



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

TINJAUAN PUSTAKA

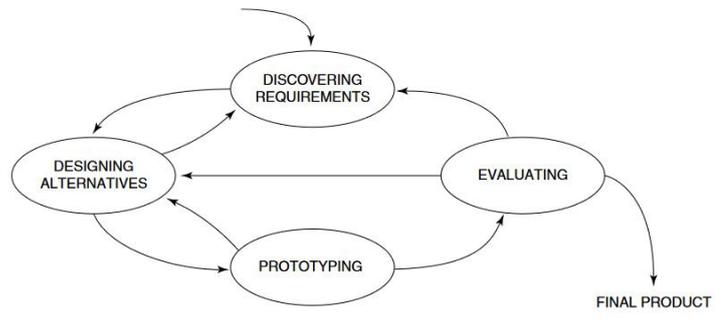
2.1 Interaction Design

Sharp, Preece, dan Rogers (2019) menjelaskan bahwa proses mendesain produk interaktif adalah membuat *user experience* yang memudahkan komunikasi dan interaksi antar sesama manusia di kehidupan sehari-hari. Dalam proses desain tersebut, maka desainer perlu mempertimbangkan aspek dari *user*, teknologi dan interaksi untuk menghasilkan *user experience* yang baik. Pada umumnya, *interaction design* dibuat oleh gabungan berbagai profesional di bidang yang berbeda-beda, hal ini memberikan potensi pengembangan proses desain *interaction design* semakin luas karena berbagai metode, ide, desain yang dihasilkan dapat berbeda-beda.

Sedangkan menurut Kolko (2011), *interaction design* adalah hasil interaksi dialog antara suatu persona dengan produk, *service*, atau sistem, dimana interaksi tidak terlihat tampak namun menjadi bagian dari kehidupan sehari-hari manusia. Interaksi yang terjadi bergantung pada kondisi bentuk, fungsi, dan teknologi yang digunakan seiring dengan perkembangan zaman. *Interaction designers* menghasilkan *interaction design* sebagai *the shapers of behaviors*, dimana setiap desain yang dibuat dipertimbangkan melalui pengamatan observasi terhadap tindakan, pikiran, dan perasaan manusia.

2.1.1 Four Basic Activities of Interaction Design

Menurut Sharp, Preece, dan Rogers (2019), *interaction design* memiliki proses dengan empat fase aktivitas dasar yang terdiri atas *discovering requirements*, *designing alternatives*, *prototyping*, dan *evaluating*. Masing-masing aktivitas dilakukan berurut namun saling berkesinambungan menjadi satu model *lifecycle*, dengan penjelasan masing-masing fase aktivitas sebagai berikut:



Gambar 2.1 Model *Lifecycle Interaction Design*
 Sumber: Sharp, Preece, & Rogers (2019)

1. *Discovering Requirement*

Proses desain dimulai dengan mencari ide dan bentuk produk interaktif yang ingin dikembangkan dari ide tersebut. Selanjutnya, desainer melakukan pencarian data dan analisis untuk menentukan seperti apa *target user* dan manfaat dari produk interaktif yang ingin dibuat.

2. *Designing Alternatives*

Berikutnya, desainer mulai mempertimbangkan ide-ide apa saja yang dapat digunakan untuk memenuhi manfaat dari produk interaktif tersebut. Dalam *interaction design*, maka ada dua tindakan aktivitas yang perlu dilakukan yaitu membuat desain konseptual dan desain konkrit. Desain konseptual menjabarkan *outline* abstrak untuk membentuk konsep model dan bagaimana cara *user* dapat berinteraksi dengan bentuk konsep model tersebut. Sedangkan desain konkrit menjabarkan detail-detail produk yang akan disertakan, seperti warna, suara, gambar, desain menu, dan desain ikon. Setiap desain alternatif akan dipertimbangan pada setiap tahap pengembangan.

3. *Prototyping*

Prototyping dilakukan dengan melibatkan *user* untuk berinteraksi dengan *prototype* yang sudah dibuat. Contoh dari teknik *prototyping* yang bisa dilakukan adalah menggunakan *prototype* kertas untuk mengevaluasi pengembangan awal desain, atau melakukan *role playing* untuk mengobservasi perkiraan interaksi *user* dengan *prototype* produk interaktif.

