RANCANG BANGUN APLIKASI PENCARIAN DAN PENAWARAN BUKU DENGAN MENERAPKAN *RESPONSIVE WEB DESIGN*

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh:
AFNI WILUJENG SETYORINI
NIM 115060801111022

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
MALANG
2015

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LEMBAR PERSETUJUAN

RANCANG BANGUN APLIKASI PENCARIAN DAN PENAWARAN BUKU DENGAN MENERAPKAN *RESPONSIVE WEB DESIGN*

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh:
AFNI WILUJENG SETYORINI
NIM 115060801111022

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Denny Sagita R., S.Kom, M.Kom	**Aryo Pinandito, S.T, M.MT**
NIK. 851124 06 1 1 0250	NIP. 198305192014041001

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI PENCARIAN DAN PENAWARAN BUKU DENGAN MENERAPKAN *RESPONSIVE WEB DESIGN*

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :

AFNI WILUJENG SETYORINI
NIM 115060801111022

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 10 Juli 2015

Penguji I,	**Penguji II,**
Agi Putra Kharisma, S.T, M.T	**Fajar Pradana, S.ST, M.Eng**
NIK. 2013048604301001	**NIK. 871121 16110371**

Penguji III,

Issa Arwani, S.Kom, M.Sc
NIP. 19830922 201212 1 003

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, 01 Juli 2015

Mahasiswa

AFNI WILUJENG SETYORINI
NIM. 115060801111022

ABSTRAK

Afni Wilujeng Setyorini, 2015. Rancang Bangun Aplikasi Pencarian dan Penawaran Buku Dengan Menerapkan Responsive Web Design. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Dosen pembimbing : Denny Sagita S.Kom, M.Kom dan Aryo Pinandito ST,M.MT

Perpustakaan sebagai tempat untuk menyimpan buku-buku atau bahan pustaka yang dapat digunakan masyarakat untuk mengakses berbagai sumber informasi lewat buku. Namun, masyarakat masih kesulitan mendapatkan bahan pustakan diperpustakaan dikarenakan koleksi buku yang tidak lengkap dan jumlah buku yang terbatas. Selain itu, masyarakat juga kesulitan mencari bahan pustaka di toko buku dikarenakan buku yang dibutuhkan sudah tidak terbit atau sudah tidak dijual belikan di toko buku. Salah satu alternatif untuk mengatasi permasalahan tersebut bisa dilakukan dengan menawarkan buku-buku yang kita miliki atau buku-buku yang sudah tidak digunakan bisa kita tawarkan kepada orang lain sehingga menjadi lebih bermanfaat. Tidak adanya media informasi yang dapat melakukan pencarian dan penawaran buku membuat seseorang kesulitan mencari informasi lokasi orang yang mempunyai buku yang kita butuhkah. Pada aplikasi ini, pengguna dapat melakukan pencarian penawaran buku yang berada di dekat area lokasi user berdasarkan jarak lokasi terdekat beserta rute lokasi yang ditampilkan melalui Google Maps API dengan memanfaatkan teknologi Geolocation dan Haversine Formula. Aplikasi ini dibangun berbasis web dengan menerapkan Responsive Web Design (RWD). RWD akan membantu pengguna untuk mengakses aplikasi secara optimal pada berbagai perangkat (*smartphone, tablet, desktop*) dan lintas *platform* dengan berbagai jenis ukuran layar. Implementasi aplikasi menggunakan PHP, javascript, HTML5 dan CSS3. Hasil pengujian menujukan penerapan teknologi *Geolocation* dan persamaan Haversine Formula mampu menentukan lokasi terdekat dengan membandingkan hasil perhitungan jarak pada aplikasi dengan jarak sebenarnya memiliki kesalahan rata-rata 0,47 kilometer atau 470 meter dan masing-masing kasus uji memiliki akuransi kesalahan kurang dari 0.5 persen (%). Berdasarkan hasil pengujian layout responsif dan pengunjian *compatibility*, Aplikasi Pencarian dan Penawaran Buku dapat beradaptasi secara optimal dalam berbagai jenis perangkat (*smartphone*, *tablet*, *laptop*) dengan berbagai ukuran layar dan kompatibel terhadap berbagai jenis browser pada perangkat bergerak maupun perangkat *desktop*.

Kata kunci: penawaran buku, pencarian buku, *responsive web design,* teknologi *geolocation*

ABSTRACT

Afni Wilujeng Setyorini, 2015. ***Building Design of Searching and Offering Book Application by Implementing Responsive Web Design.*** *Information Technology and Computer Science Program, Brawijaya University, Malang. Advisors*: Denny Sagita S.Kom, M.Kom dan Aryo Pinandito ST,M.MT

Library as the place for keeping books or literary sources which can be used to look up any information needed. However, people are, yet, having difficulties to get hold of literary sourcess in library because of several reasons such as incompleteness of books collection and restrictiveness the number of books available in the library. Besides, sometimes its difficult for people to look for literary sources in bookstores either because the books required have not been published or have not sold anymore. One alternative to overcome these problems can be done by offering books that we have or unused books to others so that it becomes more useful. The absence of media information in attempt to search book make people having trouble in finding information about the location of people possessing books we need. In this application, users can conduct book offering search in the nearby users' location based on distance route location and its nearby locations displayed through Google Maps API by using Geolocation Technology and Haversine Formula. This web-based application is built by applying Responsive Web Design (RWD). RWD indeed, will help the users to access the application optimally on a variety of devices (smartphones, tablets, desktops) and cross-platform with various types of screen size. PHP, JavaScript, HTML5 and CSS3 are used in the implementation of this application. The test results shows that the implementation of Geolocation technology and Haversine equation formula are able to determine the nearest location by comparing the calculation results between distance showed by the application and the actual distance in which has an average error of 0.47 kilometers or 470 meters. In each test case has error accuracy of less than 0.5 percent (%). Based on the results of responsive layout and compatibility, Book Search and Offer application is optimally adaptable into different types of devices (smartphone, tablet, and laptop) with a variety of screen sizes and is compatible into several types of browsers on mobile devices and desktop devices.

Keywords: *book offer, book search, responsive web design, geolocation technology.*

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Skripsi dengan judul *"*Rancang Bangun Aplikasi Pencarian dan Penawaran Buku Dengan Menerapkan Responsive Web Design*"*. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

1. Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan – bantuan baik lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :Bapak Denny Sagita Rusdiantoro, S.Kom., M.Kom dan Bapak Aryo Pinandito, S.T., M.MT selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Ibu Purwanti, ayah Suparjo dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti – hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
3. Bapak Sutrisno, Ir., MT, selaku Ketua Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, dan Drs. Mardji, MT, selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya.
4. Bapak R. Arief Setyawan, ST., MT selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
5. Seluruh Dosen Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
6. Teman-teman TIF-C 2011 dan teman-teman Program Studi Informatika/Ilmu Komputer angkatan 2011, terimakasih atas segala bantuannya dan kesan yang diberikan selama menempuh studi di Teknik Informatika Universitas Brawijaya.

7. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun yang tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, Juni 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL 10
LEMBAR PERSETUJUAN 10
LEMBAR PERNGESAHAN i10
PERNYATAAN ORISINALITAS SKRIPSI i
ABSTRAK 10
ABSTRACT vi
KATA PENGANTAR 10
DAFTAR ISI 10
BAB I 10
PENDAHULUAN 10
1.1 Latar Belakang 10
1.2 Rumusan Masalah 10
1.3 Batasan Masalah 10
1.4 Tujuan Penelitian 10
1.5 Manfaat Penelitian 10
1.6 Sistematika Penulisan 10
BAB II DASAR TEORI 10
2.1 Kajian Pustaka 10
2.2 Buku dan Minat Baca 10
2.3 Google Maps 10
2.4 Geolocation 10
2.6 Responsive Web Design 10
2.6.1 Flexible Grids 10
2.6.2 Flexible Image and Media 10
2.6.3 Media Queries 10
2.7 Konsep MVC 10
2.8 Haversine Formula 10
2.9 Rekayasa Perangkat Lunak 10
BAB III METODOLOGI 10
3.1 Metode Pengembangan 10
3.2 Studi Literatur 10
3.3 Analisis Kebutuhan 10

3.4 Perancangan Perangkat Lunak 10
3.5 Implementasi 10
3.6 Pengujian Perangkat Lunak 10
3.7 Pengambilan Kesimpulan 10
BAB IV ANALISIS DAN PERANCANGAN 10
4.1 Analisis Kebutuhan Sistem 10
4.1.1 Gambaran Umum Aplikasi 10
4.1.2 Identifikasi Aktor 10
4.1.3 Analisis Kebutuhan Fungsional 10
4.1.4 Analisis Kebutuhan Non Fungsional 10
4.2 Perancangan Perangkat Lunak 10
4.2.1 Diagram Use Case dan Activity 10
4.2.2 Sequence Diagram 10
4.2.3 Class Diagram 10
4.2.4 Perancangan Arsitektur Sistem 10
4.2.5 Perancangan Basis Data 10
4.2.6 Perancangan Layout Responsif 10
4.2.7 Perancangan Antarmuka 10
4.3 Perancangan Pengujian 10
4.3.1 Lingkungan Pengujian Perangkat Lunak 10
4.3.2 Identifikasi dan Rencana Pengujian 10
4.3.3 Perancangan Kasus Uji 10
BAB V IMPLEMENTASI 10
5.1 Spesifikasi Sistem 10
5.2 Batasan-batasan Implementasi 10
5.3 Implementasi Basis Data 10
5.4 Implementasi Class 10
5.5 Implementasi Kode Program 10
5.6 Implementasi Layout Responsif 10
5.7 Implementasi Antarmuka Aplikasi 10
5.7.1 Implementasi Antarmuka halaman utama 10
5.7.2 Implementasi Antarmuka Pencarian 10
5.7.3 Implementasi Antarmuka Posting Penawaran Buku 10

5.7.4 Implementasi Antarmuka Request Penawaran Buku 105
BAB VI PENGUJIAN DAN ANALISIS 10
6.1 Hasil Pengujian 10
6.1.1 Pengujian Validasi 10
6.1.2 Pengujian Layout Web Responsif 10
6.1.3 Pengujian Haversine Formula 10
6.1.4 Pengujian Compatibility 10
5.2 Analisis Pengujian 10
5.2.1 Analisis Pengujian Validasi 10
5.2.2 Analisis Pengujian Layout Responsive 10
5.2.3 Analisis Pengujian Jarak Lokasi 10
5.2.4 Analisis Pengujian *Compatibility* 10
BAB VII PENUTUP 10
7.1 Kesimpulan 10
7.2 Saran 10
DAFTAR PUSTAKA 10
Lampiran 1. Skenario Use Case Kofirmasi Request Masuk 10
Lampiran 2. Skenario Use Case Mengirimkan Pesan 10
Lampiran 3. Skenario use case mengubah status penawaran buku 10
Lampiran 4. Activity Diagram konfirmasi Request Masuk 10
Lampiran 5. Activity Diagram Mengirimkan Pesan 10
Lampiran 6. Activity Diagram Mengubah Status Penawaran Buku 10
Lampiran 7 Kasus Uji Validasi Konfirmasi Request Masuk 10
Lampiran 8. Kasus Uji Validasi Kirim Pesan 10
Lampiran 9. Kasus Uji Validasi Mengubah Status Penawaran Buku 10

DAFTAR GAMBAR

Gambar 2.1 Tampilan Google Maps 10
Gambar 2.2 Hasil implementasi Geolocation API 10
Gambar 2.3 Tampilan website yang menerapkan RWD 10
Gambar 2.4 Contoh penerapan flexibe image 10
Gambar 2.5 Contoh penerapan media queries 10
Gambar 2.6 Konsep MVC 10
Gambar 2.7 Diagram alir pemodelan waterfall 10
Gambar 3.1 Diagram alir metodelogi penelitian 10
Gambar 3.2 Diagram alir metode pembangunan aplikasi 10
Gambar 4.1 Gambaran Umum Sistem 32
Gambar 4.2 Diagram *use case* 10
Gambar 4.3 *Activity diagram* pencarian buku 10
Gambar 4.4 *Activity diagram* memposting iklan penawaran buku 10
Gambar 4.5 *Activity diagram* request buku yang ditawarkan 45
Gambar 4.6 *Sequence diagram* pencarian buku 10
Gambar 4.7 *Sequence diagram* memposting iklan penawaran buku 10
Gambar 4.8 *Sequence diagram* request buku yang ditawarkan 50
Gambar 4.9 *Class diagram* pencarian buku 10
Gambar 4.10 *Class diagram* posting iklan penawaran buku 10
Gambar 4.11 *Class diagram* request buku yang ditawarkan 53
Gambar 4.12 *Class diagram* aplikasi 10
Gambar 4.13 Perancangan arsitektur aplikasi 56
Gambar 4.14 Diagram object 10
Gambar 4.15 *Physical Data Model* (PDM) 10
Gambar 4.16 Perancangan layout *desktop* dengan ukuran *fixed* 10
Gambar 4.17 Perancangan layout desktop dengan *fluid grid* 10
Gambar 4.18 Perancangan layout *tablet* 10
Gambar 4.19 Perancangan layout *smartphone* 10
Gambar 4.20 Perancangan antarmuka halaman utama pada *desktop* 10
Gambar 4.21 Perancangan halaman utama pada *smartphone* dan *tablet* 10
Gambar 4.22 Perancangan antarmuka pencarian buku pada *desktop* 10

Gambar 4.23 Perancangan antarmuka pencarian pada *smartphone* atau *tablet*... 10
Gambar 4.24 Perancangan antarmuka posting penawaran buku pada *desktop*..... 10
Gambar 4.25 Perancangan antarmuka posting pada *tablet* atau *smartphone*..... 10
Gambar 4.26 Perancangan antarmuka detail buku pada *desktop*..... 10
Gambar 4.27 Perancangan antarmuka detail buku pada smartphone atau tablet.. 10
Gambar 4.28 Perancangan antarmuka request penawaran buku pada *desktop*..... 10
Gambar 4.29 Perancangan antarmuka request pada tablet atau smartphone 10
Gambar 5.1 Diagram konseptual entity relationship..... 10
Gambar 5.2 Tampilan antarmuka halaman utama pada desktop 10
Gambar 5.3 Tampilan antarmuka halaman utama pada perangkat *tablet*..... 10
Gambar 5.4 Tampilan antarmuka halaman utama pada *smartphone*..... 10
Gambar 5.5 Tampilan antarmuka pencarian buku pada perangkat *desktop*..... 10
Gambar 5.6 Tampilan antarmuka pencarian penawaran buku pada *tablet*..... 10
Gambar 5.7 Tampilan antarmuka pencarian buku pada *smartphone*..... 10
Gambar 5.8 Tampilan antarmuka posting penawaran buku pada *desktop*..... 104
Gambar 5.9 Tampilan antarmuka posting penawaran buku pada tablet..... 10
Gambar 5.10 Tampilan antarmuka posting penawaran buku pada *smartphone*... 10
Gambar 5.11 Tampilan antarmuka request buku pada desktop 106
Gambar 5.12 Tampilan antarmuka request penawaran buku pada tablet 107
Gambar 5.13 Tampilan antarmuka request penawaran buku pada *smartphone* . 107

