EKSPLORASI PLATFORM PENGEMBANGAN PERANGKAT LUNAK MINIM KODE BUBBLE



Disusun Oleh:

N a m a : Yazid Alif Anugrah
NIM : 19523189

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2024

HALAMAN PENGESAHAN DOSEN PEMBIMBING

EKSPLORASI PLATFORM PENGEMBANGAN PERANGKAT LUNAK MINIM KODE BUBBLE

TUGAS AKHIR

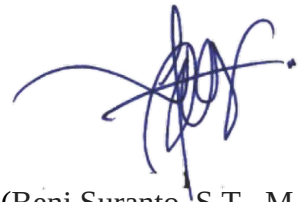
Disusun Oleh:

N a m a : Yazid Alif Anugrah

NIM : 19523189

Yogyakarta,

Pembimbing,



(Beni Suranto, S.T., M.SoftEng.)

HALAMAN PENGESAHAN DOSEN PENGUJI

EKSPLORASI PLATFORM PENGEMBANGAN PERANGKAT LUNAK MINIM KODE BUBBLE

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 17 Juli 2024

Tim Penguji

Beni Suranto, S.T., M.SoftEng. _____

Anggota 1

Izzati Muhimmah, S.T., M.Sc., PH.D. _____

Anggota 2

Erika Ramadhani, S.T., M.Eng. _____

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Yazid Alif Anugrah

NIM : 19523189

Tugas akhir dengan judul:

EKSPLORASI PLATFORM PENGEMBANGAN PERANGKAT LUNAK MINIM KODE BUBBLE

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apa pun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 9 Agustus 2024



(Yazid Alif Anugrah)

HALAMAN PERSEMBAHAN

Alhamdulillahi Rabbil ‘Alamin. Segala puji syukur terpanjatkan kepada Allah SWT, Tuhan semesta alam, Shalawat serta Salam tetap tercurahkan kepada junjungan Nabi Muhammad SAW, semoga kita semua mendapatkan syafaatnya di hari Kiamat kelak, Aamiin.

Skripsi ini saya persembahkan kepada keluarga saya di Samarinda, Kalimantan Timur, yang selalu memberikan dukungan kepada saya sehingga saya bisa berkuliah dan menyelesaikan kuliah di Universitas Islam Indonesia. Skripsi ini juga saya persembahkan kepada teman-teman informatika seperjuangan saya.

Saya juga berterima kasih kepada Bapak Beni Suranto S.T., M.Soft.Eng. sebagai dosen pembimbing penulis karena telah membimbing penelitian ini sampai di titik selesai. Sekali lagi penulis ucapkan terima kasih untuk segala doa, dukungan, serta motivasi yang telah diberikan.

HALAMAN MOTO

مَنْ جَدَّ وَجَدَ

Artinya: "Barang siapa bersungguh-sungguh, maka ia akan berhasil.".

KATA PENGANTAR

Assalamu'alaikum Wr.Wb .

Alhamdulillah Rabbil''alamin, puji dan syukur penulis panjatkan kepada Allah SWT, karena berkat dan rahmat-Nya penulis dapat menyelesaikan tugas akhir dengan judul "**Eksplorasi Platform Pengembangan Perangkat Lunak Minim Kode Bubble**".

Tugas akhir ini disusun sebagai salah satu syarat untuk mencapai gelar sarjana (S1) pada Program Studi Informatika Universitas Islam Indonesia. Selama menyelesaikan tugas akhir banyak hal dan rintangan yang harus dihadapi.

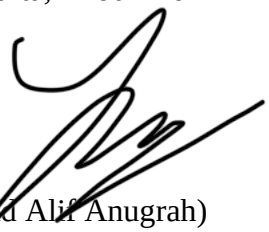
Selama menyelesaikan tugas akhir baik dalam penulisan laporan maupun dalam pengembangan *website* penulis telah banyak menerima bimbingan, pengarahan, petunjuk, saran serta fasilitas yang membantu hingga akhir dari penulisan laporan ini. Oleh karena itu, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dan membimbing selama mengerjakan tugas akhir, khususnya kepada :

1. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Prodi Program Studi Informatika Universitas Islam Indonesia.
2. Bapak Beni Suranto, M.Soft.Eng. selaku dosen pembimbing tugas akhir dan dosen pembimbing akademik yang selalu sabar membimbing saya selama berkuliah di UII.
3. Diri saya sendiri yang telah berjuang selama menyelesaikan tugas akhir.
4. Bapak saya Rusdi Salman dan ibu saya Dessy Ariana yang selalu memberikan dukungan baik materiil maupun moril, serta adik saya Yasmin Rani Meilida.
5. Gita Wahyuni yang telah sabar dan membantu saya selama saya mengerjakan tugas akhir ini dan selama saya berada di Yogyakarta.

Dalam penulisan tugas akhir, penulis juga tidak luput dari kekurangan dan jauh dari kata sempurna. Maka dari itu penulis mengharapkan kritik dan saran yang berarti agar dikemudian hari dapat menjadi lebih baik. Besar harapan penulis dengan adanya tugas akhir ini dapat bermanfaat bagi para pembaca dan dapat menjadi referensi bagi mahasiswa statistika lainnya.

Wassalamualaikum Wr.Wb.

Yogyakarta, 12 Juli 20124



(Yazid Alif Anugrah)

SARI

Pengembangan *software* telah mengalami evolusi yang sangat pesat. Penmgembangan *software* telah dilakukan dengan berbagai metode, seperti metode pengembangan minim pengkodean *(Low-Code Development)*. Tren ini terus menurus mengalami perkembangan yang sangat signifikan seiring dengan perkembangan zaman dan kebutuhan yang semakin cepat dan fleksibilitas dalam pengembangan *software*. *Platform* Bubble.io adalah salah satu *platform* pengembangan perangkat lunak yang berbasis *website* meliputi pengembangan desain UI/UX yang memiliki fitur utama yaitu *Drag and Drop* dan *Design Import* yang sangat membantu desainer dan pengembang perangkat lunak dalam membuat sebuah desain *prototype* UI sebuah *website*. Dalam penelitian ini pengujian *prototype website* yang telah dibuat menggunakan metode *Black box Testing* dan *Usability Testing*. Tujuan penelitian ini adalah diharapakan dapat menghasilkan sebuah *prototype website* sesuai tema dengan memanfaatkan fitur-fitur yang didiberikan oleh *platform* Bubble.io, serta pengujian *prototype* UI/UX yang dibuat menggunakan *platform* tersebut. Harapan hasil penelitian ini bahwa *Platfom* ini memudahkan para pengembang pemula untuk membuat website dengan mudah tanpa harus memiliki pengetahuan dasar tentang bahasa dan aturan dalam pengkodean.

Kata kunci: Perangkat Lunak Minim Kode, Bubble.io, *Prototype*, Redesain Steam.

GLOSARIUM

Glosarium memuat daftar kata tertentu yang digunakan dalam laporan dan membutuhkan penjelasan, misalnya kata serapan yang belum lazim digunakan. Urutkan sesuai abjad. Contoh penulisannya seperti di bawah ini:

Compile	proses untuk mengubah berkas kode program dengan berkas lain yang terkait menjadi berkas yang siap untuk dieksekusi oleh sistem operasi secara langsung.
Debug	langkah untuk menelusuri kesalahan kode program.
Deployment	Peluncuran
Eksplorasi	Proses penyelidikan dan kegiatan mencari pengalaman baru dari situasi baru
Low-code Development	Pengembangan minim pengkodean; pengambangan suatu aplikasi dengan minim pengkodean
Prorotype	Purwarupa suatu produk untuk diuji sebelum dirilis ke publik
Section	Bagian dari sesuatu dari yang utuh
Software	Perangkat lunak dari sebuah perangkat
Testing	Pengujian atau percobaan sebuah produk untuk mencari kesalahan atau kelemahan yang selanjutnya akan diperbaiki
UI/UX	Antarmuka pengguna dan Pengalaman pengguna
Usability	Kegunaan, bagaimana sebuah produk dapat berguna untuk publik, kualitas yang menilai seberapa mudah sebuah antarmuka pengguna digunakan
Website	Situs web, program komputer yang menjalankan peladen yang menyediakan akses kepada beberapa laman
Waterfall	metode pengembangan perangkat lunak.
Wireframe	Gambar kerangka, representasi visual dasar dari sebuah aplikasi yang akan dibuat, baik antarmuka (UI) atau tata letak.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI viii
GLOSARIUM ix
DAFTAR ISI x
DAFTAR GAMBAR xiii
BAB I PENDAHULUAN 15
1.1 Latar Belakang 15
1.2 Rumusan Masalah 16
1.3 Batasan Masalah 16
1.4 Tujuan Penelitian 16
1.5 Manfaat Penelitian 16
1.6 Metodologi Penelitian Secara Umum 17
1.7 Struktur Laporan 17
BAB II LANDASAN TEORI 19
2.1 Metode Pengembangan Minim Kode 19
2.2 Platform Bubble 21
2.3 Metode Prototyping 24
2.4 Website 27
2.4.1 Pengertian Website 27
2.4.2 Indikator Website 27
2.5 Tahapan Pengembangan *Website* 31
2.6 Penelitian Terdahulu 34
BAB III METODOLOGI 38
3.1 Alur Pengembangan 38
3.2 Studi Literatur 40
3.3 Pengumpulan Data Game 41
3.4 Eksplorasi Platform Bubble 43
3.5 Pengembangan Prototype 43
3.6 Metode Pengujian 43
BAB IV HASIL DAN PEMBAHASAN 47
4.1 Hasil Kajian Pustaka 47
4.2 Hasil Rancangan 48
4.2.1 Rancangan Tampilan Home 48
4.2.2 Rancangan Tampilan Store 49
4.2.3 Rancangan Tampilan Library 50
4.3 Hasil Ekplorasi 51
4.3.1 Tampilan Halaman Home 57
4.3.2 Tampilan Halaman Store 57
4.3.3 Tampilan Halaman Browse 58
4.3.4 Tampilan Halaman Library 59
4.3.5 Tampilan Halaman Download 60

4.4 Hasil Pengujian 61
4.5 Refleksi Diri Terhadap Platform Bubble 62
BAB V KESIMPULAN DAN SARAN 64
5.1 Kesimpulan 64
5.2 Saran 65
DAFTAR PUSTAKA 66
LAMPIRAN 68

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu 34
Tabel 3.1 List Pernyataan dalam USE *Quesionnaire* 44
Tabel 3.2 Tabel Nilai untuk Pernyataan dalam USE *Quesionnaire* 45
Tabel 4.1 Fitur dalam platform Bubble 53
Tabel 4.2 Hasil Pengujian *Black box Testing* 61
Tabel 4.3 Skor Pengujian *Usability Testing* 62

DAFTAR GAMBAR

Gambar 2.1 Contoh *progamming* tradisional kode index.html Sumber: (rajgor & Marly, 2022) 20
Gambar 2.2 Contoh *progamming* tradisional kode style css.html Sumber: (rajgor & Marly, 2022) 20
Gambar 2.3 Contoh *platform Low-Code* Sumber: (rajgor & Marly, 2022).....21
Gambar 2.4 Contoh *output* dari *programming* tradsional dan *Low-Code Platform* Sumber: (rajgor & Marly, 2022) 21
Gambar 2.5 Bubble Sumber : Bubble.io 21
Gambar 2.6 Contoh Platform *Low-Code Bubble* 22
Gambar 2.7 Alur Tahapan Metode *Prototyping* Sumber : (Muhtadi & Zuanedi, 2021) 26
Gambar 2.8 Tiga Kriteria Evaluasi *Web* Sumber: (Asriani, 2011) 30
Gambar 2.9 Contoh Tahap Design Dengan Figma 32
Gambar 2.10 Contoh Kode Pengembangan *Website* 33
Gambar 2.11 Contoh Hosting *Website*.....34
Gambar 3.1 Alur Pengembangan 38
Gambar 3.2 Platform Steam.....42
Gambar 3.3 Platform Steamdb Data dan Rating Game 42
Gambar 3.4 Komunitas Game di Discord.....42
Gambar 4.1 Hasil Rancangan Tampilan *Home*.....49
Gambar 4.2 Hasil Rancangan Tampilan *Store*.....50
Gambar 4.3 Hasil Rancangan Tampilan *Library*.....51
Gambar 4.4 Tampilan Utama *Bubble.io* 51
Gambar 4.5 *Element Tree* 53
Gambar 4.6 *Visual Element* 53
Gambar 4.7 *Workflow* 54
Gambar 4.8 *Design Import* 55
Gambar 4.9 *Styles* 55
Gambar 4.10 *Templates* 56
Gambar 4.11 *Plugin* 56
Gambar 4.12 Hasil dari *prototype* tampilan *home* 57
Gambar 4.13 Hasil dari *prototype* tampilan *store*.....58
Gambar 4.14 Hasil dari *prototype* tampilan *Browse*.....59

Gambar 4.15 Hasil dari *prototype* tampilan *Library* 60
Gambar 4.16 Hasil dari *prototype* tampilan *Download* 60

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada era digital saat ini, kebutuhan akan pengembangan perangkat lunak semakin meningkat. Namun, mengembangkan perangkat lunak yang rumit dan kompleks seringkali membutuhkan keahlian pemrograman tingkat lanjut. Hal tersebut membuat pengembangan perangkat lunak menjadi sulit bagi mereka, akibat tidak memiliki keterampilan pemrograman yang memadai. Pada umumnya, pengembangan sistem atau aplikasi dilakukan melalui serangkaian proses pemograman atau dengan menulis baris-baris kode program yang sering dikenal dengan istilah *Coding*. Namun dalam perkembangannya saat ini, telah hadir istilah *Low-Code* sebagai alternatif pengembangan sistem atau aplikasi dengan meminimalkan proses *Coding*, sehingga memungkinkan perangkat lunak dibuat tanpa keterampilan *pemrograman* tingkat lanjut.

Tools atau alat-alat yang digunakan dalam metode *low-code* mempunyai peran yang sangat penting dalam mendukung pengembangan system atau aplikasi dengan menggunakan metode *Low-Code.* Saat ini sudah hadir banyak alat bantu yang dapat digunakan dalam pengembangan sistem atau aplikasi melalui metode *Low-Code*, baik pada *platform web* maupun *mobile.* Beberapa yang populer masuk dalam kategori *Content Management System* seperti Wordpress, Drupal, Joomla, dan lain-lain. Beberapa lainnya baru mulai bermunculan dalam tahun-tahun dimulainya revolusi industry 5.0 seperti glideapps, Bubble, webflow, dan lain-lain.

Salah satu *platform* pengembangan perangkat lunak *Low-Code* yang populer saat ini adalah *Bubble*. *Bubble* membuat pengembangan perangkat lunak menjadi lebih mudah dan memungkinkan orang yang tidak memiliki pengalaman atau pengetahuan di bidang pengembangan perangkat lunak untuk membuat program dalam bentuk aplikasi seluler, situs *web*, ataupun aplikasi lainnya. Akan tetapi masih terdapat keterbatasan dalam pengunaan *platform Bubble*. Keterbatasan tersebut diantaranya meliputi pembatasan penggunaan *database*, pembatasan integrasi dengan *platform* lain, dan pembatasan penggunaan bahasa pemrograman yang digunakan.