4. *Evaluating*

Proses *evaluating* dilakukan untuk mengembangkan produk final sehingga dapat memenuhi manfaat yang ditentukan. Evaluasi didasarkan atas penentuan manfaat, eksplorasi alternatif desain, dan pembuatan *prototype*. Dari hasil evaluasi tersebut akan dipertimbangkan pengembangan desain atau alternatif apa yang perlu ditambahkan selanjutnya.

2.1.2 Design Principles

Sharp, Preece, dan Rogers (2019) menyatakan bahwa *design principles* berfungsi sebagai panduan tolak ukur desainer akan apa yang harus disediakan dan yang perlu dihindari pada saat mendesain *interaction design*. *Design principles* pada *interaction design* bersifat abstrak dan tidak kaku, karena prinsip-prinsip didasarkan oleh perpaduan teori, pengalaman, dan nalar. Prinsip tidak mengatur secara harfiah bagaimana desainer harus mendesain, tetapi lebih berperan sebagai *trigger* desainer agar dapat memenuhi *feature-feature* yang diperlukan pada sebuah *interaction design*.

Berikut adalah prinsip-prinsip desain yang umumnya digunakan, yaitu:

1. *Visibility*

Visibility fungsi yang jelas akan memudahkan *user* untuk mengambil keputusan akan tindakan yang ingin dilakukan selanjutnya,

contohnya seperti penjelasan Don Norman (1988) mengenai kontrol pada mobil. Kontrol yang mengoperasikan mobil seperti kontrol indikator, lampu depan, klakson, dan lampu *hazard* di tata sedemikian rupa sehingga memudahkan pengemudi pada saat menggunakan mobil. Jika *visibility* fungsi tidak terlihat jelas maka *user* juga akan kesulitan untuk mengakses fungsi yang ingin digunakan. Contohnya seperti fasilitas elevator, *faucet*, dan lampu dengan sensor otomatis dengan fungsi aktivasi tersembunyi, sehingga *user* perlu menebak-nebak cara menggunakannya dan rentan frustrasi.

2. *Feedback*

Visibility yang dapat di mengerti akan menghasilkan *feedback* yang baik untuk memperlancar proses *user interaction*. Proses *feedback* adalah menginformasikan hasil dari tindakan yang telah dilakukan dan telah dituntaskan, dan setelahnya *user* dapat melanjutkan aktivitas interaksi selanjutnya. Menyesuaikan kombinasi antara jenis-jenis aktivitas dengan interaktivitas yang sesuai penting untuk mendapatkan *feedback* yang tepat. *Feedback* pun dapat muncul dalam bentuk audio, *tactile*, verbal, visual ataupun kombinasi dari berbagai jenis. Contohnya, seperti pada saat menulis di kertas dengan pena, maka *feedback* yang didapatkan adalah berupa tulisan dari hasil proses tindakan menulis tersebut.

3. *Constraints*

Constraining diaplikasikan untuk membatasi interaksi-interaksi apa saja yang dapat diakses dan tidak dapat diakses oleh *user*. Contohnya, seperti teks menu berwarna abu-abu yang mengartikan bahwa fungsi menu tersebut tidak dapat digunakan. *Constraining* digunakan untuk mencegah *user* melakukan kesalahan atau memilih opsi yang berakibat fatal.

4. *Consistency*

Konsistensi umumnya digunakan untuk berbagai fungsi pada *interface* yang memiliki fungsi atau tujuan yang mirip, sehingga mudah di pelajari karena memiliki proses operasional yang sama. Konsistensi biasanya diaplikasikan pada sistem yang simpel, contohnya seperti fungsi klik kiri pada *mouse*. Namun, fungsi yang konsistensi akan mempersulit *user* jika ditawarkan pada *interface* dengan berbagai pilihan fungsi dengan proses operasional dan tujuan yang berbeda-beda. Contohnya, seperti aplikasi pengolah dokumen teks yang memiliki berbagai tombol dengan fungsi yang beragam. Untuk mengatasinya, tombol-tombol dapat dibagi menjadi beberapa kategori menu sesuai dengan fungsinya sehingga mudah untuk diakses oleh *user*.

