

PERANCANGAN APLIKASI MULTIMEDIA INTERAKTIF BELAJAR BERHITUNG UNTUK USIA BALITA BERBASIS WEB

Aswar Hanif
AMIK BSI Jakarta

aswar.awf@bsi.ac.id

Abstrak – Proses perkembangan otak manusia, 90% terjadi pada 5 tahun pertama hidupnya. Ini berarti bahwa masa balita adalah penting bagi perkembangan otak. Proses perkembangan otak dalam tiap tahap dipengaruhi oleh faktor-faktor penting, termasuk di dalamnya adalah respon yang diterima dan aktivitas. Permainan berhitung akan berhasil jika anak diberikan kesempatan untuk berpartisipasi dan distimulasi untuk menyelesaikan soal dan masalah. Penggunaan teknologi, seperti aplikasi interaktif, dapat membantu seorang anak untuk belajar berhitung lebih mudah. Dalam aplikasi ini, seorang balita dapat belajar berhitung menggunakan warna, gambar, dan suara, membuat proses belajar lebih menarik dan menyenangkan. Aplikasi ini juga dirancang untuk menaikkan tingkat kesulitan dengan lambat dan bertahap, sehingga si anak tidak akan kesulitan untuk mengikuti. Proses pengembangan aplikasi dilakukan menggunakan Software Development Life Cycle (SDLC) yang menyediakan aturan dan proses, mulai dari tahap perencanaan dan analisis hingga tahap pengujian, untuk memastikan proyek pengembangan aplikasi dapat menghasilkan produk yang diinginkan. Keuntungan aplikasi tidak hanya dirasakan oleh anak-anak, tetapi juga oleh orangtua yang menginginkan anak-anak mereka untuk bermain sekaligus belajar.

Kata Kunci: aplikasi permainan, balita, berhitung

I. PENDAHULUAN

Proses perkembangan otak seseorang, 90% terjadi pada 5 tahun pertama hidupnya. Ini berarti masa balita menjadi sangat penting bagi perkembangan otak. Proses perkembangan otak di setiap fase dipengaruhi oleh faktor-faktor penting, yang meliputi pengalaman sehari-hari, respons yang diterima, asupan nutrisi, dan aktivitas. Tahapan perkembangan kemampuan membaca, menulis, dan berhitung (*calistung*) bukanlah keterampilan yang dapat begitu saja dikuasai anak. Melihat gambar adalah bentuk membaca yang paling sederhana. Balita usia 3-5 tahun diharapkan sudah memiliki ketertarikan untuk “membaca” gambar, simbol, dan logo yang ada disekitarnya. Untuk itu salah satunya anak membutuhkan *exposure* yang tinggi pada buku bergambar.

Keberadaan buku bergambar dan seorang pembimbing adalah utama dalam pembelajaran balita. Permasalahannya adalah bahwa buku rawan mengalami kerusakan dan penurunan kualitas gambar dengan pemakaian terus menerus, dan keberadaan seorang pembimbing, bisa orangtua maupun guru dalam dalam pendidikan usia dini, yang tidak bisa menemani setiap saat, membatasi kegiatan belajar seorang anak. Penggunaan teknologi, seperti aplikasi *multimedia* interaktif, dapat membantu seorang anak untuk belajar lebih nyaman dan mudah.

Menggunakan permainan untuk belajar telah lama digunakan secara efektif. Permainan berhitung akan berhasil jika anak diberikan kesempatan untuk berpartisipasi dan distimulasi untuk menyelesaikan masalah mereka sendiri. Permainan juga akan berhasil jika tingkat kesulitan disesuaikan dengan target *audience* penggunanya.

Dengan adanya aplikasi *multimedia* interaktif berbasis *web*, diharapkan guru atau orangtua dengan anak balita bisa mendapatkan sebuah sarana untuk anak bermain sekaligus belajar. Penggunaan basis *web* diharapkan dapat memberikan keluasaan akses, karena *web browser* yang tersedia di banyak alat elektronik yang mengakses *internet*. Ini juga dapat dijadikan sebagai bagian dari *website* profil organisasi seperti PAUD (Pendidikan Anak Usia Dini), atau bisa menjadi salah satu tautan, yang diharapkan akan memberikan nilai tambah yang berguna bagi organisasi, penggunanya, dan orangtua yang memiliki anak usia balita secara umum.

