IMPLEMENTASI ONE TIME PASSWORD BERBASIS GAMBAR PADA WEBSITE ECOMMERCE PTIIK UNTUK MENINGKATKAN SEKURITAS

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

WINNY AYU PARAMITA

NIM. 0910680092

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

MALANG

2014

LEMBAR PERSETUJUAN

IMPLEMENTASI ONE TIME PASSWORD BERBASIS GAMBAR PADA WEBSITE ECOMMERCE PTIIK UNTUK MENINGKATKAN SEKURITAS

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :
WINNY AYU PARAMITA
NIM. 0910680092

Telah diperiksa dan disetujui oleh:

Pembimbing I	Pembimbing II,
Himawat Aryadita, ST., M.Sc	**Aswin Suharsono, ST., MT.**
NIP. 19801018 200801 1 003	**NIK. 840919 06 1 1 0251**

LEMBAR PENGESAHAN

IMPLEMENTASI ONE TIME PASSWORD BERBASIS GAMBAR PADA WEBSITE ECOMMERCE PTIIK UNTUK MENINGKATKAN SEKURITAS

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh :

WINNY AYU PARAMITA
NIM. 0910680092

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 7 Januari 2014

Penguji I	**Penguji II**
Denny Sagita Rusdianto, S.Kom., M.Kom	**Dr. Eng Herman Tolle, ST., MT.**
NIK. 851124 06 1 1 0250	**NIP. 197408232000121001**

Penguji III

Aryo Pinandito, S.T., M.MT.
NIK. 830519 16 1 1 0374

Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN
ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 Januari 2014

Mahasiswa,

Winny Ayu Paramita

NIM. 0910680092

Kata Pengantar

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena hanya dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul "Implementasi *One Time Password* Berbasis Gambar Pada *Website Ecommerce* PTIIK Untuk Meningkatkan Sekuritas".

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan skripsi, diantaranya:

1. Orang tua yang telah memberikan dukungan moral dan material
2. Bapak Himawat Aryadita, ST., M.Sc, selaku dosen pembimbing I yang telah memberikan ilmu dan saran untuk proposal skripsi ini.
3. Bapak Aswin Suharsono, ST., MT., selaku dosen pembimbing II yang juga memberikan ilmu dan saran untuk proposal skripsi ini.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
5. Seluruh mahasiswa Program Teknologi Informasi dan Ilmu Komputer, khususnya teman – teman yang telah membantu terealisasinya skripsi ini.
6. Rekan-rekan kantor PT. Cendana Teknika Utama tempat dimana saya bekerja yang telah memberikan dukungan dan memberikan keleluasaan waktu untuk saya menyelesaikan skripsi ini

Penulis sadar bahwa skripsi ini masih banyak kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan skripsi ini. Semoga skripsi ini dapat memberikan manfaat dan kebaikan bagi banyak pihak serta bernilai ibadah di hadapan Allah SWT. Aamiin.

.

Malang, Desember 2013

Penulis

ABSTRAK

Winny Ayu Paramita. 2013. : Implementasi *One Time Password* Berbasis Gambar Pada *Website Ecommerce* PTIIK Untuk Meningkatkan Sekuritas. Skripsi Program Studi Informatika/ Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing : Himawat Aryadita, ST., M.Sc dan Aswin Suharsono, ST., MT.

Data yang tersimpan di internet belum terjamin keamanannya sehingga perlu diamankan dengan sistem autentikasi yang tepat. Mekanisme autentikasi yang sering dipakai sekarang adalah *textual password* dimana banyak dan bervariasinya karakter menyebabkan pengguna mudah lupa pada *password*nya. Mekanisme autentikasi *One Time Password* (OTP) berbasis gambar diharapkan dapat mengatasi dua masalah tersebut. OTP adalah mekanisme *login* menggunakan *password* yang hanya dapat digunakan satu kali saja. *Password* berbasis gambar digunakan karena percobaan ilmiah menunjukkan bahwa manusia sangat baik dalam mengidentifikasi, mengingat, dan menampung pola gambar daripada pola teks. OTP berbasis gambar yang diimplementasikan pada website *ecommerce* Fakultas PTIIK Universitas Brawijaya ini bertujuan untuk mendesain sistem autentikasi yang aman dari serangan *Man In The Middle pasif* dan *Brute Force* kemudian ingin mengetahui bagaimana respon pengguna. Pengujian validasi menggunakan metode *black-box* menghasilkan sistem 100% valid. *User acceptance testing* dilakukan dengan membagikan 50 kuesioner yang menghasilkan sebanyak 34,7% responden belum merespon positif metode ini karena beberapa faktor yaitu karena karakter pada gambar selalu berubah sehingga pengguna perlu melihat satu persatu, gambar kurang umum sehingga susah diingat dan faktor keterbiasaan. Pengujian keamanan dengan melakukan serangan *Man In The Middle* pasif dan serangan *brute force* melalui program yang telah ada menghasilkan bahwa *password* tidak dapat ditemukan. *Password* dapat diketahui apabila penyerang mengetahui alur dari metode ini kemudian mengimplementasikannya ke dalam sebuah program *brute force*.

Kata Kunci : *one time password, graphical password,* keamanan, *ecommerce*

ABSTRACT

Winny Ayu Paramita. 2013. : ***Implementation of Images Based One Time Password in PTIIK's Ecommerce Website for Increase The Security. Undergraduate Thesis of Informatic Engineering Study Program, Information Technology and Computer Science Program, Brawijaya University, Malang. Advisor:*** **Himawat Aryadita, ST., M.Sc** ***and*** **Aswin Suharsono, ST., MT.**

Data which stored on the internet is not secured enough so it needs to be secured by right authentication system. Authentication mechanism that is often used now is textual passwords where many and varied characters cause users easily forget the password. One Time Password (OTP) authentication mechanism image based is expected to solve both issues. OTP is a mechanism to log in using password that can only be used one time. Image-based passwords is used because a scientific experiment show that humans are very good in identifying, remembering, and accommodate image patterns rather than text patterns. OTP image based which are implemented in ecommerce website PTIIK UB Faculty aims to design a secure authentication system of the Man In The Middle attacks and Brute Force passive then wanted to know how the user response. Validation testing using a black-box system produces 100 % valid. User acceptance testing is done by distributing 50 questionnaires that produce that 34.7 % respondents have not responded positively to this method because of these factors: because the characters in the picture is always changing so the user needs to see one by one, the picture is less common so it is difficult to remember and this method is unconventional. Testing security by performing passive Man In The Middle attacks and brute force attacks through existing programs resulted that password can not be found. Passwords can be known if the attacker knows the flow of this method then implementing it in a brute-force program.

Keywords : *one time password, graphical password, security, ecommerce*

DAFTAR ISI

Daftar Gambar

Daftar Tabel

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini menyebabkan maraknya kegiatan jual beli secara online. *Ecommerce* merupakan salah satu teknologi yang digunakan untuk melakukan proses perdagangan yang dilakukan melalui *World Wide Web*. Setiap pengguna yang ingin terlibat di dalam kegiatan *ecommerce* perlu mendaftarkan diri dengan mengisi data-data yang nantinya diperlukan untuk menunjang segala transaksi yang ada.

Data yang tersimpan di ruang internet belum tentu terjamin keamanannya. Data tersebut bisa dicuri atau disadap untuk kepentingan tertentu. Oknum-oknum tertentu bisa saja mencoba semua kemungkinan password yang mungkin (*Brute Force*) atau berusaha menyadap informasi yang kita kirim ke server dengan menggunakan serangan *Man in The Middle* untuk mencoba masuk ke dalam akun kita. Untuk mengatasi serangan-serangan itu, data dan informasi yang disimpan bisa diselamatkan dengan menggunakan sistem autentikasi yang tepat. Autentikasi adalah proses validasi pengguna pada saat memasuki sistem. Autentikasi bertujuan untuk membuktikan siapa anda sebenarnya, apakah anda benar-benar orang yang diklaim sebagai dia (*who you claim to be*) [RFD-11]. Salah satu mekanisme autentikasi yang digunakan untuk mencegah berbagai bentuk pencurian identitas dengan memastikan bahwa kombinasi nama atau password pengguna tidak dapat digunakan sebanyak dua kali adalah One Time Password (OTP). OTP merupakan mekanisme login ke dalam sebuah jaringan atau layanan dengan menggunakan password yang unik yang hanya dapat digunakan satu kali saja. [GEM-06]

Mekanisme autentikasi dibagi ke dalam tiga bentuk umum yaitu *token-based system, biometrics-based system,* dan *knowledge-based system. Token-based system* adalah autentikasi yang menggunakan objek yang bersifat eksklusif dan hanya dimiliki oleh pengguna tertentu sebagai identifikasi. *Biometrics-based system* menggunakan bagian tubuh tertentu misalnya sidik jari untuk proses

autentikasi. Sedangkan *knowledge-based system* terbagi menjadi dua yaitu *textual password* dan *graphical password* yang keduanya sama-sama menggunakan informasi dimana hanya diketahui oleh pengguna tersebut [HAR-10].

Mekanisme autentikasi yang sering dipakai sekarang adalah *textual password*. Banyak dan bervariasinya karakter pada *textual password* menyebabkan pengguna mudah lupa password yang telah dibuat sendiri. Hal ini ditunjang dengan percobaan ilmiah yang menunjukkan bahwa manusia sangat baik dalam mengidentifikasi, mengingat, dan menampung pola gambar daripada pola teks [SRN-67]. Untuk itu dalam penelitian kali ini penulis mengimplementasikan proses autentikasi dengan menggunakan gambar dan memadukan dengan teknologi OTP, di mana dalam penelitian sebelumnya, *"Secured Authentication Protocol System using Images"* telah dapat mengatasi masalah yang dihadapi dan dapat menjamin kerahasiaan dan autentikasi saat mengirim pesan [GAR-10].

Berdasarkan paparan informasi di atas, penulis mengambil judul skripsi "*Implementasi One Time Password Berbasis Gambar pada Website Ecommerce PTIIK untuk Meningkatkan Sekuritas*". Implementasi difokuskan pada kemanan data pada saat proses autentikasi.

1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang yang telah dikemukakan, maka masalah yang akan diteliti oleh penulis adalah:

1. Bagaimana mendesain sistem autentikasi yang aman untuk *website ecommerce* khususnya *website ecommerce* fakultas PTIIK Universitas Brawijaya?
2. Bagaimana implementasi OTP berbasis gambar pada *website ecommerce* fakultas PTIIK Universitas Brawijaya?
3. Apakah OTP berbasis gambar pada website *ecommerce* PTIIK dapat mencegah serangan *Man In The Middle* dan *Brute Force*?
4. Bagaimana respon dari pengguna saat menggunakan metode OTP berbasis gambar pada proses autentikasi?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Penelitian difokuskan pada metode *One Time Password* (OTP) berbasis gambar saat melakukan proses daftar, *login*, ubah akun dan *reset password* pada pembeli dan penjual
2. Pilihan gambar sebanyak 36 gambar untuk *password* sebagai percobaan pertama sudah ditetapkan.
3. Gambar yang telah dipilih dapat dipilih kembali sebagai *password*nya dan minimal enam karakter (tiga gambar) yang harus dimasukkan sebagai *password* dengan urutan gambar yang tidak boleh berubah.
4. Pengembangan *website ecommerce* PTIIK ini menggunakan metode *waterfall* yang terdapat proses *reuse* di dalamnya
5. Pengujian keamanan yang dilakukan adalah untuk mencegah seseorang mengetahui password pengguna dari serangan *Man In The Middle* berbentuk pasif dengan menggunakan *Wireshark* dan dari serangan *Brute Force* dengan menggunakan program yang sudah ada yaitu *Fireforce* dan program yang dibuat dengan menggunakan PHP Curl
6. Website *ecommerce* dalam penelitian ini menggunakan bahasa pemrograman PHP dengan *framework Code Igniter* dan basis data MySQL.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mendesain sistem autentikasi yang aman pada *website ecommerce* fakultas PTIIK Universitas Brawijaya
2. Mengimplementasikan OTP berbasis gambar pada *website ecommerce* fakultas PTIIK Universitas Brawijaya
3. Mencegah adanya serangan *Man in The Middle Attack* dan *Brute Force* pada *website ecommerce* fakultas PTIIK Universitas Brawijaya
4. Mengetahui respon pengguna terhadap metode OTP berbasis gambar pada *website ecommerce* PTIIK

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1.5.1 Bagi Penulis

1. Dapat lebih memahami tentang metode OTP berbasis gambar.
2. Mengimplementasikan pengetahuan tentang keamanan jaringan dan pemograman web

1.5.2 Bagi Pembaca/ Pengguna

1. Mendapatkan wawasan akan pengimplementasian dari OTP berbasis gambar pada website *ecommerce*.
2. Mendapatkan rasa aman pada data-datanya yang tersimpan di *website ecommerce*

1.6 Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan, dan waktu pengerjaan.

BAB II Dasar Teori

Membahas teori-teori yang mendukung dalam pengembangan dan perancangan implementasi *one time password* berbasis gambar pada website *ecommerce*.

BAB III Metodologi dan Perancangan

Membahas tentang metode yang digunakan dalam penulisan yang terdiri dari studi literatur, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis dan membahas tentang analisa kebutuhan dari implementasi *one time password* berbasis gambar pada website *ecommerce* dan kemudian merancang hal-hal yang berhubungan dengan analisa tersebut

BAB IV Implementasi

Membahas tentang hasil perancangan dari analisis kebutuhan dan implementasi sistem aplikasi.

BAB V Pengujian

Memuat tentang hasil pengujian dan analisis terhadap sistem yang telah direalisasikan.

Bab VI Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

BAB II
TINJAUAN PUSTAKA

Pada bab dua, terdiri dari kajian pustaka dan dasar teori. Kajian pustaka adalah membahas penelitian yang telah ada dan yang diusulkan. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Pada penelitian ini, dasar teori yang diperlukan adalah dasar teori yang berdasarkan latar belakang dan rumusan masalah.

2.1 Kajian pustaka

Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan. Penelitian sebelumnya berjudul "*Secured Authentication Protocol System using Images"*. Penelitian yang diusulkan saat ini akan diimplementasikan pada *website ecommerce* PTIIK. Perbedaan antara penelitian sebelumnya dan yang diusulkan adalah pada proses autentikasi dan pengujian yang dilakukan. Pada penelitian sebelumnya menggunakan *hidden characters* untuk melakukan proses *mapping* terhadap komponen-komponen yang ada sedangkan pada penelitian yang diusulkan menggunakan tabel yang menyimpan urutan gambar dan urutan karakter untuk proses *mapping* terhadap komponen-komponen yang ada. Perbedaan yang kedua adalah pada sisi pengujian. Penelitian sebelumnya lebih difokuskan pada pengujian keamanan sedangkan penelitian yang diusulkan akan menguji dari sisi keamanan dan dari sisi pengguna (*user acceptance testing*)

2.2 Autentikasi

Autentikasi adalah suatu mekanisme yang digunakan oleh suatu sistem untuk mengidentifikasi user yang berhak mengakses informasi pada sistem tersebut. Mekanisme autentikasi yang ada saat ini dibagi ke dalam tiga bentuk umum:

a. *Token-based system* menggunakan suatu objek yang eksklusif yang hanya dimiliki oleh *user* tertentu sebagai bentuk identifikasi. Salah satu contohnya adalah penggunaan kartu ATM, dimana setiap pemilik memiliki satu kartu untuk proses autentikasi.
b. *Biometrics-based system* menggunakan ciri khas tubuh tertentu dari *user* sebagai proses autentikasi. Cara ini memiliki tingkat keamanan yang

sangat tinggi, namun belum banyak digunakan karena membutuhkan biaya yang cukup mahal dan proses identifikasi yang cukup lama. Beberapa contohnya adalah *fingerprints*, iris *scan*, dan *facial recognition*.

c. *Knowledge-based system* menggunakan suatu informasi yang hanya diketahui oleh *user*. Sistem ini dibagi ke dalam dua kategori umum yaitu tekstual *password* dan *graphical password*.[HAR-10]

Autentikasi dapat dikatakan aman apabila terhindar dari *password attacks*. *Password attacks* yang dapat terdiri dari:

1. *Brute Force Attacks*

   Pada tipe serangan ini, semua kombinasi *password* yang mungkin dicoba untuk menemukan *password* pengguna. *Brute force attack* biasanya digunakan untuk membuka *password* yang dienkripsi dimana *password* tersebut disimpan di form teks terenkripsi.

2. *Dictionary Attack*

   Tipe serangan ini relatif lebih cepat daripada *brute force*. tidak seperti *brute force* yang mencoba semua kemungkinan, *dictionary attack* mencoba menemukan *password* dengan kata-kata yang sering digunakan untuk dijadikan *password*. Meskipun *dictionary attack* lebih cepat daripada *brute force, dictionary attack* ini juga mempunyai keterbatasan. *Dictionary attack* bisa saja tidak dapat menemukan *password* pengguna karena tidak terdapat pada daftar *password* tersebut

3. *Phishing Attacks*

   Phising attacks adalah serangan berbasis *web* dimana penyerang mengalihkan pengguna ke *website* palsu untuk mendapatkan *password* atau kode PIN dari pengguna tersebut.

4. *Shoulder Surfing*

   Shoulder surfing adalah salah satu alternatif memata-matai dimana penyerang mengamati pergerakan dari pengguna saat memasukkan *password*

5. *Key Loggers*

   Program ini memonitor aktivitas pengguna dengan merekam setiap tombol ditekan oleh pengguna. Penyerang menginstal *software key logger*

ke dalam sistem pengguna, baik dengan menginstal sendiri atau dengan menipu pengguna untuk mengklik agar menginstal *file* tersebut dalam sistem. *Key logger* membuat *file log* dari tombol ditekan oleh pengguna dan kemudian mengirimkan *file log* ke alamat *e-mail* penyerang. Penyerang kemudian mendapatkan *password* dan dapat mengakses ke sistem target

6. *Video Recording Attack*

   Dalam jenis serangan para penyerang dengan bantuan kamera yang dilengkapi ponsel atau kamera mini, menganalisis rekaman video dari pengguna yang memasukkan *password*.

7. *Replay Attacks*

   Replay attack adalah bagian dari *Passive Man In the Middle Attack*. *Man in the middle* pasif adalah serangan pada jaringan dimana penyerang "mendengar" percakapan antara pengirim (AP) dan penerima (*Client*) seperti mengambil sebuah informasi yang bersifat rahasia seperti pada proses autentikasi, lalu *hacker* menggunakan informasi tersebut untuk berpura-pura menjadi client yang terautentikasi [RIS-12].

Pada skripsi kali ini, penulis menggunakan *brute force* dan *man in the middle* pasif untuk pengujian keamanan autentikasi.

2.3 Tekstual Password

Tekstual *password* adalah bentuk yang paling umum dan sering digunakan saat ini. Tekstual *password* menggunakan karakter alfanumerik (ASCII) untuk mengidentifikasi *user*. *User* akan diminta untuk memasukkan kombinasi dari beberapa karakter pada proses autentikasi.

Berdasarkan studi yang dilakukan, user lebih sering menggunakan kata-kata yang pendek dan mudah diingat sebagai *password*. Sayangnya, bentuk *password* seperti ini sangat mudah ditebak. Di sisi lain, *password* yang susah ditebak cukup sulit untuk diingat. Dengan kemampuan user yang hanya mampu mengingat beberapa *password* saja, *user* cenderung mencatat semua *password* atau menggunakan *password* yang sama pada semua akun yang dimiliki. Selain masalah daya ingat manusia, saat ini juga sedang marak penggunaan *spyware*

untuk menangkap informasi berupa "*username*" dan "*password*" yang diketikkan *user* untuk dikirim ke penyerang. Hal ini menandakan bahwa tekstual *password* yang menggunakan input dari *keyboard* cukup mudah untuk diserang. Belajar dari kelemahan-kelemahan yang dimiliki textual *password*, kemudian diciptakanlah suatu metode baru yang dikenal dengan *Graphical Password*. [HAR-10]

2.4 Graphical Password

Metode *Graphical Password* pertama kali dikemukakan oleh G. Blonder pada 1996. Graphical password dianggap mampu menggantikan tekstual *password* sebagai metode autentikasi *user*. Hal ini mengacu pada penelitian psikologis bahwa gambar lebih mudah untuk dikenali dan diingat oleh memori manusia jika dibandingkan dengan teks.

Berdasarkan teknik identifikasinya, *graphical password* terbagi dua yaitu:

a. *Recognition-based password*, pada model ini *user* diminta untuk mengenali gambar yang dipilih pada awal registrasi.
b. *Recall-based password*, pada model ini *user* diminta untuk membuat kembali gambar yang telah digambar atau dipilih pada awal registrasi.

Berdasarkan tipe latar gambar yang digunakan, *graphical password* terbagi dua yaitu:

a. *Image-based password*, pada model ini *password* menggunakan gambar sebagai *background* dari *password*. Model ini biasanya menggunakan teknik identifikasi *Recognition-based*..
b. *Grid-based password*, pada model ini *password* menggunakan *grid* sebagai *background* dari *password*. Model ini biasanya menggunakan teknik identifikasi *Recall-based*. [HAR-10]

2.5 *One Time Password*

One Time Password merupakan mekanisme *login* ke dalam sebuah jaringan atau layanan dengan menggunakan *password* yang unik yang hanya dapat digunakan satu kali saja. Hal ini digunakan untuk mencegah berbagai bentuk pencurian identitas dengan memastikan bahwa kombinasi nama atau *password* pengguna tidak dapat digunakan sebanyak dua kali. *One time password* adalah bentuk dari autentikasi yang kuat dan menawarkan perlindungan yang lebih

efektif untuk rekening bank, jaringan perusahaan dan sistem lain yang berisi data sensitif.

Dewasa ini sebagian besar jaringan perusahaan, situs *ecommerce* dan komunitas *online* hanya membutuhkan nama pengguna dan *password* statis untuk *login* dan akses ke dalam data yang bersifat pribadi dan sensitif. Meskipun metode autentikasi ini tidak menyusahkan, *password* statis merupakan bentuk perlindungan yang paling tidak aman terhadap pencurian identitas *online* seperti: *phishing, keyboard logging, serangan man in the middle* dan metode lain.

One time password dapat dihasilkan dalam beberapa cara dan masing-masing memiliki manfaat yang berbeda dari segi keamanan, kenyamanan, biaya dan akurasi. Solusi autentikasi yang kuat mengatasi keterbatasan *password* statis dengan memasukkan langkah keamanan tambahan. *One time password* yang bersifat sementara melindungi akses jaringan dan identitas digital end-user. *One time password* menambahkan tingkat tambahan perlindungan dan membuatnya sangat sulit bagi penipu untuk mengakses informasi yang tidak sah, jaringan atau rekening *online*.[GEM-06]

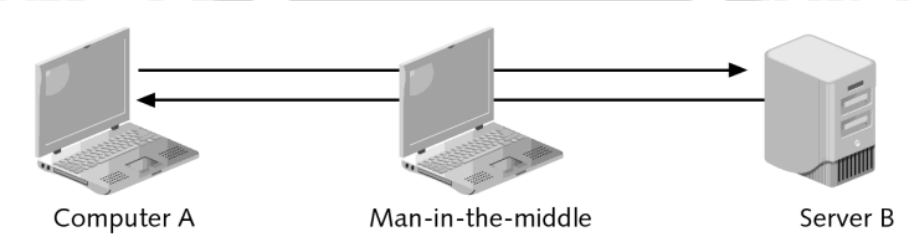
2.6 *Ecommerce*

Electronic commerce (ecommerce) adalah suatu penjualan secara elektronis, yang bisa dilakukan dari jarak jauh (teknologi marketing) yang digunakan di luar toko. Untuk tempat yang jauh sekalipun tetap dilakukan perdagangan dengan memanfaatkan *ecommerce*. Perubahan cara dan bentuk perdagangan telah mengubah, menggeser dan menaklukkan cara bisnis global yang tidak mengenal jarak dan waktu. Kegiatan yang dilakukan juga menjadi tidak banyak lagi diwakili oleh tenaga manusia di saat terjadi peningkatan keterpaduan telekomunikasi dan komputasi secara integral. [TAZ-10]

Pembeli yang akan berbelanja di toko *online* dapat menggunakan fasilitas *shopping cart. Shopping cart* adalah sebuah *software* di situs *web* yang mengijinkan pelanggan untuk melihat toko yang anda buka kemudian memilih item barang untuk diletakkan dalam kereta dorong yang kemudian membelinya saat melakukan *check out.* Konsep *shopping cart* ini meniru kereta belanja yang biasanya digunakan orang untuk berbelanja di pasar swalayan. *Shopping cart* biasanya berupa formulir dalam web.

2.7 *Man In The Middle*

Man in the middle dapat dilakukan pada sebuah jaringan ketika dua komputer saling berkomunikasi. Jenis serangan ini membuat seolah-olah hanya dua komputer yang terlibat saat sedang mengirim dan menerima data, sedangkan sebenarnya diantara mereka terdapat sebuah komputer yang dapat melihat lalu lintas data yang sedang dikirim.



Gambar 2.1 Serangan Man in The Middle
Sumber: [CDM-11]

Serangan *man in the middle* bisa berbentuk pasif atau aktif. Untuk penelitian ini, peneliti menggunakan *man in the middle* berjenis pasif sebagai langkah pengujian keamanan. Pada serangan pasif *man in the middle*, penyerang menangkap data yang terdeteksi.[CDM-11]

2.8 *Brute Force*

Sebuah serangan *brute force* adalah salah satu serangan yang melibatkan kode atau *password* dengan mencoba semua kombinasi yang mungkin sampai yang benar ditemukan. Ini bukan sesuatu yang mudah dilakukan dan mungkin membutuhkan waktu yang sangat lama, tergantung pada jenis enkripsi yang digunakan dan tingkat keamanan pada sistem tertentu. Hal ini dianggap sebagai jalan terakhir setelah penyerang tidak dapat mengakses sistem melalui cara lain . Tergantung pada jumlah atau kunci yang digunakan pada data yang terenkripsi, serangan *brute force* mungkin atau mungkin tidak dipertimbangkan. Saat sebuah kombinasi *password* mempunyai panjang kunci yang signifikan, seorang penyerang dapat memutuskan untuk mencoba memecahkan kode. Probabilitas untuk menemukan kombinasi yang tepat dalam waktu singkat menurun dengan setiap kunci tambahan yang digunakan. Setiap sistem dinilai berdasarkan seberapa mudah atau sulitnya untuk melancarkan serangan *brute force* terhadap sistem tersebut. Sebuah sistem yang bagus tidak akan mudah terganggu oleh serangan ini

dan dengan demikian sistem tersebut akan bernilai tinggi. Dalam teori serangan *brute force* selalu dicapai Namun, ada beberapa yang benar-benar dapat diserang dan akan membutuhkan milyaran tahun untuk menyelesaikan .

Serangan *brute force* dapat sangat efektif ketika berdiri sendiri atau bersama dengan serangan jenis lain. Pada dasarnya serangan *brute force* berhasil berdasarkan dari keahlian penyerang atau kelemahan dari sistem yang diserang. Salah satu cara serangan *brute force* dapat berhasil adalah dengan menggabungkan *dictionary attack* . Penggunaan kata-kata konvensional memudahkan pengguna individu untuk mengingat kode akses mereka , meskipun dengan mengorbankan tingkat keamanan yang lebih baik. Serangan *brute force* dengan mengambil keuntungan dari taktik ini dengan menggunakan *dictionary attack* untuk meningkatkan kemungkinan bahwa kode dari sebuah pengguna dapat ditemukan cukup cepat [ROL-13]. Namun pada penelitian kali ini, penulis tidak menggunakan metode *dictionary attack.* Penulis menggunakan metode *brute force* dengan mencoba kemungkinan yang ada dengan *range* karakter tertentu.

