**APLIKASI MARKETPLACE BARANG BEKAS**

**BERBASIS WEB MENGGUNAKAN**

**IONIC DAN MARIADB**

**PROPOSAL PROYEK AKHIR**

Oleh :  
**Adrian Priabdi Fauzi (4311711001)**

**Fajar Widiarto (4311711005)**

**Indra Rianto Saputra (4311711006)**

**Indah Audina (4311711029)**

Disusun untuk pengajuan proposal Proyek Akhir mata kuliah Pemrograman Perangkat Bergerak



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK MULTIMEDIA DAN JARINGAN  
POLITEKNIK NEGERI BATAM  
BATAM**

**2019**

**Abstrak**

Aplikasi merupakan penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentaun dengan Bahasa pemerograman tertentu. Barang bekas merupakan suatu barang yang telah digunakan oleh pemiliknya dan tidak digunakan atau dibutuhkan lagi tetapi masih layak untuk digunakan. Aplikasi marketplace barang bekas adalah aplikasi berbasis web dan native atau bisa disebut dengan hybrid yang dapat dijalankan baik menggunakan smartphone android maupun smartphone IOS untuk mencari dan menawarkan barang bekas kepada user. Aplikasi ini dapat membantu pengguna atau konsumen dalam mencari barang bekas diinginkan dan sesuai kebutuhan. Aplikasi ini memiliki beberapa fitur seperti find item, item with categories, dan management system. Selain itu terdapat pula fitur pendukung seperti Compare, On TKP, dan feedback user.

**Kata kunci :** barang bekas, marketplace, aplikasi marketplace barang bekas, android, ios, hybrid.

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Beberapa tahun belakangan ini, sampah merupakan masalah yang sangat sulit diatasi karena sampah semakin hari semakin bertambah. Sampah bukan hanya tidak berguna tetapi ada juga yang masih berguna, seperti barang – barang bekas . Barang bekas merupakan barang yang memiliki tingkat kelayakan yang masih baik dan bisa digunakan. Banyak manusia memiliki barang bekas tetapi kebanyakan mereka akan membuangnya kedalam tempat sampah atau menyimpannya didalam gudang. Sehingga membuat beberapa tempat seperti rumah atau kantor menjadi berantakan dan tidak tertata. Dengan adanya barang bekas maka ada beberapa manusia yang sangat ingin mengambil atau mengolah barang bekas tersebut agar tidak menjadi sampah yang membuat penyemaran lingkungan semakin buruk. Akan tetapi banyak manusia yang tidak tahu dimana dan bagaimana mengambil barang bekas tersebut sehingga kebanyakan manusia akan membeli dari pemulung, mencari ketempat jual beli besi tua bahkan mencari – cari menggunakan media sosial.

Dengan kemajuan teknologi dan semakin banyaknya manusia memiliki smartphone dari kalangan bawah maupun kalangan atas, Kami mengusulkan untuk pembuatan sebuah aplikasi yang dapat memberi tempat kepada pemilik barang bekas dan pencari barang bekas agar dapat mendapatkan barang bekas yang diinginkan, berkomunikasi dan saling terhubung agar memberikan keuntungan bagi kedua belah pihak. Selain itu dengan aplikasi ini maka Kita akan ikut berpartisipasi dalam mengurangi sampah dan menjaga lingkungan agar tetap baik dengan mengolah barang – barang bekas tersebut. Sehingga ide Kami adalah untuk membuat sebuah aplikasi marketplace barang bekas yang berbasis Ionic dan MariaDB agar dapat membantu dalam menyelesaikan masalah lingkungan yang terjadi saat ini.

1. **Rumusan Masalah**

Rumusan masalah yang dimiliki adalah :

1. Bagaimana membuat aplikasi marketplace yang menjual atau menawarkan barang bekas sebagai tempat untuk mempermudah dalam mencari barang bekas yang diinginkan oleh pencari barang bekas?
2. Bagaimana cara menghubungkan antara pemilik barang bekas dengan pencari barang bekas ?
3. **Batasan Masalah**

Agar penelitian tidak terlalu luas cakupannnya, maka diperlukan batasan – batasan masalah pada proyek akhir ini, yaitu sebagai berikut :

