Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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

LANDASAN TEORI

2.1. Dehidrasi

Dehidrasi didefinisikan sebagai kondisi dimana tubuh kehilangan cairan yang diperlukan agar dapat menjalankan fungsi-fungsinya dengan normal (MIMS Pte Ltd, 2013). Dehidrasi adalah suatu keadaan dimana terjadi ketidakseimbangan kadar elektrolit dalam tubuh akibat dari hilangnya cairan atau air (Singh, 2015). Dehidrasi terjadi karena pengeluaran air lebih banyak daripada jumlah yang masuk, dan kehilangan cairan ini juga disertai dengan hilangnya elektrolit.

Pada dehidrasi, terjadi keseimbangan negatif cairan tubuh akibat penurunan asupan cairan dan meningkatnya jumlah air yang keluar (lewat ginjal, saluran cerna atau *insensible water loss*/IWL), atau karena adanya perpindahan cairan dalam tubuh. Berkurangnya volume total cairan tubuh menyebabkan penurunan volume cairan intrasel dan ekstrasel. Manifestasi klinis dehidrasi erat kaitannya dengan deplesi volume cairan intravaskuler. Proses dehidrasi yang berkelanjutan dapat menimbulkan syok *hipovolemia* yang akan menyebabkan gagal organ dan kematian (Leksana, 2015).

Berdasarkan perbandingan jumlah natrium dengan jumlah air yang hilang, dehidrasi dibedakan menjadi tiga tipe yaitu dehidrasi isotonik, dehidrasi hipertonik, dan dehidrasi hipotonik. Variasi kadar natrium mencerminkan jumlah cairan yang hilang dan memiliki efek patofisiologi berbeda.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

1. Dehidrasi isotonik (isonatremik). Tipe ini merupakan yang paling sering (80%). Pada dehidrasi isotonik kehilangan air sebanding dengan jumlah natrium yang hilang, dan biasanya tidak mengakibatkan cairan ekstrasel berpindah ke dalam ruang intraseluler. Kadar natrium dalam darah pada dehidrasi tipe ini 135-145 mmol/L dan osmolaritas efektif serum 275-295 mOsm/L.
2. Dehidrasi hipotonik (hiponatremik). Natrium hilang yang lebih banyak daripada air. Penderita dehidrasi hipotonik ditandai dengan rendahnya kadar natrium serum (kurang dari 135 mmol/L) dan osmolalitas efektif serum (kurang dari 270 mOsml/L). Karena kadar natrium rendah, cairan intravaskuler berpindah ke ruang ekstravaskuler, sehingga terjadi deplesi cairan intravaskuler. Hiponatremia berat dapat memicu kejang hebat; sedangkan koreksi cepat hiponatremia kronik (2 mEq/L/jam) terkait dengan kejadian mielinolisis pontin sentral.
3. Dehidrasi hipertonik (hipernatremik). Hilangnya air lebih banyak daripada natrium. Dehidrasi hipertonik ditandai dengan tingginya kadar natrium serum (lebih dari 145 mmol/L) dan peningkatan osmolalitas efektif serum (lebih dari 295 mOsm/L). Karena kadar natrium serum tinggi, terjadi pergeseran air dari ruang ekstravaskuler ke ruang intravaskuler. Untuk mengkompensasi, sel akan merangsang partikel aktif (idiogenik osmol) yang akan menarik air kembali ke sel dan mempertahankan volume cairan dalam sel. Saat terjadi rehidrasi cepat untuk mengoreksi kondisi hipernatremia, peningkatan aktivitas osmotik sel tersebut akan

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

menyebabkan influks cairan berlebihan yang dapat menyebabkan pembengkakan dan ruptur sel edema serebral adalah konsekuensi yang paling fatal. Rehidrasi secara perlahan dalam lebih dari 48 jam dapat meminimalkan risiko ini. (Leksana, 2015)