Oleh karena itu, diperlukan penelitian mendalam mengenai *platform* pengembangan *software Low-Code Bubble*, dengan tujuan untuk mengeksplorasi dan mengidentifikasi keterbatasan yang ada dalam penggunaannya serta mencari solusi untuk mengatasi keterbatasan yang telah disebutkan. Dengan adanya penelitian ini, diharapkan dapat

memudahkan pengembangan perangkat lunak dan meningkatkan efisiensi pengembangan perangkat lunak tanpa perlu melalui proses *Coding* yang rumit. penelitian ini berharap dapat memberikan wawasan yang lebih mendalam tentang potensi low-code development dalam mendukung inovasi teknologi. Selain itu, penelitian ini juga bertujuan untuk memberikan gambaran tentang manfaat yang dapat diperoleh oleh pemula dalam dunia pengembangan aplikasi, serta sejauh mana *low-code* dapat menjadi solusi efektif dalam memenuhi kebutuhan pengembangan aplikasi yang beragam.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya, maka rumusan masalah pada penelitian ini yaitu:

1. Bagaimana metode pengembangan perangkat lunak minim kode dilakukan dengan perangkat Bubble?
2. Bagaimana kelebihan dan kekurangan perangkat Bubble untuk pengembangan perangkat lunak minim kode?

1.3 Batasan Masalah

Batasan masalah dalam penelitian yaitu:

1. *Software Bubble* yang digunakan dalam penelitian ini yaitu *Bubble Engine* versi 21.
2. Pendalaman *Software Bubble* dalam penelitian ini dikhususkan untuk membuat sebuah *website* versi *desktop*.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini yaitu:

1. Unuk mengetahui bagaimana perangkat lunak minim pengkodean *Bubble* bisa mempermudah para developer.
2. Untuk mengetahui proses dan cara kerja dari pengembangan perangkat lunak *Low-Code Bubble*.

1.5 Manfaat Penelitian

1. Manfaat untuk peneliti :
    a. Penelitian ini diharapkan dapat memberikan wawasan mengenai teknologi *Low-Code* khususnya *Bubble*.
    b. Penlitian ini diharapkan dapat memberikan pegetahuan lebih lanjut mengenai arsitektur, fitur, dan cara kerja *platform Bubble*.

c. Penelitian ini diharapkan dapat membantu peneliti selanjutnya dalam pengembangan solusi lebih lanjut untuk teknologi *Low-Code*.

2. Manfaat untuk pengembang :
   a. Penelitian ini diharapkan dapat memudahkan pengembang dalam melakukan pengembangan *website* tanpa memerlukan pegkodean yang rumit.
   b. Penilitian ini diharapkan dapat memangkas biaya dan waktu dalam mengembangkan suatu *project*.
   c. Penilitian ini diharapkan dapat meningkatkan keterampilan dan memicu inovasi dalam pengembangan fitur dalam suatu *project*.

1.6 Metodologi Penelitian Secara Umum

Metode-metode yang digunakan dalam penilitian ini adalah sebagai berikut :

1. Studi literatur
2. Eksplorasi perangkat *Bubble* dengan mengimplementasikan sebuah studi kasus pengembangan perangkat lunak menggunakan metode *prototyping*

1.7 Struktur Laporan

Struktur penelitan laporan Tugas Akhir ini terdiri dari 5 bagian diantaranya yaitu :

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang masalah yang diangkat, rumusan masalah, lingkup masalah, tujuan penelitian, manfaat penilitian, metodologi penilitian secara umum dan struktur laporan.

BAB II : LANDASAN TEORI

Bab ini berisi mengenai dasar teori yang digunakan dan mendasari penulisan tugas akhir. Meliputi teori yang mendukung dalam penelitian dan metode pengembangan hingga platform yang digunakan serta penelitian terdauhulu yang dijadikan acuan.

BAB III : METODOLOGI PENELITIAN

Bab ini berisi tentang metode analisis yang digunakan, menjelaskan tahapan-tahapan dalam metode *prototyping* yang digunakan yang harus dilalui untuk mencapai hasil akhir dan kesimpulan dalam penelitian.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi penjabaran hasil yang didapatkan selama mengeksplorasi *platform Bubble Low Code* dan hasil *website* dengan menggunakan *platform Bubble Low Code*.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil penelitian yang telah dilakukan oleh penulis untuk mengembangkan kekurangan dari hasil penelitian dan untuk proses pengembangan selanjutnya supaya lebih baik.

BAB II
LANDASAN TEORI

2.1 Metode Pengembangan Minim Kode

Low-Code development (pengembangan minim kode) merupakan sebuah cara untuk mendesain dan mengembangkan perangkat lunak dengan cepat dan dengan *coding* yang seminimal mungkin (Virta, 2018). Sedangkan menurut (rajgor & Marly, 2022) yang dimaksud dengan *Low-Code development* (pengembangan minim kode) merupakan sebuah pendekatan pengembanagan *visual drag-and-drop* yang menyederhanakan pendeployan aplikasi dengan mengurangi koding manual sebanyak mungkin.

Platform Minim Kode (*Low-Code Platform*) merupakan sebuah alat untuk *programmer* dan *non-programmer*. *Platform* Minim Kode (*Low-Code Platform*) merupakan salah satu alternatif pengembangan perangkat lunak. *Platform* Minim Kode (*Low-Code Platform*) memungkinkan pengembang untuk meminimalkan pengkodean secara manual dengan bahasa pemrogaman yang telah umum ditemui. Penggunaan fitur *drag* and *drop* merupakan salah satu fitur yang dapat membuat proses pengembangan menjadi lebih cepat. Pengembang dapat menggunakan *Platform* Minim Kode (*Low-Code Platform*) untuk membangun aplikasi dan *website* dengan tampilan visual aplikasi dan *website* secara langsung dan dapat diintegrasikan dengan *database* (Threestayanti, 2019). *Platform* ini bisa memungkinkan para pengembang untuk merancang *website* dan aplikasi hanya dengan pengetahuan tingkat lanjut.

Perbedaan yang mendasar antara *platform Low-Code* dan *programming* tradisonal yaitu seperti pada Gambar 2.1, Gambar 2.2, dan Gambar 2.3.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"
    integrity="sha512-E8QSvWZ0eCLGk4km3hxSsNmGWbLtSCSUcewDQPQWZF6pEU8GlT8a5fF32wOl1i8ftdMhssTrF/OhyGWwonTc
XA==="
    crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <title>Generate UUID</title>
</head>
<body>
    <div class="main">
        <div class="btn_uuid">
            <button id="btn">Generate UUID</button>
        </div>
        <div class="btn_uuid">
         <label id="show_uuid">Click the Button to Generate UUID</label>
        </div>
    </div>
</body>
<script>
let uuid_txt;
document.getElementById("btn").addEventListener("click", myFunction);
function myFunction() {
    uuid_txt = crypto.randomUUID();
    document.getElementById("show_uuid").innerHTML = uuid_txt;
}
</script>
</html>
```

Gambar 2.1 Contoh *progamming* tradisional kode index.html Sumber: (rajgor & Marly, 2022)



