

BAB II. LANDASAN TEORI

2.1 Bejek

Bejek merupakan sebuah penyedia jasa transportasi yang melayani angkutan melalui jasa ojek. Bejek didirikan pada tahun 2015 di Kabupaten Lumajang oleh Agung Subhakti. Pada awal didirikannya Bejek menggandeng beberapa tunakarya untuk dijadikan *driver* dengan maksud untuk mengurangi jumlah pengangguran di kabupaten Lumajang. Proses pelayanan Bejek pada saat itu dilakukan hanya melalui media WhatsApp. Hingga tahun 2017 Bejek telah meluncurkan aplikasinya dan sudah diunduh sebanyak hampir 10.000 kali di Google Play pada sistem operasi Android. Pada aplikasinya, Bejek mempunyai layanan pembayaran *digital* yang bernama Be-Pay. Bejek menawarkan beberapa fitur dan layanan, yaitu:

1. Bejek Motor, layanan transportasi kendaraan menggunakan armada motor
2. Bejek Mobil, layanan transportasi kendaraan menggunakan armada mobil
3. Bejek Makanan, layanan pesan antar makanan oleh kurir
4. Bejek Barang, layanan pesan antar barang oleh kurir
5. Bejek Pulsa, layanan pembayaran pulsa
6. Bejek *Travel*, layanan *travel* mobil antarkota
7. Bejek Wisata, layanan pengantaran ke tempat wisata
8. Bejek Pijat, layanan pijat profesional
9. Be-Pay, layanan pembayaran *digital*

Bejek juga dikenal dengan logo berwarna oranye, di dalamnya memiliki arti oranye sebagai warna yang hangat sehingga mempunyai daya tarik dan meningkatkan minat pengguna. Sayap memiliki arti sebagai armada yang dapat berkeliling kemana saja. Hal tersebut dimaksudkan Bejek dapat melayani penggunanya secara maksimal dimanapun dan kapanpun. Sedangkan huruf B adalah simbol dari Bejek itu sendiri. Logo Bejek dapat dilihat pada gambar 2.1.



Gambar 2. 1 Logo Aplikasi Bejek

Seiring berjalannya waktu, Bejek terus menambah jumlah *driver* dan mitra kios di bidang kuliner sehingga saat ini Bejek memiliki lebih dari 15 *driver* tetap dan lebih dari 45 mitra kios kuliner di aplikasinya. Bejek sering digunakan oleh sebagian besar masyarakat di Kabupaten Lumajang. Saat ini Bejek telah memperluas wilayah operasi hingga ke kota Jember untuk sebagian layanannya.

2.2 *User Interface*

User Interface atau UI adalah suatu istilah yang digunakan untuk menggambarkan tampilan dari mesin atau komputer yang berinteraksi langsung dengan pengguna. Desain dan penyusunan tampilan antarmuka perlu diperhatikan untuk menghasilkan tampilan yang bagus (Schlatter & Levinson, 2013). Tampilan UI dapat berupa bentuk, warna, dan tulisan yang didesain semenarik mungkin. Secara sederhana, UI adalah bagaimana tampilan sebuah produk yang dilihat oleh pengguna.

Ada 17 prinsip umum desain *user interface* menurut (Mayhew, 2008) yang harus dipahami para perancang sistem, terutama untuk mendapatkan hasil maksimal dari tampilan yang dibuat, diantaranya:

- a. *User Compatibility* yang bisa berarti kesesuaian tampilan dengan tipikal dari *user*. Karena berbeda *user* bisa jadi kebutuhan tampilannya berbeda. Misalnya jika aplikasi diperuntukkan bagi anak-anak, maka jangan menggunakan istilah atau tampilan orang dewasa.
- b. *Product Compatibility* istilah ini mengartikan bahwa produk aplikasi yang dihasilkan juga harus sesuai. Memiliki tampilan yang sama/serupa, baik untuk *user* yang awam maupun yang ahli.