1. Pemilik barang bekas dapat menambahkan barang – barang apa saja yang bisa ditawarkan kepada pencari barang bekas.
2. Pencari barang bekas dapat mencari barang – barang bekas yang diinginkan sesuai kategori yang tersedia pada aplikasi.
3. Aplikasi ini masih dalam sebatas mempertemukan antara pencari barang bekas dengan pemilik barang bekas.
4. **Tujuan**

Tujuan dari proyek akhir ini adalah untuk merancang dan membangun aplikasi yang dapat menyelesaikan rumusan masalah, yaitu :

1. Menerapkan aplikasi marketplace barang bekas sebagai tempat yang mempermudah pencari barang bekas untuk mendapatkan barang yang diinginkan.
2. Mempermudah dalam menghubungkan antara pemilik barang bekas dengan pencari barang bekas.
3. **Manfaat**

Dari tujuan diatas, dapat diambil beberapa manfaat, yaitu :

1. Mempermudah pencari barang bekas dalam mencari barang bekas yang diinginkan.
2. Sebagai tempat pertama dalam mencari barang bekas yang diinginkan.
3. Para pemilik barang bekas dapat menawarkan dan menjual barang bekas tersebut agar mengurangi sampah – sampah baik di rumah maupun di kantor.
4. Ikut berpartisipasi dalam mengurangi sampah bagi lingkungan.

**BAB II**

**LANDASAN TEORI**

1. **Aplikasi Mobile**

Aplikasi menurut kamus Besar Bahasa Indonesia (KBBI) adalah penerapan rancang sistem untuk mengolah data menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Berarti aplikasi adalah program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Menurut Ibisa, aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya.

Dari penjelasan diatas dapat disimpulkan bahwa aplikasi mobile adalah suatu program yang berjalan pada perangkat mobile dengan menggunakan Bahasa pemrograman tertentu untuk membantu dan melaksanakan tugas khusus dari pengguna.

1. **Jenis Aplikasi Mobile**

Aplikasi Native Aplikasi native adalah aplikasi yang dibangun dengan bahasa pemrograman spesifik untuk platform tertentu. Contoh populernya yaitu penggunaan Objective-C atau Swift untuk platform iOS (Apple). Adapun platform Android yang menggunakan bahasa pemrograman Java atau Kotlin. Membangun aplikasi native harus menyediakan pengalaman produk yang optimal pada perangkat mobile. Meskipun begitu membutuhkan biaya yang tinggi untuk membangun aplikasi lintas platform dan tetap mempertahankan aplikasi native agar tetap terupdate.

Aplikasi web adalah aplikasi yang dibangun dengan standar dan berbasis web yang diakses menggunakan web browser. Aplikasi web bisa menjadi pilihan yang terbaik untuk membangun produk yang diinginkan. Karena benar-benar minim persyaratan, dengan kata lain akses ke  
fitur dari perangkat mobile seperti Push Notification tidak diperlukan. Aplikasi web bisa menjadi pilihan yang murah dari sisi budget. Namun sayangnya, aplikasi web tidak bisa didistribusikan melalui toko aplikasi native seperti App Store atau Google Play.

Aplikasi hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android. Aplikasi hybrid biasanya menggunakan browser untuk mengijinkan aplikasi web mengakses berbagai fitur di device mobile seperti Push Notification, Contacts, atau Offline Data Storage. Beberapa tools untuk mengembangkan aplikasi hybrid antara lain Phonegap, Rubymotion dan lain-lain. Keuntungan membangun aplikasi hybrid diantaranya pemeliharaan project menjadi semakin mudah jika dibandingkan dengan aplikasi  
native. Aplikasi hybrid juga, bisa dibangun secara cepat untuk keperluan cross platform dan  
dana yang bisa menjadi lebih hemat jika dibandingkan dengan native.

1. **Framework Ionic**

Ionic adalah framework yang dikhususkan untuk membangun aplikasi mobile hybrid dengan HTML5, CSS dan AngularJS. Ionic menggunakan Node.js SASS, AngularJS sebagai engine-nya. Ionic dilengkapi dengan komponen-komponen CSS seperti button, list, card, form, grids, tabs, dan masih banyak lagi. Jadi Ionic itu merupakan teknologi web yang bisa digunakan untuk membuat suatu aplikasi mobile. Karena hybrid maka aplikasi hanya dibuat 1 kali tetapi sudah bisa dirilis di lebih dari 1 platform alias cross-platform.

1. **MariaDB**

MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB.