```
main {
    border: 2px solid rgb(73, 71, 71);
    border-radius: 10px;
    width: 50%;
}
.btn_uuid{
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 10px;
}
#btn{
    background-color: chartreuse;
    padding: 20px;
    border-radius: 10px;
}
```

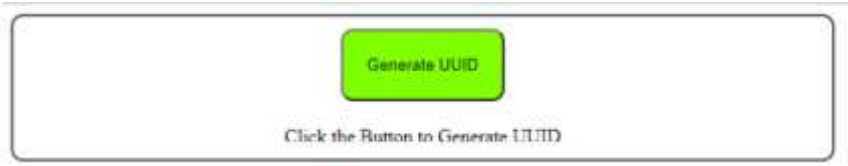
Gambar 2.2 Contoh *progamming* tradisional kode style css.html Sumber: (rajgor & Marly, 2022)

Pada Gambar 2.2 dan Gambar 2.3 merupakan contoh sebuah pengkodean tradisonal menggunakan HTML, CSS dan javaScript untuk menghasilkan sebuah UUID (*Universal Unique Identifier*). Dalam membuat UUID dengan *coding* tradisonal *programming* setidaknya membutuhkan sebanyak 50 baris kode.



Gambar 2.3 Contoh *platform Low-Code* Sumber: (rajgor & Marly, 2022)

Pada gambar 2.3 merupakan sebuh contoh pembuatan UUID dengan menggunakan *platform Low-Code.* Degan menggunakan *platform Low-Code* hanya tinggal men- *drag-and-drop* 1 tombol dan 1 label serta hanya membutuhkan 2 baris kode.



Gambar 2.4 Contoh *output* dari *programming* tradsional dan *Low-Code Platform* Sumber: (rajgor & Marly, 2022)

2.2 Platform Bubble

Bubble adalah sebuah platform minim pengkodean visual. *Bubble* memiliki fungsi utama mengembangkan *website* dan aplikasi *web* dengan *No-Code* yang memiliki fungsi lebih lanjut. *Bubble* pertama kali diperkenalkan pada tahun 2012 oleh Emmanuel Straschnov dan Josh Haas. Pada awalnya *Bubble* merupakan *Platform Bootstrap* selama 7 tahun. Pada tahun 2019 *Bubble* telah mengumpulkan sebanyak 6 juta USD dari *SignalFire*, Neo, Nas, Eric Ries dan para pendiri Warby Parker, Allbirds, Okta, Harry's. (Ericksson, 2020). URL dari *platform* Bubble yaitu https://bubble.io/.



Gambar 2.5 Bubble Sumber : Bubble.io



Gambar 2.6 Contoh Platform *Low-Code Bubble*

Dalam (Groc, 2020) terdapat beberapa hal yang dapat dan tidak dapat dilakukan dalam *platform Bubble*, yaitu :

1. Hal yang dapat dilakukan dalam *platform Bubble* :
    a. Membangun aplikasi web yang responsif

       Bubble adalah salah satu lingkungan pemrograman visual tercanggih untuk membangun aplikasi web yang kuat. Baik ingin membuat web responsif atau aplikasi web yang berfungsi pada ukuran layar apa pun, mungkin dapat dilakukan dengan *Bubble*.
    b. Membangun aplikasi web yang progressif

       Aplikasi web progresif (*Progressive Web Apps*) mudah dipasang hanya dengan 1 klik perangkat Android, sehingga dapat diakses dari layar beranda sama seperti aplikasi lainnya.
    c. Membangun aplikasi mobile *hybrid*

       Aplikasi *hybrid* menawarkan keuntungan besar bagi developer yang ingin sering memperbarui aplikasinya karena tidak perlu mengirimkan ulang versi baru.
    d. Menjalankan kode kostum

       Bubble menyediakan banyak fitur asli untuk membangun aplikasi web yang hebat. Sebagian besar alat tanpa kode tidak memungkinkan perluasan kemampuan dengan baris kode tradisional. Bubble menonjol dalam hal ini: memungkinkan pengembang menjalankan kode khusus untuk memperluas aplikasi mereka ketika mereka menemukan batasan.

e. Koneksi ke servis eksternal dan perangkat keras melalui melalui *APIs*

Platform *Bubble* menawarkan alat yang kuat, *The API connector*, untuk menghubungkan aplikasi ke sistem perangkat lunak eksternal atau perangkat keras. Secara teknis, hampir dapat terhubung ke segala sesuatu melalui *API*, seperti mesin bertenaga *AI* untuk mengklasifikasikan sekumpulan gambar yang sangat besar atau mengendalikan perangkat *IoT*.

f. Terhubung ke database eksternal

Secara bawaan, data aplikasi dibaca dan disimpan di server *AWS*. Editor *Bubble* menyediakan alat bantu, *SQL Database Connector*, yang memungkinkan pengembang untuk menghubungkan aplikasi ke database eksternal dan menjalankan queri *SQL* dari dalam *Bubble*

g. Meningkatkan aplikasi/ web seiring dengan pertumbuhan bisnis

Bubble dapat meningkatkan skalabilitas dan performa dan menawarkan berbagai paket hosting terjangkau yang sesuai dengan kebutuhhan performa aplikasi.

h. Mengelola akses dan privasi data

Bubble adalah *platform* yang *open-ended* dibandingkan dengan alat tanpa lainnya. Secara default, semua data terbuka untuk umum ketika Anda membuat aplikasi *Bubble*. Namun, editor *Bubble* menawarkan beberapa alat untuk mengonfigurasi akses data di aplikasi Anda. Memang, aturan privasi memberikan kekuatan untuk melindungi informasi dan memastikannya aman.

2. Hal yang tidak dapat dilakukan dalam *pltaform Bubble* :

a. Membangun *native apps*

Platform Bubble tidak menyediakan *engine* khusus untuk membangun dan menggunakan aplikasi *native* di bagian toko aplikasi.

b. *Host* aplikasi di server pribadi

Bubble menyediakan *hosting* untuk para pelanggan mereka pada server *Cloud Amazon Web Services* (*AWS*) atau instansi khusus. Untuk saat ini, belum memungkinkan untuk meng-*host* sendiri aplikasi *Bubble* pada layanan *hosting* pribadi atau server lokal

c. Menggunakan bahasa lain selain Javascript

Pengembang *Bubble* dapat memperluas fitur aplikasi dengan kode khusus. Tetapi *Bubble* hanya mendukung bahasa *Javascript* karena inti *engine* dari *Bubble* ditulis menggunnakan bahasa *Javascript.*

d. Mengembangkan algoritma kompleks

Bahasa visual *Bubble* bukanlah alat yang paling tepat untuk mengembangkan algoritma yang kompleks, terutama operasi rekursif atau perulangan data

e. Membuat gim tingkat lanjut

Bubble tidak bisa membuat dan menjalankan gim 3d yang kompleks dengan *Bubble engine* karena bubble dibuat dibuat untuk membuat aplikasi

f. Eksport kode

App *Bubble* hanya bisa dijalankan di *platform Bubble*. Jika ingin memutuskan untuk pindah dari *platform Bubble*, harus membuat ulang logika aplikasi *no-code* nya.

Bubble mempunyai keunggulan dalam proses pembuatan sebuah *website* yaitu (Ericksson, 2020):

1. Lebih aman dari kebanyakan aplikasi web yang dibuat oleh tim kecil karena didalam *platform Bubble* telah mengakomodasi banyak aspek yang diperlukan oleh seorang *developer*.
2. Dengan menggunakan *Bubble* dapat mengurangi sekitar 80% waktu pengembangan dibandingkan dengan menggunakan pengkodean tradisional.
3. Terdapat opsi untuk menerapkan klaster *Bubble* sendiri didalam *region AWS* pilihan anda sendiri.

2.3 Metode Prototyping

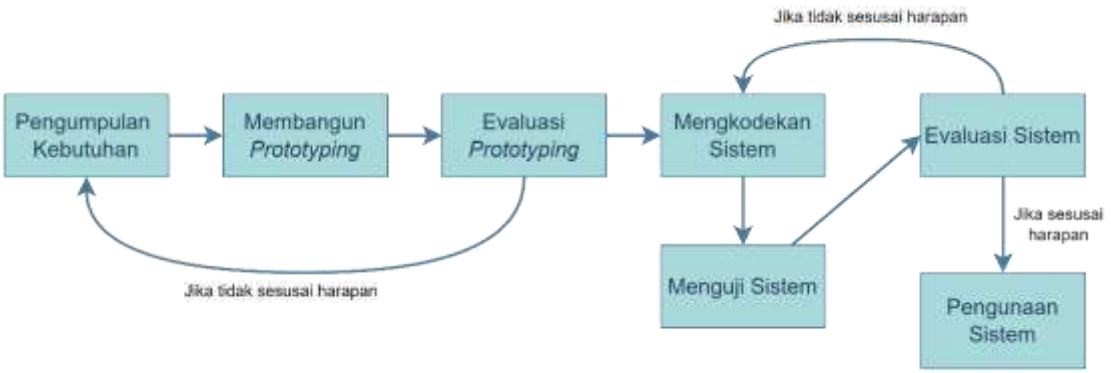
Proses pengembangan sistem seringkali menggunakan pendekatan prototipe (*prototyping*). Metode *prototyping* ini sangat baik digunakan untuk menyelesesaikan masalah kesalahpahaman antara pengguna dan analis yang ditimbulkan akibat pengguna tidak mampu mendefinisikan secara jelas kebutuhannya. *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang menggunakan pendekatan untuk membuat rancangan dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh calon pengguna (Prasetyo, 2017). Yang dimaksud dengan metode *Prototyping* menurut (Chandramita, 2019) yaitu suatu proses pembuatan *software* yang bersifat berulang dengan perencanaan cepat dimana terdapat umpan balik yang memungkinkan terjadinya perulangan dan perbaikan *software* sampai

memenuhi kebutuhan dari pengguna. Dengan metode *prototyping*, pengguna juga dapat memiliki suatu gambaran awal/dasar tentang program serta melakukan pengujian awal yang didasarkan pada konsep model kerja (*working model*) (Chandramita, 2019). Pengertian lain dari *prototype* yaitu demonstrasi awal dari sebuah perangkat lunak yang menjelaskan konsep dan tampilan. Pada umumnya dari proses demonstrasi tersebut muncul lebih banyak mengenai masalah-masalah dan solusinya, sehingga kebutuhan sistem yang sebelumnya tidak dideskripsi secara detail dapat diketahui. Selain itu *prototype* juga dapat digunakan sebagai teknik pengurangan resiko seperti kemunculan *error* dan hal-hal yang terlewatkan dalam persyaratan (Sommervile, 2011).

Metode prototyping akan menghasilkan sistem sebagai perantara anatara pengembang dan pengguna, hal tersebut karena dapat memudahkan pengguna dalam memlih sistem yang seusai dari apa yang diharapkan dengan pembuatan model *software* yang sederhana, dimana penggambaran dasar yang digunakan untuk dijadikan rancangan (Muhtadi & Zuanedi, 2021). *Prototyping* mempunyai tujuan yaitu mengembangkan model awal software menjadi sebuah sistem yang final. Tujuan sebuah *prototyping* bagi pengembang sistem yaitu untuk mengumpulkan informasi dari pengguna sehingga pengguna dapat berinteraksi dengan model *prototype* yang dikembangkan, karena *prototype* menggambarkan versi awal dari sistem untuk kelanjutan sistem sesungguhnya yang lebih besar. Terdapat keuntungan menggunakan metode *prototype* pada proyek perangkat lunak, yaitu (Sommervile, 2011):

1. Kegunaan sistem yang lebih baik.
2. Kesesuaian system yang lebih dekat dengan kebutuhan pengguna.
3. Kualitas desain yang lebih baik.
4. Keterpeliharaan system yang lebih baik.
5. Usaha Pengembanga yang lebih ringan.

Menurut (Muhtadi & Zuanedi, 2021) tahapan proses *prototyping* meliputi pengumpulan kebutuhan, membangun *prototyping*, evaluasi *prototyping*, mengkodekan sistem, menguji sistem, evalusi sistem, serta penggunaan sistem.



Gambar 2.7 Alur Tahapan Metode *Prototyping* Sumber : (Muhtadi & Zuanedi, 2021)

Berdasarkan Gambar 2.7 mengenai alur tahapan metode *prototyping* yaitu :

1. Pengempulan kebutuhan

   Pada tahapan ini pengembang melakukan identifikasi perangkat lunk dan semua kebutuhan sistem yang akan dibuat.

2. Membangun prototyping

   Membuat rancangan sementara yang berfokus pada alur program kepada pengguna.

3. Evaluasi prototyping

   Evaluasi dilakukan untuk mengetahui apakah model prototype sudah sesuai dengan harapan.

4. Mengkodekan sistem

   Jika *prototype* disetujui maka akan diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Pengujian sistem

   Setelah perangkat lunak sudah siap, perangkat lunak harus melewati pengujian. Pengujian ini biasanya dilakukan dengan *White Box Testing*, *Black Box Testing*, dan lain-lain.

6. Evaluasi sistem

   Pengguna melakukan evaluasi apakah perangkat lunak sudah sesuai dengan apa yang diharapkan atau tidak. Jika ya, lakukan tahap selanjutnya. Jika tidak, ulangi tahap mengkodekan sistem dan pengujian sistem.

7. Mengunakan Sistem

   Perangkat lunak yang telah diuji dan disetujui siap untuk digunakan.

2.4 Website

2.4.1 Pengertian Website

Menurut Agus Hariyanto dalam jurnal (Destiningrum & Adrian, 2017) Web merupakan kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (hyperlink). Sehingga yang dimaksud dengan *Website* yaitu sebuah *software* yang berfungsi untuk menampilkan dokumen pada suatu web yang membuat pengguna dapat mengakses internet melalui *software* yang terkoneksi dengan internet.

2.4.2 Indikator Website

Website yang baik tidak hanya terbatas pada masalah desain dan seni, tetapi juga harus menganut *methodology*, standar-standar baku pemograman yang telah ada. Hal ini penting karena begitu banyak hal yang dapat mempengaruhi mutu dari sebuah *website.* Adapun kriteria-kriteria *Website* yang baik menurut Suyanto dalam (Rahmasari & Haryadi, 2022) memperhatikan beberapa aspek, yaitu :