II. LANDASAN TEORI

2.1. HTML

HTML merupakan singkatan dari *HyperText Markup Language*. HTML digunakan untuk membangun suatu halaman *web*. Sekalipun banyak orang menyebutkan sebagai suatu bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman, karena seperti tercermin dari namanya, HTML adalah suatu bahasa *mark up*. HTML digunakan untuk melaksanakan *mark up* (penandaan) terhadap sebuah dokumen teks. Tanda tersebut digunakan untuk menentukan format atau *style* dari teks yang ditandai. (Levine, 2000)

2.2. HTML5

HTML5 adalah bahasa *markup* yang digunakan untuk penataan dan konten menyajikan di *World Wide Web*. Ini adalah versi kelima dan saat ini adalah standar HTML.

Diterbitkan bulan Oktober 2014 oleh *World Wide Web Consortium (W3C)* untuk meningkatkan bahasa dengan dukungan untuk *multimedia* terkini,

sekaligus menjaga untuk mudah dibaca oleh manusia dan secara konsisten dipahami oleh komputer dan perangkat seperti *web browser*, *parser*, dan lain-lain. *HTML5* dimaksudkan tidak hanya untuk menggantikan *HTML 4*, tetapi juga *XHTML 1* dan *DOM level 2 HTML*.

HTML5 termasuk termasuk didalamnya, model pengolahan rinci untuk mendorong implementasi yang lebih *interoperable*. *HTML5* memperluas, meningkatkan dan merasionalisasi *markup* yang tersedia untuk dokumen, dan memperkenalkan *markup* dan memperkenalkan aplikasi pemrograman antarmuka (API) untuk aplikasi *web* yang kompleks. Untuk alasan yang sama, *HTML5* juga merupakan calon untuk aplikasi *mobile cross-platform*, karena memiliki fitur yang dirancang dengan memikirkan perangkat bertenaga rendah.

Banyak fitur sintaksis baru disertakan. Untuk secara *native* memasukkan dan menangani *multimedia* dan konten grafis, elemen baru seperti `<video>`, `<audio>` dan `<canvas>` ditambahkan, dan dukungan untuk *scalable vector graphics (SVG)* konten dan formula *MathML*. Untuk memperkaya konten semantik dokumen, elemen struktur halaman baru seperti `<utama>` `<section>`, `<article>`, `<header>`, `<footer>`, `<aside>`, `<nav>` dan `<figure>`, ditambahkan. Atribut baru diperkenalkan, beberapa elemen dan atribut telah dihapus, dan lain-lain seperti `<a>`, `<cite>` dan `<menu>` telah berubah, didefinisikan ulang atau standar. (Kesteren & Pieters, 2011)

2.3. CSS

CSS singkatan dari *Cascading Style Sheets* dan merupakan Bahasa yang terpisah, namun saling melengkapi, *HTML*. CSS adalah apa yang kita gunakan untuk menerapkan presentasi (*style*) terhadap konten pada halaman *web*. (Lazaris, 2013)

Bersama dengan *HTML* dan *JavaScript*, CSS adalah teknologi dasar yang digunakan oleh sebagian besar situs untuk membuat halaman *web* secara visual menarik, *user interface* untuk aplikasi *web*, dan antarmuka pengguna untuk banyak aplikasi *mobile*. CSS dirancang terutama untuk memungkinkan pemisahan konten dokumen dari presentasi dokumen, termasuk aspek-aspek seperti tata letak, warna, dan *font*. Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak fleksibilitas dan kontrol dalam spesifikasi karakteristik presentasi, memungkinkan beberapa halaman *HTML* untuk berbagi format dengan menentukan CSS relevan dalam *file CSS* terpisah, dan mengurangi kompleksitas dan pengulangan dalam konten struktural.

2.4. Bootstrap

Bootstrap merupakan produk *open source* dari Mark Otto dan Jacob Thornton yang saat awalnya dirilis, keduanya adalah karyawan di *Twitter*. Saat itu ada kebutuhan untuk membakukan *toolset frontend* insinyur di perusahaan.

Sejak *Bootstrap* diluncurkan pada Agustus 2011, *bootstrap* telah meroket popularitasnya.