2.9 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan disiplin ilmu yang berkaitan dengan semua aspek produksi perangkat lunak dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah sistem digunakan. Dalam definisi rekayasa perangkat lunak terdapat dua frase kunci:

1. Disiplin rekayasa

   Perekayasa membuat suatu alat bekerja. Mereka menerapkan teori, metode, dan alat-alat dimana dapat digunakan secara tepat. Namun, mereka menggunakan secara selektif dan selalu mencoba untuk menemukan solusi dari suatu masalah bahkan ketika tidak terdapat dalam teori maupun metode. Perekayasa juga mengakui bahwa mereka harus bekerja dalam tekanan organisasi dan finansial sehingga merak mencari solusi dalam kondisi tersebut.

2. Semua aspek dari produksi perangkat lunak

   Perangkat lunak tidak hanya peduli dengan proses teknis dari pengembangan perangkat lunak. Hal ini juga mencakup kegiatan seperti

manajemen proyek perangkat lunak dan pengembangan alat, metode, dan teori untuk mendukung produksi perangkat lunak.

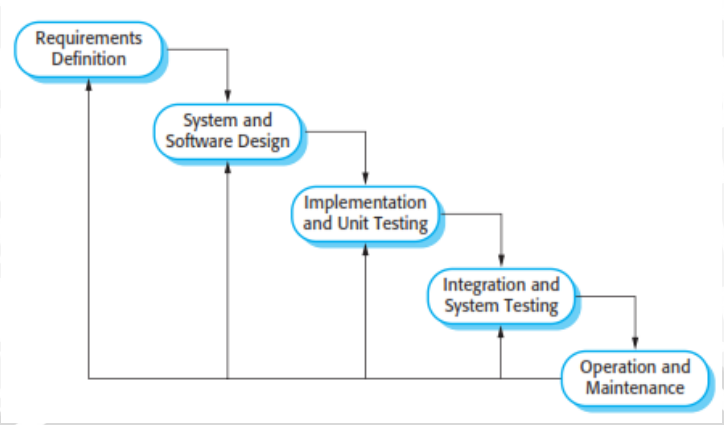
Secara umum, perekayasa perangkat lunak mengadopsi pendekatan yang sistematis dan terorganisir untuk bekerja, karena hal ini cara yang paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun, rekayasa adalah segala sesuatu tentang memilih metode yang paling sesuai untuk suatu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan [IAN-07].

2.10 Software Process Model

Model proses perangkat lunak adalah representasi yang sederhana dari proses perangkat lunak. Setiap model proses merupakan proses dari perspektif tertentu, dan dengan demikian hanya menyediakan informasi parsial tentang proses tersebut. Misalnya, model proses aktivitas yang menunjukan kegiatan dan urutan namun tidak menunjukan peran orang yang terlibat dalam aktivitas ini. Sebuah model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat proyek dan aplikasi, metode dan alat-alat yang akan digunakan, dan kontrol *deliverable* yang diperlukan. Beberapa model proses perangkat lunak yang sering digunakan para pengembang perangkat lunak adalah *waterfall model, incremental development,* dan *reuse-oriented software engineering.*

2.10.1 Waterfall Model

Secara umum, perekayasa perangkat lunak memakai pendekatan yang sistematis dan terorganisir terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun demikian, rekayasa ini sebenarnya mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan [IAN-07]. Proses pengembangan menggunakan model proses *waterfall* ini terlihat pada Gambar 2.2.



Gambar 2.2 Pemodelan waterfall
Sumber: [IAN-07]

Model proses untuk rekayasa perangkat lunak dipilih sesuai dengan sifat dari proyek dan aplikasi yang akan dibuat. Salah satu dari model proses yang digunakan adalah *waterfall model*. Model proses *waterfall* ini merekomendasikan pendekatan yang sistematis dan terurut (*systematic and sequential approach*) untuk pengembangan perangkat lunak yang dimulai dari analisis kebutuhan (*requirement analysis*), perancangan (*design*), implementasi (*coding*), pengujian (*testing*), dan pemeliharaan (*maintenance*).

2.10.2 Software Reuse

Software Reuse pada dasarnya adalah penggunaan kembali perangkat lunak yang telah ada. Hal ini sering terjadi ketika orang-orang yang bekerja pada proyek mengetahui desain atau kode yang mirip dengan apa yang dibutuhkan. Mereka mencari dan memodifikasi sesuai kebutuhan lalu menggabungkan ke dalam sistem. Pendekatan *reuse* mengandalkan komponen perangkat lunak yang dapat digunakan kembali dan mengintegrasikan kerangka untuk komposisi komponen sistem. [IAN-07]

Tahapan utama dari *reuse* secara langsung mencerminkan dasar pembangunan kegiatan:

1. Analisis komponen

   Mengingat pencarian spesifikasi kebutuhan dilakukan untuk komponen untuk mengimplementasikan spesifikasi tersebut. Biasanya tidak ada

komposisi yang tepat dan komponen yang dapat digunakan hanya menyediakan beberapa fungsi yang diperlukan.

2. Kebutuhan dalam modifikasi

   Selama tahap ini kebutuhan dianalisis menggunakan informasi tentang komponen yang telah ditentukan. Kemudian dimodifikasi untuk mencerminkan komponen yang tersedia. Dimana satu kondisi tidak memungkinkan memodifikasi, kegiatan analisis komponen dapat dilakukan kembali untuk mencari solusi alternatif.

3. Desain sistem dengan *reuse*

   Selama fase ini mendesain *framework* pada sistem atau *framework* yang tersedia akan digunakan kembali. Para desainer memperhitungkan komponen yang digunakan kembali dan mengatur *framework* untuk mengembangkannya.

4. Pengembangan dan integrasi perangkat lunak

   Perangkat lunak yang tidak dapat diperoleh secara eksternal maka akan dikembangkan dan komponen diintegrasikan untuk menciptakan sistem baru.

Pada penelitian ini penulis menggunakan metode *waterfall* yang didalamnya terdapat tahapan *reuse* sehingga tahapan pada penelitian ini adalah: analisis kebutuhan, analisis komponen, perancangan, implementasi, dan pengujian.

2.11 Pengujian Perangkat Lunak

Pengembangan sistem perangkat lunak melibatkan serangkaian kegiatan produksi di mana peluang untuk keteledoran manusia sangat besar. Kesalahan dapat muncul pada awal proses dimana kemungkinan terjadi kekeliruan pada tujuan atau tujuan tidak terspesifikasi secara tepat. Karena ketidakmampuan manusia dalam membuat sesuatu yang sempurna, maka pengembangan perangkat lunak disertai dengan aktivitas penjaminan kualitas. Pengujian perangkat lunak merupakan elemen penting dari jaminan kualitas perangkat lunak dan merepresentasikan tinjauan utama spesifikasi, desain, dan pembuatan kode [PRE-10].

2.11.1 Pengujian Validasi

Pada kulminasi pengujian terintegrasi, perangkat lunak secara lengkap dirakit sebagai suatu paket; kesalahan *interfacing* telah diungkap dan dikoreksi, dan seri akhir dari pengujian perangkat lunak, yaitu pengujian validasi dapat dimulai. Validasi dapat ditentukan dengan berbagai cara, tetapi definisi yang sederhana adalah bahwa validasi berhasil bila perangkat lunak berfungsi dengan cara yang dapat diharapkan secara bertanggung jawab oleh pelanggan. Validasi perangkat lunak dicapai melalui sederetan pengujian *black-box* yang memperlihatkan konformitas dengan persyaratan. Rencana pengujian menguraikan kelas-kelas pengujian yang akan dilakukan, dan prosedur pengujian menentukan *test case* spesifik yang akan digunakan untuk mengungkap kesalahan dalam konformitas dengan persyaratan. Baik rencana dan prosedur didesain untuk memastikan apakah semua persyaratan fungsional dipenuhi; semua persyaratan kinerja dicapai; dokumentasi betul dan direkayasa oleh manusia: dan persyaratan lainnya dipenuhi (transportabilitas, kompatibilitas, pembetulan kesalahan, maintanabilitas)

Black-box testing atau *behavioral testing* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan [PRE-10]

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface.*
3. Kesalahan dalam struktur data atau akses *basis data* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi

2.11.2 User Acceptance Testing

User Acceptance Testing (UAT) adalah salah satu prosedur proyek perangkat lunak akhir dan kritis yang harus terjadi sebelum perangkat lunak baru dikembangkan akan tergulir keluar ke pasar. Selama UAT, pengguna perangkat lunak sebenarnya menguji perangkat lunak untuk memastikan dapat menangani tugas-tugas yang diperlukan dalam skenario dunia nyata dan sesuai dengan spesifikasi [MAX-13].

UAT langsung melibatkan pengguna yang dituju perangkat lunak. UAT dapat diimplementasikan dengan membuat perangkat lunak yang tersedia untuk percobaan *beta* gratis di internet atau melalui tim pengujian yang terdiri dari pengguna perangkat lunak yang sebenarnya

2.11.3 Pengujian Keamanan

Pengujian (*Testing*) keamanan saat proses autentikasi dilakukan dengan dua jenis serangan yaitu *Man In The Middle* dan *Brute Force*. Untuk serangan Man In The Middle digunakan software penyadap *Wireshark*. *Wireshark* adalah sebuah *network packet analyzer* yang mencoba menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut sedetail mungkin [FAG-11].

Sedangkan untuk serangan *brute force*, peneliti menggunakan program *Fireforce* dan program yang dibuat dengan menggunakan CURL. *FireForce* adalah ekstensi *Firefox* yang dirancang untuk melakukan serangan *brute-force* pada *form GET* dan *POST*. *FireForce* dapat menggunakan kamus atau menghasilkan *password* berdasarkan beberapa jenis karakter. *FireForce* dapat digunakan pada *platform* apa saja yang menjalankan *browser web Firefox* [FRF-13]. CURL alias URL Client merupakan sebuah *library* yang berbasis *command line*, dimana dapat digunakan untuk memasukkan parameter ke dalam *web request*. *Libcurl* membuat semua jenis komunikasi antar sever mungkin terjadi dengan berbagai macam cara. Bisa dengan protokol http, https, ftp, gopher, telnet, dict, dan ldap dengan dukungan HTTPS *certificate*, HTTP POST, HTTP PUT, FTP *uploading*, *upload* berbasis HTTP *form*, proxy, cookies, bahkan autentikasi user dan password. Curl adalah *porting* ke PHP sebagai modul opsional dan dapat

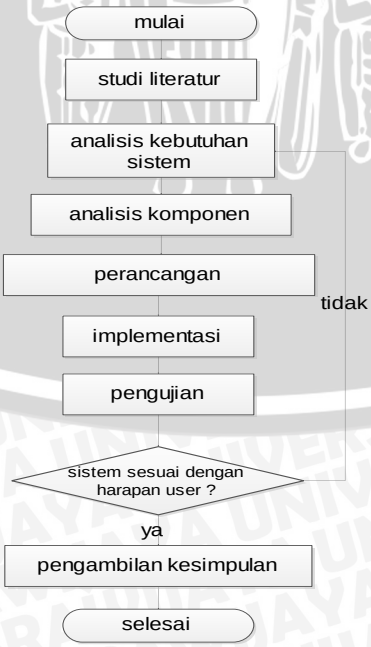
berguna untuk mendapatkan informasi pengintaian, atau akses tidak sah ke URL yang ditunjuk. Curl dapat digunakan bersama dengan *script* PHP untuk serangan *brute force*, serangan pengintai, *spoofing*, dan pencurian data. [HAK-13]

BAB III
METODE PENELITIAN DAN PERANCANGAN

Bab ini menjelaskan langkah-langkah pengembangan proses aplikasi yang ditempuh dalam penyusunan skripsi. Metode penelitian terdiri dari studi literatur untuk dasar teori, metode yang digunakan dalam perancangan, serta pengujian dan analisis. Sedangkan dalam perancangan menjelaskan analisis kebutuhan dan perancangan aplikasi.

3.1 Metode Penelitian

Pengembangan proses aplikasi dikembangkan dengan metode pengembangan perangkat lunak SDLC menggunakan model *waterfall* yang didalamnya terdapat *reuse* karena tahapan pengembangan aplikasi yang dilakukan oleh penulis sesuai dengan tahapan model *waterfall* yang akan digambarkan pada Gambar 3.1. Sedangkan untuk pemakaian *reuse*, penulis menyisipkan salah satu tahapan dari *software reuse* yaitu analisis komponen karena pada pengembangan aplikasi ini, penulis melakukan proses *reuse* terhadap komponen-komponen yang ada. Komponen-komponen tersebut digunakan untuk menunjang fungsionalitas dari *website ecommerce*. Pada skripsi ini, pengembangan *reuse* mayoritas dipakai untuk fungsi transaksi kegiatan penjualan pembelian.



Gambar 3.1 Diagram alir runtutan pengerjaan skripsi secara umum

3.1.1 Studi Literatur

Metode ini digunakan untuk mendapatkan dasar teori sebagai sumber acuan untuk penulisan skripsi dan pengembangan aplikasi. Teori dan pustaka yang berkaitan dengan tugas akhir ini meliputi :

a. *One Time Password*
b. *E-commerce*
c. *Man In the Middle*
d. *Brute Force*
e. Rekayasa Perangkat Lunak
    a. *Waterfall Model*
    b. *Software Reuse*
f. Pengujian Perangkat Lunak
    a. Teknik Pengujian
        i. Pengujian keamanan saat proses autentikasi
        ii. *User acceptance testing*
        iii. *Black-box Testing*
    b. Strategi Pengujian
        i. Menguji keamanan autentikasi dengan serangan *Brute Force* dan *Man In The Middle*
        ii. Mengetahui respon dari *user*
        iii. Pengujian Validasi

Studi literatur menjelaskan dasar teori yang digunakan sebagai penunjang dan pendukung penulisan skripsi. Teori penunjang dan pendukung skripsi ini meliputi *One Time Password, E-commerce, Man In the Middle,* dan *Brute Force*. Sumber atau referensi yang digunakan antara lain buku, jurnal, laporan penelitian, dan bantuan mesin pencari (*search engine*) internet.

3.1.2 Analisis Kebutuhan

Kegiatan analisis kebutuhan aplikasi meliputi analisis spesifikasi aplikasi. Metode analisis menggunakan bahasa pemodelan UML *(Unified Modeling Language). Use Case Diagram* digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas aplikasi dari perspektif *user*. Analisis kebutuhan

dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) aplikasi yang kemudian akan dimodelkan dalam diagram *use case*

3.1.3 Analisis Komponen

Pada tahapan pengembangan aplikasi ini, penulis menyisipkan salah satu tahapan dari metode pengembangan reuse yaitu analisis komponen. Analisis komponen dilakukan untuk mencari komponen dalam mengimplementasikan fungsionalitas dari *website ecommerce*. Komponen-komponen yang diperlukan dalam mengimplementasikan sesuai spesifikasi kebutuhan adalah *web framework, library cart, library jcrop* dan fungsi *reset email*

3.1.4 Perancangan

Setelah menentukan kebutuhan untuk membangun aplikasi, maka langkah selanjutnya yaitu desain atau perancangan sistem. Perancangan sistem yang akan dibuat ini secara garis besar meliputi:

a. Pemodelan *use case diagram*
b. Pemodelan *class diagram*
c. Pemodelan *activity diagram*
d. Perancangan algoritma aplikasi
e. Perancangan *user interface*

3.1.5 Implementasi

Implementasi aplikasi dilakukan dengan mengacu kepada perancangan aplikasi. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman berorientasi objek yaitu menggunakan implementasi basis data MySQL dengan *software* XAMPP 1.7.4, dan bahasa pemrograman PHP dengan *framework* CodeIgniter.

Secara umum, alur penggunaan *website* adalah pengunjung dapat melihat halaman *website* tanpa harus *login* dahulu. Jika pengunjung ingin terlibat dalam transaksi jual beli, pengunjung harus melakukan proses *register* menggunakan metode OTP berbasis gambar terlebih dahulu sebagai penjual atau pembeli. Jika pengunjung telah melakukan *register* sebagai penjual, pengunjung tersebut dapat *login* menggunakan metode OTP berbasis gambar sebagai penjual kemudian

mengunggah foto produk yang akan dijualnya beserta deskripsinya. Dan jika pengunjung telah melakukan *register* menggunakan metode OTP berbasis gambar sebagai pembeli, pengunjung tersebut dapat *login* menggunakan metode OTP berbasis gambar sebagai pembeli kemudian memilih produk yang ingin dibeli dan melakukan proses pembelian.

3.1.6 Pengujian dan Analisis

Pengujian dari aplikasi ini berkaitan dengan *Black-box Testing*, pengujian keamanan, dan *User Acceptant Testing*. Proses pengujian *black box* dilakukan melalui tahapan pengujian validasi. Pada pengujian validasi akan digunakan teknik pengujian *black box* (*Black Box Testing*).

Pengujian keamanan dilakukan dengan melakukan percobaan serangan *Man In The Middle* pasif dan *Brute Force* pada *website ecommerce* yang telah dibuat. Serangan *Man In The Middle* bisa dilakukan dengan menggunakan aplikasi *sniffer* yaitu Wireshark, sedangkan serangan *Brute Force* dilakukan dengan menggunakan *Fireforce* dan program yang dibuat dengan menggunakan PHP Curl

User Acceptance Testing dilakukan untuk mengetahui bagaimana respon pengguna atas metode OTP berbasis gambar ini. Pengujian dilakukan dengan melibatkan 50 pengguna yang mendaftarkan diri ke dalam *website ecommerce* ini. Kemudian pengguna akan mengisi kuisioner yang telah disediakan oleh penulis.

3.2 Perancangan

Untuk mengembangkan aplikasi, tahap perancangan dilakukan dengan cara menganalisis kebutuhan aplikasi dan merancang aplikasi sesuai dengan kebutuhan

3.2.1 Analisis Kebutuhan

Proses analisis kebutuhan diawali dengan penjabaran gambaran umum implementasi OTP berbasis gambar pada *website ecommerce*, identifikasi aktor, analisis data yang akan disimpan, penjabaran tentang daftar kebutuhan yang kemudian memodelkannya ke dalam diagram *use case*, analisis komponen, dan perancangan. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan – kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

Kebutuhan yang digunakan dalam pembuatan skripsi ini meliputi:

- Kebutuhan Hardware, meliputi:
  - Komputer PC / Laptop
- Kebutuhan Software, meliputi:
  - Microsoft Windows sebagai sistem operasi
  - Adobe Dreamweaver CS5 sebagai platform pengembangan
  - XAMPP 1.7.4 sebagai *server*
  - Edraw Max sebagai *tool* pembuatan diagram pemodelan sistem
- Data yang dibutuhkan meliputi:
  - Data gambar sebanyak 36 untuk pilihan *password*
  - Data *user* yang terdiri dari *id user*, *username*, *password*, *email*, status dan *level*.
  - Data produk yang dijual

3.2.1.1 Deskripsi Umum Perangkat Lunak

Aplikasi yang akan dikembangkan pada proyek skripsi ini adalah sebuah aplikasi berbasis *web ecommerce* yang proses autentikasinya menggunakan OTP berbasis gambar. Aplikasi berbasis web ini akan digunakan di Fakultas PTIIK Universitas Brawijaya. Secara fungsionalitas website ini tidak berbeda dengan website ecommerce pada umumnya, namun proses autentikasi yang diterapkan menggunakan metode *One Time Password* (OTP) berbasis gambar. Alur sistem secara umum digambarkan seperti gambar di bawah ini.

Gambar 3.2 Arsitektur sistem secara umum

Pengunjung yang mengunjungi website ini dapat melihat barang-barang yang dijual. Namun untuk melakukan kegiatan jual beli secara online, pengunjung tersebut harus mendaftarkan dirinya sebagai pembeli atau penjual. Setelah mendaftar, penjual dapat menjual produknya dan pembeli dapat membeli barang yang tersedia di *web*site ini. Sedangkan *administrator* bertugas memproses pesanan pembelian dari pembeli.

Proses daftar, *login*, ubah, dan *reset password* menggunakan metode OTP berbasis gambar. Pengguna mengisi data untuk akunnya seperti biasa. Namun untuk mengisi *field password*, pengguna perlu memilih minimal tiga gambar dari 36 gambar yang tersedia. Kemudian angka dan huruf yang tertera pada gambar yang telah menjadi pilihannya dimasukkan ke *field password*. Gambar-gambar yang dipilih itulah yang nantinya akan menjadi *password* pengguna tersebut. Angka dan huruf yang tertera pada gambar dan juga posisi gambar akan berubah setiap pengguna *reload* halaman. Hal ini mengakibatkan karakter yang dimasukkan oleh pengguna selalu berubah-ubah setiap ia melakukan proses daftar, *login*, ubah, atau *reset password*. Untuk itu, pengguna hanya perlu mengingat gambar yang telah dipilih secara berurutan untuk masuk ke dalam aplikasi.

3.2.1.2 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor – aktor yang akan berinteraksi dengan *website ecommerce* di Fakultas PTIIK Universitas Brawijaya Malang yang melakukan autentikasi dengan menggunakan OTP berbasis gambar. Tabel 3.1 memperlihatkan aktor – aktor yang terlibat beserta penjelasannya masing-masing yang merupakan hasil dari proses identifikasi aktor.

Tabel 3.1 Identifikasi Aktor

Aktor	Deskripsi
Pengunjung	Pengunjung adalah pengguna yang dapat melakukan proses *register*
Pembeli	Pembeli adalah pengguna yang dapat *login* ke dalam sistem setelah sebelumnya *register* untuk melakukan transaksi pembelian lebih lanjut

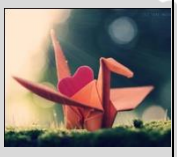
Penjual	Penjual adalah pengguna yang dapat *login* ke dalam sistem setelah sebelumnya *register* untuk melakukan penjualan

3.2.1.3 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur penyimpanan data yang dibutuhkan perangkat lunak. Struktur penyimpanan data pada aplikasi ini disusun berdasarkan analisis data sebagai berikut :

a. Data gambar yang digunakan sebagai pilihan *password user*. Berikut di bawah ini merupakan tabel daftar gambar yang digunakan:

Tabel 3.2 Data Gambar

b. Data pembeli yang terdiri dari *id user*, *username*, *password*, *email*, status dan *level* beserta detail dari profil pembeli
c. Data produk yang diupload oleh *administrator* atau penjual

3.2.1.4 Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional yang harus disediakan oleh sistem. Daftar kebutuhan ini disebut dengan *Software Requirement Specification* (SRS). Daftar kebutuhan fungsional dapat dilihat pada tabel-tabel di bawah ini.

Tabel 3.3 Spesifikasi kebutuhan fungsional pengunjung

Nomor SRS	Kebutuhan	*Use Case*
SRS_001_01	Sistem harus mampu menyediakan fasilitas pendaftaran untuk menjadi pembeli atau penjual	*Register*
SRS_001_02	Sistem harus mampu menyediakan fasilitas untuk mengisi *feedback*	Mengisi *feedback*

Tabel 3.4 Spesifikasi kebutuhan fungsional pembeli

Nomor SRS	Kebutuhan	*Use Case*
SRS_002_01	Sistem harus mampu menyediakan fasilitas *login* dengan menggunakan metode OTP berbasis gambar sehingga hanya yang sudah mendaftar saja yang bisa *login*	*Login*
SRS_002_02	Sistem harus menyediakan fasilitas me*reset password* bagi pembeli yang lupa akan *password*nya	Reset *Password*
SRS_002_03	Sistem harus mampu menyediakan fasilitas untuk mengisi *feedback*	Mengisi *feedback*

Tabel 3.5 Spesifikasi kebutuhan fungsional penjual

Nomor SRS	Kebutuhan	*Use Case*

SRS_003_01	Sistem harus mampu menyediakan fasilitas *login* dengan menggunakan metode OTP berbasis gambar sehingga hanya yang sudah mendaftar saja yang bisa *login*	*Login*
SRS_003_02	Sistem harus menyediakan fasilitas me*reset password* bagi pembeli yang lupa akan *password*nya	Reset *Password*
SRS_003_03	Sistem harus mampu menyediakan fasilitas untuk mengisi *feedback*	Mengisi *feedback*

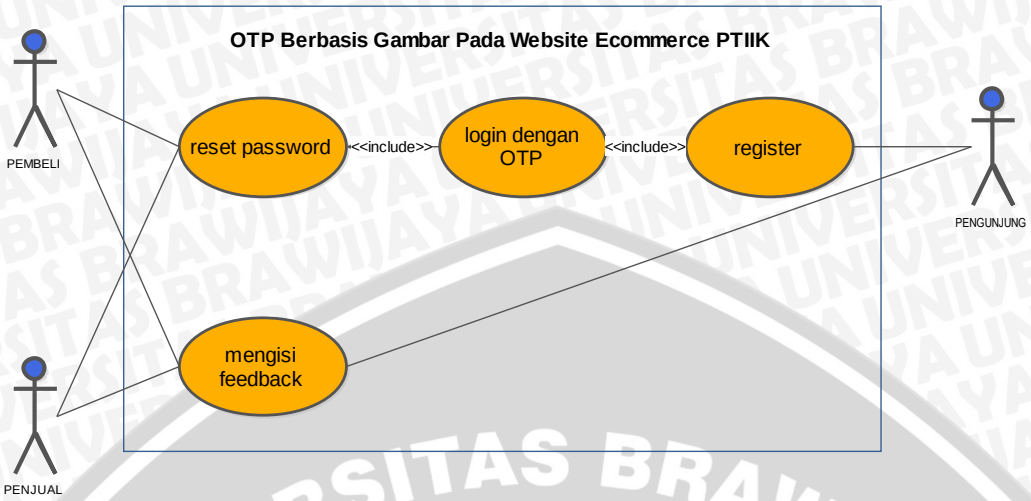
Daftar kebutuhan non-fungsional ditunjukkan pada tabel 3.6.

Tabel 3.6 Spesifikasi kebutuhan non-fungsional

Parameter	**Deskripsi Kebutuhan**
Security	Sistem keamanan perangkat lunak sebaiknya dapat terhindar dari ancaman serangan jaringan khususnya dari Man in The Middle dan Brute Force
Acceptance	Metode OTP berbasis gambar pada sistem ini sebaiknya dapat diterima atau direspon positif oleh pengguna

3.2.1.5 Diagram use case

Diagram *use case* adalah salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor. Diagram *use case* implementasi OTP berbasis gambar pada *website ecommerce* PTIIK dapat dilihat pada Gambar 3.4



Gambar 3.3 Diagram Use Case

3.2.1.6 Skenario Use Case

Masing-masing *use case* yang terdapat pada diagram *use case*, dijabarkan dalam skenario *use case* secara detail. Skenario *use case* akan memuat nama *use case*, aktor di dalam *use case* tersebut, tujuan dari *use case*, deskripsi global tentang *use case*, kondisi awal yang harus dipenuhi, dan kondisi akhir yang diharapkan setelah berjalannya fungsional *use case*. Selain itu juga akan diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh aktor (aliran utama atau *main flow)*, serta kejadian alternatif yang akan terjadi jika suatu kondisi tidak bisa terpenuhi (aliran alternatif atau *alternatif flow)*.