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait 8
Tabel 2.2 Browser desktop yang mendukung Geolocation API 10
Tabel 2.3 Perangkat Bergerak yang mendukung Geolocation API 10
Tabel 4.1 Identifikasi aktor 10
Tabel 4.2 Spesifikasi kebutuhan aplikasi 10
Tabel 4.3 Kebutuhan non fungsional 36
Tabel 4.4 Skenario use case pencarian buku 10
Tabel 4.5 Skenario use case memposting penawaran buku 10
Tabel 4.6 Skenario use case request buku yang ditawarkan 43
Tabel 4.7 Rancangan tabel data user 10
Tabel 4.8 Rancangan tabel data penawaran buku 10
Tabel 4.9 Rancangan tabel komentar 10
Tabel 4.10 Rancangan tabel permintaan 60
Tabel 4.11 Rancangan tabel pesan 60
Tabel 4.12 Identifikasi dan rencana pengujian 10
Tabel 4.13 Kasus uji pencarian buku 10
Tabel 4.14 Kasus uji memposting penawaran buku 10
Tabel 4.15 Kasus uji request buku yang ditawarkan 76
Tabel 4.16 Kasus uji pengujian aplikasi pada layar *smartphone* 10
Tabel 4.17 Kasus Uji pengujian aplikasi pada layar *tablet* 10
Tabel 4.18 Kasus uji pengujian aplikasi pada layar laptop 10
Tabel 4.19 Kasus uji pengujian akuransi jarak 10
Tabel 4.20 Kasus uji pengujian compatibilty 10
Tabel 5.1 Spesifikasi perangkat lunak 10
Tabel 5.2 Implementasi class pada file program 10
Tabel 6.1 Hasil Pengujian Validasi 10
Tabel 6.2 Hasil pengujian antarmuka aplikasi pada layar *smartphone* 10
Tabel 6.3 Hasil pengujian antarmuka aplikasi pada layar *tablet* 10
Tabel 6.4 Hasil pengujian antarmuka aplikasi pada layar *laptop* 10
Tabel 6.5 Studi kasus posisi perangkat user 10
Tabel 6.6 Studi kasus lokasi user penawaran buku 10
Tabel 6.7 Hasil perhitungan jarak lokasi dengan Haversine Formula 10

Tabel 6.8 Hasil pengujian kompatibilitas sistem 10

DAFTAR KODE

Kode 2.1 Program untuk menampilkan Google Maps 10
Kode 2.2 Program penerapan Geolocation API 10
Kode 2.3 Program CSS penerapan flexible grids 10
Kode 2.4 CSS konvensi nilai pixel menjadi satuan persen 10
Kode 2.5 CSS penerapan responsive font-size 10
Kode 2.6 CSS penerapan flexible image and media 10
Kode 2.7 Contoh kode CSS untuk menampilkan object secara responsive 10
Kode 5.1 Script PHP pada class bukuControl untuk proses pencarian buku 10
Kode 5.2 Script PHP pada class bukuModel untuk hasil pencarian buku 10
Kode 5.3 Script Javascript untuk melakukan pencarian buku 10
Kode 5.4 Script PHP pada class dashboardControl untuk posting penawaran 10
Kode 5.5 Script PHP pada class bukuModel untuk menyimpan data penawaran 10
Kode 5.6 Script PHP pada class *permintaanControl* untuk request buku 92
Kode 5.7 Script PHP pada class *permintaanModel()* 93
Kode 5.8 Kode CSS untuk mengatur ukuran *fluid grid* 94
Kode 5.9 Kode CSS untuk mengatur antarmuka aplikasi pada *desktop*. 96
Kode 5.10 Kode CSS untuk mengatur antarmuka aplikasi pada *tablet*. 96
Kode 5.11 Kode CSS untuk mengatur antarmuka pada *smartphone*. 97
Kode 5.12 Implementasi antarmuka halaman utama 10
Kode 5.13 Implementasi antarmuka halaman pencarian buku 10
Kode 5.14 Implementasi antarmuka posting penawaran buku 10
Kode 5.15 Implementasi antarmuka posting penawaran buku 106

BAB I
PENDAHULUAN

1.1 Latar Belakang

Buku adalah jendela dunia, maksudnya segala informasi yang ada di penjuru dunia bisa diketahui melalui membaca buku. Buku sebagai sumber informasi bisa kita dapatkan melalui perpustakaan maupun toko buku. Perpustakaan sebagai tempat untuk menyimpan buku-buku atau bahan pustaka yang dapat digunakan masyarakat untuk mengakses berbagai sumber informasi lewat buku. Kelebihan lain dari perpustakaan yaitu kita bisa mendapatkan buku-buku lama yang sudah tidak terbit atau tidak ada di toko buku. Namun, permasalahan yang dikeluhkan masyarakat terhadap perpustakaan saat ini yaitu ketidaklengkapan koleksi buku dan keterbatasan jumlah buku. Hal ini dibuktikan, berdasarkan kritik dan saran yang dapat dilihat di website perpustakaan Universitas Brawijaya dan Perpustakaan Umum kota Malang, banyak pengunjung perpustakaan yang mengeluhkan ketidaklengkapan koleksi dan jumlah buku yang masih sedikit. Sehingga mengakibatkan pengunjung perpustakaan tidak mendapatkan bahan pustaka yang sesuai dengan kebutuhan mereka. Permasalahan lainnya yaitu kesulitan mencari pustaka di toko buku dikarenakan buku yang kita butuhkan sudah tidak terbit atau sudah tidak dijual belikan di toko buku.

Salah satu alternatif lain untuk mendapatkan buku yang tidak ada diperpustakaan dan toko buku, bisa dilakukan dengan cara menawarkan buku-buku yang sudah tidak terpakai untuk dijual, dipinjamkan atau disewakan dengan harga yang lebih murah kepada orang lain. Namun, tidak adanya media informasi pencarian dan penawaran buku menyulitkan seseorang untuk melakukan pencarian dan penawaran buku.

Dari permasalahan diatas, dibutuhkan sebuah media informasi pencarian dan penawaran buku yang membantu masyarakat untuk menemukan atau mencari buku yang tidak tersedia diperpustakaan dan toko buku. Informasi yang dibutuhkan masyarakat dalam pencarian buku yaitu informasi lokasi pengguna yang menawarakan bukunya. Informasi lokasi pencarian buku akan ditampilkan penawaran terdekat yang berada disekitar area lokasi pengguna dengan memanfaatkan teknologi *Geolocation* dan Haversine Formula. Penerapan teknologi *Geolocation* dan Haversine Formula pada aplikasi dapat membantu

pengguna melakukan pencarian buku dengan ditampilkan jarak lokasi sehingga pengguna dapat menemukan penawaran buku yang berada disekitar area yang dekat dengan lokasinya. Teknologi *Geolocation* berfungsi untuk melakukan pelacakan lokasi pengguna, sehingga akan didapatkan *latitude* dan *longitude* berdasarkan informasi lokasi perangkat *user* dari GPS, *WiFi*, *IP address*, *Cell ID* (Wibowo, 2014). Selanjutnya, koordinat tersebut akan dibandingkan dengan koordinat alamat *user* pemilik buku dengan menggunakan Haversine Formula. Penerapan Haversine Formula pada Aplikasi Pencarian dan Penawaran Buku dikarenakan metode ini memiliki kemungkinan kesalahannya hanya mencapai 0.5% untuk menentukan lokasi terdekat berdasarkan jarak tempuh (Bhawiyuga, 2011).

Perkembangan perangkat bergerak seperti *tablet* dan *smartphone* yang semakin memudahkan pengguna untuk mengakses internet kapanpun dan dimanapun. Berdasarkan survei terbaru Baidu (2015), 65.7 persen pengguna Internet di Indonesia mengakses internet terbesar melalui smartphone diikuti oleh dekstop (52 persen), laptop (45 persen), dan tablet (1.9 persen) (Rizkia, 2015). Permasalahan yang terjadi, akses internet pada aplikasi web denga menggunakan perangkat bergerak seperti *tablet* dan *smartphone* menyulitkan pengguna saat mengakses halaman web dikarenakan ukuran layar yang jauh lebih kecil dibandingkan layar *desktop* komputer. Sebuah situs web jika tampilannya versi *mobile* sama dengan versi PC/Notebook akan menyulitkan pengguna saat mengakses halaman web. Konten halaman web akan terlihat sangat kecil dan kurang nyaman saat dibaca jika diakses melalui perangkat bergerak, sehingga harus diperbesar dulu tampilannya kemudian harus menggeser layar ke samping.

Ketidaknyamanan yang dirasakan pengguna pada saat mengkases halaman web di perangkat bergerak dapat diatasi dengan menerapkan teknologi *Responsive Web Design*. *Responsive Web Design* adalah sebuah teknik yang digunakan untuk membuat *layout* web menyesuaikan dengan tampilan *device* atau perangkat yang digunakan pengunjung web baik ukuran maupun orientasi, sehingga tampilan yang berada di desktop komputer dengan tampilan di perangkat bergerak atau *mobile device* akan tetap sama nyamannya dengan tampilan halaman web tersebut saat ditampilkan di *desktop* komputer (Syachbana & Akib, 2013).

Dari permasalahan diatas, aplikasi ini diharapkan dapat dijadikan sebagai media informasi pencarian dan penawaran buku untuk memudahkan pengguna mencari buku yang dibutuhkan dengan ditampilkan informasi lokasi penguna yang menawarkan bukunya. Penerapan teknologi Geolocation dan Haversine Formula pada aplikasi diharapkan dapat membantu pengguna melakukan pencarian buku dengan ditampilkan jarak lokasi sehingga pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya. Selain itu, penerapan teknik *Responsive Web Design* diharapkan aplikasi dapat beradapatasi secara optimal dalam berbagai jenis perangkat (*smartphone, tablet, desktop*) dan mendukung beberapa jenis browser yang digunakan.

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang permasalahan, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana tingkat akurasi persamaan Haversine Formula dalam perhitungan jarak lokasi sehingga membantu pengguna menemukan penawaran buku yang dekat dengan lokasinya berdasarkan jarak lokasi?
2. Bagaimana perancangan dan implementasi menggunakan *teknik responsive web design* pada aplikasi pencarian dan penawaran buku yang tampilannya dapat menyesesuaikan ukuran layar dari perangkat yang digunakan?
3. Bagaimana pengaruh penerapan teknik *Responsif Web Design* (RWD) pada aplikasi pencarian dan penawaran buku terhadap perangkat yang digunakan pengguna yang memiliki ukuran layar yang berbeda?

1.3 Batasan Masalah

Agar masalah yang dirumuskan lebih terfokus, maka pembuatan aplikasi ini dibatasi dalam hal:

1. Aplikasi hanya menyajikan pertukaran informasi untuk pencarian dan penawaran buku bekas.
2. Aplikasi hanya dapat melakukan penawaran buku dengan tujuan dijual atau disewakan
3. Informasi pencarian buku terdekat ditampilkan berdasarkan hasil perhitungan jarak terpendek.

4. Aplikasi dibangun untuk digunakan di beberapa jenis ukuran layar perangkat berbeda (perangkat bergerak atau desktop PC) yang terhubung dengan internet.
5. Data uji coba jarak lokasi hanya mengambil data alamat lokasi di area malang yang digunakan sebagai uji coba.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini yaitu:

1. Membangun aplikasi sebagai media pertukaran informasi yang dapat membantu masyarakat untuk menawarkan dan mencari buku yang tidak tersedia di perpustakaan atau toko buku dengan menampilkan informasi lokasi pemilik buku beserta jarak lokasi.
2. Membangun sebuah aplikasi yang dapat menyesuaikan tampilannya pada perangkat yang digunakan penguna yang memiliki ukuran layar yang berbeda, sehingga tampilan yang berada di desktop komputer dengan tampilan di perangkat bergerak tetap memberikan kenyamanan user saat mengakses aplikasi.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi pengguna. Manfaat dari penelitian adalah sebagai berikut:

1. Pengguna dapat menerapkan sistem sebagai media informasi Pencarian dan Penawaran Buku yang menampilkan informasi pencarian buku yang terhubung dengan orang-orang yang menawarkan bukunya.
2. Menghasilkan sebuah aplikasi yang bersifat *responsive* atau menyesuaikan secara otomatis atau melakukan resize sesuai dengan ukuran tampilan perangkat yang digunakan pengguna.
3. Pengguna dapat memanfaatkan buku-buku bekas atau buku yang sudah tidak terpakai menjadi lebih bermanfaat untuk orang lain, serta dapat menghemat biaya untuk mendapatkan buku yang kita inginkan.

1.6 Sistematika Penulisan

Untuk mencapai tujuan yang diharapkan, sistematika penulisan laporna skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Bab Pendahuluan berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan pada penelitian Rancang Bangun Aplikasi Pencarian dan Penawaran Buku.

BAB II Tinjauan Pustaka

Bab Tinjauan Pustaka menjelaskan tentang kajian pustaka penelitian terkait dengan dasar teori dan teori penunjang yang diperlukan untuk mendukung pembuatan Aplikasi Pencarian dan Penawaran Buku

BAB III Metodologi

Bab Metodelogi membahas tentang alur tahapan perancangan aplikasi yaitu studi literatur, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis.

BAB IV Analisis dan Perancangan

Bab Analisis dan Perancangan membahas tentang dan perancangan sistem yang digambarkan dengan diagram UML *(Unified Modelling Language*) sebagai dasar tahap implementasi yang akan dilakukan sesudahnya.

BAB V Implementasi dan Pemabahasan

Membahas tentang implementasi dari hasil perancangan analisis kebutuhan yang telah dibuat. Pada Implementasi pengembangan Aplikasi Pencarian dan Penawaran Buku menggunakan teknik *Responsive Web Design* dengan bahasa pemograman PHP, *JavaScript*, HTML5 dan CSS3.

BAB VI Pengujian

Bab Pengujian membahas proses dan hasil pengujian terhadap sistem yang telah dibuat. Pengujian aplikasi menggunakan pengujian validasi (*validation testing),* pengujian perhitungan jarak terdekat dan pengujian *layout web* responsif *(responsive test).*

BAB VII Penutup

Bab penutup memuat kesimpulan tentang hasil yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan serta saran untuk pengembangan sistem lebih lanjut.

BAB II
TINJAUAN PUSTAKA

Bab Tinjauan Pustaka berisi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka adalah membahasa penelitian yang telah ada dan yang diusulkan. Dasar teori adalah membahas teori yang mendukung penyusunan penelitian yang diusulkan. Teori dasar yang akan dibahas pada bab ini yaitu buku dan minat baca, *Geolocation, Responsive Web Design (RWD),* Google Map API, konsep MVC, Haversine Formula.

2.1 Kajian Pustaka

Banyak sumber acuan yang digunakan untuk mengembangkan Aplikasi Pencarian dan Penawaran Buku. Sumber acuan atau kajian pustaka yang digunakan berupa buku-buku penunjang dan penelitian sebelumnya yang pernah dilakukan oleh pihak lain. State of the art terkait penelitian sebelumnya dijelaskan pada Tabel 2.1.

Penelitian yang berjudul "Pemanfaatan *Geolocation* dan Haversine Formula dalam Perancangan Sistem Informasi Geografis (GIS) (Studi Kasus: Pariwisata Kabupaten Semarang)". Penelitian ini menghasilkan sistem dengan pencarian lokasi wisata yang berdasarkan latitude dan longitude serta jarak tempuh dari suatu objek dengan memafaatkan teknologi Geolocation untuk mengetahui latitude dan longitude dari GPS, Cell ID, Wifi maupun IP address yang terhubung oleh perangkat. Koordinat yang diperoleh dari teknologi Geolocation tersebut kemudian dilakukan proses perhitunggan seberapa besar jarak lokasi wisata menggunakan Haversine Formulla (Widyatmmoko, 2012).

Penelitian Selanjutnya berjudul "Perancangan Website Menggunakan Responsive Web Design". Penelitian ini membahas tentang perancangan website menggunakan responsive web design. Dari hasil penerapan teknik *responsive web design*, website dapat beradaptasi dengan optimal dalam perangkat apapun yang digunakan untuk mengaksesnya, berupa respon secara otomatis atau penyesuaian ukuran web terhadap ukuran layar perangkat yang digunakan

sehingga memberikan kenyamanan pengguna pada saat mengakses web tersebut (Syachbana & Akib, 2013).

Penelitian usulan "Rancang Bangun Aplikasi Pencarian dan Penawaran Buku menggunakan Teknik Responsive Web Design" akan dibangun dengan menerapkan teknik responsive web design agar aplikasi dapat berjalan responsif sesuai dengan ukuran *layout device* atau perangkat yang digunakan. Selain itu, aplikasi ini juga dapat memanfaatkan teknologi *Geolocation*. Pemanfaatan *Geolocation* dan Haversine Formula untuk melakukan pencarian penawaran buku yang berada di dekat area lokasi user berdasarkan jarak lokasi terdekat beserta rute lokasi yang ditampilkan melalui Google Maps API. Perbandingan penelitian tekait dijelaskan pada Tabel 2.1.

