

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

User Experience (UX) adalah sebuah kualitas pengalaman yang dimiliki seorang pengguna saat sedang berinteraksi dengan suatu sistem yang spesifik. Sistem tersebut dapat berupa hanya sebuah desain. Desain dalam hal ini mencakup berbagai macam hal, sebagai contoh desain pintu, produk, bahkan website dan sebuah sistem operasi yang menyangkut dengan penggunaan oleh seorang manusia. Menurut Tom tullis dan Bill Albert dalam buku *Measuring The User Experience, User Experience* melihat keseluruhan interaksi individual terhadap sesuatu, mencakup pemikiran, perasaan dan persepsi dari hasil interaksi yang dilakukan tersebut. Dari pemahaman mengenai *User Experience* tersebut maka penulis memahami nilai dari *user experience* menjadi sangat penting dalam pembuatan suatu sistem. Penelitian banyak dilakukan untuk meeningkatan *user experience* agar user dapat lebih dipermudah dalam menyelesaikan tujuannya saat menggunakan sistem. Penggunaan *Augmented Reallity* adalah salah satu solusi dalam meningkatkan *user experience*. *Augmented Reallity* menyediakan suatu objek secara tidak nyata (*virtual*) lalu menggabungkannya dengan dunia nyata secara simulasi, hal ini tentu menyebabkan terjadinya interaksi yang berbeda dan dapat menyebabkan peningkatan user experience.

Dalam penelitiannya, Nyoman Aditya G.S. (2012) menyimpulkan bahwa penggunaan *Augmented Reallity* dapat meningkatkan *User Experience*. Peningkatan yang terjadi adalah pada 3 dari 5 matrik *User Experience* yaitu kegunaan sistem (*Usefulness*), kepuasan pengguna (*Satisfaction*) dan keyakinan mengambil keputusan (*Decision Confidence*) sedangkan matrik yang menurun kemudahan penggunaan sistem (*Ease of Use*) dan kemudahan belajar sistem (*Ease of Learning*). Penggunaan

Self Reported Metrics dan kuisioner dilakukan oleh peneliti untuk mengevaluasi sistem yang telah dibuat. Peneliti menekankan penelitiannya pada salah satu faktor utama *User Experience* yaitu *Usability*. Peneliti menekankan penelitian pada bagian *Usability*, dengan kata lain penekanan penelitian menggunakan *Usability Testing*.

Pada penelitian oleh So-Yeon (2008) mengenai *User Experience* pada teknologi Web 3D yang digunakan dalam demonstrasi produk dan bagaimana faktor *Usability* dapat dipengaruhi oleh interaksi pengguna terhadap sistem. Dalam penelitian ini So-Yeon dkk membandingkan *User Experience* dari Web 3D dengan Web 2D konvensional yang diaplikasikan pada website demonstrasi produk furniture. Penelitian ini mengukur *User Experience* dengan melakukan penilaian terhadap 3 aspek utama yaitu *usability*, *technology acceptance*, dan *presence*. Untuk mengukur *usability* digunakan beberapa metode tetapi tetap mengarah pada 2 kategori yaitu *:objective performance measure* mengukur seberapa mampu pengguna dalam menggunakan sistem dan *subjective user preference/satisfaction measure* mengukur seberapa puas pengguna dalam menggunakan sistem. Sedangkan untuk mengukur *User Acceptance of technology* peneliti menggunakan *Technology acceptance model (TAM)*. Dari ke 3 aspek utama penilaian tersebut So-Yeon mengatakan bahwa Web 3D lebih efektif dari Web 2D ketika mendemonstrasikan produk yang dapat dikustomisasi atau dapat dipasangkan dengan berbagai bentuk, warna atau bahan yang tergantung dari selera pasar. *User Experience* dalam berinteraksi dengan Web 3D juga berbeda dari interaksi dengan Web 2D konvensional sehingga menghasilkan *usability* yang lebih baik. Visualisasi 3D menstimulasi *sense of presence* pengguna yang merupakan kombinasi dari interaktifitas, kejelasan, kenikmatan yang membuat pengguna secara aktif terikat pada evaluasi produk dibandingkan 2D yang lebih pasif.

Pada jurnal yang ditulis oleh Maryam Moayeri (2010), mengenai pengambilan data secara *online* menggunakan perangkat lunak penguji *usability*, membahas keunggulan pengambilan data yang dilakukan secara *online* daripada pengambilan data secara tradisional (langsung). Metode pengambilan data secara *online* ini dapat

menjadi solusi atas kesulitan yang terjadi saat mengambil, menganalisa dan menampilkan data dari pengguna. Penggunaan perangkat lunak untuk mengambil data pengukuran yang diperlukan dapat mempermudah peneliti untuk melakukan perhitungan cepat seperti berapa lama waktu yang dibutuhkan saat mengakses situs, dan seberapa sering pengguna mengklik pada suatu halaman. Hanya dengan penggunaan satu tombol bisa merekam semua aktifitas pengguna untuk data dalam penelitian. Perangkat lunak yang digunakan bisa menampilkan grafik atau chart atau gambar lainnya yang dapat mempermudah peneliti melakukan penampilan (presentasi) data.

Dari beberapa jurnal dan karya ilmiah di atas, penulis dapat menarik 3 kesimpulan yaitu :

1. *Augmented Reality* dapat digunakan untuk meningkatkan *User Experience*.
2. Dalam pengukuran *User Experience* dapat dilakukan dengan menggunakan *Usability Testing*.
3. Pengambilan data secara online menggunakan suatu perangkat lunak atau sistem dapat lebih efektif dalam pencarian, penghitungan dan bahkan penyajian data untuk mencapai hasil dalam penelitian.

Dari 3 kesimpulan di atas, maka sistem pengukuran *User Experience* dibuat untuk menjadi suatu alat agar dapat membantu penulis dalam mengukur suatu peningkatan *User Experience* yang terjadi pada sebuah sistem, dan dalam penggunaannya akan menggunakan prinsip dari *Usability Testing*.

2.2 Landasan Teori

2.2.1 *Augmented reality* adalah variasi dari *Virtual Environments* (VE) yang menggabungkan lingkungan dunia nyata dengan informasi virtual

(lingkungan virtual) untuk mensimulasikan kehadiran benda yang tidak ada di pengguna secara nyata

Aplikasinya sudah banyak dilakukan antara lain :

1. Dalam dunia medik : *Augmented reality* telah digunakan untuk visualisasi dan latihan untuk pembedahan
2. Produksi, perakitan dan perbaikan suatu produk: *Augmented reality* telah membantu dalam proses perancangan suatu produk yang belum ada, misal dengan desain 3 dimensi sehingga mempermudah langkah-langkah dalam proses produksi. contohnya adalah Boeing yang menggunakan teknologi *augmented reality* untuk membimbing mekanik untuk membangun dalam sistem kelistikan pesawat.
3. Memberi keterangan dan visualisasi : *Augmented reality* dapat digunakan untuk memberikan keterangan ataupun lingkungan dengan informasi umum atau rahasia. Contohnya pada ECRC (*European Computer-industry Research Centre*) pengguna dapat menunjuk bagian dari model mesin dan sistem akan menunjukkan nama dan info bagian yang ditunjuk.
4. Perencanaan jalur robot : mengoperasikan robot pada dunia nyata akan sulit jika jaraknya jauh, karena jeda (delay) pada komunikasi, maka dibuatlah virtualisasinya terlebih dahulu, sehingga ketika virtualisasi sudah memberikan hasil, robot bisa diperintah tanpa membuang waktu karena kesalahan.
5. Entertainment : *augmented reality* mulai dijadikan cara baru dalam memainkan video game, contoh nyatanya adalah konsol *video game* dari Microsoft, Xbox dengan Kinect sebagai input dan implementasi *augmented reality* nya

6. Pesawat militer : *augmented reality* digunakan pada tampilan helm pilot, untuk membidik senjata pada pesawat. (Ronald T Azuma, 1997:1-9).

Dengan demikian, penulis mengambil kesimpulan bahwa *augmented reality* telah banyak membantu dalam proses virtualisasi dan meningkatkan *user experience* pada setiap contoh di atas.

2.2.2 Interaksi Manusia dan Komputer (IMK)

Interaksi terjadi di antara manusia dan komputer untuk mendapatkan suatu hasil yang diinginkan tepat waktu

Istilah *Human Computer Interaction (HCI)* atau Interaksi Manusia dan Komputer (IMK) mulai muncul pertengahan tahun 1980-an sebagai bidang studi yang baru. Tujuan IMK adalah perancangan dan evaluasi antarmuka pemakai (*user interface*) yang berdasarkan prinsip manfaat (*usable*) artinya sistem tersebut dapat berfungsi dengan baik, aman (*safe*) artinya sistem tersebut bisa untuk mengembangkan dan meningkatkan keamanan (*safety*), utilitas (*utility*), mengacu kepada fungsionalitas sistem atau sistem tersebut dapat meningkatkan efektifitas dan efisiensi kerjanya, ketergunaan (*usability*) disini dimaksudkan bahwa sistem yang dibuat tersebut mudah digunakan dan mudah dipelajari baik secara individu ataupun kelompok. efektifitas (*effectiveness*) dan efisiensinya (*eficiency*).

Tujuan IMK adalah untuk membuat kita merancang produk interaktif yang akan mendukung orang banyak dalam kehidupannya sehari-hari. Setiap produk yang digunakan seseorang disebut memiliki *user experience* : koran, botol saos, dan lainnya (Yvonne Rogers, 2007:15) secara sederhana menjelaskan bagaimana manusia merasakan kepuasan ketika menggunakan produk, seperti saat mereka membuka atau menutup tutup botol kecap. Ini

mengapa kita harus menyadari pentingnya membangun desain yang baik untuk meninggalkan kesan kepada konsumen akan produk yang mereka pakai.

Berikut empat faktor dasar membangun desain interaktif bagi *user* :

- Identifikasi hal diperlukan dan mencari solusi untuk meningkatkan *User Experience*
- Membuat beberapa alternatif desain sesuai dengan requirement
- Membuat versi desain interaktif yang dapat menimbulkan interaksi dengan *User*
- Mengevaluasi kekekurangan dan melakukan pengujian kembali

Dari faktor di atas dapat disimpulkan IMK bukan hanya mengenai *user interface* yang baik, tapi juga menjadi termasuk aspek penggunaannya dan implementasi rancangannya.

2.2.3 User Experience (UX)

User Experience (UX) adalah sebuah kualitas pengalaman yang dimiliki seseorang saat berinteraksi dengan sebuah desain yang spesifik. Desain tersebut dapat berupa desain sebuah gelas, mainan, website sampai dengan desain yang lebih besar dan kompleks.

Menurut Tom Tullis dan Bill Albert dalam buku *Measuring The User Experience*, *User Experience* melihat keseluruhan interaksi individual terhadap sesuatu, mencakup pemikiran, perasaan dan persepsi dari hasil interaksi yang dilakukan tersebut.

2.2.4 Mengukur User Experience Menggunakan Usability Testing.

Menurut Tom Tullis dan Bill Albert dalam buku mereka berjudul *Measuring the User Experience :Collecting, Analyzing and Presenting*

Usability Metrics(2008:1) menyatakan pendapat mereka bahwa (*usability metrics*) adalah sarananya yang sangat baik untuk mengukur *UX* dari suatu produk. Metrik (*metric*) adalah cara mengukur atau mengevaluasi sebuah kejadian atau hal tertentu dengan kata lain bisa disebut juga tolak ukur. *Usability metrics* menunjukkan dan menjadi tolak ukur interaksi antara pengguna dan sesuatu yang digunakan, dan dinilai melalui aspek efektifitasnya (mampu menyelesaikan tugas), efisiensi (banyaknya usaha/waktu yang dibutuhkan untuk dapat menyelesaikan tugas), dan kepuasan (tingkat kepuasan pengguna saat menggunakan sistem untuk melakukan tugasnya) (Tullis & Albert, 2008 hlm 7-8)

Berikut adalah daftar *usability metrics* yang akan digunakan untuk mengukur *user experience* dan yang akan digunakan dalam sistem :

2.2.4a Performance Metrics

Performance metrics menjadi sarana terbaik sebagai metrik oleh banyak orang mengukur *user experience*, metrik ini adalah cara terbaik untuk mengevaluasi efektifitas dan efisiensi sistem yang ada (Tullis & Albert, 2008 hlm 7-8). Jika para pengguna melakukan kesalahan (*error*), sistem bisa mencatatnya, dan bisa dijadikan acuan untuk pengembangan lebih lanjut oleh pihak pengembang sistem. Jika para pengguna membutuhkan waktu sangat lebih lama untuk menyelesaikan tugas, sistem bisa mencatat waktunya dan bisa dijadikan acuan untuk pengembangan lebih lanjut. Berikut adalah yang akan digunakan pada sistem.

a.*Task-Success*: Salah satu *usability metric* adalah selesai atau tidaknya pengguna dapat menyelesaikan tugasnya dengan sistem. Cara mengukur yang akan digunakan adalah :

- Binary Success : mengukur kesuksesan suatu tugas pada sistem dan membandingkannya dengan sistem lainnya dengan cara mengukur berhasil dan tidak (1 dan 0) (Tullis & Albert, 2008 hlm 65-66). Sebagai contoh sederhana pada tabel berikut ini

Tabel 1. Tingkat kesuksesan tugas (pengguna melihat produk hingga memutuskan untuk membeli (atau tambahkan ke keranjang))

Pengunjung website	Sistem 1 (tanpa <i>augmented reality</i>)	Sistem 2 (Menggunakan <i>augmented reality</i>)
A	1	1
B	0	1
C	1	1
D	0	0
E	0	1
Rata-rata	40%	80%

NB : 1 adalah sukses (jadi membeli), 0 adalah gagal (tidak jadi membeli)

b. *Time-On-Task* : metrik yang mengacu pada waktu penyelesaian sebuah tugas. Biasanya terkait dengan efisiensi sebuah produk, pada umumnya, waktu penyelesaian tugas jika lebih cepat, lebih baik *user experience* nya. Tapi ada beberapa pengecualian, misal dalam *game*, penyelesaian *task* yang terlalu cepat akan tidak disukai oleh pengguna. Contoh lain misal mengakses halaman dengan fitur *augmented reality* dan mengakses halaman tanpa *augmented reality*, yang tanpa jelas lebih cepat, tapi belum tentu menjadi lebih baik, karena pengguna sedang memanfaatkan fitur *augmented reality*

untuk mencoba kacamata sehingga menghabiskan waktu lebih lama daripada cuma sekedar melihat gambar kacamata dengan tampilan 2 dimensi. (Tullis & Albert, 2008 hlm 74)

c. Efisiensi : Cara mengukur efisiensi bukan hanya dengan cara menghitung waktu saja, ada cara lain yaitu dengan menghitung tindakan (effort) yang dibutuhkan untuk melakukan tugas tertentu. Sebagai contoh, pada website : Yang akan dihitung adalah jumlah halaman dan banyaknya masing-masing halaman tersebut telah ditampilkan pada masing-masing sistem, baik non maupun dengan fitur berbasis *augmented reality*, lalu datanya dibandingkan satu sama lain. (Tullis & Albert, 2008 hlm 86)

2.2.4b Self-Reported Metrics

Self-Reported metrics adalah cara lain untuk mengetahui *usability* tentang sistem, yaitu dengan cara meminta langsung kepada pengguna mengatakan tentang pengalaman mereka menggunakan sistem. Penggunaan metrics dengan aturan tertentu (bisa diukur) tetap dilakukan, agar pengguna tidak memberikan pendapat secara bebas (tidak bisa diukur) contohnya : bagus/jelek, puas/tidak puas, mudah/susah. Berikut adalah acuan yang akan digunakan untuk cara menampilkan skala rating, cara pengambilan survey dan cara menyimpulkan. (Tullis & Albert, 2008 hlm 121)

a. Semantic Differential Scales : cara untuk menampilkan tingkat pengalaman pengguna, yaitu dengan menampilkan 2 hal yang berlawanan dalam bentuk skala (5-7 poin) (Tullis & Albert, 2008 hlm 124)

contoh :

Pertanyaan		1	2	3	4	5	
kinerja?	Buruk	○	○	○	○	○	Bagus

kenyamanan?	Frustrasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fun
	Susah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Mudah

b. *Post-Session Ratings* : survey nantinya akan ditampilkan setelah pengguna selesai menyelesaikan semua interaksinya dengan sistem (misal : ketika selesai mengakses situs, dan akan menutup *tab* pada *browser*, maka akan ada *pop-up* pertanyaan untuk meminta persetujuan untuk menjalankan sistem survey) (Tullis & Albert, 2008 hlm 131).

c. *Aggregating Individual Task Ratings* : untuk melihat *usability* pada *post-session* dengan mengambil keseluruhan data survey dan dirata-ratakan berdasarkan poin yang ada, lalu ditarik kesimpulan. Asumsinya adalah dengan demikian, penulis menangkap kesan terakhir dari pengalaman pengguna terhadap sistem. (Tullis & Albert, 2008 hlm 137)

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Deskripsi Sistem

Sistem yang akan dibangun terbagi menjadi 3 bagian yaitu :

1. Sistem Toko Kacamata *Online* Biasa

Sistem ini adalah sebuah situs yang berupa toko kacamata *online* yang memiliki halaman tampilan kacamata kacamata secara biasa, 2 Dimensi. Sistem ini dibangun untuk menjadi suatu acuan dasar pengukuran *user experience*. Dalam sistem ini pengguna akan dapat mengakses kacamata kacamata dengan tampilan biasa dengan mempertimbangkan bentuk kacamata sesuai keinginan pengguna.