5. *Affordance*

Affordance adalah kemampuan suatu objek untuk memberikan petunjuk arahan cara menggunakan objek tersebut (Norman, 1988). Contoh *affordance* pada *interface* adalah penggunaan elemen grafis seperti *scrollbar* yang otomatis mengarahkan *user* untuk digerakkan ke atas dan ke bawah.

2.1.3 Interaction Types

Menurut Sharp, Preece, dan Rogers (2019) berikut adalah jenis-jenis tipe interaksi:

1. *Instructing*

Bentuk interaksi berupa instruksi yang diberikan oleh *user* kepada suatu sistem untuk melakukan suatu hal. Interaksi tersebut antara lain seperti menulis dan mengirim suatu perintah, memilih opsi pada menu *windows* atau pada *multitouch screen*, mengucapkan perintah, memberi gestur, memencet *button*, atau menggunakan kombinasi

dari berbagai *function keys*. Contoh model *instructing* adalah *home entertainment system*. Kelebihannya tipe interaksi *instructing* dapat dijalankan dengan cepat dan efisien.

2. *Conversing*

Bentuk interaksi berupa dialog antara *user* dengan suatu sistem, dimana *user* dapat berbicara langsung kepada suatu *interface* atau menuliskan pertanyaan, lalu dibalas oleh sistem dalam bentuk teks atau suara. Sistem bekerja sebagai partner dialog *user*, karena sistem akan merespon *user* layaknya manusia pada umumnya. Biasanya tipe interaksi *conversing* digunakan pada aplikasi untuk mencari informasi tertentu atau untuk mendiskusikan isu tertentu, contohnya seperti Siri, *speech system* yang digunakan pada produk Apple.

3. *Manipulating*

Bentuk interaksi berupa *user* yang memanipulasi suatu objek baik secara virtual maupun langsung, seperti membuka, memegang, menutup, atau memindahkan. Tindakan tersebut dilakukan melalui suatu *physical controllers* atau gestur gerakan, contohnya seperti *game console* Wii.

4. *Exploring*

Bentuk interaksi berupa *user* yang dapat mengeksplorasi dengan bergerak langsung di suatu lingkungan virtual (3D, *augmented reality*, *virtual reality*) atau lingkungan nyata (*smart rooms* dan *ambient enviroments* dengan teknologi sensor)

5. *Responding*

Bentuk interaksi dimulai dari inisiatif sistem untuk berinteraksi, dan *user* akan memilih untuk merespon atau tidak. Contoh penggunaan

interaksi ini adalah aplikasi Google Lens yang dapat mengidentifikasi jenis *breed* binatang anjing berdasarkan foto saja.

2.2 Interactivity

Menurut Crawford (2003), interaktivitas adalah proses dua pihak yang saling mendengarkan, berpikir, dan berbicara.

2.2.1 Interactivity Principle

Menurut Tomaszewski (2004), Meadows (2003) menyatakan bahwa *interactivity* dibentuk dari prinsip-prinsip berikut:

1. *Input/Output*

User dapat mempengaruhi sistem (*input*) dan memperlihatkan keadaan sistem tersebut (*output*). *Input* harus jelas dan waktu respons antara *input* dan *output* harus singkat.

2. *Inside/Outside*

Membentuk suatu sistem interaktif didasari atas pengalaman dari dalam dan luar kerangka. Maksud dari dalam kerangka adalah pengalaman dan interpretasi oleh *user* pada sistem tersebut, sedangkan luar kerangka adalah hasil observasi yang dilihat, seperti *frame rate* atau kualitas suara.

3. *Open/Closed*

Sistem terbuka tidak dapat diprediksi dan bisa berubah, sedangkan sistem tertutup akan selalu memberikan respons yang sama. Meadows memberikan contoh bentuk sistem tersebut seperti menendang batu bata dan menendang manusia. Setiap batu bata akan merespons dengan sama sedangkan respons manusia akan berbeda-beda. Sistem terbuka lebih *engaging* jika dibandingkan dengan sistem tertutup.