Tanda pertama dehidrasi yaitu rasa haus yang mudah diatasi dengan memberikan minum. Jika hilangnya cairan tubuh tidak diganti dengan apapun dalam beberapa hari, akan berkembang menjadi dehidrasi berat yang membutuhkan penanganan medis segera. Kondisi seperti berkeringat banyak, *heat stroke* berat, diare berat, atau muntah-muntah, diabetes melitus, serta diabetes insipidus seringkali menyebabkan dehidrasi. Jika tidak diatasi, dehidrasi dapat mengakibatkan kerusakan ginjal, syok, bahkan kematian. (MIMS Pte Ltd, 2013)

Dalam mengetahui tingkat dehidrasi yang diderita, salah satunya dapat dilakukan dengan cara mengukur tingkat dehidrasi dengan menggunakan table derajat dehidrasi. Berikut adalah tabel derajat berdasarkan Maurice King dan tabel derajat WHO

2.1.1. Derajat Dehidrasi WHO

Tabel 2.1 Derajat Dehidrasi Berdasarkan (World Health Organization, 2005)

	A	B	C
Keadaan Umum	Baik, sadar	Gelisah, rewel	Lesu, lunglai atau tidak sadar
Mata	Normal	Cekung	Sangat Cekung
Air Mata	Ada	Tidak Ada	Kering
Mulut dan Lidah	Basah	Kering	Sangat Kering

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

Rasa Haus	Minum biasa, tidak haus	Haus, ingin minum banyak	Malas minum atau tidak bisa minum
Periksa: Turgor Kulit	Kembali cepat	Kembali lambat	Kembali sangat lambat
Hasil Pemeriksaan	Tanpa Dehidrasi	Dehidrasi Ringan / Sedang	Dehidrasi Berat
Terapi	*Plan A*	*Plan B*	*Plan C*

Perhitungan skor derita dehidrasi pada tabel 2.1 adalah apabila terdapat kurang dari 2 tanda pada kolom B dan C maka masuk kedalam kategori Tanpa Dehidrasi. Sedangkan apabila terdapat lebih dari 2 tanda pada kolom B maka masuk kedalam kategori Dehidrasi Ringan/ Sedang. Dan apabila terdapat 2 atau lebih tanda pada kolom C maka masuk kedalam kategori Dehidrasi Berat (Leksana, 2015).

2.1.2. Skor Maurice King

Tabel 2.2 Tabel Skor Maurice King

Bagian Tubuh yang Diperiksa	**Nilai**		
	0	1	2
Keadaan umum	Sehat	Gelisah, cengeng, ngantuk, apatis	Mengigau, koma/syok
Elastis kulit	Normal	Sedikit kurang	Sangat kurang
Mata	Normal	Cekung	Sangat cekung
Ubun-ubun besar	Normal	Sedikit kurang	Sangat kurang
Mulut	Normal	kering	Kering & Sianosis