Bootstrap telah berkembang dari menjadi proyek yang seluruhnya digerakkan oleh CSS untuk kemudian memasukkan sejumlah *plugin JavaScript* dan ikon-ikon yang berjalan seiring dengan *form* dan tombol. Pada dasarnya, *bootstrap* memungkinkan perancangan *web responsive* dan memfiturkan desain 12-kolom, 940px-wide *grid* yang *robust*. Salah satu yang menarik adalah build tool di *website Bootstrap*, di mana Anda dapat menyesuaikan membangun sesuai kebutuhan, memilih CSS dan fitur *JavaScript* ingin dimasukkan ke dalam situs Anda. Semua ini memungkinkan pengembangan antarmuka *web* untuk melambung ke depan, membangun dengan pondasi desain dan penegembangan stabil yang melihat ke depan. Memulai *bootstrap* adalah sederhana seperti meletakkan beberapa CSS dan *JavaScript* ke dalam akar situs Anda. (Spurlock, 2013)

2.5. iSpring PowerPoint to Flash and HTML5 Converter

iSpring menambahkan *tab* ekstra untuk *toolbar PowerPoint* untuk akses ke semua fungsinya. *Software* ini mengkonversi Presentasi *PowerPoint (PPT)* ke *Flash* atau *HTML5*. Teknologi *AccuPoint* dan *HyperPoint* yang dikembangkan oleh *iSpring* mengkonversi presentasi *PowerPoint* ke *Flash*, *HTML5*, dan video dengan akurasi tertinggi.

2.6. Prototipe

Sebuah *prototipe* adalah sampel awal, model, atau peluncuran produk dibangun untuk menguji konsep atau proses atau bertindak sebagai sesuatu yang akan direplikasi atau dipelajari. (Blackwell & Manar, 2015) *Prototyping* berfungsi untuk memberikan spesifikasi untuk sebuah sistem bekerja nyata daripada sebuah teori.

2.6. Adobe Photoshop

Ada banyak program aplikasi yang dapat digunakan untuk melakukan manipulasi gambar. *Photoshop* adalah salah satu program pengolah *image (image processing)* yang banyak digunakan untuk membuat dan memanipulasi gambar, baik untuk keperluan percetakan, *internet*, maupun elektronik. Keunggulan dan kekuatan *Photoshop* terletak antara lain pada kemudahan, fasilitas, dan fleksibilitasnya. *Photoshop* adalah salah satu program pengolah *image*, khususnya untuk gambar *bitmap*. Gambar *bitmap* yaitu gambar yang terbentuk dari elemen gambar (*pixel*), dengan setiap *pixel*nya mempunyai warna tertentu.

II III. METODE PENELITIAN

3.1. Software Development Life Cycle (SDLC) Model Waterfall

Model *SDLC Waterfall* adalah proses penembangan *software* secara *sequensial* di mana proses dianggap mengalir ke bawah (mirip dengan air terjun), melalui tahap-tahap yang harus dieksekusi untuk secara sukses membangun sebuah *software*. Model *Waterfall* mendefinisikan beberapa tahap berurutan yang harus diselesaikan sesuai urutannya,

dan hanya meneruskan ke tahap selanjutnya saat tahap yang mendahuluinya telah benar-benar terselesaikan secara tuntas. Pada dasarnya, model Waterfall terdiri dari lima tahap: Analisis, desain, implementai, *testing*, dan *maintenance*. (Bassil, 2012)

1. Analisis

Merupakan proses pengumpulan kebutuhan untuk menentukan spesifikasi *software*. Terdiri dari:

- a. *Functional Requirements*
- b. *Content Requirements*
- c. *Quality Requirements*
- d. *System Environment Requirements*
- e. *User Interface Requirements*
- f. *Evolution Requirements*
- g. *Project Constraint*

2. Desain

Desain perangkat lunak adalah proses multistep yang meliputi:

- a. Desain Antar Muka
- b. Desain Basis Data
- c. Desain Struktur Navigasi

3. Implementasi

Dari desain yang sudah dibuat, diimplementasikan ke perangkat lunak atau pembuatan aplikasi

4. Testing

Setelah implementasi desain ke dalam aplikasi, dilakukan evaluasi apakah aplikasi yang dihasilkan sesuai dengan analisis kebutuhan. Tahap testing bisa dilakukan dengan dua pendekatan:

- *Black-Box Testing*.
Menguji dari spesifikasi fungsional tanpa menguji desain dan kode program.
- *White-Box Testing*.
Menguji dari segi desain dan kode.