**BAB III**

**ANALISIS DAN PERANCANGAN**

1. **Metode Penyelesaian Masalah**
2. **Breadth Search First**

Breadth-first search (BFS) adalah salah satu algoritma pencarian/penelusuran dengan cara traversal melebar pada sebuah graf atau pohon ruang status. BFS mengawali penelusuran dengan cara mengunjungi suatu simpul pada graf (atau pohon) status. Kemudian, BFS mengunjungi semua simpul yang bertetangga dengan simpul tersebut secara preorder. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya. Jika graf berbentuk pohon berakar, maka semua simpul pada aras d dikunjungi lebih dulu dari simpul-simpul pada aras d+1. Atau dalam kata lain, BFS akan melakukan pencarian secara melebar terlebih dahulu (breadth-first).

Dalam penggunaannya, algoritma ini memerlukan sebuah antrian q untuk menyimpan simpul-simpul yang telah dikunjungi. Antrian ini diperlukan sebagai acuan dalam memilih simpul-simpul yang akan dikunjungi berikutnya, yaitu simpul-simpul yang bertetanggaan dengan simpul yang sudah ada di antrian. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan tabel boolean untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali.

Urutan algoritma dari Breadth Search Firsh adalah masukkan simpul root ke dalam antrian, kemudian periksa antrian terdepan apakah memiliki anak simpul, jika ya, masukan semua anak simpul ke dalam antrian dan hapus antrian terdepan, jika antrian kosong berhenti, tapi jika tidak kembali ke langkah dua

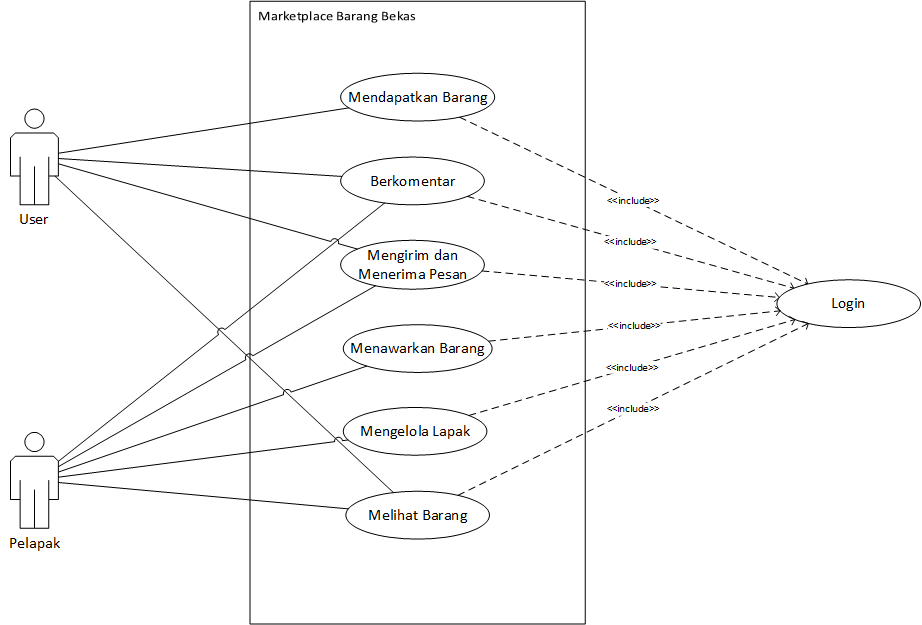
1. **Depth Search First**

Depth-first search (DFS) adalah algoritma pencarian/penelusuran traversal mendalam pada sebuah graf atau pohon ruang status. DFS mengawali penelusuran dengan cara mengunjungi suatu simpul pada graf (atau pohon) status. Kemudian, DFS mengunjungi simpul tetangga dari simpul tersebut yang dipilih secara preorder, terus menerus ke simpul tetangga dari simpul tetangga tadi secara rekursif sampai dengan simpul tetangga yang dikunjungi tidak memiliki tetangga lagi. Kemudian, DFS akan melakukan backtrack-ing ke simpul sebelumnya.