1. *Usability*

   Usability (kemudahan pengguna) merupakan pengalaman pengguna dalam berinteraksi dengan aplikasi atau situs *web* sampai pengguna dapat mengoperasikannya dengan mudah dan cepat. Untuk mencapai tingkat *usability* yang ideal, sebuah *website* harus memenuhi lima syarat berikut :

   a. Mudah untuk dipelajari
   b. Efisien dalam penggunaan
   c. Mudah untuk diingat
   d. Tingkat kesalahan rendah
   e. Kepuasaan pengguna

2. Sistem Navigasi

   Aspek navigasi (struktur) berkaitan dengan cara atau mekanisme perpindahan dari satu situs ke situs yang lain (*menu system*) di dalam sebuah system *website*. Navigasi dapat ditampilkan dalam berbagai media seperti teks, *image* atau animasi. Syarat navigasi yang baik yaitu :

   a. Mudah dipelajari
   b. Konsisten

c. Kemungkinan *feedback*
d. Muncul dalam konteks
e. Memberikan alternatif lain
f. Memberikan perhitungan waktu dan Tindakan
g. Menyediakan pesan visual yang jelas
h. Menggunakan label yang jelas dan mudah dipahami
i. Mendukung tujuan dan perilaku pengguna

3. Desain Visual (*Graphic Design*)

Desain yang baik sekurang-kurangnya memiliki komposisi warna yang baik dan konsisten, *layout* grafik yang konsisten, teks yang mudah dibaca, penggunaan grafik yang memperkuat isi teks, penggunaan animasi pada tempat yang tepat, isi animasi yang memperkuat isi teks, dan secara keseluruhan membentuk suatu pola yang harmonis.

4. Isi (*Contents*)

Konten atau isi adalah roh utama sebuah *website*, karena itu haruslah menarik, relevan, dan sesuai untuk target audien situs yang dituju. Situs *web* sebaiknya memiliki arsip dari konten data-data lama. Untuk objektivitas dan ketepatan informasi yang terdapat pada konten, lebih baik bila konten tersebut merupakan hasil kompilasi data dan diperkuat dengan pendapat pihak-pihak berwenang. Miliki data-data atau penjelasan pendukung. Sertakan juga daftar referensi dari sumber yang berwenang. Bila konten

berbentuk multimedia, usahakan berhubungan dengan isi situs *web*. Streaming bersama isi situs *web*. Sinkronkan antara audio dan visual. Jika perlu aksesnya dikelompokkan pada halaman tertentu dan diberikan informasi mengenai besar file dan total waktu pemutarannya.

5. Kompatibilitas (*Compatibility*)

Situs web harus kompatibel dengan berbagai perangkat tampilannya (*browser*), harus memberikan alternatif bagi *browser* yang tidak dapat

melihat situs tersebut.

6. *Loading Time*

Loading Time merupakan seberapa cepat respon ketika diakses. Bila sebuah situs *web* dapat tampil lebih cepat, kemungkinan besar *user* akan kembali mengunjungi situs tersebut, apalagi bila ditunjang dengan konten dan tampilan yang menarik.

7. *Functionality* (Aspek Teknologi)

   Fungsionalitas ini menyangkut beragam fasilitas dan kemudahan yang tersedia disitus *web*. Seberapa baik sebuah situs *web* bekerja dari aspek teknologinya, hal ini bisa melibatkan programmer dengan *script*-nya, misalnya HTML, (DHTML), PHP, ASP, COLDFUSION, CGI, dan SSI.

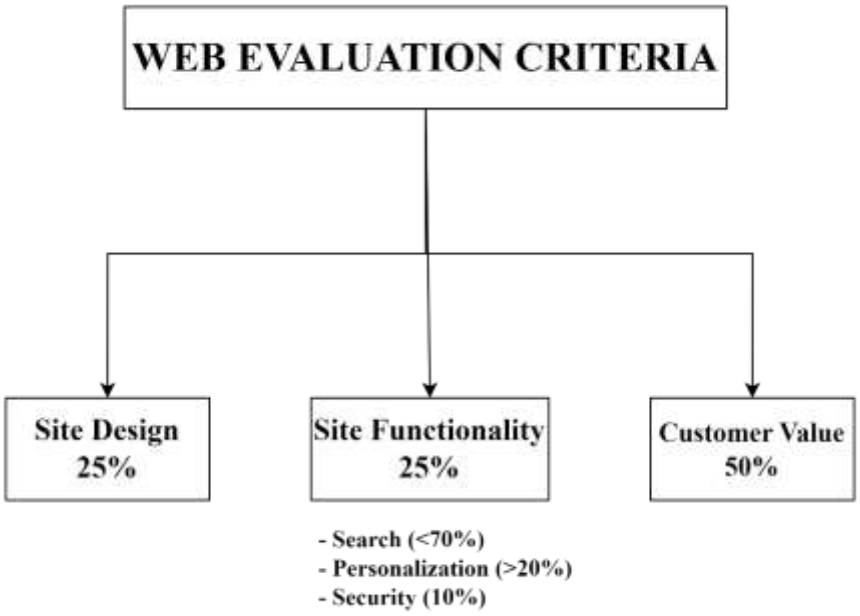
8. *Accesibility* (Kemudahan Akses)

   Halaman *web* harus dapat digunakan oleh setiap orang, tanpa memandang usia dan keadaan fisiknya. Hambatan infrastruktur juga harus diperhatikan, seperti akses internet yang lambat, spesifikasi komputer, penggunaan *browser*, dan lain-lain, yang dapat mempengaruhi akses seseorang, termasuk berbagai teknologi baru seperti PDA dan ponsel. *Web* juga harus dapat diakses melalui teknologi tersebut.

9. *Interactivity*

   Interactivity merupakan timbali balik pengguna dengan *website*. Salah satu tujuan dalam membangun sebuah web adalah berkomunikasi dua arah antara pengakses web dan pemilik web (Husna & Nafisah, 2019). Interaktivitas adalah hal-hal yang melibatkan pengguna situs *web* sebagai *user experience* dengan situs *web* itu sendiri. Dasar dari interaktivitas adalah *hyperlinks* dan mekanisme *feedback*. Hyperlinks digunakan untuk membawa pengunjung ke sumber berita, topik lebih lanjut, topik terkait, atau lainnya. Seperti *link* yang berbunyi *More info about this, glossar, related links*. Untuk mekanisme *feedback,* contohnya seperti kritik, komentar, pertanyaan, polling atau *survey*. Keuntungan dari adanya mekanisme *feedback* adalah user dapat memberitahu pengelola bila ada kesalahan pada situs seperti *missinglink*, *dead link*, atau kesalahan lainnya. *User* juga dapat memberikan kritik dan saran demi kemajuan situs.

Menurut kajian Indrajit (2001) dalam jurnal (Asriani, 2011),Terdapat tiga kriteria utama yang harus diperhatikan di dalam sebuah *website,* yaitu :



Gambar 2.8 Tiga Kriteria Evaluasi *Web* Sumber: (Asriani, 2011)

1. *Site Design*

   Menyangkut hal-hal yang berkaitan dengan tampilan *website* dan system menu yang dipergunakaan. Ada dua sub-kriteria yang harus dipertimbangkan dalam membuat desain sebuah situs, yaitu : Navigasi dan Estetika. Aspek navigasi berkaitan dengan cara atau mekanisme perpindahan dari satu situs ke situs yang lain (menu *system*) di dalam sebuah sistem *website.*

2. *Site functionality*

   Menyangkut beragam fasilitas dan kemudahan yang tersedia di *website.* Dari tiga buah sub-kriteria yang ada, fasislitas pencarian (*searching*) merupakan yang terpenting (70%). Hal ini disebabkan karena seluruh profil Perusahaan yang sedemikian besar harus diwakilkan oleh sebuah *website* yang terbatas sehingga dibutuhkan sebuah "*information desk*" atau "*reception desk*" yang dapat dikunjungi pelanggan sewaktu-waktu.

3. *Customer Value*

   Menyangkut berbagai aspek manfaat yang dapat secara langsung dirasakan para user yang mengakses *website*. *Content* merupakan hal pertama yang dipandang dapat memberikan manfaat kepada pelanggan. Seorang individu mengakses ke *website* tertentu karena ingin data atau informasi terkait dengan kebutuhannya. Jika yang bersangkutan tidak dapat menemukan apa yang diinginkannya didalam *website* terkait, maka yang bersangkutan akan mencarinya di *website-website* lain.

2.5 Tahapan Pengembangan *Website*

Pengembangan *website* (*Website Development*) merupakan sebuah proses dalam pembuatan dan memelihara sebuah *website* menggunakan sebuah teknlogi seperti *HTML, PHP, dan JavaScript*. Tahapan pengembangan *website* berdasarkan pendekatan yang dikemukakan oleh Howcroft dan Varol (2000) dalam (Mauko, Setiohardjo, & Noach, 2017) yaitu:

1. Tahap Analisis

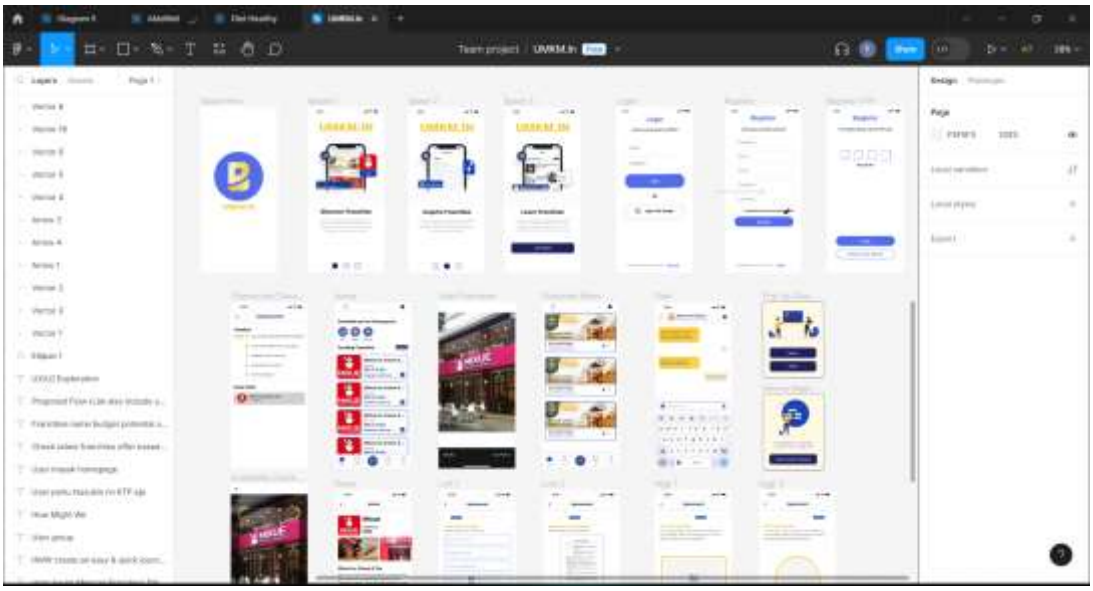
   Pada tahap analisis berkaitan dengan penetapan tujuan stategis terkait pengembangan *website* dan analisis bagaimana situs *website* dapat mencapai tujuan srategis. Pada tahap analisis ini dibagi menjadi 6 tugas yaitu:

   a. Analisis teknologi, dimana pada bagian ini mempunyai tujuan untuk mengidentifikasi semua komponen teknologi dan peralatan yang dibutuhkan dalam membangun dan mendukung situs *website* yang akan dikembangkan.
   b. Analisis informasi, pada bagian ini mempunyai tujuan untuk mengidentifikasi informasi yang dibutuhkan pengguna, baik statis (halaman web) maupun dinamis (berdasarkan request ke server database).
   c. Analisis keterampilan, pada bagian ini mempunyai tujuan untuk mengidentifikasi beragam keterampilan yang dibutuhkan untuk menyelesaikan proyek.
   d. Analisis pengguna, pada bagian ini mempunyai tujuan untuk mengidentifikasi semua pengguna situ yang dimaksud. Bagian ini merupakan proses yang jauh lebih kompleks daripada pengembangan system informasi tradisional sebagai jangkauan pengguna.
   e. Analisis biaya, pada bagian ini mempunyai tujuan untuk memperkirakan biaya dalam pengembangan situs *website*.
   f. Analisis resiko, pada bagian ini mempunyai tujuan untuk menganalisis resiko utama yang terkait dengan pembengan situs *website*.

2. Tahap Design

   Pada tahap design dilakukan perancangan sistem berdasarkan dokumen tujuan yang dihasilkan dari tahap pertama, diantaranya desain informasi yang menggambarkan struktur web, struktur data dan fungsi dari setiap fitur serta desain grafis yang meliputi tata letak, warna, gambar dan animasi. Pada tahap ini juga dilakukan pengujian

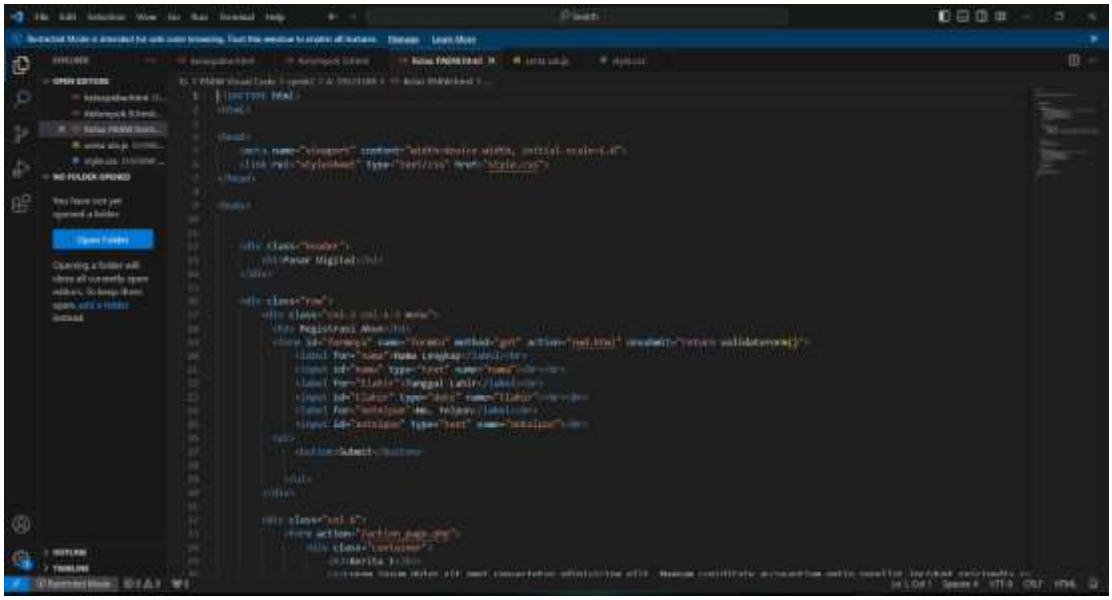
terhadap desain untuk menemukan inkonsistensi atau kesalahan yang tidak sesuai dengan sasaran dan tujuan yang diuraikan pada tahap pertama serta untuk memastikan apakah sistem dapat menghasilkan informasi yang dibutuhkan oleh pengguna dengan benar. Pada tahap design ini dirancang dengan menggunakan Figma



Gambar 2.9 Contoh Tahap Design Dengan Figma

3. Tahap Generation

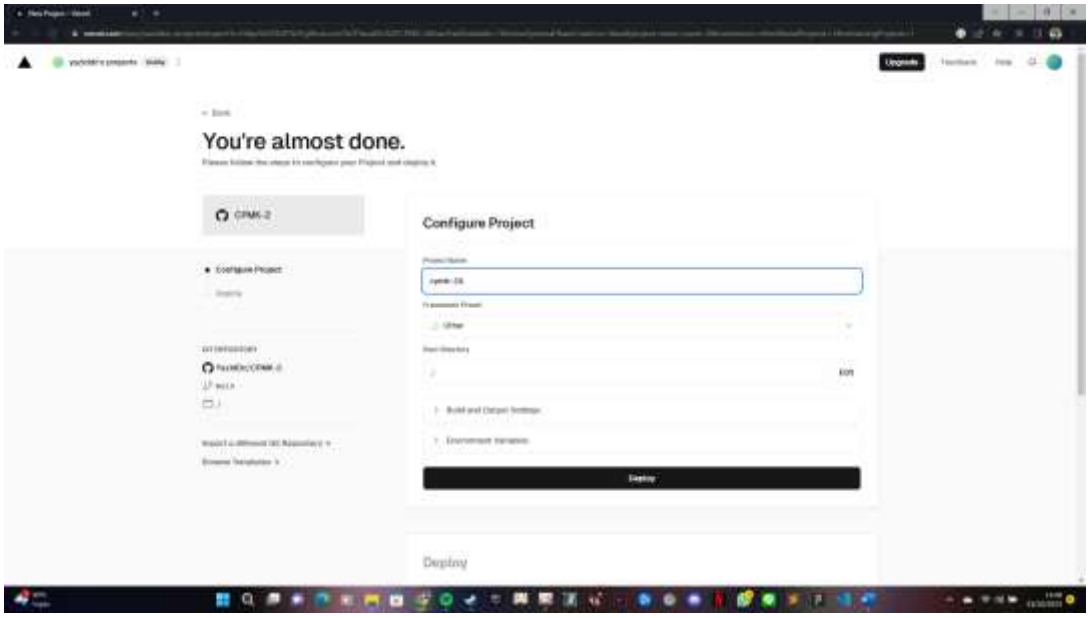
Tahap ini mencakup pengembangan situs *website* berdasarkan desain yang telah dibuat sebelumnya dengan melakukan penulisan *source* code program untuk membangun sistem dengan menggunakan bahasa HTML dan CSS sebagai perancangan antarmuka, serta PHP sebagai bahasa pemrograman yang membentuk logika dari pemrosesan sistem Semua sumber daya untuk pengembangan situs *website* seperti perangkat keras dan perangkat lunak akan dipilih dalam tahapan ini. Langkah pengkodean melihat generasi semua perangkat lunak yang terhubung dengan situs dan pemasangannya ke *server Web* yang relevan. Ini mungkin hanya melibatkan posting situs server Web, tapi juga bisa melibatkan tugas yang lebih rumit, seperti koneksi *database*. Pengujian adalah salah satu bidang yang paling kompleks dan sulit dari proyek pengembangan *website*. Ini bahkan lebih kompleks dari pada sistem informasi tradisional, karena aplikasi *Web* sering dikembangkan untuk sekelompok pengguna yang luas (sering tidak diketahui) di lingkungan teknologi yang berbeda. Situs *website* harus diuji terhadap sebanyak mungkin lingkungan dan kombinasi teknologi ini supaya dapat memaksimalkan hal yang potensial.



Gambar 2.10 Contoh Kode Pengembangan *Website*

4. Tahap Implementation

Tahap implementation merupakan tahap akhir dan yang paling sederhana namun paling penting. Pada tahap ini situs *website* yang telah dikembangkan selanjutnya dilakukan hosting. Hosting merupakan layanan berbasis internet sebagai tempat penyimpanan data atau tempat menjalankan aplikasi ditempat terpusat yang disebut dengan *server* dan dapat diakses melalui jaringan internet. Untuk memastikan pengunjung merasakan manfaat dari *website* maka kehadiran sebuah situs harus dirasakan dan isinya harus dianggap bernilai. Selama tahap ini, situs harus terdaftar di mesin pencari utama, bersama dengan metode promosi lainnya seperti memberi tahu newsgroup yang relevan, pencetakan Alamat domain situs Web pada kop lembaga, kartunama, dll. Pada tahap ini juga dilakukan *maintenance* atau perawatan sistem.



Gambar 2.11 Contoh Hosting *Website*

2.6 Penelitian Terdahulu