2. Sistem Toko Kacamata *Online* dengan *Augmented Reality*

Sistem ini adalah sebuah situs yang berupa toko kacamata *online* yang memiliki halaman tampilan kacamata kacamata secara biasa, 2 Dimensi, tetapi memiliki fitur cermin *virtual* sebagai wujud dari *Augmented Reality*. Sistem ini dibangun untuk menjadi suatu acuan dalam mengukur peningkatan *user experience*. Berdasarkan landasan teori yang penulis kemukakan pada bab 2, penggunaan *Augmented Reality* dapat meningkatkan *user experience*. Pengguna akan memiliki kesempatan mengakses kacamata dan mencobanya secara *virtual* sehingga bisa membayangkan bentuk kacamata di wajah pengguna.

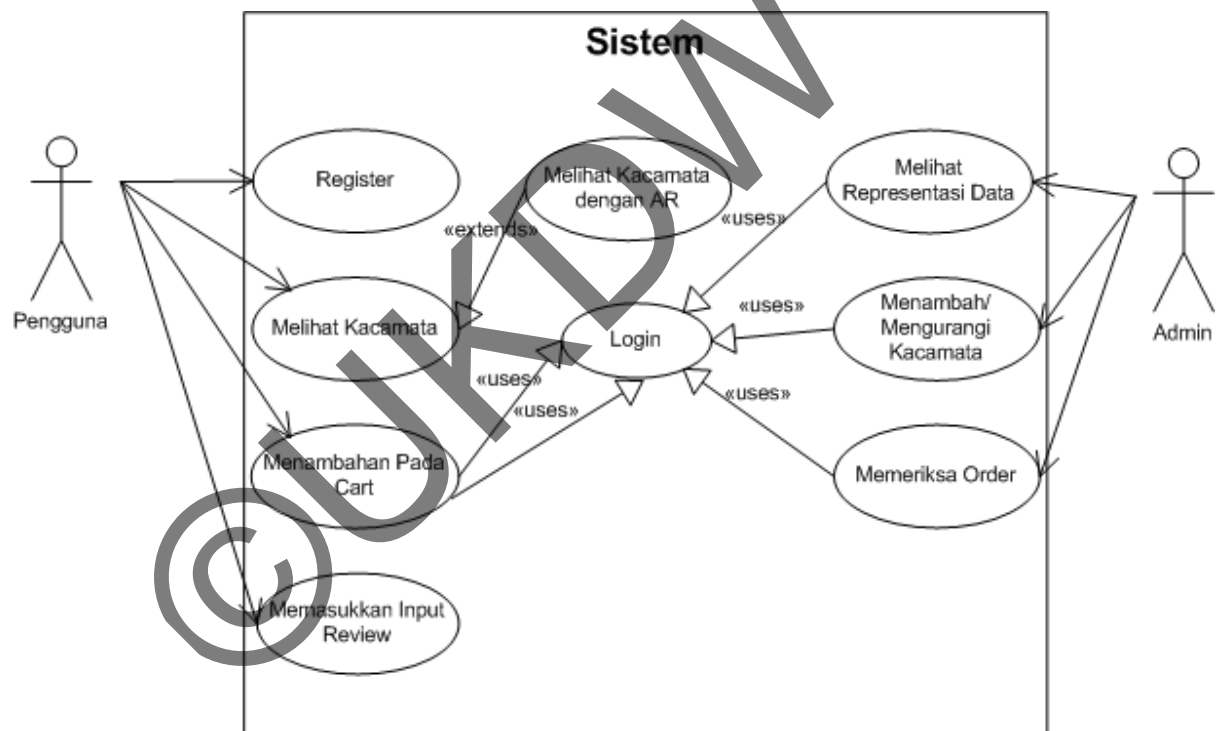
3. Sistem Pengukuran *User Experience*

Sistem ini adalah sebuah sistem yang dibangun untuk mengukur *user experience* pada kedua situs toko *online* yang dijelaskan sebelumnya. Sistem dibangun menggunakan PHP, javascript dan web-service dengan mempertimbangkan beberapa metrik-metrik pengukuran *user experience*. Sistem akan memiliki fitur merepresentasikan data dalam bentuk data *diagram* dan *chart* dalam setiap kategori metrik pengukuran, sehingga data

yang ada lebih mudah dibandingkan dari masing-masing toko *online*, karena berupa data yang tetap dan dapat diukur oleh sistem (bisa diukur menggunakan angka sebagai contoh satuan waktu).

3.2 Use Case Diagram

Pada bagian ini akan menerangkan hubungan antar sistem dengan pengguna dan sistem dengan admin. Pengguna dapat melakukan aktifitas seperti register, melihat kacamata, melakukan transaksi sedangkan admin dapat melakukan aktifitas menambah, menghapus dan mengubah informasi kacamata, melihat transaksi yang dilakukan pengguna dan dapat juga melihat data untuk pengukuran *user experience*.



Gambar 3.1 Use Case Diagram sistem

- Pengguna

1. Register : proses mendaftarkan data diri pengguna ke sistem yang ada.
2. Melihat kacamata : proses ketika pengguna memilih kacamata dan melihat-lihat kacamata yang tersedia pada sistem.

3. Menambah barang pada cart : proses ketika pengguna memilih untuk memasukkan kacamata kedalam cart (keranjang belanja)
4. Memasukkan input jempol : proses ketika selesai menggunakan sistem, akan ada masukan dari *user* untuk survei singkat menggunakan tombol jempol ke atas atau jempol ke bawah.
5. Mencoba kacamata dengan AR : proses tambahan (*extend*) dari melihat kacamata, yaitu mencoba kacamata menggunakan cermin *virtual*. (*Augmented Reality*)

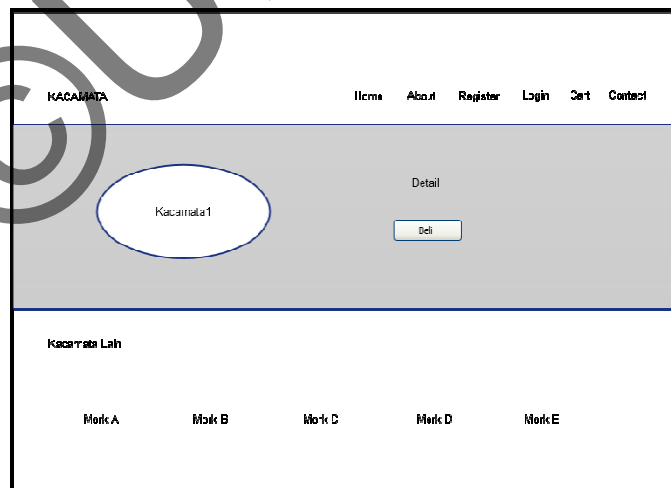
- Admin

1. Melihat representasi data : proses melihat data yang telah direpresentasikan dari sistem pengukuran *user experience* dalam bentuk tabel dan grafik.
2. Menambah kacamata : proses menambah kacamata terbaru ke dalam sistem.
3. Memeriksa *order* : proses melihat dan melakukan pemeriksaan ulang untuk proses transaksi selanjutnya.

3.3 Desain Antarmuka Sistem

3.3.1 Antarmuka Toko Online

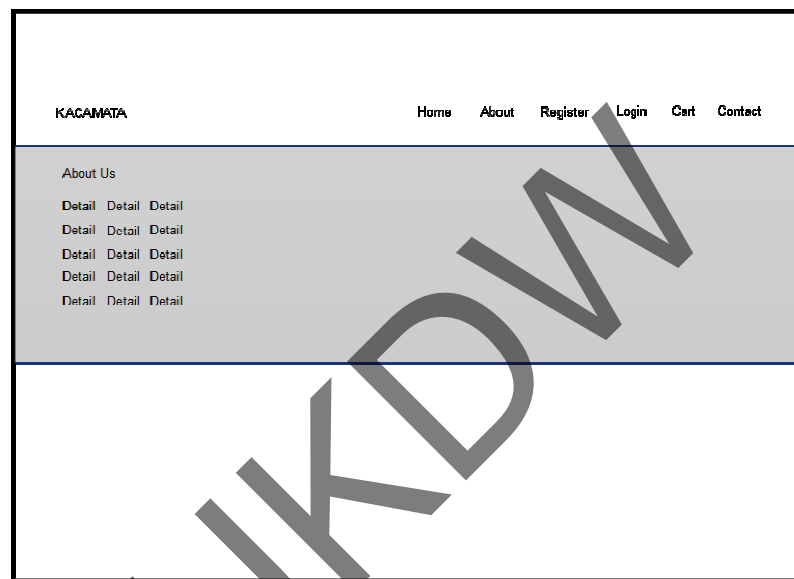
3.3.1.1 Halaman Awal



Gambar 3.2 Halaman Awal Situs Kacamata

Gambar 3.2 menampilkan rancangan antarmuka untuk halaman awal pada situs kacamata. Terdapat *header* nama toko, *link* untuk *home*, *about*, *login*, *registrasi*, *cart* dan *contact us*. Pada bagian halaman awal tersedia tempat untuk menampilkan beberapa tampilan merk kacamata yang tersedia.

3.3.1.2 Halaman About Us



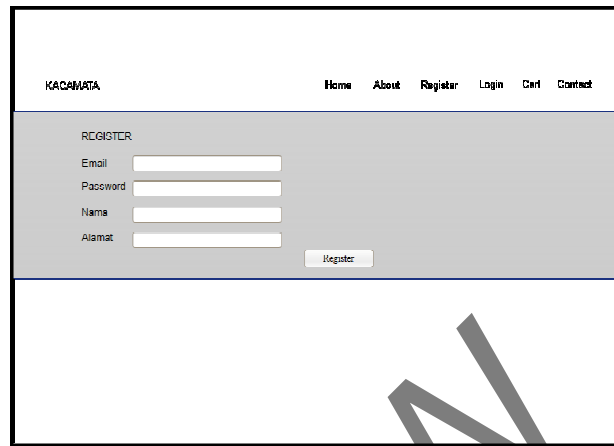
Gambar 3.3. Halaman Tentang Kami Situs Kacamata

Gambar 3.3 menampilkan rancangan antarmuka untuk halaman tentang toko yang ditampilkan pada situs kacamata yaitu berupa nama dan penjelasan singkat mengenai toko kacamata. Terdapat *header* nama toko dan *link* menuju ke halaman lainnya. (*home*, *about*, *login*, *registrasi*, *cart* dan *contact us*.)

3.3.1.3 Halaman Register

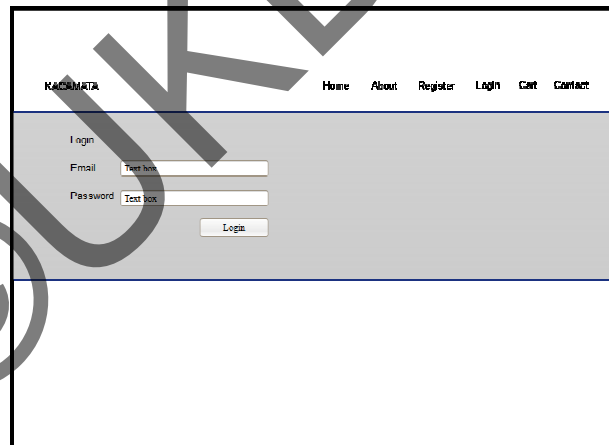
Gambar 3.4 berikut menampilkan rancangan antarmuka pada halaman pendaftaran (*Register*) pada situs kacamata. Terdapat *header* toko kacamata, form register yaitu *text box* untuk *email*, *password*, nama dan alamat serta tombol register.

Terdapat juga link menuju halaman lainnya (*home, about, login, registrasi, cart dan contact us.*)



Gambar 3.4 Halaman Register Situs Kacamata

3.3.1.4 Halaman Login

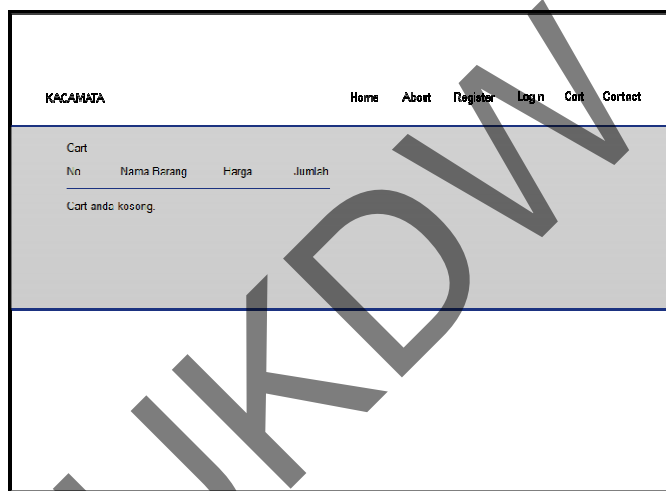


Gambar 3.5 Halaman Login Situs Kacamata

Gambar 3.5 menampilkan rancangan antar muka untuk halaman *login* pada situs kacamata. Terdapat *headernama* toko, link menuju halaman lainnya (*home, about, login, registrasi, cart dan contact us.*) dan form login yang terdiri dari *text box* untuk *email* dan *password* serta tombol *login*.

3.3.1.5 Halaman Cart

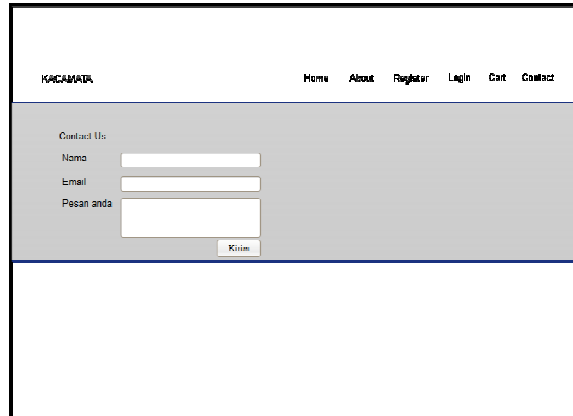
Gambar 3.6 berikut ini adalah untuk menampilkan rancangan antarmuka halaman *cart* pada situs *kacamata*. Terdapat *headernama* toko, terdapat juga link ke halaman lainnya (*home, about, login, registrasi, cart* dan *contact us.*). dan *cart* yang berupa *text* menampilkan data urutan barang, nama barang, harga dan jumlah yang telah ditambahkan ke dalam *cart*. Jika *cart* kosong, maka akan menampilkan bahwa *cart* sedang dalam posisi kosong.



Gambar 3.6 Halaman Cart Situs Kacamata

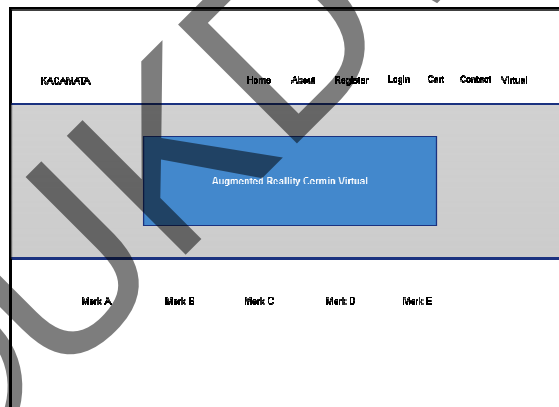
3.3.1.6 Halaman Contact Us

Gambar 3.7 berikut ini menampilkan rancangan antarmuka untuk halaman *contact us* pada situs *kacamata*. Terdapat *header* toko *kacamata*, *link* menuju halaman lainnya (*home, about, login, registrasi, cart* dan *contact us.*) dan terdapat form untuk *contact us* berupa *text box* untuk nama, email dan pesan yang akan dikirimkan serta tombol kirim. Pada halaman ini digunakan oleh *user* agar dapat mengontak toko *online* secara langsung melalui *form* yang tersedia di halaman toko *online* yang nantinya bisa di jawab oleh admin toko *online*.



Gambar 3.7 Halaman Contact Us Situs Kacamata

3.3.1.7 Halaman Virtual

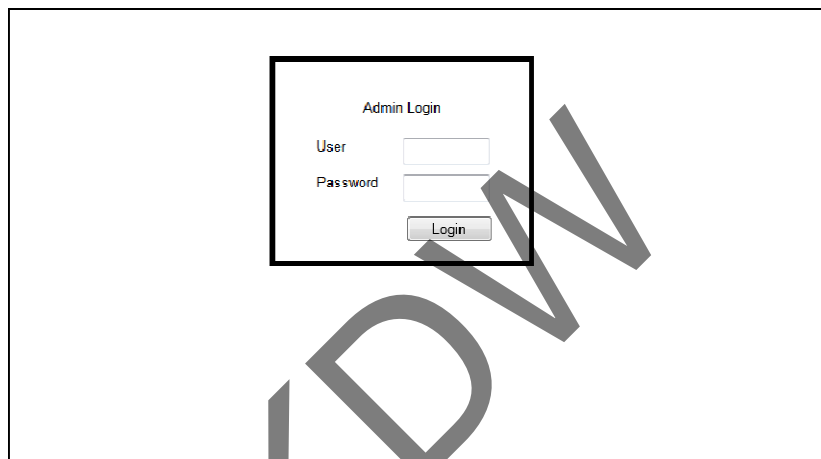


Gambar 3.8 Halaman Virtual Situs Kacamata

Gambar 3.8 menampilkan rancangan antarmuka untuk halaman *virtual* pada situs kacamata. Terdapat *header* nama toko, link menuju halaman lainnya (*home, about, login, registrasi, cart dan contact us.*), link merk yang lain dan sebuah *plugin* cermin *virtual* untuk implementasi *augmented reality*. Pada halaman ini *user* dapat mengakses fitur cermin *virtual* yang diimplementasikan sebagai bentuk *augmented reality*. Digunakan untuk mencoba kacamata secara *virtual* dan membandingkan kacamata.