5. Maintenance

Dukungan atau pemeliharaan sebuah *web* atau aplikasi dapat dilakukan dengan beberapa cara, seperti:

- a. Melakukan *update* terhadap konten, menyesuaikan dengan informasi yang berubah atau bertambah.
- b. Pemeliharaan *hardware* dan *software* agar selalu berjalan baik.
- c. Melakukan *back-up database secara berkala* untuk menghindari kehilangan data
- d. Melakukan evaluasi terhadap kritik dan saran pengguna

3.2. Bahan Penelitian

Untuk memperoleh bahan penelitian ini dibutuhkan data-data dan literatur-literatur disertai dengan penganalisaan untuk memperoleh hasil yang diinginkan. Untuk itu pengambilan data dalam penulisan ini adalah:

1. Observasi

Dilakukan pengamatan terhadap cara pembimbing (Guru dan Orangtua) dalam mengajarkan anak balita mengenal dan mengingat angka, di PAUD (Pendidikan Anak Usia Dini) maupun di rumah.

2. Studi Pustaka

Mempelajari, meneliti, dan mengumpulkan informasi dari referensi-referensi yang menunjang, baik untuk konten situs *web* maupun teori-teori yang dibutuhkan untuk pengembangan aplikasi secara keseluruhan.

IV. PEMBAHASAN

4.1. Analisis Kebutuhan

1. *Functional Requirements*

Fungsi-fungsi yang dibutuhkan oleh *website* dan aplikasi *multimedia* interaktif untuk balita dijelaskan di tabel 1.

2. *Content Requirements*

a. Halaman Beranda

Halaman ini akan berisi gambar utama, nama situs, dan sedikit teks deskripsi situs. Tautan-tautan ke multimedia dan aplikasi juga akan dimasukkan semua ke badan halaman *home* ini.

b. Halaman Mengenal Angka

Halaman ini akan berisi judul video, *multimedia* berbentuk video, dan sedikit teks deskripsi untuk video.

c. Halaman Mengingat Angka

Halaman ini akan berisi judul aplikasi, aplikasi interaktif untuk mengingat angka.

d. Halaman Menebak angka

Halaman ini akan berisi aplikasi interaktif untuk menebak angka

3. *Quality Requirements*

a. *Usability*

- Navigasi situs harus mudah digunakan oleh pembimbing (guru atau orangtua)
- Aplikasi interaktif harus bisa digunakan oleh anak usia balita

b. *Portability*

- Desain situs *responsive*, sehingga bisa diakses melalui berbagai ukuran resolusi layar
- Aplikasi harus bisa berjalan pada *browser desktop* maupun *mobile*

4. *System Environment Requirements*

Lingkungan yang dibutuhkan oleh *website* dan aplikasi *multimedia* interaktif untuk balita adalah:

a. *Web Server*

Karena aplikasi dan situs tidak memiliki konten dinamis, *web server* tidak memerlukan *software* khusus untuk mengolah *server-side script* dan *database*.

b. *Client hardware requirement*

Desain *responsive* membuat situs bisa dijalankan diberbagai resolusi perangkat. Aplikasi akan berfungsi diberbagai resolusi perangkat, tapi membutuhkan resolusi minimal relatif untuk visibilitas.

- c. *Client software requirement Browser web* yang kompatibel dengan *HTML5*, *CSS3*, dan *Javascript*. Koneksi internet jika dihosting di *internet*, koneksi *LAN* jika di *intranet*, atau bisa dijalankan local.

d. Lain-lain

Tabel 1. Functional Requirements

<i>Requirement ID</i>	<i>Requirement Statement</i>	Kondisi	Keterangan
FR100	Website akan bisa menampilkan daftar multimedia dan aplikasi	Wajib	
FR101	<i>Menampilkan multimedia dan aplikasi berdasarkan kategori</i>	Wajib	
FR102	<i>Menampilkan multimedia dan aplikasi berdasarkan yang paling populer</i>	Optional	
FR103	<i>Menampilkan multimedia dan aplikasi berdasarkan angka yang difiturkan</i>	Optional	
FR200	Konten multimedia memiliki kontrol	Wajib	Play, pause, timing, volume, fullscreen
FR201	Konten multimedia memiliki fitur auto-play	Optional	
FR300	Konten aplikasi memiliki kontrol	Wajib	Play, pause, volume, fullscreen, repeat
FR301	Konten aplikasi memiliki fitur auto-play	Wajib	