2.2 Buku dan Minat Baca

Buku adalah sumber ilmu. Berbagai macam ilmu dituliskan ke dalam sebuah buku. Tidak hanya informasi yang ada dalam negeri yang didapatkan, melainkan informasi tentang dunia, bahkan alam semesta. Hanya saja masalahnya, budaya membaca belum menjadi budaya masyarakat kita. Dilihat dari Badan Pusat Statistik (BPS) tahun 2010, masyarakat Indonesia belum menjadikan membaca sebagai sumber utama mendapatkan informasi. Orang lebih memilih menonton TV (85,9%) atau mendengarkan radio (40,3%) ketimbang ,membaca Koran (23,5%) (Guswid, 2010).

Ada banyak faktor yang menyebabkan minat baca masyarakat Indonesia tergolong rendah. Pertama, kurangnya sarana dan pra sarana,khususnya perpustakaan dengan buku-buku yang bermutu dan memadai. Kedua, harga buku di Indonesia tergolong mahal.perlu ratusan ribu untuk membeli enslikopedi yang berkualitas. Berbeda dengan Negara lain seperti Australia ,buku-buku di bagikan secara gratis. Hal tersebut menjadikan masyarakat malas untuk membaca buku karena untuk membaca buku harus mengeluarkan uang yang cukup banyak. Ketiga, banyak keluarga di Indonesia yang belum menanamkan budaya membaca (Putri, 2010).

Tabel 2.1 Penelitian Terkait

No	Judul	Objek **Input dan Parameter**	Metode **Proses**	Output **Hasil Penelitian**
1	Pemanfaatan *Geolocation* dan Haversine Formula dalam Perancangan Sistem Informasi Geografis (GIS) (Studi Kasus: Pariwisata Kabupaten Semarang) (Widyatmmoko, 2012)	Data Masukan: - Latitude dan longitude lokasi object wisata - Jarak radius	Metode: - Memafaatkan teknologi geolocation dan Haversine Formula	Menghasilkan sistem dengan pencarian lokasi wisata yang berdasarkan latitude dan longitude serta jarak tempuh dari suatu objek
2	Perancangan Website Menggunakan Responsive Web Design (Syachbana & Akib, 2013).	Website	Metode: Teknik Responsive Web Design	website dapat beradaptasi dengan optimal dalam perangkat apapun yang digunakan untuk mengaksesnya, berupa respon secara otomatis atau penyesuaian ukuran web terhadap ukuran layar perangkat yang digunakan
3	Rancang Bangun Aplikasi Pencarian dan Penawaran Buku menggunakan Teknik Responsive Web Design (Usulan)	- Data Buku - Latitude dan longitude lokasi user penawar buku - Website	Metode: 1. teknologi geolocation dan Haversine Formula Profile Matching 2. Teknik Responsive Web Design Proses: - Pelacakan lokasi *user* dengan	Menghasilkan sebuah aplikasi web yang dapat digunakan di beberapa ukuran layar perangkat bergerak ataupun komputer. Selain itu, aplikasi dapat melakukan pencarian penawaran buku yang

			menggunakan teknologi *Geolocation*, sehingga akan didapatkan *latitude* dan *longitude* dari sebuah perangkat yang dilacak berdasarkan informasi lokasi *user* dari GPS, *WiFi, IP address, Cell ID*. Selanjutnya, koordinat tersebut akan dibangingkan dengan koordinat alamat *user* pemilik buku dengan menggunakan Haversine Formula sehingga *user* dapat menemukan alamat pemilik buku yang dekat dengan lokasinnya. - Impementasi responsive web design menggunakan teknik *system grid, fluid grid* dan *media queries*	berada disekitar area lokasi terdekat user.

Sumber : (Syachbana & Akib, 2013), (Widyatmmoko, 2012)

2.3 Google Maps

Google Maps adalah Google Maps merupakan sebuah layanan peta dunia virtual berbasis Web yang disediakan Google. Layanan ini gratis dan bisa ditemukan serta diakses di *http://maps.google.com*. Google Maps menawarkan peta yang dapat digeser (*panned*), diperbesar (*zoom in*), diperkecil (*zoom out*) dan dapat diganti dalam beberapa mode. Google Maps menyediakan fitur pencarian rute (*routing*), penunjuk arah dari satu objek peta ke objek yang lain (*direction*), dan juga pencari tempat (*place*).

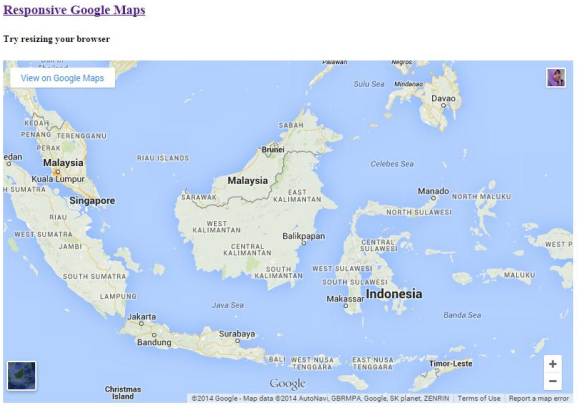
Google Maps menyediakan layanan API *(Application Programming Interface)* yang memungkinkan developer lain untuk memanfaatkan aplikasi ini di aplikasi buatannya. Google Maps API adalah suatu library yang berbentuk *JavaScript*. Cara membuat Google Maps untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript.*

Didalam aplikasi Google Maps API terdapat berbagai fungsi pemograman yang digunakan untuk mengolah dan menampilkan data pemetaan. Untuk dapat menggunakan fungsi-fungsi dari Google Maps API kita harus memanggil fungsi-fungsi tersebut dengan bahasa pemograman *JavaScript*. merupakan contoh halaman *web* yang menampilkan peta dari Google Maps (Putra, 2012).

```
<script src="http://maps.googleapis.com/maps/api/js"></scrip>
<script>
function initialize() {
  var mapProp = {
    center:new google.maps.LatLng(51.508742,-0.120850),zoom:5,
    mapTypeId:google.maps.MapTypeId.ROADMAP
  };
  var map=new
google.maps.Map(document.getElementById("Google Maps"),mapProp);
}
google.maps.event.addDomListener(window,'load',initialize);
</script>
```

Kode 2.1 Program untuk menampilkan Google Maps
Sumber: (w3schools, 2011)

Apabila kode diatas dijalankan, maka *browser* akan menampilkan halaman *web* yang menampilkan peta dari Google Maps serperti .



Gambar 2.1 Tampilan Google Maps

2.4 Geolocation

Geolocation adalah teknologi yang memungkinkan aplikasi untuk mengindentifikasi lokasi geografis dari pengguna atau perangkat komputasi melalui berbagai mekanisme pengunpulan data. *Geolocation API* memungkinkan untuk berbagi lokasi dengan website yang terpercaya seperti menampilkan lokasi geografis pengguna pada peta atau menunjukkan tempat-tempat menarik di sekitar lokasi saat itu. Informasi lokasi geografis geografis diperoleh melalui *IP addres*, koneksi jaringan *wireless* (Wi-Fi), menara BTS *(Base Transceiver Station)* atau melalui perangkat GPS (*Global Positioning System*) yang terintegrasi (rumahdot, 2012). Saat ini *API Geolocation W3C* didukung oleh *browser desktop* yang ditunjukan pada tabel 2.1:

Tabel 2.2 Browser desktop yang mendukung Geolocation API

Browser	Version
Firefox	3.5+
Chrome	5.0+
Safari	5.0+
Opera	10.6+
Internet Explorer	9.0+

Sumber: (rumahdot, 2012)

Ada juga dukungan untuk API *Geolocation W3C* pada perangkat bergerak, seperti yang ditunjukan pada :

Tabel 2.3 Perangkat Bergerak yang mendukung Geolocation API

Browser	Version
Android	2.0+
iPhone	3.0+

Opera Mobile	10.1+
Symbian	S60 3rd & 5th generation

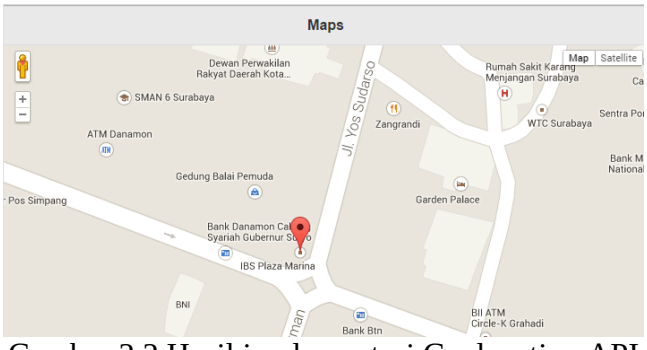
Sumber: (rumahdot, 2012)

Kode 2.2 merupakan contoh penerapan *Geolocation API* dengan menggunakan HTML 5 dari peta Google Maps API:

```
  var map;
    var mapOptions = {
      zoom: 6
    };
    map = new google.maps.Map(document.getElementById('map-
canvas'),
        mapOptions);

    // Try HTML5 geolocation
    if(navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(function(position)
{
        var pos = new
google.maps.LatLng(position.coords.latitude,
position.coords.longitude);

        var infowindow = new google.maps.InfoWindow({
          map: map,
          position: pos,
          content: 'Location found using HTML5.'
        });
        map.setCenter(pos);
      }, function() {
        handleNoGeolocation(true);
      });
    } else {
      // Browser doesn't support Geolocation
      handleNoGeolocation(false);
    }
  }

  function handleNoGeolocation(errorFlag) {
    if (errorFlag) {
      var content = 'Error: The Geolocation service failed.';
    } else {
      var content = 'Error: Your browser doesn\'t support
geolocation.';
    }

    var options = {
      map: map,
      position: new google.maps.LatLng(60, 105),
      content: content
    };
```

Kode 2.2 Program penerapan Geolocation API
Sumber: (rumahdot, 2012)

Apabila kode 2.2 dijalankan, browser akan menampilkan peta lokasi pengguna saat ini yang ditandai pada marker berwarna merah pada Gambar 2.2



Gambar 2.2 Hasil implementasi Geolocation API

2.6 Responsive Web Design

Responsive Web Design (RWD) merupakan salah satu teknik yang dapat membuat proses perancangan aplikasi dan situs web untuk berbagai jenis perangkat menjadi lebih mudah. RWD memberikan keunggulan untuk dapat diterapkan ke dalam perancangan situs web sehingga situs dapat diakses melalui *smartphone, tablet, desktop,* ataupun *smartTV* tanpa memperlihatkan perbedaan yang terlalu besar dalam hal penggunaan. Dengan menggunakan konfigurasi *responsive design*, sebuah situs *web* mampu memberikan respon secara otomatis terhadap ukuran layar, sehingga situs dapat beradaptasi dengan optimal dalam perangkat apapun yang digunakan pengguna untuk mengaksesnya (Ramadhani, 2013).

Gambar 2.3 Tampilan website yang menerapkan RWD
Sumber: (Ramadhani, 2013)

Ada 3 unsur inti yang harus ada pada RWD agar bisa bekerja dengan baik, yaitu pada *Flexible Grids, Fluid Images dan Media queries*. Dari ketiga unsur tersebut diharapkan sebuah desain nantinya bisa beradaptasi pada berbagai layar berbeda, tidak peduli beberapapun banyaknya perangkat yang ada. Dengan RWD semua bisa dikontrol hanya dalam satu desain saja (Ramadhani, 2013).

2.6.1 Flexible Grids

Flexible Grids adalah salah satu pilar dalam RWD, yang merupakan cara baru dalam mengatur posisi, *layout, margin* dan *spacing* dari suatu halaman *web*. Pada umumnya *layout* dan ukuran *text* didefinisikan dalam satuan *px (pixel)* untuk ukuran *layout* dan *text*. Namun, untuk desain responsif saat ini para pengembang lebih banyak mengunakan satuan % (*percent*) sebagai penggantinya, ini dimaksudkan agar desain pada *layout* tidak tampak kaku dan bisa *flexible* ketika ditampilkan pada resolusi layar yang berbeda.

Dalam Flexible Grids terdapat teknik untuk untuk mengganti desain lama yang masih menggunakan satuan *px* dengan desain yang lebih *responsive* yaitu menggunakan formula perhitungan pada Flexible Grids. Desain yang belum resposif biasanya lebih banyak menggunakan satuan *px* seperti dalam contoh CSS yang ditunjukan pada Kode 2.3 (Ramadhani, 2013). :

```
.sidebar{
width:300px;
}
.content{
width:700px;
}
.wrap{
width:1024px;
}
```

Kode 2.3 Program CSS penerapan flexible grids
Sumber: (rumahdot, 2012)

Dari code css diatas lebar *grid* utama adalah dalam reolusi *width*:1024px dan untuk *sidebar width*:300px dan *content width*:700px. Untuk mengkonversi nilai px *(pixel)* agar bisa diganti dengan satuan % (*percent*) menggunakan formula dan kalkulasi sederhana seperti berikut (Ramadhani, 2013):

Rumus perhitungan konversi nilai pixel menjadi satuan persen

Target / Context = Result

300px / 1024px = 29.296875%; /* 300 (originally 300) / 1024 */

700px / 1024px = 68.359375%; /* 700 (originally 700) / 1024 */

Dari hasil kalkulasi diatas maka bisa didapatkan nilai *percent* dalam koding CSS pada Kode 2.4:

```
.sidebar{
width:29.29%;
}
.content{
width:68.35%;
}
.wrap{
width:1024px;
}
```

Kode 2.4 CSS konvensi nilai pixel menjadi satuan persen
Sumber: (rumahdot, 2012)

Formula diatas sebenarnya bisa juga digunakan untuk menghitung ukuran text (*font size*), Katakanlah konteks normal untuk ukuran *font* dalam *body* adalah 16 piksel. Jika ingin menetapkan bahwa H1 harus 24 pixel, maka bisa diganti dengan satuan cm dengan cara menghitungnya sebagai berikut:

24px / 16px = 1,5cm

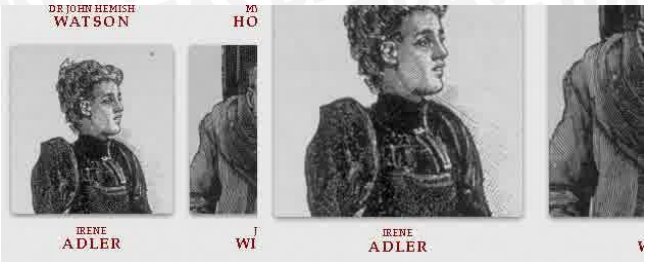
Dengan code CSS akan didapat koding yang ditujukan pada Kode 2.5:

```
h1 {
font-size: 1.5em;
}
```

Kode 2.5 CSS penerapan responsive font-size
Sumber: (rumahdot, 2012)

2.6.2 Flexible Image and Media

Flexible Image merupakan salah satu pilar RWD untuk menampilkan gambar atau media lainya seperti video secara dinamis sesuai dengan ukuran dan orientasi layar pada perangkat. Flexible Image dapat dilakukan dengan melakukan *scaling*. Gambar (image) dengan menggunakan satuan cm dan px misalkan *width*:400px dan *height*:500px tidak akan bekerja baik dengan RWD, langkah termudah adalah dengan mengganti satuannya menjadi % contoh: *width*:100% dan *height*:100%. Tidak semua gambar harus menggunakan %, tetapi bisa menggunakan alternatif *auto* sebagai gantinya (Ramadhani, 2013).