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

Denyut nadi /menit	>120	Sedang (120-140)	>140

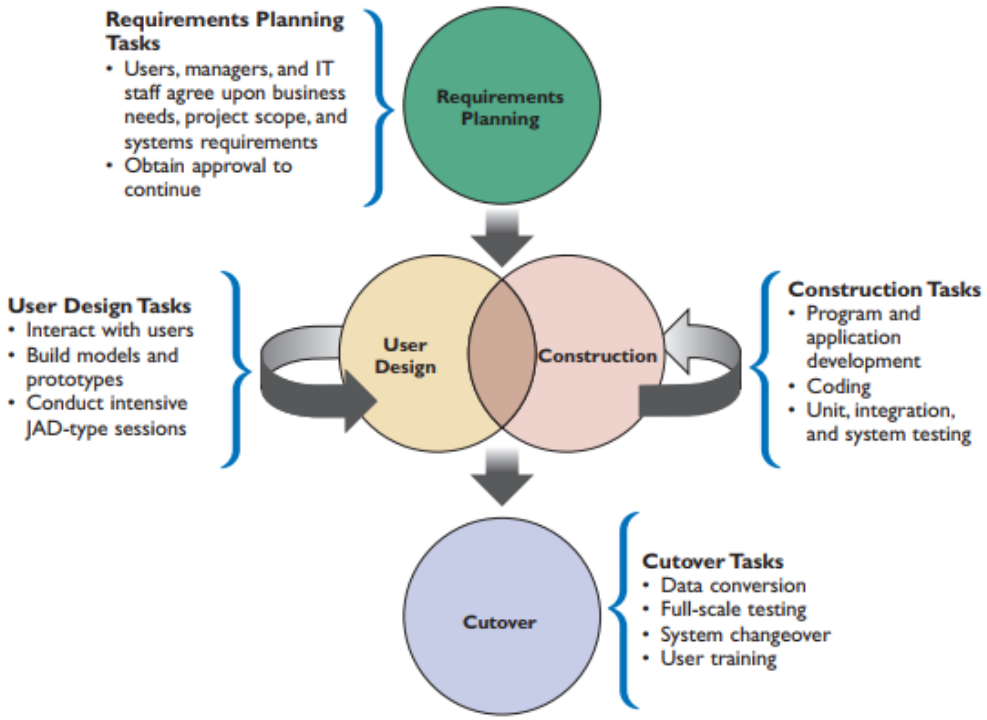
Berdasarkan tabel 2.2, untuk menentukan elastisitas kulit, kulit perlu dicubit selama 30-60 detik, kemudian dilepas. Jika bentuk kulit kembali normal dalam waktu 2 sampai 5 detik menandakan indikasi dehidrasi ringan, 5 sampai 10 detik menandakan dehidrasi sedang, dan jika kulit kembali dalam waktu lebih dari 10 detik maka tingkat dehidrasi yang diderita yaitu dehidrasi tinggi.

Berdasarkan skor yang ditemukan pada penderita, dapat ditentukan derajat dehidrasinya, yaitu dehidrasi ringan (skor 0 sampai 2), dehidrasi sedang (skor 3 sampai 6), dan dehidrasi berat (skor >7) (Iswari, 2011).

2.2. *Rapid Application Development* (RAD)

Rapid application development (RAD) adalah teknik atau metode pengembangan sistem yang membutuhkan waktu relatif cepat dalam menghasilkan output sistem informasi yang berfungsi dengan baik. RAD biasa digunakan oleh perusahaan untuk mengurangi biaya dan waktu pengembangan suatu sistem serta meningkatkan probabilitas keberhasilan. Proses RAD memungkinkan pengguna untuk memeriksa model kerja sedini mungkin, kemudian menentukan apakah model sudah memenuhi kebutuhan serta menyarankan perubahan yang diperlukan. Model tersebut kemudian dimodifikasi sesuai dengan keinginan *user* dan proses interaktif berlanjut sampai sistem benar-benar berkembang dan *user* merasa puas.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018



Gambar 2.1 Empat fase RAD (Sumber : Shelly & Roseblatt, 2012)

Sesuai dengan gambar 2.1, model RAD memiliki 4 fase, yaitu: (Shelly & Roseblatt, 2012)

1. *Requirement Planning*

   Pada fase ini, *user* melakukan pembuatan *business needs*, lingkup proyek, dan keperluan sistem. Fase *requirement planning* akan berakhir apabila semua kebutuhan telah diketahui dengan benar.

2. *User Design*

   Pada fase ini, dilakukan perancangan model sesuai dengan kebutuhan yang telah didiskusikan pada fase sebelumnya.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

3. *Construction*

   Fase *construction* berfokus pada pembangunan sistem atau aplikasi. Fase ini mirip dengan SDLC namun pada model RAD, tetap memungkinkan adanya perubahan atau *improvement* pada sistem.

4. *Cutover*

   Fase terakhir dari metode RAD adalah untuk melakukan *review* serta *testing* sistem atau aplikasi. Apabila sistem tersebut sudah lolos tahap pengujian maka dilakukan *user training* sehingga hasil nya berupa sistem yang telah dibangun dan didistribusikan dengan baik dan cepat.