3.3.1.8 Halaman Login Admin

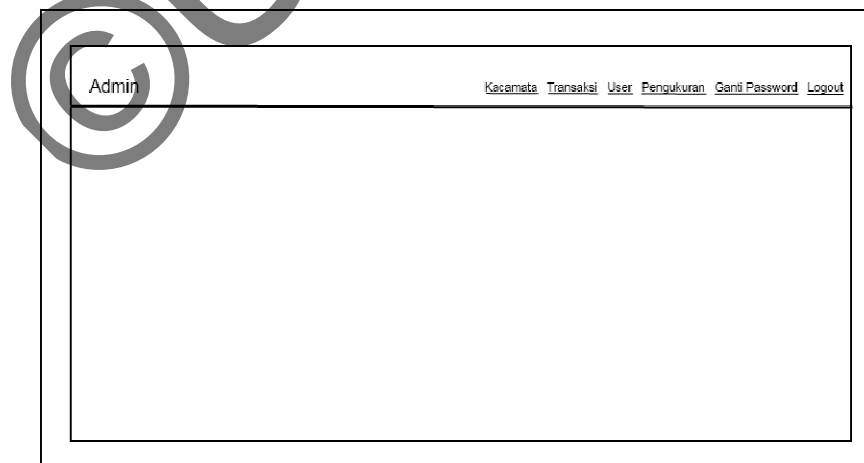
Gambar 3.9 berikut ini menampilkan rancangan antarmuka untuk halaman login admin, terdapat form yang berisi *text box* untuk mengisi *username* dan *password* serta tombol login untuk mengirimkan informasi login admin ke *server*.



The image shows a simple web form titled "Admin Login". It contains two text input fields: one labeled "User" and one labeled "Password". Below these fields is a button labeled "Login". The entire form is enclosed in a rectangular border.

Gambar 3.9 Halaman Login Admin

3.3.1.9 Halaman Awal Admin



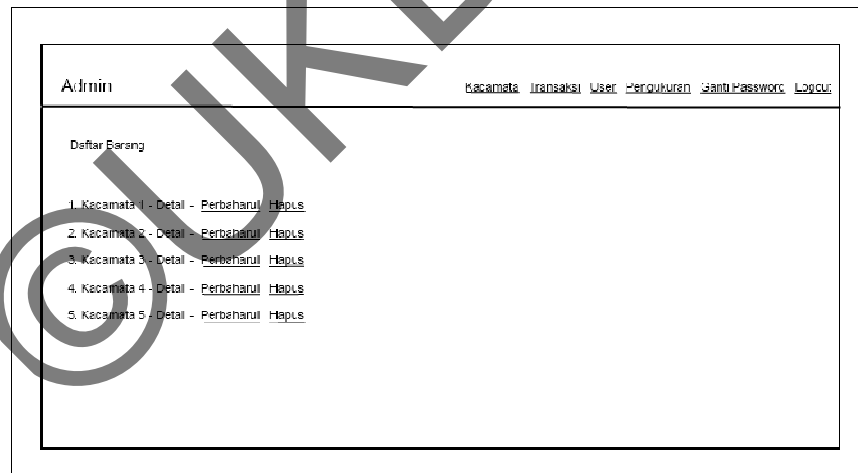
The image shows a dashboard layout for an admin user. At the top left, the word "Admin" is displayed. At the top right, there is a horizontal menu with several links: "Kacamata", "Transaksi", "User", "Pengukuran", "Ganti Password", and "Logout". The main area of the dashboard is a large, empty rectangular box.

Gambar 3.10 Halaman Awal Admin

Gambar 3.10 menampilkan rancangan antar muka untuk halaman awal admin, berisikan gambar admin sebagai *header*, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu *kacamata*, *transaksi*, *user*, *pengukuran*, *ganti password*, *password*. Serta halaman tengah yang masih kosong hanya untuk awalan admin.

3.3.1.10 Halaman Kacamata– Admin

Gambar 3.11 berikut ini menampilkan rancangan antar muka untuk halaman kacamata pada admin, berisikan gambar admin sebagai *header*, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu *kacamata*, *transaksi*, *user*, *pengukuran*, *ganti password*, *password*, lalu pada tengah terdaftar daftar barang yaitu daftar kacamata serta detail masing-masing kacamata serta link untuk perbaharui dan hapus kacamata.



Gambar 3.11 Halaman Kacamata - Admin

3.3.1.11 Halaman Transaksi – Admin

Gambar 3.12 berikut ini menampilkan rancangan antar muka untuk halaman transaksi pada admin, berisikan gambar admin sebagai *header*, *link-link* agar admin

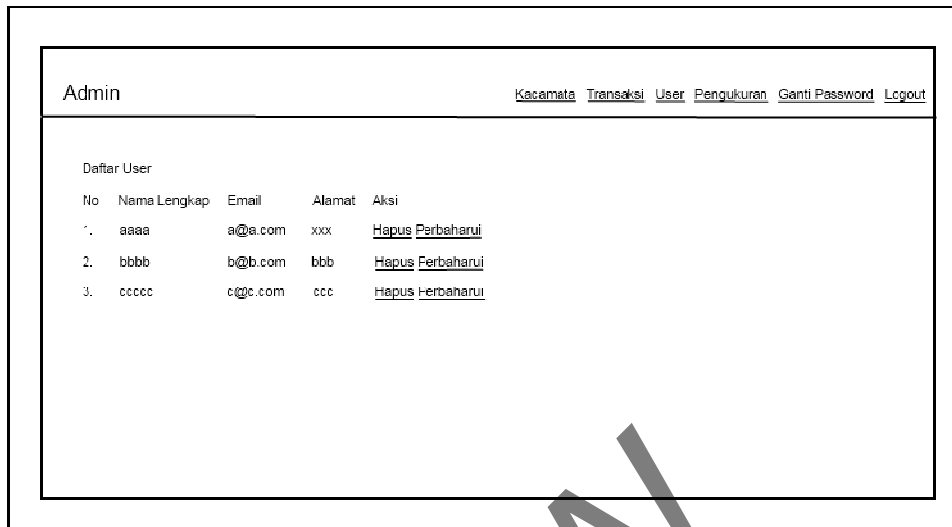
dapat mengakses setiap halaman untuk fitur pada admin yaitu *kacamata*, *transaksi*, *user*, *pengukuran*, *ganti password*, *password* , lalu pada tengah terdapat daftar transaksi, yaitu berisikan informasi kode transaksi, waktu, *user* yang melakukan, nama *user* dan link untuk hapus dan melihat detail setiap transaksi.

Admin					
					Kacamata Transaksi User Pengukuran Ganti Password Logout
Daftar Transaksi					
No	Kode Transaksi	Waktu	User Email	Nama Lengkap	Aksi
1.	xxxxx	xxxx	xx@x.com	xxx xxx	Hapus Detail
2.	yyyyy	yyyy	yy@y.com	yyy yyy	Hapus Detail
3.	zzzzz	zzzz	zz@z.com	zzz zzzz	Hapus Detail
4.	aaaaa	aaaa	aa@a.com	aaa aaaa	Hapus Detail

Gambar 3.12 Halaman Transaksi - Admin

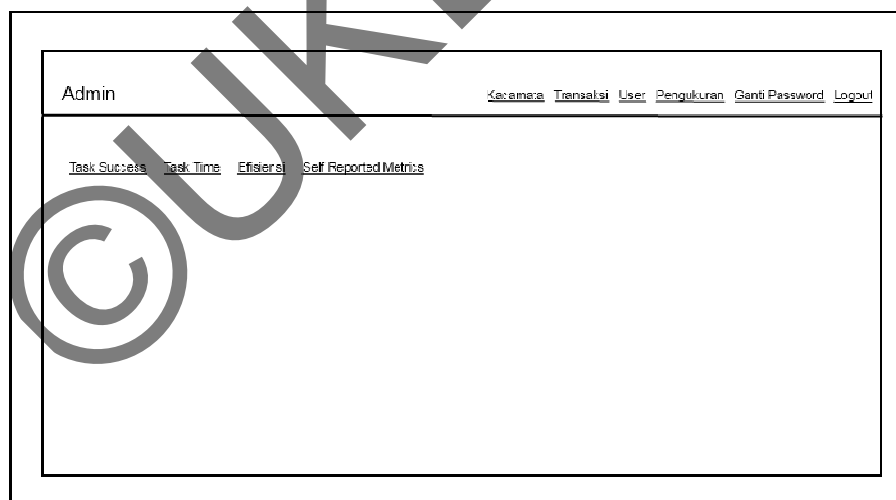
3.3.1.11 Halaman *User* – Admin

Gambar 3.13 berikut menampilkan rancangan antar muka untuk halaman *user* pada admin, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu *kacamata*, *transaksi*, *user*, *pengukuran*, *ganti password*, *password* dan pada tengah berisikan daftar *user* dan informasi masing-masing *user* yaitu nama lengkap, email dan alamat serta link untuk hapus serta perbaharui informasi *user*. Pada halaman ini, admin dapat melakukan pembaharuan terhadap informasi *user* dan dapat juga menghapus *user* yang terdaftar pada sistem. Selain itu juga tentu admin dapat melihat informasi *user* yang tersedia pada basis data sistem dan detailnya secara lengkap. *User* yang dimaksud dalam sistem ini adalah *user* yang telah mendaftar sebagai konsumen toko *kacamata*.



Gambar 3.13 Halaman *User*- Admin

3.3.1.12 Halaman Pengukuran – Admin

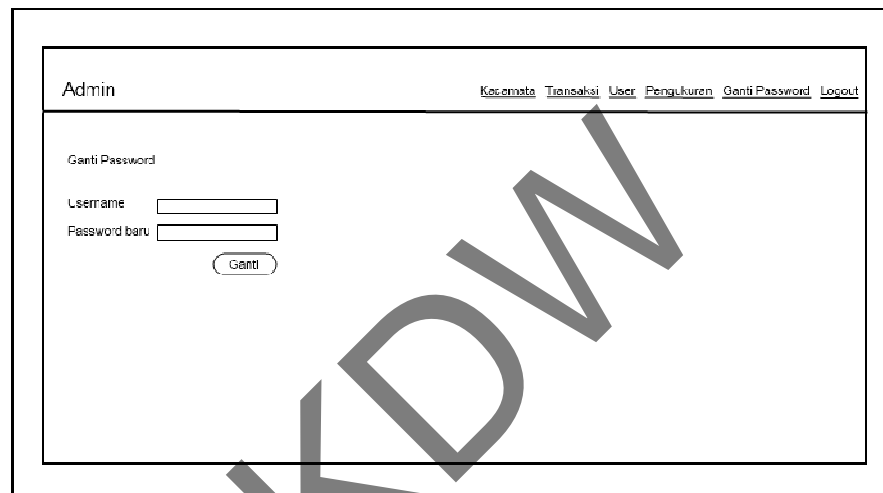


Gambar 3.14 Halaman Pengukuran – Admin

Gambar 3.13 menampilkan rancangan antarmuka untuk pengukuran *user experience*. Terdapat gambar admin sebagai *header*, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu kacamata, transaksi, *user*,

pengukuran, ganti *password*, *password* , lalu di tengah terdapat pilihan berupa link yang merujuk pada masing-masing metrik pengukuran *user experience* yaitu *Task Success* , *Task Time*, *Efficiency* dan *Self Reported Metrics*.

3.3.1.13 Halaman Ganti *Password* – Admin



The image shows a web interface for an admin user to change their password. At the top left, the word 'Admin' is displayed. To its right is a horizontal navigation menu with links for 'Kacamata', 'Transaksi', 'User', 'Pengukuran', 'Ganti Password', and 'Logout'. The main section of the page is titled 'Ganti Password'. It contains a form with two input fields: 'Username' and 'Password baru'. Below these fields is a button labeled 'Ganti'.

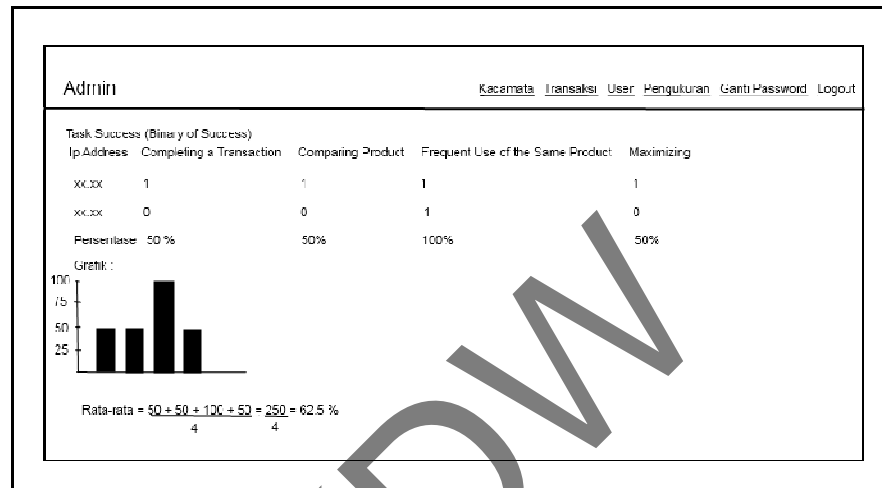
Gambar 3.15 Halaman Ganti *Password* - Admin

Gambar 3.15 menampilkan rancangan antar muka untuk *gantipassword* untuk admin, terdapat gambar sebagai *header* untuk admin di atas, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu *kacamata*, *transaksi*, *user*, *pengukuran*, *ganti password*, *password* dan di tengah terdapat form untuk mengisi *username* dan form untuk *password* baru, serta tombol *ganti* untuk mengirimkan informasi ke *server*.

3.3.1.14 Halaman Pengukuran – *Task Success*

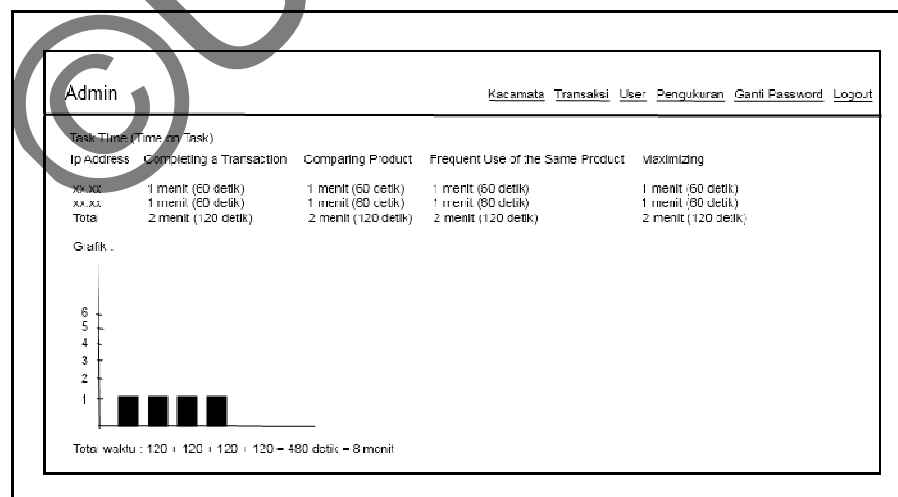
Gambar 3.16 menampilkan rancangan antar muka untuk halaman pengukuran *task success* pada admin, berisikan gambar admin sebagai *header*, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu *kacamata*, *transaksi*, *user*, *pengukuran*, *ganti password*, *password* dan di tengah

terdapat informasi berupa daftar *ip address*, dan nilai *task success* dalam 1 dan 0 untuk masing-masing tes skenario, serta ditampilkan dalam bentuk grafik dan rata-rata data.



Gambar 3.16 Halaman Pengukuran – *Task Success*

3.3.1.15 Halaman Pengukuran – *Task Time*

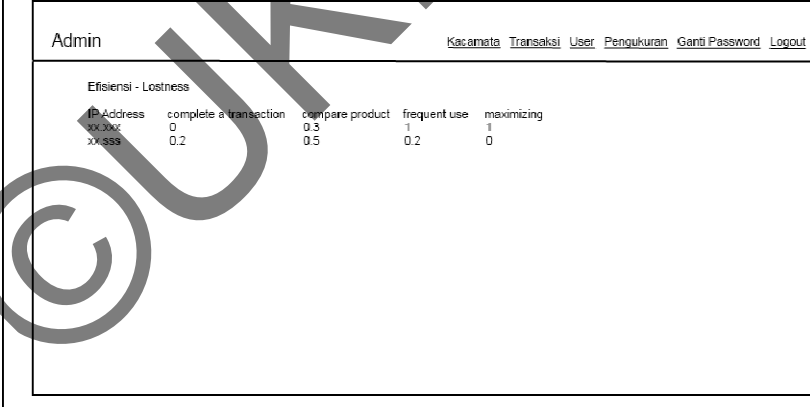


Gambar 3.17 Halaman Pengukuran – *Task Time*

Gambar 3.17 menampilkan rancangan antar muka untuk halaman pengukuran *task time* pada admin, berisikan gambar admin sebagai *header*, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu kacamata, transaksi, *user*, pengukuran, ganti *password*, *password* dan di tengah terdapat informasi berupa daftar *ip address*, dan nilai *task success* dalam 1 dan 0 untuk masing-masing tes skenario, serta ditampilkan dalam bentuk grafik dan rata-rata data.

3.3.1.16 Halaman Pengukuran – Efisiensi (*Lostness*)

Gambar 3.18 menampilkan rancangan antar muka untuk halaman pengukuran Efisiensi (*Lostness*) pada admin, berisikan gambar admin sebagai *header*, *link-link* agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu kacamata, transaksi, *user*, pengukuran, ganti *password*, *password* dan di tengah terdapat informasi berupa daftar *ip address*, dan nilai *lostness* dari setiap skenario tes.