Gambar 2.4 Contoh penerapan flexibe image
Sumber: (Marcotte, 2010)

Untuk menampilkan ukuran gambar sesuai *size* aslinya bisa diterapkan menggunakan kode 2.6:

```
img{
height:auto;
width:auto;
max-width:100%
}
```

Kode 2.6 CSS penerapan flexible image and media
Sumber: (Marcotte, 2010)

Untuk *video, iframe, embed dan object* pada intinya hampir sama caranya dengan *responsive* pada gambar yaitu tetap memberi satuan %. Saat ini *embed video* yang paling banyak digunakan adalah Youtube. Namun terlepas dari Youtube, ada juga yang menanam langsung pada *Self Hosting* dan biasanya untuk pemanggilannya mengunakan *tag* <*video*>. Membuat *responsive* pada video dalam *iframe* sangatlah mudah, seperti menangani *responsive* pada gambar, kita tinggal mengontrolnya langsung melalui koding CSS. Kode 2.7 merupakan contoh koding CSS untuk penampilkan video dalam ukuran penuh sesuai dengan ukuran lebar *div*.

```
.ResponsiveWrapper {
 position: relative;
 padding-bottom: 56.25%; /* 16:9 */
 padding-top: 25px;
 height: 0;
}
.ResponsiveWrapper iframe, video, object, embed {
 position: absolute;
 top: 0;
 left: 0;
 width: 100%;
 height: 100%;
}
```

Kode 2.7 Contoh kode CSS untuk menampilkan object secara responsive
Sumber: (Marcotte, 2010)

2.6.3 Media Queries

Media Query adalah salah satu fitur dalam CSS3 yang memungkinkan tampilan *web* untuk menentukan kapan aturan CSS tertentu harus diterapkan. Hal ini memungkinkan untuk menerapkan CSS khusus untuk perangkat bergerak. Media Queries bisa dikatakan otak dari *Design Web Responsive* karena memiliki kontrol penuh terhadap *design responsive template* tampilan *website*. Contoh tampilan website yang menerapkan Media Queries ditunjukan pada .

Gambar 2.5 Contoh penerapan media queries
Sumber: (Marcotte, 2010)

Ada 3 cara untuk pemanggilan *media queries*:

1. Menggunakan @import untuk mengimport aturan Style dari Style Sheet lainnya.

```
@import url(style768min.css) screen and (min-width: 768px);
```

2. *Menambahkan tag Media Queries max-width untuk mengurutkan ukuran resolusi layar dari resolusi besar ke kecil dan tag min-width untuk resolusi layar dari resolusi kecil ke besar.*

```
@media screen and (max-width:768px){
/* CSS styles */
}
@media screen and (max-width:320px){
/* CSS styles */
}
@media screen and (max-width:240px){
/* CSS styles */
}
```