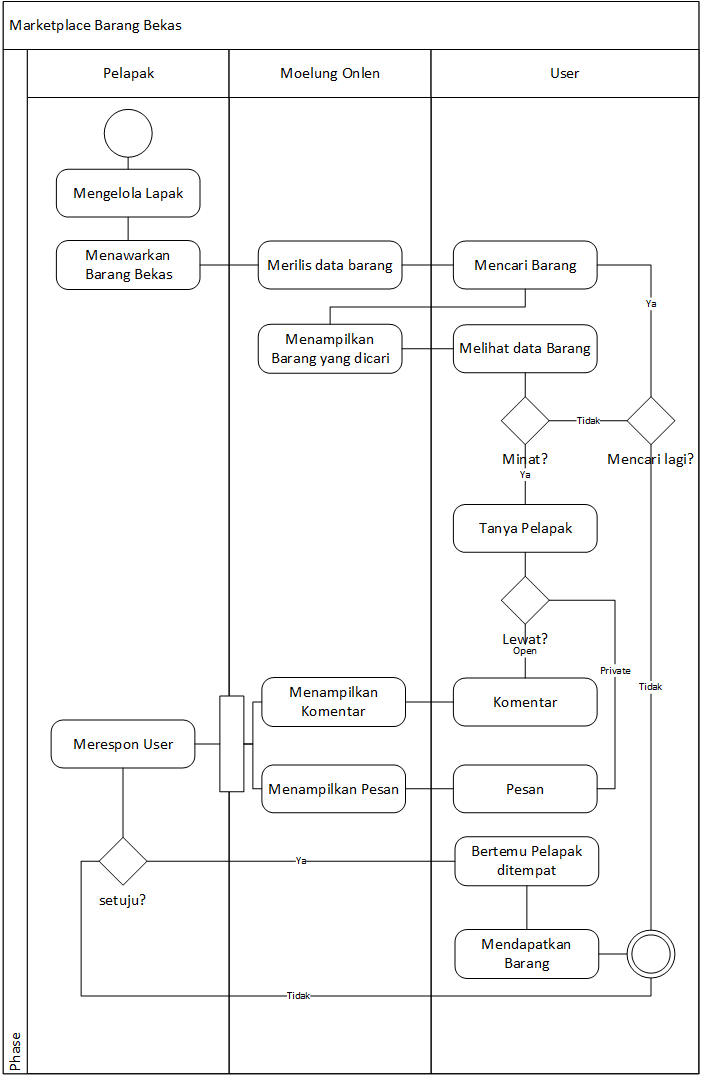
Jika simpul ini masih memiliki tetangga yang belum dikunjungi, DFS akan kembali melakukan pencarian rekursif pada tetangga tersebut sampai menemukan simpul yang tidak memiliki tetangga lagi. Jika simpul ini tidak memiliki anak yang belum dikunjungi, DFS akan melanjutkan backtrack-nya ke simpul sebelumnya lagi secara terus menerus dan rekursif sampai ditemukan sebuah simpul yang masih memiliki tetangga, dan kemudian melakukan pencarian rekursif pada simpul tetangga tersebut. Jika graf berbentuk pohon berakar, maka semua simpul pada aras d dikunjungi lebih dulu dari simpul-simpul pada aras d+1. Atau dalam kata lain, DFS akan melakukan pencarian secara mendalam terlebih dahulu (depth-first). Algoritma ini dapat diimplementasikan menggunakan stack atau secara rekursif.

Urutan Algoritm DFS dengan menggunakan stack. Langkah – langkahnya adalah masukkan simpul root ke dalam tumpukan dengan push. Lalu mmbil dan simpan isi elemen (berupa simpul pohon) dari tumpukan teratas. Kemudian hapus isi stack teratas dengan prosedur pop. Setelah itu periksa apakah simpul pohon yang disimpan tadi memiliki anak simpul. Jika ya, push semua anak simpul yang dibangkitkan ke dalam stack dan jika tumpukan kosong berhenti, tapi jika tidak kembali ke langkah dua.

