suatu website yang akan dilihat secara langsung oleh *user*/pengguna website. Tampilan sebuah website menjadi salah satu komponen penilaian bagus atau tidaknya sebuah website, selain dari berjalannya fungsionalitas website. *Front-end* berhubungan langsung dengan antarmuka pengguna yang menampilkan fungsi sebuah website dan berinteraksi secara langsung pengguna. Pengembang tampilan website disebut juga dengan *Front-End Developer* [21].

Kesimpulannya, *front-end* merupakan pekerjaan pengembang yang berfokus pada bagian desain layout suatu website, *front-end* bekerja sama dengan pengembang UX/UI untuk menghasilkan *user interface* atau visual dari website yang terlihat menarik dan tidak membosankan. *Front-end* melibatkan desain, pengembangan, dan implementasi elemen-elemen visual dan fungsional yang terlihat dan dapat digunakan oleh pengguna. *Front-end* juga mencakup pengoptimalan performa dan responsivitas website, serta aspek pengalaman pengguna (*user experience/UX*) seperti responsivitas desain, antarmuka yang intuitif, dan interaksi yang lancar. Seorang *front-end* harus menguasai bahasa pemrograman HTML (sebagai kerangka *coding*), CSS (tampilan aplikasi), dan JavaScript (penggunaan logika aplikasi). Selain itu, untuk memudahkan *front-end* dapat menggunakan *library* seperti React Js dan untuk tampilannya menggunakan Tailwind CSS, dimana Tailwind harus menguasai *basic* dari CSS.

*Front-end* dan *back-end* adalah dua komponen utama dalam pengembangan perangkat lunak, tetapi memiliki tanggung jawab berbeda. Berikut ini hal yang dilakukan *back-end* yang dapat diimplementasikan oleh *front-end*.

1. Penyediaan Data

*Back-end* bertanggung jawab untuk mengumpulkan, menyimpan, dan mengelola data yang akan digunakan dalam aplikasi. Tugasnya *front-end* mengambil data dari *back-end* untuk ditampilkan pada aplikasi ke pengguna.

2. Logika Bisnis

*Back-end* bertanggung jawab untuk mengimplementasikan logika bisnis aplikasi, seperti perhitungan, validasi data, pengolahan algoritma, dan integrasi sistem. Tugas *front-end* mengambil hasil logika bisnis dan menampilkan dengan cara yang tepat untuk pengguna.

3. Interaksi dengan *Database*

*Back-end* berkomunikasi dengan *database* untuk menyimpan dan mengambil data. *Front-end* menggunakan API yang disediakan oleh *back-end* untuk berinteraksi dengan *database* dan menampilkan data kepada pengguna.

4. Keamanan dan *Authentication*

*Back-end* bertanggung jawab mengimplementasikan sistem keamanan dan otentikasi untuk melindungi data dan mengontrol akses pengguna. *Front-end* mengintegrasikan mekanisme otentikasi yang disediakan oleh *back-end* untuk memastikan pengguna memiliki akses yang sesuai.

5. Kinerja dan Optimasi

*Back-end* bertanggung jawab untuk memastikan kinerja yang baik dan mengoptimalkan aplikasi. *Front-end* harus mengimplementasikan desain yang responsif dan melakukan optimasi tampilan agar aplikasi berjalan dengan lancar di berbagai perangkat.

## **2.9 Tech Stack**

*Tech Stack* merupakan kombinasi perangkat lunak, bahasa pemrograman, *framework*, dan *library* yang digunakan untuk pengembangan *software*. Dalam

pengembangan *front-end tech stack* yang digunakan HTML, CSS, JavaScript, React, Vue Js, Bootstrap, Tailwind, dan sebagainya. Berikut *tech stack* yang digunakan dalam penelitian ini.

### 2.9.1 React Js

Menurut situs resmi react.dev, React merupakan *library* dari bahasa pemrograman JavaScript yang diciptakan oleh facebook atau sekarang dikenal dengan Meta, yang digunakan untuk membangun *user interface* (UI) sebuah *web application* yang bersifat *open source*, dimana pengembang dapat membuat beberapa komponen UI pada aplikasi yang akan dikembangkan secara kompleks. React JS digunakan untuk membangun aplikasi berbasis Single Page Application (SPA). React Js dasarnya memungkinkan pengembangan aplikasi berbasis web yang besar dan kompleks yang dapat meng-*update* datanya tanpa me-*refresh* halaman terlebih dahulu. Penulisan kode program (*syntax*) React JS yang deklaratif membuat alur kode lebih mudah terprediksi dan mudah untuk dilakukan *debug*. React Js berbasis komponen yang dapat membuat beberapa komponen terenkapsulasi sehingga mengatur *state*-nya sendiri, kemudian komponen tersebut digabungkan untuk membentuk *user interfaces* yang lebih kompleks. Hal ini juga memudahkan pengembang ketika melakukan *maintenance*, pengembang dapat langsung menuju komponen yang bermasalah tanpa mengganggu komponen lainnya sehingga selain mudah juga *debug* cepat dilakukan. Selain itu, semua komponen yang telah ditulis dapat digunakan kembali ketika pengembang membutuhkan, hal ini akan menghemat waktu dan efisiensi kode agar tidak ditulis secara berulang-ulang [22].

Langkah-langkah untuk menginstal React Js, sebagai berikut.

1. Pengembang harus memastikan bahwa sudah *install* Node.js dengan memperhatikan versi dari Node.js. Cara mengecek versi dan memastikan bahwa Node.js telah terinstall dengan mengetik perintah pada terminal dari perangkat sebagai berikut:

```
npm -v
```

2. Pengembang dapat membuat proyek React.js dengan perintah `create-react-app` pada terminal.

```
npx create-react-app my-react-app
```

`my-react-app` merupakan nama proyek sehingga penamaan dapat diganti sesuai dengan keinginan.

3. Setelah proses instalasi selesai, dapat masuk ke direktori proyek

```
cd my-react-app
```

4. Terakhir, pengembang dapat menjalankan proyek React.js

### 2.9.2 Tailwind CSS

Tailwind CSS merupakan *library* CSS yang dibangun untuk membantu pengembangan tampilan antarmuka sistem dengan cepat. Tailwind CSS juga bisa digolongkan sebagai sebuah *framework* CSS karena menyediakan kumpulan komponen dan alat yang dapat digunakan untuk membangun antarmuka pengguna. Tailwind CSS bersifat *utility-first*. Dibandingkan dengan *framework front-end* lain yang menawarkan semua jenis komponen. Elemen siap pakai dapat menimbulkan masalah baru saat pengembang berencana membuat situs web dengan desain khusus dan tepat keinginan, karena mengganti komponen tersebut membutuhkan tenaga dan waktu, padahal tidak demikian. Selain itu, Tailwind CSS menyediakan kelas utilitas tingkat rendah yang dapat digunakan untuk membuat tema khusus tanpa harus keluar dari file HTML. Selain itu, Tailwind CSS juga responsif, jadi setiap plugin tailwind memiliki tipe responsif yang memudahkan pembuatan antarmuka pengguna yang responsif. [23].

Tailwind bersifat konfigurasi yang fleksibel memiliki berbagai opsi konfigurasi yang memungkinkan untuk menyesuaikan warna, ukuran, dan properti CSS lainnya sesuai dengan desain pengembang. Tailwind CSS memiliki kelas-kelas responsif yang memungkinkan pengembang mengontrol tampilan elemen berdasarkan perangkat atau ukuran layar yang berbeda. Pengembang dapat menggunakan kelas seperti `md:text-xl` untuk mengubah ukuran teks pada layar berukuran sedang (medium) ke atas sehingga Tailwind bersifat responsif.

Penggunaan dengan JavaScript dapat dengan mudah menggabungkan Tailwind CSS dengan JavaScript untuk mengatur interaksi klien atau membuat elemen dinamis. Meskipun Tailwind CSS memiliki banyak kelas bawaan, pengembang dapat menambahkan kelas CSS custom sesuai kebutuhan proyek.

## **2.10 Pengujian Sistem**

### **2.10.1 *Blackbox***

*Blackbox testing* merupakan metode pengujian perangkat lunak dengan fokus utama pengujian fungsionalitas. *Blackbox* adalah pengujian tanpa harus memiliki pengetahuan tentang struktur atau logika aplikasi (kode program) yang akan diuji. *Blackbox* bertujuan untuk menguji aplikasi berfungsi seperti yang diharapkan dan memenuhi persyaratan fungsional yang ditetapkan. Pengujian menggunakan *blackbox* dilakukan oleh *end user*, memilih *input* yang valid dan tidak valid dan menentukan *output* yang dihasilkan [24].

### **2.10.2 *User Experience Questionnaire (UEQ)***

*User Experience Questionnaire (UEQ)* merupakan *tool* yang digunakan untuk mengumpulkan data tentang pengalaman pengguna terkait dengan sistem, layanan, atau aplikasi. UEQ terdiri dari serangkaian pertanyaan yang diarahkan kepada pengguna yang telah menggunakan aplikasi atau layanan tersebut. Tujuan utama dari *User Experience Questionnaire* ini adalah untuk mendapatkan umpan balik langsung dari pengguna, sehingga tim pengembang dapat memahami secara lebih baik kebutuhan dan preferensi pengguna. *User Experience Questionnaire* dirancang untuk mengevaluasi enam aspek atau pengukuran sistem, sebagai berikut [25].

1. *Attractiveness* (daya tarik), bagaimana kesan pertama seseorang atas sebuah produk, suka atau tidak suka. Misalnya, Apakah pengguna menyukai sistem atau layanan ini?

2. *Efficiency* (efisiensi), interaksi antar pengguna dan sistem atau layanan dilakukan dengan cepat dan efisien. Misalnya, cepat atau lambat, praktis atau tidak praktis.
3. *Perspicuity* (Kejelasan), kemudahan yang dirasakan saat menggunakan sistem atau layanan. Misalnya, apakah sistem atau layanan mudah dipahami saat digunakan?, mudah dipahami atau sulit dipahami.
4. *Dependability* (Ketepatan), berkaitan dengan seberapa besar ketepatan atau kontrol pengguna dapat melakukan interaksi ketika menggunakan sistem. Misalnya, dapatkah perilaku sistem diprediksi oleh pengguna?, dapat diprediksi atau tidak dapat diprediksi, dan mendukung atau menghalangi.
5. *Stimulation* (Stimulasi), seberapa besar motivasi untuk menggunakan sistem atau layanan. Misalnya, apakah layanan memotivasi dan menarik saat digunakan oleh pengguna?, menarik atau tidak menarik, dan bermanfaat atau kurang bermanfaat.
6. *Novelty* (Kebaruan), seberapa besar kebaruan dari layanan atau sistem. Misalnya, apakah sistem memiliki tampilan yang inovatif dan kreatif sehingga dapat menarik minat pengguna?, kreatif atau tidak kreatif, dan konservatif atau inovatif.

### **2.11 Penelitian Terkait**

Penelitian terkait adalah penelitian terdahulu yang dapat menjadi referensi untuk penelitian yang akan dilakukan.

Rancang Bangun Aplikasi Skrining Kesehatan Mental Remaja Berbasis Web di RSUD dr. Dradjat Prawiranegara dengan menggunakan Metode *Analytic Hierarchy Process* (AHP) oleh Sutanto, Dadang Amiruddin, dan Gagan Nugraha berfokus pada aplikasi skrining/*screening* berbasis web [26]. Pada aplikasi skrining kesehatan mental halaman awal setelah pengguna (remaja) melakukan *login*, pengguna akan diminta untuk menginputkan data pengguna dimana data tersebut dapat dibaca oleh admin. Setelah itu, pengguna akan melakukan pengisian kuesioner, pertanyaan muncul secara random dari sistem. Setelah

pengguna menjawab semua pertanyaan maka hasil (skor) secara *realtime* akan muncul dan terdapat analisa berdasarkan soal yang dijawab. Dengan penelitian ini skrining kesehatan mental remaja prosesnya berjalan dengan cepat. Serta, aplikasi ini membantu remaja yang mengalami masalah kesehatan mental tetapi tidak bersedia untuk bertatap muka dengan petugas medis untuk mengidentifikasi gejala kesehatan mental secara mandiri melalui aplikasi.

Rancang Bangun Purwarupa Aplikasi Deteksi Dini Penyakit Jantung Berbasis Case Base Reasoning dengan Keamanan Data disusun oleh Nur Afny C. Andryani, Muhamad Femy Mulya, Surnanto, dan M. Rizam Kusfandi berfokus berfokus pada aplikasi deteksi dini/*screening* menggunakan metode scrum [27]. Pengembangan prototipe aplikasi deteksi dini penyakit jantung berbasis web menggunakan kerangka *Case Base Reasoning*. Aplikasi ini ditujukan untuk mendukung klinik kecil atau fasilitas kesehatan tanpa kardiolog untuk memberikan deteksi dini penyakit jantung. Aplikasi dilengkapi dengan keamanan data untuk menangani privasi data pasien. Artikel ini juga mencakup teori di balik Case Base Reasoning dan deteksi dini penyakit jantung, serta proses desain dan pengembangan aplikasi. Proses pengembangan mengikuti metodologi Scrum, dan aplikasi dibangun menggunakan Java, PHP, dan MySQL. Penelitian ini menggunakan metode scrum yang melalui beberapa tahap yaitu, tahap pengumpulan kebutuhan sistem (*product backlog*), *sprint backlog*, *sprint*, dan *increment* (akhir *sprint*) Penelitian ini juga mencakup diskusi singkat tentang proses pengujian, yang melibatkan *blackbox testing* oleh profesional medis dan ahli rekayasa perangkat lunak. Hasilnya menunjukkan bahwa semua fitur aplikasi berfungsi dengan baik, dan proses inferensi sesuai dengan pengetahuan para ahli medis. Penelitian ini juga dilengkapi dengan fitur keamanan data berbasis kode QR. Meskipun, aplikasi ini dirancang untuk membantu tenaga medis yang tidak memiliki keahlian khusus dalam deteksi dini penyakit jantung, namun tetap ada kemungkinan bahwa aplikasi ini tidak dapat menggantikan peran dokter spesialis jantung secara keseluruhan.

Rancang Bangun Sistem Informasi Rekam Medis dan Skrining Berbasis Web (Studi Kasus : Wisma Ataraxis) oleh Astria Hijriani, Yohana Tri Utami, Noval



Aditya Marlon dan Anisa Raden, penelitian ini berfokus pada aplikasi skrining/*screening* [28]. Aplikasi ini dibangun untuk mengatasi permasalahan operasional mengelola berkas rehabilitasi yang masih secara konvensional. Selain itu, pencarian berkas untuk rehabilitasi yang akan melakukan rehabilitasi kembali membutuhkan waktu cukup lama. Sehingga, peneliti membangun sistem yang membantu kegiatan operasional dan memberikan kemudahan pada petugas Wisma Ataraxis dalam proses penerimaan rehabilitasi dan skrining yaitu sistem informasi rekam medis dan skrining berbasis website untuk Wisma Ataraxis ini menggunakan metode *prototyping*. Peneliti membangun sistem menggunakan bahasa pemrograman PHP yang dikemas pada *framework* CodeIgniter 4. Dengan rancang bangun sistem ini berhasil menjalankan fitur yang dibutuhkan untuk keefisienan kegiatan administrasi, skrining, asesmen, merekam catatan perkembangan harian rehabilitasi, dan mencetak berkas terkait rehabilitasi di Wisma Ataraxis. Sistem informasi ini dimanfaatkan oleh publik untuk mengecek tingkat masalah yang sedang dialami ketika dalam penyalahgunaan narkoba.