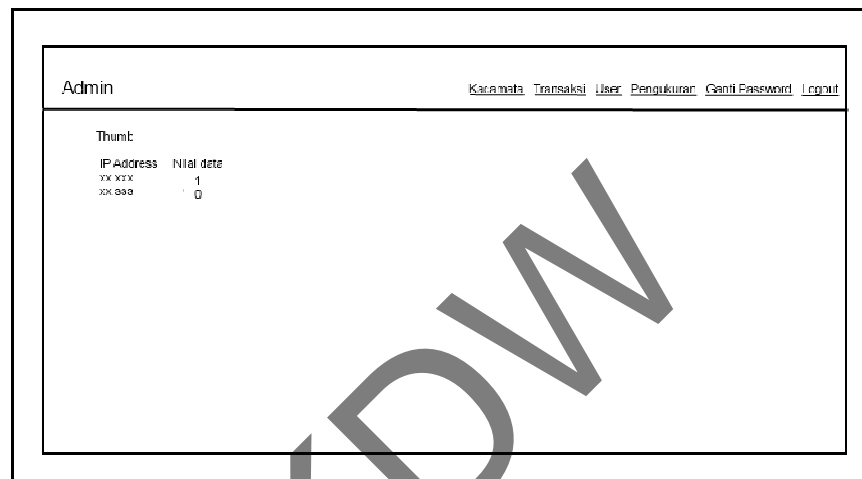
Admin		Kacamata Transaksi User Pengukuran Ganti Password Logout			
Efisiensi - Lostness					
IP Address	complete a transaction	compare product	frequent use	maximizing	
xx-xx-xx	0	0.3	1	1	
xx-xx-xx	0.2	0.5	0.2	0	

Gambar 3.18 Halaman Pengukuran – Efisiensi (*Lostness*)

3.3.1.17 Halaman Pengukuran – *Self Reported Metrics*

Gambar 3.19 berikut ini menampilkan rancangan antarmuka untuk halaman pengukuran *self reported metrics* pada admin, berisikan gambar admin sebagai

header, link-link agar admin dapat mengakses setiap halaman untuk fitur pada admin yaitu *kacamata, transaksi, user, pengukuran, ganti password, password* dan di tengah terdapat informasi berupa daftar *ip address*, dan nilai *self reported metrics* bernilai 1 jika baik dan 0 jika kurang baik.



The screenshot shows an Admin interface with a navigation menu at the top: [Kacamata](#), [Transaksi](#), [User](#), [Pengukuran](#), [Ganti Password](#), and [Logout](#). Below the menu is a table titled "Thunt:" with the following data:

IP Address	Nilai	date
XX.XXX.XXX.XXX	1	
XX.XXX.XXX.XXX	0	

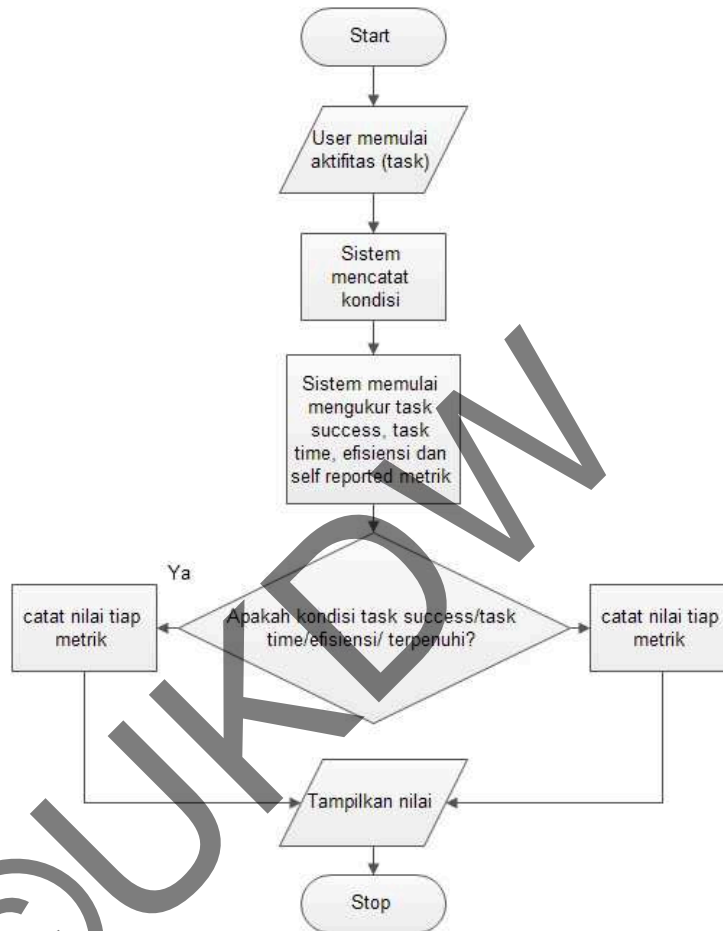
Gambar 3.19 Halaman Pengukuran – *Self Reported Metric*

3.4 Modul Pada Sistem

3.4.1 Modul Sistem Secara Umum

Modul sistem secara umum menunjukkan alur kerja sistem keseluruhan secara kasar untuk setiap modul yang ada pada sistem. Sistem mencatat kondisi awal dan memeriksa kondisi, mengukur data dan memeriksa kondisi akhir dari suatu tugas yang diberikan. Jika kondisi terpenuhi maka sistem akan mencatat nilai yang baru lalu menampilkannya, jika kondisi tidak terpenuhi maka sistem akan menampilkan nilai yang telah ditetapkan di awal untuk setiap masing-masing skenario tes. Penjelasan alur untuk skenario tes lainnya akan dijelaskan pada modul-modul *flowchart* terpisah. Kondisi awal dan akhir pada setiap skenario tes adalah berbeda antara satu dan lainnya. Sehingga modul ditampilkan dalam

flowchart secara kasaran saja tidak secara spesifik. Gambar 3.19 berikut menunjukkan *flowchart* keseluruhan sistem secara umum.



Gambar 3.20 Flowchart sistem secara umum

3.4.2 Modul Penghitungan *Time on Task*

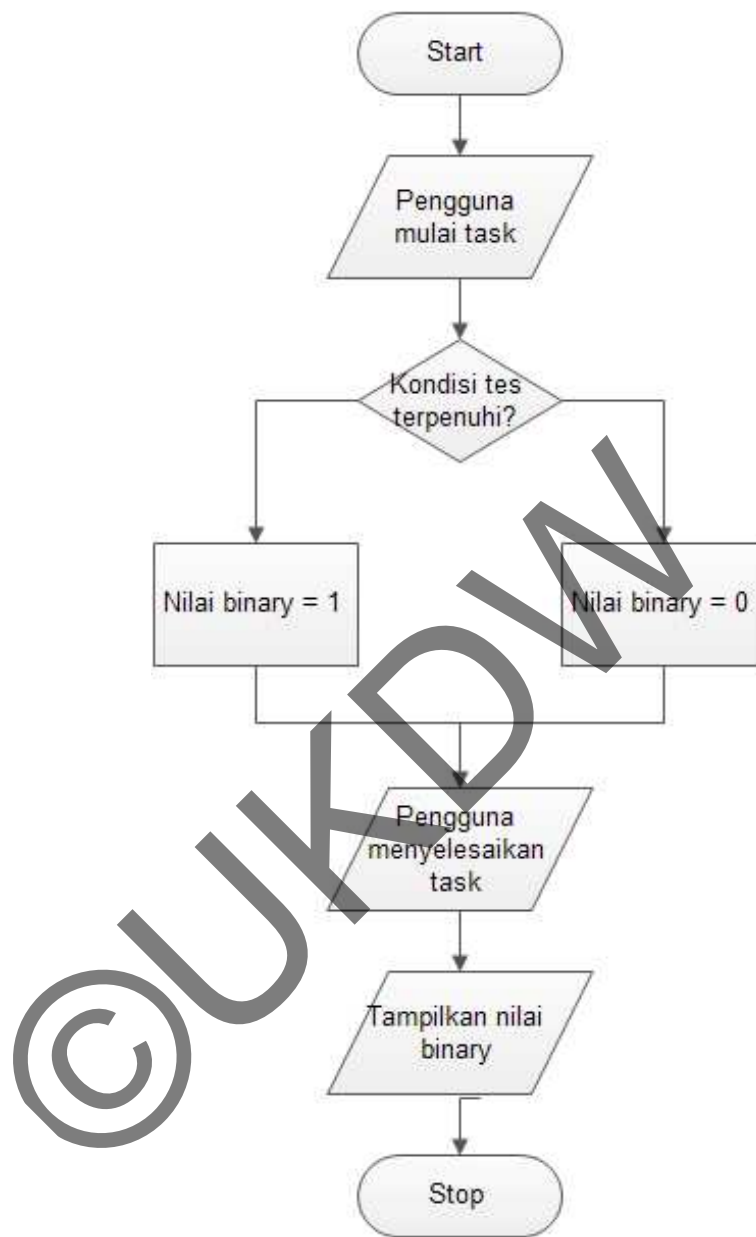
Modul penghitungan *Time on Task* adalah untuk menghitung lamanya waktu yang diperlukan untuk melakukan suatu tugas/skenario tes. Gambar 3.20 berikut ini menampilkan alur cara kerja sistem mengeksekusi modul penghitungan *time on task* secara kasaran. Kondisi awal dan akhir tiap skenario tes adalah berbeda. Jika kondisi tidak terpenuhi atau dengan kata lain gagal, maka nilai akan sama dengan 0



Gambar 3.21 Flowchart modul penghitungan *Time on Task*

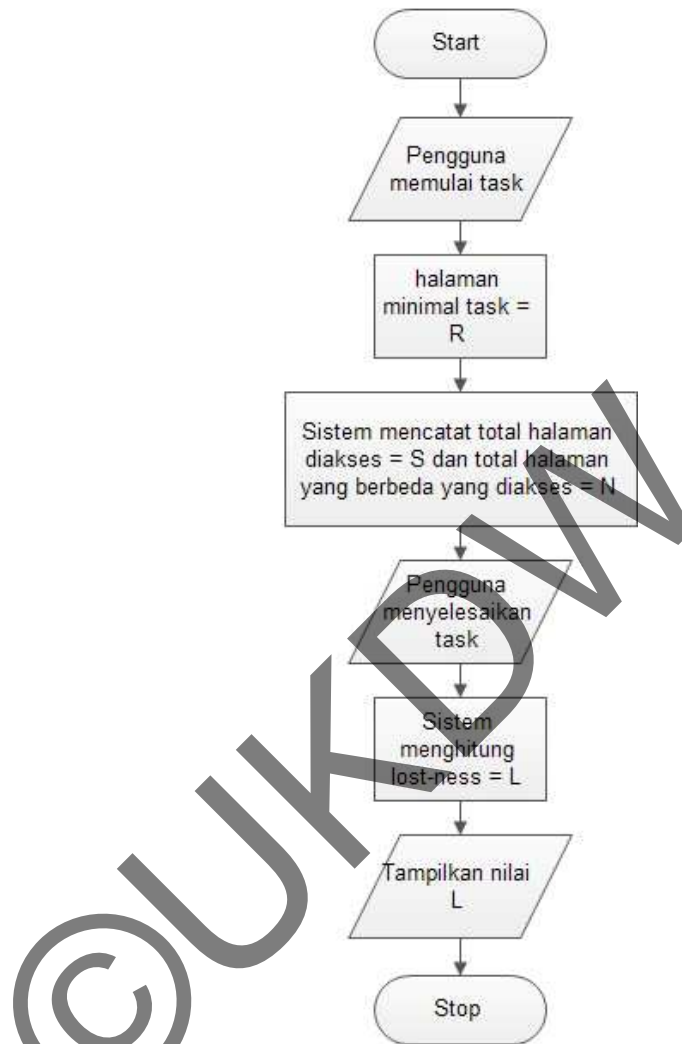
3.4.3 Modul Penghitungan *Task Success (Binary of Success)*

Modul penghitungan *task success (binary of success)* adalah untuk menilai sukses tidaknya suatu tugas/ skenario dilakukan, nilai sukses adalah 1 dan nilai gagal akan sama dengan 0. Gambar 3.21 menampilkan alur cara kerja untuk modul penghitungan *task success (binary of success)*. Nilai yang ditampilkan adalah 1 jika kondisi terpenuhi dan pengguna telah menyelesaikan skenario tes. Jika kondisi tidak terpenuhi maka sistem akan mencatat dan menampilkan nilai awal yaitu 0 sebagai informasi bahwa skenario tes gagal dilakukan.



Gambar 3.22 Flowchart Modul Penghitungan *Task Success* (*Binary of Success*)

3.4.4 Modul Penghitungan Efisiensi (*Lost-ness*)



Gambar 3.23 Flowchart Modul Penghitungan Efisiensi (*Lost-ness*)

Pada sistem ini, efisiensi diukur dengan menggunakan penghitungan *lostness*. Fungsinya adalah untuk mengukur seberapa banyak usaha yang terbuang untuk menyelesaikan suatu tugas dengan menggunakan rumus *lost-ness* (Smith, 1996). Semakin kecil nilai *lostness* maka semakin baik tandanya *user experience* pada sistem tersebut. Nilai terbaik adalah nilai yang terkecil (tidak ada usaha yang hilang) yaitu dinilai dengan 0 dan nilai terburuk adalah nilai yang terbesar (usaha

terlalu banyak memakan langkah) yaitu bernilai satu. Berikut adalah rumus *lostness* :

$$L = \text{sqrt} [(N/S - 1)^2 + (R/N - 1)^2]$$

Keterangan:

L : *lostness*

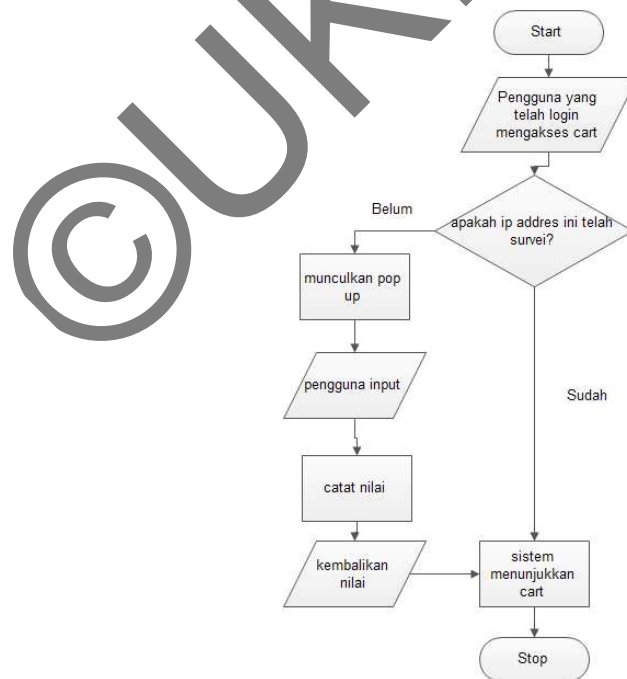
N : Jumlah halaman yang berbeda yang dikunjungi saat melakukan skenario tes

S : Jumlah halaman keseluruhan yang dikunjungi saat melakukan skenario tes

R : Jumlah halaman minimal untuk melakukan skenario tes

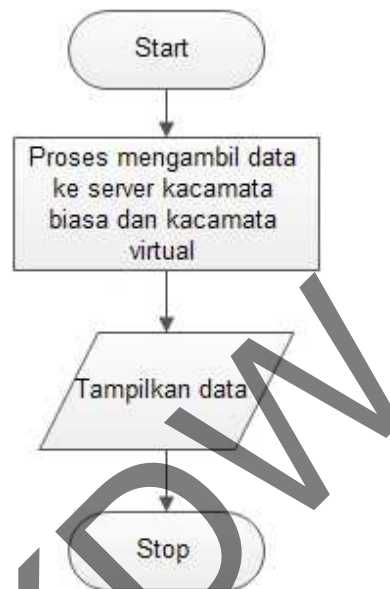
3.4.5 Modul Pencatatan *Self Reported Metrics*

Modul ini untuk mengukur nilai dari survei yang diarahkan kepada pengguna, input akan Cuma berbunyi iya dan tidak, dan nilai bernilai 1 dan 0.



Gambar 3.24 Flowchart Modul *Self Reported Metrics*

3.4.6 Modul Penampil Data dari 2 Toko Kacamata.



Gambar 3.25 Flowchart Modul Penampil Data.

Modul ini untuk mengambil nilai dari dua toko *online* kacamata lalu untuk ditampolkan secara sekaligus agar dapat dibandingkan.

3.5 Skenario Pengujian

Berikut adalah 5 dari 10 skenario yang paling sering dilakukan untuk melakukan pengujian *usability* menurut Tom tullis dan Bill Albert dalam buku mereka yang berjudul *Measuring the UserExperience :Collecting, Analyzing and Presenting Usability Metrics*(2008:49)

3.5.1 Menyelesaikan sebuah transaksi

Pada saat pengguna mengakses toko *online*, sistem akan mengukur kesuksesan dalam menyelesaikan suatu transaksi. Transaksi dalam toko *online* ini adalah kondisi awal mulai dari mengakses halaman awaldan lalu kondisi pengecekan berakhir adalah dengan

membeli kacamata yang tersedia di keranjang belanja. Metrik yang akan digunakan adalah *Task-Success* (dengan menggunakan *binary success*), *Time on Task*, Efisiensi.

3.5.2 Membandingkan kacamata

Sistem akan mengukur ketika pengguna membandingkan kacamata secara keseluruhan. Kondisi pengecekan dimulai ketika halaman awal diakses dan akan dianggap sukses ketika telah mengecek minimal 1 kacamata yang berbeda dan skenario berakhir ketika telah pengguna telah menambahkan satu kacamata ke keranjang belanja karena telah dianggap pengguna sukses menentukan pilihan. Metrik yang akan digunakan adalah *Task-Success* (dengan menggunakan *binary success*), *Time on Task*, Efisiensi.

3.5.3 Evaluasi membandingkan jenis kacamata yang sama

Sistem akan mengukur ketika pengguna membandingkan kacamata. Pengguna apakah balik untuk mengecek kacamata yang sama. Kondisi pengecekan dimulai ketika halaman awal diakses dan akan dianggap sukses dan berhasil ketika mengecek minimal 1 kacamata yang telah diakses sebelumnya. dan skenario berakhir ketika telah pengguna telah menambahkan satu kacamata ke keranjang belanja karena telah dianggap pengguna sukses menentukan pilihan. Metrik yang digunakan adalah *Task-Success* (dengan menggunakan *binary success*), *Time on Task*, Efisiensi.