2.3. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah metode visualisasi dan pendokumnetasian desain sistem perangkat lunak yang banyak digunakan. UML mirip seperti DFD, namun UML menggunakan konsep *object-oriented analysis and design* (OOAD) sedangkan DFD menggunakan konsep struktural. UML digunakan untuk menggambarkan proses bisnis dan kebutuhan *user* secara umum, desain UML juga dapat ditulis dalam bahasa pemrograman apapun. Beberapa *tools* UML yang dapat digunakan untuk pembuatan *user requirements* adalah (Shelly & Roseblatt, 2012):

2.3.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case diagram* dapat sangat membantu untuk menyusun

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

requirement sebuah sistem, mengkomunikasikan rancangan dengan *user*, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Terdapat tiga simbol dalam *use case diagram*, yaitu:

1. Oval

Gambar 2.2 Simbol oval pada *use case diagram*

Simbol oval seperti pada gambar 2.2 pada *use case diagram* digunakan untuk mendeskripsikan sebuah *action* atau *event*.

2. *Stick Figure*

Gambar 2.3 Simbol aktor pada *use case diagram*

Stick Figure pada gambar 2.3 diatas digunakan untuk mendeskripsikan aktor yang berperan dalam suatu sistem atau aplikasi.

3. *Line*

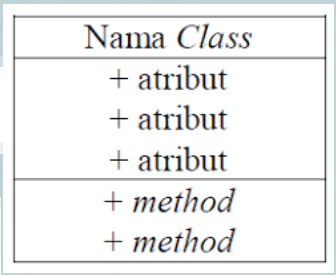
Gambar 2.4 Simbol *line* pada *use case*

Line atau disebut juga dengan asosiasi seperti pada gambar 2.4 digunakan untuk menghubungkan aktor dengan *action* yang dapat dikerjakan.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

2.3.2. Class Diagram

Class diagram merupakan *logical* model yang menunjukkan objek dan relasi yang terdapat pada *use case diagram*. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.



Gambar 2.5 Komponen *class diagram*

Tiga komponen utama dalam *class diagram* yaitu nama, atribut, dan metode seperti pada gambar 2.5. Terdapat *line* yang menunjukkan relasi antar kelas dan label yang mengidentifikasi *action* antar dua kelas.

2.3.3. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang seperti sebuah *flowchart*. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

2.4. Android

Android merupakan *platform* perangkat lunak dan sistem operasi yang diperuntukkan bagi perangkat *mobile*. Dibangun dengan menggunakan Linux, android dikembangkan oleh Google sejak tahun 2005 hingga akhirnya resmi diluncurkan pada tahun 2007. Android merupakan perangkat lunak *open source*

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

yang memungkinkan *developer* untuk mengembangkan aplikasi-aplikasi yang dapat dijalankan pada system operasi tersebut. Adapun bahasa pemrograman yang biasa digunakan oleh para *developer* untuk membangun sebuah aplikasi android adalah menggunakan *Java Programming Language* (Verma, Arora, & Verma, 2017).

2.5. User Interface

2.5.1. The Eight Golden Rules

Ben Shneiderman mengemukakan 8 *golden rules principle* yang merupakan sebuah aturan *interface* untuk menyempurnakan implementasi sebuah sistem dalam hal berinteraksi dengan manusia. Adapun, 8 *golden rules* terdiri dari: (Shneiderman & Plaisant, 2018)

1. *Strive for consistency*

Konsisten dalam penggunaan jenis huruf, ukuran huruf, dan spasi serta peletakan merupakan hal yang terpenting untuk membuat sebuah aplikasi terlihat baik. Pemilihan warna yang akan digunakan pada aplikasi juga harus diperhatikan dan disesuaikan dengan konten dari aplikasi tersebut. Selain itu, *user* biasanya tidak ingin mempelajari hal baru yang membuat waktu *user* terbuang karena menggunakan aplikasi.

2. *Cater to universal usability*

Memperhatikan sasaran dari aplikasi, baik itu dari usia maupun jenis kelamin. Selain itu juga harus memperhitungkan apakah *user* merupakan seseorang yang sudah biasa menggunakan teknologi atau *user* pemula.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

3. *Offer informative feedback*

Adanya fitur yang memungkinkan aplikasi untuk memberikan *feedback* ketika terjadi sesuatu pada saat pengoperasian aplikasi. Contohnya jika *user* menekan tombol *close*, aplikasi sebaiknya menampilkan *popup alert* yang digunakan untuk mengkonfirmasi apakah *user* benar-benar ingin keluar dari aplikasi ataukah hanya salah menekan tombol.