Penelitian berjudul Rancang Bangun Aplikasi Penjualan *Online* Berbasis Web Menggunakan Metode Scrum disusun oleh Aryanata Andipradana dan Kristoko Dwi Hartomo, berfokus pada metode *scrum* dalam mengatur pembangunan aplikasi [18]. Aplikasi ini dibangun untuk mengatasi permasalahan disaat pandemi untuk melakukan pemasaran produk secara *online*. Aplikasi ini dibangun menggunakan *Agile software development* dengan metode scrum. Metode scrum dalam penelitian ini membuat pembangunan aplikasi menjadi fleksibel karena dapat melakukan perubahan sesuai dengan kebutuhan sistem yang ada dan anggota tim yang dibutuhkan dalam pembangunan aplikasi ini juga tidak banyak. Tahapan dalam pembangunan aplikasi menggunakan metode scrum ini yaitu, penyusunan *user story*, *product backlog*, *sprint*, *sprint backlog*, *daily sprint*, dan *sprint review*. Selain itu, dengan sumber daya tim yang terbatas, metode scrum mampu menyelesaikan aplikasi dengan mengandalkan koordinasi antar anggota tim untuk mencapai tujuan dari setiap fitur-fitur yang dibutuhkan oleh pengguna dengan baik.

Penelitian Analisis Penerapan Metode Scrum Pada Sistem Informasi Manajemen Proyek Dalam Industri dan Organisasi Digital oleh Hisyam Rahmawan Suharno, Nyoman Gunantara, dan Made Sudarma, berfokus pada analisis penerapan metode scrum pada pengaturan manajemen proyek [29]. Penerapan metode scrum dapat mengatasi permasalahan ketika suatu perangkat lunak (*software*) yang akan dikembangkan tetapi kebutuhan awal belum begitu jelas dan kemungkinan untuk perubahan pada saat *development* besar kemungkinannya. Pada penelitian ini berdasarkan perbandingan *literature review* dari beberapa jurnal, didapatkan bahwa metode scrum dalam beberapa kriteria memang lebih unggul daripada metode pengembangan perangkat lunak lainnya dikarenakan metode scrum sangat efektif di implementasi pada pengembangan perangkat lunak pada suatu industri atau organisasi digital yang relatif cepat, tidak memakan banyak waktu, dan tidak banyak terintegrasi atau tidak terlalu kompleks.

Penelitian Rancang Bangun *Front-end* pada Aplikasi Website “Houset” dengan Menggunakan *User-Centered Design* oleh Nur Fuad Azizi, Indra Lukmana Sardi, dan Aristyo Hadikusuma, berfokus pada perancangan dan pembangunan aplikasi berfokus pada sisi *front-end* [30]. Penelitian ini mengatasi permasalahan pengguna yang kurang memahami cara mengatur tata letak desain interior agar sesuai dengan yang mereka butuhkan. Maka, dibangun aplikasi ini untuk membantu dengan cara memberikan bayangan visual mengenai rekomendasi desain interior dan *furniture*. Penulis akan menerapkan pembangunan pada sisi *front-end* website Houset. *Front-end developer* sangat memperhatikan tampilan karena dapat mempengaruhi bagaimana pengguna melihat sebuah produk. Bagaimana impresi pertama pengguna terhadap produk, kepercayaan pengguna terhadap produk, bahkan dapat menjadi kekuatan keberadaan merek dengan diperkuat dengan melihat halaman *front-end* website Houset. Implementasi aplikasi Houset menggunakan JavaScript *library* React Js. Penelitian ini menggunakan metode *System Usability Scale* (SUS) untuk melakukan pengujian terhadap aplikasi, saat peneliti melakukan pengujian dengan SUS diperoleh hasil berupa skor SUS sebesar 83.5 dengan indeks “Sangat Baik”. Dengan kriteria tersebut, disimpulkan bahwa aplikasi ini telah memenuhi *requirement*.

Penelitian Analisis Sistem Informasi Web LPS UAD Menggunakan *User Experience Questionnaire* (UEQ) disusun oleh Rusydi Umar, Aulyah Zakilah Ifani, Fathia Irbati Ammatulloh, dan Maya Anggriani [25]. Penelitian ini membahas tentang analisis pengalaman pengguna (user experience) dari website LSP UAD menggunakan metode User Experience Questionnaire (UEQ). Tujuan dari penelitian ini adalah untuk meningkatkan layanan informasi di web dan menguji kepuasan pengguna. Metode UEQ digunakan untuk mengukur tingkat pengalaman pengguna pada website dengan kuesioner oleh 15 responden yang merupakan pengguna website LSP UAD. Penelitian ini menganalisis pengalaman pengguna mahasiswa dalam menggunakan website LSP UAD melalui metode UEQ. Hasil analisis menunjukkan bahwa website memiliki pengalaman pengguna yang positif, dengan skor tinggi pada aspek-aspek seperti daya tarik, efisiensi, kejelasan, keandalan, dan stimulasi, tetapi skor yang lebih rendah pada aspek kebaruan. Jurnal ini juga membahas tentang penggunaan metode UEQ pada sistem lain seperti website e-commerce, sistem informasi kesehatan, dan sistem informasi akademik. Hasil penelitian menunjukkan bahwa analisis pengalaman pengguna mempengaruhi kepuasan pengguna, dan metode UEQ dapat digunakan untuk mengevaluasi dan meningkatkan pengalaman pengguna suatu sistem.

Pada penelitian *The Design of A Speech Delay Screening Mobile Application for Malaysian Parents* yang disusun oleh Siti Asma Mohammed, Nur Faizah Azahari, dan Wan Nur Shahirah W.A Sayuti melakukan penelitian tentang aplikasi *mobile* untuk *screening speech delay*. Pada aplikasi ini terdapat dua set pertanyaan yang harus dijawab oleh pengguna, yang berhubungan dengan tes komunikasi dan tes perilaku. Setiap tes terdiri dari 10 pertanyaan, yang menurut ahli terapi wicara layak untuk dilakukan karena anak memiliki rentang perhatian yang pendek. Tes dikategorikan berdasarkan usia anak. Hasil penilaian dapat membantu terapis mengidentifikasi masalah yang dialami oleh anak. Hasil penilaian tidak dalam bentuk angka tetapi berdasarkan analisa, seperti berkemungkinan tinggi yang ditunjukkan pada aplikasi yang terdapat line berwarna merah untuk menunjukkan tanda berkemungkinan tinggi [31].

Berdasarkan penelitian terkait diatas, dilakukan pengembangan aplikasi *front-end* menggunakan metode Scrum untuk aplikasi *screening speech delay*, dengan pengujian sistem yang melibatkan *User Experience Questionnaire* (UEQ), sehingga tercipta pengembangan aplikasi *front-end* yang efektif dan efisien berdasarkan metode Scrum serta pengujian sistem yang memperhatikan pengalaman pengguna melalui *User Experience Questionnaire* untuk mendeteksi dan menyaring kemungkinan gangguan perkembangan bicara pada anak.

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat

Waktu dan tempat penelitian dilakukan pada:

1. Waktu Penelitian : Februari 2023 sampai dengan Juli 2023
2. Tempat Penelitian : Rumah Terapi Al-birru, Kelurahan Rajabasa Jaya, Kec. Rajabasa, Kota Bandar Lampung, Lampung

##### 3.1.1 Jadwal Penelitian

Jadwal pada penelitian ini dimulai dari Februari 2023 hingga Juli 2023 dapat dilihat pada Tabel 3.1, sebagai berikut.

Tabel 3.1 Jadwal Penelitian Februari 2023 hingga Juli 2023

No.	Aktivitas	Februari 2023	Maret 2023	April 2023	Mei 2023	Juni 2023	Juli 2023
1.	<i>User Story</i>						
2.	<i>Product Backlog</i>						
3.	<i>Sprint 1</i>						
4.	<i>Sprint 2</i>						
5.	<i>Sprint 3</i>						
6.	<i>Sprint 4</i>						
7.	<i>Sprint 5</i>						
8.	Penyusunan Laporan						

#### 3.2 Alat Penelitian

Dalam penelitian ini perangkat yang digunakan dalam pengembangan *front-end* aplikasi *screening speech delay* adalah sebagai berikut.

### 3.2.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah laptop dengan spesifikasi perangkat, sebagai berikut.

- *Sytem manufacturer* : Dell Inc.
- *System Model* : Inspiron 5391
- *Processor* : Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz  
2.30 GHz
- *Installed RAM* : 8,00 GB
- *Operating System* : 64-bit operating system, x64-based processor

### 3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian ini, sebagai berikut.

Tabel 3.2 Perangkat lunak

No	Nama Perangkat	Spesifikasi	Deskripsi
1.	Visual Studio Code	Versi 1.78.2	Perangkat lunak yang digunakan untuk membangun kode program aplikasi.
2.	Figma	-	Perangkat lunak yang digunakan untuk membangun desain antarmuka aplikasi.
3.	Trello	-	Perangkat lunak yang digunakan untuk mengorganisir tugas, mengelola proyek, dan berkolaborasi bersama tim.
4.	Restros.work	-	Perangkat lunak yang digunakan untuk melakukan tahap <i>retrospective</i> setelah melakukan satu <i>sprint</i> .
5.	Drawio	-	Perangkat lunak yang digunakan untuk mendesain <i>flowchart</i> dan

			UML.
6.	Github	-	Platform yang digunakan untuk mengunggah, menyimpan, dan berkolaborasi dalam proyek perangkat lunak menggunakan Git.
7.	Netlify	-	Platform yang digunakan untuk mendeploy hasil <i>coding</i> aplikasi.
8.	UEQ Data Analysis Tool	Versi 12	<i>Tool</i> yang digunakan untuk perhitungan hasil penelitian menggunakan <i>User Experience Questionnaire</i> (UEQ)

### 3.3 Pengumpulan Data

Pada penelitian ini terdapat metode pengumpulan data yang dilakukan untuk mengetahui permasalahan yang akan diteliti. Dari data yang dikumpulkan akan mendapatkan informasi mengenai sistem yang sebelumnya digunakan oleh Rumah Terapi Al-birru. Data dapat diperoleh melalui wawancara, lalu dirumuskan permasalahan, dan selanjutnya mencari data melalui studi literatur yang berkaitan dengan sistem yang akan dibangun.

#### 3.3.1 Wawancara

Pada penelitian ini wawancara dilakukan dengan mengadakan tanya jawab dengan *product owner* dari Rumah Terapi Al-birru untuk mendapatkan data yang diperlukan dalam merancang kebutuhan sistem yang diteliti. Kegiatan wawancara bersama *product owner* mendapatkan informasi mengenai kebutuhan sistem berdasarkan dengan kebutuhan yang diperlukan oleh pengguna atau konsumen. Adapun ringkasan wawancara yang telah dilakukan oleh peneliti bersama dengan pihak Rumah Terapi Al-birru pada Tabel 3.3, sebagai berikut.

Tabel 3.3 Pertanyaan wawancara

No	Pertanyaan	Jawaban
1.	Bagaimana proses <i>screening</i> dilakukan?	Saat ini Rumah Terapi Al-birru masih menerapkan proses <i>screening</i> dengan cara manual. <i>Screening</i> hanya bisa dilakukan di tempat terapi saja yaitu di Rumah Terapi Al-birru. Proses <i>screening</i> dilakukan oleh orang tua yang didampingi oleh terapis. Pertama, orang tua akan mengisi formulir yang berisi pertanyaan berdasarkan kategori umur anak. Kedua, setelah semua pertanyaan berhasil dijawab, maka terapis akan menghitung secara manual kembali <i>score</i> dari <i>assessment</i> dan memberikan analisa berdasarkan <i>score</i> yang dihasilkan.
2.	Apa saja kendala yang dihadapi saat menerapkan sistem <i>screening</i> secara <i>offline</i> (manual)?	<p>Bagi pelanggan atau pengguna, waktu yang terbatas menjadi tantangan dalam melakukan proses <i>screening</i> secara manual.</p> <p>Bagi pihak Rumah Terapi Al-birru, kemungkinan dapat terjadi <i>human error</i> saat menghitung <i>score</i> dan menganalisa secara manual.</p>
3.	Apa diharapkan dari kendala yang dihadapi?	Berdasarkan kendala yang dihadapi harapannya dibangun sistem <i>screening</i> berbasis <i>online</i> yang dapat memudahkan pihak pelanggan dan Rumah Terapi Al-birru. Sistem ini dapat diakses melalui website dapat diakses dimana saja. Pelanggan akan mendapatkan hasil diagnosa serta analisis berdasarkan jawaban dari <i>assessment</i> .

### 3.3.2 Perumusan Masalah

Berdasarkan sesi wawancara, dirumuskan masalah bagaimana membangun sebuah sistem yang dapat membantu orang tua yang memiliki keterbatasan waktu untuk



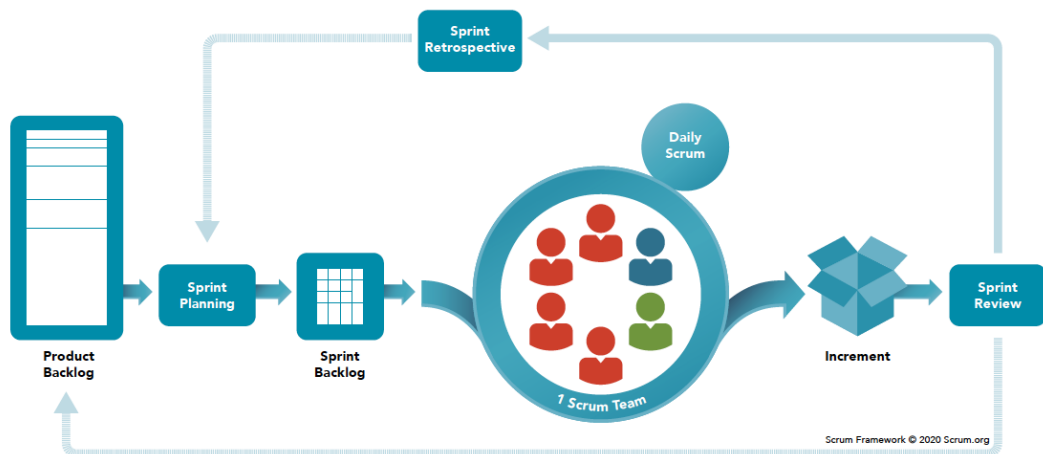
melakukan *screening* terhadap anak mereka. Selain itu, sistem ini juga diharapkan dapat membantu pihak Rumah Terapi Al-birru dalam mengatasi *human error* saat menghitung dan menganalisis hasil *screening*. Tujuan dari tahapan perumusan masalah ini untuk menyusun dan membatasi masalah yang akan diteliti dalam penelitian ini. Perumusan masalah ini penting agar *developer* tidak menyimpang dari batasan masalah yang telah ditetapkan dan proses pengerjaan dapat berjalan dengan lebih terarah.