3.5.4 Memaksimalkan usability dari produk yang penting (Registrasi)

Produk yang penting adalah produk yang keberadaannya bertujuan agar pengguna dapat menyelesaikan suatu tugas penting,

tugas yang jika tidak dikerjakan berpengaruh buruk terhadap hasilnya (Tom Tullis dan Bill Albert, 2008:53). Tugas yang sangat penting dalam penelitian ini adalah menyelesaikan sebuah transaksi, sehingga produk yang keberadaannya sangat penting dalam sistem adalah fitur registrasi. Kondisi pengecekan akan dimulai ketika pengguna mulai mengakses halaman registrasi, dan akan sukses jika pengguna telah sukses mengirim semua data registrasi. Metrik yang akan digunakan adalah : *Task-Success* (dengan menggunakan *binary success*), *Time on Task*, Efisiensi.

3.5.5 Menemukan total *user experience* yang bersifat positif

Sistem akan mengukur masukan dari pengguna, berupa fitur survei. Ketika pengguna menyelesaikan transaksi, sistem akan memunculkan *pop up* untuk pilihan bagus (Ya) dan kurang bagus (Tidak). Pilihan akan dimasukkan pengguna. Yang bagus akan bernilai 1 dan kurang bagus akan bernilai 0. Jumlah yang baik akan dijumlah pada akhirnya. Metrik yang akan digunakan adalah *self reported metrics*. Pertanyaan yang akan diajukan adalah untuk mengetahui apakah situs toko online memudahkan *user* menemukan kacamata yang diinginkan. Pertanyaannya adalah : “Apakah toko online ini memudahkan anda untuk menemukan kacamata yang anda inginkan?”.

BAB IV

IMPLEMENTASI DAN ANALISIS SISTEM

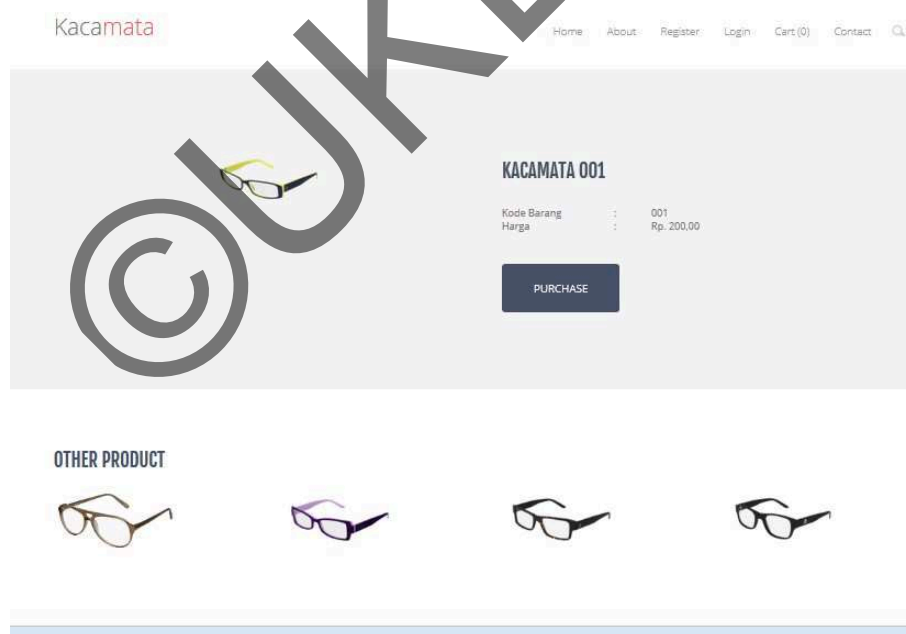
4.1 Implementasi Sistem

4.1.1 Implementasi Antarmuka

Sistem yang dibangun terdiri dari sistem toko *online* biasa, sistem toko *online* dengan *augmented reality* yang dalam bentuk PHP dan sistem pengukuran user experience yang dalam bentuk PHP, javascript dan *web service* yang terintegrasi dengan kedua toko *online*

Berikut adalah implementasi antarmuka yang penting untuk pengukuran *user experience*.

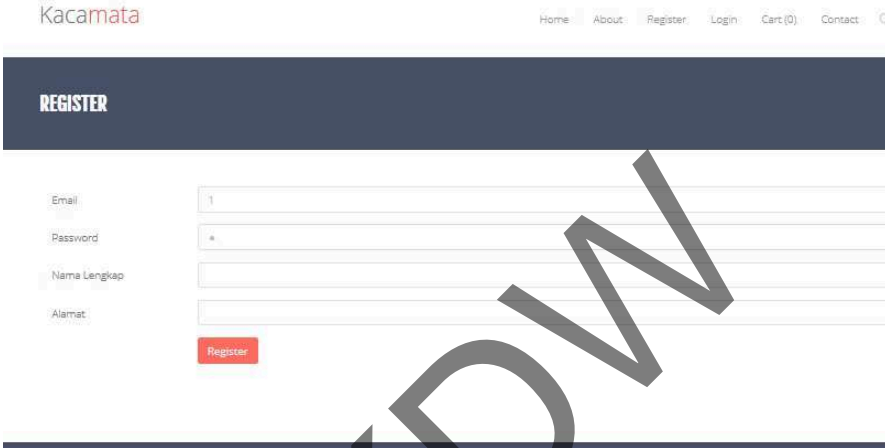
1) Halaman awal



Gambar 4.1 Halaman Awal Toko Kacamata

Gambar 4.1 menampilkan halaman awal dari situs kacamata, menjadi awal dan parameter pertama dari setiap *task* yang akan dikerjakan dan diukur. Juga merupakan halaman kacamata pertama.

2) Halaman Register

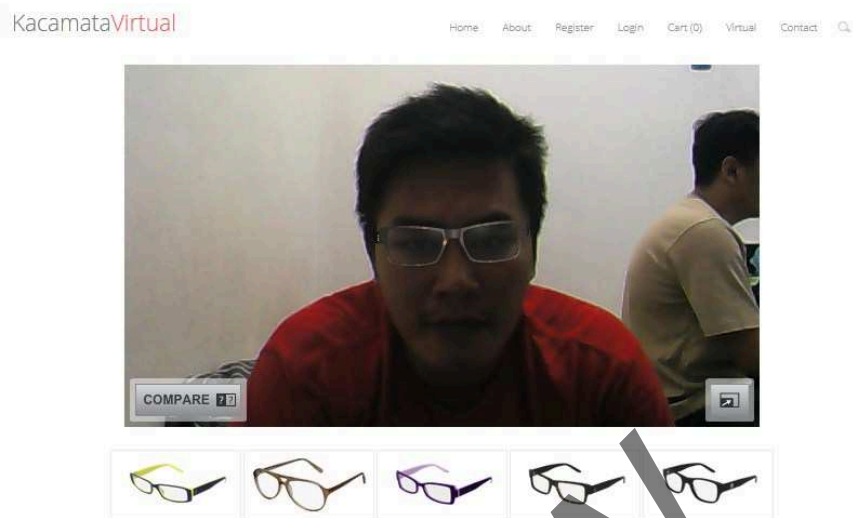


Gambar 4.2 Halaman Register

Gambar 4.2 menampilkan halaman untuk registrasi, halaman ini adalah tempat parameter mengukur kesuksesan dari skenario tes untuk registrasi.

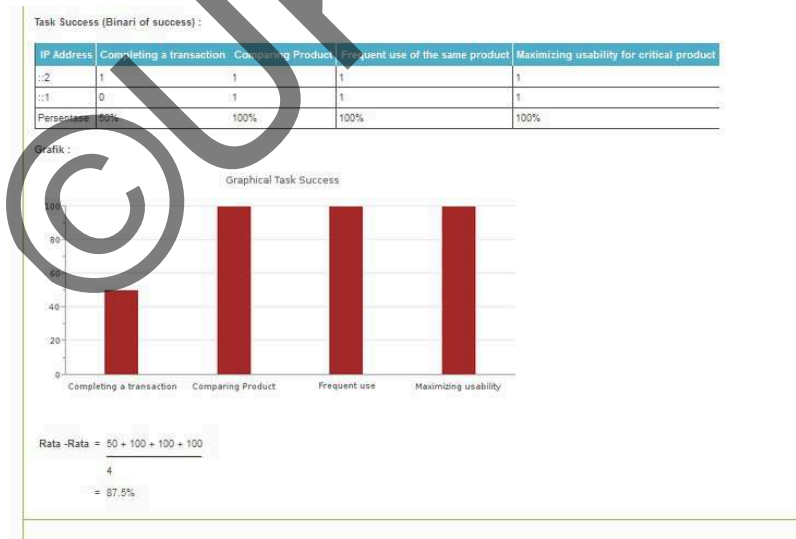
3) Halaman kacamata dengan *augmented reality* (cermin virtual)

Gambar 4.3 berikut ini akan menampilkan kacamata dengan fitur cermin virtual yang menggunakan *augmented reality*. Halaman ini berpengaruh terhadap *task time* pada sistem kacamata *online* berbasis *augmented reality* pada skenario tes membandingkan kacamata.



Gambar 4.3 Halaman Kacamata dengan *Augmented Reality*

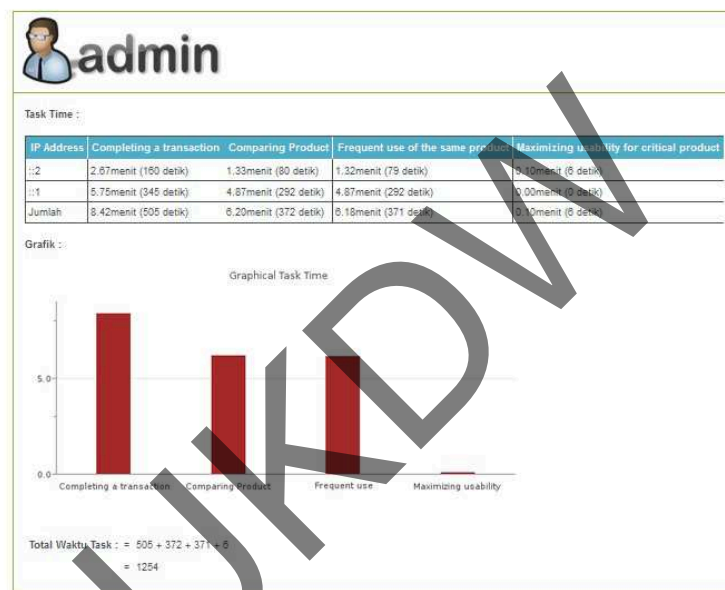
4) Halaman Pengukuran *user experience – task success*



Gambar 4.4 Halaman Pengukuran *User Experience – Task Success*.

Gambar 4.4 menampilkan data hasil pengukuran *user experience* yaitu hasil kesuksesan skenario tes, nilai kesuksesan direpresntasikan dalam bentuk tabel dan grafik serta rata-rata dari nilai data yang ada.

5) Halaman Pengukuran *User Experience – Time on Task*



Gambar 4.5 Pengukuran *User Experience – Time on Task*

Gambar 4.5 menampilkan data hasil pengukuran *user experience* dengan metrik *time on task*, data direpresentasikan dalam bentuk tabel dan grafik.

6) Halaman Pengukuran *User Experience – Efisiensi*

Gambar 4.6 berikut ini menampilkan data hasil pengukuran *user experience* dengan menggunakan metrik efisiensi yaitu hasil dari perhitungan rumus *lost-ness*. Ditampilkan dalam bentuk tabel.

Lost Ness :

IP Address	Completing a transaction	Comparing Product	Frequent use of the same product	Maximizing usability for critical product
...2	0,80	0,71	0,71	0,88
...1	0	0,80	0,80	0

Gambar 4.6 Halaman Pengukuran *User Experience* –Efisiensi (*Lost-ness*)

7) Halaman Pengukuran *User Experience* – *Self Reported Metrics*

admin [Kacamata](#) [Transaksi](#) [User](#) [Pengguna](#)

Thumb

IP Address	Value
103.247.42.195	0
120.175.238.93	1
118.99.89.119	1
114.121.152.195	1

Gambar 4.7 Halaman Pengukuran *User Experience* – *Self Reported Metrics*

Gambar 4.7 menampilkan data alamat ip dan nilai dari survey yang dimasukkan oleh pengguna, berupa nilai 1 dan 0 untuk nilai jika memudahkan pengguna untuk mencari kacamata yang dia inginkan.

4.2.2 Implementasi pemrograman

1) *Task Success – Binary of Success*

a) Fungsi checkout.

```
public function checkout($ip){
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','checkout');
    $query = $this->db->get('aktivitas');
    $task_value = 0;
    if($query->num_rows() > 0){
        $task_value = 1;
    }
    return $task_value;
}
```

Pada fungsi *checkout* ini, sistem mengecek nilai kesuksesan tes skenario setiap *ip address* (nanti akan memanggil variabel *\$ip* dari tabel *aktivitas_ip_address*). Sistem lalu mencari pada *database*, data yang aktif (*aktivitas_status=1*) kata *checkout* yang dimiliki oleh setiap URL (dari tabel *aktivitas_uri* , karena ketika *user* sukses melakukan transaksi, akan terjadi pencatatan *url checkout*).

Setelah melakukan *query*, sistem memeriksa *num_rows*, jika lebih dari 0 maka ada minimal 1 kata *checkout* ditemukan, berarti *user* telah sukses melakukan *checkout* minimal sebanyak 1 kali. Nilai awal tes skenario

(\$task_value) jika tidak ditemukan bernilai 0, dan jika ditemukan nilai akan ditentukan menjadi 1. Terakhir, nilai akan dikembalikan.

b) Fungsi compare_product

```
public function compare_product($ip){
    $this->db->distinct();
    $this->db->select('aktivitas_uri');
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','index');
    $query = $this->db->get('aktivitas');
    $task_value = 0;
    if($query->num_rows() > 1){
        $task_value = 1;
    }
    return $task_value;
}
```

Pada fungsi compare_product ini, sistem mengecek nilai kesuksesan tes skenario setiap *ip address* (\$ip dari tabel aktivitas_ip_address). Sistem akan memulai dengan mengelompokkan (distinct) dari tabel aktivitas uri (distinct mengelompokkan dari apa yang muncul dengan perintah select). Lalu sistem akan mengecek data yang aktif (aktivitas status=1). Sistem lalu mencari kata dalam aktivitas_uri, yang mengandung kata index, karena ketika mengakses halaman kaca mata setiap kaca mata berada dibawah *link* index, dari kaca mata 1 hingga kaca mata 5, sehingga memudahkan pencatatan

halaman. Nilai dari *task succes* (task_value) awal adalah 0. Sistem akan melakukan *query* dari tabel aktivitas, dan melakukan pengecekan, jika data yang mengandung kata index (num_rows > 1) maka nilai akan diubah menjadi 1. Terakhir sistem akan mengembalikan nilai.

c) Fungsi frequent_use

```
public function frequent_use($ip){
    $query = "SELECT aktivitas_uri, count(*) as frequent
            from aktivitas
            where aktivitas_status=1
            and aktivitas_ip_address='$ip'
            and aktivitas_uri like '%index%'
            group by aktivitas_uri";
    $execute = $this->db->query($query);
    $task_value = 0;
    foreach($execute->result() as $r){
        $frequent_use = $r->frequent;
        if($frequent_use >= 2){
            $task_value = 1;
            return $task_value;
            exit();
        }
    }
    return $task_value;
}
```

Pada fungsi `frequent_use` ini, query ditentukan sebagai perintah atas tabel `aktivitas_uri` dan melakukan fungsi `count` sebagai "frequent", nilainya akan dikembalikan dalam nama field "frequent". Sistem mengecek data aktif, alamat ip dan kata "%index%" (url yang sama persis), dikelompokkan berdasarkan `aktivitas_uri`.

Lalu query dijalankan dan nilai task didefinisikan = 0. Sistem lalu melakukan perulangan untuk menjalankan query, dan mengecek hasilnya. jika menemukan kata "%index%" pada query, maka "frequent" akan ditambahkan dan jika nilainya telah mencapai 2 atau lebih, nilai task yang tadinya = 0 akan diubah menjadi 1, lalu nilai akan dikembalikan.

d) Fungsi `maximizing_usability`

```
public function maximizing_usability($ip){
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','register');
    $query = $this->db->get('aktivitas');
    $task_value = 0;
    if($query->num_rows() > 0){
        $task_value = 1;
    }
    return $task_value;
}
```

Pada fungsi `maximizing_usability` ini, cara pengecekannya sama seperti pada fungsi `checkout`, tetapi kata yang akan dicari oleh sistem adalah kata `register`.

2. Task Time – (Time on Task)

a) Fungsi checkout_time)

```
public function checkout_time($ip){
    //get list aktivitas
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->order_by('id_aktivitas','asc'); //yang pertama
    $query_a = $this->db->get('aktivitas');
    $first_time = $query_a->row()->aktivitas_time;
    //pre($query_a->result());

    //get checkout time
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','checkout');
    $this->db->order_by('id_aktivitas','asc');
    $query_b = $this->db->get('aktivitas');
    //pre($query_b->result());
    $task_value = 0;
    if($query_b->num_rows() > 0){
        $success_time = $query_b->row()->aktivitas_time;
        //pre($success_time);
        $task_value =
get_date_diff($success_time,$first_time);
    }
    //pre($task_value);
}
```

```
        return $task_value;
    }
```

Pada fungsi ini , sistem akan memeriksa data yang aktif dan ip address lalu mengurutkan id_aktivitas dari yang terkecil. lalu mendefinisikan isi query_a yaitu mengambil dari tabel aktivitas, dan mendefinisikan first_time sebagai waktu pertama dari aktivitas_time. Sistem akan mengecek lagi data yang sama (aktif dan ip address) lalu mengecek kata "checkout", lalu jika ketemu, sistem lalu akan mengurutkannya dari yang index id_aktivitas terkecil, lalu mendefinisikan query_b dari tabel aktivitas.