2.2.2 Interactivity Plots

Menurut Meadows (2003), berikut adalah *interactivity plots*:

1. *Nodal Plot*

Nodal Plot terdiri dari beberapa tahap yang diurutkan dari awal dan akhir dalam satu baris. Setiap urutan tahap tersebut akan menjadi tempat dimana peserta menyelesaikan tugas atau level yang diperlukan untuk maju ke tahap selanjutnya. Jika peserta gagal, maka peserta harus mengulang kembali level tersebut.

2. *Modulated Plot*

Modulated Plot menawarkan berbagai pilihan pada tahap-tahap tertentu. Struktur tahap ini bisa dijalankan secara acak, dilewati, atau dipilih suatu keputusan tertentu yang nantinya akan mengubah hasil cerita. Namun semakin banyak pilihan yang diberikan, maka semakin banyak kemungkinan alur cerita yang akan muncul, dan semakin sulit untuk menentukan apakah cerita tersebut menjadi menarik atau koheren.

3. *Open Plot*

Hanya ada sedikit alur dalam *open plot*, karena biasanya diadakan suatu simulasi yang bergantung pada peraturan tertentu. Daripada hanya mengikuti satu alur cerita saja, peserta bisa mengembangkan karakter dan lingkungan dalam alur cerita tersebut. Misalnya seperti pada games Ultima Online, Sims, dan Age of Empires

2.3 Aplikasi

Menurut Cuello dan Vittone (2018), aplikasi adalah perangkat lunak yang setara dengan program komputer, namun dioperasikan pada telepon genggam. Aplikasi bisa diakses tanpa menunggu dan tanpa bergantung pada internet, dikarenakan pengalaman user yang lebih praktis dengan alur navigasi yang jelas untuk mengakses konten aplikasi.

2.3.1 Perancangan Aplikasi

Menurut Cuello dan Vittone (2018), berikut adalah proses perancangan aplikasi:

1. *Conceptualization*

Ide aplikasi dikembangkan melalui riset menyeluruh, serta pertimbangan kebutuhan dan masalah calon pengguna aplikasi tersebut.

2. *Defining*

Persona dan *User Journey* ditentukan untuk menjadi fondasi jenis dan fungsi aplikasi.

3. *Design*

Berdasarkan hasil diskusi mengenai konsep aplikasi yang ingin dibuat, maka dapat dimulai pembentukan *wireframe* sebagai awalan dari *prototype* dini untuk diuji kepada pengguna. Pengembangan desain visual dan *programming* juga mulai dikembangkan.

4. *Development*

Programmer mulai menyesuaikan kerangka struktur aplikasi berdasarkan fungsi dan desain aplikasi yang telah ditentukan. *Bug* dan kinerja fungsi aplikasi juga mulai diperbaiki agar memenuhi standar pasar.

5. *Publishing*

Bersamaan dengan perilisan aplikasi di toko, maka dimulai observasi terhadap data analitik, statistik, dan komentar user terhadap aplikasi. Hasil observasi berupa data perilaku dan performa aplikasi akan menjadi landasan evaluasi untuk memperbaiki dan mengembangkan aplikasi ke depannya.

2.3.2 Jenis-jenis Aplikasi

Setiap jenis aplikasi memiliki karakteristik dan batasan masing-masing, hal tersebut akan mempengaruhi *programming*, desain visual, dan interaksi aplikasi. Maka berdasarkan jenis, berikut tipe-tipe aplikasi menurut Cuello dan Vittone (2018):

1. Aplikasi *native*

Aplikasi *native* adalah aplikasi didesain sesuai dengan bahasa *Software Development Kits* (SDKs), yaitu sistem operasi *software* seperti Android, iOS, dan Windows Phone. Aplikasi *native* bisa diunduh melalui *app store* dan perlu di *update* secara berkala. Karakteristik aplikasi *native* adalah kemampuan aplikasi untuk memberi notifikasi pada saat digunakan dan memiliki akses pada fungsi kamera dan sensor *device*.