1. **Gambaran Umum Perangkat Lunak**
2. **Use Case**

****

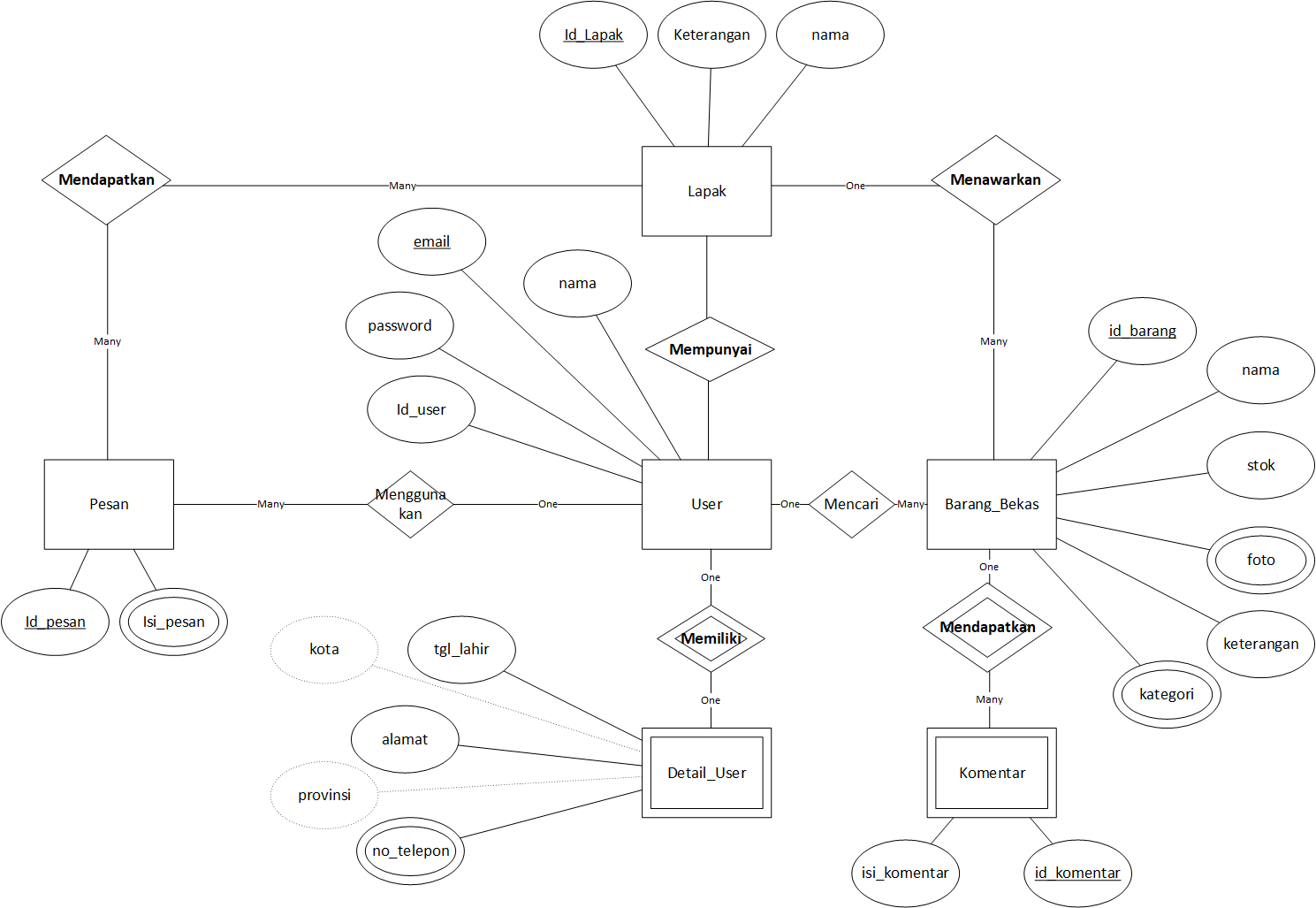
1. **Activity**

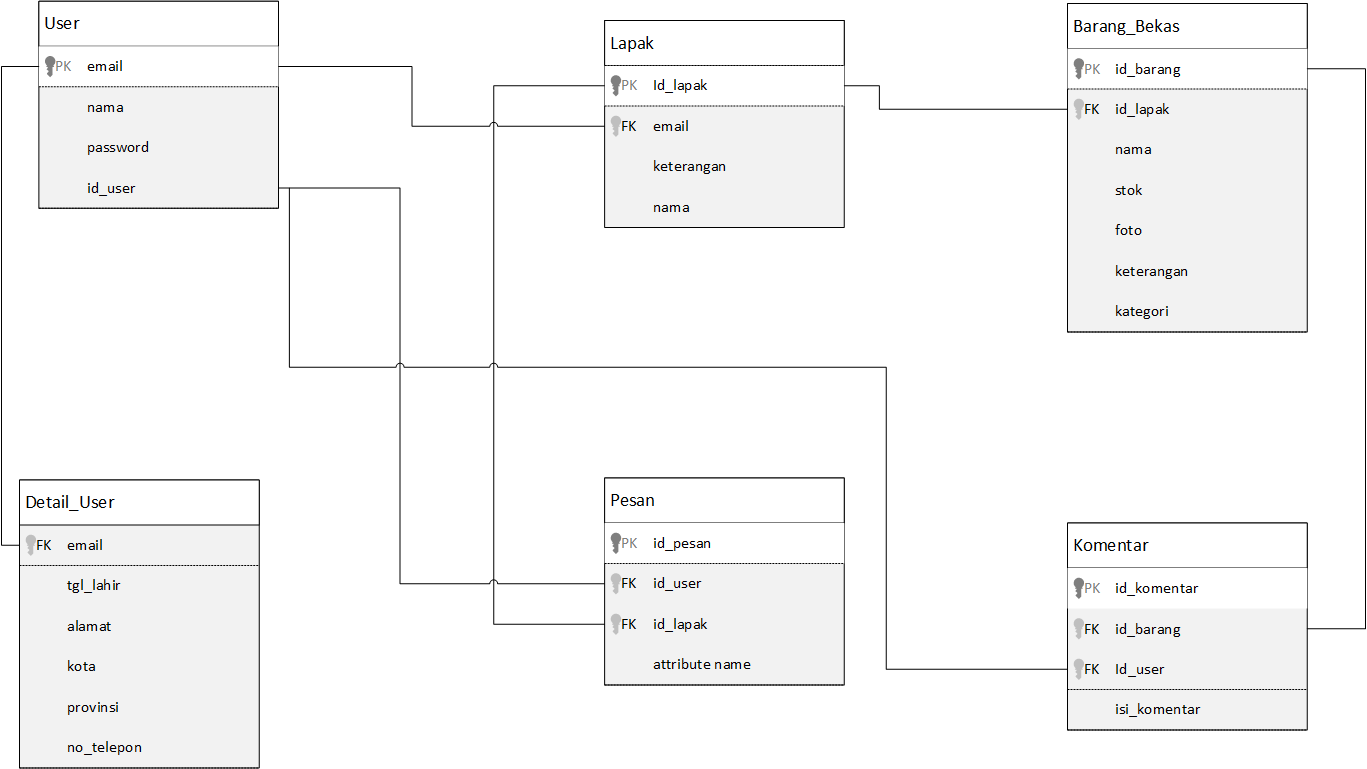
****

1. **Kebutuhan Fungsional**

|  |  |
| --- | --- |
| **Kode Fungsional** | **Penjelasan** |
| F001 | Menampilkan barang – barang bekas dari user pelapak. |
| F002 | Memasukkan komentar pada barang – barang yang ditawarkan oleh pelapak. |
| F003 | Mengirim pesan antar user dengan user pelapak. |
| F004 | Menampilkan keterangan barang bekas. |
| F005 | Menampilkan keterangan lapak. |
| F006 | Mengelola lapak. |
| F007 | Mengelola user. |