Tabel 3.7 Use case Register

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_001_01
Nama	*Register*
Tujuan	Untuk mendaftar menjadi pembeli atau penjual.
Deskripsi	*Use Case* ini menjelaskan bagaimana pengunjung melakukan proses pendaftaran untuk menjadi pembeli atau penjual
Aktor	Pengunjung
Skenario Utama	
Kondisi Awal	*Website ecommerce* sudah berjalan
Aksi Aktor	Reaksi Sistem
1. Pengunjung membuka *link* untuk *register*	1. Sistem mengacak gambar dan mengacak angka dan huruf untuk ditampilkan bersamaan dengan *form* pendaftaran pada halaman *register*. Susunan angka

2. Pengunjung memilih daftar sebagai pembeli atau penjual kemudian mengisi semua *field* yang ada. Khusus untuk mengisi password, pengunjung diharuskan memilih minimal tiga gambar dan memasukkan karakter yang tertera pada gambar-gambar yang dipilih tersebut di *field password* dan konfirmasi *password*	dan huruf dan gambar tersebut dimasukkan ke dalam database 2. Sistem menerima masukan dari pengunjung kemudian dmasukkan ke dalam database. Khusus untuk masukan password, karakter yang dimasukkan tersebut dicocokan dengan susunan id gambar dan karakter acak yang telah dimasukkan ke dalam database. Setelah itu *id* gambar tersebut dimasukkan ke dalam database sebagai *password*.
Skenario Alternatif 1 : Jika masukan tidak sesuai dengan *rule*	
	Menampilkan *alert* peringatan
Skenario Alternatif 2 : Jika *username* sudah digunakan	
	Menampilkan pesan bahwa *username* yang dimasukkan sudah terdaftar
Skenario Alternatif 3 : Jika karakter yang dimasukkan tidak sesuai dengan yang ditampilkan	
	Menampilkan pesan bahwa angka tidak tercantum
Kondisi Akhir	Pengunjung sudah terdaftar sesuai dengan *username* yang dimasukkan dan *password* gambar yang dipilih

Tabel 3.8 *Use case* Mengisi *Feedback*

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_001_02
Nama	Mengisi *feedback*
Tujuan	Untuk mengisi kuesioner
Deskripsi	*Use Case* ini menjelaskan bagaimana pengunjung dapat mengisi *feedback* berupa kuesioner
Aktor	Pengunjung
Skenario Utama	
Kondisi Awal	*Website ecommerce* sudah berjalan
Aksi Aktor	Reaksi Sistem
1. Pengunjung memilih menu	1. Sistem menampilkan kuesioner yang berasal dari *form google docs*

Feedback 2. Pengunjung mengisi kuesioner secara lengkap	2. Jawaban dari pengunjung disipan di dalam *google docs*
Kondisi Akhir	Pengunjung telah mengisi kuesioner

Tabel 3.9 Use case Login

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_002_01
Nama	*Login*
Tujuan	Untuk menyeleksi anggota yang sah.
Deskripsi	*Use case* ini menjelaskan bagaimana pembeli melakukan *login* untuk dapat menampilkan halaman utama sisi pembeli.
Aktor	Pembeli
Skenario Utama	
Kondisi Awal	Halaman *home website* sudah terbuka
Aksi Aktor	Reaksi Sistem
1. Pembeli membuka *link* untuk *login*	1. Sistem mengacak gambar dan mengacak angka dan huruf untuk ditampilkan bersamaan dengan *form login* pada halaman *login.* Susunan karakter dan gambar tersebut dimasukkan ke dalam database
2. Pembeli mengisi semua *field* yang ada. Khusus untuk mengisi *field* password, pembeli diharuskan memasukkan karakter yang tertera pada gambar-gambar yang dipilih saat melakukan *register*	2. Sistem menerima masukan dari pembeli kemudian karakter tersebut dicocokan susunan karakter dan gambar yang telah dimasukkan ke dalam database untuk dicocokkan kembali dengan *password* yang telah didaftarkan oleh pembeli
Skenario Alternatif 1 : Jika masukan tidak sesuai dengan yang terdaftar di database	
	Menampilkan pesan bahwa *username* atau *password* tidak benar
Skenario Alternatif 2 : Jika karakter yang dimasukkan tidak sesuai dengan yang ditampilkan	
	Menampilkan pesan bahwa angka tidak tercantum
Skenario Alternatif 3 : Jika masukan tidak sesuai dengan *rule* yang ada	
	Menampilkan *alert* peringatan
Kondisi Akhir	Pembeli dapat mengakses halaman khusus pembeli dan dapat melakukan kegiatan pembelian

Tabel 3.10 Use case Reset password

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_002_02
Nama	Reset *password*
Tujuan	Untuk menyetel ulang *password* karena lupa.
Deskripsi	*Use case* ini menjelaskan bagaimana pembeli yang lupa akan *password*nya bisa menyetel ulang kembali dengan *password* yang baru
Aktor	Pembeli
Skenario Utama	
Kondisi Awal	Pembeli sudah membuka halaman *login* namun lupa dengan *password*nya
Aksi Aktor	Reaksi Sistem
1. Pembeli membuka *link* Lupa Password 2. Pembeli memasukkan alamat *email*nya yang digunakan untuk mendaftar 3. Pembeli membuka link di *email* yang telah dikirim oleh sistem 4. Pembeli mengisi field password dan konfirmasi password dengan memasukkan karakter yang tertera pada gambar-gambar yang dipilihnya	1. Sistem menampilkan form untuk memasukkan alamat *email* 2. Sistem mengirim link untuk me*reset* ke alamat email yang terdaftar 3. Sistem mengacak gambar dan mengacak angka danhuruf untuk ditampilkan bersamaan dengan form *reset password*. Susunan karakter dan gambar tersebut dimasukkan ke dalam database 4. Sistem menerima masukan dari pembeli kemudian karakter tersebut dicocokan dengan susunan karakter dan gambar yang telah dimasukkan ke dalam database kemudian didapatkan *id* gambar sebagai password. Selanjutnya meng*update id* gambar yang lama dengan *id* gambar yang baru
Skenario Alternatif 1 : Jika *email* yang dimasukkan tidak terdaftar	
	Menampilkan pesan bahwa email tidak ditemukan
Skenario Alternatif 2 : Jika masukan dari pembeli tidak sesuai dengan aturan yang ada	
	Menampilkan pesan pembetulan
Skenario Alternatif 3 : Jika karakter yang dimasukkan tidak sesuai dengan yang ditampilkan	
	Menampilkan pesan bahwa angka tidak tercantum
Kondisi Akhir	*Password* pembeli ter*update* oleh *password* baru yang baru saja dimasukkan

Tabel 3.11 *Use case* Mengisi *Feedback*

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_002_03
Nama	Mengisi *feedback*
Tujuan	Untuk mengisi kuesioner
Deskripsi	*Use Case* ini menjelaskan bagaimana pembeli dapat mengisi *feedback* berupa kuesioner
Aktor	Pembeli
Skenario Utama	
Kondisi Awal	Pembeli sudah dalam keadaan *login*
Aksi Aktor	Reaksi Sistem
1. Pembeli memilih menu Feedback 2. Pembeli mengisi kuesioner secara lengkap	1. Sistem menampilkan kuesioner yang berasal dari *form google docs* 2. Jawaban dari pembeli disimpan di dalam *google docs*
Kondisi Akhir	Pembeli telah mengisi kuesioner

Tabel 3.12 Use case Login

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_003_01
Nama	*Login*
Tujuan	Untuk menyeleksi penjual yang sah.
Deskripsi	*Use case* ini menjelaskan bagaimana pembeli melakukan *login* untuk dapat menampilkan halaman utama sisi penjual.
Aktor	Penjual
Skenario Utama	
Kondisi Awal	Halaman *home website* sudah terbuka
Aksi Aktor	Reaksi Sistem
1. Penjual membuka *link* untuk *login*	1. Sistem mengacak gambar dan mengacak angka dan huruf untuk ditampilkan bersamaan dengan *form login* pada halaman *login.* Susunan angka-angka dan gambar tersebut dimasukkan ke dalam database
2. Penjual mengisi semua *field* yang ada. Khusus untuk mengisi *field* password, penjual diharuskan memasukkan karakter yang tertera pada gambar-gambar yang dipilih saat melakukan	2. Sistem menerima masukan dari penjual kemudian karakter tersebut dicocokan dengan susunan karakter dan gambar yang telah dimasukkan ke dalam database untuk kemudian dicocokkan dengan *password* yang telah didaftarkan oleh penjual

register	
Skenario Alternatif 1 : Jika masukan tidak sesuai dengan yang terdaftar di database	
	Menampilkan pesan bahwa *login* gagal
Skenario Alternatif 2 : Jika karakter yang dimasukkan tidak sesuai dengan yang ditampilkan	
	Menampilkan pesan bahwa angka tidak tercantum
Skenario Alternatif 3 : Jika masukan tidak sesuai dengan *rules* yang ada	
	Menampilkan pesan pembetulan
Kondisi Akhir	Penjual dapat mengakses halaman khusus penjual dan dapat melakukan kegiatan penjualan

Tabel 3.13 Use case Reset password

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_003_02
Nama	Reset *password*
Tujuan	Untuk menyetel ulang *password* karena lupa.
Deskripsi	*Use case* ini menjelaskan bagaimana penjual yang lupa akan *password*nya bisa menyetel ulang kembali dengan *password* yang baru
Aktor	Penjual
Skenario Utama	
Kondisi Awal	Penjual sudah membuka halaman *login* namun lupa dengan *password*nya
Aksi Aktor	Reaksi Sistem
1. Penjual membuka *link* Lupa Password 2. Penjual memasukkan alamat *email*nya yang digunakan untuk mendaftar 3. Penjual membuka link di *email* yang telah dikirim oleh sistem 4. Penjual mengisi field password dan konfirmasi password dengan memasukkan karakter yang tertera pada gambar-gambar yang dipilihnya	1. Sistem menampilkan form untuk memasukkan alamat *email* 2. Sistem mengirim link untuk me*reset* ke alamat email yang terdaftar 3. Sistem mengacak gambar mengacak angka dan huruf untuk ditampilkan bersamaan dengan form *reset password*. Susunan karakter dan gambar tersebut dimasukkan ke dalam database 4. Sistem menerima masukan dari penjual kemudian karakter tersebut dicocokan dengan susunan karakter dan gambar yang telah dimasukkan ke dalam database. Kemudian didapatkan *id* gambar sebagai password. Selanjutnya meng*update id* gambar yang lama dengan *id* gambar yang baru

Skenario Alternatif 1 : Jika *email* yang dimasukkan tidak terdaftar	
	Menampilkan pesan bahwa email tidak ditemukan
Skenario Alternatif 2 : Jika masukan dari penjual tidak sesuai dengan aturan yang ada	
	Menampilkan pesan pembetulan
Skenario Alternatif 3 : Jika karakter yang dimasukkan tidak sesuai dengan yang ditampilkan	
	Menampilkan pesan bahwa angka tidak tercantum
Kondisi Akhir	*Password* penjual ter*update* oleh *password* baru yang baru saja dimasukkan

Tabel 3.14 *Use case* Mengisi *Feedback*

Skenario Kasus Pada Sistem	
Nomor *Use Case*	SRS_003_03
Nama	Mengisi *feedback*
Tujuan	Untuk mengisi kuesioner
Deskripsi	*Use Case* ini menjelaskan bagaimana penjual dapat mengisi *feedback* berupa kuesioner
Aktor	Penjual
Skenario Utama	
Kondisi Awal	Penjual sudah dalam keadaaan *login*
Aksi Aktor	Reaksi Sistem
1. Penjual memilih menu Feedback 2. Pengunjung mengisi kuesioner secara lengkap	1. Sistem menampilkan kuesioner yang berasal dari *form google docs* 2. Jawaban dari penjual disimpan di dalam *google docs*
Kondisi Akhir	Penjual telah mengisi kuesioner

3.2.2 Analisa Komponen

Penulis menyisipkan salah satu tahapan dari metode pengembangan reuse yaitu analisis komponen. Analisis komponen dilakukan untuk mencari komponen yang dapat diimplementasikan untuk fungsionalitas dari *website ecommerce*. Komponen-komponen yang dibutuhkan oleh Impelementasi OTP Berbasis Gambar pada *Website Ecommerce* PTIIK adalah *framework* untuk pengembangan aplikasi beserta komponen untuk fitur yang terdapat pada *website ecommerce*. Tabel 3.13 menampilkan komponen yang dipilih untuk mengimplementasikan OTP Berbasis Gambar pada *Website Ecommerce* PTIIK.

Tabel 3.15 Daftar komponen

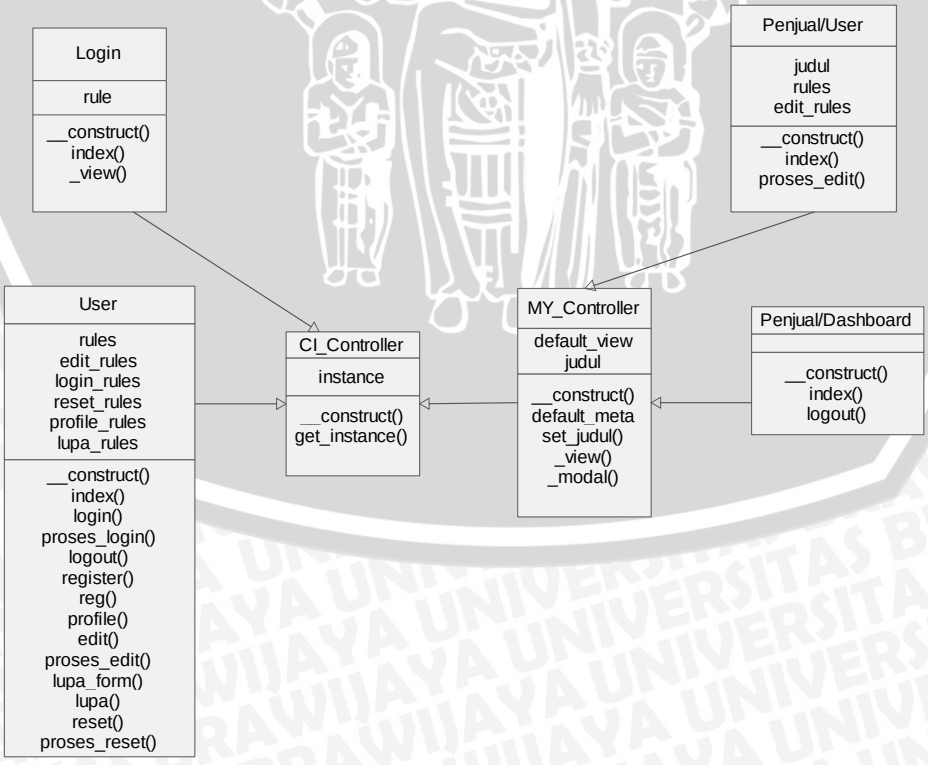
No	Komponen	Keterangan
1	*Framework Codeigniter*	*Web framework* untuk membuat *web*
2	Library cart	*Library* untuk menangani sistem keranjang belanja
3	Library jcrop	*Library* untuk upload image
4	Fungsi *Reset Password*	Fungsi untuk *reset password* dengan mengirim *email*

3.2.3 Perancangan

Pembuatan desain aplikasi dimodelkan dengan beberapa diagram. Pada proses desain dilakukan identifikasi terhadap *class-class* yang dibutuhkan yang dimodelkan dalam *class diagram*. Pada proses rinci yang terjadi di dalam sistem secara rinci dimodelkan dalam *activity diagram.*

3.2.3.1 Perancangan Kelas Aplikasi

Pemodelan kelas memberikan gambaran pemodelan elemen-elemen kelas yang membentuk sebuah perangkat lunak. Kelas bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan. Gambar 3.4 menunjukkan pemodelan diagram kelas dari aplikasi yang dibuat.



Gambar 3.4 Diagram Kelas

Semua kelas pada sistem mengalami pewarisan dari kelas Controller pada *framework CodeIgniter*. Pewarisan pada tiap kelas dikarenakan kelas-kelas tersebut merupakan penambahan dari *controller-controller* pada sistem agar dapat beroperasi sesuai dengan spesifikasi kebutuhan. Berikut penjelasan dari beberapa kelas pada Gambar 3.4

1. Kelas CI_Controller

Tabel 3.16 Penjelasan Kelas CI_Controller

Nama Kelas	CI_Controller
Deskripsi	Kelas dari komponen *framework CodeIgniter* yang berfungsi sebagai kelas utama yang dapat diakses oleh *controller* lain

2. Kelas MY_Controller

Tabel 3.17 Penjelasan Kelas MY_Controller

Nama Kelas	MY_Controller
Deskripsi	Kelas yang dibuat sebagai *extends* dari *core Controller*. MY_Controller digunakan untuk halaman admin yang berisi method-method untuk autentikasi, pengaturan css, dan pemanggilan view

3. Kelas User

Tabel 3.18 Penjelasan Kelas User

Nama Kelas	User
Deskripsi	Kelas ini dibuat untuk mendefinisikan proses autentikasi *user*

4. Kelas Login

Tabel 3.19 Penjelasan Kelas Login

Nama Kelas	Login
Deskripsi	Kelas ini berfungsi untuk melakukan verifikasi data yang masuk untuk dialihkan ke halaman yang sesuai dengan levelnya

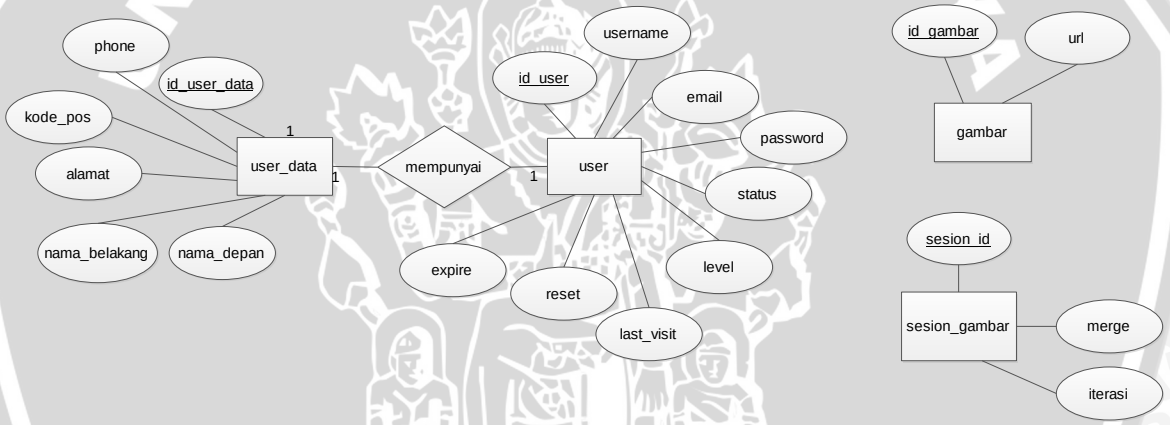
5. Kelas Penjual/ User

Tabel 3.20 Penjelasan Kelas Penjual/User

Nama Kelas	Penjual/User
Deskripsi	Kelas ini berfungsi untuk mengelola akun dari penjual

3.2.3.2 Perancangan Basis data

Basis data berfungsi sebagai tempat menyimpan data. Perancangan basis data digunakan untuk merancang basis data yang akan dibuat agar masukan dan keluaran program sesuai dengan apa yang diharapkan. Perancangan basis data mengambil acuan dari proses analisis data yang dilakukan pada tahap analisis kebutuhan



Gambar 3.5 Diagram *relationship*

Berikut ini merupakan struktur tabel serta keterangan masing masing tabel dan *field* yang ada pada *database*. Entitas user data merepresentasikan tabel user_data, yang berisi detail data diri pembeli. Struktur tabel user_data ditunjukkan pada Tabel 3.21

Tabel 3.21 Struktur tabel user_data

No.	Nama *Field*	Tipe	Lebar
1	id_user_data	Integer	10
2	nama_depan	Varchar	50
3	nama_belakang	Varchar	50
4	Alamat	Text	
5	kode_pos	Integer	10
6	Phone	Integer	10

Entitas *user* merepresentasikan tabel user, yang berisi data akun pembeli. Struktur tabel user ditunjukkan pada Tabel 3.22

Tabel 3.22 Struktur tabel user

No.	Nama *Field*	Tipe	Lebar
1	id_user	Integer	10
2	username	Varchar	50
3	email	Varchar	50
4	password	Varchar	50
5	status	Enum	(‘0’, ‘1’)
6	level	Integer	10
7	last_visit	Varchar	50
8	reset	Varchar	100
9	expire	Varchar	50

Entitas gambar merepresentasikan tabel gambar, yang berisi data gambar pilihan *password* unruk pembeli dan penjual. Struktur tabel gambar ditunjukkan pada Tabel 3.23

Tabel 3.23 Struktur tabel gambar

No.	Nama *Field*	Tipe	Lebar
1	id_gambar	Varchar	3
2	url	Varchar	200

Entitas session gambar merepresentasikan tabel session_gambar, yang berisi data susunan id gambar dan susunan karakter setiap halaman di*refresh*. Struktur tabel session_gambar ditunjukkan pada Tabel 3.24

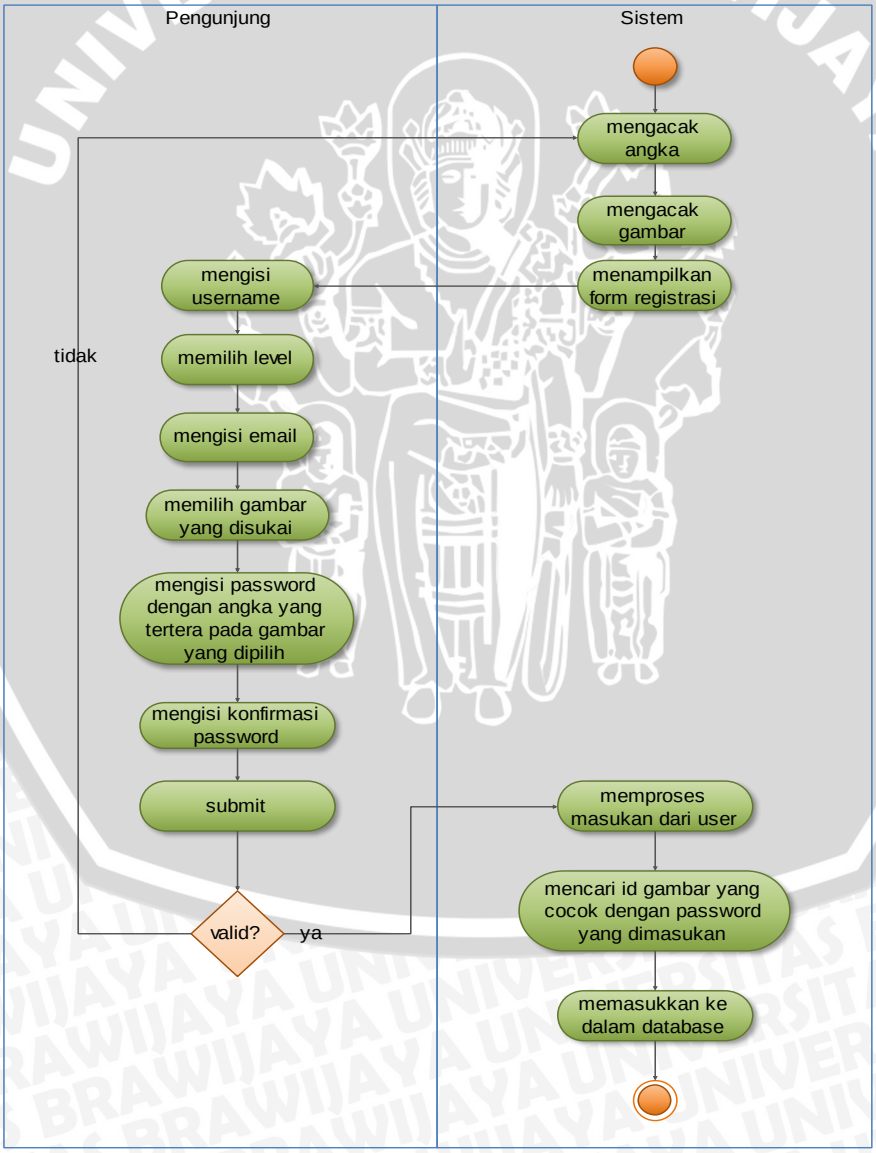
Tabel 3.24 Struktur tabel session _gambar

No.	Nama *Field*	Tipe	Lebar
1	session_id	Integer	3
2	merge	Varchar	250
3	Iterasi	Varchar	250

3.2.3.3 Perancangan Aktivitas Aplikasi

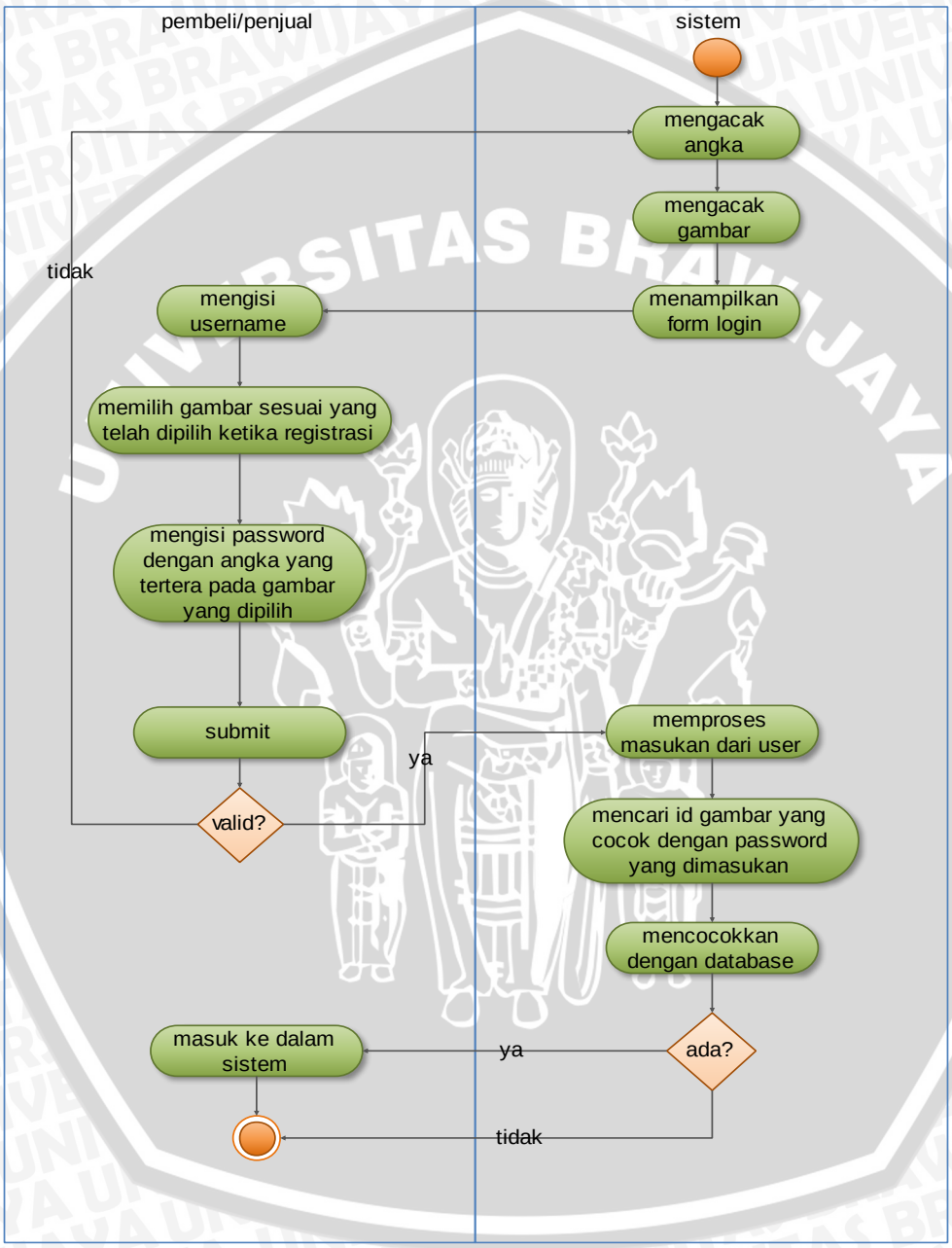
Diagram aktivitas menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, kemungkinan yang mungkin terjadi, dan bagaimana proses berakhir. Diagram aktivitas juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Gambar 3.6 merupakan diagram aktivitas *register*. Diagram aktivitas ini menggambarkan alur ketika pengunjung melakukan proses *registrasi*.