2. Aplikasi *web*

Aplikasi *web* adalah aplikasi yang di desain berdasarkan HTML, Javascript, dan CSS. Aplikasi *web* tidak bergantung pada SDKs sehingga dapat beradaptasi dengan berbagai *platform* yang berbeda. Pada umumnya aplikasi *web* memiliki visualisasi yang serupa pada saat diakses melalui *browser website*, namun hal ini membuat visual aplikasi terlihat generik. Karena aplikasi *web* dapat bekerja melalui *website* maka akan selalu *terupdate* otomatis, namun aplikasi *web* masih bergantung pada koneksi internet.

3. Aplikasi *hybrid*

Aplikasi *hybrid* adalah gabungan dari dua aplikasi sebelumnya. Sistem aplikasi *hybrid* serupa dengan aplikasi *web*, namun hasil akhirnya terlihat seperti aplikasi *native*. Maka pada umumnya, developer tetap menggunakan kode yang sama meskipun *software* dan distribusi *app store* berbeda tempat. Aplikasi *hybrid* memiliki

akses seperti aplikasi *native* dan memiliki visual yang dapat disesuaikan dengan *platform* berbeda-beda.

2.3.3 Kategori Aplikasi

Kategori aplikasi dibedakan berdasarkan konten yang ditawarkan, setiap kategori aplikasi memiliki *interface* dan potensi monetisasi yang berbeda-beda. Berikut adalah berbagai kategori aplikasi berdasarkan konten menurut Cuello dan Vittone (2018):

1. *Entertainment*

Pada umumnya merupakan *gaming apps*. Penataan grafis, animasi dan efek suara biasanya digunakan untuk menarik perhatian *user*.

2. *Social*

Aplikasi sosial memiliki tujuan untuk membangun interaksi komunikasi antar *user*, contohnya seperti Facebook, Path, Twitter, dan Instagram. Aplikasi gratis tidak dipungut biaya, namun keuntungan aplikasi sosial biasanya didapatkan melalui data personal *user* atau *in-app purchases*.

3. *Utilities and Productivity*

Pada umumnya aplikasi berbentuk *to do list* atau aplikasi untuk *work teams* yang digunakan pada sektor bisnis, dan berfungsi efisien untuk menyelesaikan *tasks* atau menyelesaikan masalah tertentu.

4. *Educational and Informative*

Aplikasi *educational* dan *informative* memiliki fungsi utama sebagai penyedia informasi pengetahuan dan berita, sehingga akses konten, *legibility*, navigasi, dan *search tools* perlu didesain dengan baik karena merupakan fungsi utama.

5. Creation

Jenis aplikasi *creation* biasanya digunakan untuk membuat suatu kreasi, misalnya aplikasi untuk mengedit video, foto, suara, maupun menulis.

2.4 User Interface

Menurut Wood (2018), *interface* adalah kontak poin dari interaksi antara manusia dan mesin. *User interface* yang digunakan pada komputer, *smartphone*, tablet atau konsol gim memiliki pola visual interaktif '*front end*' yang dikenal sebagai *graphical user interfaces* (GUIs), yang di dukung oleh sistem *delivery* terprogram '*back end*'. Kebutuhan dan ekspektasi *user* harus menjadi fokus utama desainer dalam mendesain suatu UI sehingga mendapatkan hasil *usability*, *functionality* dan *aesthetics* desain yang baik. *Interface design* menjalani proses iteratif dengan melakukan *user research*, *ideation*, *testing*, *building* bersama tim desain UI, *target user*, dan klien untuk mendapatkan desain interaktif *user experience*.

2.4.1 User Interface Elements

Menurut UXPin (2020), berikut adalah tiga jenis elemen *user experience* yaitu:

1. Input elements

Input elements adalah elemen yang bertanggungjawab menangani berbagai *input* yang berbeda dari *user* dan menangani proses validasi *input*. Contohnya seperti: *dropdowns*, *combo boxes*, *buttons*, *toggles*, *text/password fields*, *date pickers*, *checkboxes*, *radio buttons*, dan *confirmation dialogues*.