- c. *Task Compatibility* berarti fungsional dari *task*/tugas yang ada harus sesuai dengan tampilannya. Misal untuk pilihan *report*, orang akan langsung mengartikan akan ditampilkan laporan.
- d. *Work Flow Compatibility*, aplikasi bisa dalam satu tampilan untuk berbagai pekerjaan. Misal untuk kirim *mail*, maka kita harus membuka tampilan tersendiri untuk daftar alamat.
- e. *Consistency* yaitu konsistensi dalam desain. contohnya jika menggunakan istilah *save* yang berarti simpan, maka gunakan istilah tersebut pada semua komponen yang memerlukan data disimpan.
- f. *Familiarity* pemilihan *icon* yang biasa digunakan atau yang biasa dipahami. Misalnya pemilihan *icon* disket akan lebih familiar jika digunakan untuk perintah menyimpan.
- g. *Simplicity*, kemampuan untuk melihat segala sesuatu secara sederhana. Aplikasi harus menyediakan pilihan *default* untuk suatu pekerjaan.
- h. *Direct Manipulation* adalah manipulasi secara langsung. Misalnya untuk mempertebal huruf, cukup gunakan ctrl+B.
- i. *Control* yaitu berikan kontrol penuh pada *user*.
- j. WYSIWYG, *What You See Is What You Get* yaitu buatlah tampilan mirip seperti kehidupan nyata *user* dan memastikan fungsionalitas yang ada berjalan sesuai tujuan.
- k. *Flexibility*, *tool*/alat yang bisa digunakan *user*. Jangan hanya terpaku pada *keyboard* atau *mouse* saja.
- l. *Responsiveness* yaitu tampilan yang dibuat harus ada responnya. Misalnya yang sering kita lihat ketika ada tampilan *please wait...* 68%..
- m. *Invisible Technology* yaitu *user* tidak penting mengetahui algoritma apa yang digunakan. Contohnya untuk mengurutkan pengguna tidak perlu mengetahui algoritma yang digunakan *programmer* (*max sort*, *bubble sort*, *quick sort*, dst)
- n. *Robustness* yaitu dapat mengakomodir kesalahan *user*.
- o. *Protection* yaitu melindungi *user* dari kesalahan yang umum dilakukan. Misalnya dengan memberikan fitur *back* atau *undo*.
- p. *Ease of Learning* yaitu aplikasi yang mudah dipelajari

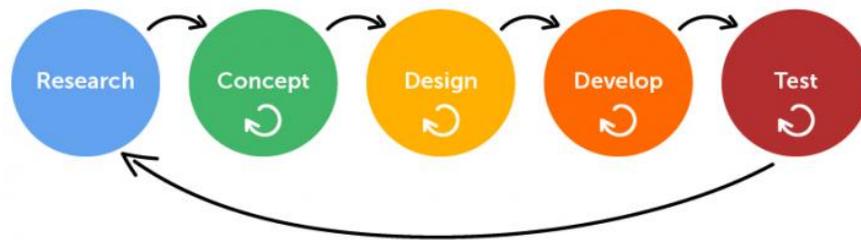
- q. *Ease of use* yaitu aplikasi harus mudah digunakan.

2.3 *User Experience*

User Experience (UX) adalah seluruh aspek yang berfokus dengan pengalaman seorang pengguna dalam menggunakan sebuah produk, seberapa mudah pengguna memahami cara kerja sebuah produk, bagaimana perasaan pengguna saat menggunakan produk, dan bagaimana cara pengguna mencapai tujuannya di dalam penggunaan produk tersebut (Benyon, 2019). *User experience* berfokus pada preferensi, persepsi, emosi, tanggapan fisik dan psikologi pengguna yang terjadi sebelumnya, selama dan setelah menggunakan suatu produk. *User experience* yang baik adalah dimana produk yang sedang dilihat ataupun dipakai membuat perasaan pengguna nyaman dan pengguna ingin menggunakan lagi produk tersebut. *User experience* yang baik akan membantu pengguna bekerja lebih cepat, meningkatkan efisiensi dan produktivitas pengguna dan mengurangi kesalahan pengguna dalam penggunaan produk (Robert & Lesage, 2011).

2.4 *User Centered Design*

User Centered Design (UCD) adalah pendekatan desain dengan proses berdasarkan informasi mengenai orang-orang yang menggunakan produk tersebut. Dengan kata lain, desainer harus berfokus pada pengguna dan kebutuhannya dalam setiap tahap dari proses desain (Irene, 2020). Proses desain pada UCD berfokus pada kebutuhan pengguna. Produk yang dikembangkan dengan pendekatan UCD, dioptimalkan untuk *end-user* serta ditekankan pada bagaimana kebutuhan atau keinginan *end-user* terhadap penggunaan suatu produk (Barbaroux, 2015). Desain dirancang dengan adaptasi terhadap *behavior* atau perilaku pengguna dalam menggunakan produk sehingga produk yang dikembangkan tidak memaksa pengguna untuk mengubah perilakunya ketika menggunakan produk tersebut (Abrams dkk., 2004). Tujuannya adalah agar produk yang dikembangkan dapat bermanfaat serta meningkatkan kemudahan bagi pengguna sehingga dapat mengoptimalkan fungsionalitas dari produk yang dikembangkan sesuai dengan kebutuhan pengguna. Adapun siklus proses *User Centered Design* dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Proses UCD