5. User Interface Requirements

User interface yang dibutuhkan oleh *website* dan aplikasi *multimedia* interaktif untuk balita adalah:

- a. Halaman Beranda
 - Memiliki navigasi utama yang *responsive* dan melayang dibagian paling atas halaman.
 - Gambar utama dan deskripsi pendek ditempatkan dibagian paling terlihat saat situs dibuka.
 - Konten *multimedia* dan aplikasi ditampilkan berdasarkan kategori, yang bisa diakses dengan *scroll* ke bawah atau melalui navigasi utama.
 - Konten *multimedia* video Mengenal Angka ditampilkan menggunakan modal box sehingga tidak perlu pindah halaman.
- b. Halaman Mengenal Angka
 - Konten judul, video, dan deskripsi ditampilkan jelas dan utama.
 - Fungsi untuk menutup modal box dibuat jelas di atas kanan dan satu lagi di bagian bawah/akhir.
- c. Halaman Mengingat Angka
 - Navigasi utama untuk kembali atau ke bagian lain.

- Aplikasi dengan yang ditampilkan menyesuaikan dengan resolusi (*responsive*).
- d. Halaman Menebak angka
 - Navigasi utama untuk kembali atau ke bagian lain.
 - Aplikasi dengan yang ditampilkan menyesuaikan dengan resolusi (*responsive*).

6. Evolution Requirements

Perkembangan yang harus disiapkan oleh *website* dan aplikasi *multimedia* interaktif untuk balita adalah:

- Rancangan akan bisa menambah jumlah *multimedia* dan aplikasi, termasuk jumlah kategorinya.
- Rancangan akan bisa menggunakan *database* untuk menyajikan konten dinamis.

7. Project Constraints

Batasan-batasan proyek perancangan *website* dan aplikasi *multimedia* interaktif untuk balita adalah:

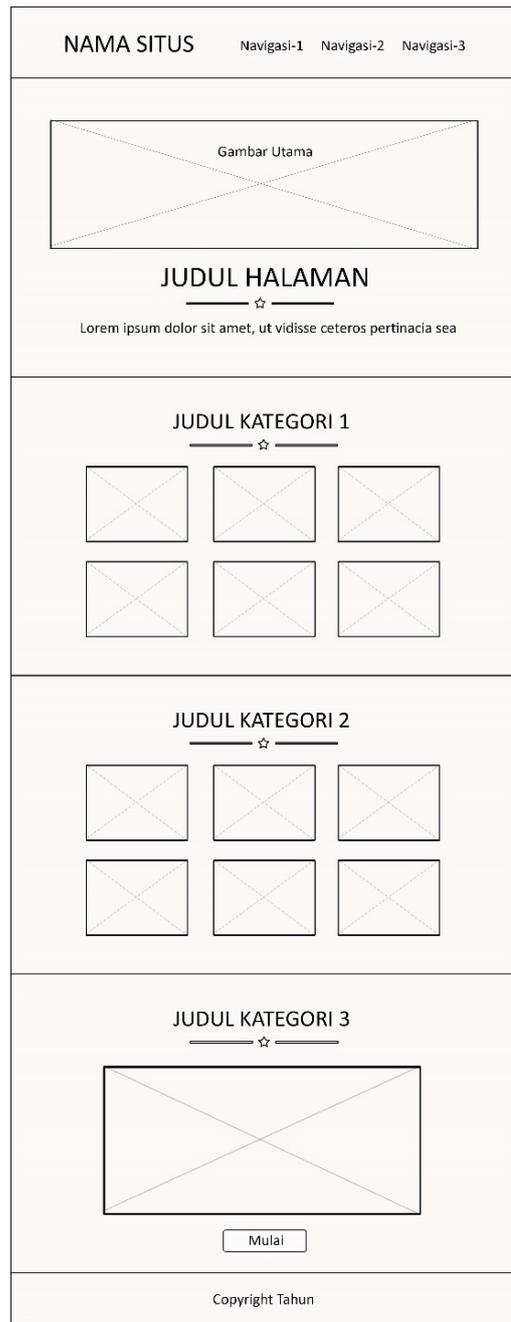
- a. Anggaran
 - Disesuaikan dengan kebutuhan dan kemampuan.
- b. Waktu pengembangan: Total 14 hari.
 - Analisis Kebutuhan: 3 hari.
 - Desain: 2 Hari
 - Implementasi: 8 hari
 - Testing: 1 hari