2. Output elements

Output elements adalah elemen yang bertanggungjawab menunjukkan hasil respons dari *input user*, sehingga *output elements*

bergantung pada *input* dan sistem operasional. Contohnya seperti: *alerts, warnings, success* dan *error messages*.

3. *Helper elements*

Sisa elemen-elemen yang tersisa termasuk ke dalam kategori *helper elements*. Contohnya seperti: *notifications, breadcrumbs, icons, sliders, notifications, progress bars*, dan *tooltips*.

2.5 User Experience

Menurut Marsh (2016), *user experience* adalah proses mendesain *user experience*, melalui *feedback* opini langsung dan subjektif dari *user experience* individu mengenai aplikasi atau situs yang dibuat. Proses *user experience* didasari oleh riset mengenai pengertian *user*, lalu pengembangan ide untuk memenuhi kebutuhan *user* dan kebutuhan bisnis, kemudian kedua kebutuhan tersebut disesuaikan menjadi solusi dan diuji apakah bekerja dengan baik atau tidak.

2.5.1 5 Main Ingredients of UX

Menurut Marsh (2016), berikut adalah 5 unsur penting yang perlu dipertimbangkan dalam mendesain suatu UX, yaitu:

1. *Psychology*

Desainer harus pertama-tama mempertimbangkan pandangan *user* secara psikologi mengenai *user experience* yang akan dibuat, karena *user experience* didesain untuk *user*. Maka dari itu, desainer perlu mengesampingkan keinginan dan selera pribadi dalam mendesain *user experience* agar sesuai dengan psikologi *user*.

2. *Usability*

Desainer perlu menguji apakah *user experience* yang diperlukan membingungkan atau tidak. *User experience* didesain sedemikian rupa sehingga mudah diakses dan dimengerti oleh *user*.

3. *Design*

Desainer perlu mengesampingkan selera desain, karena fokus pada *user experience* adalah apakah desain yang dibuat bekerja dengan baik atau tidak.

4. *Copywriting*

User experience membutuhkan teks *UX copy* yang simpel dan sejelas mungkin.

5. *Analysis*

Desainer perlu menganalisa proses mendesain *user experience* secara menyeluruh agar dapat mengembangkan *user experience* dengan baik.

2.5.2 **User Experience Elements**

Menurut Brenton (2011), berikut adalah elemen-elemen yang disusun berurut sehingga membentuk suatu *user experience*, yaitu:

1. *The Strategy Plane*

Desainer harus mengerti apa yang diinginkan oleh audiens dan menyesuaikan keinginan audiens untuk mencapai tujuan objektif produk.

2. *The Scope Plane*

Desainer mulai menentukan aspek fungsionalitas produk, dengan menyusun daftar “*feature set*” produk secara terperinci dan menyusun jenis konten elemen apa saja yang akan digunakan.

3. *The Structure Plane*

Desainer mulai membentuk ruang dengan kerangka *interaction design* dan menentukan bagaimana sistem akan merespon *user*. Pendataan informasi dilakukan dengan membuat *information architecture*.

4. *The Skeleton Plane*

The Skeleton Plane dibagi menjadi tiga komponen utama. Yang pertama adalah *information design*, dimana informasi yang dijabarkan dapat dimengerti dengan baik. Yang kedua adalah *interface design*, dengan menyusun elemen-elemen *interface* untuk memudahkan *user* berinteraksi dengan fungsi sistem. Yang ketiga adalah *navigation design*, dengan menggunakan set elemen layar yang memberikan pengarahan *user* terhadap *information architecture*.

5. *The Surface Plane*

Desainer mulai memperhatikan bagaimana *sensory experience* yang ada pada hasil akhir produk.

2.6 Informasi GERD

Dari hasil wawancara, dr. Christian menyatakan bahwa penyakit GERD terjadi pada saat produksi asam lambung berlebihan pada tubuh terjadi berulang-ulang akibat dari pencernaan yang terganggu. Pencernaan yang terganggu terjadi akibat dari tidak adanya makanan yang masuk ke dalam tubuh, sehingga asam lambung merusak dinding lambung dan menyebabkan rasa nyeri pada perut. Hapsari, Putri, Rahardja, Utari, dan Syam (2017) menyatakan bahwa menurut penelitian epidemiologi yang dilakukan, ditemukan bahwa setelah tahun 2005 angka penderita GERD di area Asia Tenggara dan Asia Barat (6.3-18.3%) lebih tinggi dibandingkan dengan di area Asia Timur (5.2-8.5%).