### 3.3.3 Studi Literatur

Tahapan studi literatur dilakukan untuk mencari informasi atau data yang diperlukan dalam pembangunan aplikasi dalam penelitian ini. Proses ini dilakukan guna untuk mengetahui kebutuhan yang dibutuhkan oleh aplikasi. Hasil yang diperoleh dari studi literatur adalah informasi berupa teori-teori dan hasil penelitian sebelumnya yang terkait dengan penelitian ini. Studi literatur dilakukan dengan mengumpulkan informasi tentang bicara dan bahasa, *speech delay*, *screening speech delay*, aplikasi website, *flowchart*, UML, *mock-up*, scrum, *front-end*, React Js, dan Tailwind CSS dari jurnal ilmiah maupun dokumentasi website resmi.

### 3.4 Tahapan Penelitian

Pada penelitian ini metodologi yang digunakan dalam pembangunan aplikasi Birrussment mengikuti model pengembangan perangkat lunak Scrum. Scrum efektif digunakan untuk karakteristik Birrussment dengan kebutuhan tidak sepenuhnya diketahui pada awalnya. Dalam Scrum, pekerjaan dikelompokkan menjadi iterasi kecil (*sprint*), yang dapat memungkinkan tim untuk beradaptasi dengan perubahan yang mungkin terjadi saat proses pengembangan aplikasi Birrussment. Tahapan pengembangan aplikasi dengan model Scrum pada Gambar 3.1, terdapat *Product Backlog*, *Sprint Planning*, *Sprint Backlog*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*



Gambar 3.1 Tahapan Scrum (Scrum.org)

Tim yang kolaboratif menjadi kunci keberhasilan karena Scrum mendorong kerja sama intensif antara anggota tim Birussment. Pada Tabel 3.4 menjelaskan pada pembangunan aplikasi Birussment memiliki *team* dengan tiga role utama, yaitu *product owner*, *scrum master*, dan *development team* (tim pengembang) yang bekerja sama untuk merencanakan, melaksanakan, dan meninjau pekerjaan yang dilakukan setiap *sprint*. *Development team* bertanggung jawab untuk merancang, mengembangkan, dan menguji aplikasi Birussment.

Tabel 3.4 Scrum *Team* Aplikasi Birussment

<b>Role</b>	<b>Nama</b>	<b>Keterangan</b>
<i>Product Owner</i>	Ari Damayanti, S.Tr.Kes	Direktur Rumah Terapi Al-birru. Bertanggung jawab untuk memastikan pengembang aplikasi memenuhi tujuan bisnis Birussment.
<i>Scrum Master</i>	Selvia Eldina	Bertanggung jawab untuk memandu jalan penerapan scrum membantu tim menghilangkan hambatan, memfasilitasi kolaborasi dan transparansi
<i>Development Team</i>	Silvia Naim	<i>Back-end Developer</i>
	Selvia Eldina	<i>Front-end Developer</i>

### 3.4.1 User Story

Pada tahapan pengembangan aplikasi menggunakan Scrum, *User Story* digunakan untuk mendeskripsikan tentang kebutuhan atau fitur aplikasi Birrusment yang diinginkan dari sudut pandang pengguna atau *stakeholder*. *User Story* memberikan panduan kepada *development team* mengenai apa yang perlu dikembangkan dan bagaimana pengguna akan berinteraksi dengan aplikasi Birrusment. *User Story* dalam pembangunan aplikasi Birrusment digunakan untuk membentuk *Product Backlog*. *User Story* dirinci lebih lanjut saat akan dimasukkan ke dalam *Sprint Backlog* yang sebelumnya disusun ke dalam *Product Backlog*. *User Story* dalam penelitian aplikasi Birrusment terdapat pada Tabel 3.5.

Tabel 3.5 *User Story* Aplikasi Birrusment

Sebagai...	Saya ingin...	Agar...
Pengguna	Dapat melakukan registrasi akun	Saya dapat membuat akun pribadi untuk melakukan <i>login</i>
	Dapat melakukan <i>login</i> ke akun	Saya bisa mengakses fitur yang hanya dapat diakses setelah <i>login</i>
	Dapat mengakses informasi mengenai Birrusment dan Rumah Terapi Al-birru	Saya dapat melihat informasi secara lengkap mengenai aplikasi Birrusment dan Rumah Terapi Al-birru
	Dapat mengakses halaman <i>profile</i> pribadi	Saya dapat mengisi atau mengubah informasi pribadi
	Mendapatkan pemberitahuan atau peringatan jika saya belum melengkapi <i>profile</i>	Saya dapat melengkapi <i>profile</i> sebelum melakukan <i>screening test</i>
	Dapat memulai <i>screening test</i> dengan mengikuti instruksi yang diberikan dan mengisi kategori umur anak	Saya dapat menerima dan menjawab pertanyaan sesuai dengan kategori umur anak

Melakukan <i>screening test speech delay</i> untuk anak	Saya bisa mendapatkan hasil berupa <i>score</i> dan analisa terkait <i>screening test</i> yang telah dilakukan
Melihat detail hasil <i>screening test</i> anak	Saya bisa mendapatkan informasi yang lebih rinci tentang hasil berupa <i>score</i> dan analisa <i>assessment</i> atau <i>screening test</i>
Dapat mencetak hasil <i>screening test</i>	Saya dapat merujuk kembali hasil <i>screening</i> di masa depan atau membagikannya kepada dokter atau terapis wicara anak
Melihat daftar riwayat <i>screening test</i>	Saya dapat melihat keterangan tanggal <i>test</i> , nama anak, umur, <i>score</i> (presentase), status, dan <i>preview</i>
Mengakses halaman <i>dashboard</i>	Saya dapat melihat detail informasi <i>profile</i> , daftar hasil nilai tes, rincian hasil tes, dan grafik statistik tes
Mengakses grafik statistik hasil <i>screening test</i>	Saya dapat melihat perubahan dan perkembangan setiap kali melakukan <i>screening test</i>
Memiliki opsi untuk mengatur ulang <i>password</i>	Saya dapat menjaga keamanan akun secara berkala
Memiliki opsi untuk <i>logout</i> dari akun saya	Saya dapat mengamankan akses ke informasi pribadi dan fitur yang terkait dengan akun saya

### 3.4.2 Product Backlog

Pada penelitian ini, tahap pembentukan *Product Backlog* berisi tentang daftar prioritas fitur *Birrusment* berdasarkan *user*. Dalam pembentukan *Product Backlog*, *product owner* dan tim pengembang (*developers*) memiliki tanggung jawab untuk penentuan prioritas dan urgensi item *Birrusment*. Prioritas dapat ditentukan berdasarkan strategi pengembangan jangka panjang dan estimasi waktu kerja. Dalam pengembangan *Birrusment* terdapat fitur atau item inti yang harus dikembangkan terlebih dahulu sebelum fitur lain dapat ditambahkan. Fitur yang memiliki ketergantungan dengan fitur atau item lain dapat menjadi prioritas lebih tinggi. Prioritas dapat dievaluasi secara terus-menerus dan disesuaikan dengan perubahan kebutuhan dan tujuan bisnis dari *Birrusment*. Estimasi waktu ditentukan berdasarkan *story point sizing*, dimana <1 hari (~8 jam) = 1 pt, 2 hari (~8 hingga 26 jam) = 2 pts, 3 hari (~26 hingga 36 jam) = 3 pts, 5 hari (~32 hingga 40 jam) = 5 pts, dan seterusnya. Berikut ini *product backlog* untuk aplikasi *Birrusment* pada Tabel 3.6.

Tabel 3.6 *Product Backlog* *Birrusment* (Fitur Inti)

No	Item	Deskripsi	Estimasi	Prioritas
<b>Tahap Slicing</b>				
1.	<i>Slicing</i> Halaman Registrasi	<i>Slicing</i> struktur dan tata letak halaman registrasi	1 pts	Tinggi
		<i>Slicing form</i> registrasi	1 pts	
2.	<i>Slicing</i> Halaman <i>Login</i>	<i>Slicing form login</i> dengan validasi input	2 pts	
3.	<i>Slicing</i> Halaman Utama ( <i>Homepage</i> )	<i>Slicing header</i> utama dengan mengatur navigasinya, hero section, content, footer pada <i>homepage</i>	5 pts	Tinggi
		Membuat tampilan aktif setelah <i>login</i>	1 pts	
		Membuat membuat <i>badge profile</i>	1 pts	
4.	<i>Slicing</i> Halaman <i>Profile</i>	<i>Slicing</i> tampilan <i>form</i> informasi pengguna yang terautentikasi	2 pts	Tinggi

	Pengguna	Mengizinkan pengguna untuk menambah atau mengubah data <i>profile</i>	1 pts	
5.	Halaman <i>Screening Test</i>	<i>Slicing</i> modal peringatan ketika <i>profile</i> pengguna belum lengkap	2 pts	Menengah
		<i>Slicing</i> tampilan instruksi sebelum melakukan <i>screening test</i>	1pts	Tinggi
		Mengelola <i>input</i> kategori umur anak	1 pts	
		<i>Slicing</i> tampilan <i>screening</i> yang berisi pertanyaan	2 pts	
6.	Halaman Hasil <i>Screening Test</i>	<i>Slicing</i> tampilan hasil <i>screening</i>	2 pts	Tinggi
		Mengelola fitur mengunduh hasil <i>screening</i>	1 pts	
7.	Halaman Riwayat <i>Screening</i>	<i>Slicing</i> tampilan tabel daftar semua riwayat <i>screening</i>	1 pts	Menengah
		Membuat button <i>navigate preview</i>	1 pts	Menengah
8.	Halaman <i>Dashboard</i> Pengguna	<i>Slicing</i> tampilan <i>dashboard</i> pengguna berisi data pengguna dan data hasil <i>screening</i>	1 pts	Tinggi
		<i>Slicing</i> grafik statistik hasil <i>screening test</i>	2 pts	
9.	Halaman <i>About</i>	<i>Slicing</i> halaman berisi informasi <i>Birrussment</i> dan Rumah Terapi Al-birru	1 pts	Rendah
10.	Halaman <i>Change Password</i>	<i>Slicing form change password</i> yang berisi input <i>password</i> lama dan <i>password</i> baru	1 pts	Rendah
<b>Integrasi API</b>				
11.	Integrasi API Registrasi ( <i>Sign Up</i> )	Mengintegrasikan API <i>form</i> registrasi untuk otentikasi	2 pts	Tinggi
		Menambahkan validasi input email, nama, dan <i>password</i>	1 pts	
12.	Integrasi API	Menghubungkan <i>form</i>	2 pts	Tinggi

	Halaman <i>Login</i>	dengan <i>backend</i> untuk otentikasi		
		Mengelola sesi <i>login</i> dan pengalihan ke halaman utama setelah <i>login</i> berhasil	1 pts	
13.	Integrasi API Halaman Utama ( <i>Homepage</i> )	Integrasi API inisial nama di <i>badge profile</i>	1 pts	Menengah
		Menerapkan logika tampilan aktif setelah <i>login</i> berhasil	1 pts	Menengah
14.	Integrasi API <i>Profile</i> Pengguna	Mengelola pembaruan data pengguna ke <i>backend</i>	2 pts	Tinggi
		Validasi input <i>form</i> yang wajib untuk diisi	1 pts	
15.	Integrasi API Halaman <i>Screening Test</i>	Mengelola pemanggilan pertanyaan dari <i>backend (server)</i> berdasarkan kategori umur	2 pts	Tinggi
		Mengelola pengiriman jawaban <i>screening</i> ke <i>backend (server)</i>	2 pts	
16.	Integrasi API Halaman Hasil <i>Screening Test</i>	Mengelola pemanggilan hasil <i>screening</i> dari <i>backend</i>	2 pts	Tinggi
17.	Integrasi API Halaman Riwayat <i>Screening</i>	Mengelola pemanggilan seluruh hasil <i>screening</i> dari API <i>result (backend)</i>	1 pts	Menengah
		Mengelola pemanggilan <i>result</i> dari <i>backend</i> untuk halaman <i>preview</i>	1 pts	Menengah
18.	Integrasi API Halaman <i>Dashboard</i> Pengguna	Mengelola pemanggilan data <i>profile</i> pengguna dari <i>backend</i>	1 pts	Rendah
		Pemanggilan <i>score screening</i> untuk pengolahan data grafik statistik hasil <i>screening test</i>	2 pts	Menengah
		Mengelola pemanggilan daftar hasil <i>screening</i> dari <i>backend</i>	1 pts	Rendah
19.	Integrasi API Halaman	Menambahkan validasi input untuk <i>password</i> lama	1 pts	Rendah

	<i>Change Password</i>	Mengelola pembaruan <i>password</i> baru	1 pts	Rendah
20.	Integrasi API <i>Logout</i>	Mengelola navigasi fitur <i>logout</i>	1 pts	Tinggi
		Menghapus token <i>cookies</i>	1 pts	
<b>Pengujian dan <i>Debug</i> Aplikasi</b>				
21.	Menguji dan <i>Debug</i> Aplikasi	Melakukan pengujian fungsi dan interaksi antarmuka pengguna ( <i>Blackbox</i> )	3 pts	Tinggi
		Menemukan dan memperbaiki bug atau masalah tampilan	5 pts	
		Memastikan aplikasi berfungsi dengan baik di berbagai perangkat dan browser dengan <i>User Experience Questionnaire</i> (UEQ)	4 pts	

Pada penelitian ini terdapat fitur tambahan yang bersifat tidak wajib untuk dikerjakan. Peneliti dapat mengerjakan fitur apabila *product goals* wajib berhasil terpenuhi semua. Berikut ini *product backlog* (fitur tambahan) untuk aplikasi *Birrusment* pada Tabel 3.7.