Sistem mendefinisikan nilai task awal = 0, lalu mengecek jika num_rows pada query_b lebih besar dari 0, jika ya, maka sistem akan mendefinisikan success_time yaitu nilai dari aktivitas_time pada query_b, lalu sistem mendefinisikan nilai task dari selisih waktu antara success_time dan first_time. Nilai task akan dikembalikan

b) Fungsi compare_product_time

```
public function compare_product_time($ip){
    //get time product yang diakses pertama kali
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->order_by('id_aktivitas','asc'); //kecil ke besar
    $this->db->like('aktivitas_uri','index'); //index, letak product
    $query_a = $this->db->get('aktivitas');
    $first_time = $query_a->row()->aktivitas_time;
    //pre($query_a->result());
}
```

```

        //get add to chart time
        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db->like('aktivitas_uri','add');
        $this->db->order_by('id_aktivitas','asc');
        $query_b = $this->db->get('aktivitas');
        //pre($query_b->result());

        $task_value = 0;
        if($query_b->num_rows() > 0){
            $success_time = $query_b->row()->aktivitas_time;
            $task_value =
            get_date_diff($success_time,$first_time);
        }
        //pre($task_value);
        return $task_value;
    }

```

Pada fungsi ini, cara kerjanya sama seperti fungsi checkout_time. Fungsi ini berbeda saat melakukan pengecekan dengan fungsi checkout_time, checkout_time mengecek dengan mencari kata “checkout”, tetapi fungsi ini memulai mendefinisikan query_b setelah menemukan kata “add” (yaitu ketika pengguna melakukan “add to cart”).

c) Fungsi frequent_use_time

```
public function frequent_use_time($ip){
    //get time product yang diakses pertama kali
    $query = "SELECT id_aktivitas, aktivitas_uri, aktivitas_time,
count(*) as jumlah from aktivitas
    where aktivitas_status = 1
    and aktivitas_ip_address = '$ip'
    and aktivitas_uri not in ('welcome/index')
    and aktivitas_uri like '%index%'
    group by aktivitas_uri
    order by id_aktivitas asc";
    $x = $this->db->query($query);
    //pre($x->result());
    //exit();

    //cek ketersediaan data
    $task_value = 0;
    if($x->num_rows() > 0){
        $y = 0;
        //cek frequent disini
        foreach($x->result() as $r){
            if($r->jumlah > 1 ){
                $y = $r->jumlah;
                $first_time = $r->aktivitas_time;
                break;
            }
        }
        if($y != 0){
```

```

//get add to chart time
$this->db->where('aktivitas_status',1);
$this->db->where('aktivitas_ip_address',$ip);
$this->db->like('aktivitas_uri','add');
$this->db->order_by('id_aktivitas','asc');
$query_b = $this->db->get('aktivitas');

$task_value = 0;
if($query_b->num_rows() > 0){
    $success_time = $query_b->row()-
>aktivitas_time;
    $task_value =
get_date_diff($success_time,$first_time);
}
}
return $task_value;
}

```

pada fungsi ini , sistem akan memulai query untuk mengambil waktu produk yang diakses pertama kali.

query didefinisikan melalui Id_aktivitas, aktivitas_uri, aktivitas_time dan akan di hitung sebagai jumlah dari tabel aktivitas.

Sistem mengecek data aktif, ip address, dan yang tidak terdapat di direktori 'welcome/index' dan mengecek url 'index' yang sama persis dan akan dikelompokkan dengan aktivitas_uri, sedangkan diurutkan dari kecil berdasarkan id_aktivitas. x didefinisikan dari query, lalu nilai task didefinisikan = 0.

Sistem akan mengecek apabila ada num_rows yang ditemukan (url yg sama persis yg memiliki kata "index"), jika ada yang ditemukan, maka y akan di set = 0, lalu sistem akan melakukan kondisi perulangan untuk mendefinisikan first_time yang akan di ambil dari aktivitas_time setelah dicek dengan kondisi dalam perulangan.

Sistem akan mengecek lagi data aktif, ip address, dan kondisi 'add to cart' lalu mengurutkannya dari kecil berdasarkan id_aktivitas, lalu mendefinisikan query_b dari tabel aktivitas.

sistem mendefinisikan nilai task = 0 lalu dilakukan kondisi pengecekan melalui jumlah query_b yang lebih besar dari num_rows jika ketemu, maka success_time didefinisikan. dan akhirnya nilai task didefinisikan dari selisih success_time dan first_time. lalu nilai akan dikembalikan.

d) Fungsi maximizing_usability_time

```
public function maximizing_usability_time($ip){  
    //get register time  
    $this->db->where('aktivitas_status',1);  
    $this->db->where('aktivitas_ip_address',$ip);  
    $this->db->where('aktivitas_uri','welcome/register');
```



```

$this->db->order_by('aktivitas_time','desc');
$a = $this->db->get('aktivitas',1,1); //limit 1 offset 1

$task_time = 0;
//jika data ada
if($a->num_rows() == 1){
    $baris = $a->row();
    $time_register = $baris->aktivitas_time;
    $session = $baris->aktivitas_session;
    //pre($this->db->last_query());

    //get register success time
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_session',$session);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db-
>where('aktivitas_uri','welcome/register_success');
    $this->db->order_by('aktivitas_time','desc');
    $b = $this->db->get('aktivitas',1,0); //limit 1 offset 0
    //pre($this->db->last_query());

    if($b->num_rows() > 0){
        $data = $b->row();
        $time_success_register = $data->aktivitas_time;

        $task_time =
get_date_diff($time_success_register,$time_register);

```

```

        //pre($time_register);
        //pre($time_success_register);
    }
}

//pre($task_time);
return $task_time;
}

```

Pada fungsi ini, sistem mengukur waktu registrasi. Dimulai dari mengecek data yang aktif, ip address, lalu mengecek url yang terdapat kata "welcome/register", diurutkan dari paling besar berdasarkan aktivitas_time, nilai awal task = 0.

Cara pengecekannya jika ada data ($\text{num_rows} == 1$) maka sistem akan melakukan penghitungan. waktu awal adalah time_register yang didapatkan dari pengecekan data awal dan melakukan penetapan sesi. lalu sistem akan melakukan lagi pengecekan data aktif, sesi yang sama, ip address, yang memiliki kata "welcome/register", diurutkan dari paling besar berdasarkan aktivitas_time, time_success_register didapat kan dari pengecekan kedua dan diambil dari aktivitas_time, nilai task yang akan dikembalikan jika kondisi terpenuhi adalah selisih antara time_success_register dan time_register.

3. Efisiensi (*Lost-ness*)

a) Fungsi `lostness_complete_transaction`

```

public function lostness_complete_transaction($ip){
    //get checkout
    $this->db->where('aktivitas_status',1);
}

```

```

$this->db->where('aktivitas_ip_address',$ip);
$this->db->where('aktivitas_uri','welcome/checkout');
$this->db->order_by('id_aktivitas','asc');
$query_a = $this->db->get('aktivitas');
//cek jika ada checkout
$hasil = 0;
if($query_a->num_rows() != 0){
    //CARI N
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db-
>where_not_in('aktivitas_uri',array('welcome/index'));
    $this->db->where('id_aktivitas <',$query_a->row()-
>id_aktivitas);
    $this->db->group_by('aktivitas_uri');
    $query_n = $this->db->get('aktivitas');
    $n = $query_n->num_rows();
    //pre($n);
    //pre($this->db->last_query());
    //pre($query_n->result());

    //CARI S
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db-
>where_not_in('aktivitas_uri',array('welcome/index'));
    $this->db->where('id_aktivitas <',$query_a->row()-
>id_aktivitas);

```

```

$query_s = $this->db->get('aktivitas');
$s = $query_s->num_rows();


```

Pada fungsi ini sistem memeriksa lost-ness dari checkout, sistem akan memeriksa kondisi jika checkout terjadi yaitu berdasar data aktif, ip address, kata "welcome checkout" lalu mengurutkannya, serta mendefinisikan query_a dari tabel aktivitas. Nilai task (hasil) didefinisikan = 0,

lalu kondisi pengecekan dilakukan, jika terdapat chekcout, maka sistem akan mencari nilai n (halaman yang berbeda), lalu mencari s (total alamat) berdasarkan num_rows dari aktivitas, dan r (halaman minimum) didefinisikan pengembang = 4, karena untuk menyelesaikan transaksi ada 4 halaman yang minimal di akses, halaman home (lalu klik add to cart), halaman register, halaman login, lalu cart (dan klik beli). lalu fungsi akan memanggil fungsi rumus, dan mengembalikan nilai.

b) Fungsi lostness_compare_product

```
public function lostness_compare_product($ip){
    //get add to chart
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','add');
    $this->db->order_by('id_aktivitas','asc');
    $query_a = $this->db->get('aktivitas');
    //pre($query_a->result());
    //cek jika ada checkout
    $hasil = 0;
    if($query_a->num_rows() != 0){
        //CARI N
        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db->
>where_not_in('aktivitas_uri',array('welcome/index'));
        $this->db->where('id_aktivitas <',$query_a->row()-
>id_aktivitas);
        $this->db->group_by('aktivitas_uri');
        $query_n = $this->db->get('aktivitas');
        $n = $query_n->num_rows();
        //pre($n);
        //pre($this->db->last_query());
        //pre($query_n->result());

        //CARI S
        $this->db->where('aktivitas_status',1);
```

```

        $this->db->where('aktivitas_ip_address',$ip);
        $this->db-
>where_not_in('aktivitas_uri',array('welcome/index'));
        $this->db->where('id_aktivitas <',$query_a->row()-
>id_aktivitas);

        $query_s = $this->db->get('aktivitas');
        $s = $query_s->num_rows();
        //pre($s);

        //SET R
        $r = 3;

        //Panggil fungsi
        $hasil = $this->rumus($n,$s,$r);
    }
    //pre($hasil);
    return $hasil;
}

```

Pada fungsi ini , sistem melakukan pengukuran lostness terhadap tes skenario membandingkan produk. Dimulai pengecekan dilakukan seperti fungsi lainnya : keaktifan data, ip address, lalu mengecek kata "add" untuk kondisi berhenti. lalu mendefinisikan nilai task = 0.

Kondisi pengecekan dimulai ketika num_row dari query_a tidak kosong, lalu fungsi memulai memeriksa keaktifan data, ip address, halaman yang tidak berada pada array yang memiliki kata "welcome/index" dan

dikelompokkan berdasarkan aktivitas_uri, lalu fungsi mendefinisikan query_n. dan mencari nilai n, s dan r serta memanggil fungsi rumus,, hasilnya nanti akan dikembalikan.

c) Fungsi lostness_frequent_use

```
public function lostness_frequent_use($ip){  
    //get add to chart  
    $this->db->where('aktivitas_status',1);  
    $this->db->where('aktivitas_ip_address',$ip);  
    $this->db->like('aktivitas_uri','add');  
    $this->db->order_by('id_aktivitas','asc');  
    $query_a = $this->db->get('aktivitas');  
    //pre($query_a->result());  
  
    //cek jika ada checkout  
    $hasil = 0;  
    if($query_a->num_rows() != 0){  
        //cek frequent use nya  
        $sid = $query_a->row()->id_aktivitas;  
        $query = "SELECT id_aktivitas, aktivitas_uri, count(*)  
as jumlah from aktivitas  
        where aktivitas_status = 1  
        and id_aktivitas < '$sid'  
        and aktivitas_ip_address = '$ip'  
        and aktivitas_uri not in ('welcome/index')  
        and aktivitas_uri like '%index%'
```

```

group by aktivitas_uri
order by id_aktivitas asc";
$query_f_use = $this->db->query($query);


```



```

        //pre($this->db->last_query());
        //pre($query_n->result());

        //CARI S
        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db-
>where_not_in('aktivitas_uri',array('welcome/index'));
        $this->db->where('id_aktivitas <','$query_a-
>row()->id_aktivitas);

        $query_s = $this->db->get('aktivitas');
        $s = $query_s->num_rows();
        //pre($s);

        //SET R
        $r = 3;

        //Panggil fungsi
        $hasil = $this->rumus($n,$s,$r);
    }
    //echo $y;
    //exit();
}

    //pre($hasil);
    return $hasil;
}

```

Pada fungsi ini , sistem melakukan pengukuran lostness terhadap tes skenario membandingkan produk. Dimulai pengecekan dilakukan seperti fungsi lainnya : keaktifan data, ip address, lalu mengecek kata "add" untuk kondisi berhenti. lalu mendefinisikan nilai task = 0.

Lalu sistem akan mengecek jika ada yg menambahkan to cart, berdasarkan num_rows (tidak sama dengan 0), lalu mengecek jika ada aktivitas penggunaan produk yang sama dan terlebih dulu memeriksa dari keaktifan data, ip dan lainnya. Sistem memeriksa dengan menggunakan kata yg sama persis yg mengandung "index", dan akan ditampung dalam "jumlah", jika "jumlah" lebih dari 1, maka pemeriksaan akan dilakukan, dan hasilnya akan ditampung dalam variabel y,

Jika ada produk yang diakses lebih dari 1 kali (variabel y tidak sama dengan 0), maka sistem akan mencari nilai n, s dan r ditentukan = 3, lalu sistem akan memanggil fungsi rumus dan mengembalikan hasilnya.

d) Fungsi `lostness_maximizing_usability`

```
public function lostness_maximizing_usability($ip){
    //get success uri
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->where('aktivitas_uri','welcome/register_success');
    $this->db->order_by('id_aktivitas','asc');
    $query_a = $this->db->get('aktivitas');
    //pre($query_a->result());

    //cek data
    $hasil = 0;
```

```

        if($query_a->num_rows() != 0){
            //CARI N
            $this->db->where('aktivitas_status',1);
            $this->db->where('aktivitas_ip_address',$ip);
            $this->db-
>where_not_in('aktivitas_uri',array('welcome/index'));
            $this->db->where('id_aktivitas <',$query_a->row()-
>id_aktivitas);

            $this->db->group_by('aktivitas_uri');
            $query_n = $this->db->get('aktivitas');
            $n = $query_n->num_rows();
            //pre($n);
            //pre($this->db->last_query());
            //pre($query_n->result());

            //CARI S
            $this->db->where('aktivitas_status',1);
            $this->db->where('aktivitas_ip_address',$ip);
            $this->db-
>where_not_in('aktivitas_uri',array('welcome/index'));
            $this->db->where('id_aktivitas <',$query_a->row()-
>id_aktivitas);

            $query_s = $this->db->get('aktivitas');
            $s = $query_s->num_rows();
            //pre($s);

            //SET R
            $r = 3;

```

```
        //Panggil fungsi
        $hasil = $this->rumus($n,$s,$r);

    }
    //pre($hasil);
    return $hasil;
}
}
```

Pada fungsi ini, pengukuran dilakukan dengan cara memeriksa keaktifan data, alamat ip dan kondisi yang sukses registrasi, lalu akan diurutkan dari kecil ke besar berdasarkan id_aktivitas.

nilai task akan didefinisikan menjadi 1. lalu sistem memeriksa yang sukses registrasi, setelah itu sistem akan mencari nilai n, s dan r dan memanggil fungsi rumus. Terakhir sistem akan mengembalikan nilai.

4.2 Analisis Sistem

Analisa sistem bertujuan untuk mengetahui nilai dari hasil pengukuran dari sistem terhadap kedua situs toko *online* agar dapat diukur perbedaannya. Pengujian dilakukan terhadap total 5 skenario tes dengan menggunakan 4 metrik pengukuran *user experience* terhadap 20 pengguna. Penyajian tabel data adalah berdasarkan metrik-metrik pengukuran *user experience* dengan berisikan perbandingan hasil setiap skenario tes antara situs kaca mata biasa dan situs kaca mata dengan cermin *virtual* pada setiap tabel. Hasil dan analisisnya dapat dilihat pada tabel-tabel berikut, dan perincian data setiap tabel dapat dilihat di lampiran.

Tabel 4.1 Pengukuran dengan metrik *Task Success – Binary of Success*

Skenario Tes	<i>Task Success – Binary of Success</i>			
	Situs Kacamata Biasa		Situs Kacamata Virtual	
	Pengguna Sukses (total 20)	Persentase	Pengguna Sukses (total 20)	Persentase
Menyelesaikan transaksi	3	15%	8	40%
Membandingkan produk	20	100%	12	60%
Evaluasi produk yang sama	14	70%	11	55%
Registrasi	6	30%	10	45%

Pada tabel 4.1. terlihat bahwa penurunan kesuksesan saat membandingkan produk (40%) dan evaluasi produk yang sama (15%), berarti terjadi permasalahan, menurut pengakuan 6 dari 8 pengguna yang gagal membandingkan produk, hal ini dikarenakan kecepatan internet yang lamban, sehingga membuat mereka gagal untuk mengakses halaman virtual Tetapi peningkatan terjadi pada bagian *checkout* (25%) dan *registrasi* (15%).

Walau pada situs kacamata *online* berbasis *Augmented Reality*. pengguna yang sukses mengakses menjadi lebih sedikit dibandingkan toko kacamata biasa, tetapi mengalami peningkatan dalam transaksi. Peningkatan terjadi ini menunjukkan bahwa *augmented reality* dapat digunakan untuk meningkatkan penjualan.