Penelitian terdahulu mempunyai manfaat sebagai tolak ukur penelitian dan membantu pemahaman yang lebih baik terhadap penelitian yang sedang dilakukan, sehingga dengan penelitian terdahulu membantu mengenal lebih mengenai *Low-Code* serta *platform Bubble*. Tabel 2.1 merupakan penelitian terdahulu yang dijadikan acuan dalam penelitian ini.

Tabel 2.1 Penelitian Terdahulu

Judul	Tahun	Penulis	Pembahasan
Implementasi Black Box Testing dan Usability Testing pada *website* MI iftahul Ulum Warugunung Surabaya	2023	(Abdillah, Kurniastuti, Susanto, & Yudianto, 2023)	Penelitian ini bertujuan untuk melakukan pengujian *website* sekolah pada MI Miftaul Ulum Warugunung Surabaya dengan menggunakan black box testing dan *usability testing*. Hasil black box testing menunjukkan dari 30 fitur sistem yang diuji, semua fitur berhasil berjalan sukses sesuai dengan hasil yang diharapkan. Pengujian *usability* testing menghasilkan nilai *usability* sebesar 0,90 dengan mendekati nilai 1 sehingga

Judul	Tahun	Penulis	Pembahasan
			website termasuk sistem informasi yang baik.
Penerapan Low-Code Platform dalam Pengembagan Aplikasi Presensi	2022	(Akbar & Idris, 2022)	Penelitian ini menggunakan Low-Code platform *Outsystems* dalam pengembangan aplikasi presensi yang dibe nama ACCAbsensi. Pengunaan Low-Code dalam jurnal ini mempunyai tujuan untuk mengatasi permasalahan secara cepat dan tepat serta mempermudah kinerja pengembang karena dapat memudahkan pengembengaan dan meminimalisir waktu pengembangan.
Pengembanga Aplikasi Layanan Multiguna Menggunakan Low-Code Platform	2021	(Abednego, Suyoto, & Julianto, 2021)	Penelitian ini mempunyai tujuan untuk mengembangkan aplikasi layanan multiguna mengenai layanan penngajuan pinjaman dana dari Astra Credit Companies untuk Masyarakat dengan jamninan berupa BPKB. Pengembangan aplikasi tersbut menggunakan *platform Low-Code Outsystems* karena dapat dengan cepat membuat sebuah perangkat lunak, dimana system pembuatan *User Interface*nya yaitu secara *drag and drop*, dan dapat bekerja dengan s*ystem cloud computing*.
Could I Build My Product on Bubble.io	2020	(Ericksson, 2020):	Penelitian ini mempunyai tujuan untuk menyakinkan kita bahwasannya kita bisa membuat *website* dan aplikasi didalam Bubble.io. Ericksson menyatakan bahwa Bubble sangat membantu dalam

Judul	Tahun	Penulis	Pembahasan
			memangkas waktu sebanyak 80% ketika membuat *website* dan aplikasi dibandingkan dengan cara coding tradisional. Bubble dapat terkoneksi dengan API menggunakan Bubble's API Connector plugin. Bubble bisa terintegrasi dengan layanan umum Seperti : OAuth (open aunthentication), Goggle Analytics, Figma for Design, Google Maps APIs, DLL. Ericksson juga menyatakan bahwa Bubble bisa meluncurkan app kalian di klaster Bubble di Amazon Web Services (AWS), Paket berbayar yang disediakan oleh Bubble juga memberikan kapasitas ekstra pada klaster ini hingga ke tingkat produksi.
Relation of low-code development to standard software development: Case Biit Oy Master's	2018	(Virta, 2018)	Penelitian ini mempunyai beberapa tujuan antara lain mengetahui pengembangan *Low-Code*, memungkinkan *Low-Code* ditawarkan kepada para pengembang perangkat lunak, *platform* apa yang memiliki *Low-Code* didalamnya, dan mengenai perbedaan *Low*-Code dengan pengembangan perangkat lunak standar. Keuntungan yang didapatkan setelah menggunakan Perangkat lunak *Low-Code Salesforce* adalah dapat menyederhanakan proses otomatisasi, kecepatan pengembangan meningkat, dan tidak diperlukan unit tes didalamnya.

Judul	Tahun	Penulis	Pembahasan
			Namun, perangkat lunak Low-Code *Salesforce* juga memiliki beberapa kerugian dalam penggunaannya seperti alur kerja didalamnya tidak dapat menangani volume data yang besar, ada beberapa karyawan di Biit, yang diwawancari oleh Tatu Virta lebih memilih untuk melakukan pengkodean secara manual karena lebih efisien dan lebih cepat untuk dibuat, dan lain sebagainya.

BAB III
METODOLOGI

Bab ini berisi tentang langkah pengerjaan Tugas Akhir. Dimana Tugas Akhir ini dilakukan berdasarkan langkah-langkah yang sistematis dengan tujuan memperjelas dan mempermudah penyelesaian masalah. Metodologi yang digunakan dalam pengerjaan Tugas Akhir ini terdiri dari lima (5) bagian utama yaitu studi litelatur, pengumpulan data game, eksplorasi *platform bubble* dan pengembangan *prototyping*, serta pengujian.

3.1 Alur Pengembangan

Gambar 3.1 menunjukkan alur pengembangan dalam mengeksplorasi dan perancangan sebuah *website* menggunakan *platform Bubble* didalam penelitian ini. Dimana langkah-langkahnya yaitu studi litelatur, pengumpulan data game, eksplorasi *platform bubble* dan pengembangan *prototyping*, serta pengujian.



Gambar 3.1 Alur Pengembangan

Tahapan atau alur pengembangan dalam penelitian ini dapat dituliskan sebagai berikut :

1. Studi Literatur

   Pada tahapan ini, peneliti mengkaji berbagai jurnal, artikel dan penelitian terdahulu yang sesuai dengan topik penelitian untuk mendapatkan berbagai informasi mengenai permasalahan, metode dan solusi. Studi literatur yang dilakukan yaitu untuk mengetahui mengenai *platform* Bubble serta pembuatan *website* yang baik dan benar dengan menggunakan *platform* Bubble.io.

2. Pengumpulan data game

   Tahap pengumpulan data *game* yaitu melakukan pengumpulan dan identifikasi game apa saja yang akan ditampilkan dalam website *marketplace game* yang akan dibuat. Untuk melakukan hal ini, informasi tentang tren game sangat penting. Pengambilan data *game* dilakukan dengan memanfaatkan informasi *trend game* berdasarkan *platform* Steamdb dan hasil diskusi dari komunitas Discord. Analisis tren dari *platform* Steamdb tersebut memberikan wawasan tentang popularitas dan permintaan untuk game tertentu. Selain itu, interaksi dengan komunitas juga menjadi sumber informasi berharga, karena pendapat dan preferensi pengguna dapat memengaruhi pilihan game yang akan disajikan.

3. Eskplorasi *Platform Bubble*

   Dalam penelitian ini, eksplorasi platform Bubble versi 29 dilakukan menggunakan perangkat laptop Asus TUF 505DT. Proses eksplorasi dimulai dengan mengunduh, menjalankan, dan mempelajari *tools* yang tersedia dalam platform Bubble. Laptop Asus TUF 505DT dipilih karena kapabilitasnya yang cukup untuk menjalankan platform tersebut dengan lancar. Platform bubble bisa di donwload di

   Selama proses eksplorasi, dilakukan analisis terhadap fitur-fitur yang ada, pengujian performa, serta pemahaman mendalam tentang cara kerja *platform* dan secara aktif menggunakan *platform* ini untuk memahami fitur-fitur yang disediakan, proses pengembangannya, serta kemampuan dan keterbatasannya dalam mendukung pengembangan perangkat lunak minim kode. Hal ini dilakukan karena *platform* Bubble masih tergolong baru sebagai *platform* pengembengan perangkat lunak minim kode.

4. Pengembangan Prototype

   Pengembangan *Prototype* dalam penelitian ini memanfaatkan fitur desain visual yang disediakan oleh *platform* Bubble.io. *Prototype* yang sedang dikembangkan adalah sebuah *marketplace* untuk game. Dalam proses pengembangan ini, peneliti

menggunakan *template* yang telah tersedia di Bubble.io sebagai landasan. Kemudian, peneliti melakukan penyesuaian *template* tersebut sesuai dengan kebutuhan, seperti pengaturan warna, tata letak, dan jenis huruf yang digunakan. Selain itu, data-game yang telah dikumpulkan sebelumnya juga ditambahkan ke dalam *prototype*. Pendekatan ini memungkinkan peneliti untuk mengetahui secara langsung apa yang terlibat dalam mengembangkan perangkat lunak minim kode dengan Bubble dan mengukur tingkat kesulitan, kekurangan serta kelebihan dari pengalaman menggunakan Bubble.io untuk membuat perangkat lunak minim kode.

5. Pegujian

   Setelah menyelesaikan tahap pengembangan *prototyping* menggunakan platform Bubble, langkah selanjutnya adalah melakukan pengujian untuk memastikan fungsionalitas dan kelayakan penggunaan. Pengujian prototyping juga dilakukan untuk melihat apakah semua fitur bekerja dengan baik, dan apakah ada masalah teknis atau kekurangan lain yang perlu diperbaiki. Dengan ini akan membantu untuk mengetahui seberapa efektif Bubble dalam pengembangan perangkat lunak minim kode. Metode yang digunakan dalam pengujian ini adalah Black box *testing* dan Usability *testing*. Black box testing dilakukan untuk menguji fungsionalitas sistem tanpa perlu mengetahui detail internal dari kode program. Sementara itu, Usability testing bertujuan untuk mengevaluasi seberapa mudah pengguna dapat menggunakan sistem. Melalui kedua metode ini, peneliti dapat mengidentifikasi potensi masalah dan kekurangan dalam sistem, serta memperbaikinya. Proses pengujian ini penting untuk memastikan bahwa *marketplace* gim yang dikembangkan memiliki kinerja yang optimal.

3.2 Studi Literatur

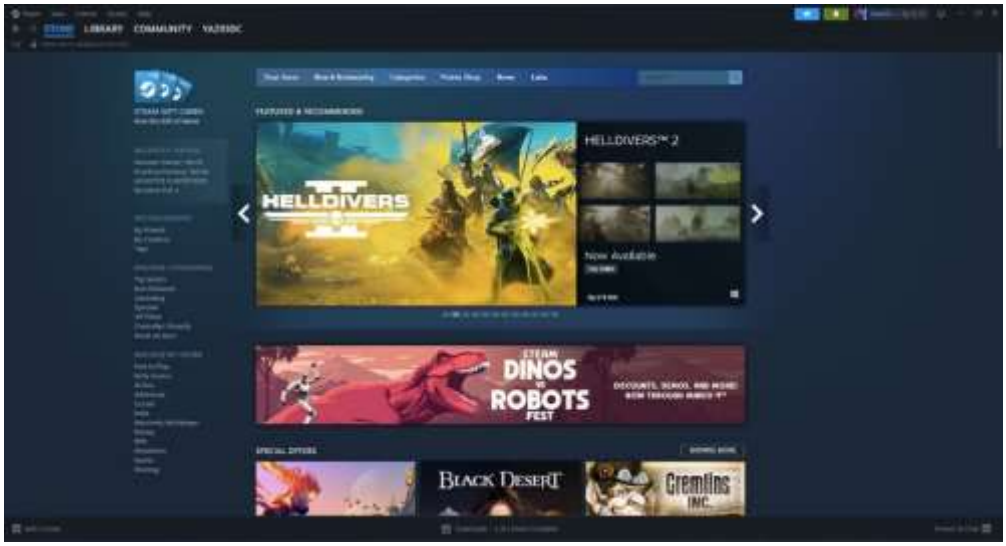
Studi literatur merupakan serangkaian kegiatan dalam metode penelitian untuk mengumpulkan, menelaah, dan menganalisis berbagai sumber informasi yang relevan dengan topik penelitian. Tahap ini merupakan bagian penting dari metode penelitian yang melibatkan pengumpulan, penelaahan, dan analisis berbagai sumber informasi yang relevan dengan topik penelitian untuk dapat membantu dalam membangun basis pengetahuan yang kuat untuk penelitian yang akan dilakukan, memberikan konteks dan latar belakang yang diperlukan untuk pemahaman yang lebih baik tentang masalah penelitian serta membantu dalam memberikan wawasan lebih mengenai topik yang akan diteliti. Tujuan studi literatur dalam penelitian ini

yaitu untuk dapat memahami, mengidentifikasi dan menambah referensi mengenai pengembangan perangkat lunak minim kode dan *platform* Bubble.io. Studi literatur dalam penelitian ini dilakukan dengan mengumpulkan berbagai informasi yang relevan dan mengkaji berbagai literatur yang berkaitan dengan penelitian pengembangan perangkat lunak minim kode dan *platform* Bubble.io. Literatur-literatur tersebut diperoleh dari:

1. Buku-buku dan jurnal-jurnal penelitian serta artikel baik yang diterbitkan baik dari dalam negeri maupun luar negeri.
2. Informasi dari internet seperti youtube dan blog yang terpercaya.

3.3 Pengumpulan Data Game

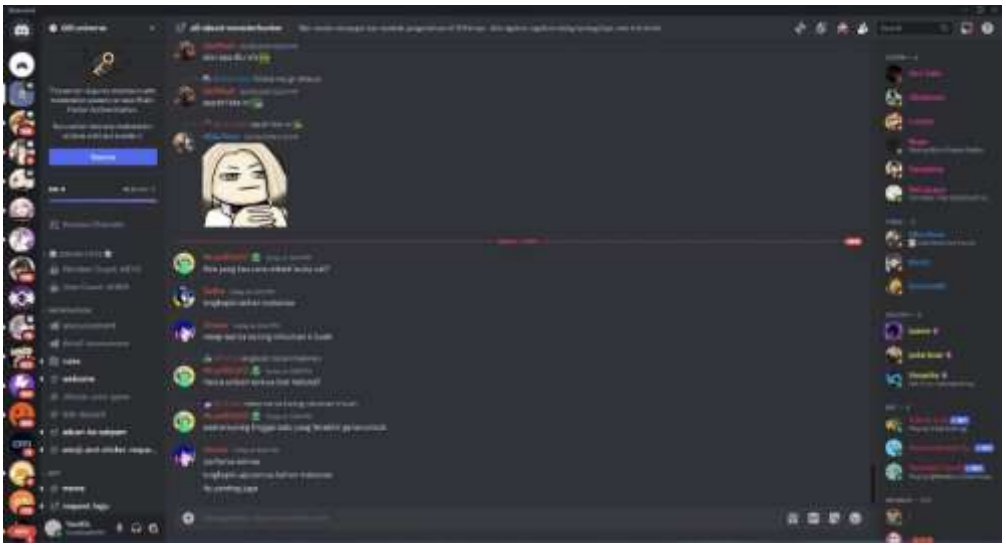
Pengumpulan data *game* dalm penelitian ini yaitu dengan melakukan *market research game* untuk mengetahui *game* dan *genre game* apa saja yang sedang populer dan sering dimainkan atau *game* apa saja yang banyak penikmatnya. Referensi-referensi yang dihasilkan tersebut yang nantinya dijadikan acuan untuk mengetahui *game* apa saja yang akan ditampilkan dalam prototyping *website marketplace game*. Dalam langkah pengumpumlan data dan pencarian data *game* melalui *platform* Steamdb dan komunitas *game* di Discord. Dengan menggunakan *platform* Steamdb maka dapat melihat statitsik mengenai informasi *game* seperti jumlah pemain secara *real-time*, ulasan pengguna. Sedangkan komunitas *game* dalam Discord memungkinkan untuk mengetahui informasi dengan cara berinteraksi secara langsung dengan para penggemar *game*, dan dapat mendengarkan rekomendasi dari para anggota komunitas *game* serta dapat menambah wawasan dari hasil diskusi. Proses pengumpulan data yang cermat dan komprehensif adalah langkah penting dalam memastikan bahwa *game-game* yang dipilih memenuhi kebutuhan dan minat pengguna di dalam *prototyping marketplace game* yang akan dibuat.