1. *Research*, pada tahap ini dapat dilakukan dengan *interview* informal dan survei. Fase *research* merupakan kunci untuk menciptakan pengalaman pengguna yang tepat.
2. *Concept*, tahap yang mendasari akan terbentuknya tujuan, apa yang diharapkan, bagaimana ukuran keberhasilannya (*success metric*), dan prioritas apa yang seharusnya ada. Tahap ini bisa juga dikatakan sebagai tahap *ideating*. *UX Designer* juga meminta *feedback* dari pengguna mengenai ide-ide yang dirancang.
3. *Design*, fase ini bertujuan untuk menarik wawasan dengan data yang dibuat pada fase *develop*. Ide-ide dari fase *concept* dapat diwakili oleh *paper prototype*, *wireframe*, dan sebagainya yang semuanya sengaja dibuat dengan *low-fidelity* untuk menunda percakapan yang berkaitan dengan identitas grafis, pencitraan *brand* atau detail visual.
4. *Develop*, tujuan dari tahap ini adalah untuk pada fase ini *UX designer* membangun sebuah *prototype* desain dan berkolaborasi dengan *developer* untuk mendiskusikan ide rancangan yang dibuat.
5. *Test*, di fase ini adalah ketika desain *high-fidelity* berperan, aset konten dan digital diciptakan, dan versi *high-fidelity* produk divalidasi dengan *stakeholder* dan *end-user* melalui sesi *user testing*. Pada aktivitas ini, *UX designer* juga mengkomunikasikan pemahaman mereka kepada *end-user* untuk membantu mengkonfirmasi valid atau tidaknya asumsi apa pun yang dibuat.

2.5 System Usability Scale

System Usability Scale (SUS) adalah salah satu metode uji pengguna yang menyediakan alat ukur yang “*quick and dirty*” dan dapat diandalkan. Diaplikasikan

dengan menggunakan 10 pertanyaan berbentuk kuesioner yang diikuti dengan 5 opsi jawaban untuk setiap pertanyaan, mulai dari Sangat Setuju hingga Sangat Tidak Setuju (Mustami, 2016). Untuk contoh pertanyaan diperlihatkan pada tabel 2.1.

Tabel 2. 1 10 Pertanyaan SUS

No.	Pertanyaan
1.	Saya berpikir akan menggunakan sistem ini lagi
2.	Saya merasa sistem ini rumit untuk digunakan
3.	Saya merasa sistem ini mudah digunakan
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
8.	Saya merasa sistem ini membingungkan
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Pertanyaan Metode uji pengguna ini diperkenalkan oleh John Brooke di tahun 1986 yang dapat digunakan untuk mengevaluasi berbagai jenis produk maupun servis, termasuk di dalamnya *hardware*, *software*, perangkat *mobile*, *website* dan aplikasi. Pada pengembangan yang dilakukan oleh penulis, metode SUS digunakan untuk mengevaluasi *usability* dari desain solusi yang dihasilkan. Dari hasil tersebut dapat diketahui tingkat kesesuaian kebutuhan pengguna terhadap desain solusi sebelum dilakukan iterasi. (Alathas, 2018) *System usability scale* didapatkan dari hasil olahan nilai jawaban kuesioner. Adapun aturan perhitungan sebagai berikut :

1. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapat dari skor pengguna akan dikurangi 1.

2. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna.
3. Skor SUS didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5.

Aturan perhitungan skor untuk berlaku pada 1 responden. Untuk perhitungan selanjutnya, nilai SUS dari masing-masing responden dicari nilai rata-ratanya dengan menjumlahkan semua nilai dan dibagi dengan jumlah responden. Berikut rumus menghitung nilai SUS:

$$\bar{X} = \frac{\sum x}{n}$$

Keterangan:

- \bar{x} : *Score* rata- rata
 Σx : Jumlah *score* SUS
 n : Jumlah reponden

Score SUS memiliki rentang nilai 0-100. Untuk mengetahui kualitas perangkat lunak yang dibuat, dirujuk ke grafik *percentile rank* terhadap SUS score (Kalbach, 2016) berikut ini :

Tabel 2. 2 *Adjective Rating* SUS

SUS Score	Grade	Adjective Rating
>80,3	A	Excellent
>74 – 80,3	B	Good
>68 - <74	C	Okay
51 – 68	D	Poor
<51	E	Awful

Untuk itu dalam menentukan SUS score seperti ketentuan sebagai berikut:

- a. *Grade* A : dengan nilai lebih besar atau sama dengan 80,3.
- b. *Grade* B : dengan nilai lebih besar sama dengan 74 dan lebih kecil 80,3.
- c. *Grade* C : dengan nilai lebih besar 68 dan lebih kecil 74.
- d. *Grade* D : dengan nilai lebih besar sama dengan 51 dan lebih kecil 68.
- e. *Grade* E : dengan nilai lebih kecil dari 51.