Tabel 4.2 Pengukuran dengan metrik *Time on Task*

Skenario Tes	<i>Task Time – Time on Task (dalam detik)</i>	
	Situs Kacamata Biasa (total dalam detik)	Situs Kacamata Virtual (total dalam detik)
Menyelesaikan transaksi	540	4760
Membandingkan produk	1000	3197
Evaluasi produk yang sama	994	2770
Registrasi	105	651

Pada tabel 4.2 terlihat bahwa semua skenario tes yang dilakukan pengguna mengalami peningkatan waktu pada bagian toko kacamata dengan fitur cermin virtual, menurut 5 pengguna toko kacamata (dipilih acak) dengan fitur cermin virtual yang ditanyakan, hal ini dikarenakan fitur cermin virtual dirasa cukup mengasyikkan

Dengan metrik ini dirasa cukup sukses, karena peningkatan *user experience* tidak semua diukur dengan lebih cepatnya waktu *task* (Tom Tullis dan Bill Albert, 2008 hlm 74), karena pengguna mengaku mengalami pengalaman yang menyenangkan dan fungsi *checkout_time*, nilai yang gagal (kondisi tidak terpenuhi) adalah 0 menit.

Tabel 4.3 Pengukuran dengan Metrik Efisiensi - *Lostness*

Skenario Tes	Efisiensi (<i>Lostness</i>)	
	Situs Kacamata Biasa	Situs Kacamata Virtual
Menyelesaikan transaksi	0.95	0.91
Membandingkan produk	0.72	0.90
Evaluasi produk yang sama	0.73	0.84
Registrasi	0.97	0.95

Pada tabel 4.3 terlihat bahwa efisiensi pada situs kacamata virtual meningkat 0.04 (dari 0.95 menjadi 0.91, karena *lostness* semakin kecil semakin baik) dan pada registrasi meningkat 0.03 (dari 0.97 menjadi 0.95). Dan mengalami penurunan pada membandingkan produk serta evaluasi produk yang sama dikarenakan masalah yang sama seperti pada metrik *task success* yaitu koneksi internet yang lamban.

Tabel 4.4 Pengukuran dengan *Self Reported Metrics*

Total user positif	<i>Self Reported Metrics</i>		
	Total	Jumlah positif	Persentase
Situs Kacamata Biasa	4	3	75%
Situs Kacamata Virtual	8	8	100%

Pada tabel 4.3 terlihat jumlah pengguna yang memasukkan nilai survei jika situs kacamata membantu pengguna untuk menemukan kacamata yang dia cari, pada situs kacamata biasa menampilkan 3 pengguna menganggap terbantu dari total 4 yang memasukkan nilai survei dan dengan persentase 75% dan pada situs kacamata *online* berbasis *Augmented Reality*., mengalami peningkatan nilai menjadi 8 orang yang memasukkan survey dengan persentase 100%.

4.3 Kelebihan dan Kekurangan Sistem.

Berdasarkan implementasi sistem, didapatkan keunggulan dan kelemahan sistem. Sistem dapat mengukur *User Experience* secara otomatis melalui metrik-metrik yang ditentukan. Representasi data yang ditampilkan karena dalam bentuk angka dan grafik, sangat membantu admin dalam lebih baik memahami kesalahan yang terjadi (pada skenario tes yang spesifik) sehingga tidak perlu melakukan survei secara

menyeluruh (seperti pada kasus penurunan nilai kesuksesan skenario tes yang diakibatkan oleh relatif beratnya kaca mata virtual bagi beberapa user).

Kekurangan sistem, sistem dalam hal kode pemrograman terintegrasi permanen ke masing-masing website (menjadi kode di dalam situs) sehingga tidak dinamis, penambahan tes skenario dilakukan secara manual dengan pemrograman. Penarikan solusi harus ditarik oleh admin sendiri karena sistem hanya menampilkan data yang telah diukur sebelumnya. Pengembangan kedepannya bisa ditambahkan dengan metrik-metrik lain agar pengukuran *user experience* menjadi lebih spesifik dan menjadi lebih mudah untuk penarikan solusi (misal menambahkan metrik *error* pada sistem sehingga mengetahui apa dan di mana kesalahan yang terjadi)

©UKDW

Kode Program (Pengukuran pada Situs Kacamata Online berbasis Augmented Reality)

```
<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class User_ex extends CI_Controller {

    public function index(){

        $this->load->view('user_ex/index');

    }

    public function task_success(){

        $data = array();

        $query = "SELECT distinct aktivitas_ip_address from aktivitas
where aktivitas_status=1";

        $q = $this->db->query($query);

        foreach($q->result() as $r){

            $task[$r->aktivitas_ip_address][] = $this->checkout($r-
>aktivitas_ip_address);

            $task[$r->aktivitas_ip_address][] = $this-
>compare_product($r->aktivitas_ip_address);

            $task[$r->aktivitas_ip_address][] = $this-
>frequent_use($r->aktivitas_ip_address);

            $task[$r->aktivitas_ip_address][] = $this-
>maximizing_usability($r->aktivitas_ip_address);

        }

        $task_name[] = 'Completing a transaction';

    }

}
```

```
$task_name[] = 'Comparing Product';

$task_name[] = 'Frequent use of the same product';

$task_name[] = 'Maximizing usability for critical product';

$data['global'] = $q;

$data['task'] = $task;

$data['task_name'] = $task_name;

$data['total_data'] = $q->num_rows();

$this->load->view('user_ex/task_success',$data);
}

public function checkout($ip){

    $this->db->where('aktivitas_status',1);

    $this->db->where('aktivitas_ip_address',$ip);

    $this->db->like('aktivitas_uri','checkout');

    $query = $this->db->get('aktivitas');

    $task_value = 0;

    if($query->num_rows() > 0){

        $task_value = 1;

    }

    return $task_value;

}

public function compare_product($ip){

    $this->db->distinct();
```

```
$this->db->select('aktivitas_uri');

$this->db->where('aktivitas_status',1);

$this->db->where('aktivitas_ip_address',$ip);

$this->db->like('aktivitas_uri','bandingkan');

$query = $this->db->get('aktivitas');

$task_value = 0;

if($query->num_rows() > 1){

    $task_value = 1;

}

return $task_value;

}

public function frequent_use($ip){

    $query = "SELECT aktivitas_uri, count(*) as frequent

    from aktivitas

    where aktivitas_status=1

    and aktivitas_ip_address='$ip'

    and aktivitas_uri like '%bandingkan%'

    group by aktivitas_uri";

    $execute = $this->db->query($query);

    $task_value = 0;

    foreach($execute->result() as $r){

        $frequent_use = $r->frequent;

        if($frequent_use >= 2){

            $task_value = 1;

            return $task_value;

        }

    }

}
```

```
        exit();
    }
}

return $task_value;
}

public function maximizing_usability($ip){
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','register');
    $query = $this->db->get('aktivitas');
    //pre($this->db->last_query());
    $task_value = 0;
    if($query->num_rows() > 0){
        $task_value = 1;
    }
    return $task_value;
}

public function task_success_bar($nilai_satu, $nilai_dua,
$nilai_tiga, $nilai_empat,$jenis){
    require_once ('j/jpgraph.php');
    require_once ('j/jpgraph_bar.php');

    $x = array('Completing a transaction',
               'Comparing Product',
               'Frequent use',
```

```
'Maximizing usability');  
  
$y = array($nilai_satu, $nilai_dua, $nilai_tiga, $nilai_empat);  
  
$graph = new Graph(650,320,'auto');  
  
$graph->SetScale("textlin");  
  
$graph->yaxis->SetTickPositions(array(0,20,40,60,80,100,120,140),  
array(10,30,50,70,90,110));  
  
$graph->SetBox(false);  
  
$graph->ygrid->SetFill(false);  
$graph->xaxis->SetTickLabels($x);  
$graph->yaxis->HideLine(false);  
$graph->yaxis->HideTicks(false,false);  
  
$bplot = new BarPlot($y);  
  
$graph->Add($bplot);  
  
$bplot->SetColor("white");  
if($jenis == 'biasa'){  
    $bplot->SetFillColor("#cc1111");  
}  
else{  
    $bplot->SetFillColor("#11cccc");  
}  
  
$bplot->SetWidth(45);
```

```
$graph->title->Set("Graphical Task Success");

$graph->Stroke();

}

/*-----TIME ON TASK-----*/

public function task_time(){

    $data = array();

    $query = "SELECT distinct aktivitas_ip_address from aktivitas
where aktivitas_status=1";

    $q = $this->db->query($query);

    foreach($q->result() as $r){

        $task[$r->aktivitas_ip_address][] = $this-
>checkout_time($r->aktivitas_ip_address);

        $task[$r->aktivitas_ip_address][] = $this-
>compare_product_time($r->aktivitas_ip_address);

        $task[$r->aktivitas_ip_address][] = $this-
>frequent_use_time($r->aktivitas_ip_address);

        $task[$r->aktivitas_ip_address][] = $this-
>maximizing_usability_time($r->aktivitas_ip_address);

    }

    $task_name[] = 'Completing a transaction';

    $task_name[] = 'Comparing Product';

    $task_name[] = 'Frequent use of the same product';

    $task_name[] = 'Maximizing usability for critical product';

    $data['task'] = $task;
```

```
$data['task_name'] = $task_name;

$data['total_data'] = $q->num_rows();

$this->load->view('user_ex/task_time',$data);
}

public function checkout_time($ip){
    //get checkout time
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','checkout');
    $this->db->order_by('id_aktivitas','asc');
    $query_b = $this->db->get('aktivitas');
    $task_value = 0;
    if($query_b->num_rows() > 0){
        //get list aktivitas
        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db->order_by('id_aktivitas','asc'); //yang
        $query_a = $this->db->get('aktivitas');
        $first_time = $query_a->row()->aktivitas_time;
        //pre($query_a->result());

        $success_time = $query_b->row()->aktivitas_time;
        $task_value = get_date_diff($success_time,$first_time);
    }
}
```

pertama

```
        return $task_value;
    }

    public function compare_product_time($ip){

        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db->order_by('id_aktivitas','asc'); //kecil ke besar
        $this->db->like('aktivitas_uri','bandingkan'); //compare
        $this->db->group_by('aktivitas_uri');
        $query_a = $this->db->get('aktivitas');
        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db->like('aktivitas_uri','add');
        $this->db->order_by('id_aktivitas','asc');
        $query_b = $this->db->get('aktivitas');

        $task_value = 0;
        if($query_b->num_rows() > 0 && $query_a->num_rows() >= 2){
            $first_time = $query_a->row()->aktivitas_time;
            $success_time = $query_b->row()->aktivitas_time;
            $task_value = get_date_diff($success_time,$first_time);
        }

        return $task_value;
    }

    public function frequent_use_time($ip){

        $query = "SELECT id_aktivitas, aktivitas_uri, aktivitas_time,
```



```
count(*) as jumlah from aktivitas

    where aktivitas_status = 1

    and aktivitas_ip_address = '$ip'

    and aktivitas_uri like '%bandingkan%'

group by aktivitas_uri

order by id_aktivitas asc";

$x = $this->db->query($query);

$task_value = 0;

if($x->num_rows() > 0){

    $y = 0;

    foreach($x->result() as $r){

        if($r->jumlah > 1 ){

            $y = $r->jumlah;

            $first_time = $r->aktivitas_time;

            break;

        }

    }

    if($y != 0){

        //get add to chart time

        $this->db->where('aktivitas_status',1);

        $this->db->where('aktivitas_ip_address','$ip');

        $this->db->like('aktivitas_uri','add');

        $this->db->order_by('id_aktivitas','asc');

        $query_b = $this->db->get('aktivitas');

        $task_value = 0;

        if($query_b->num_rows() > 0){

            //$first_time = $x->row()->aktivitas_time;
```

```

                                $success_time = $query_b->row()-
>aktivitas_time;

                                $task_value =
get_date_diff($success_time,$first_time);

                                }

                                }

                                //pre($task_value);

                                return $task_value;

                                }

public function maximizing_usability_time($ip){

    $this->db->where('aktivitas_status',1);

    $this->db->where('aktivitas_ip_address',$ip);

    $this->db->where('aktivitas_uri','welcome/register');

    $this->db->order_by('aktivitas_time','desc');

    $a = $this->db->get('aktivitas',1,1); //limit 1 offset 1

    $task_time = 0;

    //jika data ada
    if($a->num_rows() == 1){

        $baris = $a->row();

        $time_register = $baris->aktivitas_time;

        $session = $baris->aktivitas_session;

        $this->db->where('aktivitas_status',1);

        $this->db->where('aktivitas_session',$session);

        $this->db->where('aktivitas_ip_address',$ip);

```

```
        $this->db->where('aktivitas_uri','welcome/register_success');

        $this->db->order_by('aktivitas_time','desc');

        $b = $this->db->get('aktivitas',1,0); //limit 1 offset 0
        //pre($this->db->last_query());

        if($b->num_rows() > 0){

            $data = $b->row();

            $time_success_register = $data->aktivitas_time;

            $task_time =
            get_date_diff($time_success_register,$time_register);
            }

        }

        return $task_time;
    }

    public function task_time_bar($nilai_satu, $nilai_dua, $nilai_tiga,
    $nilai_empat, $jenis){

        require_once ('j/jpgraph.php');
        require_once ('j/jpgraph_bar.php');

        $x = array('Completing a transaction',
                    'Comparing Product',
                    'Frequent use',
                    'Maximizing usability');

        $y = array($nilai_satu, $nilai_dua, $nilai_tiga, $nilai_empat);

        $graph = new Graph(650,320,'auto');
```

```

        $graph->SetScale("textlin");

        $graph->yaxis->SetTickPositions(array(0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80),
array(3,8,13,18,23,28,33,38,43,48,53,58,63,68,73,78));

        $graph->SetBox(false);

        $graph->ygrid->SetFill(false);

        $graph->xaxis->SetTickLabels($x);

        $graph->yaxis->HideLine(false);

        $graph->yaxis->HideTicks(false,false);

        $blplot = new BarPlot($y);
        $graph->Add($blplot);

        $blplot->SetColor("white");
        if($jenis == 'biasa'){
            $blplot->SetFillColor("#cc1111");
        }else{
            $blplot->SetFillColor("#11cccc");
        }

        $blplot->SetWidth(45);
        $graph->title->Set("Graphical Task Time");

        $graph->Stroke();

    }

/*-----LOSTNESS-----*/
-----*/

    public function lostness(){

        $query = "SELECT distinct aktivitas_ip_address from aktivitas

```

```
where aktivitas_status=1";

    $q = $this->db->query($query);

    foreach($q->result() as $r){

        $lost[$r->aktivitas_ip_address][] = $this->lostness_complete_transaction($r->aktivitas_ip_address);

        $lost[$r->aktivitas_ip_address][] = $this->lostness_compare_product($r->aktivitas_ip_address);

        $lost[$r->aktivitas_ip_address][] = $this->lostness_frequent_use($r->aktivitas_ip_address);

        $lost[$r->aktivitas_ip_address][] = $this->lostness_maximizing_usability($r->aktivitas_ip_address);

    }

    $task_name[] = 'Completing a transaction';
    $task_name[] = 'Comparing Product';
    $task_name[] = 'Frequent use of the same product';
    $task_name[] = 'Maximizing usability for critical product';

    $data['task'] = $lost;
    $data['task_name'] = $task_name;
    $this->load->view('user_ex/lostness',$data);
}

public function rumus($n,$s,$r){

    $a = ($n/$s - 1) * ($n/$s - 1);

    $b = ($r/$n - 1) * ($r/$n - 1);

    $c = $a + $b;

    $y = sqrt($c);
```

```

        $z = number_format($y,'2','.','');

        return $z;

    }

    public function lostness_complete_transaction($ip){

        $this->db->where('aktivitas_status',1);

        $this->db->where('aktivitas_ip_address',$ip);

        $this->db->where('aktivitas_uri','welcome/checkout');

        $this->db->order_by('id_aktivitas','asc');

        $query_a = $this->db->get('aktivitas');

        $hasil = 1;

        if($query_a->num_rows() != 0){

            $this->db->where('aktivitas_status',1);

            $this->db->where('aktivitas_ip_address',$ip);

            $this->db->where_not_in('aktivitas_uri',array('welcome/index'));

            $this->db->where('id_aktivitas <', $query_a->row()->id_aktivitas);

            $this->db->group_by('aktivitas_uri');

            $query_n = $this->db->get('aktivitas');

            $n = $query_n->num_rows();

            $this->db->where('aktivitas_status',1);

            $this->db->where('aktivitas_ip_address',$ip);

            $this->db->where_not_in('aktivitas_uri',array('welcome/index'));

            $this->db->where('id_aktivitas <', $query_a->row()->id_aktivitas);

            $query_s = $this->db->get('aktivitas');

```

```

        $s = $query_s->num_rows();

        $r = 5;

        $hasil = $this->rumus($n,$s,$r);
    }
    return $hasil;
}

public function lostness_compare_product($ip){
    $this->db->where('aktivitas_status',1);
    $this->db->where('aktivitas_ip_address',$ip);
    $this->db->like('aktivitas_uri','add');
    $this->db->order_by('id_aktivitas','asc');
    $query_a = $this->db->get('aktivitas');
    $hasil = 1;
    if($query_a->num_rows() != 0){
        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db->where_not_in('aktivitas_uri',array('welcome/index'));
        $this->db->where('id_aktivitas <', $query_a->row()->id_aktivitas);

        $this->db->group_by('aktivitas_uri');
        $query_n = $this->db->get('aktivitas');
        $n = $query_n->num_rows();
        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address',$ip);
        $this->db-

```

```

>where_not_in('aktivitas_uri',array('welcome/index'));

        $this->db->where('id_aktivitas <', $query_a->row()-
>id_aktivitas);

        $query_s = $this->db->get('aktivitas');

        $s = $query_s->num_rows();

        $r = 2;

        $hasil = $this->rumus($n,$s,$r);

    }

    return $hasil;

}

public function lostness_frequent_use($ip){

    $this->db->where('aktivitas_status',1);

    $this->db->where('aktivitas_ip_address',$ip);

    $this->db->like('aktivitas_uri','add');

    $this->db->order_by('id_aktivitas','asc');

    $query_a = $this->db->get('aktivitas');

    $hasil = 1;

    if($query_a->num_rows() != 0){

        $id = $query_a->row()->id_aktivitas;

        $query = "SELECT id_aktivitas, aktivitas_uri, count(*)
as jumlah from aktivitas

        where aktivitas_status = 1

        and id_aktivitas < '$id'

        and aktivitas_ip_address = '$ip'

        and aktivitas_uri not in ('welcome/index')