4. *Design dialogs to yield closure*

Aplikasi yang baik juga harus mempunyai perbedaan tampilan untuk memberi tahu pengguna aktifitas yang sedang ia lakukan dalam aplikasi tersebut. Misalnya jika *user* melakukan *login* ke dalam aplikasi, tampilan sebelum *login* dan setelah *login* harus dibuat berbeda agar memudahkan *user* untuk mengetahui apakah ia sudah masuk atau belum.

5. *Prevent errors*

Aplikasi yang baik hendaknya memungkinkan pemberian solusi kepada para *user* ketika *user* melakukan kesalahan. Seperti adanya fitur untuk melakukan *backup* data.

6. *Permit easy reversal of actions*

Human *error* merupakan hal yang tidak dapat dihindari, oleh karena itu dalam sebuah aplikasi, harus ada fitur yang memungkinkan *user* untuk melakukan perubahan atau menghapus hal yang sudah dilakukan sebelumnya.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

7. *Support internal locus of control*

Aplikasi harus memiliki fitur yang memungkinkan *user* untuk mengubah informasi akun atau data diri yang dimilikinya sesuai dengan yang dikehendaki.

8. *Reduce short term memory load*

Pada poin ini adalah mengutamakan konsistensi tata letak setiap komponen pada aplikasi agar memudahkan *user* dalam mengingat tampilan pada satu halaman yang digunakan pada halaman lain.

2.5.2. Android Rules for User Interface

Perancangan antar muka pada aplikasi mobile, baik itu android maupun iOS haruslah memikirkan interaksi dengan *user*. Berikut adalah beberapa panduan perancangan *user interface* berdasarkan *website android developer* (AndroidDeveloper, 2018):

1. Implementasi Navigasi yang Efektif
   - Penggunaan *Back Navigation* yang baik

     Back Button pada aplikasi android sebenarnya tidak dibutuhkan, karena pada dasarnya *handphone* android sendiri sudah menyediakan *Back Button* pada perangkat nya. Untuk itu, perlu diketahui bahwa *Back Button Navigation* yang dimaksud disini biasanya disiapkan apabila *user* berada pada *WebView* dimana *Back Button* pada aplikasi difungsikan untuk menuju ke *web page* yang dikunjungi sebelumnya.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018

- Pembuatan *Swipe Views* dengan *Tab*

  Swipe View pada aplikasi android memudahkan *user* untuk mengganti tab yang berisi konten halaman yang ditampilkan pada *screen* dengan cara menggeser layer secara *horizontal*.

2. Pemberian Notifikasi

   - *Update* Notifikasi

     Aplikasi yang dapat memberikan notifikasi secara berkala untuk satu aktifitas yang sama haruslah memperhatikan untuk melakukan *update* terhadap notifikasi yang lama dibandingkan dengan membuat notifikasi baru lagi.

   - Menampilkan *Progress* Pada Notifikasi

     Beberapa notifikasi biasanya mengandung *animated progress bar* yang berfungsi untuk memberi tahu *user progress* yang telah dibuat oleh aplikasi tanpa *user* harus membuka aplikasi tersebut. Jenis notifikasi ini biasanya digunakan apabila *user* melakukan pengunduhan data dari aplikasi.

3. Mendukung Fitur *Swipe to Refresh*

Aplikasi memungkinkan *user* untuk melakukan *refresh* secara manual meskipun pada dasarnya aplikasi tersebut melakukan *refresh* secara berkala. Hal ini dapat berguna untuk aplikasi seperti prediksi cuaca, dll. *Refresh* dilakukan dengan cara melakukan *swipe* secara *vertical* pada halaman yang ingin di *refresh*.

Rancang bangun aplikasi..., Putri Hanna Belinda, FTI UMN, 2018