3. *Pemanggilan melalui link untuk* Style Sheet *secara terpisah.*

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 768px)" href="style768.css" />
```

2.7 Bootstrap

Bootstrap adalah kumpulan alat yang gratis untuk membuat website dan aplikasi web. Ini berisi HTML dan CSS berbasis desain template untuk tipografi, bentuk, tombol, navigasi dan komponen antarmuka lainnya, serta opsional ekstensi JavaScript. Bootstrap dikembangkan oleh Mark Otto dan Jacob Thornton di Twitter sebagai kerangka untuk mendorong konsistensi di alat internal. Sebelum Bootstrap, berbagai perpustakaan yang digunakan untuk pengembangan antarmuka, yang menyebabkan inkonsistensi dan beban pemeliharaan tinggi. Bootstrap memiliki dukungan yang relatif lengkap untuk HTML5 dan CSS 3, sehingga kompatibel dengan semua browser utama. Informasi dasar dari kompatibilitas situs atau aplikasi yang tersedia untuk semua perangkat dan browser. Terdapat konsep kompatibilitas parsial yang membuat informasi dasar dari sebuah situs web yang tersedia untuk semua perangkat dan browser. Misalnya, sifat diperkenalkan pada CSS3 untuk sudut dibulatkan, gradien dan bayangan yang digunakan oleh Bootstrap meskipun kurangnya dukungan oleh browser web lama.Bootstrap adalah modular dan pada dasarnya terdiri dari serangkaian stylesheet LESS yang menerapkan berbagai komponen toolkit. Sebuah stylesheet disebut bootstrap.less mencakup komponen-komponen stylesheet. Pengembang dapat menyesuaikan file Bootstrap sendiri, memilih komponen yang ingin mereka gunakan dalam proyek mereka.CSS Framework ini hadir dengan lebar standar 940 pixel, layout grid. Pengembang dapat menggunakan tata letak variable-width. Untuk kedua kasus, toolkit memiliki empat variasi untuk menggunakan resolusi yang berbeda dan jenis perangkat: ponsel, portrait dan landscape, tablet dan PC dengan resolusi rendah dan tinggi. Setiap variasi menyesuaikan lebar kolom (Riyadi, 2014).

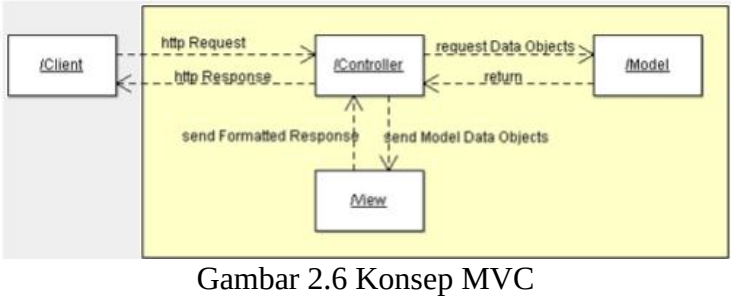
Bootstrap dilengkapi dengan beberapa komponen JavaScript dalam bentuk plugin jQuery. Mereka menyediakan elemen antarmuka pengguna tambahan seperti kotak dialog, tooltips, dan carousels. Mereka juga memperluas fungsionalitas dari beberapa elemen antarmuka yang ada, termasuk misalnya fungsi auto-complete untuk bidang masukan. Dalam versi 2.0, plugin JavaScript berikut ini didukung: *Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Tutup, Carousel* dan *Typeahead*. Untuk menggunakan Bootstrap dalam sebuah halaman HTML, pengembang cukup mengunduh Bootstrap CSS

stylesheet dan termasuk link dalam file HTML yang tersedia di website resmi http://getbootstrap.com/ (Riyadi, 2014).

2.8 Konsep MVC

Model-View-Controller atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasinya kebanyakan framework dalam aplikasi *website* adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna dan bagian yang menjadi *control* dalam sebuah aplikasi *web*.

Konsep MVC adalah konsep pemisahan antara *logic* dengan tampilan dan database. Manfaat konsep ini adalah, membuat *coding logic* lebih simple, karena sudah di pisah dengan *code* untuk tampilan dan membuat programmer dapat bekerja secara terpisah dengan *designer*. *Programmer* mengerjakan *logic*, sedangkan *designer* berkutat dengan design dan tampilan. Gambaran konsep MVC ditunjukan pada



Gambar 2.6 Konsep MVC
Sumber: (Rossa & Shalahudin, 2013)

Berikut modul yang terdapat pada arsitektur MVC (Rossa & Shalahudin, 2013) :

1. Model

Model adalah layer yang bertanggung jawab untuk melakukan hubungan dengan *database*. Model pada umunnya digunakan untuk mengambil data dari *database* atau menyimpan data ke dalam *database.*

2. View

View adalah komponen dari aplikasi yang menampilkan antarmuka untuk pengguna *(user interface).* Layer ini mengandung keseluruhan detail dari implementasi *user interface*. Disini, komponen *grafis* menyediakan representasi

proses internal aplikasi dan menuntun alur interaksi *user* terhadap aplikasi. Tidak ada layer lain yang berinteraksi dengan *user*, melainkan hanya *view*.

3. Controller

Controller adalah layer yang menyediakan detail alur program dan transisi antar layer (*model* dan *view*), dan juga bertanggung jawab akan penampungan *events* yang dibuat oleh *user* dari *view* dan melakukan *update* terhadap komponen *model* menggunakan data yang dimasukkan oleh *user*.

2.9 Haversine Formula

Haversine Formula adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bumi berdasarkan garis bujur dan lintang. Penggunaan rumus ini mengasumsikan pengabaian efek *ellipsoidal*, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Penggunaan metode cukup akurat untuk sebagian besar perhitungan dan kemungkinan kesalahan hanya 0.5%. Berikut adalah rumus haversine yang ditunjukan pada dan (Longitudestore, 2012):

$$haversine\left(\frac{d}{r}\right) = haversin(\varphi_1 - \varphi_2) + \cos(\varphi_1)\cos(\varphi_2) haversin\,(\lambda_1 - \lambda_2) \quad \text{....(2-1)}$$

Persamaan 2.1 Rumus Haversine Formula
Sumber: (Longitudestore, 2012)

$$haversine = sin^2\left(\frac{\varphi}{2}\right) = \frac{1-\cos(\varphi)}{2} \quad \text{.....(2-2)}$$

Persamaan 2.2 Fungsi Haversine Formula
Sumber: (Longitudestore, 2012)

Keterangan:

haversine	: fungsi Haversin Formula
d	: jarak antara dua titik bumi
r	: radius dari lingkaran bumi
(φ_1,φ_2)	: latitude
(λ_1,λ_2)	: longitude

Sebagai contoh perhitungan dari rumus Haversine Formula, dimisalkan *latitude* dan *longitude* pengguna adalah 7.9633666 dan 112.6128482. Sedangkan *latitude* dan *longitude* lokasi yang dituju adalah -7.970177 dan 112.627524. Dengan menggunakan rumus Haversine Formula, maka langkah-langkah untuk menghitung jarak adalah sebagai berikut:

φ1 = -7.9633666
φ2 = -7.970177
Δφ = -7.970177-(-7.9633666) = 0.0068104
Δλ = 112.627524-112.6128482 = 0.0146758
r = 6371.

$$d = 2 * 6371.1 \arcsin(\sqrt{\sin^2\left(\frac{0.0068104}{2}\right) + cos\,(-7.9633666)cos(-7.970177)\sin^2\left(\frac{0.0146758}{2}\right)})$$

d = 1.784778645

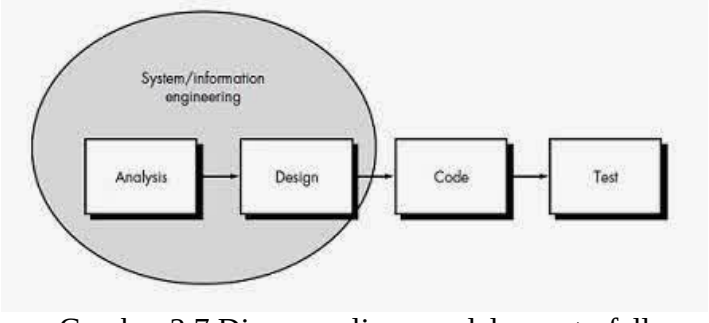
Melalui perhitungan diatas didapatkan jarak anatara pengguna dan lokasi yang dituju adalah sekitar 1.79 kilometer atau 1784.8 meter.

2.10 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak adalah penerapan yang sistematis, disiplin serta pendekatan yang terukur terhadap pengembangan pengoperasian dan pemeliharaan perangkat lunak. Seorang *software engineer* atau sekumpulan *software engineer* harus menggabungkan strategi pengembangan perangkat lunak yang meliputi proses, metode dan alat bantu yang digunakan dalam proses pengembangan. Strategi ini disebut sebagai model proses. Model proses untuk rekayasa perangkat lunak dipilih sesuai dengan sifat dari proyek dan aplikasi yang akan dibuat. Terdapat beberapa model proses untuk rekayasa perangkat lunak antara lain *linear sequential model (waterfall), prototyping, Rapid Aplication Developmet (RAD), incremental model dan spiral model* (Pressman, 2001). Model proses yang digunakan dalam penelitian ini adalah Model Waterfall.

Metode waterfall merupakan metode yang sering digunakan oleh penganalisis sistem pada umumnya. Model Waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean dan pengujian. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya (Rossa & Shalahudin, 2013). Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan. Model *waterfall* meliki kelebihan prosesnya mudah dipahami dan stuktur sistemnya jelas. Model waterfall digunakan pada proses pengembangan perangkat lunak dikarenakan kebutuhan *user* telah dipahami dan

kemungkinan adanya perubahan kebutuhan *user* kecil. Tahapan-tahapan pengembangan sistem menggunakan model *waterfall* dijelaskan pada Gambar 2.7.



Gambar 2.7 Diagram alir pemodelan waterfall
Sumber: (Rossa & Shalahudin, 2013)

Penjelasan diagram alir pemodelan waterfall pada adalah sebagai berikut:

1. Analisis kebutuhan

   Pada tahap ini dilakukan pengumpulan kebutuhan Aplikasi Pencarian dan Penawaran Buku yaitu kebutuhan pengguna, kebutuhan perangkat keras, kebutuhan perangkat lunak dan basis data. Tahap ini dinyatakan selesai apabila seluruh kebutuhan sudah terkumpul dan kemudian kebutuhan fungsional sistem telah dimodelkan ke dalam diagram *use case*.

2. Desain

   Pada tahap ini mentraslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Hasil dari tahapan desain perangkat lunak dapat direpresentasikan berupa diagram UML seperti class diagram, activity diagram, sequential diagram dan diagram entitas (ERD) serta perancangan antar muka aplikasi.

3. Pembuatan Kode Program

   Pada tahap ini adalah pembuatan program perangkat lunak sesuai desain yang telah dibuat pada tahap desain. Hasil implementasi berupa aplikasi berbasis *web responsive* dengan memanfaatkan teknologi *geolocation API* yang dapat menampilkan informasi pencarian dan penawaran buku.

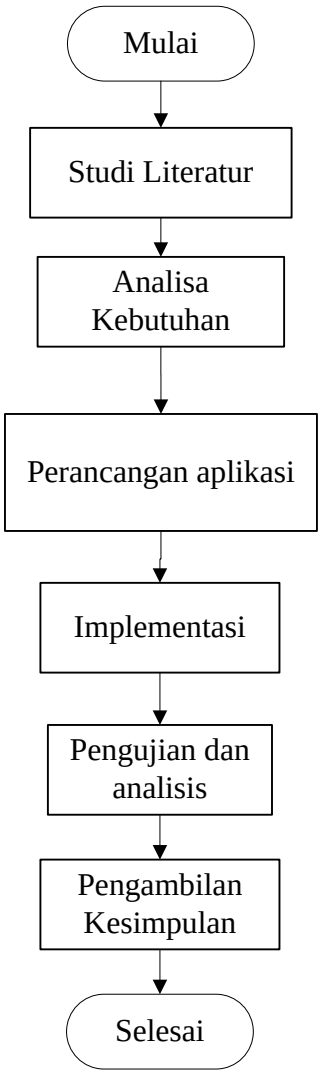
4. Pengujian

   Pada tahap ini dilakukan pengujian sebelum aplikasi dipublikasikan untuk meminimalisir kesalahan *(error)* dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Pengujian Aplikasi Pencarian dan Penawaran Buku menggunaan teknik pengujian validasi, pengujian jarak, pengujian

layout web responsif *(responsive test)* dan pengujian kompatibilitas sistem terhadap berbagai jenis *browser*. Pada tahap ini dikatakan selesai, apabila seluruh pengujian sudah dilakukan. Kemudian dilakukan analisis sistem untuk membandingkan hasil yang diharapkan dari analisis kebutuhan apakah sudah sesuai dengan hasil sistem yang sudah diimplementasikan sehingga didapatkan suatu kesimpulan.

BAB III
METODOLOGI

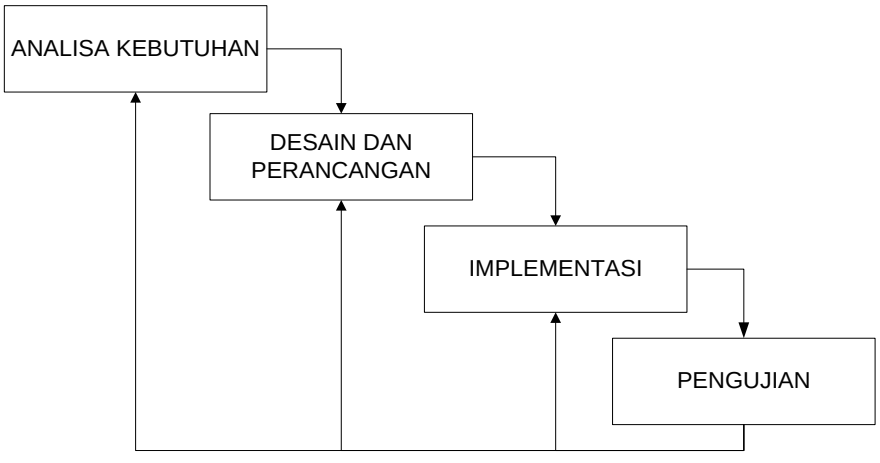
Pada bab ini berisi tentang metodologi penelitian yang akan dilakukan dalam perancangan dan pembangunan Aplikasi Pencarian dan Penawaran Buku. Metodologi penelitian akan menjelaskan tentang tahapan yang akan dilakukan dalam proses penelitian. Gambar 3.1 merupakan langkah-langkah yang akan dilakukan dalam penelitian meliputi studi literatur, perancangan, implementasi dan pengujian dari aplikasi yang akan dikembangkan. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan perangkat lunak selanjutnya.



Gambar 3.1 Diagram alir metodelogi penelitian

3.1 Metode Pengembangan

Agar penelitian yang dilakukan lebih terarah maka akan digunakan suatu metode. Dalam penelitian ini metode yang digunakan dalam pengembangkan mengunakan metode *waterfall* yang ditunjukan pada Gambar 3.2.



Gambar 3.2 Diagram alir metode pembangunan aplikasi

Gambar 3.2 merupakan tahapan-tahapan pengembangan sistem menggunakan model *waterfall.* Model Waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari studi literature, analisis kebutuhan, perancangan, implementasi dan terakhir pengujian dan analisis. Artinya, jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya (Rossa & Shalahudin, 2013). Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan. Model *waterfall* memiliki kelebihan prosesnya mudah dipahami dan stuktur sistemnya jelas. Model waterfall digunakan pada proses pengembangan perangkat lunak dikarenakan kebutuhan *user* telah dipahami dan jelas serta kemungkinan adanya perubahan kebutuhan *user* kecil.

3.2 Studi Literatur

Studi literatur adalah mencari referensi dasar teori yang digunakan untuk menunjang pengimplementasian aplikasi pencarian dan penawaran buku menggunakan teknik *responsive web design.* Dasar teori tersebut diperoleh dari beberapa sumber yaitu : buku, jurnal, laporan penelitian dan bantuan mesin pencari (search engine) internet. Dasar Teori yang digunakan dalam mengimplementasikan aplikasi meliputi : Buku dan minat baca, Google Maps,

Geolocation, Responsive Web Design, Konsep MVC, Haversine Formula. Tujuan dilakukan studi literatur adalah untuk memperkuat permasalahan serta sebagai dasar teori dalam melakukan studi dan menjadi dasar untuk mengimplementasikan aplikasi pencarian dan penawaran buku.