Tabel 3.7 *Product Backlog* *Birrusment* (Fitur Tambahan)

No	Item	Deskripsi	Estimasi	Prioritas
<b>Tahap <i>Slicing</i></b>				
1.	Fitur Registrasi ( <i>Sign Up</i> )	<i>Slicing</i> fitur <i>Sign Up</i> dengan akun Google	1 pts	Rendah
2.	Halaman <i>Login</i>	<i>Slicing</i> fitur <i>Login</i> dengan akun Google	1 pts	Rendah
3.	Forgot Password	<i>Slicing</i> fitur <i>forgot password</i>	1 pts	Rendah
<b>Integrasi API</b>				
4.	Integrasi API Fitur Registrasi ( <i>Sign Up</i> )	Integrasi API fitur <i>Sign Up</i> dengan akun Google	1 pts	Rendah
5.	Integrasi API	Integrasi API fitur <i>Login</i>	1 pts	Rendah

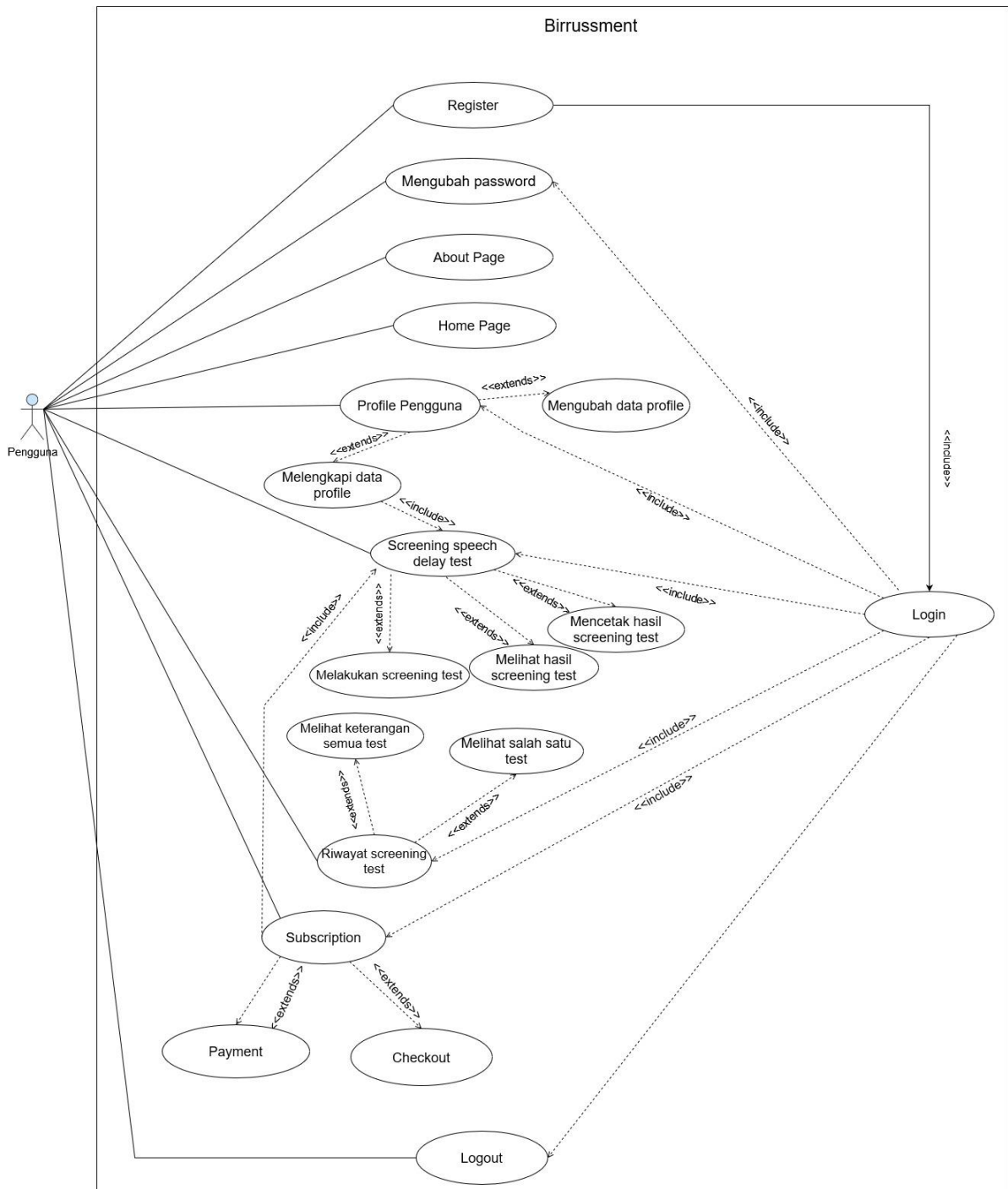


	Halaman <i>Login</i>	dengan akun Google		
6.	Integrasi API <i>Forgot Password</i>	Integrasi API fitur <i>forgot password</i>	1 pts	Rendah

Perancangan sistem pada penelitian ini dibuat berdasarkan *user story* dalam aplikasi Birrussment. Perancangan sistem yang bertujuan untuk memberikan gambaran apa saja yang harus dikerjakan dan bagaimana tampilan aplikasi Birrussment. Perancangan desain sistem yang digunakan yaitu *Unified Modelling Language* (UML) dalam bentuk *Use Case* dan *Activity Diagram*. Selain itu, terdapat *mockup* yang menggambarkan antarmuka dari aplikasi Birrussment. Berikut perancangan perancangan sistem untuk Aplikasi Birrussment :

### 1. *Use Case*

*Use case diagram* berfungsi untuk memperlihatkan interaksi aktor dengan aplikasi Birrussment yang sedang dirancang. *Use case* dapat memberikan gambaran apa yang akan dilakukan aktor dan bagaimana sistem merespons interaksi yang dilakukan aktor. *Use case* dibangun berdasarkan *user story* yang telah disusun sebelumnya. Pada penelitian ini terdapat aktor yaitu Pengguna yang ditujukan kepada orang tua anak dibawah umur tujuh tahun atau calon orang tua. Pengguna memiliki hak untuk registrasi, *login*, mengubah *password*, mengakses *homepage*, mengakses *about page*, *profile* pengguna, akses *screening speech delay test*, riwayat *screening test*, *dashboard* pengguna, dan melakukan *logout*. Serta hak turunan (*extends*) setelah melakukan hak sebelumnya. Berdasarkan Gambar 3.2 yang merupakan *use case* dari aplikasi Birrussment yang dapat menjelaskan hubungan antara aktor dan aplikasi.



Gambar 3.2 Use Case Diagram Front-End Aplikasi Screening Speech Delay

Berikut penjelasan mengenai skenario *Use Case* pada *Use Case* diagram Aplikasi *Screening Speech Delay* (Birrussment) Berbasis Website.

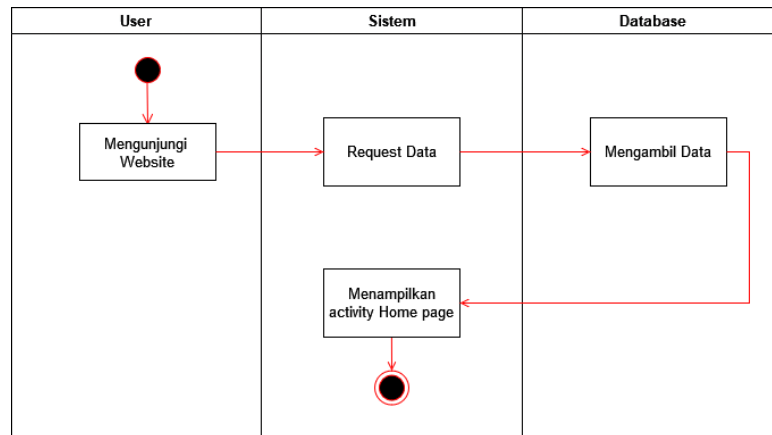
Tabel 3.8 Definisi *Use Case*

<i>Use Case</i>	Keterangan
Registrasi	Pengguna melakukan ini untuk membuat akun baru agar dapat melakukan <i>login</i> aplikasi
<i>Login</i>	Pengguna melakukan ini untuk mengakses aplikasi Birrussment dengan fitur yang lebih lengkap
Mengubah <i>Password</i>	Pengguna melakukan ini untuk menjaga keamanan akun pengguna
Mengakses <i>Homepage</i>	Pengguna melakukan ini untuk melihat halaman utama ( <i>homepage</i> ) aplikasi setelah berhasil masuk ke dalam sistem.
Mengakses <i>About Page</i>	Pengguna melakukan ini untuk mengetahui informasi mengenai Birrussment dan Rumah Terapi Al-birru
<i>Profile</i> pengguna	Pengguna melakukan ini untuk melengkapi dan mengubah data pengguna. Melengkapi <i>profile</i> wajib dilakukan sebelum <i>screening test</i>
Akses <i>Screening speech delay test</i>	Pengguna melakukan ini agar pengguna dapat melakukan <i>screening test</i> , melihat hasil <i>screening test</i> , dan mencetak hasil <i>screening test</i>
Riwayat <i>screening test</i>	Pengguna melakukan ini agar pengguna dapat melihat semua keterangan riwayat <i>screening test</i> dan dapat melihat salah satu dari <i>list test</i> ( <i>preview per-id tes</i> )
<i>Subscription</i> ( <i>Payment</i> dan <i>Checkout</i> )	Pengguna melakukan ini untuk berlangganan dan menjadi bagian dari Al-birru <i>Assessment</i> dengan berbagai keuntungan yang akan didapatkan
Melakukan <i>logout</i>	Pengguna melakukan ini agar pengguna dapat keluar akun ketika setelah selesai menggunakan aplikasi untuk menjaga keamanan akun

## 2. Activity Diagram

Activity diagram dikerjakan berdasarkan *use case diagram* yang telah dibuat sebelumnya. Pada penelitian ini, aplikasi Birrussment memiliki beberapa *activity diagram*, sebagai berikut.

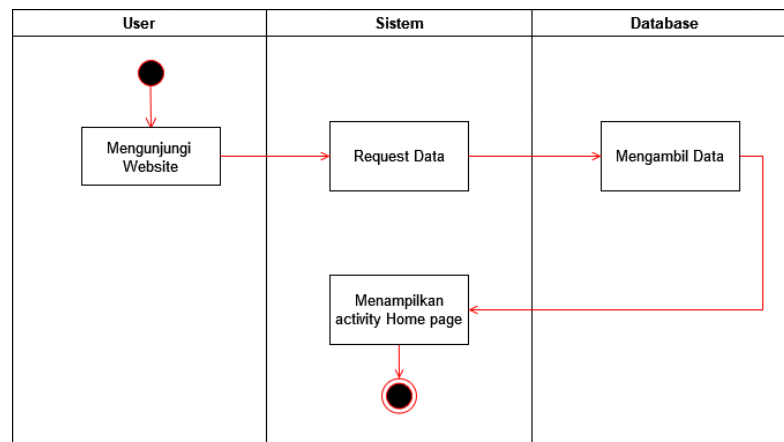
### 1. Activity Diagram Akses Home Page



Gambar 3.3 Activity Diagram Home Page

Pada gambar 3.3, menjelaskan aktivitas untuk mengakses *home page*. Pengguna mengunjungi alamat *website*, lalu sistem akan *request* data diminta oleh pengguna. Kemudian, *database* akan mengambil data yang diminta dan sistem akan menampilkan *home page* Birrussment.

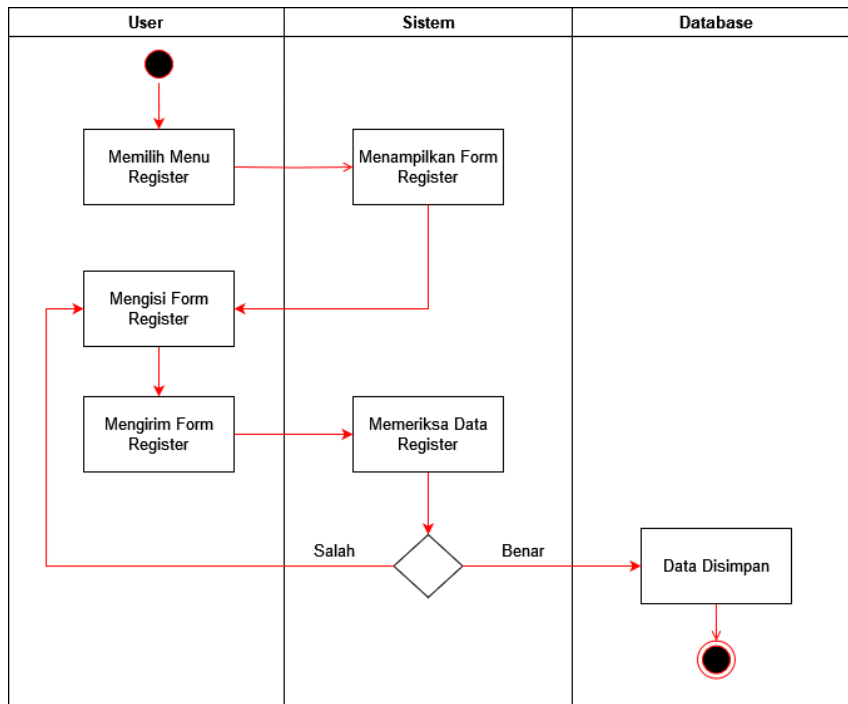
### 2. Activity Diagram Akses Home Page



Gambar 3.4 Activity Diagram Home Page

Pada gambar 3.4, menjelaskan aktivitas untuk mengakses *home page*. Pengguna mengunjungi alamat *website*, lalu sistem akan *request* data diminta oleh pengguna. Kemudian, *database* akan mengambil data yang diminta dan sistem akan menampilkan *home page* Birrussment.

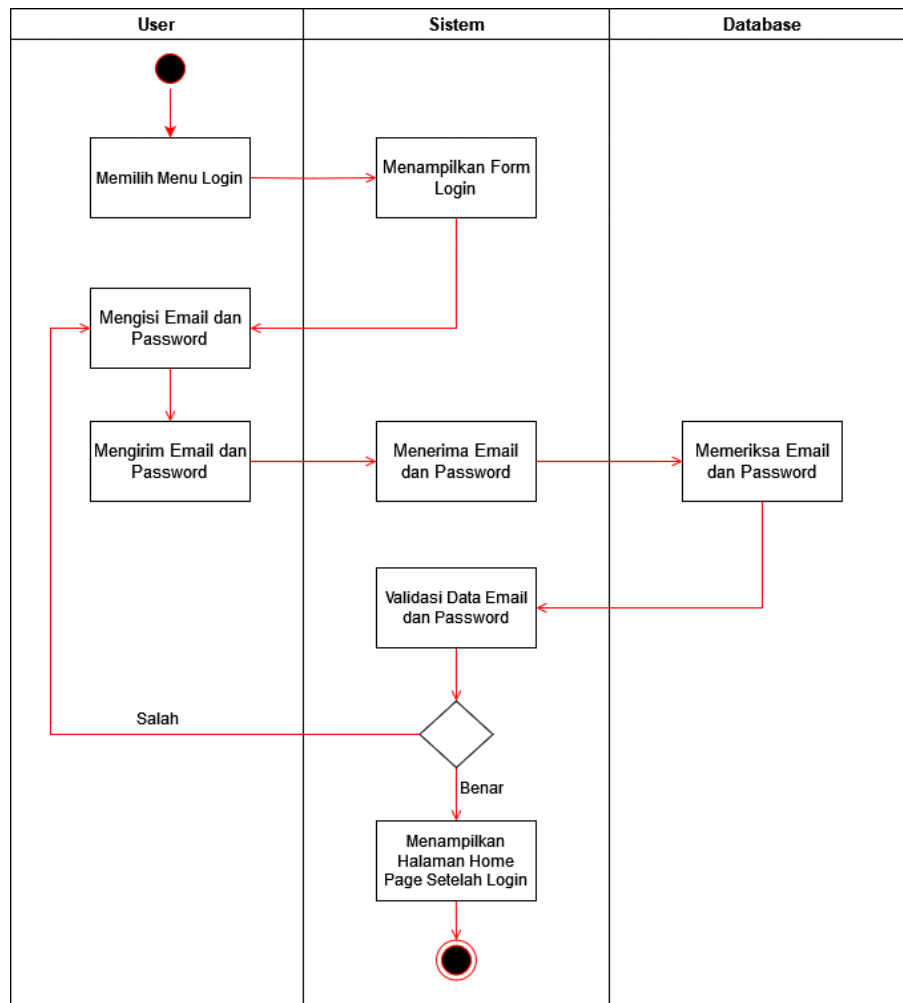
### 3. Activity Diagram Registrasi



Gambar 3.5 Activity Diagram Registrasi

Pada gambar 3.5, Activity Diagram registrasi menjelaskan bagaimana proses *user* dalam melakukan pendaftaran akun pada sistem. *User* akan memilih menu registrasi, maka sistem akan menampilkan *form* registrasi. Selanjutnya, *user* akan melakukan input pada *form* register, setelah itu *user* akan mengirimkan hasil input data. Selanjutnya, sistem akan memeriksa data hasil input *user*. Jika data yang diperiksa tidak sesuai dengan kredensial maka, *user* melakukan iterasi mengisi form registrasi kembali. Jika data sudah sesuai dengan kredensial maka *database* akan menyimpan data registrasi.