Gambar 3.2 Platform Steam



Gambar 3.3 Platform Steamdb Data dan Rating Game



Gambar 3.4 Komunitas Game di Discord

3.4 Eksplorasi Platform Bubble

Eksplorasi *platform* Bubble dalam penelitian ini berfokus pada versi 21 dari Bubble Engine dan akan ditekankan pada pembuatan *website* desktop. Tujuan utama dari eskplorasi *platform* Bubble adalah untuk memahami bagaimana Bubble dapat mempermudah pekerjaan pengembang dan memahami proses serta cara kerja dari pengembangan perangkat lunak dengan pendekatan minim kode di Bubble.. Eksplorasi platform Bubble akan dilakukan untuk memahami fitur dan cara kerja *platform* untuk mengembangakan perangkat lunak yang minim kode.

3.5 Pengembangan Prototype

Setelah melakukan tahapan eksplorasi *platform* Bubble.io, selanjutkan melakukan pengembangan *prototype*. Melalui pembuatan prototype ini, peneliti dapat lebih mengetahui tentang manfaat, kelebihan dan keterbatasan penggunaan *platform* Bubble.io dalam mengembakan sebuah perangkat lunak terutama *website*. Pada tahapan ini akan dibuat sebuah purwarupa dari sistem yang akan dibuat. Menurut kamus besar Bahasa Indonesia pengertian purwarupa adalah rupa yang pertama atau rupa awal. Sehingga, purwarupa dapat disebut sebagai rupa awal yang dibuat untuk mewakili skala sebenarnya sebelum dikembalikan atau justru dibuat khusus untuk pengembangan sebelum dibuat dalam skala sebenarnya.

3.6 Metode Pengujian

Metode pengujian yang dilakukan dalam penelitian ini yaitu dengan menggunakan metode *Black Box testing* dan *Usability testing*. Pengujian dengan menggunakan metode *Black Box testing* berfokus pada *input* dan *output* sebuah *website*, apakah sebuah *website* sudah layak untuk digunakan atau belum dengan melihat data yang diuji. Menurut Hamza (2019) dalam penelitian (Abdillah, Kurniastuti, Susanto, & Yudianto, 2023) *Black box testing* merupakan salah satu jenis pengujian perangkat lunak yang berfokus memastikan masalah yang dapat muncul pada perangkat lunak ketika digunakan. Pengujian *black box testing* diperlukan untuk mengetahui apakah *website* yang dikembangkan berjalan sesuai dengan yang diharapkan atau tidak. Pengujian *black* box bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi. Pengujian dengan menggunakan *black box testing* cenderung untuk menemukan kesalahan sebagai berikut (Jaya, 2018):

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka
3. Kesalahan pada struktur data dan akses basis data.

4. Kesalahan performansi (*performance error*). (Sommervile, 2011)
5. kesalahan inisialisasi dan terminasi.

Pada pengujian dengan menggunakan metode *Usability testing* menggunakan kuisioner yang sesuai dengan standar internasional yaitu USE Quesionnaire yang dikembangkan STC *Usability and User Experience Community* dari Arnold M. Lund. Kuisioner USE merupakan bentuk kuisioner untuk membantu dalam pengukuran *usability* produk maupun jasa secara subyektif (Sasongko, Jayanti, & Risdiansyah, 2020). Kuesioner ini terdiri dari 4 dimensi yaitu kegunaan (*Usefulness*), kemudahan untuk digunakan (*Ease of Use*), kemudahan dipelajari (*Ease of Learning*) dan kepuasan (*Satisfaction*). Kuisioner tersebut dibuat dalam bentuk lima (5) poin dengan model skala *Likert* untuk mengukur tingkat persetujuan pengguna terhadap pernyataan kuesioner. Tabel 2.1 menunjukan list pernyataan untuk pengujian dengan menggunakna metode *usability testing*.

Tabel 3.1 List Pernyataan dalam USE *Quesionnaire*

No.	**Pernyataan**
Usefulness	
1.	Apakah informasi yang Anda butuhkan mudah ditemukan pada aplikasi ini?
2.	Seberapa membantu fitur pencarian dalam menemukan game atau konten lain?
3.	Apakah fitur rekomendasi game relevan dengan preferensi Anda?
4.	Seberapa baik aplikasi ini mendukung kebutuhan Anda dalam berinteraksi dengan komunitas game?
5.	*Website* ini mudah digunakan
Ease of Use	
6.	Seberapa mudah Anda dapat menavigasi menu utama?
7.	Apakah tata letak aplikasi ini intuitif dan logis?
8.	Website ini mudah dipahami oleh saya
9.	Seberapa mudah Anda dapat mengatur pengaturan pribadi dan preferensi dalam aplikasi?
10.	Saya bisa menggunakan *website* ini tanpa instruktsi tertulis
Easy of Learning	
11.	Saya melihat *website* ini konsisten

No.	Pernyataan
12.	Seberapa cepat Anda bisa belajar menggunakan fitur-fitur baru dalam aplikasi ini?
13.	Seberapa mudah Anda memahami ikon dan label pada aplikasi ini?
14.	Apakah Anda merasa memerlukan waktu lama untuk beradaptasi dengan desain baru ini?
15	Seberapa jelas dan mudah dipahami petunjuk penggunaan yang diberikan dalam aplikasi?
Satisfaction	
16.	Seberapa puas Anda dengan tampilan visual dari aplikasi ini?
17.	Apakah Anda merasa nyaman menggunakan aplikasi ini untuk waktu yang lama?
18.	Seberapa puas Anda dengan kecepatan respon aplikasi ini?
19.	Apakah Anda merasa aplikasi ini memberikan pengalaman pengguna yang menyenangkan?
20	Secara keseluruhan, seberapa puas Anda dengan redesign UI/UX dari aplikasi ini?

Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial yang merupakan skala kontinum bipolar, pada ujung sebelah kiri (angka rendah) menggambarkan suatu jawaban yang bersifat *negative*. Sedang ujung sebelah kanan (angka tinggi), menggambarkan suatu jawaban yang bersifat positif (Rahadi, 2014). Tabel 3.2 merupakan pengukuran skala likeart untuk memberikan nilai pada pernyataan dalam pengujian yang digunakan dalam penelitian.

Tabel 3.2 Tabel Nilai untuk Pernyataan dalam USE *Quesionnaire*

LS	**1**	**2**	**3**	**4**	**5**
Nilai	STS	TS	N	S	SS

Keteranngan :

PK = Pernyataan Kuesioner

STS = Sangat Tidak Setuju

TS = Tidak Setuju
N = Netral
S = Setuju
SS = Sangat Setuju

Dengan produk yang diuji memiliki kegunaan yang baik. Berikut adalah interpretasi umum dari nilai usability (Hesti Wahyuningtyas, 2022) :

1. 0 - 0,2: Usability sangat rendah - Pengguna mengalami kesulitan besar dalam menggunakan sistem atau produk.
2. 0,2 - 0,4: Usability rendah - Banyak masalah usability yang mempengaruhi pengalaman pengguna.
3. 0,4 - 0,6: Usability sedang - Ada beberapa masalah usability yang perlu diperbaiki.
4. 0,6 - 0,8: Usability tinggi - Pengguna sebagian besar puas dengan sistem atau produk, tetapi masih ada ruang untuk perbaikan.
5. 0,8 - 1: Usability sangat tinggi - Sistem atau produk sangat mudah digunakan dan memberikan pengalaman pengguna yang sangat baik.

BAB IV
HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dari setiap proses yang ada pada penelitin ini dan membahas hasil dari tampilan *platform* bubble.io, serta tampilan *website* yang dihasilkan dengan menggunakan *platform* bubble.io

4.1 Hasil Kajian Pustaka

Hasil kajian pustaka dari penelitian ini dengan mencari literatur penelitian terdahulu dari bubble.io dan pembuat *website* sejenis. Literatur pembanding pertama adalah "*Could I Build My Product On Bubble.io*" (Ericksson, 2020). Literatur ini membahas pengalaman selama menggunakan *platform* bubble.io yang dilihat dari sudut pandang *developer*. Hasil literatur ini mencakup berbagai macam hal diantaranya Keamanan, berapa banyak waktu yang dihemat, kelebihan, kekurangan dan dll. Keamanan dari *platform bubble* menurut (Ericksson, 2020) adalah *platform* bubble.io memiliki tingkat keamanan yang bagus dibandingkan dengan keamanan web yang di buat oleh tim kecil, yang mana pihak bubble.io sudah menamakan berbagai macam aspek dalam keamanan *platform* mereka sehingga para developer hanya perlu mengimplementasikan aplikasi mereka sesuai dengan kemampuan masing-masing. Penelitian ini juga menjelaskan bahwa pada saat menggunakan bubble dapat mengurangi sekitar 80% waktu untuk membuat *website* dari *full-stack developer* atau sebagai tim. Contoh beberapa hal yang menghemat waktu diantara nya tidak perlu menginstal dan mengkonfigurasi database, tidak perlu user signup, login ataupun mengautentikasi.

Literatur kedua adalah "Pengembangan Aplikasi Layanan Multiguna Menggunakan *Low-Code* Platform" (Abednego, Suyoto, & Julianto, 2021). Literatur ini membahas tentang pengembangan aplikasi menggunakan *platform Outsystems*. Literatur ini membahasa kelebihan dan kekuranagan pada *platform Outsystems* berdasarkan *review* yang dilakukan oleh peneliti. Kelebihan dari *platfrom Outsystems* yaitu dapat mengatasi input yang diluar batas kenormalan, seperti mengatasi input yang kosong dan input angka yang tidak dapat di isi dengan huruf. *Platform Outsystems* juga dapat mengatasi perubahan fitur atau desain dengan cepat, seperti perubahan dari yang sebelumnya terdapat *dropdown* area menjadi tidak ada *dropdown* area. Sedangkan untuk keurangannya yaitu memerlukan beberapa *plugins* atau beberapa fungsi lain nya dalam pengembangan suatu aplikasi. Tema dari aplikasi yang sedang

dikembangkan juga sering mengalami tabrakan dengan aplikasi lain nya pada fase awal pengembangan.

Literatur ketiga adalah "Implementasi *Black box Testing* dan *Usability Testing* Pada *Website* Sekolah MI Miftahul Ulum Warugunung Surabaya" (Abdillah, Kurniastuti, Susanto, & Yudianto, 2023) literatur ini membahas tentang melakukan pengujian *website* sekolah pada MI Miftaul Ulum Warugunung Surabaya dengan menggunakan *black box testing* dan *usability testing*. Hasil literatur ini berfokus bagaimana metode *black box testing* dan *usability testing* berpengaruh terhadadap hasil uji *website*. Metode *black box testing* berfokus pada pengujian fungsional sistem. Hasil black box testing menunjukkan dari 30 fitur sistem yang diuji, semua fitur berhasil berjalan sukses sesuai dengan hasil yang diharapkan. Selanjutnya, metode *usability testing* berfokus pada kepuasan pengguna dalam pengalaman menggunakan sistem. Pada metode *usability testing* menggunakan USE *Quesionnaire* yang terdiri dari empat dimensi diantaranya *usefulness* (kegunaan), *ease of use* (kemudahan untuk digunakan), *ease of learning* (kemudahan untuk dipelajari) dan *satisfaction* (kepuasan).

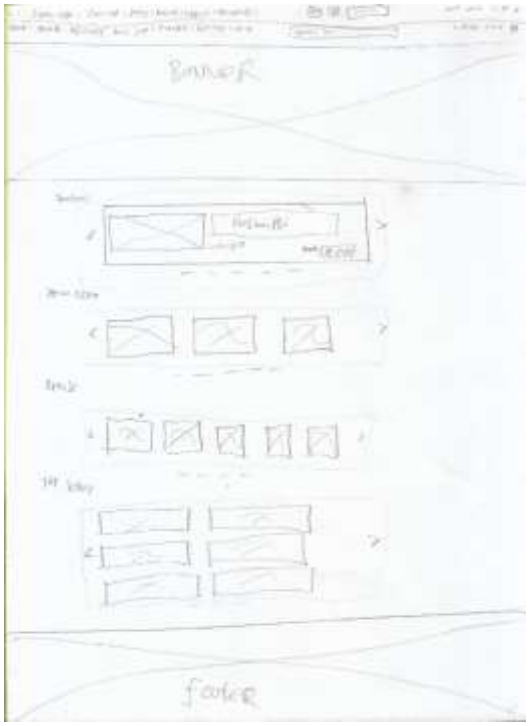
Berdasarkan hasil kajian pustaka penelitian sebelumnya, perbandingan *platfom* bubble.io dengan *platform* pembuat website sejenis dapat diketahui bahwa *platform* bubble.io memiliki beberapa keunggulan seperti dapat mengurangi sekitar waktu membuat sebuah *website* sekitar 80%, memiliki keamanan cukup tinggi dan mampu mengimpor desain website dari aplikasi eksternal seperti Figma.

4.2 Hasil Rancangan