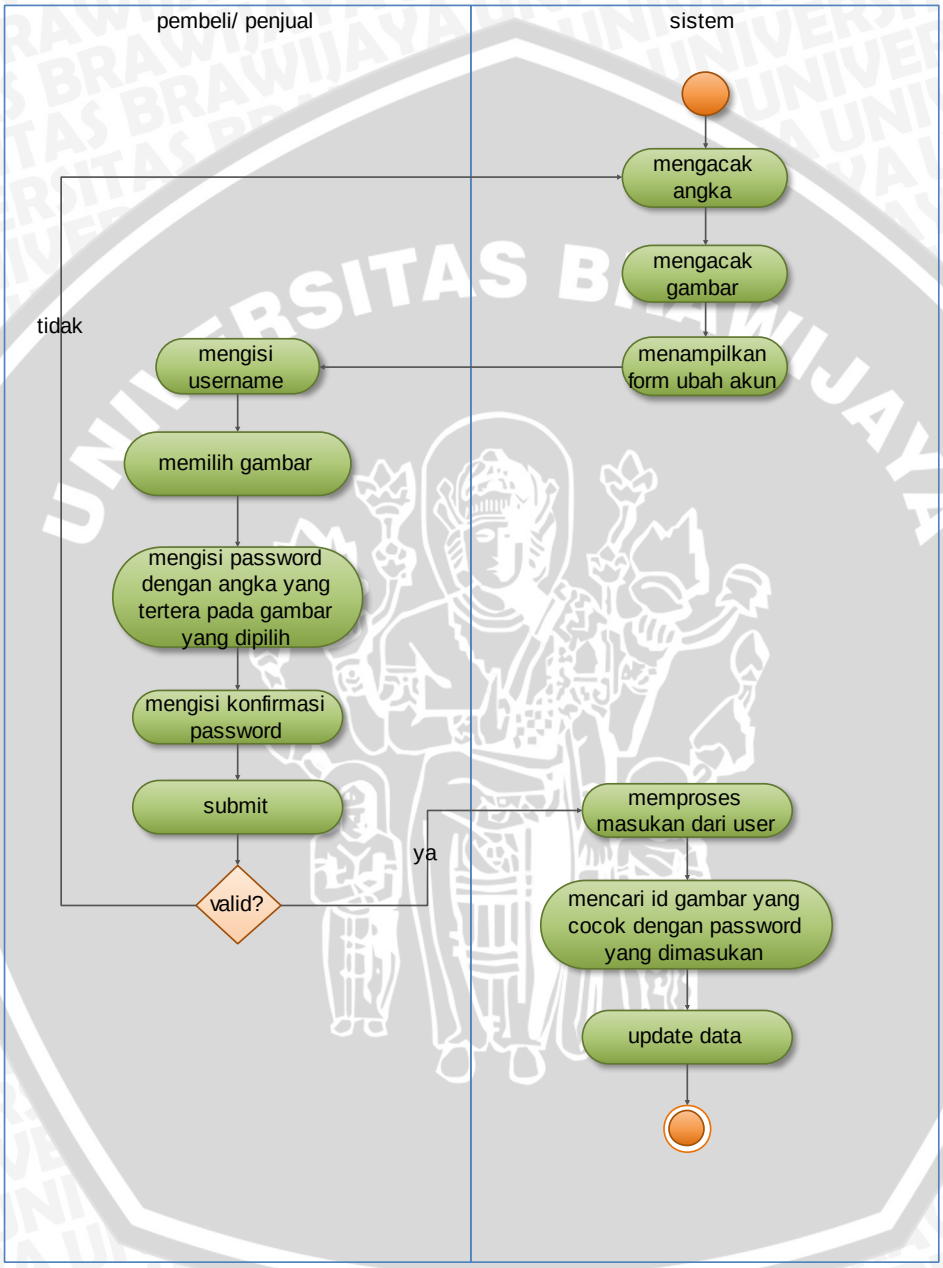
Gambar 3.6 Diagram Aktivitas *Register*

Gambar 3.7 merupakan diagram aktivitas *login*. Diagram aktivitas ini menggambarkan alur ketika pengguna yang telah terdaftar baik pembeli maupun penjual melakukan proses *login*.



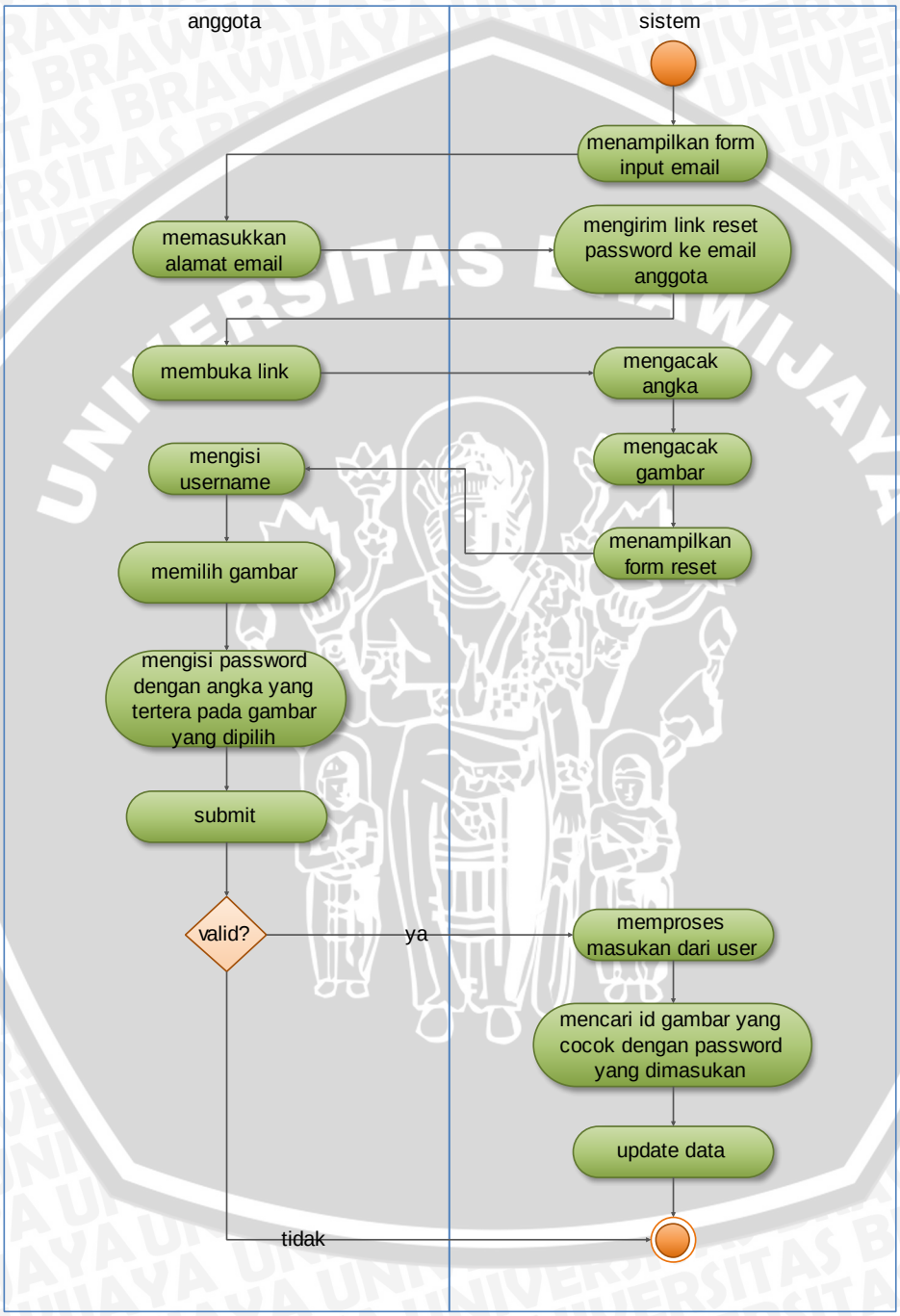
Gambar 3.7 Diagram Aktivitas *Login*

Gambar 3.8 merupakan diagram aktivitas mengubah akun. Diagram aktivitas ini menggambarkan alur ketika pembeli atau penjual melakukan proses ubah akun personalnya.



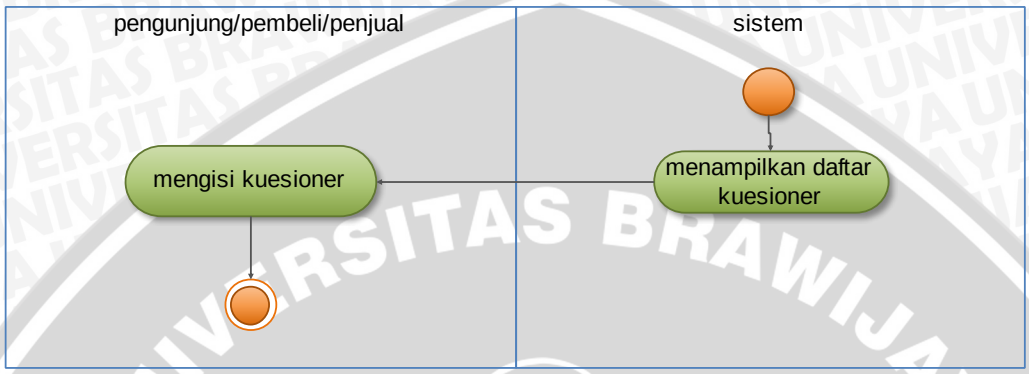
Gambar 3.8 Diagram Aktivitas Ubah Akun

Gambar 3.9 merupakan diagram aktivitas *reset password*. Diagram aktivitas ini menggambarkan alur ketika pembeli atau penjual melakukan proses *reset password* karena lupa pada *password*nya.



Gambar 3.9 Diagram Aktivitas Reset *Password*

Gambar 3.10 merupakan diagram aktivitas mengisi *feedback* untuk pengguna yang bertindak sebagai pengunjung, pembeli atau penjual. Diagram aktivitas ini menggambarkan alur ketika pengunjung, pembeli atau penjual melakukan proses mengisi *feedback*.

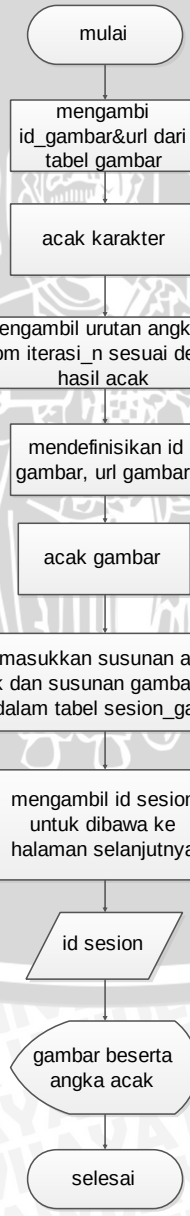


Gambar 3.10 Diagram Aktivitas Mengisi *Feedback*

3.2.3.4 Perancangan Algoritma Aplikasi

Perancangan algortima pada aplikasi dimodelkan dalam bentuk *flowchart*. Adapun algoritma yang dimodelkan adalah beberapa algoritma yang merupakan algoritma utama pada aplikasi, yaitu algoritma proses menampilkan gambar beserta karakter yang acak, proses *register*, proses *login*, proses *edit* akun dan proses *reset password*.

Flowchart proses menampilkan gambar acak beserta karakter acak dapat dilihat pada Gambar 3.11. Proses ini digunakan sebagai fitur memilih *password* untuk pembeli dan penjual.

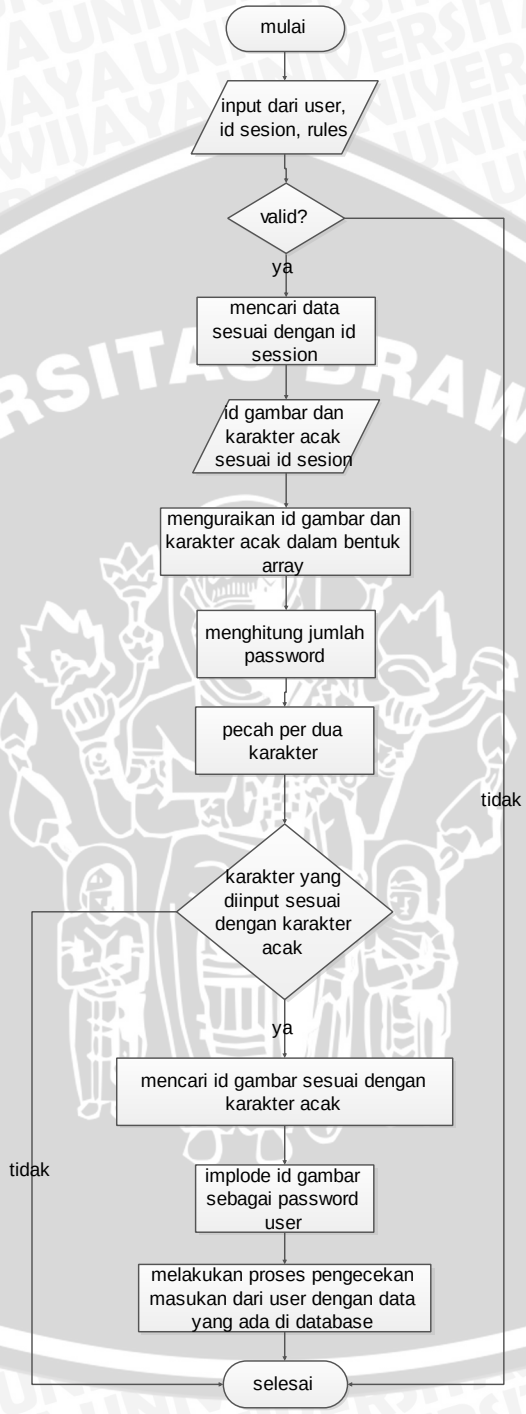


Gambar 3.11 *Flowchart* Proses Menampilkan Gambar Acak Beserta Karakter Acak

Flowchart proses *register* untuk pembeli dan penjual dapat dilihat pada Gambar 3.12

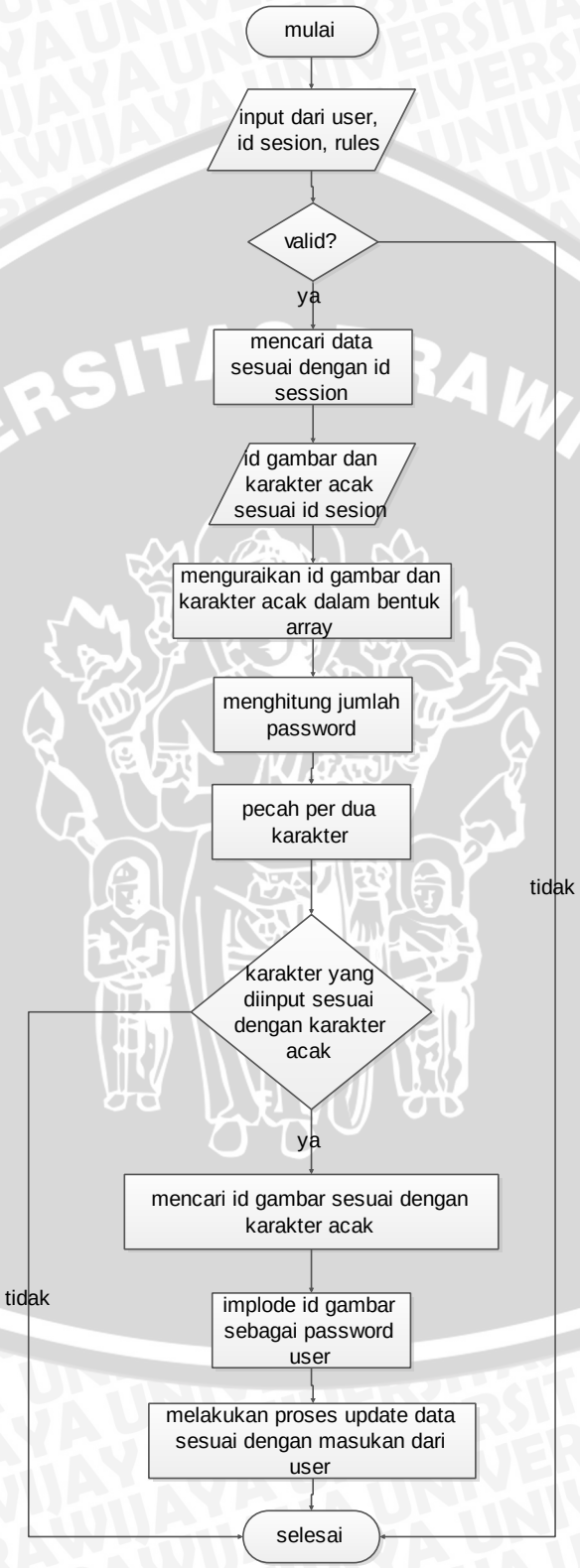
Gambar 3.12 *Flowchart* Proses Algoritma Register

Flowchart proses *login* untuk pembeli dan penjual dapat dilihat pada Gambar 3.13



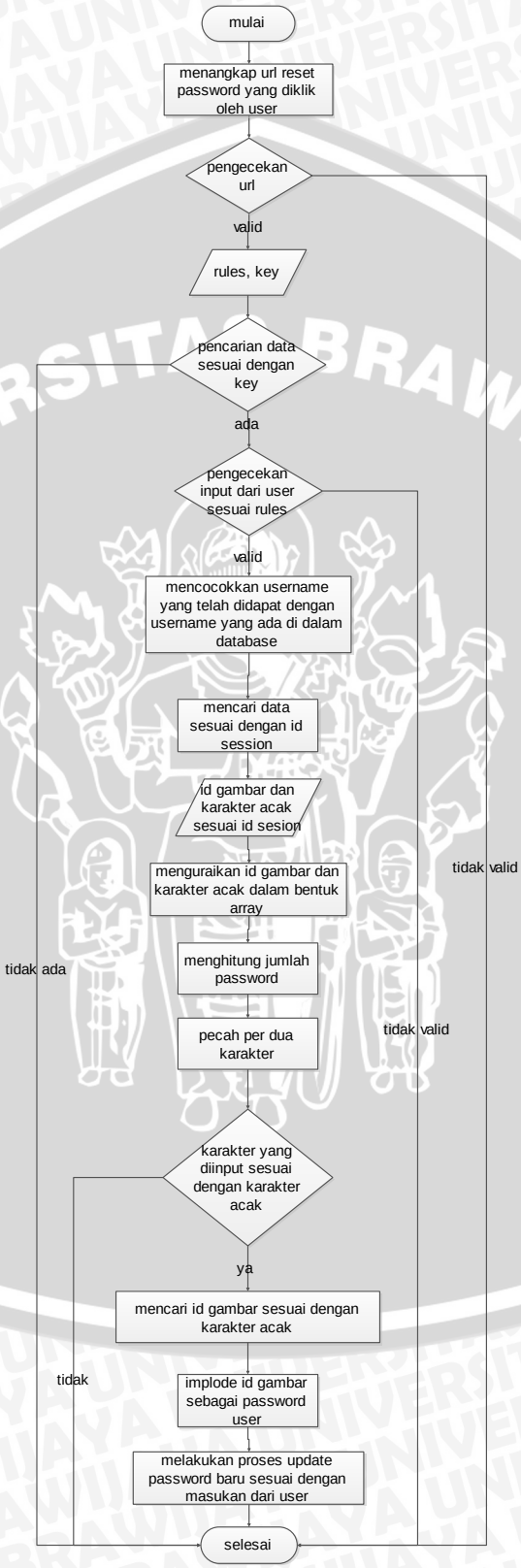
Gambar 3.13 *Flowchart* Proses Algoritma *Login*

Flowchart proses *edit* akun untuk pembeli dan penjual dapat dilihat pada Gambar 3.14



Gambar 3.14 *Flowchart* Proses Algoritma *Edit* Akun

Flowchart proses *reset password* untuk pembeli dan penjual yang lupa terhadap *password*nya dapat dilihat pada Gambar 3.15



Gambar 3.15 *Flowchart* Proses Algoritma *Reset Password*

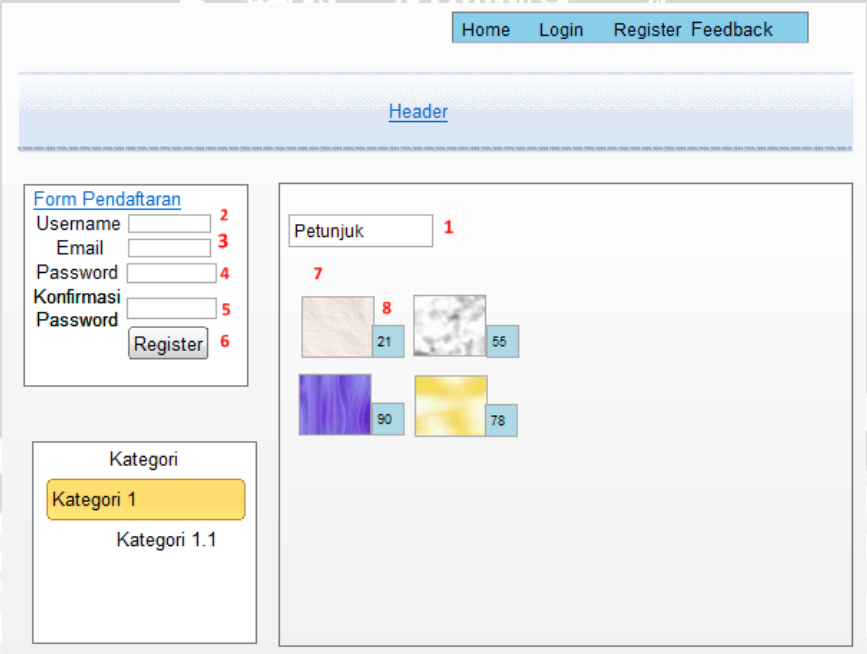
3.2.3.5 Perancangan Antarmuka Aplikasi

Pada bagian ini akan dijelaskan tentang perancangan antarmuka *website ecommerce* PTIIK yang menggunakan metode OTP berbasis gambar. Antarmuka aplikasi ini akan digunakan oleh pengguna untuk berinteraksi dengan sistem. Antarmuka aplikasi ini dibagi menjadi empat, yaitu antarmuka untuk halaman pengunjung, halaman pembeli, halaman penjual dan halaman *administrator*.

1. Perancangan Antarmuka Halaman Pengunjung

   a. Halaman *Register*

   Halaman *register* merupakan salah satu antarmuka pengguna untuk halaman pengunjung. Halaman *register* berfungsi bagi pengunjung untuk mendaftarkan diri sebagai pembeli atau penjual. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhuan SRS_001_01. Gambar 3.16 akan menunjukkan perancangan tampilan antarmuka dari halaman *register*.



Gambar 3.16 Tampilan Antarmuka Halaman *Register*

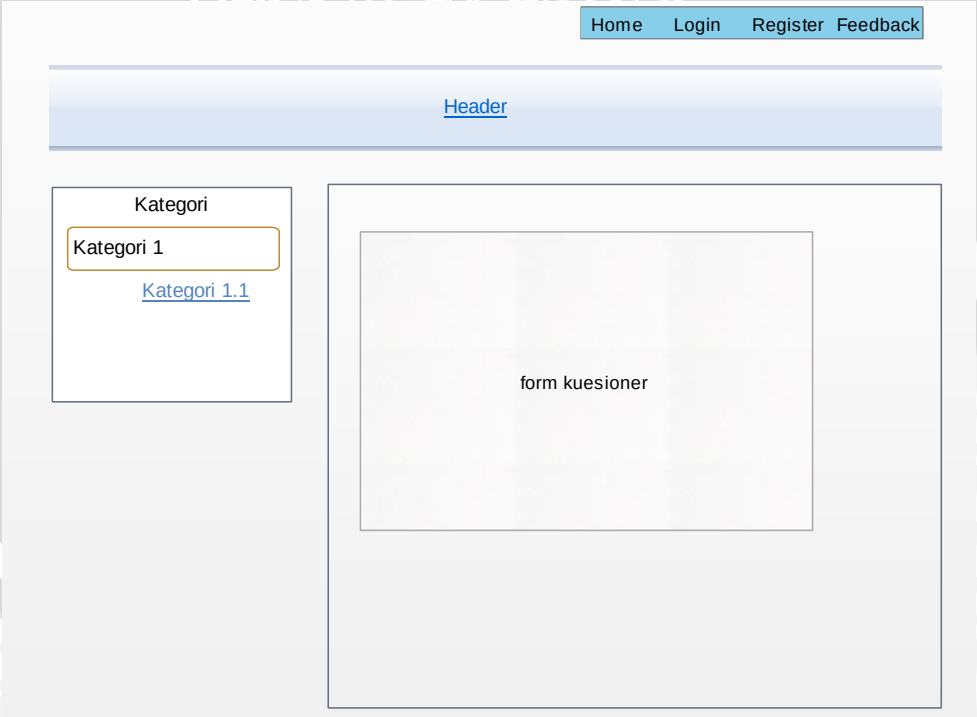
Gambar 3.16 memiliki keterangan sebagai berikut :

1. Petunjuk sebagai acuan mengisi *form* pendaftaran
2. *Field Username* untuk mengisi nama pengguna.

3. *Field Email* untuk mengisi alamat *email* pengguna
4. *Field Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password*.
5. *Field* Konfirmasi *Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password*.
6. Tombol Register untuk menjalankan proses pendaftaran.
7. Gambar pilihan untuk *password* pengguna
8. Karakter yang mewakili gambar *password*

b. Halaman *Feedback*

Halaman *feedback* merupakan salah satu antarmuka pengguna untuk halaman pengunjung. Halaman *feedback* berfungsi bagi pengunjung untuk mengisi kuesioner. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhuan SRS_001_02. Gambar 3.17 akan menunjukkan perancangan tampilan antarmuka dari halaman *feedback*.