3.3 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengetahui kebutuhan pengguna terhadap aplikasi yang akan dikembangkan. Hal ini perlu dilakukan agar aplikasi yang dikembangkan sesuai dengan kebutuhan pengguna. Dibagian ini juga dilakukan identifikasi siapa saja yang akan menggunakan aplikasi, dan informasi apa saja yang bisa digunakan oleh mereka. Identifikasi kebutuhan dilakukan dengan cara melakukan observasi terhadap permasalahan ketersediaan pustaka di pepustakaan Universitas Brawijaya dan perpustakaan kota Malang dengan melihat informasi kritik dan saran dari pengunjung perpustaakaan. Observasi bertujuan untuk mengetahui kondisi lapangan.

Setelah kebutuhan pengguna didapatkan, langkah selanjutnya adalah menggambarkan aplikasi yang akan dikembangkan. Gambaran aplikasi ini bertujuan untuk mengetahui gambaran awal mengenai aplikasi dan fitur-fitur apa saja yang ada diaplikasi yang dikembangkan.

Setelah mendeskripsikan aplikasi yang akan dikembangkan, langkah selanjutnya adalah mengumpulkan dan menganalisis data. Langkah ini diperlukan agar karekteristik dari masing-masing data diketahui. Dengan mengetahui karakteristik data, bisa diketahui pula data mana yang dibutuhkan data mana yang tidak dibutuhkan.

3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Proses perancangan aplikasi terdiri dari perancangan fungsional sistem yang dimodelkan dalam diagram use case. Kemudian, dilakukan perancangan perangkat lunak menggunakan diagram UML *(Unified Modeling Language)* berbasis *object oriented* yang terdiri dari *activity digram, sequence diagram* dan *class digram*. Perancangan UML diawali dengan, perancangan *activity diagram* untuk memodelkan aktivitas antara pengguna dan sistem yang berjalan berdasarkan skenario use case. Selanjutnya, dilakukan

perancangan *sequence diagram* untuk menggambarkan interaksi atau aliran pesan antar objek saat menjalankan aktivitas dari suatu *use case* yang digambarkan sesuai urutan waktu. Setelah perancangan *sequence diagram*, dilakukan perancangan class diagram untuk memberikan gambaran pemodelan elemen-elemen class yang membentuk sebuah perangkat lunak. *Class-class* yang ada pada *class diagram* ditentukan atau ditemukan dengan cara memeriksa objek-objek dan method-method dalam *sequence diagram*.

Setelah perancangan diagram UML dilakukan, langkah selanjutnya adalah perancangan arsitektur sistem. Perancangan arsitektur aplikasi menjelaskan tentang arsitektur aplikasi. Arsitektur yang digunakan pada Aplikasi Pencarian dan Penawaran Buku menggunakan pola arsitektur *MVC(Model-View-Controller)* dengan menggunakan teknologi *Geolocation API. Geolocation* berfungsi untuk melacak lokasi pengguna sistem yang mengakses melalui jaringan komunikasi yang digunakan user untuk mengetahui lokasi terdekat pengguna yang menawarkan bukunya.

Setelah perancangan arsitektur, langkah selanjutnya adalah perancangan basis data. Perancangan basis data adalah melakukan perancangan data dari data yang sudah dikumpulkan dan dianalisis pada proses analisis kebutuhan kemudian dimodelkan ke dalam *object diagram*. Langkah selanjutnya yaitu melakukan perancangan layout responsive untuk menerapkan teknik *Responsive Web Design*, setelah itu dilakukan perancangan antarmuka aplikasi. Perancangan antarmuka merupakan perancangan awal yang akan digunakan dalam implementasi antarmuka pengguna pada Aplikasi Pencarian dan Penawaran Buku.

Langkah terakhir yaitu melakukan peracangan pengujian atau *design uji*. Perancangan pengujian bertujuan untuk mendesain kasus uji atau *test case* yang akan diujikan pada tahap pengujian sistem untuk mengetahui kesesuaian hasil yang diharapkan dengan hasil pengujian pada tahap pengujian sistem.

3.5 Implementasi

Tahap Implementasi bertujuan untuk menghasilkan produk awal yang selanjutnya diuji untuk memastikan apakah hasilnya sesuai dengan yang diharapkan. Implementasi perangkat lunak mengacu pada perancangan perangkat

lunak yang sudah dilakukan. Tahap implementasi perangkat lunak diawali dengan melakukan penjabaran spesifikasi lingkungan perancangan perangkat lunak. Kemudian dilanjutkan implementasi basis data menggunakan database MySQL. Setelah implemtasi basis data dilakukan, tahap selanjunya adalah implementasi penulisan kode program menjadi sebuah aplikasi. Implementasi perangkat lunak pada aplikasi ini, menggunakan *framework CodeIgniter* dengan menerapkan teknik *Responsive Web Design (RWD)*. Implementasi dilakukan dengan menggunakan bahasa pemograman PHP dan *JavaScript* dengan menggunakan arsitektur *Model-View-Controller (MVC).* Implementasi antarmuka berdasarkan perancangan yang telah dilakukan menggunakan HTML5 dan CSS3.

3.6 Pengujian Perangkat Lunak

Pengujian dari perangkat lunak dilakukan untuk mengetahui apakah kinerja dan performa aplikasi telah memenuhi spesifikasi kebutuhan ada di analisis kebutuhan. Strategi pengujian perangkat lunak yang akan digunakan yaitu pengujian validasi (*validation),* pengujian *layout web* responsif *(responsive test),* pengujian pengujian perhitungan jarak manual dan aplikasi, yang terakhir adalah Pengujian kompatibilitas sistem terhadap berbagai jenis browser. Pada tahap pengujian validasi digunakan metode *black-box testing* untuk menguji kebutuhan fungsional sistem sudah terpenuhi sesuai hasil yang diharapkan. Pengujian *layout web* responsif untuk mengetahui apakah antarmuka aplikasi sudah dapat berjalan *responsive* saat diakses dalam lingkungan perangkat dengan ukuran layar yang berbeda. Pengujian perhitungan jarak manual aplikasi bertujuan untuk menganalisis akuransi metode Haversine Formula dalam pencarian lokasi terdekat berdasarkan jarak lokasi menggunakan teknologi *geolocation.* Pengujian jarak manual dilakukan dengan menghitung jarak tempuh antar lokasi menggunakan *speedometer*. Pengujian kompatibilitas sistem untuk menguji apakah aplikasi sudah dapat menjalankan kebutuhan fungsional sistem dengan baik pada browser yang berbeda sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian sistem.

Setelah tahap pengujian, dilakukan analisis sistem untuk membandingkan hasil yang diharapkan dari analisis kebutuhan dan perancangan pengujian apakah

sudah sesuai dengan hasil sistem yang sudah diimplementasikan sehingga didapatkan suatu kesimpulan.

3.7 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakannya penulisan serta memberikan pertimbangan atas pengembangan aplikasi selanjutnya. Perancangan aplikasi Pencarian dan Penawaran Buku diharapkan dapat menjadi refrensi untuk pengembangan selajutnya.

BAB IV
ANALISIS DAN PERANCANGAN

Bab ini membahas mengenai analisis kebutuhan dan perancangan aplikasi. Analisis dan perancangan aplikasi memiliki dua tahap. Tahap pertama adalah proses analisis kebutuhan perangkat lunak dilanjutkan dengan tahap kedua yaitu proses perancangan perangkat lunak. Tahap analisis kebutuhan terdapat beberapa langkah yaitu melakukan penjabaran gambaran umum aplikasi, identifikasi aktor, penjabaran daftar kebutuhan lalu dimodelkan dalam diagram *use case* dan diagram *activity*. Proses perancangan perangkat lunak memiliki tahapan yaitu perancangan arsitektural, perancangan basis data, pemodelan diagram *sequence* untuk menggambarkan interaksi antar object di dalam aplikasi, pemodelan diagram *class* untuk menggambarkan perancangan struktur *class-class* yang ada pada aplikasi kemudian dilanjutkan perancangan antarmuka pengguna dan terakhir adalah perancangan pengujian sistem.

4.1 Analisis Kebutuhan Sistem

Untuk menganalisis kebutuhan aplikasi, penulis melakukan obsevasi langsung dengan melihat informasi kritik dan saran dari pengunjung perpustakaan Universitas Brawijaya dan perpustakaan umum kota Malang. Berdasarkan hasil observasi terhadap informasi kritik dan saran perpustakaan Universitas Brawijaya dan perpustakaan umum kota Malang, ada beberapa pengunjung perpustakaan yang mengeluhkan ketidaklengkapan koleksi yang dibutuhkan dan jumlah buku yang masih kurang tidak sebanding jumlah peminjam.

Keanekaragaman jenis perangkat yang dimiliki masyarakat mulai dari komputer *desktop*, *notebook* sampai perangkat bergerak dengan platform yang berbeda-beda seperti Androis, Ios, Blackberry dan lain-lain. Hal tersebut, menyulitkan seorang programmer untuk membuat sebuah aplikasi dalam versi *fullsite* (desktop) dan *mobile* dengan platform yang berbeda-beda. Perkembangan perangkat bergerak seperti *tablet* dan *smartphone* yang semakin memudahkan pengguna untuk mengakses informasi berbasis web. Akses halaman web dengan menggunakan perangkat bergerak seperti *tablet* dan *smartphone* menyulitkan

pengguna saat mengakses peta pada halaman web dikarenakan ukuran layar yang jauh lebih kecil dibandingkan layar *desktop* komputer. Konten halaman web akan terlihat sangat kecil jika diakses melalui *smartphone*, sehingga harus diperbesar dulu tampilannya kemudian harus sering melakukan *scroll*.

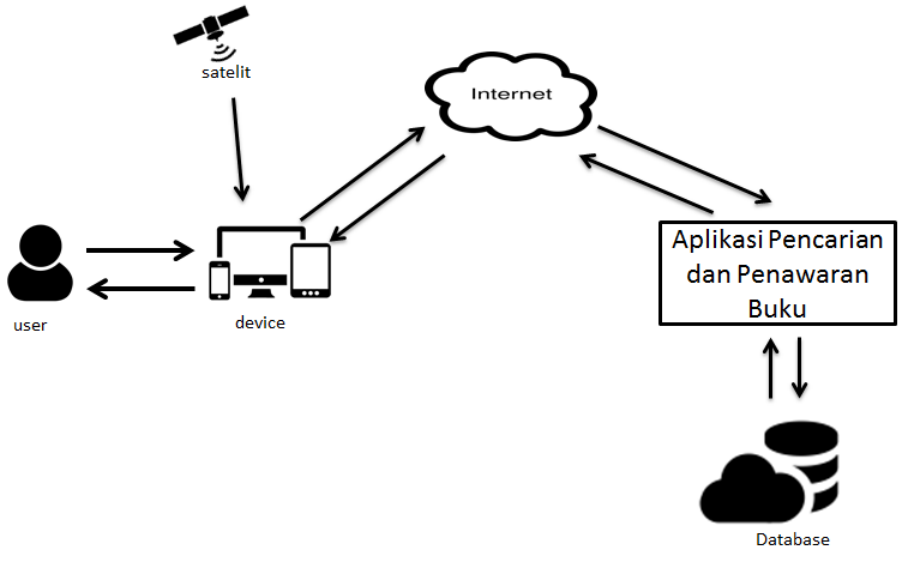
Dari analisis permasalahan tersebut, penulis merasa perlu adanya sebuah media informasi pencarian dan penawaran buku yang dapat mendukung beberapa ukuran layar perangkat mulai dari komputer *desktop* sampai perangkat *mobile*. Cara untuk membuat tampilan halaman web yang dapat menyesuaikan ukuran layar pada berbagai jenis perangkat yang dimiliki *user* yaitu menggunakan teknik *Responsive Web Design*. *Responsive Web Design* adalah sebuah teknik yang digunakan untuk membuat *layout* web menyesuaikan dengan tampilan *device* atau perangkat yang digunakan pengunjung web baik ukuran maupun orientasi, sehingga tampilan yang berada di desktop komputer dengan tampilan di perangkat bergerak atau *mobile device* akan tetap sama nyamannya dengan tampilan halaman web tersebut saat ditampilkan di *desktop* komputer.

Proses analisis kebutuhan ini diawali dengan penjabaran gambaran umum aplikasi, identifikasi aktor, penjabaran daftar kebutuhan dan kemudian dimodelkan kedalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Gambaran Umum Aplikasi

Aplikasi Pencarian dan Penawaran Buku adalah aplikasi yang dirancang sebagai media informasi pencarian dan penawaran buku-buku bekas yang sudah tidak terpakai untuk dijual atau disewakan. Disamping itu, aplikasi ini juga dapat melakukan pencarian penawaran buku yang berada di dekat area lokasi user berdasarkan jarak lokasi terdekat beserta rute lokasi yang ditampilkan melalui Google Maps API dengan memanfaatkan teknologi *Geolocation*. Pemanfaatan *Geolocation* dan Haversine Formula dalam aplikasi bertujuan agar sistem dapat melakukan perhitungan seberapa jarak kita terhadap suatu object, kemudian dengan *Google Direction Service* pengguna sistem juga dimungkinkan untuk mengetahui jalur yang bisa ditempuh untuk mencapai lokasi tersebut.

Aplikasi dirancang untuk digunakan di beberapa ukuran layar perangkat bergerak ataupun desktop PC yang terhubung dengan internet. Oleh sebab itu, dibutuhkan sebuah teknik *Responsive Web Design* pada desain *layout* aplikasi web agar dapat berjalan responsif sesuai dengan ukuran *layout device* atau perangkat yang digunakan. Sehingga, tampilan web yang berada di notebook dengan tampilan di perangkat bergerak akan tetap sama nyamannya dengan tampilan halaman web tersebut saat ditampilkan di *notebook* atau laptop. Gambaran umum sistem ditunjukan pada gambar 4.1



Gambar 4.1 Gambaran Umum Sistem

Gambar 4.1 merupakan gambaran umum aplikasi pencarian dan penawaran buku. Pengguna mengakses aplikasi melalui *web browser* pada perangkat desktop maupun perangkat bergerak yang terhubung melalui internet. Apabila perangkat terintegrasi melalui GPS maka perangkat akan menerima sinyal satelit yang memanfaatkan fungsi GPS untuk mendapatkan data koordinat perangkat pengguna saat itu. Perintah pengguna akan dikirimkan ke aplikasi dan aplikasi akan mendapatkan data yang dibutuhkan dari database. Untuk melakukan pencarian buku, pengguna masuk ke halaman pencarian dan memasukan data pencarian buku. Selain itu, pengguna dapat melakukan pencarian berdasarkan terdekat dengan lokasi user. Kemudian, aplikasi melakukan pencarian data yang dimasukan dari database. Jika buku yang dicari ada, pengguna dapat melakukan request buku kepada user yang menawarkan bukunya tersebut. Untuk melakukan request penawaran buku, pengguna harus login terlebih dahulu, pada halaman