- c. Sumber daya manusia: Total 1 orang.
 - *Web Designer*: 1 orang (merangkap)
 - *Application Design*: 1 orang (merangkap)
 - *Graphic Design*: 1 orang (merangkap)
- d. Sumber daya teknik
 - 1 desktop *windows pc*
 - *Software browser Firefox, Chrome*
 - *Software text editor Notepad++*
 - *Software Graphic Design Photoshop*
 - *Software PowerPoint*
 - *Software iSpring*
- e. Pemeliharaan: Dukungan perawatan aplikasi dan situs *web* selama 1 bulan.
- f. Konten sesuai dengan norma-norma yang berlaku.

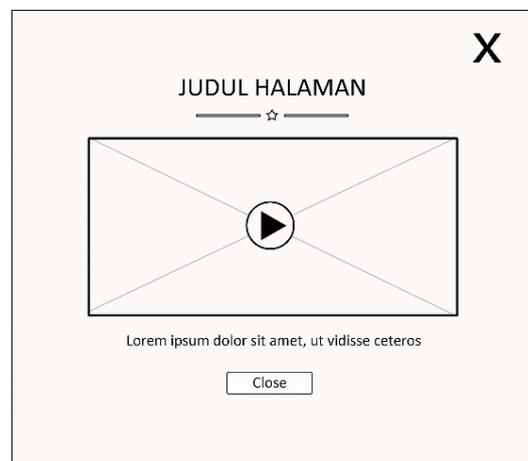
4.2. Desain Web dan Aplikasi

1. Desain Antar Muka

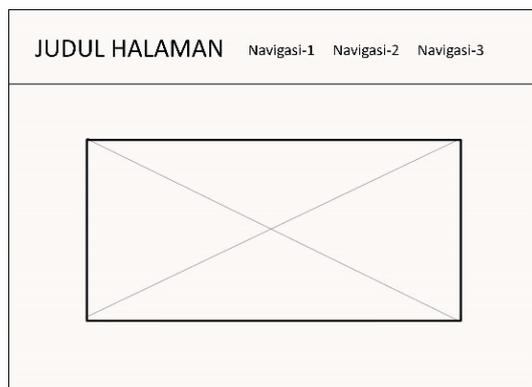
Sesuai analisis kebutuhan *User Interface*, desain antar muka dapat dilihat pada gambar 1, gambar 2, dan gambar 3.



Gambar 1. Rancangan Halaman Beranda



Gambar 2. Rancangan Modal Box Halaman Mengenai Angka



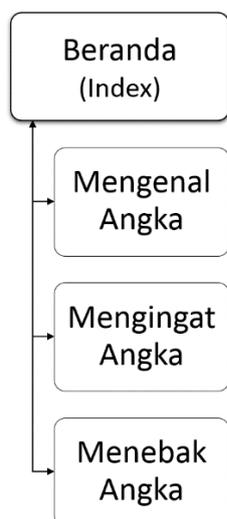
Gambar 3. Rancangan Halaman Mengingat Angka dan Menebak Angka

2. Desain Basis Data

Baik situs maupun aplikasi tidak memiliki konten dinamis, sehingga tidak dibutuhkan sebuah *database*. Karena itu tidak ada desain basis data. Mengenai konten dinamis, telah disebutkan di *evolution requirements*, agar rancangan bisa mudah menampilkannya jika ingin dikembangkan ke arah sana.

3. Desain Struktur Navigasi

Struktur navigasi dibutuhkan untuk menjelaskan hubungan antar halaman situs. Struktur navigasi yang digunakan oleh Aplikasi *Multimedia Interaktif Belajar Berhitung Untuk Usia Balita Berbasis Web* adalah struktur navigasi campuran (*composite*).