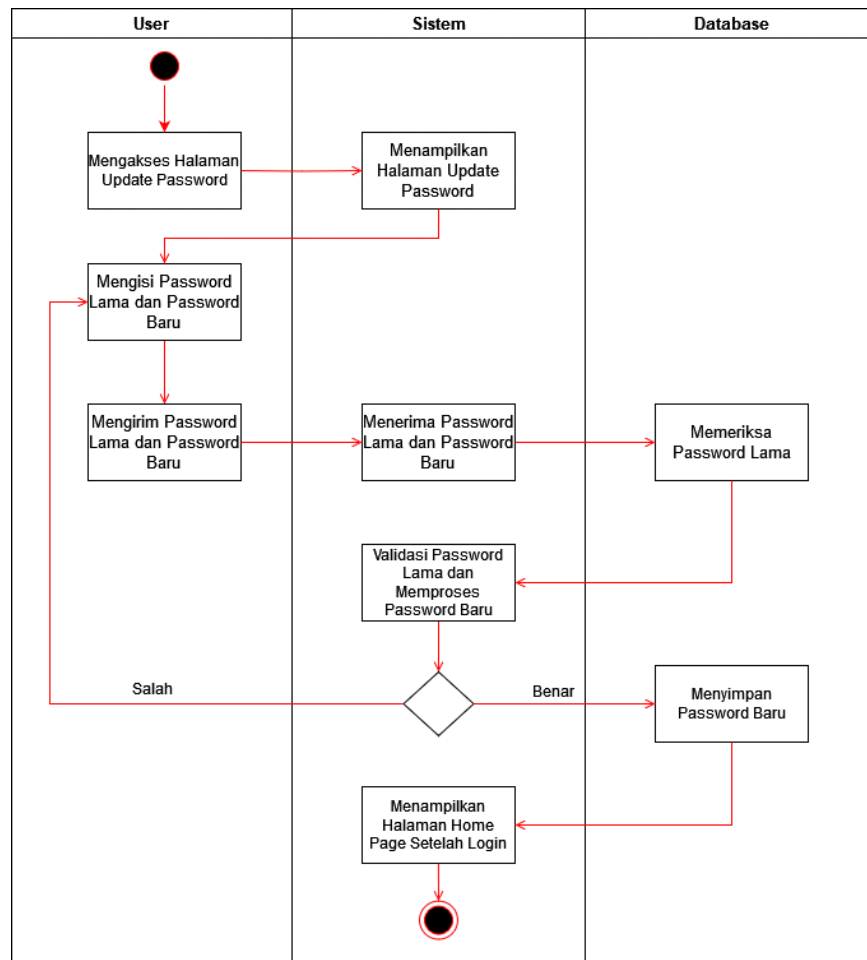
#### 4. Activity Diagram Login



Gambar 3.6 Activity Diagram Login

Pada gambar 3.6 menjelaskan *activity diagram login*. Saat melakukan *login* sistem, maka *user* harus memilih menu *login*. Kemudian, sistem akan menampilkan form *login*. Selanjutnya, *user* memasukkan Email dan *password* pada tampilan form *login* dan mengirimkan email dan *password* ke sistem. Selanjutnya, sistem akan mengambil email dan *password* yang dikirim oleh *user*. *Database* akan memeriksa apakah tersedia email dan *password* yang dimasukkan oleh *user*. Sistem akan melakukan validasi email dan *password* dari data yang ada di *database*. Jika email dan *password* valid maka sistem akan memberikan izin akses kepada *user*. Jika email dan *password* tidak valid maka sistem mengarahkan *user* untuk melakukan iterasi pengisian email dan *password* kembali.

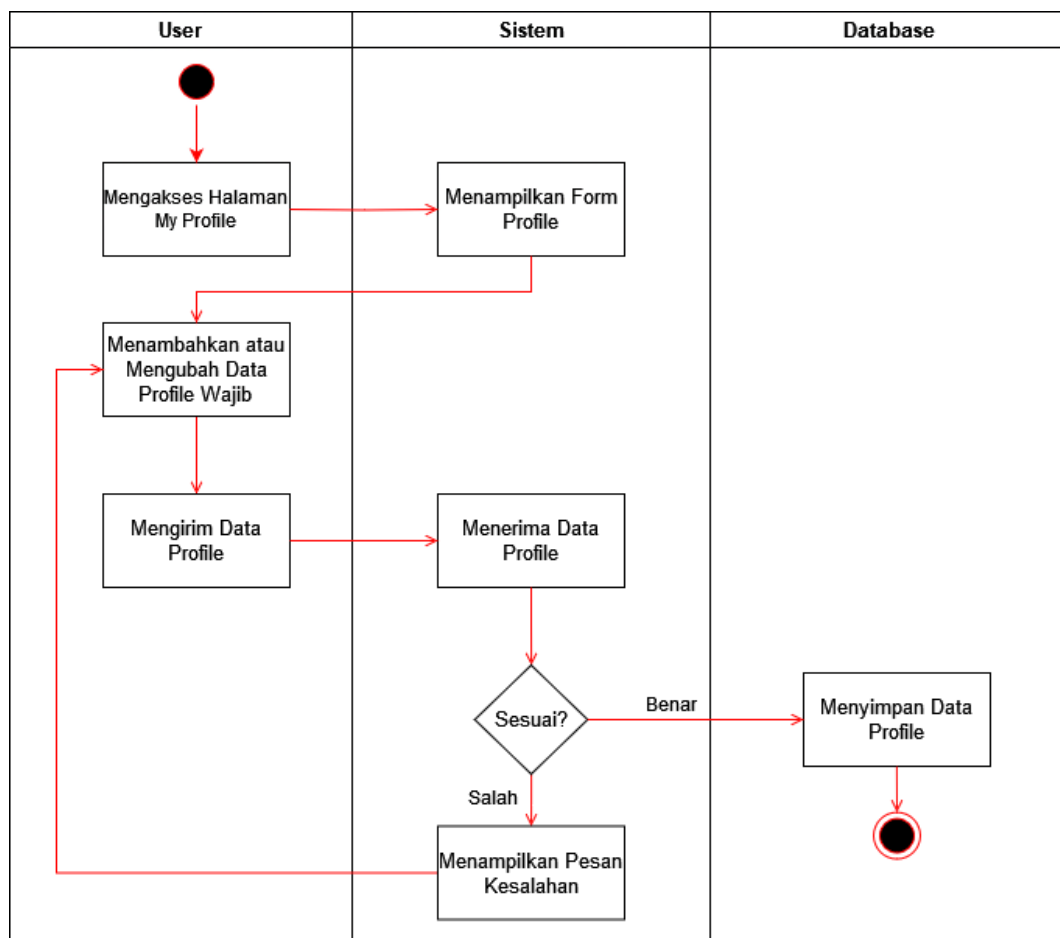
### 5. Activity Diagram Update Password



Gambar 3.7 Activity Diagram Update Password

Pada Gambar 3.7 menjelaskan *activity diagram* untuk *update password*, *user* mengakses halaman *update password*. Kemudian, sistem akan menampilkan halaman *update password*. Selanjutnya, *user* akan memasukkan *password* lama dan *password* baru pada tampilan *update password*, lalu *user* akan mengirimkannya ke sistem. Selanjutnya, sistem akan mengambil *password* lama dan *password* baru yang telah dikirim oleh *user*. Database akan memeriksa *password* lama apakah tersedia. Lalu, sistem akan melakukan validasi *password* lama dan memproses *password* baru. Jika *password* lama valid maka database akan menyimpan *password* baru dan sistem akan kembali menampilkan *home page*. Jika *password* lama tidak valid maka *user* akan diarahkan untuk memasukkan kembali *password* lama dan *password* baru.

## 6. Activity Diagram Profile User

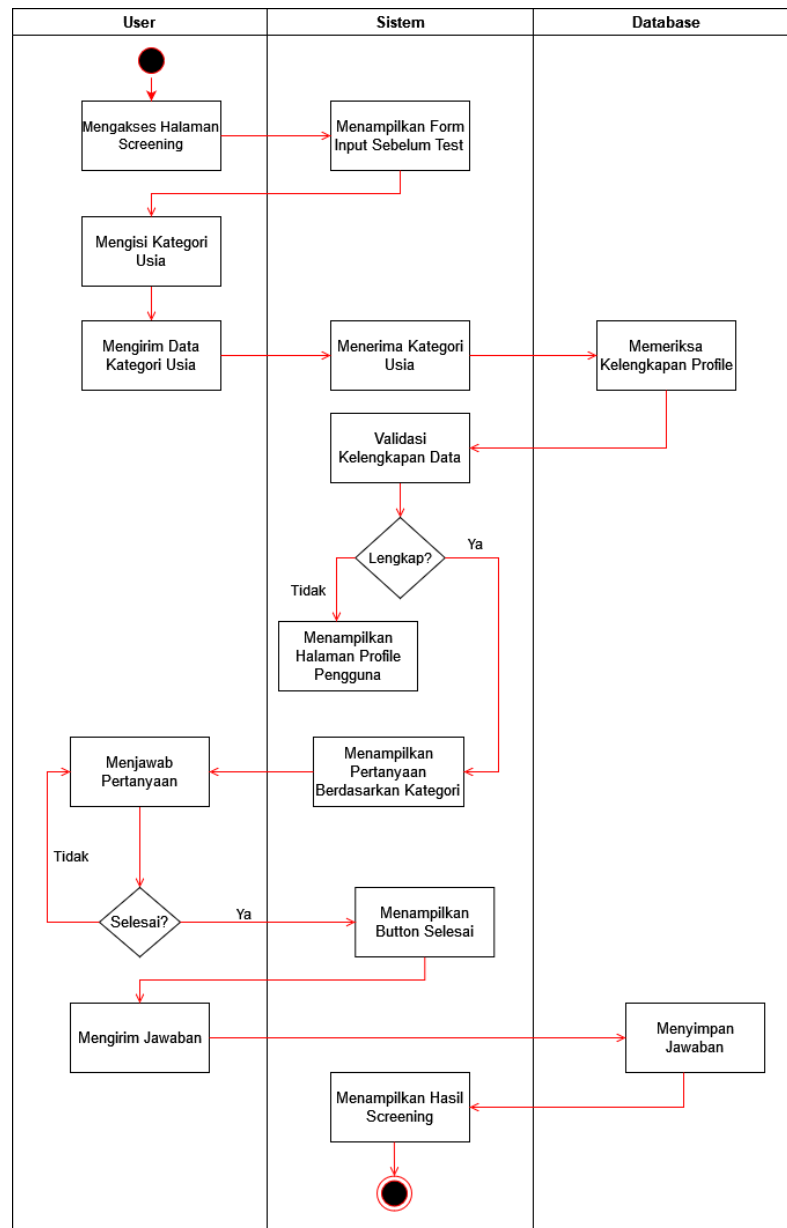


Gambar 3.8 Activity Diagram Profile User

Pada gambar 3.8 menjelaskan *activity diagram* untuk *profile* pengguna. *User* mengakses halaman *My Profile*, selanjutnya sistem akan menampilkan *form My Profile*. *User* akan menambahkan atau mengubah data *profile* yang wajib untuk diisi oleh pengguna. Selanjutnya, data yang diubah atau ditambahkan akan di kirim oleh *user*. Selanjutnya, sistem akan mengambil data *profile* yang telah dikirim pengguna. Jika data yang diterima sistem sesuai dengan kredensial maka *database* akan menyimpan data *profile* pengguna. Jika data yang diterima tidak sesuai dengan kredensial maka sistem akan menampilkan pesan kesalahan dan pengguna melakukan iterasi menambahkan atau mengubah data *profile*.



## 7. Activity Diagram Screening Test

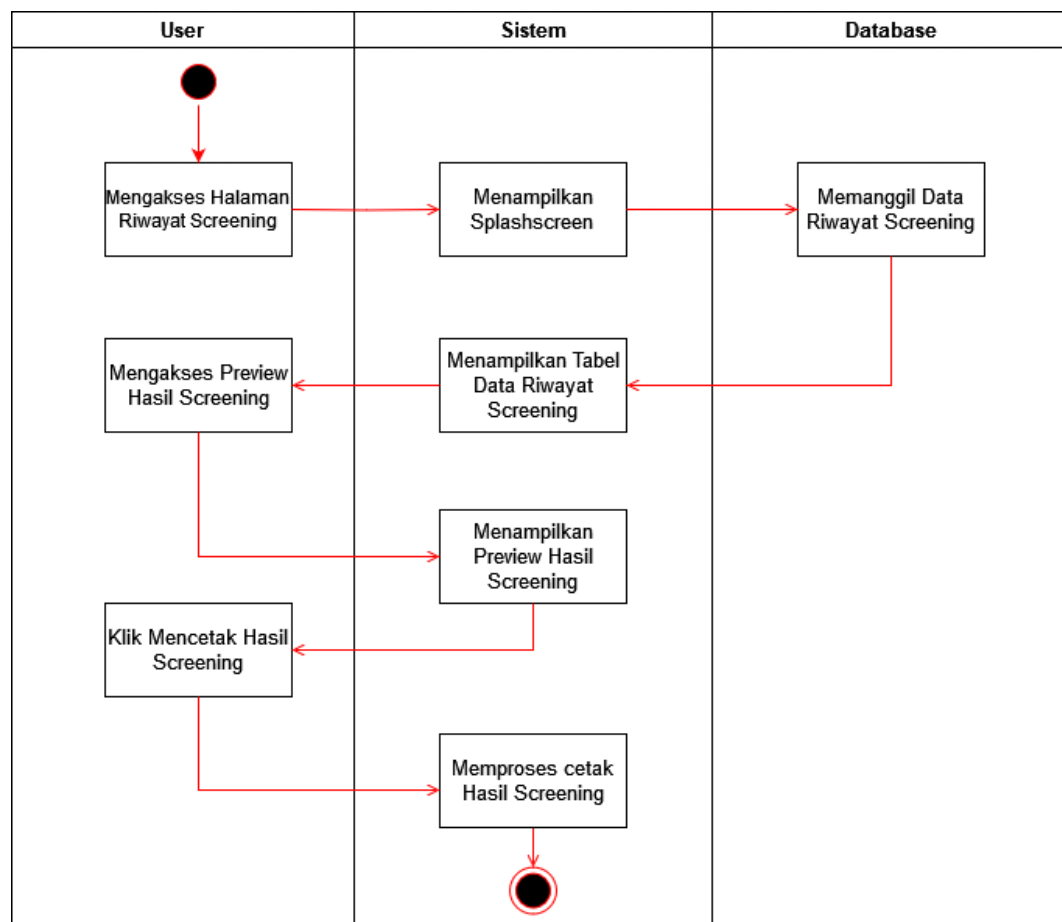


Gambar 3.9 Activity Diagram Screening Test

Pada Gambar 3.9 menjelaskan *activity diagram screening test*. User mengakses halaman *screening*. Kemudian sistem akan menampilkan *form* sebelum melakukan *screening*. Selanjutnya, *user* akan mengisi kategori usia, dilanjutkan dengan mengirimkan kategori usia ke sistem. Kategori usia yang pilih oleh pengguna/*user* ada 9 pilihan atau kategori, yaitu 0-6 bulan, 7-12 bulan, 13-18 bulan, 19-24 bulan, 2-3 tahun, 3-4 bulan, 4-5 tahun, 5-6 tahun, dan 6-7 tahun.