Radjamin, Nusi, dan Kalanjati (2019) melakukan penelitian analisis observasional kepada pasien GERD di Unit Endoskopi Divisi Gastroentero-Hepatologi Departemen SMF Penyakit Dalam FK UNAIR-RSUDS Surabaya pada bulan Januari hingga Desember 2015. Dari hasil penelitian penderita penyakit GERD berdasarkan pengelompokan umur, hasil menunjukkan bahwa 11 pasien (9,5%) berumur kurang atau sama dengan 30 tahun, 13 pasien (11,2%) dengan umur 31-40 tahun, 28 pasien (24,1%) dengan umur 41-50 tahun, 55 pasien

(47,4%) dengan umur lebih dari 50 tahun, dan 9 pasien (7,8%) yang tidak diketahui umurnya karena data yang kurang lengkap. Sedangkan untuk pengelompokan penderita penyakit GERD berdasarkan pengelompokan jenis kelamin, hasil menunjukkan bahwa 54 pasien (46,6%) adalah pria dan 62 pasien (53,4%) adalah wanita.

2.6.1 Penyebab GERD

Menurut Nandi (2018), berikut adalah penyebab terjadinya penyakit asam lambung GERD, yaitu:

1. Ketidakseimbangan Asam Lambung (*Stomach Acid Imbalances*)

Ketika manusia mengkonsumsi makanan, maka sistem pencernaan akan menghasilkan asam lambung untuk memproses makanan dan mengeliminasi bakteri jahat. Namun kadang produksi berlebih asam lambung terjadi pada sistem pencernaan, sehingga asam lambung menembus ke kerongkongan dan menyebabkan maag bersama dengan simptom-simptom GERD lainnya.

2. Hernia Hiatal

Lubang diafragma esophageal berperan sebagai pembatas antara area perut dan area dada manusia. Sistem pencernaan berjalan dari kerongkongan menuju perut melalui lubang diafragma tersebut. Namun pada kondisi tertentu, area perut membengkak hingga seukuran lubang diafragma yang menyebabkan asam refluks dan membentuk hernia.

3. Obesitas

Banyaknya lemak perut memiliki resiko besar pemicu utama penyebab penyakit GERD. LES terasit akibat perut yang semakin merenggang, sehingga memberi asam lambung akses untuk berbalik ke kerongkongan.

4. Kehamilan

Bagian uterus yang meregang pada saat kehamilan menghimpitkan area perut dan LES. Pada saat kehamilan, perubahan hormon juga merelaksasi LES. Sistem pencernaan juga melambat karena nutrisi yang juga dicerna oleh fetus.

5. Merokok

Nikotin menyebabkan otot rileks pada LES dan kerongkongan. Maka ketika LES relaks, ia akan memberi akses ke asam lambung untuk menembus masuk ke kerongkongan. Hal tersebut menghalangi kapabilitas otot untuk mendorong makanan langsung ke perut, sehingga beresiko menyebabkan makanan tersangkut. Merokok juga meningkatkan produksi asam lambung.

6. Inflamasi Kronis

Inflamasi sebagai respon terhadap proses penyembuhan tubuh. Menjadi pertahanan tubuh terhadap benda asing, luka, atau infeksi. Hal tersebut biasanya diakibatkan oleh konsumsi obat atau suplemen yang mengiritasi esofagus dan menyebabkan *heartburn*, refluks lambung

2.6.2 Cara mengatasi GERD

Menurut Nandi (2018), berikut adalah cara-cara untuk mengatasi penyakit asam lambung GERD:

1. Menghindari makanan pemicu:

a. Kopi

Kafein pada kopi dan teh memiliki konten acid yang tinggi dan dapat melonggarkan otot LES.

b. Soda

Konten soda yang kaya akan kafein, karbonasi dan tingkat acid yang tinggi dapat menyebabkan perut kembung akibat penekanan pada otot LES dan dapat berakibat fatal jika terus menerus dikonsumsi.