Gambar 4. Struktur Navigasi

4.3. Implementasi Web dan Aplikasi

Implementasi *web* dibangun menggunakan *Bootstrap* yang memastikan bahwa situs bersifat *responsive*. Ukuran konten akan mengikuti ukuran resolusi layar dari alat yang digunakan untuk mengaksesnya.

Model aplikasi dirancang menggunakan *Microsoft office PowerPoint*, yang kemudian dibangun menjadi *HTML5* menggunakan *tool iSpring*. Penggunaan *PowerPoint* sebagai alat untuk membuat model aplikasi adalah karena aplikasi sama sekali tidak membutuhkan algoritma pemrograman. Karena target *audience* aplikasi adalah anak *PAUD* usia balita, yang bahkan tidak mengenal bentuk angka, jadi tidak perlu ada perhitungan. Yang diutamakan adalah tampilan gambar, audio, dan interaksi.

Penggunaan *PowerPoint* dan *tool iSpring* mengambil contoh dari *SDLC model Rapid Application Development (RAD)*, yang menggunakan *Application Generation*, yaitu otomatisasi pembuatan aplikasi berdasarkan model yang dibuat. *iSpring* membuat aplikasi untuk *web* berbentuk *Flash* atau *HTML5*, dari model yang dibuat menggunakan *PowerPoint*. *HTML5* dipilih karena kelebihan *HTML5* dibanding media interaktif untuk *web* lainnya, seperti *Flash* adalah, bahwa *HTML5* bisa dijalankan oleh *browser* alat-alat seluler dan dijalankan *browser desktop* tanpa *add-on flash*, yang harus diunduh dan *dinstall* terlebih dahulu agar bisa dijalankan.



Gambar 5. Implementasi Halaman Beranda

1. Implementasi Halaman Beranda

Sesuai analisis kebutuhan dan desain, Halaman Beranda berisi tautan-tautan ke *multimedia* dan aplikasi dengan tipe halaman *web* yang *responsive* dan navigasi utama melayang dibagian paling atas halaman.

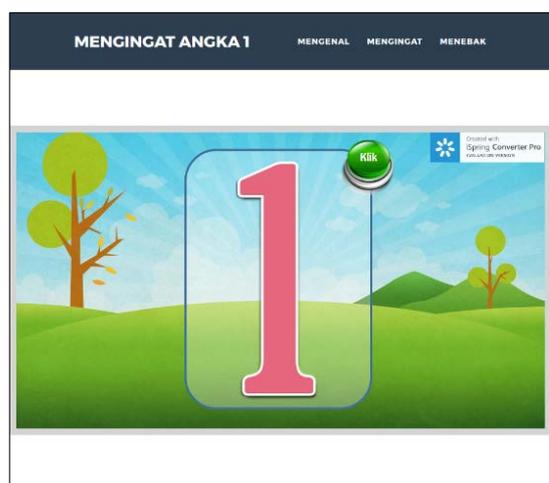
Konten *multimedia* dan aplikasi ditampilkan berdasarkan kategori, yang bisa diakses dengan *scroll* ke bawah atau melalui navigasi utama. Konten *multimedia* video Mengenal Angka ditampilkan menggunakan modal box sehingga tidak perlu pindah halaman. Lihat gambar 5.



Gambar 6. Implementasi Halaman Mengenal Angka

2. Implementasi Halaman Mengenal Angka

Sesuai analisis kebutuhan dan desain, halaman ini berisi judul video, *multimedia* berbentuk video, dan sedikit teks deskripsi untuk video. Fungsi untuk menutup modal box dibuat jelas di atas kanan dan satu lagi di bagian bawah/akhir. Lihat gambar 6.



Gambar 7. Implementasi Halaman Mengingat Angka

3. Implementasi Halaman Mengingat Angka

Halaman ini akan berisi judul aplikasi, aplikasi interaktif untuk mengingat angka, dengan

navigasi utama untuk kembali atau ke bagian lain. Lihat gambar 7.



Gambar 8. Implementasi Halaman Menebak Angka

4. Implementasi Halaman Menebak Angka

Halaman ini akan berisi judul aplikasi, aplikasi interaktif untuk menebak angka, dengan navigasi utama untuk kembali atau ke bagian lain. Lihat gambar 8.