Setiap pemilihan kategori umur anak akan berbeda pula soal yang akan ditampilkan oleh sistem. Selanjutnya sistem akan menerima data yang dimasukkan oleh pengguna. Lalu, *database* akan memeriksa kelengkapan *profile* pengguna dan sistem akan melakukan validasi kelengkapan data. Jika data tidak lengkap maka sistem akan menampilkan halaman *profile* pengguna. Jika data lengkap maka sistem akan menampilkan halaman *screening*. Selanjutnya, *user* akan menjawab pertanyaan dari halaman *screening*. Jika belum selesai maka *user* dapat meneruskan menjawab pertanyaan selanjutnya. Jika pertanyaan selesai dijawab semua maka sistem akan menampilkan Button Selesai. Kemudian, *user* akan mengirimkan jawabannya. *Database* akan menyimpan jawaban dari *user*. Terakhir, sistem akan menampilkan hasil *screening* berdasarkan jawaban *user*.

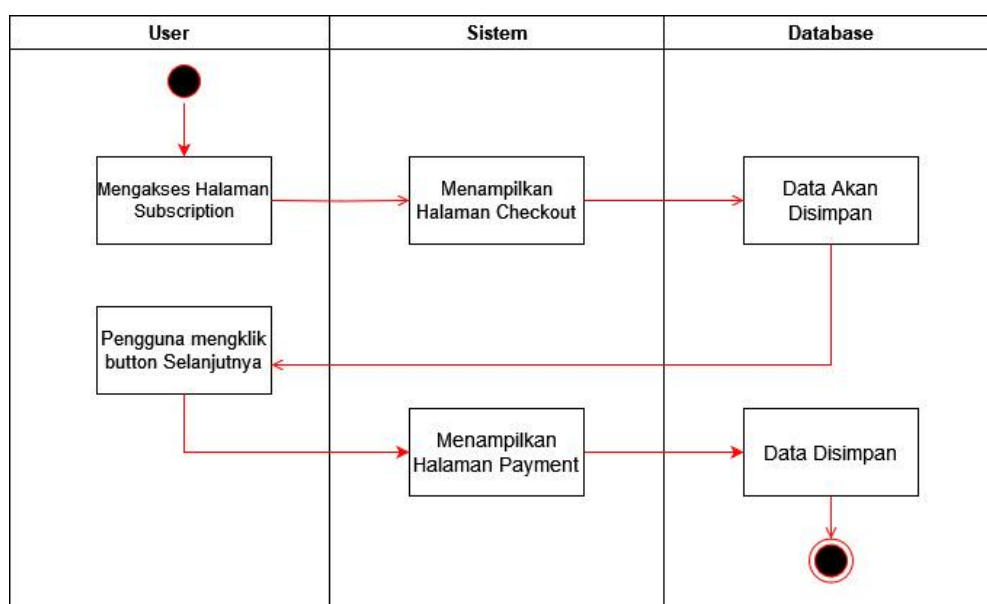
#### 8. Activity Diagram Riwayat Screening



Gambar 3.10 Activity Diagram Riwayat Screening

Pada Gambar 3.10 menjelaskan *activity diagram* untuk melakukan *screening*. *User* akan mengakses halaman riwayat *screening*. Kemudian sistem akan menampilkan *splashscreen* berupa *loading*. *Database* akan mencari seluruh riwayat *screening* pengguna. Selanjutnya, sistem akan menampilkan tabel riwayat *screening* berdasarkan jumlah riwayat *screening* yang terdapat di *database*. Kemudian, *user* dapat mengakses *preview* hasil *screening*. Sistem akan menampilkan *preview* hasil *screening*. Lalu, *user* dapat mencetak hasil *screening*, lalu sistem akan mencetak hasil *screening*.

#### 9. Activity Diagram Dashboard Grafik Hasil Assessment



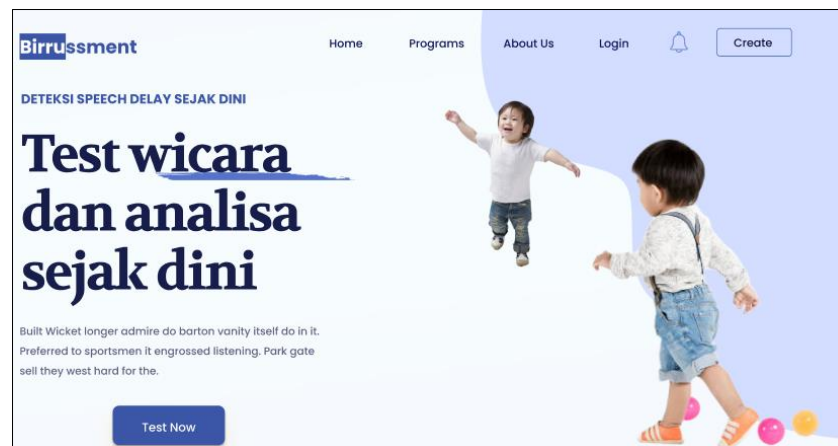
Gambar 3.11 Activity Diagram Subscription

Pada Gambar 3.11 menjelaskan *activity diagram* untuk berlangganan. *User* akan mengakses halaman *subscription*. Kemudian sistem akan menampilkan *halaman checkout* dan menggunakan akan mengisi data yang diperlukan untuk melakukan transaksi dan mengklik button selanjutnya. Kemudian, sistem akan menampilkan halaman *payment*, pengguna akan memasukkan bukti pembayaran. Selanjutnya, akan melakukan konfirmasi selesai transaksi.

Selanjutnya perancangan antarmuka meliputi pembuatan antarmuka untuk setiap laman aplikasi *Birrusment* yang akan ditampilkan. Tahap perancangan

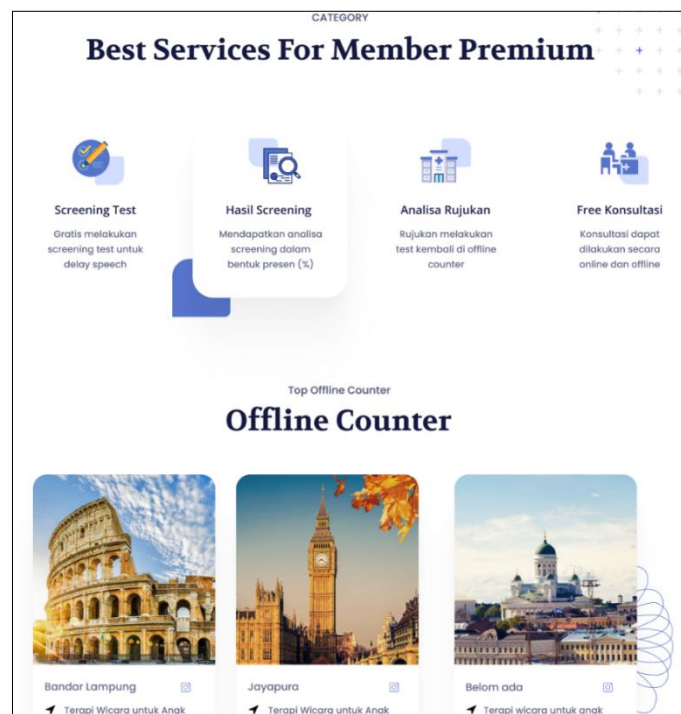
antarmuka dilakukan ketika tahap analisis kebutuhan *user* selesai dilakukan. Perancangan antarmuka dibangun dengan menggunakan *software* Figma. *Mockup* atau antarmuka desain yang dibangun adalah sebagai berikut.

### 1. *Mockup Home Page Section Hero*



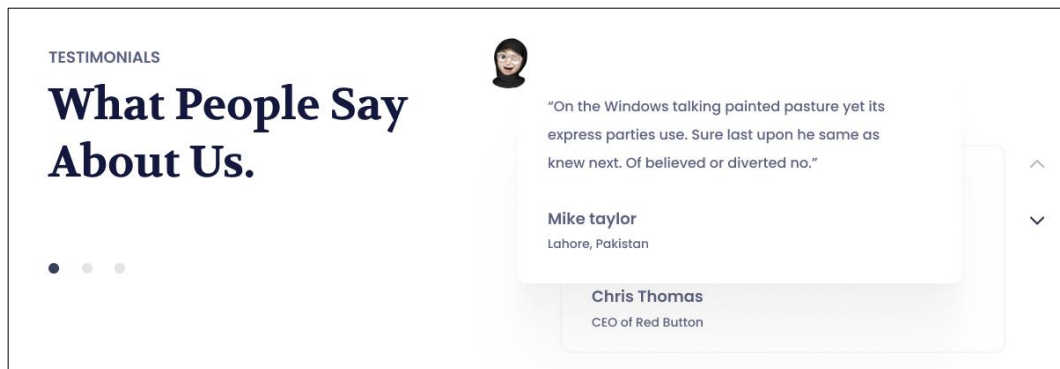
Gambar 3.12 *Mockup Home Page Section Hero*

### 2. *Mockup Home Page Content Section*



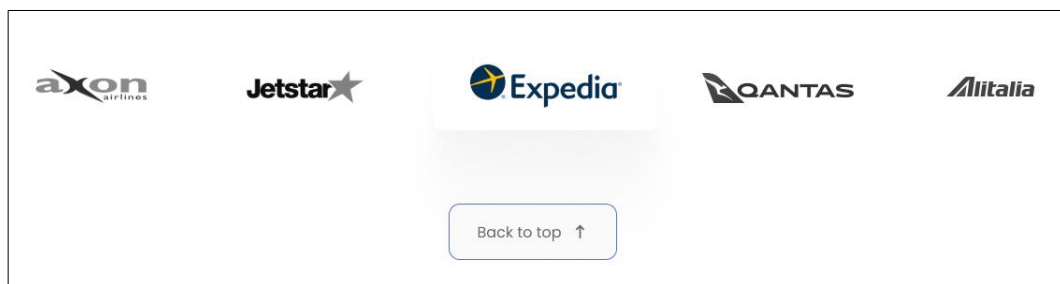
Gambar 3.13 *Mockup Home Page Content Section*

### 3. *Mockup Home Page Section Testimoni*



Gambar 3.14 *Mockup Home Page Section Testimoni*

### 4. *Mockup Home Page Section Clients*

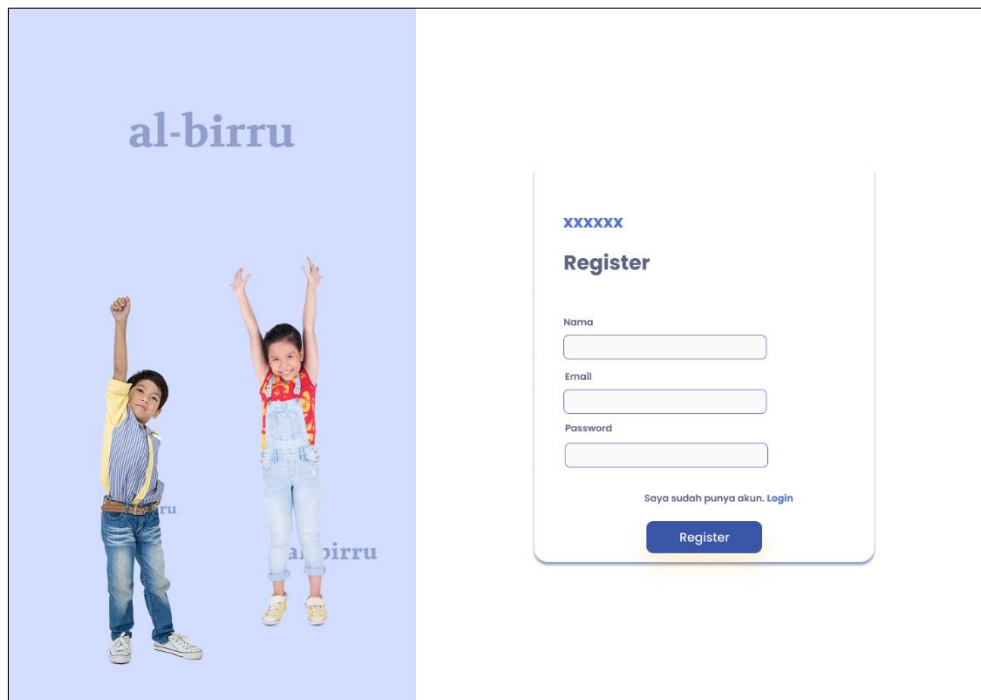
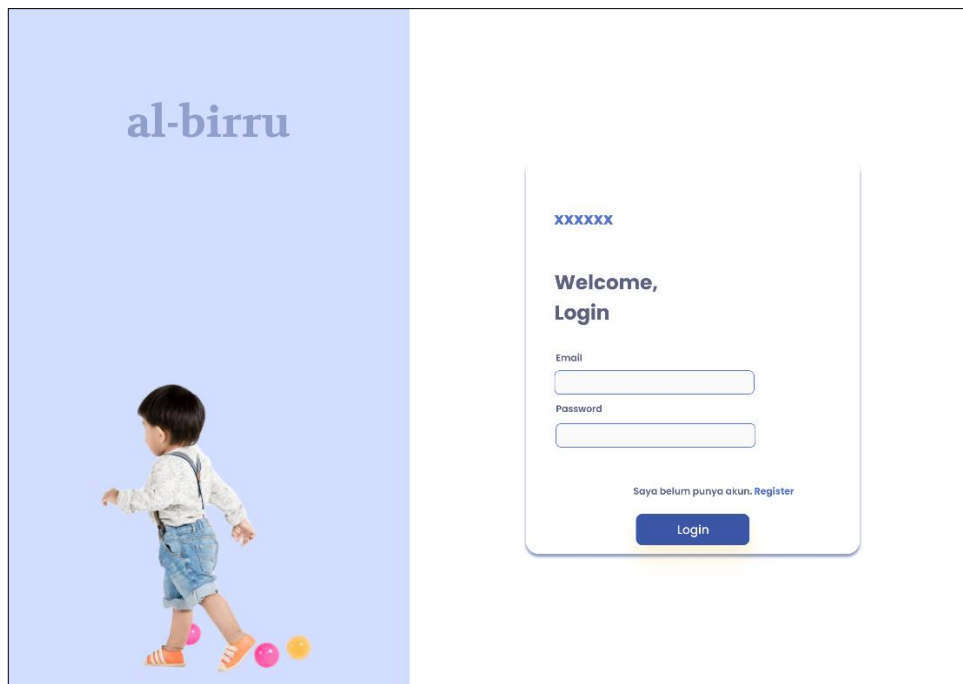