```



```

        and aktivitas_uri like '%index%'

        group by aktivitas_uri

        order by id_aktivitas asc";

        $query_f_use = $this->db->query($query);
        $y = 0;

        foreach($query_f_use->result() as $r){

            if($r->jumlah > 1 ){

                $y = $r->jumlah;

                break;

            }

        }

        if($y != 0){

            //CARI N

            $this->db->where('aktivitas_status',1);

            $this->db->where('aktivitas_ip_address', $ip);

            $this->db->

            >where_not_in('aktivitas_uri',array('welcome/index'));

            $this->db->where('id_aktivitas <', $query_a-

            >row()->id_aktivitas);

            $this->db->group_by('aktivitas_uri');

            $query_n = $this->db->get('aktivitas');

            $n = $query_n->num_rows();

            $this->db->where('aktivitas_status',1);

            $this->db->where('aktivitas_ip_address', $ip);

            $this->db->

            >where_not_in('aktivitas_uri',array('welcome/index'));

            $this->db->where('id_aktivitas <', $query_a-

            >row()->id_aktivitas);

```

```

        $query_s = $this->db->get('aktivitas');

        $s = $query_s->num_rows();

        $r = 3;

        $hasil = $this->rumus($n,$s,$r);

    }

}

return $hasil;

}

public function lostness_maximizing_usability($ip){

    $this->db->where('aktivitas_status',1);

    $this->db->where('aktivitas_ip_address',$ip);

    $this->db->where('aktivitas_uri','welcome/register_success');

    $this->db->order_by('id_aktivitas','asc');

    $query_a = $this->db->get('aktivitas');

    $hasil = 1;

    if($query_a->num_rows() != 0){

        $this->db->where('aktivitas_status',1);

        $this->db->where('aktivitas_ip_address',$ip);

        $this->db->where_not_in('aktivitas_uri',array('welcome/index'));

        $this->db->where('id_aktivitas <', $query_a->row()->id_aktivitas);

        $this->db->group_by('aktivitas_uri');

        $query_n = $this->db->get('aktivitas');
    }
}

```

```

        $n = $query_n->num_rows();

        $this->db->where('aktivitas_status',1);
        $this->db->where('aktivitas_ip_address', $ip);

        $this->db->where_not_in('aktivitas_uri',array('welcome/index'));

        $this->db->where('id_aktivitas <', $query_a->row()-
>id_aktivitas);

        $query_s = $this->db->get('aktivitas');
        $s = $query_s->num_rows();
        $r = 2;
        $hasil = $this->rumus($n,$s,$r);
    }
    return $hasil;

/*-----THUMB-----
-----*/

public function thumb(){
    $data = array();
    $this->db->where('thumb_status',1);
    $thumb = $this->db->get('thumb');
    $data['thumb'] = $thumb;

    $this->load->view('user_ex/thumb', $data);
}

/*-----DATA GABUNGAN-----
-----*/

```

```

public function get_data_service(){

    $data = array();

    $curlinit =
curl_init("http://jogjainformatika.web.id/kacamata/user_ex/get_data");

    curl_setopt($curlinit, CURLOPT_RETURNTRANSFER, true);

    $json = curl_exec($curlinit);

    curl_close($curlinit);

    $decode = json_decode($json);

    $data['task_name_k'] = $decode->task_name;
    $data['task_success_k'] = $decode->task_success;
    $data['task_time_k'] = $decode->task_time;
    $data['lostness_k'] = $decode->lostness;
    $data['thumb_k'] = $decode->thumb;
    $data['total_data_k'] = $decode->total_data;

    $query = "SELECT distinct aktivitas_ip_address from aktivitas
where aktivitas_status=1";

    $q = $this->db->query($query);

    foreach($q->result() as $r){

        $task_sukses[$r->aktivitas_ip_address][] = $this-
>checkout($r->aktivitas_ip_address);

        $task_sukses[$r->aktivitas_ip_address][] = $this-
>compare_product($r->aktivitas_ip_address);

        $task_sukses[$r->aktivitas_ip_address][] = $this-
>frequent_use($r->aktivitas_ip_address);

        $task_sukses[$r->aktivitas_ip_address][] = $this-
>maximizing_usability($r->aktivitas_ip_address);

        $task_time[$r->aktivitas_ip_address][] = $this-
>checkout_time($r->aktivitas_ip_address);

        $task_time[$r->aktivitas_ip_address][] = $this-

```

```

>compare_product_time($r->aktivitas_ip_address);

        $task_time[$r->aktivitas_ip_address][] = $this-
>frequent_use_time($r->aktivitas_ip_address);

        $task_time[$r->aktivitas_ip_address][] = $this-
>maximizing_usability_time($r->aktivitas_ip_address);

        $lost_data[$r->aktivitas_ip_address][] = $this-
>lostness_complete_transaction($r->aktivitas_ip_address);

        $lost_data[$r->aktivitas_ip_address][] = $this-
>lostness_compare_product($r->aktivitas_ip_address);

        $lost_data[$r->aktivitas_ip_address][] = $this-
>lostness_frequent_use($r->aktivitas_ip_address);

        $lost_data[$r->aktivitas_ip_address][] = $this-
>lostness_maximizing_usability($r->aktivitas_ip_address);

    }

    $this->db->where('thumb_status',1);
    $thumb = $this->db->get('thumb');

    $data['task_success_v'] = $task_sukses;
    $data['task_time_v'] = $task_time;
    $data['lostness_v'] = $lost_data;
    $data['thumb_v'] = $thumb;
    $data['total_data_v'] = $q->num_rows();
    $this->load->view('user_ex/finish',$data);
}

public function bar_success($k1,$k2,$k3,$k4,$v1,$v2,$v3,$v4){
    require_once ('j/jpgraph.php');
    require_once ('j/jpgraph_bar.php');
}

```

```
$x = array('Completing a transaction',
          'Comparing Product',
          'Frequent use',
          'Maximizing usability');

$dataly=array($k1,$k2,$k3,$k4);
$data2y=array($v1,$v2,$v3,$v4);

$graph = new Graph(650,320,'auto');
$graph->SetScale("textlin");

$theme_class=new UniversalTheme;
$graph->SetTheme($theme_class);

$graph->yaxis->SetTickPositions(array(0,30,60,90,120,150),
array(15,45,75,105,135));

$graph->SetBox(false);

$graph->ygrid->SetFill(false);
$graph->xaxis->SetTickLabels($x);
$graph->yaxis->HideLine(false);
$graph->yaxis->HideTicks(false,false);

$b1plot = new BarPlot($dataly);
$b2plot = new BarPlot($data2y);
//$b3plot = new BarPlot($data3y);

$gbplot = new GroupBarPlot(array($b1plot,$b2plot));
```

```
$graph->Add($gbplot);

$b1plot->SetColor("white");

$b1plot->SetFillColor("#cc1111");

$b2plot->SetColor("white");

$b2plot->SetFillColor("#11cccc");

$graph->title->Set("Task Success Compare");

$graph->Stroke();

}

public function bar_time($k1,$k2,$k3,$k4,$v1,$v2,$v3,$v4){

    require_once ('j/jpgraph.php');

    require_once ('j/jpgraph_bar.php');

    $x = array('Completing a transaction',

               'Comparing Product',

               'Frequent use',

               'Maximizing usability');

    $data1y=array($k1,$k2,$k3,$k4);

    $data2y=array($v1,$v2,$v3,$v4);

    $graph = new Graph(650,320,'auto');

    $graph->SetScale("textlin");

    $graph->yaxis->SetTickPositions(array(0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80),

    array(3,8,13,18,23,28,33,38,43,48,53,58,63,68,73,78));

    $graph->SetBox(false);

    $graph->ygrid->SetFill(false);
```

```
$graph->xaxis->SetTickLabels($x);  
  
$graph->yaxis->HideLine(false);  
  
$graph->yaxis->HideTicks(false,false);  
  
$b1plot = new BarPlot($data1y);  
  
$b2plot = new BarPlot($data2y);  
  
$gbplot = new GroupBarPlot(array($b1plot,$b2plot));  
  
$graph->Add($gbplot);  
  
  
$b1plot->SetColor("white");  
$b1plot->SetFillColor("#cc1111");  
  
$b2plot->SetColor("white");  
$b2plot->SetFillColor("#11cccc");  
  
$graph->title->Set("Task Time Compare");  
$graph->Stroke();  
  
}  
  
}
```


Analisa Program

a. Task Success : Binary of Success.

Tabel Kacamata

IP Address	Completing a transaction	Comparing Product	Frequent use of the same product	Maximizing usability for critical product
120.175.238.93	0	1	1	1
103.247.42.195	1	1	1	1
118.136.102.187	0	1	1	0
139.195.243.39	0	1	1	0
50.17.151.94	0	1	0	0
118.99.89.119	1	1	1	1
202.62.25.135	0	1	0	0
36.72.163.171	0	1	0	0
141.0.8.146	0	1	0	0
180.253.206.88	0	1	1	1
36.77.223.118	0	1	1	0
112.215.36.144	0	1	1	0
114.79.12.145	0	1	1	0
141.0.8.140	0	1	0	0
202.67.41.51	0	1	0	0
202.62.25.167	0	1	1	1
180.254.161.70	0	1	1	0
36.74.118.167	0	1	1	0
114.121.152.195	1	1	1	1
141.0.8.122	0	1	1	0
Persentase	15%	100%	70%	30%

Tabel Kacamata Virtual

IP Address	Completing a transaction	Comparing Product	Frequent use of the same product	Maximizing usability for critical product
103.247.42.195	1	1	1	1
36.73.98.101	1	1	1	1
202.152.202.146	0	0	0	0
111.95.66.17	0	1	1	0
118.136.230.135	0	1	1	0
103.3.221.201	0	0	0	0
107.167.99.119	0	0	0	0
110.137.76.178	1	1	1	1
112.215.66.68	0	1	1	0
200.8.230.9	0	0	0	0
200.60.76.38	0	0	0	0
106.168.209.143	1	1	1	1
111.94.196.54	1	1	0	1
110.139.117.182	1	0	0	1
114.79.13.67	1	1	1	1
180.249.184.114	0	1	1	1
180.244.160.51	1	1	1	1
180.244.125.31	0	0	0	0
36.72.0.53	0	1	1	0
66.249.76.39	0	0	0	0
Persentase	40%	60%	55%	45%

b. Tiue on Task**Tabel Kacamata**

IP Address	Completing a transaction	Comparing Product	Frequent use of the same product	Maximizing usability for critical product
120.175.238.93	0.00menit (0 detik)	0.40menit (24 detik)	0.40menit (24 detik)	0.50menit (30 detik)
103.247.42.195	4.60menit (276 detik)	2.35menit (141 detik)	2.35menit (141 detik)	0.55menit (33 detik)
118.136.102.187	0.00menit (0 detik)	0.65menit (39 detik)	0.63menit (38 detik)	0.00menit (0 detik)
139.195.243.39	0.00menit (0 detik)	2.53menit (152 detik)	2.52menit (151 detik)	0.00menit (0 detik)
50.17.151.94	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
118.99.89.119	2.52menit (151 detik)	0.12menit (7 detik)	0.12menit (7 detik)	0.47menit (28 detik)
202.62.25.135	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
36.72.163.171	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
141.0.8.146	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
180.253.206.88	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
36.77.223.118	0.00menit (0 detik)	1.35menit (81 detik)	1.33menit (80 detik)	0.00menit (0 detik)
112.215.36.144	0.00menit (0 detik)	0.92menit (55 detik)	0.90menit (54 detik)	0.00menit (0 detik)
114.79.12.145	0.00menit (0 detik)	0.90menit (54 detik)	0.88menit (53 detik)	0.00menit (0 detik)
141.0.8.140	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
202.67.41.51	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
202.62.25.167	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
180.254.161.70	0.00menit (0 detik)	0.58menit (35 detik)	0.58menit (35 detik)	0.00menit (0 detik)

	detik)	detik)	detik)	
36.74.118.167	0.00menit (0 detik)	0.97menit (58 detik)	0.97menit (58 detik)	0.00menit (0 detik)
114.121.152.195	1.88menit (113 detik)	0.35menit (21 detik)	0.35menit (21 detik)	0.23menit (14 detik)
141.0.8.122	0.00menit (0 detik)	5.55menit (333 detik)	5.53menit (332 detik)	0.00menit (0 detik)
Jumlah	9.00menit (540 detik)	16.67menit (1000 detik)	16.57menit (994 detik)	1.75menit (105 detik)

Tabel Kacamata Virtual

IP Address	Completing a transaction	Comparing Product	Frequent use of the same product	Maximizing usability for critical product
103.247.42.195	9.47menit (568 detik)	2.73menit (164 detik)	2.73menit (164 detik)	0.70menit (42 detik)
36.73.98.101	6.07menit (364 detik)	3.33menit (200 detik)	3.33menit (200 detik)	0.00menit (0 detik)
202.152.202.146	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
111.95.66.17	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
118.136.230.135	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
103.3.221.201	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
107.167.99.119	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
110.137.76.178	13.48menit (809 detik)	6.67menit (400 detik)	6.67menit (400 detik)	0.90menit (54 detik)
112.215.66.68	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
200.8.230.9	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
200.60.76.38	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
106.168.209.143	5.48menit (329 detik)	1.48menit (89 detik)	0.93menit (56 detik)	1.78menit (107 detik)

111.94.196.54	12.07menit (724 detik)	6.57menit (394 detik)	0.00menit (0 detik)	2.07menit (124 detik)
110.139.117.182	9.18menit (551 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.95menit (57 detik)
114.79.13.67	6.98menit (419 detik)	2.55menit (153 detik)	2.55menit (153 detik)	0.00menit (0 detik)
180.249.184.114	0.00menit (0 detik)	14.03menit (842 detik)	14.03menit (842 detik)	1.98menit (119 detik)
180.244.160.51	16.60menit (996 detik)	10.57menit (634 detik)	10.57menit (634 detik)	2.47menit (148 detik)
180.244.125.31	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
36.72.0.53	0.00menit (0 detik)	5.35menit (321 detik)	5.35menit (321 detik)	0.00menit (0 detik)
66.249.76.39	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)	0.00menit (0 detik)
Jumlah	79.33menit (4760 detik)	53.28menit (3197 detik)	46.17menit (2770 detik)	10.85menit (651 detik)

c. Efisiensi : *Lost-ness*

Tabel Kacamata

IP Address	Completing a transaction	Comparing Product	Frequent use of the same product	Maximizing usability for critical product
120.175.238.93	1	0.50	0.71	0.92
103.247.42.195	0.72	0.75	0.60	0.86
118.136.102.187	1	0.42	0.25	1
139.195.243.39	1	0.87	0.76	1
50.17.151.94	1	1	1	1
118.99.89.119	0.60	0.33	0.60	0.71
202.62.25.135	1	1	1	1
36.72.163.171	1	1	1	1
141.0.8.146	1	1	1	1
180.253.206.88	1	1	1	1
36.77.223.118	1	0.42	0.25	1
112.215.36.144	1	0.33	1	1

114.79.12.145	1	0.60	0.42	1
141.0.8.140	1	1	1	1
202.67.41.51	1	1	1	1
202.62.25.167	1	1	1	1
180.254.161.70	1	0.33	0.60	1
36.74.118.167	1	0.52	0.40	1
114.121.152.195	0.75	0.66	0.49	0.86
141.0.8.122	1	0.71	0.56	1
jumlah :	19.07	14.44	14.64	19.35

Tabel Kacamata Virtual

IP Address	Completing a transaction	Comparing Product	Frequent use of the same product	Maximizing usability for critical product
103.247.42.195	0.79	0.85	0.76	0.90
36.73.98.101	0.85	0.86	0.76	0.91
202.152.202.146	1	1	1	1
111.95.66.17	1	1	1	1
118.136.230.135	1	1	1	1
103.3.221.201	1	1	1	1
107.167.99.119	1	1.00	1	1
110.137.76.178	0.76	0.82	0.71	0.88
112.215.66.68	1	1	1	1
200.8.230.9	1	1	1	1
200.60.76.38	1	1	1	1
106.168.209.143	0.78	0.84	0.73	0.91
111.94.196.54	0.71	0.68	0.52	0.84
110.139.117.182	0.71	0.54	0.32	0.80
114.79.13.67	0.80	0.85	0.73	0.92
180.249.184.114	1	0.94	0.86	0.97
180.244.160.51	0.79	0.86	0.76	0.91
180.244.125.31	1	1	1	1
36.72.0.53	1	0.83	0.71	1
66.249.76.39	1	1	1	1
jumlah :	18.19	18.07	16.86	19.04

Perbandingan Rata-Rata *Lostness*

	Kacamata Biasa	Kacamata Virtual
Completing a transaction	0.9535	0.9095
Comparing Product	0.722	0.9035
Frequent use of the same product	0.732	0.843
Maximizing usability for critical product	0.9675	0.952

d. *Self Reported Metrics.***Tabel Kacamata**

IP Address	Value
103.247.42.195	0
120.175.238.93	1
118.99.89.119	1
114.121.152.195	1

Tabel Kacamata Virtual

IP Address	Value
103.247.42.195	1
36.73.98.101	1
110.137.76.178	1
106.168.209.143	1
111.94.196.54	1
110.139.117.182	1
114.79.13.67	1
180.244.160.51	1

Perbandingan Persentase Nilai Persentase

	Total Data	Sukses	Persentase
Kacamata Biasa	4	3	75%
Kacamata Virtual	8	8	100%