2.6 *User Persona*

User persona adalah dokumentasi yang berisi penjelasan tentang karakteristik pengguna digabungkan dengan tujuan, kebutuhan dan ketertarikannya yang menjadi target pengguna (Harley, 2015). Istilah *persona* diperkenalkan pertama kali oleh Alan Cooper yang menggunakan *persona* dalam perancangan interaksi secara praktis untuk menghasilkan produk *high-tech*. Isi yang didapatkan berasal dari hasil penelitian tentang pengguna yang sesuai target.

Tujuan dibuatnya *user persona* pada pengembangan yang dilakukan penulis adalah untuk melakukan pendekatan kepada calon pengguna agar lebih memahami karakteristik pengguna yang menjadi target. *User persona* mewakili target pengguna dari produk yang akan dibuat dan berfokus pada data personal sebuah karakter seperti, data demografi, sikap/tingkah laku, motivasi, pengaruh, tujuan dan kesulitan. Kasarnya seperti mengarang biodata seseorang (Ambarwati, 2020).

2.7 *Moodboard*

Menurut (Koch dkk., 2020) papan suasana hati (*Moodboard*) adalah kolase atau komposisi gambar, visual, dan objek lainnya, sering kali dibuat untuk tujuan desain atau presentasi. *Moodboard* dapat membantu desainer untuk mengetahui cara membuat gaya atau tema terpadu untuk proyek desain karena *moodboard* memegang peran penting untuk mengenalkan *brand* secara akurat. Secara umum, *moodboard* bersifat kasual sehingga bebas bagi desainer untuk menentukan seperti apa bentuknya nanti. Pada pengembangan yang dilakukan penulis, pembuatan *moodboard* dilakukan pada proses konsep untuk mengkomunikasikan kebutuhan pengguna terhadap aplikasi ke dalam bentuk visual.

2.8 *Wireframe*

Wireframe adalah rancangan awal sebuah desain web yang dibuat secara manual menggunakan *software* pengedit gambar. *Wireframe* dapat disebut sebagai *blueprint* dalam arsitektur (Anthony, 2010). Pada pengembangan yang dilakukan penulis, *wireframe* diimplementasikan sebagai *low-fidelity prototype* pada rancangan desain solusi tahap awal untuk dilakukan validasi desain. Tujuan dibuatnya *wireframe* adalah untuk menyampaikan susunan, struktur, *layout*, navigasi dan organisir konten. Maka dari itu, biasanya *wireframe* dibuat dengan

warna hitam putih. *Wireframe* lebih menekankan isi dari konten. Berikut adalah beberapa kelebihan *wireframe*:

1. Menggambarkan *layout* umum dari aplikasi
2. Membangun kepercayaan dengan *user* dan *stakeholders*
3. Menghemat biaya dan waktu

2.9 High-fidelity Prototype

Prototype adalah tahapan yang ditujukan untuk mentransformasi sifat-sifat abstrak dari sebuah ide menjadi lebih berwujud. Tahapan ini tidak hanya berupa proses visualisasi ide tetapi juga proses pembangunan ide (Pangaribuan, 2019).

High-fidelity prototype adalah sebuah rancangan yang menggambarkan sistem dengan karakteristik lebih rinci atau mendekati produk jadi. *Prototype* ini mempunyai interaksi penuh dengan pengguna dimana pengguna dapat berinteraksi dengan sistem. *High-fidelity prototype* mewakili fungsi-fungsi inti sehingga dapat mensimulasikan sebagian besar fungsi dari sistem akhir dan mempunyai penampilan yang sangat mirip dengan produk sebenarnya (Ramadhan, 2017). Pada pengembangan ini, penulis mengimplementasikan *high-fidelity prototype* pada pembuatan *mockup* untuk memvisualisasikan versi detail dan *clickable* dari *low-fidelity prototype* yang telah dibuat, seperti warna, *icon*, dll.

Dengan metode *prototyping* ini akan dihasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem. (Ogedebe & Jacob, 2012) Tujuan *prototyping* adalah mengembangkan model menjadi sistem final.