1. **Perancangan Basis Data**

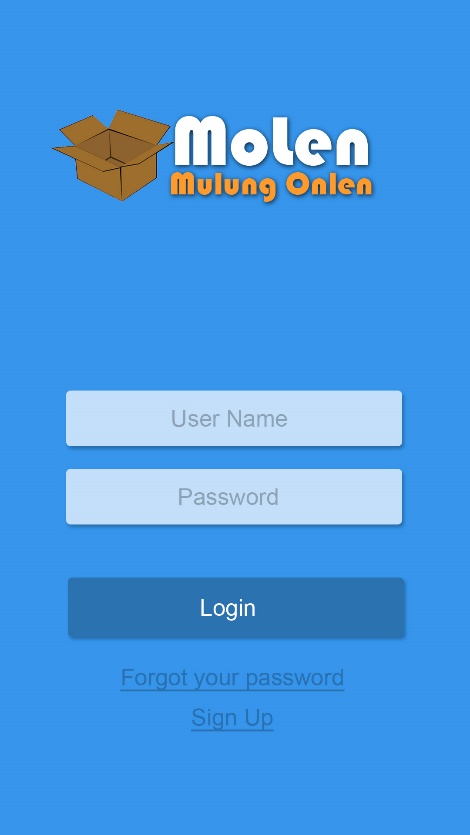
****

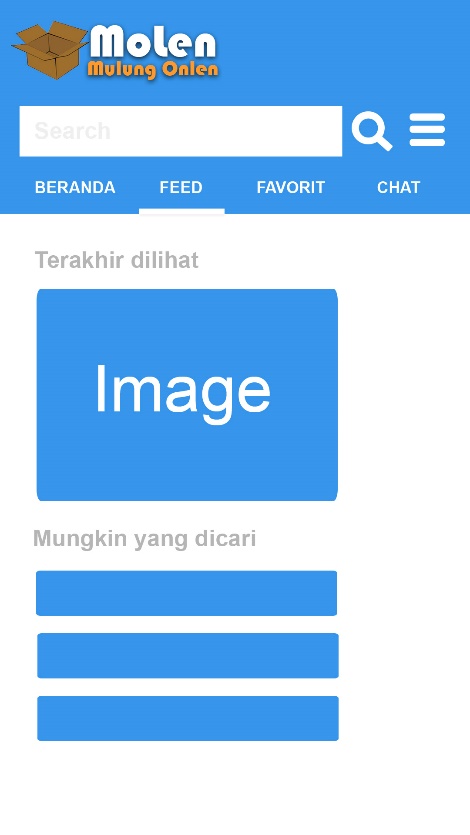
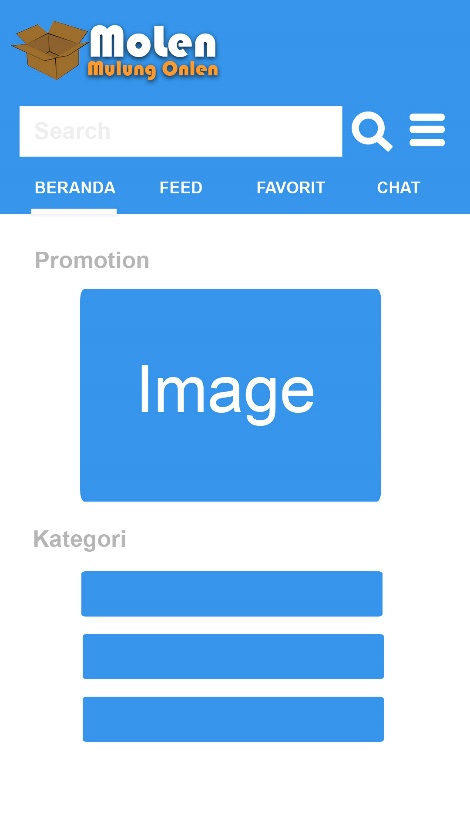
****

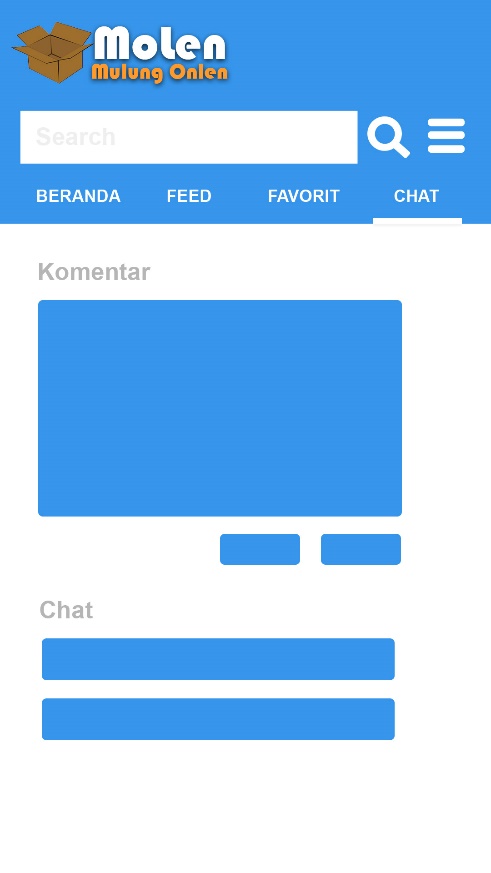
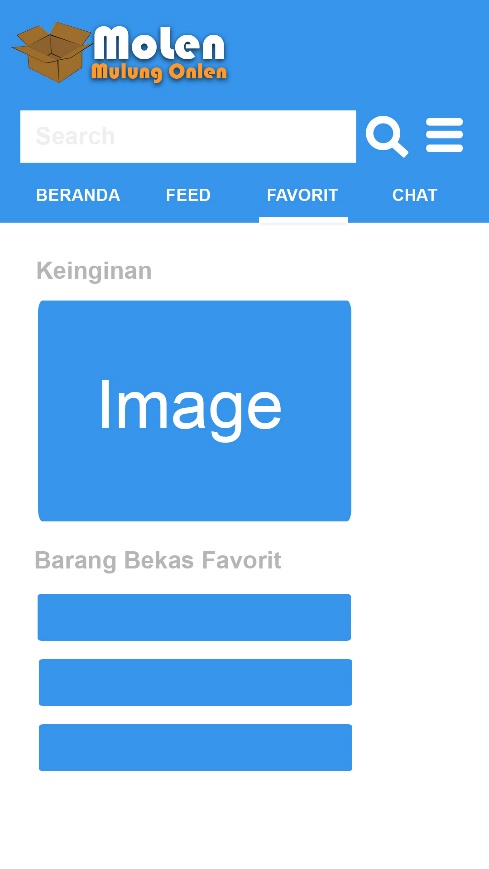
**BAB IV**