request pengguna dapat melakukan nego harga dari buku yang ditawarkan. Jika berhasil, maka buku yang kita pesan akan masuk ke *history* permintaan saya dan akan dikirimkan kepada user pemilik buku tersebut. User pemilik buku akan menerima notifikasi, jika terdapat request dari pengguna lain terhadap buku yang ditawarkannya tersebut. Kemudian user pemilik buku dapat melakukan konfirmasi permintaan buku untuk menyetujui atau menolak permintaan dari user merequest buku tersebut. Jika permintaan disetujui, user dapat melakukan pengaturan transaksi lewat fitur pesan yang tersedia. Selanjutnya, jika transaksi penawaran buku sudah mencapai kesepakatan maka user pemilik dapat merubah status buku tersebut menjadi terjual atau sedang disewakan.

4.1.2 Identifikasi Aktor

Pada tahap ini bertujuan melakukan identifikasi terhadap *user* yang berhubungan dengan sistem dan aplikasi yang dibuat. Penjelasan masing-masing *user* dilihat pada Tabel 4.1.

Tabel 4.1 Identifikasi aktor

Aktor	Deskripsi
Admin	Administrator atau admin merupakan pengguna yang bertugas untuk melakukan manipulasi data jika diperlukan. Didalam sistem, admin bertugas untuk mengelola data user dan data buku.
Non Member	*Non Member* merupakan pengguna aplikasi yang dapat melakukan akses terhadap sistem tanpa harus login terlebih dahulu dengan akses berupa melihat detail penawaran buku, melakukan pencarian buku dan melihat lokasi user yang menawarkan bukunya.
Member	Member merupakan pengguna yang sudah terdaftar dan melakukan login untuk dapat mengakses sistem. Pengguna ini memiliki akses untuk dapat menawarkan atau mengiklankan buku bekas yang dimilikinya ke dalam sistem. Selain itu, aktor ini dapat melakukan request penawaran buku yang ada pada sistem, melakukan konfirmasi request yang masuk dari user lain, mengirim pesan transaksi, mengubah status penawaran buku jika buku yang ditawarkan sudah laku terjual atau sedang disewakan serta dapat mengubah statusnya kembali jika buku sudah dikembalikan.

4.1.3 Analisis Kebutuhan Fungsional

Identifikasi kebutuhan didapat dari analisis permasalahan. Kebutuhan fungsional sistem harus dimodelkan ke dalam diagram *use case*, dan setiap kebutuhan fungsional harus dilakukan pengujian terlebih dahulu sebelum sistem dipublikasikan. Spesifikasi kebutuhan fungsional aplikasi ditunjukan pada Tabel 4.2.

Tabel 4.2 Spesifikasi kebutuhan aplikasi

Nomor	Aktor	Deskripsi kebutuhan	Usecase
SRS_01	Non Member	Sistem harus menyediakan fasilitas untuk melihat detail informasi penawaran buku dan informasi lokasi pengguna yang menawarkan bukunya.	Melihat Detail Informasi Penawaran Buku
SRS_02	Non Member	Sistem harus menyediakan fasilitas untuk menampilkan informasi lokasi pengguna saat itu dan menampilkan lokasi-lokasi pengguna yang menawarkan buku.	Melihat peta lokasi
SRS_03	Non Member	Sistem harus menyediakan fasilitas untuk melakukan pencarian buku dengan ditampilkan jarak lokasi sehingga pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya.	Pencarian Buku
SRS_04	Non Member	Sistem harus menyediakan fasilitas untuk menampilkan informasi lokasi menuju lokasi pengguna yang menawarkan bukunya atau memesan buku kepada user lain.	Melihat Rute Lokasi
SRS_05	Member	Sistem harus menyediakan fitur registrasi untuk menyimpan data pengguna yang akan menawarkan bukunya ke dalam sistem.	Registrasi
SRS_06	Member	Sistem harus menyediakan fasilitas	Memposting

		untuk menambahkan atau memposting data penawaran buku baru	Penawaran Buku
SRS_07	Member	Sistem menyediakan fasilitas untuk melakukan update data penawaran buku dan hapus data penawaran buku yang dimiliki aktor	Mengelola data penawaran buku
SRS_08	Member	Sistem menyediakan fasilitas untuk melakukan request penawaran buku kepada pengguna yang menawarkan buku tersebut.	Request buku yang ditawarkan
SRS_09	Member	Sistem menyediakan fasilitas untuk dapat melakukan konfirmasi request masuk dari user lain yang menawar buku yang kita tawarkan pada sistem.	Konfirmasi request masuk
SRS_10	Member	Sistem harus menyediakan fasilitas untuk mengubah status penawaran buku jika buku sudah terjual, sedang dipinjamkan atau sudah dikembalikan	Mengubah Status Penawaran Buku
SRS_11	Member	Sistem menyediakan fasilitas mengirim pesan antar user untuk mengatur transaksi penyewaan atau pembelian buku.	Mengirim Pesan
SRS_12	Member	Sistem menyediakan fasilitas untuk melihat notifikasi atau pemberitahuan masuk jika ada request baru atau pesan baru serta konfirmasi request terhadap buku yang dipesan.	Melihat Notifikasi
SRS_13	Admin	Sistem menyediakan fasilitas untuk melihat data user dan melakukan manipulasi data user jika diperlukan	Mengelola User
SRS_14	Admin	Sistem harus menyediakan fasilitas untuk melihat semua daftar data penawaran buku yang dilakukan oleh	Mengola data penawaran buku

		user dan melakukan manipulasi data penawaran buku jika diperlukan.	

4.1.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan *non fungsional* adalah analisis untuk mengetahui spesifikasi yang dibutuhkan oleh sistem agar dapat berjalan secara sempurna. Kebutuhan non fungsional tersebut adalah *usability dan compatibility* yang dijelaskan pada tabel 4.3

Tabel 4.3 Kebutuhan non fungsional

Parameter	**Deskripsi Kebutuhan**
Compatibility	Aplikasi harus dapat menjalankan fungsinya pada beberapa jenis browser yang berbeda pada perangkat *desktop* maupun perangkat bergerak *(mobile)*

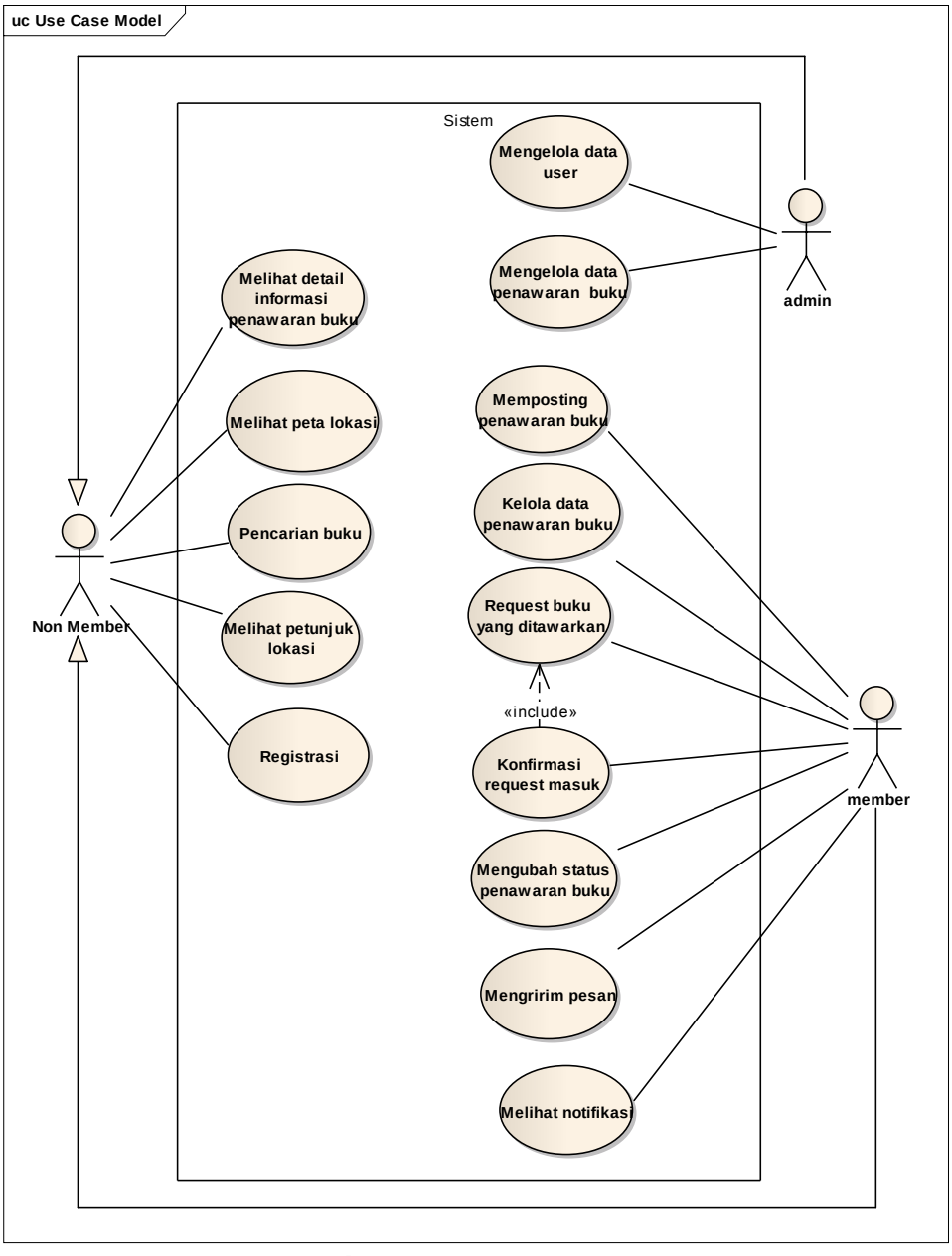
4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dalam empat tahap yaitu perancangan diagram UML *(Unified Modelling Language)* menggunakan pendekatan desain berorientasi objek yang terdiri dari *activity diagram*, *sequence diagram* dan *class diagram*. Tahap selanjutnya adalah perancangan arsitektur, perancangan basis data dan perancangan antar muka.

4.2.1 Diagram Use Case dan Activity

Pemodelan diagram use case sistem diperoleh dari kebutuhan fungsional yang digunakan untuk menggambarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. merupakan diagram usecase Aplikasi Pencarian dan Penawaran Buku yang terdiri dari tiga aktor yaitu user non member, user member dan Admin. Aktor non member dapat melakukan akses terhadap sistem tanpa harus login terlebih dahulu dengan akses berupa melihat detail penawaran buku, melakukan pencarian buku dan melihat lokasi user yang menawarkan bukunya. User member dapat melakukan posting penawaran buku yang dimilikinya atau melakukan request penawaran buku dari pengguna lain, melakukan konfirmasi permintaan buku yang masuk dan mengubah data penawaran jika penawaran buku sudah laku terjual atau sedang disewakan atau sudah dikembalikan. Aktor member memiliki relasi generalisasi dengan aktor non

member yang berarti aktor member juga dapat menjalankan aktifitas setiap use case yang dilakukan oleh aktor non member. Sedangkan admin dapat melakukan akses untuk mengelola data user dan data buku.



Gambar 4.2 Diagram *use case*

Berdasarkan diagram *use case* yang ditunjukan pada akan lebih dijelaskan secara terperinci dalam skenario *use case.* Scenario *use case* bertujuan untuk mendapatkan deskripsi secara global mengenai use case, kondisi awal dan akhir yang harus dipenuhi oleh *use case* setelah fungsionalitas telah dijalankan. Terdapat beberapa bagian penting dari skenario use case berdasarkan perancangan diagram use case diatas yaitu skenario *use case* pencarian buku yang ditunjukan

pada , skenario use case memposting penawaran buku yang ditunjukan pada , skenario use case request penawaran buku ditunjukan pada Tabel 4.6.

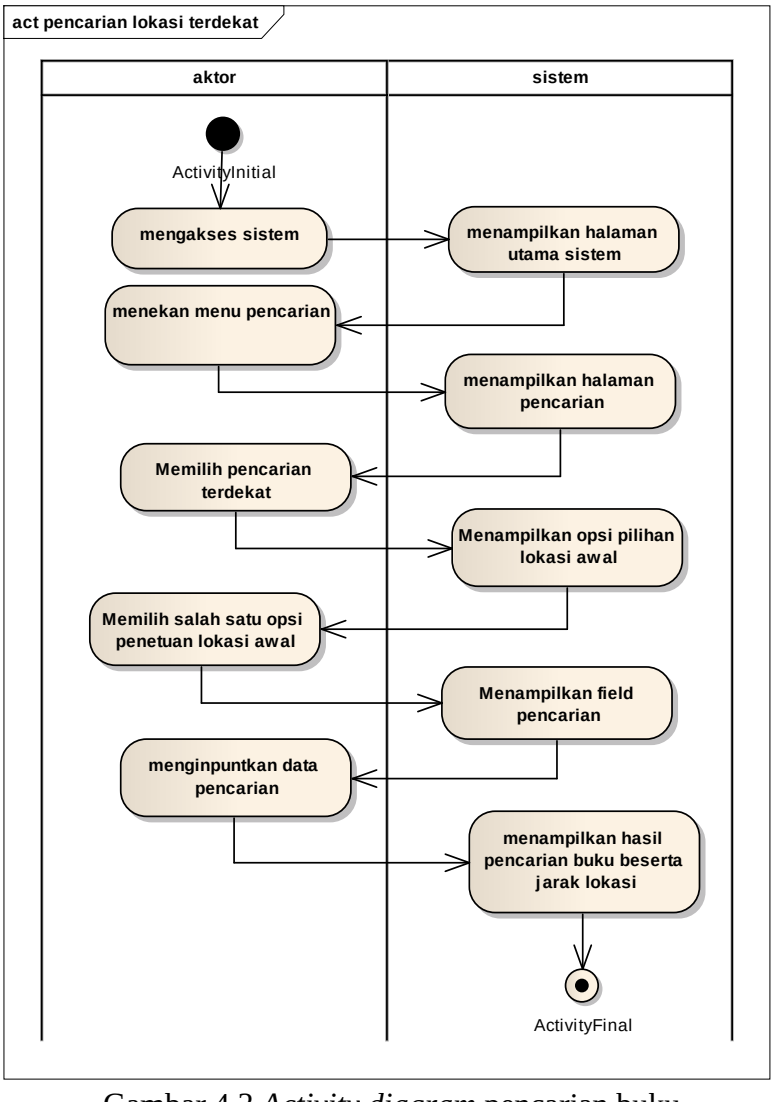
adalah skenario *use case* pencarian buku. Pada skenario *use case* pencarian buku akan dijelaskan lebih mendetail mengenai proses pencarian penawaran buku beserta jarak lokasi dengan alamat pemilik buku.

Tabel 4.4 Skenario use case pencarian buku

Identifikasi	
Nomor use case	SRS_03
Nama	Pencarian buku
Tujuan	Untuk melakukan pencarian buku yang menampilkan jarak lokasi sehingga aktor dapat menemukan penawaran buku yang dekat dengan lokasinya
Deskripsi	Use case ini memungkinkan aktor untuk melakukan menemukan penawaran buku yang dekat dengan lokasinya berdasarkan jarak terdekat
Aktor	*User* Non Member
Kondisi awal	Aktor harus mengakses sistem
Kondisi akhir	Hasil list penawaran buku akan muncul berdasarkan paramenter pencarian yang di inputkan.
Proses skenario pencarian	
Aksi aktor	Reaksi sistem
	1. Sistem menampilkan halaman utama aplikasi
2. Aktor memilih menu “cari buku”	3. Sistem menampilkan halaman pencarian
4. Aktor memilih “pilih lokasi terdekat”	5. Sistem Menampilkan opsi pilihan lokasi awal “lokasi saat ini” atau “lokasi tertentu”
6. Aktor memilih salah opsi untuk menentukan lokasi awal	7. Sistem menampilkan field pencarian berdasarkan opsi lokasi awal yang dipilih
8. Aktor menginputkan data pencarian	9. Sistem menampilkan hasil pencarian beserta jarak lokasi sesuai masukan dari aktor

menjelaskan kegiatan atau aktivitas yang dilakukan aktor pada saat melakukan pencarian penawaran buku yang berada disekitar area lokasi user berdasarkan jarak terdekat. Kondisi awal untuk melakukan pencarian lokasi terdekat yaitu aktor mengakses aplikasi dan masuk halaman utama. Pada halaman

utama, terdapat beberapa pilihan menu utama. Selanjutnya, aktor memilih menu cari buku. Sistem akan menampilkan halaman pencarian. Selanjutnya, aktor memilih “pilih lokasi terdekat”. Sistem akan opsi pilihan lokasi awal yaitu berdasarkan “lokasi saat ini” atau “lokasi tertentu”. Selanjutnya aktor memilih salah satu pilihan opsi kemudian sistem akan menampilkan field pencarian berdasarkan opsi yang dipilih. Pada field pencarian, aktor memasukan data pencarian buku yang dicari selanjutnya sistem akan menampilkan hasil pencarian buku beserta jarak lokasi sesuai masukan dari user.



Gambar 4.3 *Activity diagram* pencarian buku

merupakan aktivitas diagram pencarian buku, yang digunakan untuk menjelaskan interaksi antara aktor dengan sistem. Pada *Activity diagram* pencarian buku menjelaskan interaksi *user* dan sistem untuk melakukan pencarian buku yang menampilkan jarak lokasi sehingga aktor dapat menemukan penawaran buku yang dekat dengan lokasinya dijelaskan

pada skenario *use case* pada . Skenario use case pada dapat dimodelkan kedalam diagram *activity* pada . Untuk melakukan pencarian buku, pertama pengguna mengakses sistem lalu aplikasi akan menampilkan halawan awal sistem. Pada halaman awal aplikasi menampilkan pilihan menu utama aplikasi kemudian aktor memilih menu pencarian. Sistem akan menampilkan halaman daftar pencarian. Selanjutnya, aktor menginputkan data buku yang dicari kemudian sistem akan menampilkan hasil pencarian buku berserta jarak lokasi sehingga aktor dapat menemukan penawaran buku yang dekat dengan lokasinya. merupakan skenario *use case* memposting penawaran buku. Pada skenario memposting penawaran buku dijelaskan lebih mendetail mengenai proses melakukan posting penawaran buku baru.

Tabel 4.5 Skenario use case memposting penawaran buku

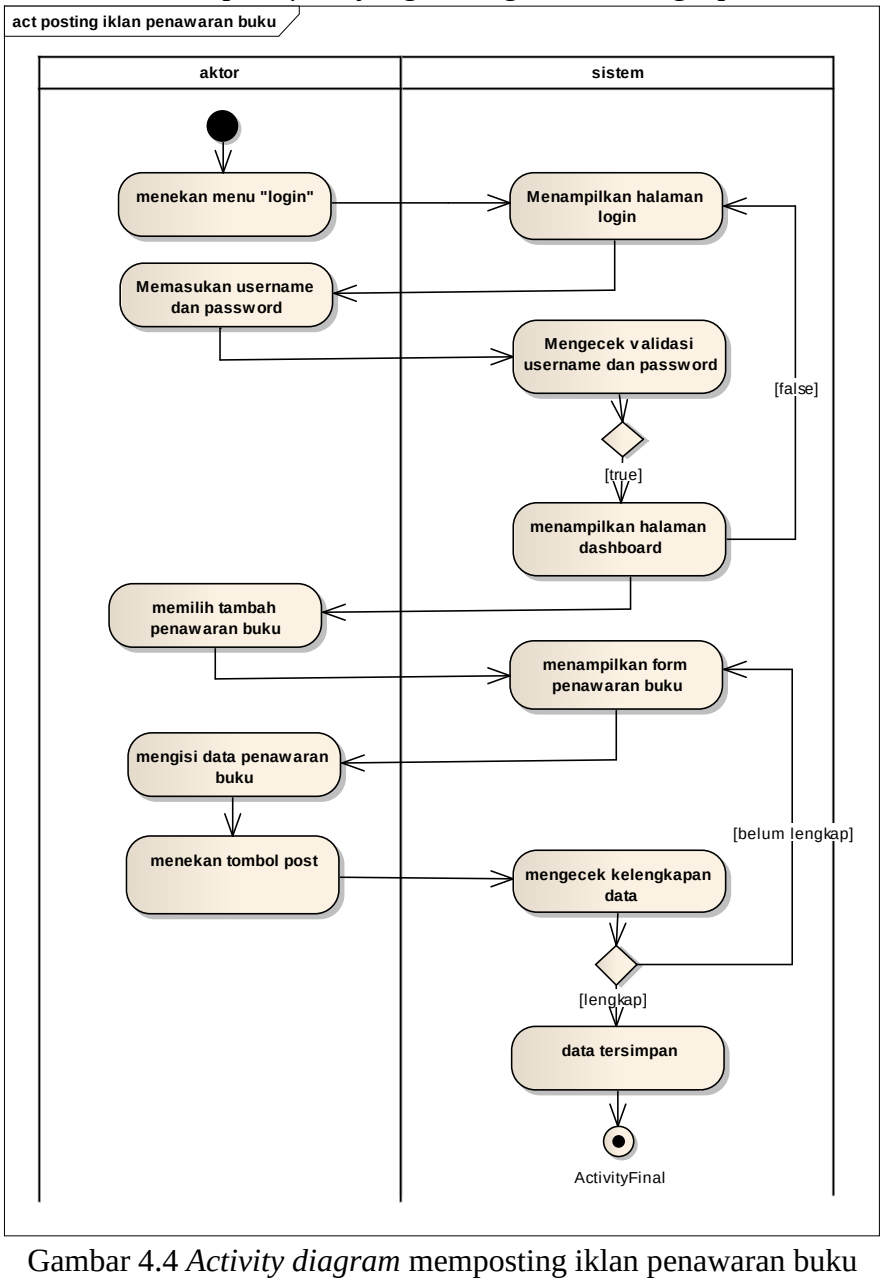
Identifikasi	
Nomor use case	SRS_06
Nama	Memposting Penawaran Buku
Tujuan	Melakukan posting penawaran buku baru
Deskripsi	Use case ini memungkinkan pengguna untuk melakukan posting penawaran buku baru.
Aktor	User Member
Kondisi awal	Aktor harus masuk halaman login
Kondisi akhir	Aplikasi menampilkan notifikasi bahwa data penawaran buku yang dimasukan berhasil disimpan ke dalam database.
Skenario utama (*basic flow*)	
Aksi aktor	Reaksi sistem
	1. Menampilkan halaman login
2. Memasukan username dan password	
	3. Mengecek validasi username dan password
	4. Sistem menampilkan halaman dashboard
5. Aktor menekan tombol “tambah penawaran”	6. Sistem menampilkan form tambah penawaran
7. Aktor memasukan data penawaran buku sesuai isian *field* yang diminta	
8. Aktor menekan tombol “post penawaran”	9. Sistem memeriksa kelengkapan isian data