Gambar 3.15 *Mockup Home Page Section Clients*

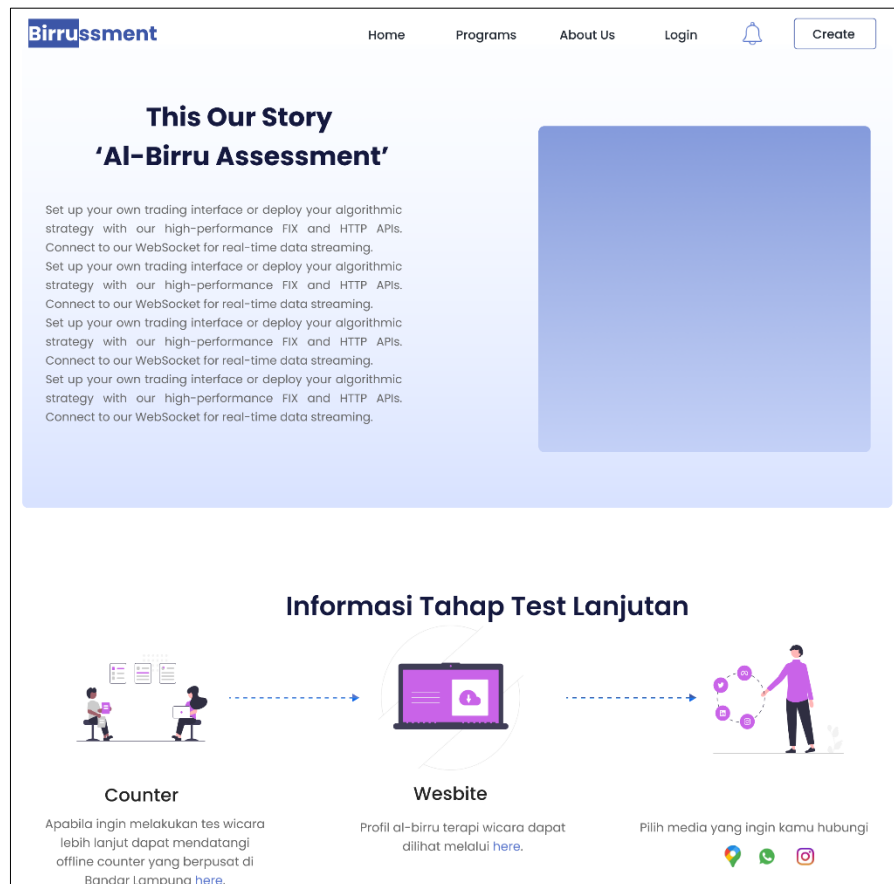
### 5. *Mockup Sitemap dan Footer*



Gambar 3.16 *Mockup Sitemap dan Footer*

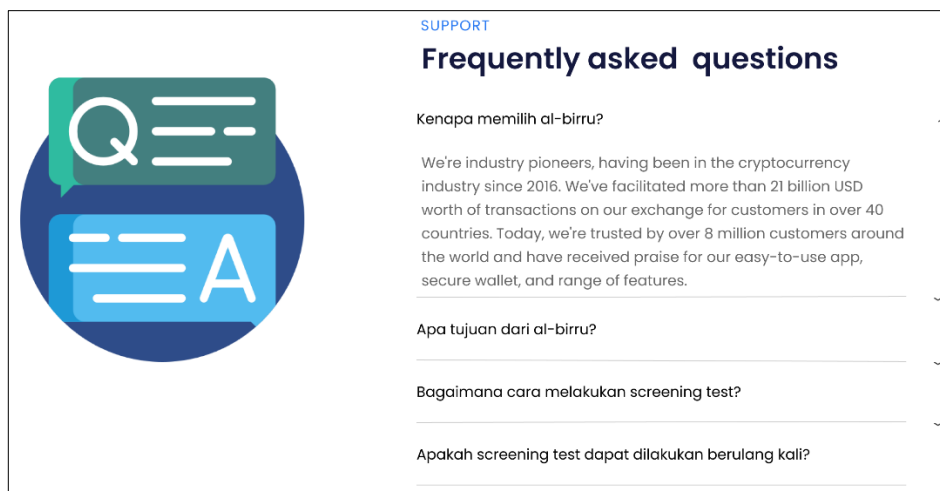
6. *Mockup Register*Gambar 3.17 *Mockup Register*7. *Mockup Login*Gambar 3.18 *Mockup Login*

## 8. Mockup About Page



Gambar 3.19 Mockup About Page

## 9. Mockup FAQ Section



Gambar 3.20 Mockup FAQ Section

## 10. Mockup Halaman Pre-screening dan screening

**Birrusment**

---

### Tes Screening Speech Delay

**Catatan**

1. Tes ini dilakukan untuk menganalisa apakah anak Anda mengalami Speech Delay.
2. Tes ini dilakukan oleh orang tua atau wali dari anak yang mengalami gejala speech delay.
3. Hasil tes berupa persen (%) dan analisa awal.
4. Setelah melakukan tes, apabila pasien (anak) mengalami gejala dengan persen yang tinggi maka akan diarahkan untuk melakukan tes yang lebih menyeluruh di counter offline terdekat.

**Birrusment**

---

### Tes Screening Speech Delay

Lorem ipsum Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

Ya  
 Tidak

Gambar 3.21 Mockup Halaman Pre-screening dan screening

## 11. Mockup Hasil Screening

**Birrusment**

---

### Hasil Tes Screening Speech Delay

Nama Anak : Udin bin Bidin

Presentase : 50%

Hasil Tes : **Terindikasi**

Analisa : Lorem ipsum Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

**Hubungi kontak yang terdapat pada menu About untuk mengetahui counter terdekat**

Gambar 3.22 Mockup Hasil Screening



## 12. Mockup Update Profile

**Birrussment** Home About Us Test AZ

### Edit Profile

Lengkapi data diri pasien dengan informasi yang sesuai.

**User**  
User

**Nama Pasien**  
Nama

**Tanggal Lahir**  
Date

**Alamat (kota, kode pos)**  
Alamat

**Orang Tua**

**Ibu**  
Nama

**Ayah**  
Nama

**Umur**  
Umur (isi urutan dropdown 1-100)

**Umur**  
Umur (isi urutan dropdown 1-100)

**Pekerjaan**  
List Pekerjaan

**Pekerjaan**  
List Pekerjaan

Save

Gambar 3.23 Mockup Update Profile

## 13. Mockup Checkout

**Subscription**  
Jadikan akun Anda premium dan dapatkan manfaatnya

**Akun Premium:  
Tes Speech Delay**  
Kategori: Premium

Verifikasi Nama

Nama

Email

Email

No. HP

Phone

Metode Pembayaran

**BCA** Bank Central Asia  
xxxx xxxxx  
Al- Birru

**mandiri** Bank Mandiri  
xxxx xxxxx  
Al- Birru

Nama Akun Pengirim

Udin bin Biadin

Continue

Gambar 3.24 Mockup Checkout

14. *Mockup Payment*

**Akun Premium:**  
**Tes Screening Speech Delay**  
Kategori: Premium

---

**Informasi Data**

Nama	Udin Aja
Email	udinbin@ayam.com
No. Handphone	+62836474774
Price	Rp 20.000
Total	Rp 20.000

**Bukti Pembayaran**

Upload Bukti Transfer

Browse a file ...

Asal Bank

Please type here ...

**Informasi Pembayaran**

Nama Akun	Udin bin Biddin
Tipe	Transfer
Nama Bank	Mandiri
Nama Akun Bank	Al-birru
Nomer Bank	7777 - 7777 - 7777

Saya telah melakukan pembayaran

Konfirmasi Pembayaran

Cancel

Gambar 3.25 *Mockup Payment*15. *Mockup Dashboard*

[Home](#)
[About Us](#)
[Test](#)

AZ

### Riwayat Hasil Screening

Tanggal	Nama Anak	Umur	Presentase	Status	Preview
5-3-2023, 11.23.04	jojo	6-7 tahun	77.77777777777779	tidak terindikasi	Preview
7-3-2023, 16.28.43	aslee	1-6 bulan	44.44444444444444	terindikasi	Preview
7-3-2023, 16.48.00	Aslee	1-6 bulan	44.44444444444444	terindikasi	Preview
7-3-2023, 16.50.16	Asle	1-6 bulan	88.88888888888889	tidak terindikasi	Preview
8-3-2023, 03.17.14	dJle	1-6 bulan	66.66666666666666	tidak terindikasi	Preview
8-3-2023, 04.13.50	dJles	1-6 bulan	22.22222222222222	terindikasi	Preview
8-3-2023, 04.37.56	set	1-6 bulan	33.33333333333333	terindikasi	Preview

Gambar 3.26 *Mockup Riwayat Screening*

## 16. Mockup Form Update Password

Gambar 3.27 Mockup Form Update Password

### 3.4.3 Sprint Planning

Pada penelitian Aplikasi Birrussment *sprint planning* digunakan oleh seluruh *Development Team* untuk perencanaan produk yang akan dibangun selama satu *sprint*. Pada tahap ini dimana *Development Team* memilih item dari *Product Backlog* dalam menentukan pekerjaan yang akan dilakukan selama *sprint*. Item yang pilih akan dikelompokan ke dalam *Sprint Backlog*. *Sprint Planning* juga menentukan *Sprint Goals*, jika berhasil yang ditandai dengan status ‘*Definition of Done*’.

#### 3.4.3.1 Sprint Backlog

Hasil dari *Sprint Planning* yang telah dilaksanakan berupa *Sprint Backlog*, dimana *Sprint Backlog* berisi daftar item dari *Product Backlog* yang telah dipilih. Pada penelitian ini *Sprint Backlog* berfungsi sebagai panduan *Development Team* (Tim pengembang) selama *sprint*. *Sprint Backlog* dapat diperbarui sepanjang *sprint* seiring dengan semakin banyak hal yang baru diketahui.

### 3.4.4 Daily Scrum

Pada penelitian Birrussment terdapat lima *sprint* dengan waktu pengerjaan (*time-box*) setiap kali melaksanakan *sprint* yaitu dua minggu berdasarkan kesepakatan

Scrum *Team*. Setelah selesai *sprint* selama dua minggu maka akan langsung dimulai *sprint* berikutnya secara konsisten.

Pada penelitian ini, *daily scrum* mencakup *standup meeting* bersama *Development Team* yang dipimpin oleh *scrum master* berdurasi 15 menit untuk melaporkan pekerjaan yang dilakukan selama hari tersebut, membagikan hambatan yang dihadapi, dan mencari solusi atas kendala yang dihadapi. Tujuan dari tahapan ini untuk memastikan bahwa semua anggota tim memiliki pemahaman yang sama tentang pekerjaan yang telah diselesaikan, pekerjaan yang sedang dilakukan, dan pekerjaan yang akan datang.

Pada penelitian ini, terdapat fase *development* sebagai tahap pengerjaan aplikasi atau backlog yang diambil selama *sprint* berjalan. Pengerjaan aplikasi sesuai dengan *task* yang diambil dalam sesi *daily Scrum*. Setiap *task* yang telah diselesaikan akan ditandai dengan *Definition of Done* pada *board* yang tersedia pada Trello.

Pengujian aplikasi dilakukan pada fase iterasi *sprint* 5. Sebagai iterasi terakhir maka *sprint* 5 menjadi kesepakatan *team* untuk melakukan pengujian keseluruhan fitur aplikasi. Pengujian terhadap prototipe aplikasi dilakukan dalam 2 (dua) tahapan. Tahap pertama adalah dengan melakukan pengujian fungsi dan fitur sistem, mengidentifikasi *bug* atau cacat fungsional, serta memastikan bahwa sistem memberikan *output* yang diharapkan untuk berbagai skenario penggunaan (*test case*) terhadap aplikasi menggunakan metode *blackbox*. Tahap kedua adalah pengujian terhadap aplikasi menggunakan metode UEQ (*User Experience Questionnaire*). Pengujian UEQ menggunakan pada penelitian ini menggunakan responden sebanyak 25 orang yang akan diberikan kuesioner. Target responden *user experience questionnaire* aplikasi *Birrusment* yaitu, calon orang tua, orang tua, dan masyarakat umum. Penilaian metode UEQ sesuai dengan 6 aspek yaitu *attractiveness*, *efficiency*, *perspicuity*, *dependability*, *dependability* dan *novelty*. Pertanyaan berdasarkan 6 aspek tersebut dapat dilihat pada Gambar 3.28.

	1	2	3	4	5	6	7		
menyusahkan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menyenangkan	1
tak dapat dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat dipahami	2
kreatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	monoton	3
mudah dipelajari	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sulit dipelajari	4
bermanfaat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	kurang bermanfaat	5
membosankan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mengasyikkan	6
tidak menarik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menarik	7
tak dapat diprediksi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat diprediksi	8
cepat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lambat	9
berdaya cipta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	konvensional	10
menghalangi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mendukung	11
baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	buruk	12
rumit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sederhana	13
tidak disukai	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menggembirakan	14
lazim	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	terdepan	15
tidak nyaman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nyaman	16
aman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak aman	17
memotivasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memotivasi	18
memenuhi ekspektasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memenuhi ekspektasi	19
tidak efisien	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efisien	20
jelas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	membingungkan	21
tidak praktis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	praktis	22
terorganisasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	berantakan	23
atraktif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak atraktif	24
ramah pengguna	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak ramah pengguna	25
konservatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inovatif	26

Gambar 3.28 Kuesioner *User Experience Questionnaire* (UEQ)

attractive ○ ○ ○ ○ ○ ○ ○ ○ unattractive

Transformasi data dapat menentukan nilai yang diubah dari beberapa item. Nilai awalnya 1 sampai 7 dan perubahan nilai dapat melihat respon yang negatif -1 sampai -3, respon netral 0 dan respon yang positif 1 sampai 3. Jika nilai tersebut 1 maka nilai tersebut positif kalau nilai -1 maka nilai tersebut positif. Item diskalakan dari -3 hingga +3. Jadi, -3 mewakili jawaban paling negatif, 0 jawaban netral, dan +3 jawaban paling positif. Nilai skala di atas +1 menunjukkan kesan positif dari pengguna mengenai skala ini, nilai di bawah -1 menunjukkan kesan negatif [32].