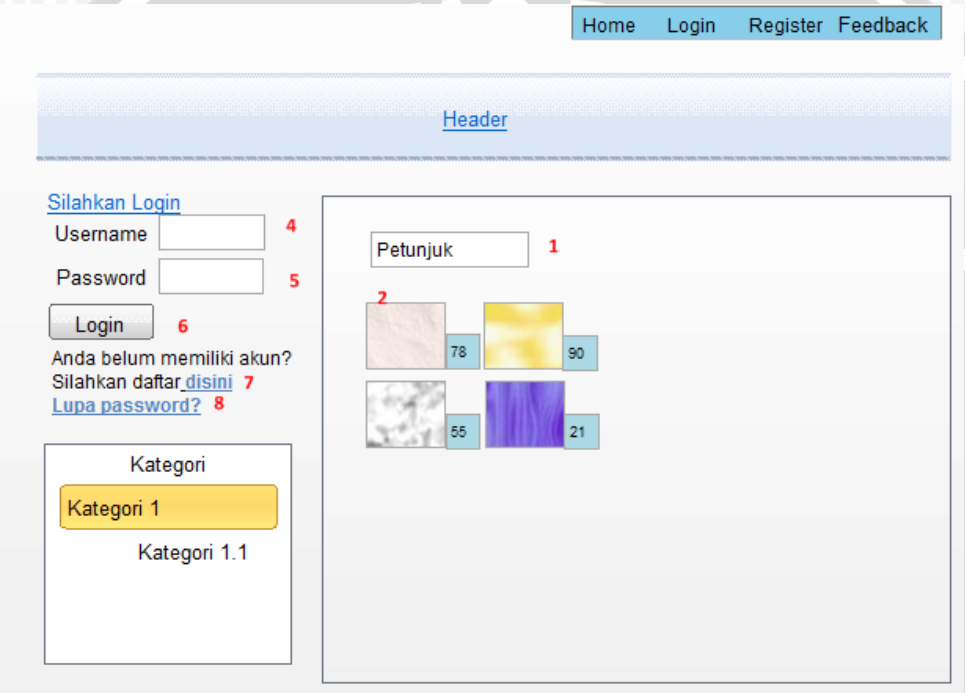


Gambar 3.17 Tampilan Antarmuka Halaman *Feedback*

2. Perancangan Antarmuka Halaman Pembeli

a. Halaman *Login*

Halaman *login* merupakan salah satu antarmuka pengguna untuk halaman pembeli. Halaman *login* berfungsi bagi pembeli atau penjual untuk masuk ke dalam aplikasi. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhuan SRS_002_01. Gambar 3.18 akan menunjukkan perancangan tampilan antarmuka dari halaman *login*.



Gambar 3.18 Tampilan Antarmuka Halaman *Login*

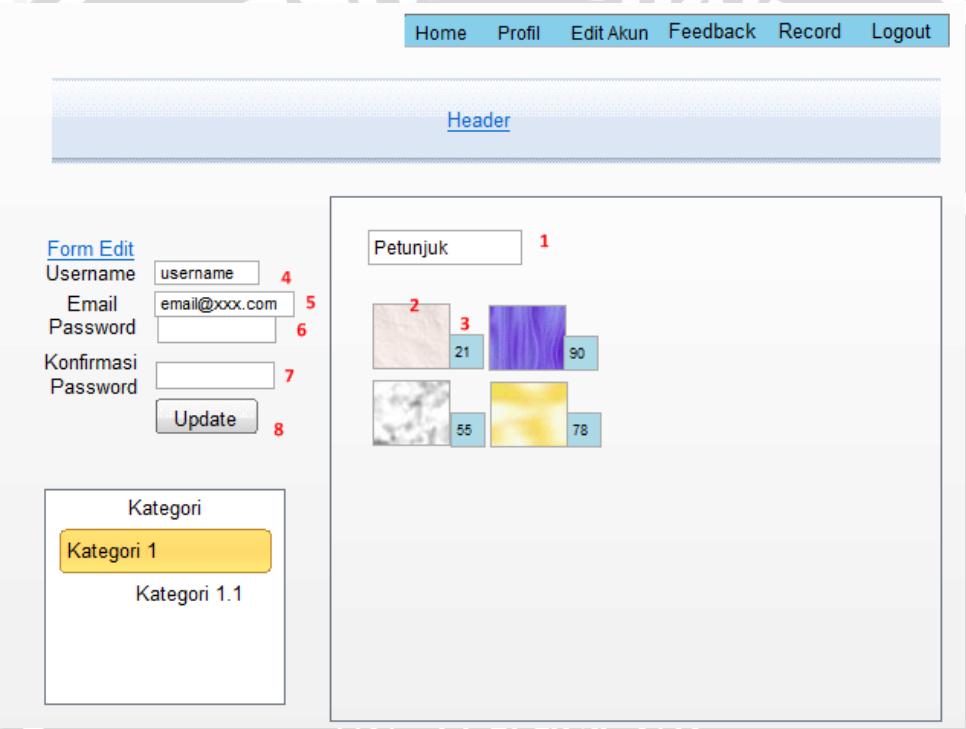
Gambar 3.18 memiliki keterangan sebagai berikut :

1. Bantuan sebagai acuan mengisi *form login*
2. Gambar pilihan untuk *password* pengguna
3. Karakter yang mewakili gambar *password*
4. *Field Username* untuk mengisi nama pengguna.
5. *Field Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password* saat proses pendaftaran
6. Tombol Login untuk menjalankan proses masuk ke dalam aplikasi.
7. *Link* untuk melakukan proses pendaftaran

8. *Link* untuk melakukan *reset password* apabila pembeli lupa akan *password*nya

b. Halaman *edit* akun

Halaman *edit* akun merupakan salah satu antarmuka pengguna untuk halaman pembeli. Halaman *edit* akun berfungsi bagi pembeli untuk mengubah *username* atau *passwordnya*. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhan SRS_002_02. Gambar 3.19 akan menunjukkan perancangan tampilan antarmuka dari halaman *edit* akun.



Gambar 3.19 Tampilan Antarmuka Halaman Edit Akun

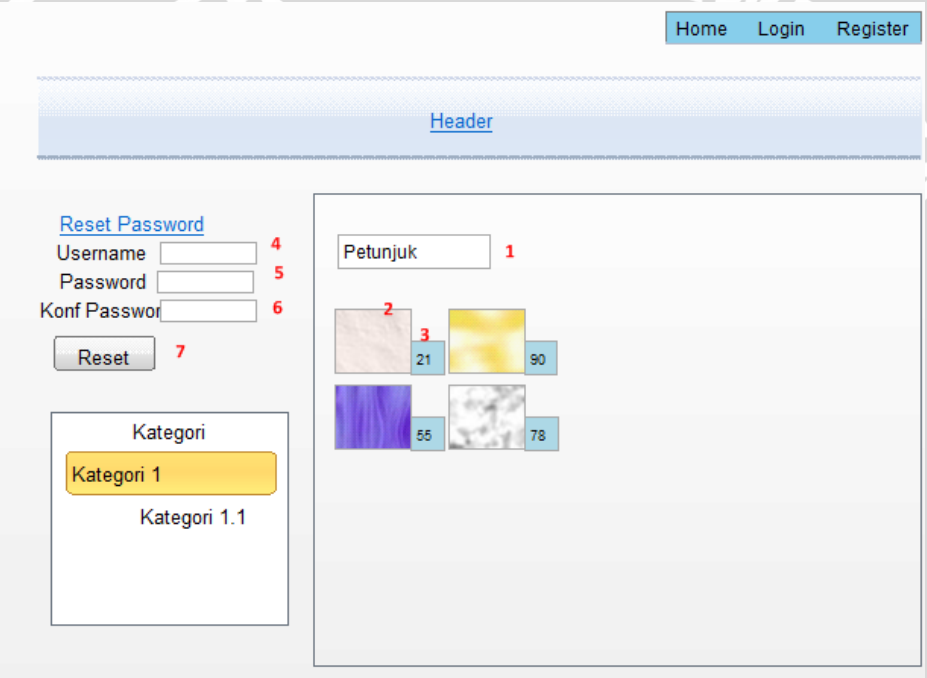
Gambar 3.19 memiliki keterangan sebagai berikut :

1. Bantuan sebagai acuan mengisi *form edit* akun
2. Gambar pilihan untuk *password* pengguna
3. Karakter yang mewakili gambar *password*
4. *Field Username* untuk mengubah nama pengguna.
5. *Field email* tidak dapat diubah
6. *Field Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password*

7. *Field* Konfirmasi *Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password*
8. Tombol Update untuk menjalankan proses ubah data yang baru.

c. Halaman *Reset Password*

Halaman *reset password* merupakan salah satu antarmuka pengguna untuk halaman pembeli. Halaman *reset password* berfungsi untuk menyetel ulang *password*nya karena pembeli lupa akan *password*nya. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhan SRS_002_03. Gambar 3.20 akan menunjukkan perancangan tampilan antarmuka dari halaman *reset password*



Gambar 3.20 Tampilan Antarmuka Halaman *Reset Password*

Gambar 3.20 memiliki keterangan sebagai berikut :

1. Bantuan sebagai acuan mengisi *form reset password*
2. Gambar pilihan untuk *password* pengguna
3. Karakter yang mewakili gambar *password*
4. *Field Username* yang berisi nama pengguna.
5. *Field Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password*
6. Tombol Reset untuk menjalankan proses menyetel ulang *password.*

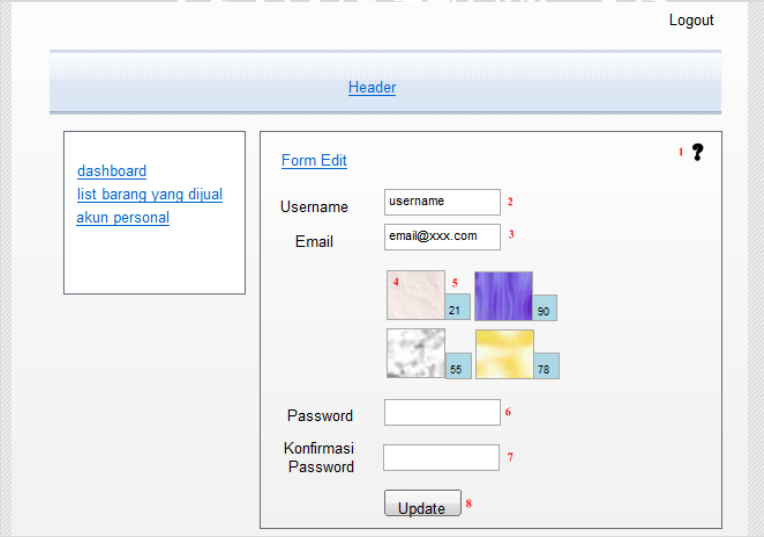
3. Perancangan Antarmuka Halaman Penjual

a. Halaman Login

Halaman *login* pada pengguna penjual digunakan untuk masuk ke dalam halaman khusus penjual. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhuan SRS_003_01. Perancangan antarmuka *login* penjual sama dengan perancangan antar muka *login* pada pembeli (lihat Gambar 3.18)

b. Halaman Edit Akun

Halaman *edit* akun merupakan salah satu antarmuka pengguna untuk halaman penjual. Halaman *edit* akun berfungsi bagi penjual untuk mengubah *username* atau *password*nya. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhan SRS_003_02. Gambar 3.21 akan menunjukkan perancangan tampilan antarmuka dari halaman *edit* akun.



Gambar 3.21 Tampilan Antarmuka Halaman *Edit* Akun

Gambar 3.21 memiliki keterangan sebagai berikut :

1. Tombol bantuan sebagai acuan mengisi *form edit* akun
2. *Field Username* untuk mengubah nama pengguna.
3. Gambar pilihan untuk *password* pengguna
4. Karakter yang mewakili gambar *password*
5. *Field Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password*

6. *Field* Konfirmasi *Password* untuk mengisi karakter sesuai dengan gambar yang dipilih sebagai *password*
7. Tombol Update untuk menjalankan proses ubah data yang baru.

c. Halaman Reset Password

Halaman *reset password* pada penjual digunakan untuk menyetel ulang *password* penjual karena penjual tersebut lupa pada *password*nya. Perancangan antarmuka ini mengacu pada spesifikasi kebutuhuan SRS_003_03. Perancangan antarmuka *reset password* penjual sama dengan perancangan antar muka *reset password* pada pembeli (lihat Gambar 3.20)

BAB IV
IMPLEMENTASI

Bab ini membahas tahapan implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak. Pembahasan terdiri atas penjelasan tentang spesifikasi sistem, batasan implementasi, implementasi basis data, implementasi tiap *class* pada *file* program, implementasi algoritma dan implementasi antarmuka.

4.1 Spesifikasi Sistem

Perangkat lunak dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

4.1.1. Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan dijelaskan pada Tabel 4.1.

Tabel 4.1 Spesifikasi perangkat keras komputer

Note Asus K42Jc	
Processor	Intel ® Core ™ i3 CPU M 370 2.40 GHz
Memory (RAM)	2.00 GB
Harddisk	Seagate Momentus 5400.6 SATA 3Gb/s 500 GB
Motherboard	IntelCore Motherboard
Monitor	14" Widescreen LED Backlit Display

4.1.2. Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang dipakai dalam proses pengembangan implementasi OTP berbasis gambar pada *website Ecommerce* PTIIK dijelaskan pada Tabel 4.2.

Tabel 4.2 Spesifikasi perangkat lunak komputer

Note Asus K42Jc	
Operating System	Microsoft Windows 7 32-bit
Programming Language	PHP
Framework	Code Igniter
Database Management System	MySQL 5.1
Software Development Tools	Adobe Dreamwaver, XAMPP

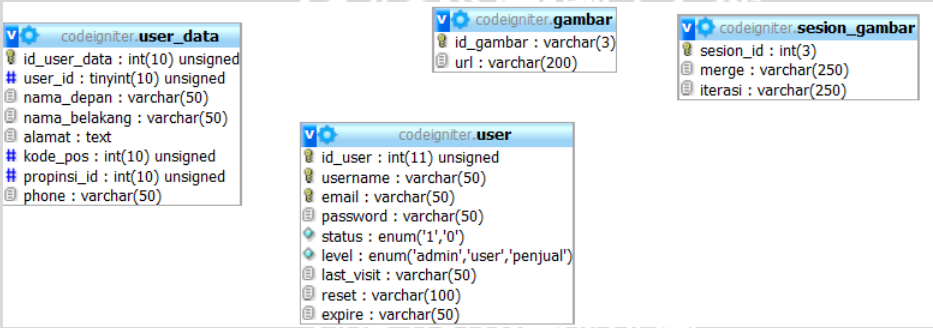
4.2 Batasan – Batasan Implementasi

Beberapa batasan dalam mengimplementasikan perangkat lunak adalah:

1. Implementasi OTP berbasis gambar pada website Ecommerce PTIIK dirancang dan dijalankan dengan menggunakan *Web Application.*
2. *Database Management System* yang digunakan adalah MySQL.
3. Implementasi difokuskan pada proses *register, login edit akun* dan *reset password* yang menggunakan metode OTP berbasis gambar

4.3 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan *database management system* MySQL. Hasil implementasi penyimpanan data ini berupa *script – script* SQL. Hasil implementasi SQL pada *database* ini dimodelkan dalam diagram konseptual *entity relationship.* Gambar 4.1 menggambarkan diagram konseptual *entitiy relationship* dari Implementasi OTP berbasis gambar pada *website ECommerce* PTIIK.



Gambar 4.1 Diagram ER konseptual dari Implementasi OTP berbasis gambar pada website Ecommerce PTIIK

4.4 Implementasi *Class* dan *Interface* Pada *File* Program

Setiap *class* yang telah dirancang pada proses perancangan direalisasikan pada sebuah *file* program dengan ekstensi *.php. Tabel 4.3 menjelaskan mengenai pasangan antara *class* dengan *file* program yang digunakan untuk mengimplementasikannya

Tabel 4.3 Implementasi *class* pada kode program

No.	*Package*	Nama *Class* atau *Interface*	Nama *File* Program
1	Controllers	User	User.php
2	Controllers/Juragan	User	User.php
3	Controllers/Sell	User	User.php

4	Models	User_m	User_m.php
5	Views	About	About.php
6	Views	Edit	Edit_mod.php
7	Views	Login	Login.php
8	Views	Lupa	Lupa.php
9	Views	Register	Register_mod.php
10	Views	Reset	Reset.php
11	Views	Reset_form	Reset_form_mod.php
12	Views	Sidebar_view	Sidebar_view.php
13	Views	User_login	User_login_mod.php
14	Views/penjual/user	List	List.php

4.5 Implementasi algoritma

Implementasi OTP berbasis gambar pada website Ecommerce PTIIK mempunyai beberapa proses (*method*) utama yang terbagi dalam beberapa *class*. Pada penulisan skripsi ini hanya dicantumkan algoritma dari beberapa proses saja sehingga tidak semua implementasi algoritma *method* akan dicantumkan. Implementasi algoritma proses yang dicantumkan antara lain adalah proses menampilkan gambar acak dan karakter acak, proses *register* pembeli dan penjual, proses *login* untuk pembeli dan penjual, proses *edit* akun untuk pembeli dan penjual, dan proses *reset password* untuk pembeli dan penjual. Implementasi algoritma ini akan direpresentasikan dalam bentuk *code* dengan bahasa pemrograman PHP berframework *CodeIgniter*.

4.5.1 Implementasi Algoritma Proses Menampilkan Gambar Acak dan Karakter Acak

Operasi pada proses menampilkan gambar acak dan karakter acak pada *field password* bertujuan untuk memberikan pilihan *password* berbasis OTP berbasis gambar pada pada proses autentikasi. Susunan gambar dan karakter selalu berubah-ubah ketika halaman di*refresh*. Proses ini digunakan pada fitur *register*, *login*, *edit* akun dan *reset password*. Gambar 4.2 merupakan salah satu penggunan algoritma ini pada proses *register*. *Method* `register()` merupakan implementasi proses menampilkan gambar acak dan karakter acak pada *field password*

```
1.  public function register(){
2.    $this->form_validation->set_rules($this->rules);
3.    $query=$this->user_m->view_gambar();
4.    $i= rand(1,3);
5.    $query_index=$this->user_m->view_index($i);
6.    $iterasi = "iterasi_".$i;
7.    foreach($query as $row){
8.        $iters[$row->url] = $row->id_gambar;
9.        $url[$row->id_gambar] = $row->url;
10.       }
11.   foreach($query_index as $row){
12.       $iter[$row->id_urutan] = $row->$iterasi;
13.       $iterses[]=$row->$iterasi;
14.       }
15.   $konstanta=array_flip($iter);
16.   shuffle($iters);
17.   $kaka = array_combine($konstanta,$iters);
18.   $coba=implode($kaka);
19.   $iterses_=implode($iterses);
20.   $query4=$this->user_m-
21.   >add_sesion_gambar($coba,$iterses_);
22.   $query5=$this->user_m->get_sesion_id($coba);
23.   foreach($query5 as $row){
24.       $_session=$row->sesion_id;
25.       }
26.   foreach($kaka as $row=>$value){
27.       $query3=$this->user_m->cek_url($value);
28.       foreach($query3 as $row3){
29.            $tampil[$row]=$row3->url;
30.            }}
31.   $this->data->_session=$_session;
32.   $this->data->iter = $iter;
33.   $this->data->tampil = $tampil;
34.   $this->data->register_page = true;
35.   $this->data->username = set_value('username');
36.   $this->data->email = set_value('email');
37.   $this->template->set_judul('My E-Commerce')
38.   ->set_css('style')
39.     ->set_parsial('sidebar','sidebar_view',$this->data)
40.     ->set_parsial('topmenu','top_view',$this->data)
41.     ->render('register_mod',$this->data);
42.       }
```

Gambar 4.2 ImplementasiAlgoritma Proses Menampilkan Gambar Acak dan Karakter Acak

Penjelasan implementasi proses menampilkan gambar acak dan karakter acak pada Gambar 4.2 yaitu:

1. Baris 3: mengambil semua gambar dari tabel gambar
2. Baris 4-6: mengacak karakter
3. Baris 7-15: mendefinisikan id gambar dan url gambar
4. Baris 16: mengacak id gambar
5. Baris 17-21: menggabungkan karakter acak dan gambar kemudian di masukkan ke dalam tabel sesion_gambar
6. Baris 22-25: mengambil session_id
7. Baris 26-34: membawa semua atribut yang diperlukan ke halaman view untuk ditampilkan gambar beserta karakter yang acak

4.5.2 Implementasi Algoritma Proses *Register* Pembeli dan Penjual

Operasi pada Proses *Register* bertujuan untuk menyediakan form pendaftaran untuk pembeli atau penjual.Form ini berisi username dan email yang harus diisi oleh pengguna dengan data yang valid. Pengguna harus memilih gambar yang telah disediakan minimal 3 gambar, sebagai password yang harus diingat dalam aplikasi ini. Gambar 4.3 merupakan *method* `reg()` sebagai implementasi proses *Register* untuk pembeli dan penjual.

```
1.  $this->form_validation->set_rules($this->rules);
2.    if($this->form_validation->run()) {
3.        $username = $this->input->post('username');
4.        $email = $this->input->post('email');
5.        $password = $this->input->post('password');
6.        $status = $this->input->post('status');
7.        $iter = $this->input->post('iter');
8.        $coba= $this->input->post('_session');
9.        $query=$this->user_m->cari_merge($coba);
10.       foreach($query as $row){
11.         $_array=$row->merge;
12.         $_iterses=$row->iterasi;
13.       $jumlah = strlen($_array);
14.       for($i=0;$i<$jumlah;$i+=2){
15.         $array_id[]= substr ($_array, $i, 2);
16.         $iterses[]= substr ($_iterses, $i, 2);
17.                                   }
18.        $num = strlen($password);
19.       for($i=0;$i<$num;$i+=2){
```

```
20.          $pec[]= substr ($password, $i, 2);
21.          }
22.        foreach($pec as $row => $value){
23.         if (!in_array($value, $iterses)){
24.             $gagal = true;
25.             break;
26.             }
27.          else {
28.             $gagal=false;
29.             }}
30.        if($gagal) {
31.           $this->data->error = '<div class="errora">Angka
32.           yang anda masukkan tidak tercantum, silahkan
33.           ulangi lagi</div>';
34.           $this->data->register_page = true;
35.           }
36.        else{
37.           $flip=array_flip($array_id);
38.           $data=array_combine($flip, $iterses);
39.        foreach($pec as $row => $value){
40.           $kons[]=array_search($value,$data);
41.           }
42.        foreach($kons as $row => $value){
43.             $id[]=array_search($value,$flip);
44.             }
45.        $password= implode('|', $id );
46.        if(!$this->autentifikasi-
47.        >tambah($username,$email,$password,$status)) {
48.          $this->data->error                =            '<div
49.          class="errora">Username     telah     digunakan,
50.          silahkan mencoba register kembali</div>';
51.          $this->data->register_page = true;
52.             }
53.        else {
54.          $this->data->sukses= '<div class="sukses">Proses
55.          registrasi berhasil</div>';
56.          $this->data->register_page = false;
57.        }
58.             }
59.                  }
60.    else {
61.        $this->data->error                =            '<div
62.        class="errora">Register  gagal,  silahkan  mencoba
63.        register kembali</div>';
64.        $this->data->register_page = true;
```

```
65.       }
66.    $query=$this->user_m->view_gambar();
67.    $i= rand(1,3);
68.    $query_index=$this->user_m->view_index($i);
69.    $iterasi = "iterasi_".$i;
70.    foreach($query as $row){
71.       $iters[$row->url] = $row->id_gambar;
72.       $url[$row->id_gambar] = $row->url;
73.       }
74.    foreach($query_index as $row){
75.       $iter[$row->id_urutan] = $row->$iterasi;
76.       $itersesi[]=$row->$iterasi;
77.       }
78.    $konstanta=array_flip($iter);
79.    shuffle($iters);
80.    $kaka = array_combine($konstanta,$iters);
81.    $coba=implode($kaka);
82.    $iterses_=implode($itersesi);
83.    $query4=$this->user_m-
84.    >add_sesion_gambar($coba,$iterses_);
85.    $query5=$this->user_m->get_sesion_id($coba);
86.    foreach($query5 as $row){
87.      $_session=$row->sesion_id;
88.      }
89.    foreach($kaka as $row=>$value){
90.       $query3=$this->user_m->cek_url($value);
91.       foreach($query3 as $row3){
92.            $tampil[$row]=$row3->url;
93.            }
94.       }
95.    $this->data->_session=$_session;
96.    $this->data->iter = $iter;
97.    $this->data->tampil = $tampil;
98.    $this->template->set_judul('PTIIKshop')
99.     ->set_parsial('sidebar','sidebar_view',$this->data)
100    ->set_parsial('topmenu','top_view',$this->data)
101    ->render('register_mod',$this->data);
102
```

Gambar 4.3 Implementasi Algoritma Proses *Register* Pembeli dan Penjual

Penjelasan implementasi algoritma proses *register* pada Gambar 4.3 yaitu:

1. Baris 2: menginisialisasi aturan untuk pengisian *form register*
2. Baris 3-8: menginisialisasi semua data yang dibawa dari halaman view

3. Baris 9: mencari data susunan gambar dan karakter sesuai dengan id sesion
4. Baris 10-17: proses untuk menguraikan data sesuai dengan id sesion
5. Baris 18: menghitung panjang karakter *password* yang dimasukkan pengguna
6. Baris 19-21: memisah *password* yang dimasukkan per dua karakter
7. Baris 22-29: mencocokkan password yang telah dipisah per dua karakter di data tabel session_gambar sesuai dengan id_sesion
8. Baris 31-34: mengeluarkan pesan "Angka yang anda masukkan tidak tercantum, silahkan ulangi lagi" dengan kondisi apabila karakter tidak ditemukan
9. Baris 37-44: proses mencari id gambar dengan mencocokan data sesuai dengan id_sesion
10. Baris 45: implode id gambar sebagai *password* pengguna
11. Baris 46-47: melakukan proses *insert* masukan dari pengguna ke dalam basis data
12. Baris 48-51: kondisi jika gagal melakukan proses *insert*
13. Baris 54-56: kondisi jika berhasil melakukan proses *insert*
14. Baris 61-64: kondisi jika masukan pengguna tidak sesuai dengan aturan pengisian *form register*
15. Baris 65-102: menampilkan halaman berikutnya