Subbab ini memuat hasil-hasil rancangan pembuatan *website* dengan metode *wireframe* secara manual dengan gambar dikertas.

4.2.1 Rancangan Tampilan Home

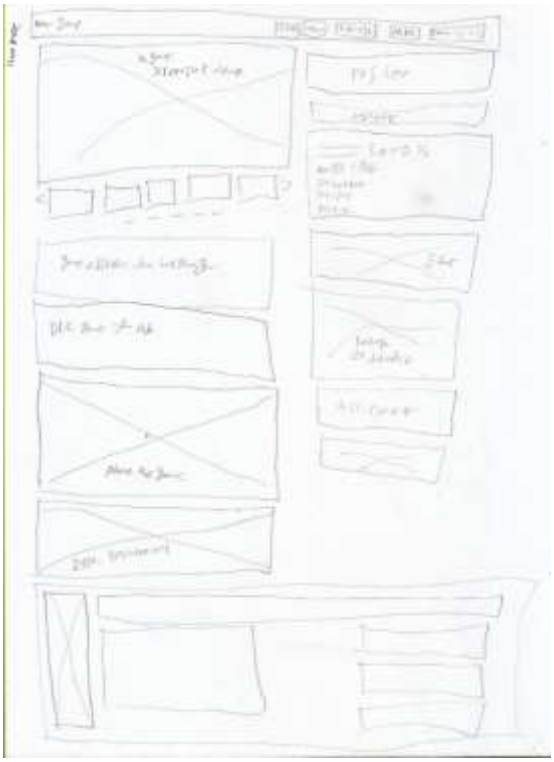
Hasil rancangan tampilan *Home* dapat dilihat pada gambar 4.1. Setiap section memiliki fungsi yang berbeda-beda. *Section* pertama berfungsi sebagai tombol navigasi kepada pengguna jika ingin ke bagian halaman mana, dengan section selanjutnya adalah *banner*, *featured, special offer*, *browse* dan *top seller*, bertujuan untuk mengenalkan kepada pengguna bahwa websitenya ini menyediakan berbagai kategori dan penawaran Istimewa kepada gim-gim yang ada di website. Selanjutnya *section* di bawah halaman adalah *footer* yang berisi *link* untuk menavigasikan pengguna.



Gambar 4.1 Hasil Rancangan Tampilan *Home*

4.2.2 Rancangan Tampilan Store

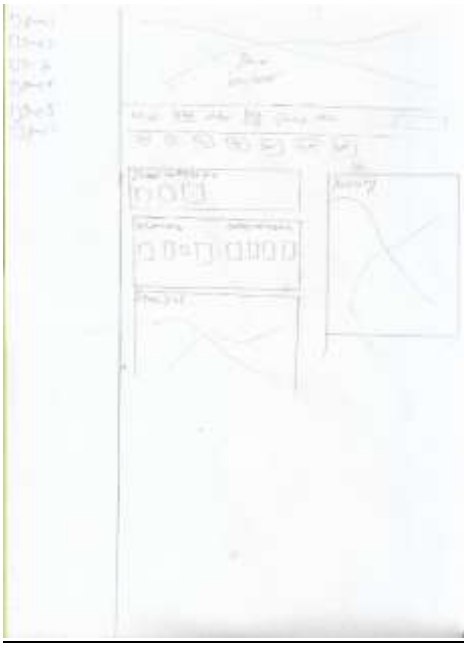
Hasil rancangan tampilan halaman *Store* dapat dilihat pada gambar 4.2. Halaman ini memiliki *section* di kiri atas untuk melihat tangkapan layar dalam gim, serta dibawahnya memiliki *section* harga game yang ditawarkan beserta konten tambahan nya jika tersedia dan juga memiliki *section* minimal spesifikasi perangkat yang diperlukan. Disebelah kanan terdapat beberapa *section* seperti poster gim, deskprisi gim, rating gim yang diberikan oleh para pengguna yang telah membeli gim, bahasa yang didukung, serta pada *section* bawah halaman merupakan tempat ulasan para pengguna yang telah membeli dan memainkan gim nya.



Gambar 4.2 Hasil Rancangan Tampilan *Store*

4.2.3 Rancangan Tampilan Library

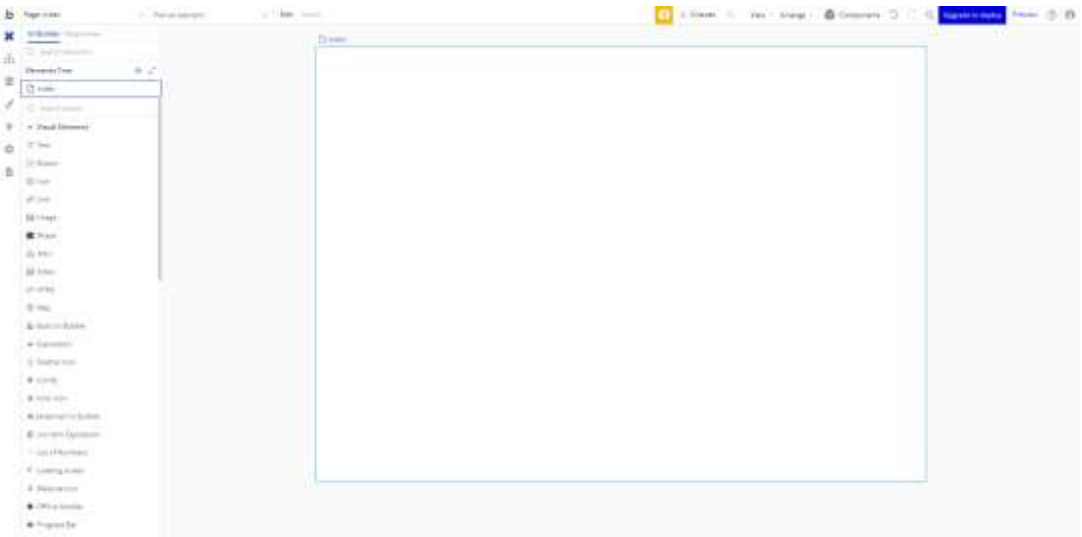
Hasil rancangan tampilan halaman *library* dapat dilihat dalam gambar 4.3. Halaman ini memiliki *section* sebelah kiri untuk melihat gim apa saja yang sudah dibeli, *section* sebelah kanan terdapat beberapa seperti bagian teman punya gim yang sama dengan kita, pencapaian yang sudah didapatkan dan belum didapatkan, tangkapan layar yang dibuat oleh komunitas dan aktivitas yang didapat oleh gim tersebut.



Gambar 4.3 Hasil Rancangan Tampilan *Library*

4.3 Hasil Ekplorasi

Subbab ini memuat hasil pembuatan yang telah dilakukan menggunakan *platform* Bubble.io. Eksplorasi *platform* ini dilakukan dengan membuat terjemahan *wireframe website* yang telah dibuat sebelumnya. *Website* ini berukuran standar desktop dengan lebar 1920 × 1080 *pixel*. Tampilan utama dari *platform* ini saat akan mendesain *website* terdapat pada Gambar 4.4.



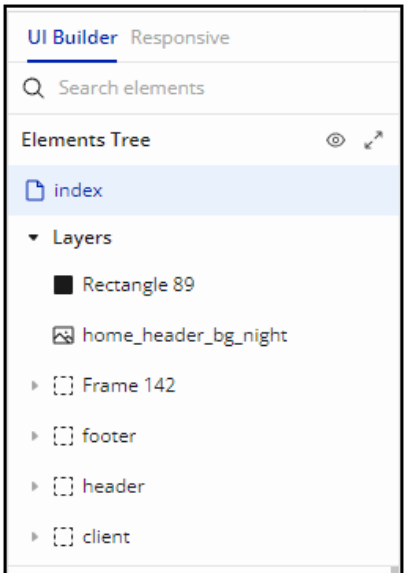
Gambar 4.4 Tampilan Utama *Bubble.io*

Dalam Gambar 4.4 terdapat halaman yang kosong untuk memulai sebuah desain, dimana memiliki ukuran *default* yaitu 1080 × 767 pixel yang dimana ukurannya bisa diubah sesuai dengan keinginan *user*. Pada bar bagian diatas terdapat fitur-fitur seperti *page,, pick an element*, edit, *bug alert*, *view*, *arrange*, *components*, *undo*, *redo*, *search tool*, *upgrade to deploy*, serta *preview*. Isi dari *page* adalah halaman-halaman *website* yang kita buat, elemen untuk hal-hal seperti *text*, gambar dll. Tombol *preview* digunakan untuk memunculkan pratinjau website yang digunakan. Pada bagian sebelah kiri terdapat menu desain, *workflow*, data, *styles*, *plugins*, pengaturan dan *logs*. Tombol desain untuk dimana para pengambang membuat dan mendesain sebuah website, menu *workflow* digunakan untuk memberi arah mau kehalaman mana sebuah tombol di website, menu data digunakan pada saat kita ingin memasukan sebuah data ke website seperti email, tanggal dll, menu *styles* digunakan untuk memodifikasi desain *website* yang pengembang ingin kan seperti warna, jenis huruf yang digunakan pada saat mendesain website, menu *plugins* untuk menambahkan fitur-fitur yang dibuat oleh komunitas yang tidak ada di dalam Bubble itu sendiri, menu pengaturan digunakan pada saat ingin mengatur berbagai macam hal seperti bahasa yang digunakan, versi Bubble itu sendiri dan mengimpor desain dari figma.

Fitur desain dari *platfrom* Bubble.io yang digunakan dalam eksplorisasi ini antara lain:

a. *UI Builder*

   UI builder dalam pengembangan ini digunakan untuk melihat elemen-elemen yang sudah disediakan oleh yaitu *element tree* dan *visual element*. *Element tree* berfungsi sebagai navigasi elemen apa saja yang sudah digunakan dalam membuat sebuah website, *visual element* berisi elemen-elemen yang bisa digunakan untuk membuat sebuah *website.* Yang dimana tinggal *drag and drop* saja elemen-elemen yang ada kedalam area *workspace* Bubble.

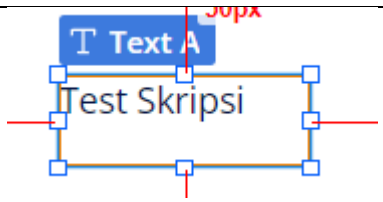


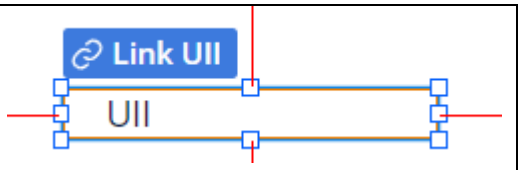
Gambar 4.5 *Element Tree*



Gambar 4.6 *Visual Element*

Tabel 4.1 Fitur dalam platform Bubble

No.	Fitur	Fungsi	Hasil
1	Text	Menambahkan text didalam Bubbble	

2	Link	Menambahkan URL untuk mengarah ke website yang ingin dituju	

b. *Workflow*

Workflow digunakan untuk menavigasi penggunkan ke halaman selanjutnya yang ingin dipilih. Dalam *workflow* ini cukup memilih elemen yang ini dibuat menjadi tombol buat kehalaman selanjutnya yang diinginkan.



Gambar 4.7 *Workflow*

c. *Design Import*

Design import dalam *platform* ini memiliki fungsi untuk mengimpor desain yang sudah kita buat dari platform figma ke bubble.io. Pengembang bisa mengimpor desain website yang sudah dibuat di figma dengan masukkan *API key* dan id file figma yang ada di *url*, fitur ini merupakan fitur resmi yang di hadirkan bubble dengan kerja sama langsung oleh figma sendiri. Pengimporan dari figma ke Bubble membutuhkan waktu Kurang lebih 3 jam (Tergantung seberapa rumit dan banyaknya halaman yang dibuat) namun ada beberapa fungsi yang harus di tambahkan dan disesuaikan kembali seperti untuk bisa menekan dan melakukan sebuah perintah pengembang harus menambah fungsi tersebut, lalu tidak semua gambar bisa tersalin karena hanya gambar dari penyimpanan perangkat internal yang tersalin dan memperbaiki semua bug yang ada. Semua ini hanya waktu kurang lebih 1 sampai 2 jam. Penggunaan platfom Bubble dalam pembuatan

prototyping website *redesign* steam hanya memerlukan waktu kurang lebih 5 jam. Jauh lebih hemat waktu dibandingkan dengan pengkodean tradisional.



Gambar 4.8 *Design Import*

d. *Styles*

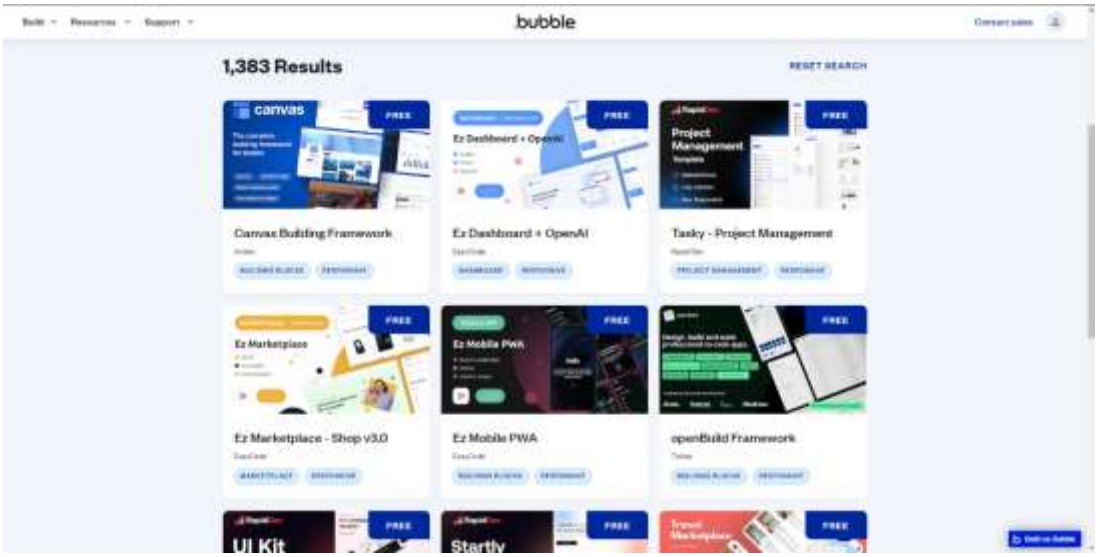
Styles dalam *platform* ini digunakan untuk mengubah gaya elemen yang dipilih, hal-hal yang bisa diubah seperti warna, ukuran, gaya latar belakang, font dll. Pengguna juga bisa menambah font yang tidak tersedia di platform Bubble.



Gambar 4.9 *Styles*

e. Templates

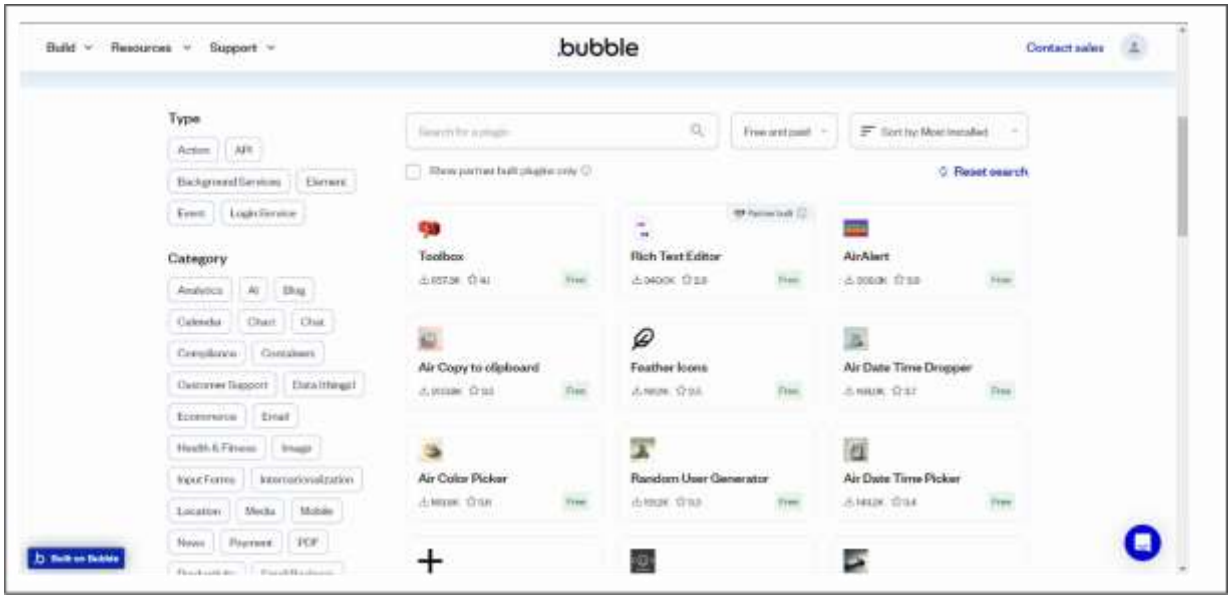
Templates merupakan fitur yang disediakan oleh platform Bubble untuk membantu mempercepat dan mempermudah proses pembuatan dengan memberikan kerangka yang sudah terstruktur. *Template* yang ada didalam platform Bubble ada 2 tipe yaitu gratis dan berbayar, *Templates* yang berbayar di hargai dari 3 sampai 500 USD. Pengembang juga bisa mensortir berbagai macam hal yang pengembang inginkan seperti kategori, harga yang dingin kan, tingkat pengalaman dan rating *Template*.



Gambar 4.10 *Templates*

f. Plugin

Plugin berfungsi situs yang ingin dikembangkan lebih fungsional dengan menghadirkan beberapa fitur pendukung yang sekiranya dibutuhkan saat diakses. Penerapan *plugin* pun dapat disesuaikan dengan kebutuhan. *Plugin* yang ada di dalam Bubble disediakan oleh komunitas dan dari Bubble langsung, pengembang dapat menemukan berbagai macam *plugin* dengan berbagai macam tipe, kategori, harga dan rating yang diberikan.



Gambar 4.11 *Plugin*

4.3.1 Tampilan Halaman Home

Tampilan dari halaman *Home* dalam website Steam berisi *banner event* diskon apa yang sedang berjalan, serta rekomendasi gim-gim yang diberikan oleh *website* dalam gambar 4.9.



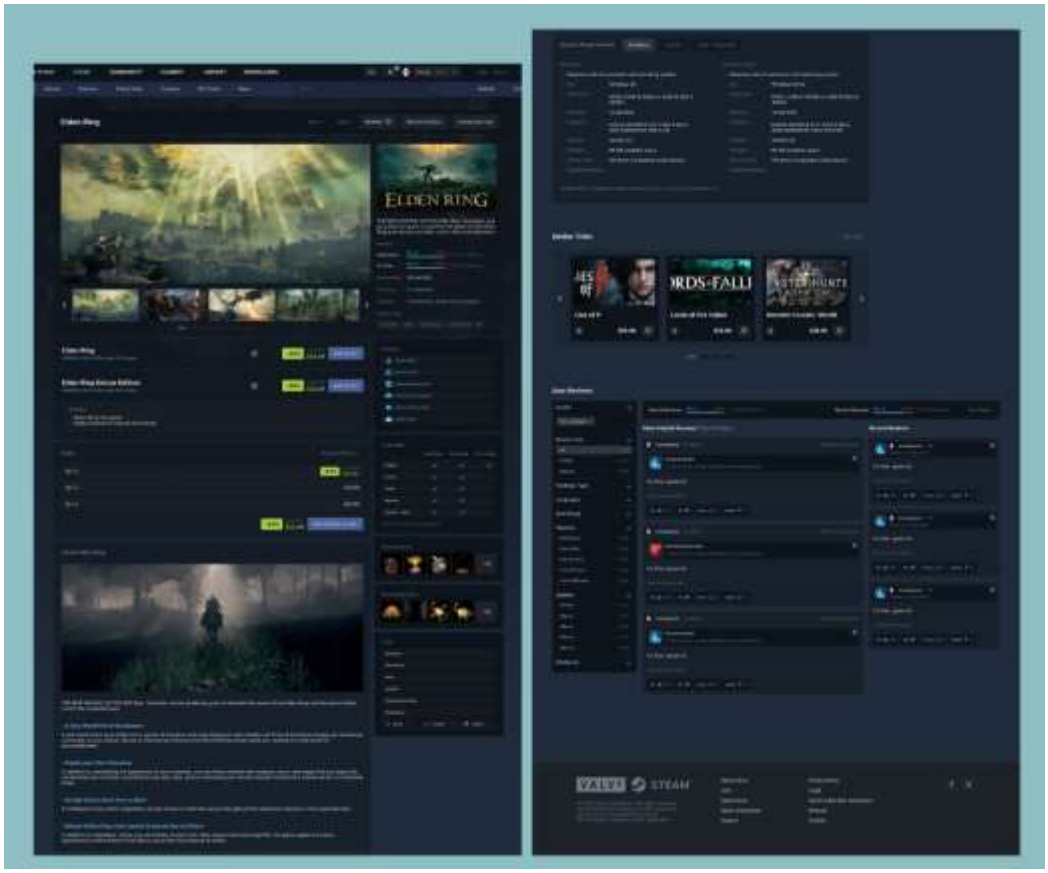
Gambar 4.12 Hasil dari *prototype* tampilan *home*

Halaman *Home* ini menampilkan berbagai macam gim yang dijual dalam berbagai kategori yang sudah disesuaikan oleh referensi masing-masing pengguna dan dibagi . Di bagian *header* terdapat berbagai macam fitur utama yang akan sering digunakan oleh para pengguna. Dalam bagian *footer* menampilkan berbagai layanan yang diberikan oleh website, serta opsi refund gim yang merasa gim yang telah dibeli oleh pengguna tidak cocok dan memiliki opsi hubungan dengan website seperti akun media sosial Facebook, X dll. Bagian *header* dan *footer* tersedia disemua halaman website ini.

4.3.2 Tampilan Halaman Store

Pada halaman *Store* berisi pilihan harga gim yang dijual, dengan pilihan konten tambahan yang bisa dibeli jika pengembang gim nya juga menjual konten tambahan. Dibagian halaman

Store juga memiliki berbgai macam fitur yang bisa menjadi alat jika ingin membeli game tersebut seperti *reviews* yang diberikan oleh para pengguna lainnya yang sudah membeli dan mencoba gim tersebut. Halaman ini memberikan minimal dan rekomendasi spesifikasi komputer supaya lancar pada saat dimainkan dan pengguna tidak menyesal saat membeli gim tersebut. Hasil tampilan dapat dilihat pada Gambar 4.10



Gambar 4.13 Hasil dari *prototype* tampilan *store*

4.3.3 Tampilan Halaman Browse

Tampilan halaman *Browse* menampilkan berbagai macam gim yang ingin dicari oleh para pengguna, pada halaman ini pengguna bisa mencari gim yang dicari dengan menggunakan *search bar* yang sudah disediakan, pengguna bisa mensortir gim yang dicari bedasarkan genre, harga, penjualan teratas dan populer. Hasil tampilan dapat dilihat pada gambar 4.11



Gambar 4.14 Hasil dari *prototype* tampilan *Browse*

4.3.4 Tampilan Halaman Library

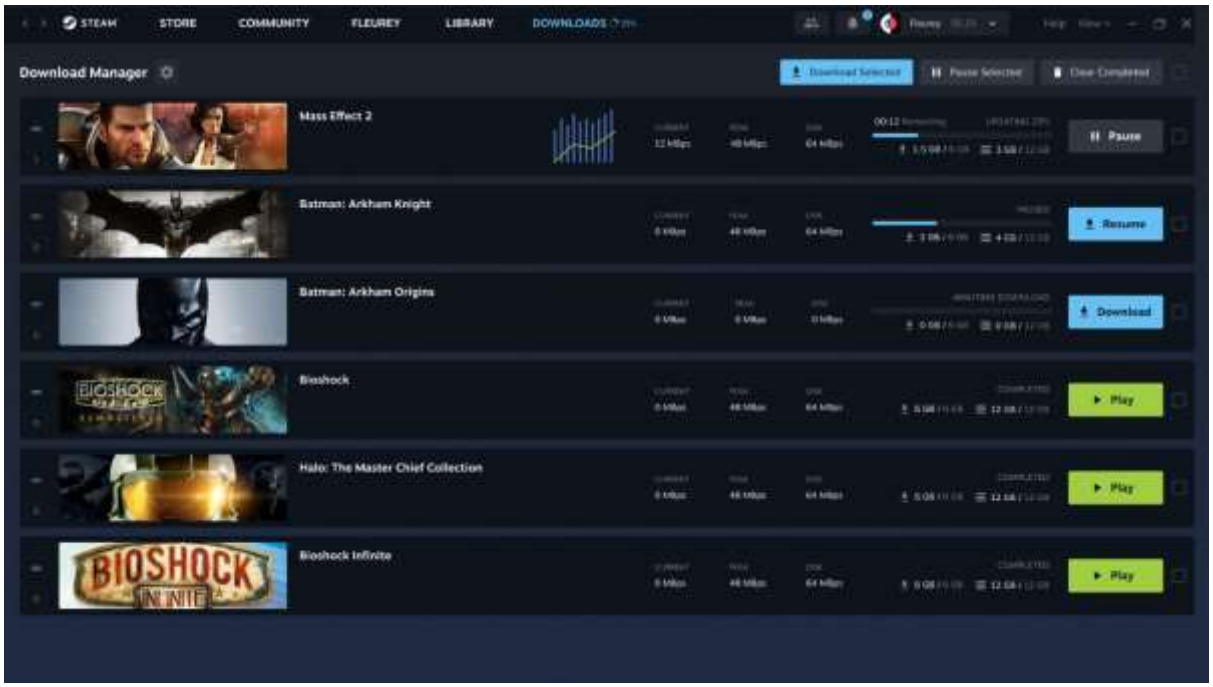
Halaman ini menampilkan semua gim yang telah dibeli oleh pengguna, pengguna bisa menyortir dan mengelompokkan gim yang telah dibeli sesuai dengan preferensi masing-masing pengguna. Hasil tampilan halaman dapat dilihat pada gambar 4.12



Gambar 4.15 Hasil dari *prototype* tampilan *Library*

4.3.5 Tampilan Halaman Download

Halaman ini menampilkan semua gim yang sedang di *download*, pengguna juga bisa melihat berapa kecepatan *dowmload*, ukuran gim dan berapa banyak waktu yang tersisa higga *download* selesai.



Gambar 4.16 Hasil dari *prototype* tampilan *Download*

4.4 Hasil Pengujian

Pengujian pada penelitian ini menggunakan metode *Black box testing* dan *Usability testing*

Tabel 4.2 Hasil Pengujian *Black box Testing*

No.	Fitur yang diuji	Skenario	Hasil yang diharapkan	Hasil Pengujian
1.	Melihat halaman *Home*	Pengguna klik menu *Home*	Pengguna berhasil masuk ke halaman *Home*	Sukses
2.	Melihat halaman *Store*	Pengguna klik menu *Buy Now*	Pengguna berhasil masuk ke halaman *Store*	Sukses
3.	Melihat halaman *Browse*	Pengguna klik menu *Home*, klik menu *Browse*	Pengguna berhasil masuk ke halaman *Browse*	Sukses
4.	Melihat halaman *Library*	Pengguna klik menu *Library*	Pengguna berhasil masuk ke halaman *Library*	Sukses