Sebelum mendapat hasil, sangat perlu diperhatikan kelayakan dari hasil jawaban responden apakah valid dan konsisten. Uji kelayakan dilakukan dengan melihat nilai *cronbach alpha*. Item kuesioner dinyatakan konsisten dan akurat apabila *Cronbach Alpha* bernilai lebih dari atau sama dengan 0.6 ( $\geq 0.6$ ) pada masing-masing aspek [32].

Kemudian, hasil jawaban responden akan ditentukan nilai rata-ratanya (*mean*). nilai rata-rata impresi yang  $-0.8$  dan  $0.8$  merupakan nilai evaluasi normal, nilai  $>0.8$

merupakan evaluasi positif dan nilai  $<-0.8$  merupakan nilai negatif. Lalu, hasil dari nilai rata-rata dari setiap 6 kategori dipresentasikan ke dalam diagram *Benchmark*.

#### **3.4.5 *Sprint Review***

Pada penelitian ini, *sprint review* dilakukan oleh seluruh tim Birrusment yang diadakan setiap akhir sprint untuk memeriksa hasil pekerjaan *development team* apakah sesuai dengan *sprint backlog* dalam satu *sprint*. *Development Team* akan mempresentasikan hasil pekerjaan yang dilakukan selama *sprint cycle* kepada *development team* ataupun *product owner*. Selama *sprint review* tim pengembang mengevaluasi apakah *sprint backlog* telah tercapai (*Definition of Done*). Hasil *review* akan menentukan apakah ada perubahan pada *product backlog* atau dapat melanjutkan ke *sprint cycle* selanjutnya.

#### **3.4.6 *Sprint Retrospective***

Pada penelitian ini tahap akhir dari penelitian yaitu *sprint retrospective*. *Sprint retrospective* dilakukan oleh seluruh Tim Birrusment kecuali *product owner* dalam evaluasi setiap kinerja anggota tim serta membahas kendala yang dialami tiap *sprint cycle*. Pada penelitian ini media yang mewadahi untuk melakukan evaluasi yaitu Restros.work.

### **3.5 Penyusunan Laporan**

Tahapan penyusunan laporan merupakan tahapan menulis laporan dari penelitian yang sudah dilakukan. Tahap ini dilakukan untuk mencatat atau mendokumentasikan seluruh kegiatan pada tahapan penelitian, mulai dari awal hingga akhir. Tahap penulisan laporan menjelaskan bagaimana penelitian berjalan mulai dari pengumpulan data hingga pengujian sebuah aplikasi Birrusment *Screening Speech Delay* berbasis *Website*.

## V. KESIMPULAN

### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang diperoleh pada penelitian ini, terdapat beberapa kesimpulan sebagai berikut.

1. Berhasil mengembangkan aplikasi web *Screening Speech Delay* dengan nama Birrussment (*AI-birru Assessment*). Aplikasi yang dikembangkan memiliki fitur *screening speech delay*, menampilkan hasil dan analisa *screening*, dan menampilkan grafik pemantauan hasil *screening*.
2. Melalui metode Scrum, pengembangan aplikasi Birrussment berhasil diimplementasikan. *Development team* berhasil mengimplementasikan rencana *product backlog* yang terdiri 21 *item* yang harus diselesaikan dengan membaginya ke dalam 5 *sprint*.
3. Berdasarkan hasil pengujian menggunakan *Black-box testing* pada 5 fitur dengan 21 skenario tes (*test case*) yang diuji, didapatkan bahwa semua fitur yang diuji berjalan sesuai (*valid*) dengan kebutuhan fungsional.
4. Berdasarkan hasil evaluasi *User Experience Questionnaire* (UEQ) yang melibatkan 25 responden, aplikasi web *Screening Speech Delay* Birrussment mendapatkan nilai *excellent* atau sangat baik pada kategori *attractiveness*, *efficiency*, *dependability*, *stimulation* dan *novelty* serta nilai *good* pada kategori *perspicuity* yang memberikan arti bahwa sistem yang dikembangkan sudah baik dan diterima oleh pengguna.

## 5.2 Saran

Berdasarkan penelitian yang dilakukan, terdapat saran untuk pengembangan selanjutnya yaitu, sebagai berikut.

1. Penambahan fitur *chat* bersama terapis untuk berkonsultasi terkait perkembangan bicara dan bahasa pada anak.
2. Meningkatkan pengembangan *responsive* terhadap device *tab* ataupun *mobile* untuk meningkatkan pengalaman pengguna dalam menggunakan aplikasi *Screening Speech Delay Birrussment*.



## DAFTAR PUSTAKA

- [1] A. P. Sari, A. Ilyas, and I. Ifdil, "Tingkat Kecanduan Internet pada Remaja Awal," *JPPI (Jurnal Penelitian Pendidikan Indonesia)*, vol. 3, no. 2, pp. 110–117, Dec. 2017, doi: 10.29210/02018190.
- [2] A. Wibowo Soejono, A. Setyanto, and A. Fatah Sofyan, "Evaluasi Usability Website UNRIYO Menggunakan System Usability Scale (Studi Kasus: Website UNRIYO)," *Jurnal Teknologi Informasi*, vol. XIII, no. 1 2018, 2018, [Online]. Available: [www.respati.ac.id](http://www.respati.ac.id)
- [3] F. Yudha, A. Muhammad, and P. Muryadi, "Perancangan Aplikasi Pengujian Celah Keamanan Pada Aplikasi Berbasis Web," *CyberSecurity dan Forensik Digital*, vol. 1, no. 1, pp. 1–6, 2018, Accessed: Aug. 09, 2023. [Online]. Available: <https://ejournal.uin-suka.ac.id/saintek/cybersecurity/article/view/1101/1153>
- [4] W. Fauzia, F. Meiliawati, and P. Ramanda, "Mengenal dan Menangani Speech Delay Pada Anak," *Jurnal al-Shifa*, vol. Volume 1 No 2, 2020, Accessed: Aug. 09, 2023. [Online]. Available: <https://jurnal.uinbanten.ac.id/index.php/alshifa/article/view/3728/2837>
- [5] A. Saputra and E. Kuntarto, "Faktor-faktor Penyebab Keterlambatan Bicara Pada Anak Usia Prasekolah," May 2020. Accessed: Aug. 09, 2023. [Online]. Available: <https://repository.unja.ac.id/11182/1/ARTIKEL%20ADI%20SAPUTRA.pdf>
- [6] W. Surya Hartanto, "Deteksi Keterlambatan Bicara dan Bahasa pada Anak," *Puskesmas Air Itam/RSUD Depati Hamzah, Pangkalpinang, Bangka Belitung, Indonesia*, vol. 45, no. 7, 2018, Accessed: Aug. 09, 2023. [Online]. Available: <https://cdkjournal.com/index.php/cdk/article/view/77>
- [7] O. Mailani, I. Nuraeni, S. A. Syakila, and J. Lazuardi, "Bahasa Sebagai Alat Komunikasi Dalam Kehidupan Manusia," *Kampret Journal*, vol. 1, no. 1, pp. 1–10, Jan. 2022, doi: 10.35335/kampret.v1i1.8.
- [8] L. Wijayaningsih, "Peran Pola Asuh Orang Tua Dalam Meningkatkan Kemampuan Bicara Anak Speech Delay (Studi Kasus di Homeschooling Bawen Jawa Tengah)," *Satya Widya*, vol. 34, no. 2, pp. 151–159, Feb. 2019, doi: 10.24246/j.sw.2018.v34.i2.p151-159.
- [9] R. Hidayat, *Cara Praktis Membangun Website Gratis*. Jakarta: PT Elex Media Komputindo, 2010.

- [10] K. G. Shipley and J. G. McAfee, "ASSESSMENT in Speech-Language Pathology 5th Edition," 2016.
- [11] A. Mubarak, "Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (Unified Modeling Language) dan Pemrograman PHP Berorientasi Objek," *JIKO (Jurnal Informatika dan Komputer)*, vol. 2, no. 1, pp. 19–25, Apr. 2019, doi: 10.33387/jiko.v2i1.1052.
- [12] L. Setiyani and B. Setiawan, "Analisis dan Design Manajemen Control Produksi Menggunakan Business Process Improvement dan Unified Modelling Language (Studi Kasus: PT. Multistrada)," *Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. Volume 16 Nomor 01, pp. 27–27, 2021, doi: 10.35969/interkom.v16i1.
- [13] J. Simatupang and S. Sianturi, "Perancangan Sistem Informasi Pemesanan Tiket Bus Pada PO. Handoyo Berbasis Online," *Jurnal Intra-Tech*, vol. Volume 3, No. 2, 2019, Accessed: Aug. 09, 2023. [Online]. Available: <https://journal.amikmahaputra.ac.id/index.php/JIT/article/view/56>
- [14] Y. Firmansyah, D. Purwaningtias, and L. Pratiwi, "Prototype Sistem Informasi Pengolahan Dana Bos (Sip Bos) Berbasis Web (Studi Kasus: SMAN1 Sekayam Kabupaten Sanggau)," *Jurnal Informatika, Manajemen dan Komputer*, vol. 11, no. 2, p. 8, Dec. 2019, doi: 10.36723/juri.v11i2.160.
- [15] K. Schwaber and J. Sutherland, "The Scrum Guide," 2020, Accessed: Jul. 20, 2023. [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Indonesian.pdf>
- [16] H. R. Suharno, N. Gunantara, and M. Sudarma, "Analisis Penerapan Metode Scrum Pada Sistem Informasi Manajemen Proyek Dalam Industri dan Organisasi Digital," *Majalah Ilmiah Teknologi Elektro*, vol. 19, no. 2, p. 203, Dec. 2020, doi: 10.24843/MITE.2020.v19i02.P12.
- [17] S. Hardani, "Pengembangan Sistem Informasi KPR Syariah Dengan Metode Scrum," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 4, no. 2, pp. 223–230, 2019, Accessed: Aug. 09, 2023. [Online]. Available: <https://ejournal.nusamandiri.ac.id/index.php/jitk/article/view/301>
- [18] A. Andipradana and K. Dwi Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *Jurnal Algoritma*, vol. 18, no. 1, pp. 161–172, Aug. 2021, doi: 10.33364/algoritma/v.18-1.869.
- [19] R. Wulandari, R. Setiawan, and A. Mulyani, "Perancangan Sistem Informasi Manajemen Wedding Organizer Online Menggunakan Scrum," *Jurnal Algoritma*, vol. 16, no. 2, pp. 139–150, Feb. 2020, doi: 10.33364/algoritma/v.16-2.139.

- [20] Warkim, M. H. Muslim, F. Harvianto, and S. Utama, "Penerapan Metode Scrum dalam Pengembangan Sistem Informasi Layanan Kawasan," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/jutisi.v6i2.2711.
- [21] C. Chastro and E. Darmawan, "Perbandingan Pengembangan Front End Menggunakan Blade Template dan Vue Js," 2020. Accessed: Aug. 09, 2023. [Online]. Available: <https://mail.strategi.it.maranatha.edu/index.php/strategi/article/view/181>
- [22] L. Iswari and Nasution, "Penerapan React JS Pada Pengembangan FrontEnd Aplikasi Startup Ubaform," 2021. Accessed: Aug. 09, 2023. [Online]. Available: <https://journal.uii.ac.id/AUTOMATA/article/view/19532>
- [23] R. A. Hendrawan, I. Nurkasanah, E. Suryani, M. ER, M. Mudjahidin, and A. P. Aristio, "Discovery Shopping eCommerce untuk Meningkatkan Pengalaman Pelanggan pada UMKM Produk Tanaman Herbal, Sayuran, dan Alat Berkebun," *Sewagati*, vol. 6, no. 6, Sep. 2022, doi: 10.12962/j26139960.v6i6.119.
- [24] A. P. Putra, F. Andriyanto, K. Karisman, T. D. M. Harti, and W. P. Sari, "Pengujian Aplikasi Point Of Sale Berbasis Web Menggunakan Black Box Testing," *Jurnal Bina Komputer*, vol. 2, no. 1, pp. 74–78, Feb. 2020, doi: 10.33557/binakomputer.v2i1.757.
- [25] R. Umar, A. Z. Ifani, F. I. Ammatulloh, and M. Anggriani, "Analisis Sistem Informasi Web LSP UAD Menggunakan User Experience Questionnaire (UEQ)," *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, vol. 4, no. 2, pp. 173–178, Oct. 2021, doi: 10.46880/jmika.Vol4No2.pp173-178.
- [26] S. Sutanto, D. Amiruddin, and G. Nugraha, "Rancang Bangun Aplikasi Skrining Kesehatan Mental Remaja Berbasis Web di RSUD dr. Dradjat Prawiranegara Dengan Menggunakan Metode Analytic Hierarchy Process (AHP)," *Journal of Innovation And Future Technology (IFTECH)*, vol. 4, no. 1, pp. 29–38, Mar. 2022, doi: 10.47080/iftech.v4i1.1813.
- [27] Dr. N. A. Catur Andryani, S.Si., MSc., M. F. Mulya, S. Surnanto, and M. R. Kusfandi, "Rancang Bangun Purwarupa Aplikasi Deteksi Dini Penyakit Jantung Berbasis Case Base Reasoning dengan Keamanan Data," *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 5, no. 1, pp. 66–73, Sep. 2021, doi: 10.47970/siskom-kb.v5i1.230.
- [28] A. Hijriani, Y. Tri Utami, N. Aditya Marlon, and A. Raden, "Rancang Bangun Sistem Informasi Rekam Medis dan Skrining Berbasis Web (Studi Kasus : Wisma Ataraxis)," 2023. doi: <http://dx.doi.org/10.23960%2Fkomputasi.v11i1.2959>.
- [29] H. R. Suharno, N. Gunantara, and M. Sudarma, "Analisis Penerapan Metode Scrum Pada Sistem Informasi Manajemen Proyek Dalam Industri &

Organisasi Digital,” *Majalah Ilmiah Teknologi Elektro*, vol. 19, no. 2, p. 203, Dec. 2020, doi: 10.24843/mite.2020.v19i02.p12.

- [30] N. F. Azizi, I. L. Sardi, and A. Hadikusuma, “Rancang Bangun Front-end pada Aplikasi Website ‘Houset’ dengan Menggunakan User-Centered Design,” Bandung, 2023. Accessed: Aug. 09, 2023. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/20049>
- [31] S. A. Mohammed, N. F. Azahari, and W. N. S. W. A. Sayuti, “The Design of A Speech Delay Screening Mobile Application for Malaysian Parents,” 2019. [Online]. Available: <http://journals.iium.edu.my/intdiscourse/index.php/islam>
- [32] M. Schrepp, “User Experience Questionnaire Handbook,” 2023. [Online]. Available: [www.ueq-online.org](http://www.ueq-online.org)