4.5.3 Implementasi Algoritma Proses *Login* Pembeli dan Penjual

Operasi pada proses *login* bertujuan untuk memberi seleksi kepada pengguna. Pengguna yang terdaftar sebagai pembeli hanya dapat mengakses halaman pembeli dan begitu juga sebaliknya, pengguna yang terdaftar sebagai penjual hanya dapat mengakses halaman aplikasi penjual saja. Pengguna dapat mengisi *field username* dan mengisi karakter pada gambar yang telah disediakan sesuai dengan gambar yang diinputkan saat *register*, sebagai *password*. Gambar 4.4 merupakan *method* `proses_login()` sebagai implementasi dari proses *Login* pembeli dan penjual.

```
$this->form_validation->set_rules($this->login_rules);
if($this->form_validation->run()) {
   $username = $this->input->post('username');
   $password = $this->input->post('password');
   $coba= $this->input->post('_session');
   $query=$this->user_m->cari_merge($coba);
   foreach($query as $row){
      $_array=$row->merge;
      $_iterses=$row->iterasi;
      }
   $jumlah = strlen($_array);
   for($i=0;$i<$jumlah;$i+=2){
      $array_id[]= substr ($_array, $i, 2);
      $iterses[]= substr ($_iterses, $i, 2);
      }
   $num = strlen($password);
   for($i=0;$i<$num;$i+=2){
      $pec[]= substr ($password, $i, 2);
      }
   foreach($pec as $row => $value){
      if (!in_array($value, $iterses)){
         $gagal = true;
         break;
         }
      else {
         $gagal=false;
         }}
   if($gagal){
      $this->data->error = '<div class="errora">Angka
      yang anda masukkan tidak tercantum, silahkan
      ulangi lagi</div>';
      }
   else{
      $flip=array_flip($array_id);
      $data=array_combine($flip, $iterses);
      foreach($pec as $row => $value){
            $kons[]=array_search($value,$data);
            }
      foreach($kons as $row => $value){
            $id[]=array_search($value,$flip);
            }
      $password= implode('|', $id );
      if(!$this->autentifikasi-
      >login($username,$password)) {
            $this->data->error = '<div
```

```
46.                 class="errora">username or password is
47.                 incorrect</div>';
48.                 }
49.           else {
50.                 $this->autentifikasi-
51.           >login($username,$password);
52.                 if($this->session-
53.           >userdata('level')==='penjual')
54.                 redirect(site_url('penjual'));
55.                 else{
56.                 $this->session->set_flashdata('pesan', '<div
57.                 class="sukses">Anda telah berhasil login,
58.                 selamat berbelanja.</div>');
59.                 redirect(site_url('user/proses_login'));
60.                  }}}}
61.   $query=$this->user_m->view_gambar();
62.   $i= rand(1,3);
63.   $query_index=$this->user_m->view_index($i);
64.   $iterasi = "iterasi_".$i;
65.   foreach($query as $row){
66.     $iters[$row->url] = $row->id_gambar;
67.     $url[$row->id_gambar] = $row->url;
68.     }
69.   foreach($query_index as $row){
70.     $iter[$row->id_urutan] = $row->$iterasi;
71.     $itersesi[]=$row->$iterasi;
72.     }
73.   $konstanta=array_flip($iter);
74.   shuffle($iters);
75.   $kaka = array_combine($konstanta,$iters);
76.   $coba=implode($kaka);
77.   $iterses_=implode($itersesi);
78.   $query4=$this->user_m-
79.         >add_sesion_gambar($coba,$iterses_);
80.   $query5=$this->user_m->get_sesion_id($coba);
81.   foreach($query5 as $row){
82.     $_session=$row->sesion_id;
83.     }
84.   foreach($kaka as $row=>$value){
85.     $query3=$this->user_m->cek_url($value);
86.     foreach($query3 as $row3){
87.     $tampil[$row]=$row3->url;
88.     }
89.   }
90.   $this->data->_session=$_session;
```

```
91. $this->data->iter = $iter;
92. $this->data->tampil = $tampil;
93. $this->data->login_page = true;
94. $this->template->set_judul('PTIIKshop')
95.   ->set_parsial('sidebar','sidebar_view',$this->data)
96.   ->set_parsial('topmenu','top_view',$this->data)
97.   ->render('user_login_mod',$this->data);
```

Gambar 4.4 Implementasi Algoritma Proses Login Pembeli dan Penjual

Penjelasan implementasi algoritma proses login pembeli dan penjual pada Gambar 4.4 yaitu:

1. Baris 2: menginisialisasi aturan untuk pengisian *form login*
2. Baris 3-5: menginisialisasi data yang dibawa dari halaman sebelumnya
3. Baris 6: mencari data susunan gambar dan karakter sesuai dengan id sesion
4. Baris 7-15: proses untuk menguraikan data sesuai dengan id sesion.
5. Baris 16: menghitung panjang karakter *password* yang dimasukkan pengguna
6. Baris 17-19: memisah *password* yang dimasukkan per dua karakter
7. Baris 20-27: mencocokkan password yang telah dipisah per dua karakter di data tabel session_gambar sesuai dengan id_sesion
8. Baris 29-31: mengeluarkan pesan "Angka yang anda masukkan tidak tercantum, silahkan ulangi lagi" dengan kondisi apabila karakter tidak ditemukan
9. Baris 34-41: proses mencari id gambar dengan mencocokan data sesuai dengan id_sesion
10. Baris 42: implode id gambar sebagai *password* pengguna
11. Baris 43: melakukan proses cek masukan dari pengguna dengan data yang terdapat dalam basis data
12. Baris 45-47: kondisi jika gagal melakukan proses pengecekan data
13. Baris 50: memanggil library autentifikasi pada fungsi login untuk memasukkan ke session
14. Baris 52-59: seleksi kondisi untuk memisahkan halaman antara penjual atau pembeli
15. Baris 61-96: menampilkan halaman selanjutnya

4.5.4 Implementasi Algoritma Proses Edit Akun Untuk Pembeli dan Penjual

Proses *Edit* Akun bertujuan untuk melakukan perubahan data username atau password dari pengguna.Pengguna disediakan field *username* dan gambar-gambar sebagai *password* untuk verifikasi pengubahan *username*.Apabila pengguna ingin mengganti *password*, maka pengguna harus memilih minimal 3 gambar baru untuk disimpan menjadi *password* barunya.Gambar 4.5 merupakan *method* `proses_edit()` sebagai implementasi dari proses edit akun.

```
1.  public function proses_edit(){
2.  $id = $this->session->userdata('user_id');
3.  $level = $this->session->userdata('level');
4.  $data = $this->user_m->get($id);
5.  if($data){
6.     $this->data->username = $data->username;
7.     $this->data->email = $data->email;
8.     }
9.  else{
10.    $this->data->username = set_value('username');
11.    $this->data->email = set_value('email');
12.    }
13. $this->form_validation->set_rules($this->edit_rules);
14. if($this->form_validation->run()) {
15.    $username = $this->input->post('username');
16.    $email = $this->input->post('email');
17.    $password = $this->input->post('new_password');
18.    $coba= $this->input->post('_session');
19.    $query=$this->user_m->cari_merge($coba);
20.    foreach($query as $row){
21.       $_array=$row->merge;
22.       $_iterses=$row->iterasi;
23.       }
24.    $jumlah = strlen($_array);
25.    for($i=0;$i<$jumlah;$i+=2){
26.       $array_id[]= substr ($_array, $i, 2);
27.       $iterses[]= substr ($_iterses, $i, 2);
28.       }
29.    $num = strlen($password);
30.       for($i=0;$i<$num;$i+=2){
31.       $pec[]= substr ($password, $i, 2);
32.       }
33.    foreach($pec as $row => $value){
34.       if (!in_array($value, $iterses)){
35.             $gagal = true;
```

```
36.             break;
37.             }
38.         else {
39.             $gagal=false;
40.             }}
41.     if($gagal) {
42.         $this->data->error = '<div class="errora">Angka
43.         yang anda masukkan tidak tercantum, silahkan
44.         ulangi lagi</div>';
45.         $this->data->edit_page = true;
46.         }
47.     else{
48.         $flip=array_flip($array_id);
49.         $data=array_combine($flip, $iterses);
50.         foreach($pec as $row => $value){
51.             $kons[]=array_search($value,$data);
52.             }
53.         foreach($kons as $row => $value){
54.             $pass[]=array_search($value,$flip);
55.             }
56.          $password= implode('|', $pass );
57.          if(!$this->autentifikasi-
58.          >ubah($id,$username,$email,$password,$level)) {
59.             $this->data->error = '<div
60.             class="errora">Username telah
61.             digunakan</div>';
62.             $this->data->edit_page = true;
63.             }
64.          else {
65.             $this->data->sukses = '<div
66.             class="sukses">Perubahan berhasil
67.             diupdate</div>';
68.              }}}
69.
70. else {
71.     $this->data->error = '<div class="errora">Update
72.     data gagal, silahkan mencoba kembali</div>';
73.     $this->data->edit_page = true;
74.     }
75. $query=$this->user_m->view_gambar();
76. $i= rand(1,3);
77. $query_index=$this->user_m->view_index($i);
78. $iterasi = "iterasi_".$i;
79. foreach($query as $row){
80.      $iters[$row->url] = $row->id_gambar;
```

```
81.     $url[$row->id_gambar] = $row->url;
82.     }
83. foreach($query_index as $row){
84.     $iter[$row->id_urutan] = $row->$iterasi;
85.     $itersesi[]=$row->$iterasi;
86.     }
87. $konstanta=array_flip($iter);
88. shuffle($iters);
89. $kaka = array_combine($konstanta,$iters);
90. $coba=implode($kaka);
91. $iterses_=implode($itersesi);
92. $query4=$this->user_m-
93. >add_sesion_gambar($coba,$iterses_);
94. $query5=$this->user_m->get_sesion_id($coba);
95. foreach($query5 as $row){
96.     $_session=$row->sesion_id;
97.     }
98. foreach($kaka as $row=>$value){
99.     $query3=$this->user_m->cek_url($value);
100     foreach($query3 as $row3){
101     $tampil[$row]=$row3->url;
102     }}
103 $this->data->_session=$_session;
104 $this->data->iter = $iter;
105 $this->data->tampil = $tampil;
106 $this->template->set_judul('PTIIKshop')
107    ->set_parsial('sidebar','sidebar_view',$this->data)
108    ->set_parsial('topmenu','top_view',$this->data)
109    ->render('edit_mod',$this->data);
```

Gambar 4.5 Implementasi Algoritma Proses Edit Akun Untuk Pembeli dan Penjual

Penjelasan implementasi algoritma proses *edit* akun untuk pembeli dan penjual pada Gambar 4.5 yaitu:

1. Baris 2-18: menginisialisasi menampilkan data yang dibawa dari halaman sebelumnya dan aturan pengisian *form edit* akun beserta menampilkan data yang dibawa dari halaman sebelumnya.
2. Baris 19: mencari data susunan gambar dan karakter sesuai dengan id sesion
3. Baris 20-28: proses untuk menguraikan data sesuai dengan id sesion
4. Baris 29: menghitung panjang karakter password yang dimasukkan pengguna

5. Baris 30-32: memisah password yang dimasukkan per dua karakter
6. Baris 33-40: mencocokkan password yang telah dipisah per dua karakter di data tabel session_gambar sesuai dengan id_sesion
7. Baris 42-45: mengeluarkan pesan "Angka yang anda masukkan tidak tercantum, silahkan ulangi lagi" dengan kondisi apabila karakter tidak ditemukan
8. Baris 48-55: proses mencari id gambar dengan mencocokan data sesuai dengan id_sesion
9. Baris 56: implode id gambar sebagai password pengguna
10. Baris 57-58: melakukan proses *update* data pengguna sebelumnya dengan data masukan dari pengguna ke dalam basis data
11. Baris 60-62: kondisi jika gagal melakukan proses *update*
12. Baris 65-67: kondisi jika berhasil melakukan proses *update*
13. Baris 71-73: kondisi jika masukan pengguna tidak sesuai dengan aturan pengisian *form edit* akun
14. Baris 75-109: menampilkan halaman selanjutnya

4.5.5 Implementasi Algoritma Proses *Reset Password* untuk Pembeli dan Penjual

Proses *Reset Password* bertujuan untuk memilih minimal tiga gambar apabila pengguna lupa *password*. Apabila pengguna lupa *password*, maka sistem akan mengirim sebuah *link* yang terhubung dengan halaman *Reset Password*. Halaman ini berisi gambar-gambar yang harus dipilih, minimal tiga gambar, sebagai *password* baru pengguna. Gambar 4.6 merupakan *method* `proses_reset()` sebagai *method* yang mengimplementasikan proses *reset password*.

```
1.  if(!$this->uri->segment(3)){
2.       redirect('user/lupa_form'); }
3.  else {
4.       $this->form_validation->set_rules($this-
5.  >reset_rules);
6.       $key = $this->uri->segment(3);
7.       $query3=$this->user_m->by_key($key);
```

```
8.        if (empty($query3)){
9.              $this->session->set_flashdata('pesan',
10.   '<script language="JavaScript">alert("link yang anda
11.   buka sudah expired, silahkan mengirim kembali email
12.   anda"); window.location = "reset";</script>');
13.             redirect('user/lupa_form');
14.              }
15.       foreach($query3 as $row3){
16.             if (empty($query3)){
17.                   redirect('user/lupa_form');
18.                   }
19.             else if (time() - $row3->expire > 3600 {
20.              $this->session->set_flashdata('pesan',
21.   '<script language="JavaScript">alert("link yang anda
22.   buka sudah expired, silahkan membuka kembali halaman
23.   reset password"); window.location =
24.   "reset";</script>');
25.                  redirect('user/lupa_form');
26.                   }
27.       $this->data->username = $row3->username;
28.       }
29.       if($this->form_validation->run()) {
30.             $query=$this->user_m->cari_username($this-
31.   >data->username);
32.             if($query){
33.             $password = $this->input-
34.   >post('new_password');
35.             $coba= $this->input->post('_session');
36.             $query=$this->user_m->cari_merge($coba);
37.             foreach($query as $row){
38.             $_array=$row->merge;
39.             $_iterses=$row->iterasi;
40.             }
41.             $jumlah = strlen($_array);
42.             for($i=0;$i<$jumlah;$i+=2){
43.                   $array_id[]= substr ($_array, $i, 2);
44.                   $iterses[]= substr ($_iterses, $i, 2);
45.                   }
46.             $num = strlen($password);
47.                   for($i=0;$i<$num;$i+=2){
48.             $pec[]= substr ($password, $i, 2);
49.                   }
50.             foreach($pec as $row => $value){
51.                               if (!in_array($value,
52.   $iterses)){
```

```
53.                                     $gagal = true;
54.                                     break;
55.                                     }
56.                                 else {
57.                                     $gagal=false;
58.                                     }
59.                             }
60.                     if($gagal) {
61.                                 $msg = '<script
62.   language="JavaScript">alert("Angka yang anda masukkan
63.   tidak tercantum, silahkan ulangi lagi");</script>';
64.                                 $this->session-
65.   >set_flashdata('pesan', $msg);
66.
67.          redirect('user/reset/'.$key);
68.                 }
69.                       else{
70.                 $flip=array_flip($array_id);
71.                 $data=array_combine($flip, $iterses);
72.
73.                 foreach($pec as $row => $value){
74.                     $kons[]=array_search($value,$data);
75.                     }
76.                 foreach($kons as $row => $value){
77.                     $id[]=array_search($value,$flip);
78.                 }
79.                 $password= implode('|', $id );
80.                 $username = $this->data->username;
81.                  if(!$this->autentifikasi-
82.   >reset_pass($username,$password)) {
83.                     echo '<script
84.   language="JavaScript">alert("Reset password gagal");
85.   window.location = "reset";</script>';
86.                             return FALSE;
87.                  } else {
88.                     $query=$this->user_m-
89.   >hapus_reset($username);
90.                    $this->session->set_flashdata('pesan',
91.   '<div class="sukses">Reset password berhasil, selamat
92.   berbelanja </div>');
93.
94.   redirect(site_url('user/proses_login'));
95.                 }
96.
97.                 }
```

```
98.                 }
99.                 else echo "username tidak sesuai";
100.                }
101.          $query=$this->user_m->view_gambar();
102.          $query2=$this->user_m->view_acak();
103.          $i= rand(1,3);
104.          $query_index=$this->user_m->view_index($i);
105.          $iterasi = "iterasi_".$i;
106.          foreach($query as $row){
107.                $iters[$row->url] = $row->id_gambar;
108.                $url[$row->id_gambar] = $row->url;
109.                }
110.          foreach($query_index as $row){
111.                $iter[$row->id_urutan] = $row->$iterasi;
112.                $iterses[]=$row->$iterasi;
113.                }
114.          $konstanta=array_flip($iter);
115.
116.          shuffle($iters);
117.          $kaka = array_combine($konstanta,$iters);
118.          $coba=implode($kaka);
119.          $iterses_=implode($iterses);
120.          $query4=$this->user_m-
121.    >add_sesion_gambar($coba,$iterses_);
122.          $query5=$this->user_m->get_sesion_id($coba);
123.          foreach($query5 as $row){
124.                $_session=$row->sesion_id;
125.                }
126.          foreach($kaka as $row=>$value){
127.                $query3=$this->user_m->cek_url($value);
128.                foreach($query3 as $row3){
129.                      $tampil[$row]=$row3->url;
130.                      }
131.          }
132.          $this->data->_session=$_session;
133.          $this->data->iter = $iter;
134.          $this->data->tampil = $tampil;
135.          $this->data->key= $this->input->post('abc');
136.          $this->data->reset = true;
137.          $this->template->set_judul('PTIIKshop')
138.           ->set_parsial('sidebar','sidebar_view',$this-
139.    >data)
140.           ->set_parsial('topmenu','top_view',$this->data)
141.           ->render('reset_form_mod',$this->data);
```

Gambar 4.6 Implementasi Algoritma Proses Reset Password untuk Pembeli dan Penjual

Penjelasan implementasi algoritma proses reset password untuk pembeli dan penjual pada Gambar 4.6 yaitu:

1. Baris 1-2: kondisi jika alamat yang dimasukkan pengguna tidak mengandung url yang diberikan melalui email
2. Baris 4-5: menginisialisasi *rules* yang telah ditetapkan
3. Baris 6: menginisialisasi *link reset*
4. Baris 7: mencari data sesuai dengan *link reset* yang telah didapatkan
5. Baris 8-18: kondisi jika data yang sesuai dengan *link reset* tidak ditemukan
6. Baris 19: menentukan batas berlakunya *link reset*
7. Baris 20-25: kondisi jika *link reset* telah melewati masa berlaku
8. Baris 26: menginisialisasi *username* yang ditemukan
9. Baris 32-33: menginisialisasi data yang dibawa dari halaman sebelumnya
10. Baris 34: mencari data sesuai dengan id session
11. Baris 35-43: menguraikan data sesuai dengan id session
12. Baris 44: menghitung panjang karakter password yang dimasukkan pengguna
13. Baris 45-47: memisah password yang dimasukkan per dua karakter
14. Baris 48-57: mencocokkan password yang telah dipisah per dua karakter di data tabel session_gambar sesuai dengan id_sesion
15. Baris 59-65: mengeluarkan pesan "Angka yang anda masukkan tidak tercantum, silahkan ulangi lagi" dengan kondisi apabila karakter tidak ditemukan
16. Baris 67-75: proses mencari id gambar dengan mencocokan data sesuai dengan id_sesion
17. Baris 76: implode id gambar sebagai password pengguna
18. Baris 77: mendeklarasikan username untuk ditampilkan
19. Baris 78-79: melakukan proses *reset password* pengguna sebelumnya dan diisikan dengan *password* baru dari pengguna ke dalam basis data
20. Baris 80-83: kondisi jika gagal melakukan proses *reset*
21. Baris 87-94: kondisi jika berhasil melakukan proses *reset*
22. Baris 101-141: menampilkan halaman selanjutnya

4.6 Implementasi Antarmuka Aplikasi

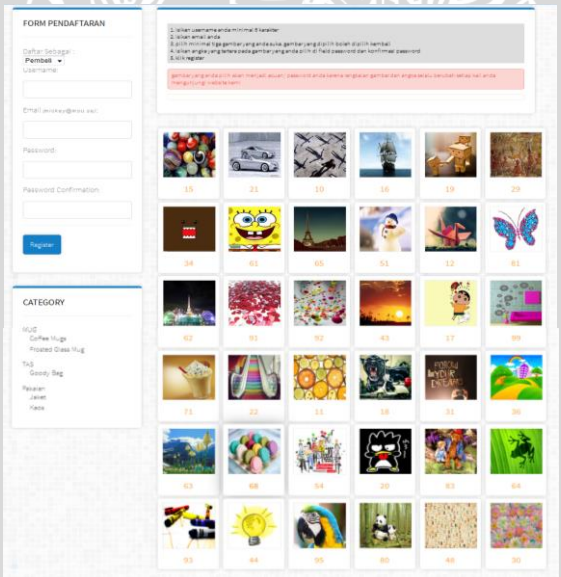
Antarmuka Implementasi OTP berbasis gambar pada *website ecommerce* PTIIK digunakan oleh pengguna untuk berinteraksi dengan aplikasi. Antarmuka aplikasi ini dibagi menjadi empat, yaitu antarmuka untuk halaman pengunjung, halaman penjual, halaman pembeli, dan halaman *administrator*.

4.6.1 Implementasi Antarmuka Halaman Pengunjung

Antarmuka pengguna untuk halaman pengunjung dapat dilihat oleh siapa saja tanpa melalui proses *login*

a. Halaman Register

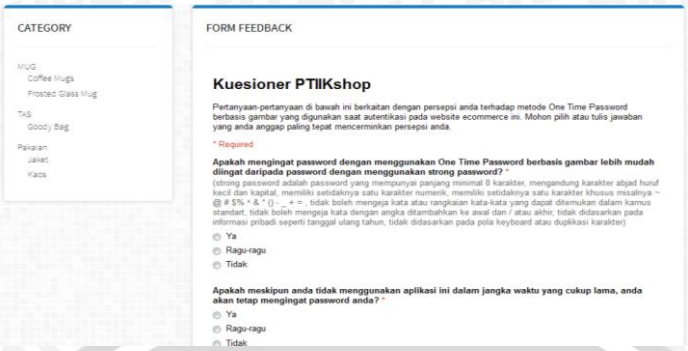
Halaman *register* menampilkan *form* pendaftaran untuk mendaftar sebagai pembeli atau penjual. Gambar 4.7 akan menunjukkan implementasi tampilan antarmuka dari halaman *registrasi* yang mengacu pada perancangan antarmuka halaman *registrasi* pengunjung Sub Bab 3.2.3.5 nomor 1 bagian a.



Gambar 4.7 Tampilan Antarmuka Halaman Register

b. Halaman Feedback

Halaman *fedback* menampilkan *form* kuesioner. Gambar 4.8 akan menunjukkan implementasi tampilan antarmuka dari halaman *feedback* yang mengacu pada perancangan antarmuka halaman *feedback* pengunjung Sub Bab 3.2.3.5 nomor 1 bagian b



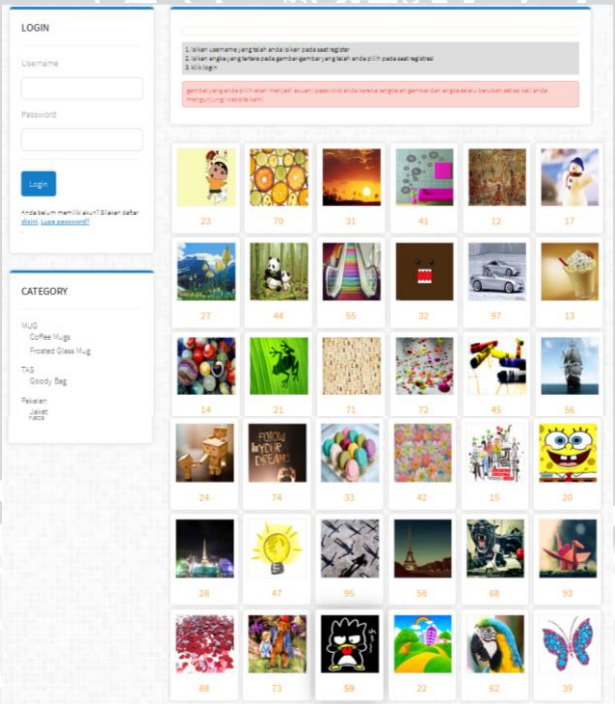
Gambar 4.8 Tampilan Antarmuka Halaman Feedback

4.6.2 Implementasi Antarmuka Halaman Pembeli

Antarmuka pengguna untuk halaman pembeli di dapat dilihat oleh siapa saja tanpa melalui proses *login.*

a. Halaman Login

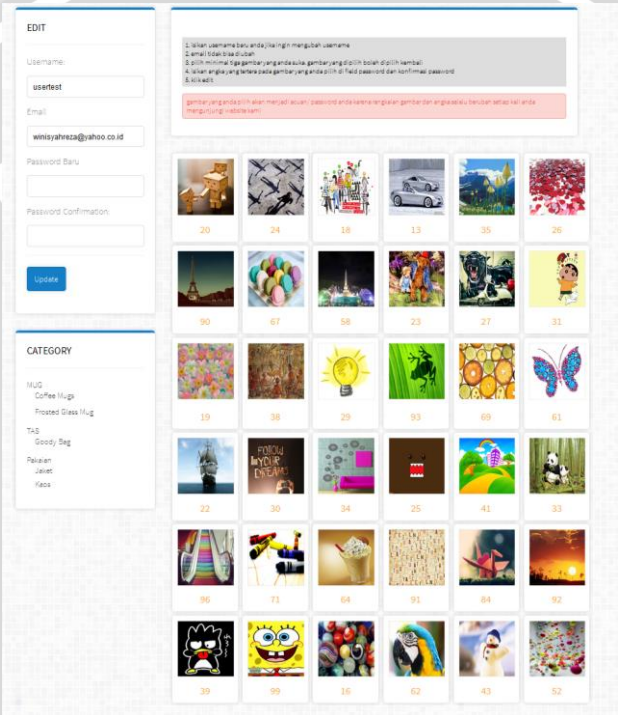
Halaman *login* merupakan salah satu antarmuka pembeli. Halaman *login* berfungsi untuk masuk ke dalam halaman pembeli. Gambar 4.9 akan menunjukkan implementasi tampilan antarmuka dari halaman *login* yang mengacu pada perancangan antarmuka halaman *login*. Sub Bab 3.2.3.5 nomor 2 bagian a.



Gambar 4.9 Tampilan Antarmuka Halaman Login

b. Halaman Edit Akun

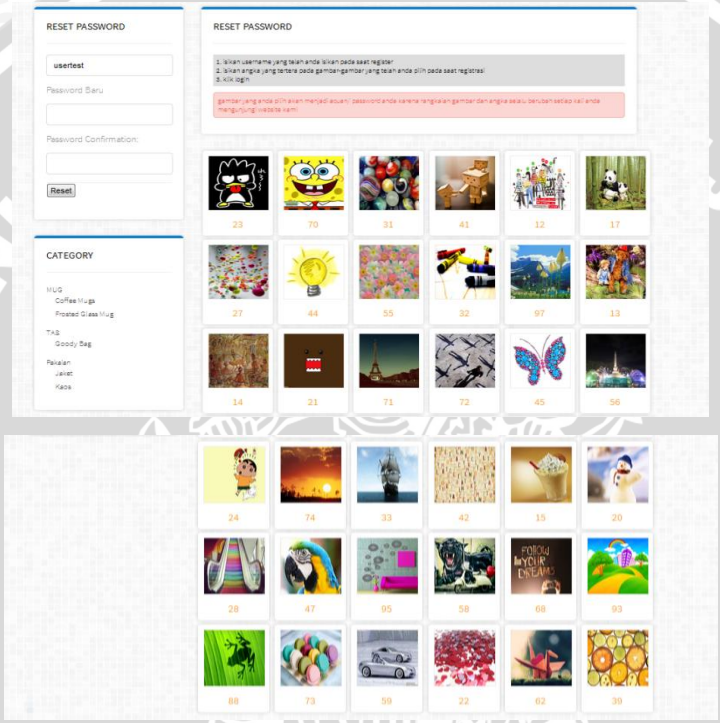
Halaman *edit* akun merupakan salah satu antarmuka pembeli.. Halaman *edit* akunberfungsi bagi pembeli untuk mengubah *username* atau *passwordnya*. Gambar 4.10 akan menunjukkan implementasi tampilan antarmuka dari halaman *edit* akun yang mengacu pada perancangan antarmuka halaman *edit* akun Sub Bab 3.2.3.5 nomor 2 bagian b.



Gambar 4.10 Tampilan Antarmuka Halaman Edit Akun

c. Halaman Reset Password

Halaman *reset password* merupakan salah satu antarmuka pembeli. Halaman *reset password* berfungsi untuk menyetel ulang password pembeli yang dikarenakan pembeli tersebut lupa dengan *password*nya Gambar 4.11 akan menunjukkan implementasi tampilan antarmuka dari halaman *reset password* yang mengacu pada perancangan antarmuka halaman *reset password* Sub Bab 3.2.3.5 nomor 2 bagian c.