5.	Mengganti *Grid* koleksi *Library*	Pengguna klik menu *Library*, klik *icon grid* disebelah *search bar*	Pengguna berhsil mengganti *grid Library*	Sukses
6.	Melihat halaman *Download*	Pengguna klik menu *Download*	Pengguna berhasil masuk ke halaman *Download*	Sukses

Tabel 4.3 Skor Pengujian *Usability Testing*

Respon	**Jumlah**	**Skor**	**Jumlah × Simbol**
Sangat Setuju	91	5	455
Setuju	88	4	352
Netral	33	3	99
Tidak Setuju	8	2	16
Setuju	0	1	0
Total Skor			**922**

Nilai *usability* ditentukan dengan menggunakan perhitungan $U = \frac{P}{Q}$; $Q = R \text{ x } S \text{ x } 5$, dimana U adalah nilai usability; P adalah skor total; Q adalah skor maksimal; R adalah jumlah semua responden yang menjawab sangat setuju; dan S adalah jumlah soal. Hasil perhitungan *usability* berdasarkan data kuesioner adalah sebagai berikut:

$$\text{Skor maksimal (Q)} = R \text{ x } S \text{ x } 5$$

$$= 12 \text{ x } 20 \text{ x } 5$$

$$= 1200$$

$$\text{Nilai usability (U)} = \frac{P}{Q}$$

$$= \frac{922}{1200}$$

$$= 0{,}77$$

Dengan nilai usability sebesar 0,77 artinya nilai tersebut mendekati nilai 1 termasuk dalam baik, karena semakin mendekati 1 maka nilai semakin baik. Hal ini menunjukan bahwa pengguna secara umum dapat menggunakannya dengan cukup mudah dan nyaman. (Fitrisia & Fadly, 2017)

4.5 Refleksi Diri Terhadap Platform Bubble

Platform Bubble ini memudahkan penulis untuk membuat sebuah *website* dengan UI dan UX yang baik, serta dengan bantuan *template* dan *Design import* pada bubble, dapat memudahkan ketika membuat desain yang cocok dan sesuai dengan *website* yang dibuat. Kelebihan dari fitur *template* dan *Design import* yaitu bersifat gratis, cepat dan memiliki

berbagai macam desain yang di berikan. Kelebihan lain adalah UI platform yang intuitif, penulis tidak perlu belajar lebih lama untuk memahami setiap fitur dasar dari *platform* Bubble, serta fitur dasar yang menurut penulis sangat esensial untuk mendesain sebuah *website*. Kelebihan selanjutnya hasil *website* yang dibuat menggunakan *platform* Bubble memiki hasil yang hampir mendekati dengan pengkodean tradisional dengan beberapa kelemahan seperti *website* yang sedikit kurang responsif dibandingkan dengan *website* pengkodean tradisional.

Namun *platform* ini memiliki kelemahan, beberapa kelemahannya yaitu beberapa fitur yang ada di bubble di bisa diakses jika kita tidak berlangganan dengan biaya dari USD 32 (Rp. 517,946.80, mengikuti kurs pada 23 juli 2024) sampai USD 399 (Rp. 6,459,289.77, Mengikuti kurs pada 23 juli 2024) per bulan ketika ingin mendapatkan pengalaman mendesain dan membuat *website* secara penuh dan tanpa batas, sehingga karena masih menggunakan dollar bukan rupiah pengembang yang memiliki dana yang terbatas akan mengalami kesusahan dalam menggunakan *platform* Bubble tidak bisa mendesain dan membuat website secara penuh dan tanpa batas. Kelemahan kedua, pada fitur *Design import* tidak bisa mengimpor 100 persen desain yang sudah dibuat di *platform* figma seperti efek blur dan gambar yang harus menggunakan file internal yang ada didalam laptop atau komputer. Kelemahan ketiga yaitu pembayaran yang terbilang cukup ribet karena *platform* bubble hanya mendukung kartu pembayaran internasional, seperti jenius dan paypal yang mana menyulitkan banyak pengembang yang ingin menggunakan 100 persen fitur yang ada didalam *platfom* Bubble.

Dari pengalaman dalam menggunakan *platform* Bubble cukup memuaskan, karena secara desain tata letak ruang kerja dan fungsi desain yang mirip dengan *platform* Figma yang bisa dibilang cukup familiar, hanya saja di platform Bubble lebih rumit dan lebih berat secara operasional sehingga memerlukan sedikit waktu untuk memahami *platform* ini, sehingga membuat pengembang mudah membuat sebuah aplikasi dan waktu membuat sebuah *website* yang awalnya membutuhkan sekitar 3 sampai 6 bulan bisa terpotong sebanyak 80% yang hanya membutuhkan sekitar 1 sampai 1,5 bulan saja. Namun, ada beberapa tantangan utama dalam eksplorasi *platform* Bubble diantaranya *plarform* ini sangat memerlukan akses internet, jika pada saat mendesain dan membuat sebuah *website* di Bubble maka proses tersebut akan terhenti sampai ada akses internet. *Platform* Bubble sangat cocok untuk berbagai macam pengembangan perangkat lunak seperti *E-commerce, CRM System, Marketplace* dll

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil eksplorasi platform bubble.io, penulis memberikan kesimpulan :

1. Untuk membuat sebuah website di platform bubble.io, pengembang dapat memanfaatkan fitur-fitur di dalam platform yang sudah di sediakan langsung oleh bubble.io atau komunitas seperti template halaman, *plugin*, *marketplace*, forum, dan dapat mengkreasikan *layout* sendiri.
2. Platfom ini memudahkan para pengembang pemula untuk membuat website dengan mudah tanpa harus memiliki pengetahuan dasar tentang bahasa dan aturan dalam pengkodean. Bubble.io memiliki fitur *design import* dari figma yang membantu dalam mendesain suatu website jika pengembang sudah memiliki desain website yang sudah dikerjakan pada platformm figma sehingga pengembang tinggal mengimport desain tersebut kedalam platform bubble akan tetapi pada saat mengimport desain dari figma ke bubble akan ada beberapa masalah desain yang harus dibenar kan di dalam bubble tersebut. Untuk mendapatkan pengalaman membuat website secara penuh dan membuka beberapa fitur yang dikunci, pengembang dapat membayar langganan dengan harga yang sudah ditetapkan oleh platfrom ini.
3. Berdasarakan hasil pengujian *redesign* website Steam yang dibuat dari platform figma lalu di impor ke platform bubble.io untuk dipoles lebih lanjut, website ini secara kesuluruhan memiliki tingkat kepuasan yang cukup, cukup *user-friendly* dan cukup responsif dengan Nilai sebesar 0,77 sehingga dapat diluncurkan kepada pengguna.
4. Platform Bubble.io menjadi salah satu solusi alternatif dalam membuat prototype website yang responsif dan mudah dengan minim pengkodean sehingga para pengembang pemula yang tidak memiliki pemahaman dasar pengkodean dapat membuat website yang mudah. Fitur *import design* bisa menjadi salah satu fitur yang sangat membantu pengembang untuk membuat sebuah website.

5.2 Saran

Saran dari penelitian ini adalah untuk penelitian selanjutnya diharapkan membandingkan antara Bubble.io dengan platform pembuat website lainnya. Dan mengeksplorasi lebih berbagai macam fitur yang ada didalam Bubble.io

DAFTAR PUSTAKA

Abdillah, M. T., Kurniastuti, I., Susanto, F. A., & Yudianto, F. (2023). Implementasi Black box Testing dan Usability Testing pada Website MI Miftahul Ulum Warugunung SUrabaya. *Jurnal Ilmu Komputer dan Desain Komunikasi Visual, 8*(1), 234-242.

Abednego, A., Suyoto, & Julianto, E. (2021). Pengembangan Aplikasi Layanan Multiguna Menggunakan Low-Code Platform (Studi Kasus: Astra Credit Companies). *Jurnal Informatika Atma Jogja, 2*(1), 10-19.

Akbar, G. M., & Idris, M. (2022). Penerapan Low-Code Platform dalam Pengembangan Aplikasi Presensi (Studi Kasus PT. Astra Sedaya Finance). *AUTOMATA, 3*(2), 121-128.

Asriani. (2011, Juli - September). Pemanfaatan Internet Marketing Dalam Pemasaran Produk Unggulan Pertanian Provinsi Gorontalo. *Komunikasi KAREBA, 1*(3), 9.

Chandramita, C. (2019). *Metode Prototyping dalam Pengembangan Sistem Informasi.* Retrieved from Academia: https://www.academia.edu/10561240/Metode_Prototyping_Dalam_Pengembangan_Si

Destiningrum, M., & Adrian, Q. J. (2017). Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi kasus: Rumah Sakit Yukum Medical Centre). *TEKNOINFO, 2*(11), 30-37.

Ericksson, W. (2020). *Could I Build My Product on Bubble.io?* The startup. Retrieved from https://medium.com/swlh/could-i-build-my-product-on-bubble-io-75923a41b9b6

Fitrisia, Y., & Fadly, M. (2017). Evaluasi Functionality dan Usability External Quality. *Jurnal Komputer Terapan, 3*(2), 193-202 .

Groc, T. (2020). *What You (Really) Can/Can't Do With Bubble*. Retrieved from Medium: https://blog.cube.fr/what-you-really-can-cant-do-with-bubble-io-f20428b5d3c5

Hesti Wahyuningtyas, I. S. (2022, Desember 2). Evaluasi Usability pada SIMPEL SMAN 12 . *Jurnal Prodi Teknik Informatika UNW "Multimatrix", 5*(2), 10-28.

Husna, H. I., & Nafisah, S. (2019). Analisis Elemen Kunci Website Berdasarkan Konsep Shedroff pada Website Perpustakaan Universitas Islam Indonesia. *Jurnal Perpustakaan, 10*(1), 1-10.

Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing boundary Value Analysis. *Jurnal Informatika : Jurnal Pengembangan IT (JPIT), 3*, 45-48.

Mauko, I. C., Setiohardjo, N. M., & Noach, F. P. (2017). Pengembangan Website Unit Penelitian dan Pengebadian Kepada Masyarakat dan Penerapan Jurnal Elektronik Berbasis Open Source di Politeknik Negeri Kupang. *Jurnal Ilmiah FLASH, 3*(2), 100-108. doi:10.32511/flash.v3i2.145

Muhtadi, A. Z., & Zuanedi, L. (2021). Implementasi Metode Prototype dalam Membangun Sistem Informasi Penjualan Online pada Toko Herbal Pahlawan. *Journal of Advances in Information and Industrial Technology (JAIIT), 3*(1), 31-41.

Prasetyo, H. (2017). *Prototyping untuk Analisis Kebutuhan dan Perancangan Sistem Monitoring Mentoring, Skripsi.* Yogyakarta: Universitas Islam Indonesia.

Rahadi, D. R. (2014). Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Andorid. *Jurnal Sistem Informasi (JSI)*, 661-671.

Rahmasari, E. A., & Haryadi, T. (2022). Kajian Interaction Design Website Museum Ranggawarsita Semarang. *DesainKomunikasi Visual*, *9*(2), 129-128.

rajgor, V. V., & Marly, S. (2022). Low Code/ No code Development Platform Compared to Traditional Development features Uses and Future. *International Research Journal of Modernization in Engineering Technology and Science*, 514-518.

Sasongko, A., Jayanti, W. E., & Risdiansyah, D. (2020). Use Questionnaire untuk Mengukur daya Guna Sistem Informasi E-Tadkzikirah. *Jurnal Khatulistiwa Informatika, 8*(2), 80-87.

Sommervile, I. (2011). *Software Egineering (Rekayasay Perangkat Lunak) Jilid 1*. Jakarta: Erlangga.

Threestayanti, L. (2019). *Apa itu Platform Low-Code development*. Retrieved from https://infokomputer.grid.id/read/121768787/apa-itu-platform-low-code-development

Virta, T. (2018). Relation of low-code development to standard software development: Case Biit Oy.

LAMPIRAN

Pertanyaan	Res 1	Res 2	Res 3	Res 4	Res 5	Res 6	Res 7	Res 8	Res 9	Res 10	Res 11
P 1	5	5	5	5	4	3	4	4	4	5	4
P 2	5	5	5	5	5	3	5	4	4	4	4
P 3	5	5	5	5	3	5	4	3	4	5	3
P 4	4	4	5	5	5	4	5	4	4	5	4
P 5	5	5	5	5	4	3	4	4	5	4	4
P 6	5	5	5	5	4	2	5	4	5	3	3
P 7	4	4	5	5	4	4	5	4	4	4	3
P 8	5	5	5	5	5	2	5	5	5	5	3
P 9	4	4	5	5	4	3	4	4	4	5	3
P 10	4	4	5	5	5	5	5	4	4	2	4
P 11	5	5	5	5	4	4	5	4	4	5	3
P 12	4	4	5	5	4	3	5	4	4	4	4
P 13	5	5	5	5	4	2	5	5	4	5	4
P 14	4	4	5	5	3	3	2	2	2	4	2
P 15	4	4	5	5	4	3	4	3	5	4	4
P 16	4	4	5	5	4	5	5	4	4	3	4
P 17	5	5	5	5	3	5	5	4	4	5	3
P 18	5	5	5	5	4	4	5	3	3	5	4
P 19	5	5	5	5	4	4	5	4	4	4	3
P 20	5	5	5	5	5	5	5	4	4	4	4