Gambar 9. Hasil Desain Responsive di Layar Alat Seluler

4.4. Testing dan Maintenance

Testing dilakukan dengan menjalankan situs dan aplikasi untuk disesuaikan dengan spesifikasi yang ditetapkan. Karena situs dan aplikasi tidak memiliki konten dinamis, *testing* yang dilakukan pada situs hanya memastikan desain *responsive* dan tautan-tautan sesuai. *Testing* pada aplikasi dilakukan dengan menjalankan aplikasi dan melihat apakah *audio* yang keluar, *timing*, *input* interaksi, dan gambar sesuai. Semua implementasi yang ada pada prototipe ini, sesuai dengan desain dan analisis kebutuhan.

Proses *mantanance* berjalan mudah sejak implementasi, karena seperti yang disebutkan di atas, situs dan aplikasi tidak memiliki konten dinamis. Penggunaan *web* yang ringan sebagai basis dari aplikasi juga membuat kebutuhan sumber daya *hardware* kecil, sehingga belum dan mungkin tidak diperlukan *maintenance hardware* sama sekali.

4.5. Cara Kerja Aplikasi

Sistem pengajaran yang digunakan aplikasi adalah sama seperti saat pembimbing (Guru atau Orangtua) mengajarkan langsung menggunakan gambar dalam buku.

1. Untuk mengenalkan angka, yaitu dengan menunjukkan gambar angka, dan menyebutkannya. Aplikasi ini mengajarkan dengan cara yang sama. Menampilkan angka dan mengeluarkan suara (*audio*) yang menyebutkan angka tersebut menggunakan format video.
2. Untuk mengingat angka, yaitu dengan menunjukkan gambar angka bersamaan dengan *audio* menanyakan gambar tersebut angka berapa. Jawaban animasi dan *audio* akan keluar saat layar diklik oleh *mouse*. Mengingat angka menggunakan animasi dan gambar yang sama dengan mengenalkan angka, yaitu satu angka berulang-ulang, dengan tambahan interaksi untuk mendapat jawabannya.
3. Untuk menebak angka, yaitu dengan menunjukkan gambar angka, 1-6, secara acak, kemudian setelah diklik akan ada animasi dan *audio* yang menjawabnya.

V. KESIMPULAN

Dengan adanya Aplikasi *Multimedia* Interaktif ini, akan lebih mempermudah pembimbing (Guru atau Orangtua) untuk mengajarkan angka kepada anak usia balita. *Multimedia* dan aplikasi interaktif membuat proses belajar lebih menyenangkan bagi anak.

Penggunaan *web* sebagai basis aplikasi membuat tidak perlunya aplikasi dibuat berkali-kali untuk tiap *environment* (lingkungan *software*), karena *web* dengan *HTML5*, *CSS*, dan *JavaScript*, dapat dijalankan oleh hampir semua *browser* pada tiap perangkat yang bisa mengakses *internet*. Dengan desain responsif pengguna aplikasi bisa memilih untuk menggunakannya di *desktop* atau *laptop* dengan tampilan yang besar dan jelas, atau di telpon seluler yang meskipun tampilannya kecil, tapi memiliki sifat portabel.

Untuk pengembangan lebih lanjut, diharapkan:

1. Menambah jumlah angka untuk aplikasi berhitung hingga 10.
2. Menggunakan konten dinamis untuk lebih memudahkan *update* dan modifikasi situs maupun aplikasi.
3. Membuat aplikasi menghitung jumlah untuk melengkapi proses edukasi dari aplikasi ini.

REFERENSI

- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (iJET)*, Vol.2, No.5.
- Blackwell, A. H., & Manar, E. (2015). *Prototype*. Retrieved from District of Columbia Public Library : <http://ic.galegroup.com/ic/scic/ReferenceDetailsPage>
- Kesteren, A. v., & Pieters, S. (2011, April 05). *HTML5 differences from HTML4*. Retrieved from World Wide Web Consortium (W3C): <https://www.w3.org/TR/2011/WD-html5-diff-20110405/>
- Lazaris, L. (2013). *Jump Start CSS*. Collingwood: SitePoint Pty. Ltd.
- Levine, A. (2000, June). *Writing HTML*. Retrieved from Maricopa Center for Learning and Instruction (MCLI): <http://www.mcli.dist.maricopa.edu>
- Spurlock, J. (2013). *Bootstrap*. Sebastopol: O'Reilly Media, Inc.