Gambar 4.11 Tampilan Antarmuka Halaman Reset Password

4.6.3 Implementasi Antarmuka Halaman Penjual

Antarmuka pengguna untuk halaman penjual berupa sebuah halaman untuk menjual produk kepada pembeli. Antarmuka halaman penjual terdiri dari halaman Login, halaman *List* Barang yang Dijual, halaman *Edit* Akun dan halaman *Reset Password*

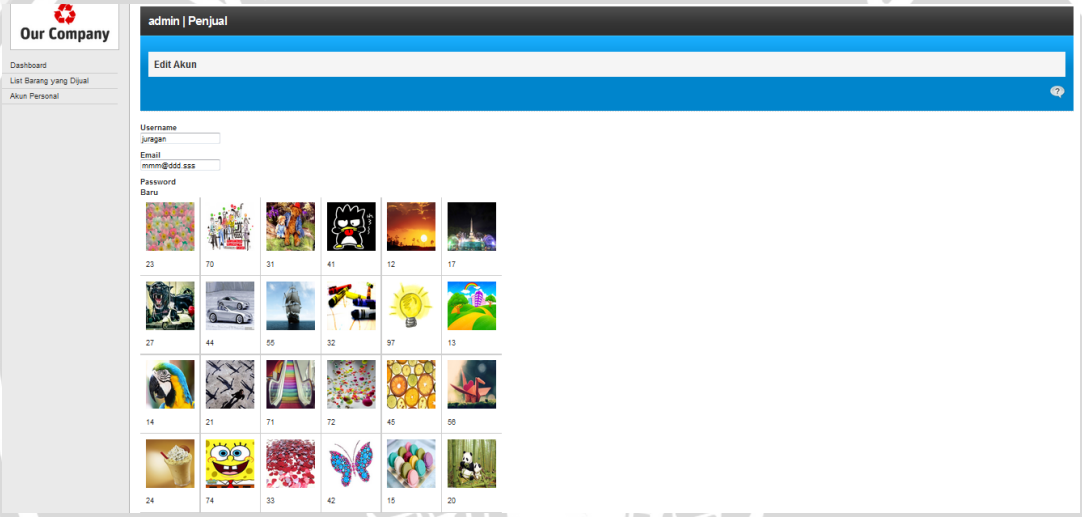
a. Halaman Login

Halaman *login* merupakan salah satu antarmuka penjual.Halaman *login* berfungsi untuk masuk kedalam halaman penjual. Implementasi tampilan antarmuka dari halaman *login* yang mengacu pada perancangan

antarmuka halaman*login* Sub Bab 3.2.3.5 nomor 3 bagian a sama dengan halaman *login* pembeli (lihat Gambar 4.9).

b. Halaman Edit Akun

Halaman edit akun merupakan salah satu antarmuka penjual. Halaman *edit* akun berfungsi berfungsi bagi penjual untuk mengubah *username* atau *password*nya. Gambar 4.12 akan menunjukkan implementasi tampilan antarmuka dari halaman *edit* akun yang mengacu pada perancangan antarmuka halaman *edit* akunSub Bab 3.2.3.5 nomor 3 bagian b



Gambar 4.12 Tampilan Antarmuka Halaman Edit Akun

c. Halaman Reset Password

Halaman *reset password* merupakan salah satu antarmuka penjual. Halaman *reset password* berfungsi untuk menyetel ulang *password* karena penjual lupa pada *password*nya. Implementasi tampilan antarmuka dari halaman *reset password* yang mengacu pada perancangan antarmuka halaman *reset pasword* Sub Bab 3.2.3.5 nomor 3 bagian c sama dengan halaman *reset password* pembeli (lihat Gambar 4.11).

BAB V
PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis terhadap Implementasi OTP Berbasis Gambar pada *Website Ecommerce* PTIIK yang telah dibangun. Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian validasi, pengujian keamanan dan pengujian *User Acceptance*. Pada pengujian validasi digunakan teknik pengujian *Black Box (Black Box Testing)*. Pada pengujian keamanan dilakukan dengan melakukan serangan *Man In The Middle* dan *Brute Force*. Pada pengujian *User Acceptance* dilakukan dengan menyebarkan kuesioner kepada user pengguna sistem.

5.1 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item - item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan. Pada skripsi ini dilakukan pengujian validasi terhadap Impelementasi OTP Berbasis Gambar pada *Website Ecommerce* PTIIK.

5.1.1 Kasus Uji Validasi

Tabel 5.1 Kasus uji untuk pengujian validasi *login* sah untuk pembeli dan penjual

Nama Kasus Uji	Kasus Uji Login Sah
Objek Uji	SRS_002_01 dan SRS_003_01
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas *login* bagi pembeli dan penjual.

Prosedur Uji	1. Pembeli dan penjual membuka *link* untuk *login.* 2. Pembeli mengisi semua *field* yang ada. Saat mengisi *field* password, pembeli diharuskan memasukkan karakter yang tertera pada gambar-gambar yang dipilih saat melakukan *register*.
Hasil yang diharapkan	Aplikasi dapat melakukan penyeleksian kondisi *login* pada *database* berdasar data yang dimasukkan dan jika penyeleksian kondisi ini benar, maka pembeli dan penjual akan mengakses ke sistem sesuai dengan hak aksesnya.

Tabel 5.2 Kasus uji untuk pengujian validasi *login* tidak sah untuk pembeli dan penjual

Nama Kasus Uji	Kasus Uji Login Tidak Sah
Objek Uji	SRS_002_01 dan SRS_003_01
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas *login* bagi Penjual dan Pembeli.
Prosedur Uji	1. Pembeli dan penjual membuka *link* untuk *login.* 2. Pembeli mengisi semua *field* yang ada dan mengisikan *username* atau *password* dengan data yang salah
Hasil yang diharapkan	Aplikasi dapat melakukan penyeleksian kondisi login pada *database* berdasar data yang dimasukkan dan jika penyeleksian kondisi ini salah, maka tidak akan mengakses ke sistem dan aplikasi menampilkan pesan kesalahan.

Tabel 5.3 Kasus uji untuk pengujian validasi register

Nama Kasus Uji	Kasus Uji Register
Objek Uji	SRS_001_01
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa

	aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas halaman untuk melakukan proses pendaftaran pembeli atau penjual
Prosedur Uji	1. Pengunjung membuka *link* untuk *register* 2. Pengunjung memilih daftar sebagai pembeli atau penjual kemudian mengisi semua *field* yang ada. Khusus untuk mengisi password, pengunjung diharuskan memilih minimal tiga gambar dan memasukkan karakter yang tertera pada gambar-gambar yang dipilih tersebut di *field password* dan konfirmasi *password*
Hasil yang diharapkan	Aplikasi menampilkan data pengunjung yang sudah terdaftar sesuai dengan *username* yang dimasukkan dan *password* gambar yang dipilih.

Tabel 5.4 Kasus uji untuk pengujian validasi mengedit akun personal

Nama Kasus Uji	Kasus Uji Mengedit Akun Personal
Objek Uji	SRS_002_02 dan SRS_003_02
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas halaman untuk mengubah akun personal pembeli dan penjual.
Prosedur Uji	1. Pembeli dan penjual membuka menu *edit* akun 2. Pembeli dan penjual mengisi *field* yang ada. Saat untuk mengisi field password, pembeli dan penjual diharuskan memasukkan karakter yang tertera pada gambar-gambar yang dipilihnya
Hasil yang diharapkan	Aplikasi menampilkan halaman dengan *Username* atau *password* pembeli dan penjual ter*update* oleh *username atau password* baru yang baru saja dimasukkan.

Tabel 5.5 Kasus uji untuk pengujian validasi *reset password*

Nama Kasus Uji	Kasus Uji Mengelola *Reset Password*
Objek Uji	SRS_002_03 dan SRS_003_03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas halaman untuk menyetel ulang *password* karena lupa
Prosedur Uji	1. Pembeli membuka *link* Lupa Password 2. Pembeli memasukkan alamat *email*nya yang digunakan untuk mendaftar 3. Pembeli membuka link di *email* yang telah dikirim oleh sistem 4. Pembeli mengisi field password dan konfirmasi password dengan memasukkan karakter yang tertera pada gambar-gambar yang dipilihnya
Hasil yang diharapkan	Masuk ke dalam halaman pembeli atau penjual tersebut dengan password yang baru

5.1.2 Hasil Pengujian Validasi

Tabel 5.6 Hasil pengujian validasi

No	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validitas
1	Kasus Uji Login Sah	Aplikasi dapat melakukan penyeleksian kondisi *login* pada *database* berdasar data yang dimasukkan dan jika penyeleksian kondisi ini benar, maka pembeli dan penjual akan mengakses ke sistem sesuai dengan	Aplikasi dapat melakukan penyeleksian kondisi *login* pada *database* berdasar data yang dimasukkan dan jika penyeleksian kondisi ini benar, maka pembeli dan penjual akan mengakses ke sistem sesuai dengan	Valid

		hak aksesnya.	hak aksesnya.	
2	Kasus Uji Login Tidak Sah	Aplikasi dapat melakukan penyeleksian kondisi login pada *database* berdasar data yang dimasukkan dan jika penyeleksian kondisi ini salah, maka tidak akan mengakses ke sistem dan aplikasi menampilkan pesan kesalahan.	Aplikasi dapat melakukan penyeleksian kondisi login pada *database* berdasar data yang dimasukkan dan jika penyeleksian kondisi ini salah, maka tidak akan mengakses ke sistem dan aplikasi menampilkan pesan kesalahan.	Valid
4	Kasus Uji Register	Aplikasi menampilkan data pengunjung yang sudah terdaftar sesuai dengan *username* yang dimasukkan dan *password* gambar yang dipilih.	Aplikasi menampilkan data pengunjung yang sudah terdaftar sesuai dengan *username* yang dimasukkan dan *password* gambar yang dipilih.	Valid
7	Kasus Uji Edit Akun Personal	Aplikasi menampilkan halaman dengan *Username* atau *password* pembeli dan penjual ter*update* oleh *username atau password* baru yang baru saja dimasukkan.	Aplikasi menampilkan halaman dengan *Username* atau *password* pembeli dan penjual ter*update* oleh *username atau password* baru yang baru saja dimasukkan.	Valid
9	Kasus Uji Mengelola *Reset Password*	Masuk ke dalam halaman pembeli atau penjual tersebut dengan password yang baru	Masuk ke dalam halaman pembeli atau penjual tersebut dengan password yang baru	Valid

5.1.3 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat konformitas antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas sistem telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

5.2 Pengujian Keamanan

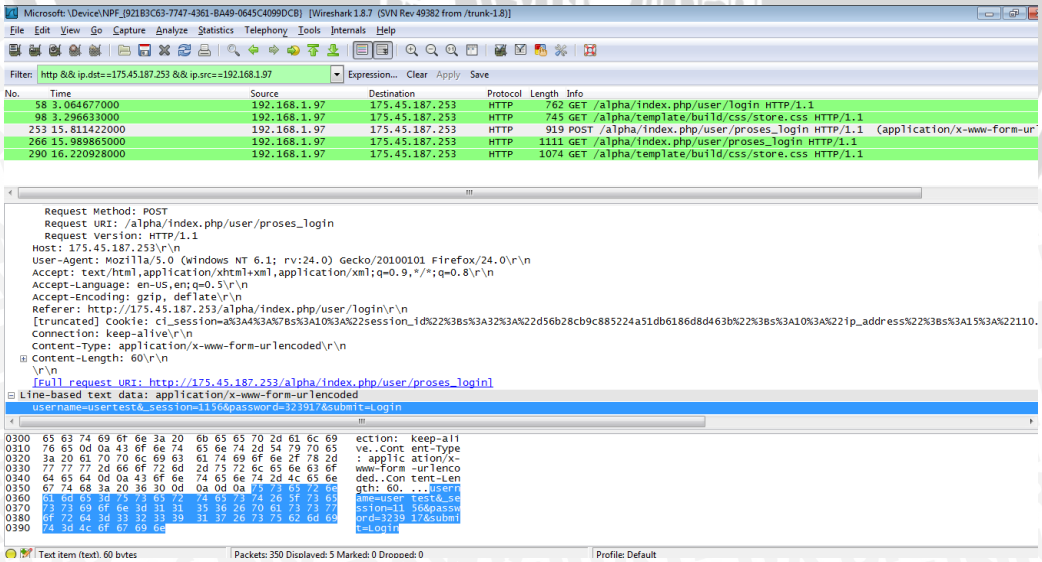
Pengujian keamanan dilakukan untuk mengetahui seberapa aman sistem yang dibangun dari serangan-serangan jaringan. Dalam penelitian ini dilakukan percobaan serangan *man in the middle* dan perhitungan kemungkinan sistem dibobol *password*nya menggunakan *brute force*.

5.2.1 Man in The Middle

5.2.1.1 Kasus Uji Man in The Middle

Pengujian dilakukan dengan menggunakan *wireshark*. *Wireshark* akan menangkap paket-paket jaringan dan menampilkan informasi paket tersebut. Pengujian dilakukan tiga kali berturut-turut saat proses *login* dengan pengguna yang sama. Berikut adalah informasi yang ditangkap oleh *wireshark* saat pengguna masuk ke dalam sistem.

Pengujian Pertama:

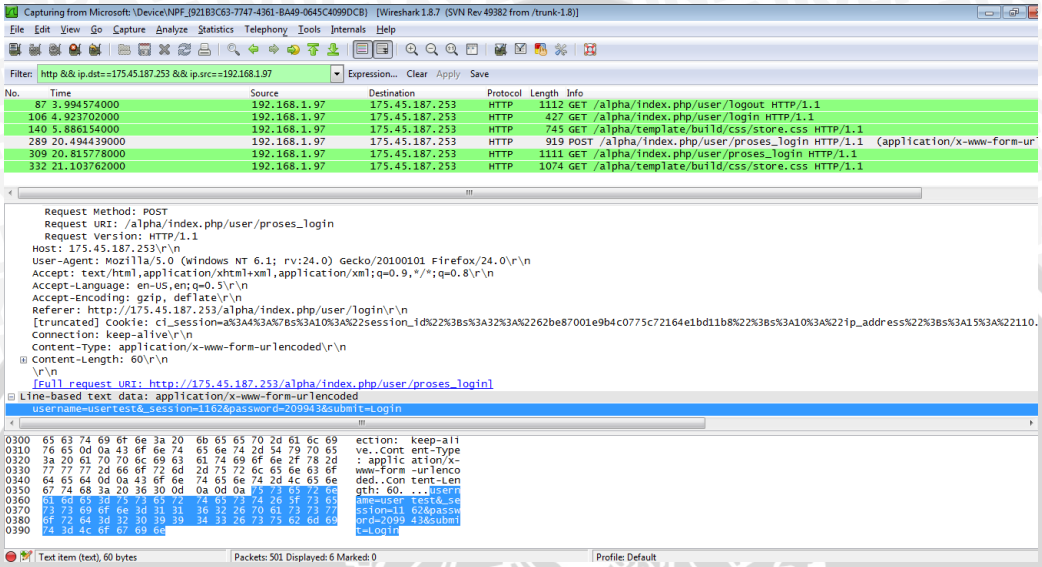


Gambar 5.1 Gambar *Printscreen* Hasil Pengujian dengan Menggunakan *Wireshark* Pengujian Pertama

Informasi yang didapat dari pengujian pertama adalah:

```
username=usertest&_session=1156&password=323917&submit=Login
```

Pengujian Kedua:

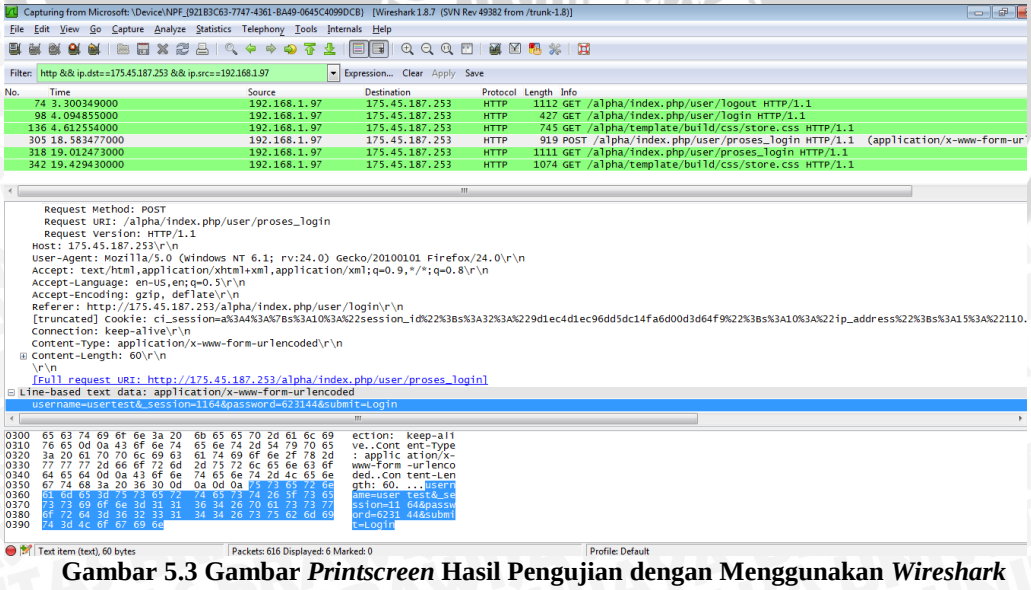


Gambar 5.2 Gambar *Printscreen* Hasil Pengujian dengan Menggunakan *Wireshark* Pengujian Kedua

Informasi yang didapat dari pengujian kedua adalah:

```
username=usertest&_session=1162&password=209943&submit=Login
```

Pengujian Ketiga:



Gambar 5.3 Gambar *Printscreen* Hasil Pengujian dengan Menggunakan *Wireshark* Pengujian Ketiga

Informasi yang didapat dari pengujian ketiga adalah:

```
username=usertest&_session=1164&password=623144&submit=Login
```

5.2.1.2 Analisis Hasil Pengujian Keamanan dari Serangan *Man in The Middle*

Pengujian keamanan dari serangan Man in The Middle dilakukan dengan menangkap informasi ketika pengguna melakukan proses *login*. Proses pengujian ini dilakukan selama tiga kali proses *login*. Berdasarkan hal tersebut maka dapat diambil kesimpulan bahwa dari tiga kali proses *login* yang dilakukan oleh pengguna yang sama, *password* yang diinputkan oleh pengguna berbeda-beda sehingga *password* sebenarnya tidak dapat diketahui melalui cara ini

5.2.2 Brute Force

5.2.2.1 Kasus Uji Brute Force

Pengujian dilakukan dengan dua tahap yaitu mencoba melakukan brute force terhadap sistem dengan menggunakan *software Fireforce* dan dengan menggunakan program buatan sendiri yang menggunakan CURL.

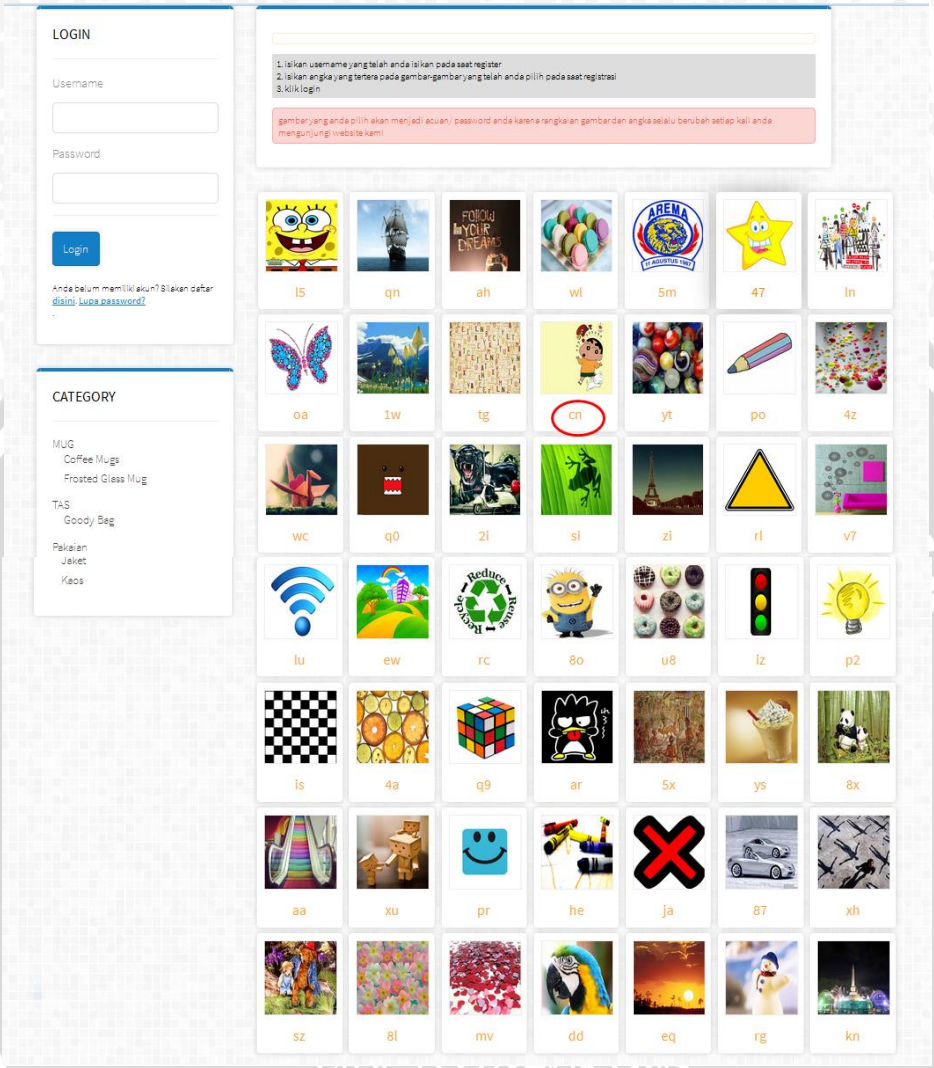
1. *Brute Force* Menggunakan *Fireforce*

Dalam percobaan pembobolan *password* ini, penulis melakukan pengujian sebanyak dua puluh kali. Penulis mengambil salah satu data pengguna yaitu pengguna yang mempunyai *username* bernama ‘aredoes’. Pengguna ‘aredoes’ ini mempunyai *password* sebagai berikut:

Tabel 5.7 Tabel Daftar Password Pengguna ‘aredoes’

Password urutan pertama	
Password urutan kedua	
Password urutan ketiga	

Urutan karakter dan gambar saat melakukan proses *brute force* adalah sebagai berikut:

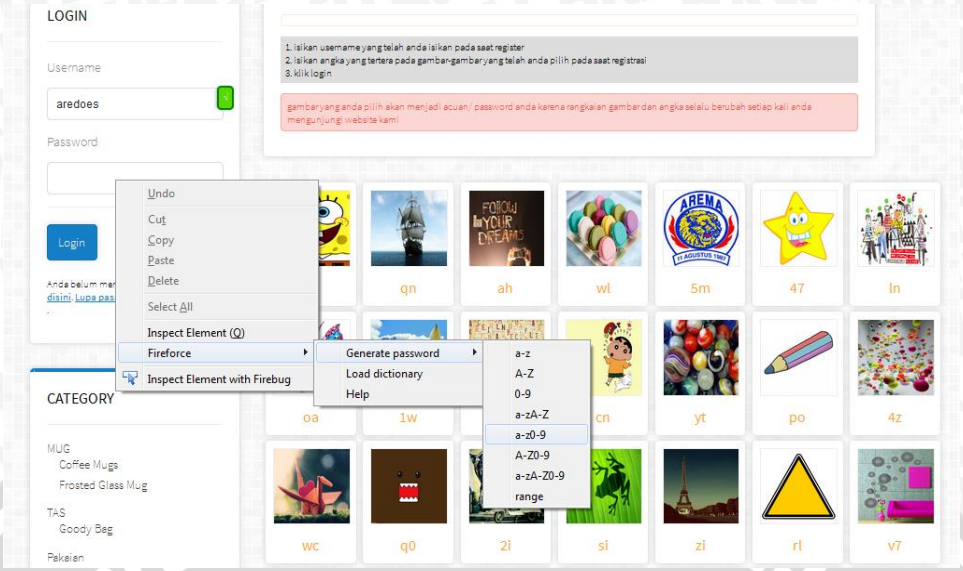


Gambar 5.4 Gambar *Printscreen* Halaman Login Saat Pengujian Brute Force

Untuk melakukan proses *brute force* dengan *Fireforce*, langkah-langkahnya adalah sebagai berikut:

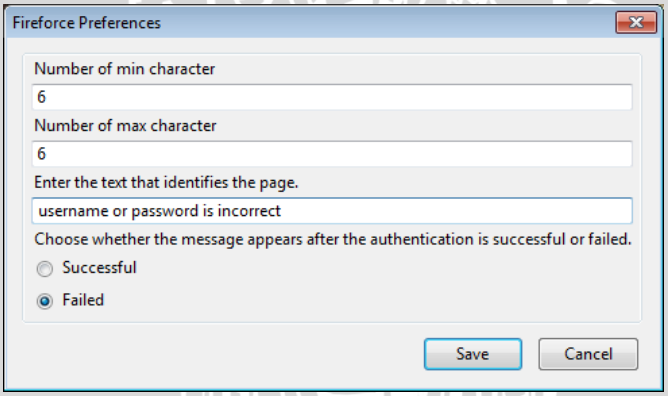
1. Memasukkan *username* pengguna
2. Klik kanan pada *field password* pilih Fireforce >> Generate Password >> a-z0-9

   Penulis memasukkan rentang karakter 0-9 dan a-z karena yang dimasukkan ke dalam *field password* adalah kombinasi antara angka dengan huruf.



Gambar 5.5 Gambar *Printscreen* Pemilihan Kombinasi Untuk Melakukan Brute Force

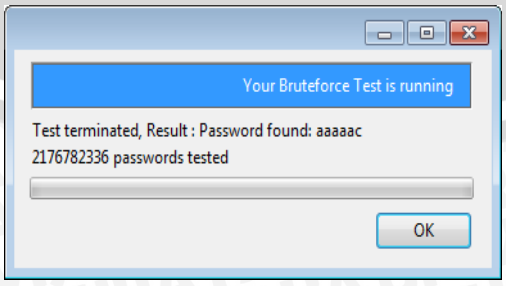
3. Penulis menguji coba panjang *password* terkecil yaitu sebanyak enam karakter atau tiga gambar



Gambar 5.6 Gambar *Printscreen* Pengisian Pengaturan Untuk Melakukan Brute Force

4. Klik save untuk melakukan proses *brute force*

Hasil yang ditemukan pada pengujian pertama adalah sebagai berikut:

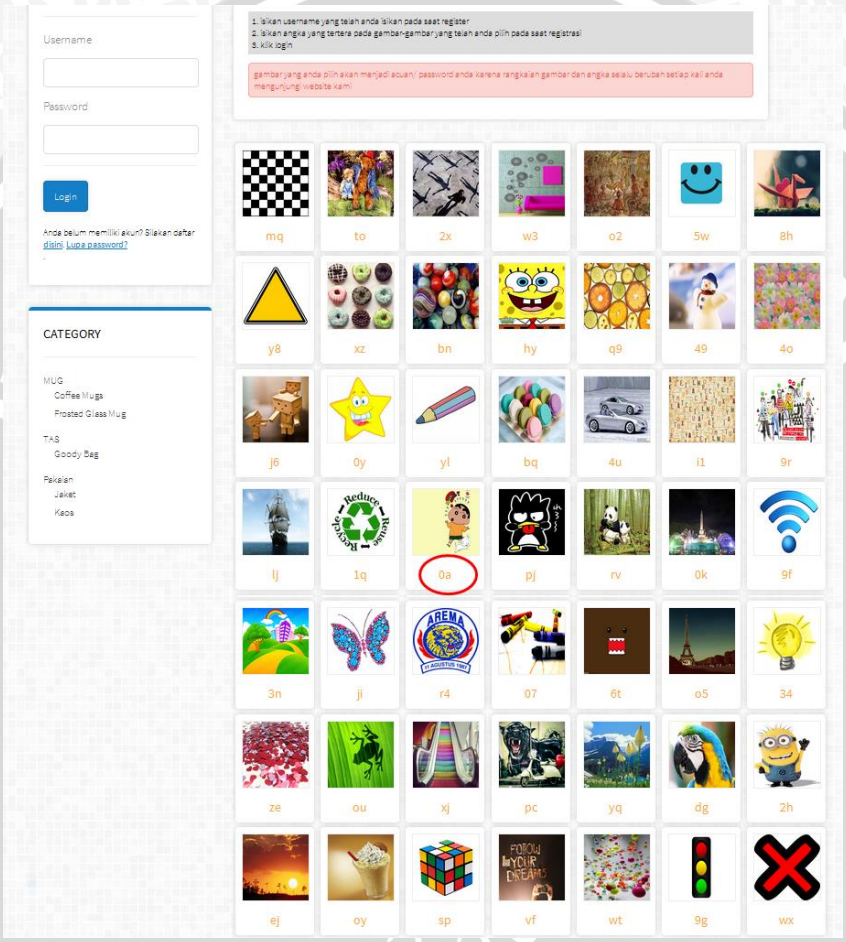


Gambar 5.7 Gambar *Printscreen* Hasil Brute Force Pengujian Pertama

Langkah-langkah di atas diulang sampai dua puluh kali pengujian.