c. Minuman berenergi

Minuman berenergi memiliki konten *acid citrus* dan kafein yang tinggi.

d. Coklat

Coklat memiliki kadar lemak, kafein, dan *theobromine* yang dapat melonggarkan otot LES dan meningkatkan refluks.

e. Makanan yang digoreng dan berlemak

Makanan yang digoreng dan berlemak lebih sulit untuk dicerna, karena menekan otot LES dan meningkatkan kemungkinan *acid reflux*.

f. Daging merah berlemak tinggi

Daging merah berlemak tinggi lebih sulit dicerna dan bertahan di perut lebih lama dibandingkan makanan lain, sehingga otot LES tertekan. Sirloin dan tenderloin dapat menjadi alternatif yang lebih baik.

g. Bawang putih dan bawang bombay

Bawang putih dan bawang bombay memiliki kandungan *allium* yang dapat memicu *heartburn* pada penderita GERD.

h. Tomat dan saus tomat

Tomat memiliki kadar *acid* yang tinggi.

i. Mint

Kadar mint yang tinggi dapat merelaksasi LES dan menyebabkan *acid reflux*.

j. Buah dan Jus Jeruk

Buah dan jus jeruk memiliki kadar *acid* yang tinggi.

k. Produk susu

Produk dengan bahan susu memiliki kadar lemak yang tinggi. Makanan dengan kadar lemak yang tinggi bertahan lebih lama di organ perut, sehingga memberatkan otot LES untuk bekerja.

l. Makanan pedas

Makanan pedas seperti tabasco, lada hitam, paprika merah, bubuk cabai dan makanan pedas lainnya dapat menyebabkan *heartburn*.

m. Alkohol

Alkohol dapat merelaksasi otot LES dan mengakibatkan *heartburn*.

2. Makan dengan porsi kecil

Hindari mengkonsumsi makanan dalam jumlah besar sekaligus, karena konsumsi makanan akan mengakibatkan renggangnya perut dan menekan otot LES yang beresiko meningkatkan asam lambung menuju esofagus. Konsumsi makanan dengan porsi kecil secara bertahap, serta mengunyah perlahan-lahan dan menyeluruh.

3. Hindari kudapan saat malam hari

Beri waktu 3 jam sebelum tidur malam untuk meminimalisir kemungkinan *heartburn*. Sistem pencernaan akan bekerja lebih efisien karena dapat mengosongkan diri ketika dalam posisi tidur.

4. Menjaga berat badan yang sehat

Hindari obesitas dengan menjaga berat badan dan gaya hidup yang sehat. Salah satunya dengan menghindari konsumsi makanan berlebihan dan makanan pemicu GERD.

5. Hindari berbaring setelah makan

Tunggu sekitar 3 jam sebelum berbaring agar sistem pencernaan tidak terganggu.

6. Menggunakan pakaian yang longgar

Menggunakan pakaian ketat dapat menekan area perut yang dapat meningkatkan resiko *acid reflux* serta *heartburn*. Maka dari itu disarankan untuk memprioritaskan kenyamanan tubuh saat berpakaian.

7. Menaikkan posisi kepala saat tidur

Dalam keadaan berbaring, produksi asam lambung akan semakin meningkat dan tidak dapat bekerja dengan baik di dalam organ perut.

8. Berhenti merokok

Merokok merupakan sebab utama *heartburn* dan meningkatnya asam lambung, sehingga tindakan preventif untuk berhenti dapat memberi perubahan signifikan untuk penyakit GERD. Konsumsi obat juga kerap dilakukan untuk mengatasi adiksi terhadap kebiasaan merokok yang disesuaikan dengan pengobatan GERD.

9. Hindari stress

Sistem pencernaan akan semakin rentan dengan asam lambung dengan semakin meningkatnya keadaan stress seseorang, yang memicu meningkatnya produksi asam lambung. Stress dapat dihindari dengan melakukan kegiatan yang dapat meminimalisir stress, seperti meditasi, berdoa, tai chi, *massage*, dan yoga secara rutin.