**IMPLEMENTASI DAN PERENCANAAN**

1. **Antarmuka Perangkat Lunak**

****

****

****

1. **Rencana Pelaksanaan**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kegiatan** | **Januari** | | | | **Februari** | | | | **Maret** | | | | **April** | | | | **Mei** | | | | **Juni** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Identifikasi Masalah |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penentuan Topik/ Judul |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Landasan Teori |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis dan Perancangan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi dan Uji Coba |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perilisan Aplikasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**DAFTAR PUSTAKA**

* https://ionicframework.com
* https://mariadb.org/learn/
* https://stackoverflow.com
* Permana, Yana, 2016, Ini Dia Perbedaan Aplikasi Native, Hybrid atau Web  
  [Online] ,(www.codepolitan.com/apa-bedanyaaplikasi-native-hybrid-dan-web, access on December, 7 2015).
* Zulva Fachrina.2015.Pencarian Barang Dalam Basis Data Online Shop Menggunakan Algoritma BFS, DFS, dan KMP. Institut Teknologi Bandung, 4 Mei
* http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Makalah2015/Makalah\_IF221\_Strategi\_Algoritma\_2015\_055.pdf

**Progress Pengerjaan Proyek**

|  |  |
| --- | --- |
| **Minggu 10** | **1 April 2019** |
| **Kegiatan** | **Hasil Kegiatan** |
| Pada minggu ini, Kami mencoba mengimplementasikan rancangan aplikasi kedalam proyek aplikasi. | Hasil dari kegiatan ini adalah Kami telah berhasil mengimplementasikan rancangan kedalam aplikasi Kami. Hanya saja masih banyak kekurangan dan bug – bug yan terjadi. Pada minggu ini, Kami baru menyelesaikan beberapa fitur seperti login, register, pengelolaan user, dan pengelolaan data barang. Untuk fitur seperti upload foto, komentar, chat, dan security aplikasi, Kami masih belum mengimplementasikan kedalam aplikasi Kami.  Link Project **: https://github.com/adrianpfauzi/molen-apps.git**  Beberapa screenshot aplikasi Kami : |

|  |  |
| --- | --- |
| **Minggu 11** | **7 April 2019** |
| **Kegiatan** | **Hasil Kegiatan** |
| Pada minggu ini, Kami masih memperbaiki fitur – fitur seperti login ke sistem, pengelolaan user untuk lihat profile dan edit profile. Selain itu masih mencoba mencari cara untuk menanamkan fitur komentar pada setiap barang yang ditawarkan | Setelah beberapa hari, Kami masih belum menemukan cara yang bagus dan baik untuk membuat fitur pengelolaan user. Kami sudah menemukan API, tetapi pengambilan data user masih didalam array json, bukan json. Sehingga ketika user mengupdate profile, maka yang terjadi adalah data yang dikirimkan ke server adalah null. Selain itu untuk fitur komentar, masih belum menemukan cara yang pas dan membuat komentar berjalan secara real time. Solusi pertama masih menggunakan halaman lain melalui komponen button. Tetapi cara tersebut terkesan ribet. Sehingga kami masih mencari cara lain agar komentar dapat ditampilkan didalam postingan barang. Untuk fitur login, kami masih mencari cara yang sesuai untuk menangani autentikasi ke aplikasi. Karena kami masih memiliki bug yang sangat fatal, yaitu ketika user tidak ada dalam database, maka tetap bisa login ke sistem.  Link Project **: https://github.com/adrianpfauzi/molen-apps.git** |