2. Brute Force Menggunakan Program yang Dibuat dengan PHP Curl

Pengujian *brute force* dengan menggunakan program yang dibuat dengan PHP Curl ini juga menggunakan pengguna bernama ‘aredoes’ dimana saat melakukan pengujian bruteforce, susunan gambar dan karakternya adalah sebagai berikut:



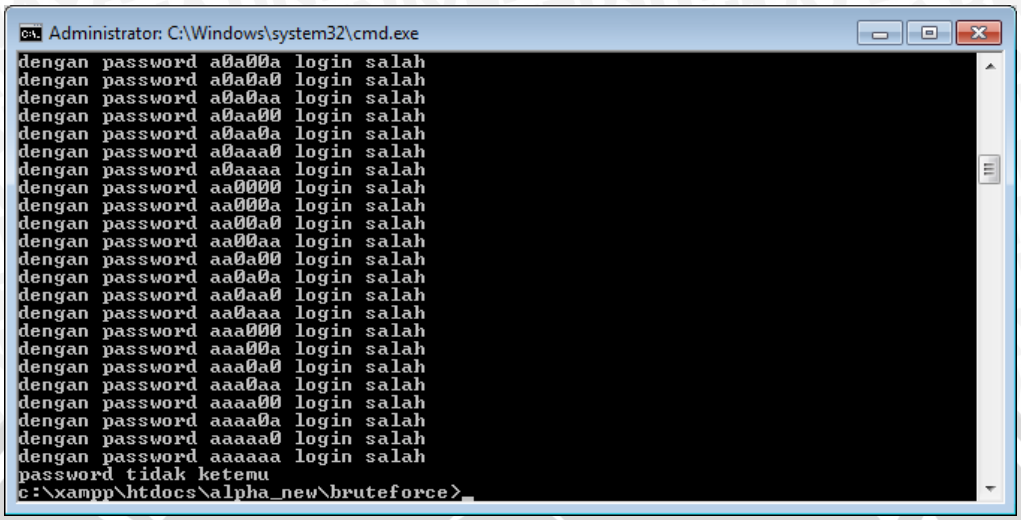
Gambar 5.8 Gambar *Printscreen* Susunan Gambar dan Karakter Ketika Melakukan Brute Force

Pengujian *brute force* menggunakan PHP Curl ini dibagi menjadi dua bagian yaitu:

a. Direct post *username* dan *password*

Direct post field username dan *password* ini merupakan hal yang dilakukan oleh program-program *brute force* pada umumnya, karena pada metode autentikasi web pada umumnya hanya menggunakan *username* dan *password*

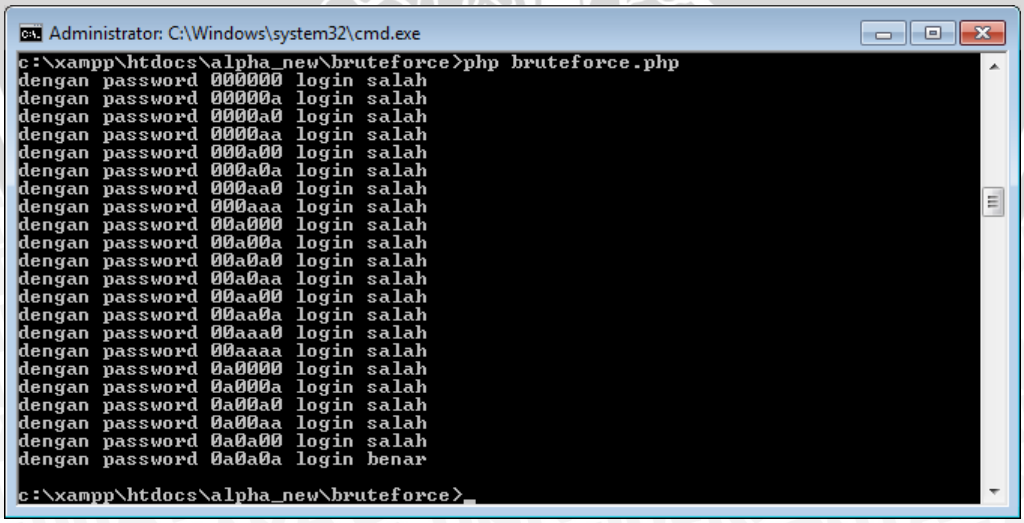
dalam proses autentikasinya. Hasil dari pengujian ini adalah password tidak ditemukan seperti gambar di bawah ini:



Gambar 5.9 Gambar *Printscreen* Hasil *Brute* Force dengan *Direct Post Field Username* dan *Password*

b. Direct post *username*, *password* dan *id session*

Direct post field username, *password* dan *id session* ini dilakukan karena pada metode autentikasi OTP berbasis gambar menggunakan tiga objek tersebut dalam proses autentikasinya. Hasil dari pengujian ini adalah password dapat ditemukan seperti gambar di bawah ini



Gambar 5.10 Gambar *Printscreen* Hasil *Brute* Force dengan *Direct Post Field Username*, *Password* dan *Id Session*

5.2.2.2 Analisis Hasil Pengujian Keamanan dari Serangan *Brute Force*

Dalam percobaan pembobolan password ini dengan menggunakan *software Fireforce*, penulis menguji sebanyak dua puluh kali dengan mencoba panjang password terkecil yaitu sebanyak enam karakter atau tiga gambar. Penguji juga mengambil salah satu data pengguna yaitu pengguna yang mempunyai *username* bernama ‘aredoes’ dengan hasil sebagai berikut.

Tabel 5.8 Tabel Daftar Hasil *Password* yang Ditemukan Melalui *Software Fireforce*

Pengujian ke-	Hasil
1	*Password found*: aaaaac
2	*Password found*: aaaaaf
3	*Password found*: aaaaab
4	*Password found*: aaaaaf
5	*Password found*: aaaaaf
6	*Password found*: aaaaaf
7	*Password found*: aaaaac
8	*Password found*: aaaaaf
9	*Password found*: aaaaab
10	*Password found*: aaaaac
11	*Password found*: aaaaab
12	*Password found*: aaaaae
13	*Password found*: aaaaae
14	*Password found*: aaaaaf
15	*Password found*: aaaaae
16	*Password found*: aaaaae
17	*Password found*: aaaaad
18	*Password found*: aaaaae
19	*Password found*: aaaaaf
20	*Password found*: aaaaaf

Dari dua puluh hasil proses pengujian, *password* yang ditemukan oleh *software fireforce* tidak berhasil membuka halaman dari pengguna ‘aredoes’. Software ini tidak dapat menemukan password sebenarnya dari pengguna ‘aredoes’ yang bernilai ‘cncncn’. Sama halnya dengan hasil pengujian dengan menggunakan *direct post username* dan *password* menunjukkan bahwa *password* tidak dapat ditemukan. Kedua pengujian ini tidak dapat menemukan *password* sebenarnya dikarenakan variabel yang diinisialisasi adalah *username* dan

password di mana pada metode OTP berbasis gambar ini terdapat satu variabel lagi yaitu *id session*. Sehingga pada pengujian dengan *direct post username*, *password* dan *id session* menghasilkan bahwa *password* dapat ditemukan. Dengan ini maka *password* dari pengguna dapat ditemukan dengan catatan sebagai berikut:

a. Penyerang mengetahui metode yang digunakan oleh OTP berbasis gambar ini yaitu dengan menambahkan satu variabel *id session*
b. Tidak memuat ulang halaman web ketika melakukan proses *brute force* karena *id session* akan selalu berubah sehingga *password* yang ditemukan tidak dapat digunakan.

5.3 User Acceptance Testing

5.3.1 Kasus Uji User Acceptance Testing

UAT (*User Acceptance Testing*) digunakan untuk mengetahui apakah sistem yang dibangun telah dapat diterima atau belum oleh pengguna sistem. UAT difokuskan pada implementasi OTP yang terdapat pada proses *autentikasi*. UAT dilakukan dengan memberikan kuesioner kepada 50 mahasiswa PTIIK untuk menilai keseluruhan dan memberikan komentar maupun saran terhadap implementasi OTP tersebut. Pada skripsi ini dilakukan UAT terhadap Implementasi OTP Berbasis Gambar pada Website Ecommerce PTIIK.

Di bawah ini adalah hasil rekapitulasi jawaban dari delapan pertanyaan yang diberikan kepada responden.

Tabel 5.8 Hasil kuesioner user acceptance testing

Pertanyaan Nomor ke-	Pertanyaan	Ya	Ragu-ragu	Tidak
1	Apakah mengingat password dengan menggunakan One Time Password berbasis gambar lebih mudah diingat daripada password dengan menggunakan strong password?	37	8	5
2	Apakah meskipun anda tidak menggunakan aplikasi ini dalam jangka waktu yang cukup lama, anda akan tetap mengingat password anda?	17	26	7

3	Menurut anda, apakah sistem autentikasi menggunakan One Time Password mudah digunakan?	40	4	6
4	Menurut anda, apakah sistem autentikasi menggunakan One Time Password merupakan ide yang bagus?	39	8	3
7	Apakah anda menyukai metode One Time Password berbasis gambar daripada metode password dengan menghafal karakter-karakter seperti biasanya?	34	12	4
9	Apakah anda lebih merasa aman memiliki password dengan metode One Time Password berbasis gambar dibandingkan dengan memiliki strong password?	29	17	4
Jumlah		196	75	29

Tabel 5.9 Hasil jawaban dari pertanyaan nomor 5

Pertanyaan	Cukup	Terlalu banyak	Terlalu sedikit
Menurut anda, apakah tiga puluh enam gambar cukup untuk menjadi alternatif pilihan password?	28	11	11

Tabel 5.10 Hasil jawaban dari pertanyaan nomor 6

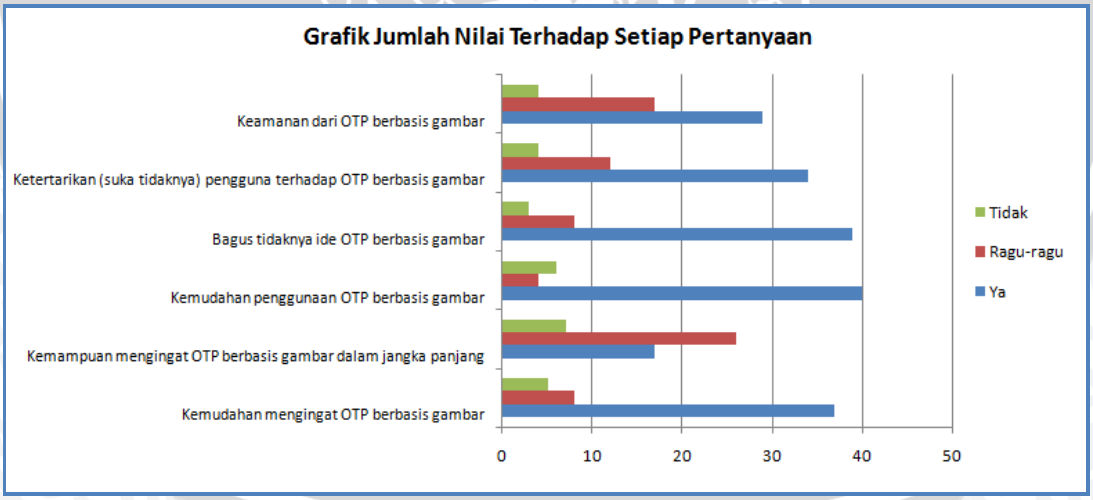
Responden	Menurut anda, berapa gambar yang bisa dijadikan alternatif pilihan password untuk pengguna?
Responden 1	100
Responden 2	20
Responden 3	9
Responden 4	100
Responden 5	150
Responden 6	16
Responden 7	50
Responden 8	13
Responden 9	30
Responden 10	100
Responden 11	36
Responden 12	24
Responden 13	9

Responden 14	10
Responden 15	20
Responden 16	9
Responden 17	10
Responden 18	50
Responden 19	9
Responden 20	10
Responden 21	25
Responden 22	160
Responden 23	12
Responden 24	9
Responden 25	99
Responden 26	30
Responden 27	40
Responden 28	64
Responden 29	50
Responden 30	9
Responden 31	9
Responden 32	128
Responden 33	36
Responden 34	25
Responden 35	25
Responden 36	25
Responden 37	25
Responden 38	100
Responden 39	100
Responden 40	24
Responden 41	16
Responden 42	20
Responden 43	30
Responden 44	100
Responden 45	9
Responden 46	20
Responden 47	30
Responden 48	40
Responden 49	50
Responden 50	50
Rata- rata	**42,7**

5.3.2 Analisis Hasil User Acceptance Testing

Proses analisis terhadap hasil pengujian sistem terhadap pengguna dilakukan dengan menghitung jumlah tiap nilai dari semua koresponden. Hasil penilaian terhadap pengguna mendapatkan nilai yang ditunjukan pada Gambar 5.7.

Pada Gambar 5.7 diperoleh bahwa 74% responden berpendapat bahwa password dengan metode OTP berbasis gambar ini mudah diingat daripada *strong password*, namun sebanyak 52% responden ragu-ragu dapat mengingat password dengan metode OTP berbasis gambar ini dalam jangka waktu yang panjang. Hal ini disebabkan gambar kurang umum sehingga susah diingat. Sebuah penelitian menunjukkan bahwa visual memory sesorang dapat bersifat subjektif namun ada sesuatu yang membuat sebuah gambar mudah diingat [SDW-11]. Kemudian dalam tingkat kemudahannya, sebanyak 80% responden menyatakan bahwa OTP berbasis gambar mudah digunakan. Kemudian sebanyak 78% responden berpendapat bahwa OTP berbasis gambar ini merupakan ide yang bagus. Responden menyukai metode OTP berbasis gambar sebanyak 68%. Dan responden yang merasa aman dengan metode OTP berbasis gambar sebanyak 58%



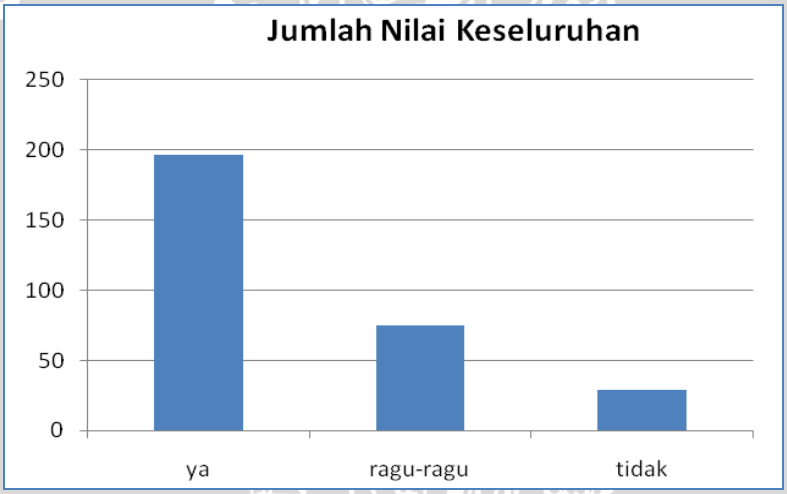
Gambar 5.8 Grafik jumlah nilai terhadap setiap pertanyaan

Selanjutnya terkait dengan jumlah pilihan gambar (pertanyaan nomor 5), sebesar 56% responden menyatakan bahwa 36 gambar cukup untuk menjadi alternatif pilihan *password*, dan sebesar 22% responden menyatakan bahwa 36

gambar terlalu banyak untuk menjadi alternatif pilihan *password*, sebesar 22% responden juga menyatakan bahwa 36 gambar terlalu sedikit.

Untuk pertanyaan "Menurut anda, berapa gambar yang bisa dijadikan alternatif pilihan *password* untuk pengguna?", dari 50 jawaban yang diberikan responden, diambil rata-rata yaitu sebesar 43 pilihan gambar. Namun dikondisikan dengan tata letak gambar pada halaman *website*, maka peneliti memutuskan untuk membulatkan ke bilangan kuadrat terdekat yaitu sebesar 49 gambar agar dapat di posisikan tujuh gambar mendatar dan tujuh gambar menurun.

Pada Gambar 5.8 merupakan jumlah keseluruhan dari tiap penilaian. Hasil dari penjumlahan nilai keseluruhan dapat diambil kesimpulan bahwa sebesar 65,3% responden dapat menerima sistem ini, sebesar 34,7% responden kurang dapat menerima sistem ini.



Gambar 5.9 Grafik jumlah keseluruhan dari tiap nilai

Secara umum responden merespon dengan baik metode OTP berbasis gambar ini. Para responden yang menyukai metode OTP berbasis gambar ini berpendapat bahwa metode ini merupakan inovasi yang menarik dan sesuatu yang visual lebih mudah diingat dibandingkan dengan sesuatu yang bersifat teks atau numerik. Meskipun demikian perlu diperhatikan bahwa dalam pengimplementasian OTP berbasis gambar ini masih ada pengguna yang belum dapat menerima karena beberapa alasan, diantaranya *password* tiap gambar saat *login* selalu berubah sehingga *user* perlu melihat satu persatu, gambarnya kurang umum sehingga susah untuk diingat dan masih belum terbiasa.

BAB VI
PENUTUP

6.1 KESIMPULAN

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. OTP berbasis gambar belum 100% aman digunakan untuk metode autentikasi pada *website ecommerce* PTIIK
2. OTP berbasis gambar pada *website ecommerce* PTIIK diimplementasikan dengan menggunakan bahasa pemrograman PHP dengan *framework Code Igniter* dan basis data MySQL.
3. Dari tiga kali proses *login* yang dilakukan oleh pengguna yang sama, *password* yang diinputkan oleh pengguna berbeda-beda sehingga *password* sebenarnya tidak dapat diketahui melalui man in the middle
4. Berdasarkan hasil pengujian serangan *brute force* dengan menggunakan *software Fireforce* dan *direct post username dan password* dengan menggunakan PHP Curl tidak berhasil menemukan *password* sebenarnya dari pengguna karena variabel yang diinisialisasi adalah *username* dan *password* di mana pada metode OTP berbasis gambar ini terdapat satu variabel lagi yaitu *id session*. Sehingga pada pengujian dengan *direct post username*, *password* dan *id session* menghasilkan bahwa *password* dapat ditemukan.
5. *Password* dari pengguna dapat ditemukan dengan catatan penyerang mengetahui metode yang digunakan oleh OTP berbasis gambar ini yaitu dengan menambahkan satu variabel *id session* dan tidak memuat ulang *form login* ketika melakukan proses *brute force* karena *id session* akan selalu berubah sehingga *password* yang ditemukan tidak dapat digunakan.
6. Berdasarkan kuesioner yang telah disebarkan, rata-rata permintaan jumlah pilihan gambar oleh responden adalah sebanyak 43 gambar. Untuk itu, peneliti membulatkan ke bilangan kuadat terdekat yaitu

sebesar 49 gambar agar dapat di posisikan tujuh gambar mendatar dan tujuh gambar menurun.

7. Dari hasil UAT, hasil yang kurang memuaskan terdapat pada pertanyaan "Apakah meskipun anda tidak menggunakan aplikasi ini dalam jangka waktu yang cukup lama, anda akan tetap mengingat password anda?". Hasil didapatkan sebanyak 52% responden ragu-ragu dapat mengingat password dengan metode OTP berbasis gambar ini dalam jangka waktu yang panjang. Hal ini disebabkan gambar kurang umum sehingga susah diingat.
8. Berdasarkan hasil analisis dari *user acceptance testing,* sebesar 65,3% pengguna merespon dengan baik metode OTP berbasis gambar ini karena mereka berpendapat bahwa metode ini merupakan inovasi yang menarik dan sesuatu yang berbentuk visual lebih mudah diingat dibandingkan dengan sesuatu yang bersifat teks atau numerik. Meskipun demikian perlu diperhatikan bahwa dalam pengimplementasian OTP berbasis gambar ini masih ada pengguna yang belum dapat menerima karena beberapa alasan, diantaranya *password* tiap gambar saat *login* selalu berubah sehingga user perlu melihat satu persatu, gambarnya kurang umum sehingga susah untuk diingat dan masih belum terbiasa.

6.2 SARAN

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain :

1. Untuk pengembangan lebih lanjut, disarankan ditambahkan metode keamanan seperti *security question* atau memblokir IP dari sebuah komputer yang telah melakukan lima kali gagal *login* untuk menghindari serangan *brute force*
2. Dalam menindaklanjuti hasil *user acceptance testing*, disarankan untuk pengembangan lebih lanjut menggunakan gambar-gambar yang mudah diingat contohnya gambar yang bersifat homogen.

3. Untuk pengembangan lebih lanjut, disarankan metode ini tidak hanya digunakan pada *web based application* tetapi juga *desktop application* atau *mobile application.*

DAFTAR PUSTAKA

[CDM-11] Ciampa, Mark. 2011. *Security+ Guide to Network Security Fundamentals, 4ed*. Diperoleh dari http://books.google.co.id/books?id=CIHYWBrg9JQC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

[FAG-11] Fajri, Ghozy. *[Tutorial-PDF] Packet Sniffing dengan Wireshark.* Diperoleh dari http://blog.uin-malang.ac.id/goji/2011/02/tutorial-pdf-paket-sniffing-dg-wireshark/

[FRF-13] www.scrt.ch/en/attack/downloads/fireforce diakses tanggal 1 Desember 2013

[GAR-10] Arumugam, G., and Sujatha, R. 2010. *Secured Authentication Protocol System Using Images*

[GEM-06] Gemalto. 2006. *One Time Password (OTP).* Diperoleh dari http://www.gemalto.com/techno/otp/

[HAK-13] http://hakipedia.com/index.php/CURL diakses tanggal 11 Januari 2013

[HAR-10] Hidayat, Ryan., Virgono, Agus., dan Usman, Koredianto. 2010. *Desain dan Implementasi Sistem Autentikasi dengan Graphical Password Berbasis Pixel Selection.*

[IAN-07] Sommerville, Ian. 2003. *Software Engineering Eight Edition.* China : China Machine Press

[MAX-13] Maxim, Bruce R. 2013. *Software Testing Strategies.* Diperoleh dari http://www.learningace.com/doc/2027567/0b918c897e643de4e7447f663a111bc7/lec25

[PRE-10] Pressman, Roger. 2010. *Software Engineering: A Practioner's Approach, 7th Edition*. Mc Graw-Hill.

[RFD-11] Firdaus, Rangga., Kurniawan, Didik., dan Simamora, Erwin. 2011. *Implementasi Metode Autentikasi One Time Password (OTPA) Berbasis Mobile Token pada Aplikasi Ujian Online (Studi Kasus: Jurusan Matematika FMIPA UNILA)*

[RIS-12] Raza, Mudassar., Iqbal, Muhammad., Sharlf, Muhammad., Halder,

Waqas. 2012. *A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication*

[ROL-13] http://www.rolo.org/brute-force-attack.html diakses tanggal 1 Desember 2013

[SDW-11] http://www.sciencedaily.com/releases/2011/05/110524104653.htm diakses tanggal 10 Januari 2013

[SRN-67] Shepard, Roger. 1967. *Recognition Memory for Words, Sentences, and Pictures*.

[TAZ-10] Tyas, Zahra. 2010. *Implementasi E-commerce untuk Ozone Distro*

LAMPIRAN

Lampiran 1 Kuesioner *user acceptance testing* untuk pengguna

Kuesioner PTIIKshop

Pertanyaan-pertanyaan di bawah ini berkaitan dengan persepsi anda terhadap metode One Time Password berbasis gambar yang digunakan saat autentikasi pada website ecommerce ini. Mohon pilih atau tulis jawaban yang anda anggap paling tepat mencerminkan persepsi anda

1. Apakah mengingat password dengan menggunakan One Time Password berbasis gambar lebih mudah diingat daripada password dengan menggunakan strong password? **(strong password adalah password yang mempunyai panjang minimal 8 karakter, mengandung karakter abjad huruf kecil dan kapital, memiliki setidaknya satu karakter numerik, memiliki setidaknya satu karakter khusus misalnya ~ @ # $% ^ & * () - _ + = , tidak boleh mengeja kata atau rangkaian kata-kata yang dapat ditemukan dalam kamus standart, tidak boleh mengeja kata dengan angka ditambahkan ke awal dan / atau akhir, tidak didasarkan pada informasi pribadi seperti tanggal ulang tahun, tidak didasarkan pada pola keyboard atau duplikasi karakter)*

   a. Ya b. Ragu-ragu c. Tidak
2. Apakah meskipun anda tidak menggunakan aplikasi ini dalam jangka waktu yang cukup lama, anda akan tetap mengingat password anda?

   a. Ya b. Ragu-ragu c. Tidak
3. Menurut anda, apakah sistem autentikasi menggunakan One Time Password mudah digunakan?

   a. Ya b. Ragu-ragu c. Tidak
4. Menurut anda, apakah sistem autentikasi menggunakan One Time Password merupakan ide yang bagus?

   a. Ya b. Ragu-ragu c. Tidak
5. Menurut anda, apakah tiga puluh enam gambar cukup untuk menjadi alternatif pilihan password?

   a. Cukup b. Terlalu Banyak c. Terlalu Sedikit

6. Menurut anda, berapa gambar yang bisa dijadikan alternatif pilihan password untuk pengguna?

   _____

7. Apakah anda menyukai metode One Time Password berbasis gambar daripada metode password dengan menghafal karakter-karakter seperti biasanya?

   a. Ya b. Ragu-ragu c. Tidak

8. Sebutkan alasan anda mengapa menyukai atau tidak menyukai metode One Time Password berbasis gambar ini?

   _____

   _____

   _____

9. Apakah anda lebih merasa aman memiliki password dengan metode One Time Password berbasis gambar dibandingkan dengan memiliki strong password?

   a. Ya b. Ragu-ragu c. Tidak

Pertanyaan Tambahan

10. Kategori produk apa yang anda inginkan untuk dijual di website ecommerce ini?

    _____

    _____

    _____