	10. Sistem menyimpan data penawaran buku baru ke dalam database
Skenario Aternatif : Jika *user* belum melengkapi data dan data tidak valid	
Aksi aktor	Reaksi sistem
	1. Sistem akan menampilkan form dengan menampilkan tanda merah pada *field* yang kosong untuk dilengkapi
2. Aktor mengisi *form* yang belum lengkap kemudian di-submit.	3. Sistem menyimpan data yang sudah dilengkapi
	4. Kembali ke skenario utama *(basic flow)* poin 10

menjelaskan kegiatan atau aktivitas yang dilakukan *user* untuk melakukan posting iklan penawaran buku yang dimiliki oleh *user*. Kodisi awal untuk melakukan posting iklan penawaran buku, *user* harus sudah terdaftar pada sistem dan masuk halaman login. Kemudian, sistem akan menampilkan halaman login dan meminta *user* untuk memasukan *username* dan *password.* User memasukan username dan password kemudian sistem akan mengecek validasi username dan password yang diinputkan oleh *user*. Jika username dan password benar maka sistem akan menampilkan halaman utama aplikasi.

Untuk melakukan posting penawaran buku, user memilih tombol tambah penawaran. Kemudian sistem akan menampilkan form tambah penawaran. Selanjutnya, user menginputkan data buku yang akan diposting sesuai data yang diminta pada form tambah penawaran lalu menekan tombol “post penawaran” jika sudah selesai memasukan data. Selanjutnya, sistem akan memerikan kelengkapan *field-field* sebelum disimpan. Jika semua field sudah lengkap sistem akan menyimpan data penawaran buku dan memberikan notifikasi bahwa data penawaran buku berhasil tersimpan. Tetapi jika masih terdapat *field* yang konsong pada form penawaran, sistem akan menampilkan form dengan menampilkan tanda

merah pada *field* yang kosong untuk dilengkapi.



Gambar 4.4 *Activity diagram* memposting iklan penawaran buku

merupakan aktivitas diagram memposting iklan penawaran buku yang digunakan untuk menjelaskan interaksi antara pengguna dengan sistem. Pada *Activity diagram* posting iklan penawaran buku menjelaskan interaksi *user* dan sistem untuk memposting iklan penawaran buku yang dimiliki oleh *user*. *Activity diagram* posting iklan penawaran buku merupakan model diagram aktivitas dari skenario use case . Pada ditunjukan proses aktivitas aktor untuk melakukan posting iklan penawaran buku. Pertama, pengguna menjalankan mengkases sitem dan menekan menu *login*, kemudian aplikasi akan menampilkan halaman login. User memasukan username dan password kemudian sistem akan mengecek

validasi username dan password yang diinputkan oleh *user*. Jika username dan password benar maka sistem akan menampilkan halaman utama aplikasi.

Untuk melakukan posting penawaran buku, user masuk pada halaman *dashboard* dengan cara memilih menu "dashboard". Selanjutnya sistem akan menampilkan halaman menu dashboard yang menampilkan daftar pernawaran buku yang dimiliki *user*. Kemudian *user* memilih tombol "tambah penawaran", sistem akan menampilkan form tambah penawaran. Selanjutnya, user menginputkan data buku yang akan diposting sesuai data yang diminta pada form tambah penawaran lalu menekan tombol "post penawaran" jika sudah selesai input data. Selanjutnya, sistem akan memerikan kelengkapan *field-field* sebelum disimpan. Jika semua field sudah lengkap sistem akan menyimpan data penawaran buku dan memberikan notifikasi bahwa data penawaran buku berhasil tersimpan. Tetapi jika masih terdapat *field* yang konsong pada form penawaran, sistem akan menampilkan form dengan menampilkan tanda merah pada *field* yang kosong untuk dilengkapi.

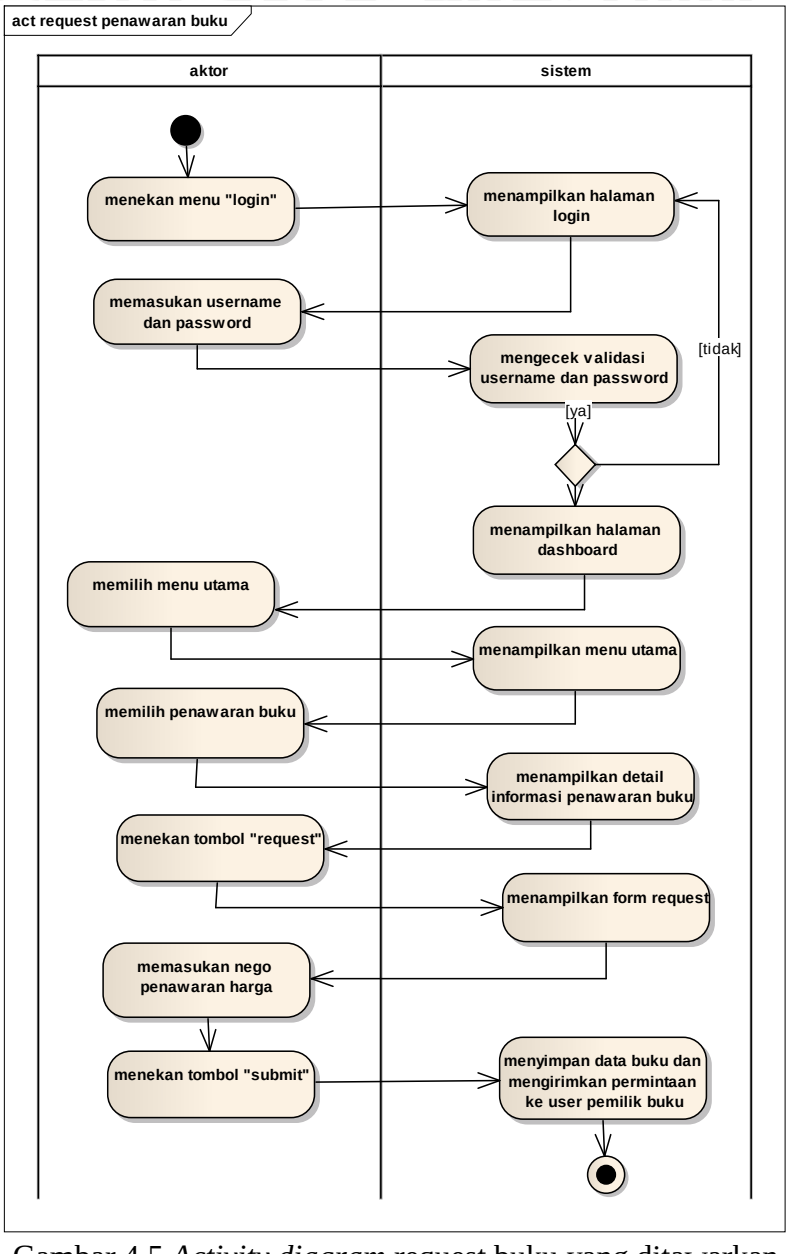
Tabel 4.6 merupakan skenario *use case* request buku yang ditawarkan. Pada skenario request penawaran buku dijelaskan lebih mendetail mengenai proses melakukan request buku yang ditawarkan pada sistem.

Tabel 4.6 Skenario use case request buku yang ditawarkan

Identifikasi	
Nomor use case	SRS_08
Nama	Request buku yang ditawarkan
Tujuan	Melakukan request atau pemesanan buku yang ditawarkan kepada user pemilik buku tersebut.
Deskripsi	Use case ini memungkinkan pengguna untuk melakukan request buku yang ditawarkan kepada user pemilik buku tersebut.
Aktor	User Member
Kondisi awal	Aktor harus masuk halaman login
Kondisi akhir	Aplikasi akan mengirimkan data permintaan buku dan menyimpan data permintaan pada history "permintaan saya"
Skenario utama (*basic flow*)	
Aksi aktor	Reaksi sistem
	1. Menampilkan halaman login
2. Memasukan username dan	

password	
	3. Mengecek validasi username dan password
	4. Sistem menampilkan halaman dashboard
5. Aktor masuk pada halaman utama sistem	6. Sistem menampilkan halaman utama aplikasi
7. Aktor memilih buku yang direquest atau dipesan	8. Sistem menampilkan detail informasi buku yang dipilih
9. Aktor menekan tombol "request"	10. Menampilkan form request
11. Memasukan nego penawaran harga jika status harga buku dapat dinego	
12. Menekan tombol "submit"	13. Sistem menyimpan data permintaan buku ke dalam history "permintaan saya" dan mengirimkan permintaan request buku kepada user pemilik buku tersebut.

Tabel 4.6 menjelaskan kegiatan atau aktivitas yang dilakukan *user* untuk melakukan request atau pemesanan buku yang ditawarkan kepada user pemilik buku tersebut. Kodisi awal untuk melakukan posting iklan penawaran buku, *user* harus sudah terdaftar pada sistem dan masuk halaman login. Kemudian, sistem akan menampilkan halaman login dan meminta *user* untuk memasukan *username* dan *password.* User memasukan username dan password kemudian sistem akan mengecek validasi username dan password yang diinputkan oleh *user.* Jika username dan password benar maka sistem akan menampilkan halaman utama aplikasi. Untuk melakukan *request* atau pemesanan buku, user masuk ke halaman utama aplikasi. Kemudian sistem akan menampilkan halaman utama aplikasi. Selanjutnya, pengguna memilih buku yang ingin dipesan kemudian sistem akan menampilkan detail informasi buku yang dipilih. Pada halaman detail informasi buku, pengguna menekan tombol "request" maka sistem akan menampilkan from penawaran harga jika status harga buku yang ditawarkan dapat dinego kemudian menekan tombol "submit". Sistem akan menyimpan data permintaan buku kedalam database dan mengirikamnya kepada user pemilik buku tersebut.



Gambar 4.5 *Activity diagram* request buku yang ditawarkan

Gambar 4.5 merupakan aktivitas diagram memposting iklan penawaran buku yang digunakan untuk menjelaskan interaksi antara pengguna dengan sistem. Pada *Activity diagram* posting iklan penawaran buku menjelaskan interaksi *user* dan sistem untuk memposting iklan penawaran buku yang dimiliki oleh *user*. *Activity diagram* posting iklan penawaran buku merupakan model diagram aktivitas dari skenario use case Tabel 4.6. Pada Gambar 4.5 ditunjukan proses aktivitas aktor untuk melakukan *request* atau pemesanan buku yang ditawarakan. Pertama, pengguna menjalankan mengakses sistem dan menekan menu *login*, kemudian aplikasi akan menampilkan halaman login. User memasukan username dan password kemudian sistem akan mengecek validasi

username dan password yang diinputkan oleh *user*. Jika username dan password benar maka sistem akan menampilkan halaman dashboard aplikasi.

Untuk melakukan request penawaran buku, user masuk pada halaman beranda dengan cara memilih menu "home". Selanjutnya sistem akan menampilkan halaman beranda yang menampilkan daftar pernawaran buku yang ditawarkan pada sistem. Selanjutnya, pengguna memilih buku yang ingin dipesan kemudian sistem akan menampilkan detail informasi buku yang dipilih. Pada halaman detail informasi buku, pengguna menekan tombol "request" maka sistem akan menampilkan from penawaran nego harga jika status harga buku yang ditawarkan dapat dinego kemudian menekan tombol "submit". Sistem akan menyimpan data permintaan buku kedalam database dan mengirikamnya kepada user pemilik buku tersebut.

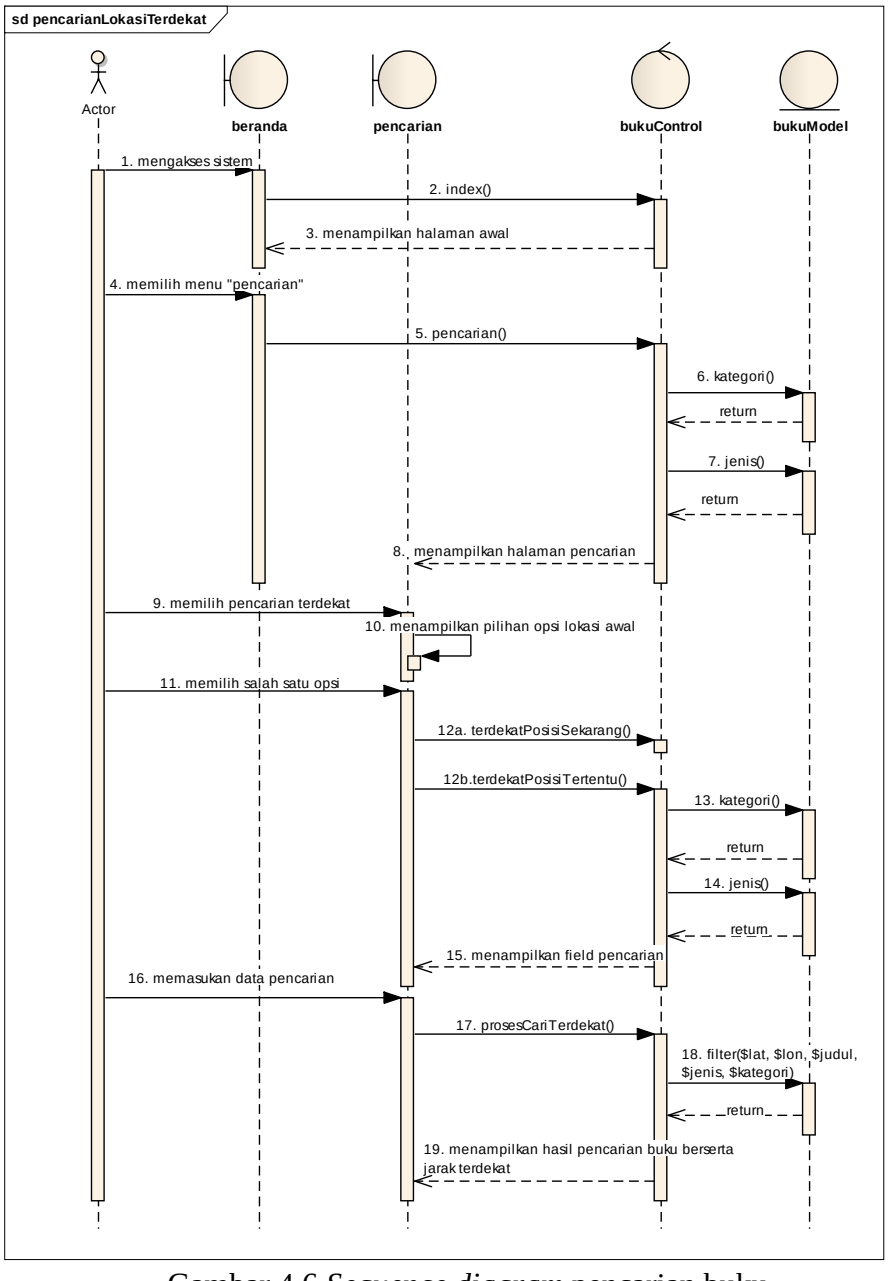
User pemilik buku akan mendapatkan notifikasi atau pemberitahuan jika buku yang ditawarkan direquest oleh pengguna lain. User pemilik buku tersebut, dapat melakukan konfirmasi persetujuan atau penolakan terhadap request atau pemesanan buku yang masuk. Skenario konfirmasi request buku yang masuk ditunjukan pada lampiran 1 dan aktivitas konfirmasi request masuk ditunjukan pada lampiran 4. Jika request atau pemesanan buku disetujui, user dapat melakukan pengaturan transaksi lewat fitur pesan yang tersedia. Skenario mengirim pesan ditunjukan pada lampiran 2 dan aktivitas mengirim pesan ditunjukan pada lampiran 5. Selanjutnya, jika transaksi penawaran buku sudah mencapai kesepakatan maka user pemilik dapat merubah status buku tersebut menjadi terjual atau sedang disewakan. Untuk skenario ubah status penawaran buku ditunjukan pada lampiran 3 dan aktivitas untuk mengubah status penawaran buku ditujukan pada lampiran 6

4.2.2 Sequence Diagram

Sequence diagram menjelaskan interaksi atau aliran pesan antar objek saat menjalankan aktivitas dari suatu *use case* yang digambarkan sesuai urutan waktu. *Sequence diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-object yang terkait). Pada aplikasi ini terdapat beberapa *Sequence Diagram* yaitu *Sequence Diagram* Pencarian lokasi penawaran buku terdekat yang ditunjukan pada , *Sequence Diagram* memposting penawaran buku yang

ditunjukan pada , *Sequence Diagram* melihat informasi request atau pemensanan buku yang ditawarkan ditunjukan pada Gambar 4.8 .

Pada *sequence diagram* pencarian buku terdekat menjelaskan aliran pesan antar objek untuk melakukan aktivitas pencarian buku terdekat yang dimodelkan pada *activity diagram* pada . *Sequence diagram* pencarian buku ditunjukan pada .



Gambar 4.6 *Sequence diagram* pencarian buku

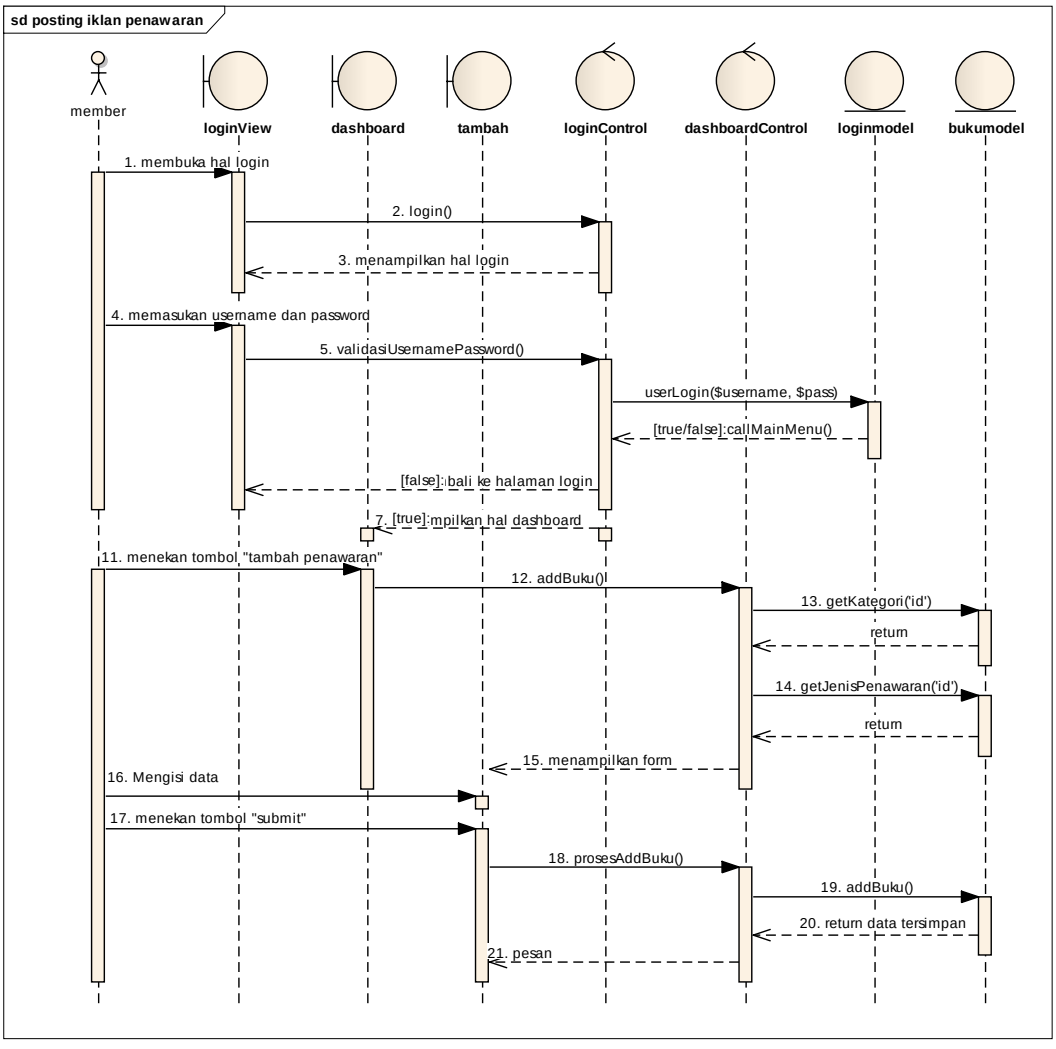
merupakan *sequence diagram* untuk melakukan pencarian penawaran buku berdasarkan penawaran yang dekat dengan lokasi *user*. Pertama, pada saat *user* mengakses sistem aplikasi akan menjalankan method *index(),* melalui *controller* akan memanggil object *beranda* yang menampilkan halaman utama

aplikasi. Selanjutnya, *user* memilih menu "pencarian", kemudian aplikasi menjalankan method *pencarian()* lalu *controller* akan melakukan pengambilan data melalui method *kategori()* dan *jenis()* pada *class bukuModel*. Selanjutnya, *controller* akan memanggil object *pencarian* yang menampilkan halaman pencarian buku. Pada halaman pencarian buku, *aktor* memilih *link* pencarian terdekat, sistem akan menampilkan pilihan opsi penentuan lokasi awal. Aktor memilih salah satu opsi yang tersedia kemudian sistem akan menjalankan *method terdekatPosisiSekarang()* jika user memilih opsi terdekat berdasarkan posisi saat ini atau sistem akan menjalankan method *terdekatPosisiTertentu()* jika user memilih opsi terdekat berdasarkan posisi yang ditentukankan. Sistem mendapatkan data latitude dan longitude lokasi user, selanjutnya data akan dikirim dan dilakukan proses perhitungan jarak melalui method *filter()* pada class *bukuModel*. Hasil pencarian berserta perhitungan jarak lokasi akan ditampilkan pada object *pencarian.*

Pada *Sequence diagram* posting iklan penawaran buku menjelaskan aliran pesan antar objek dalam aktivitas posting penawaran buku yang dimodelkan pada *activity diagram* pada . *Sequence diagram* posting iklan penawaran buku yang ditunjukkan pada Gambar 4.7. Untuk melakukan posting penawaran buku baru, user harus melakukan login terlebih dahulu. Pada saat user membuka halaman login aplikasi akan menjalan method *login(),* melalui *controller* akan memanggil object *loginView* yang menampilkan halaman login. Selanjutnya, u*ser* memasukan *username* dan *password* kemudian aplikasi menjalankan method *validasiUsernamePassword()* pada *loginControl* untuk mengecek validasi username dan password yang tersimpan pada database melalui method *userLogin($username,$pass)* pada *loginModel.* Jika username dan password benar, aplikasi akan menampilkan halaman *dashboard* aplikasi berisi data penawaran buku.

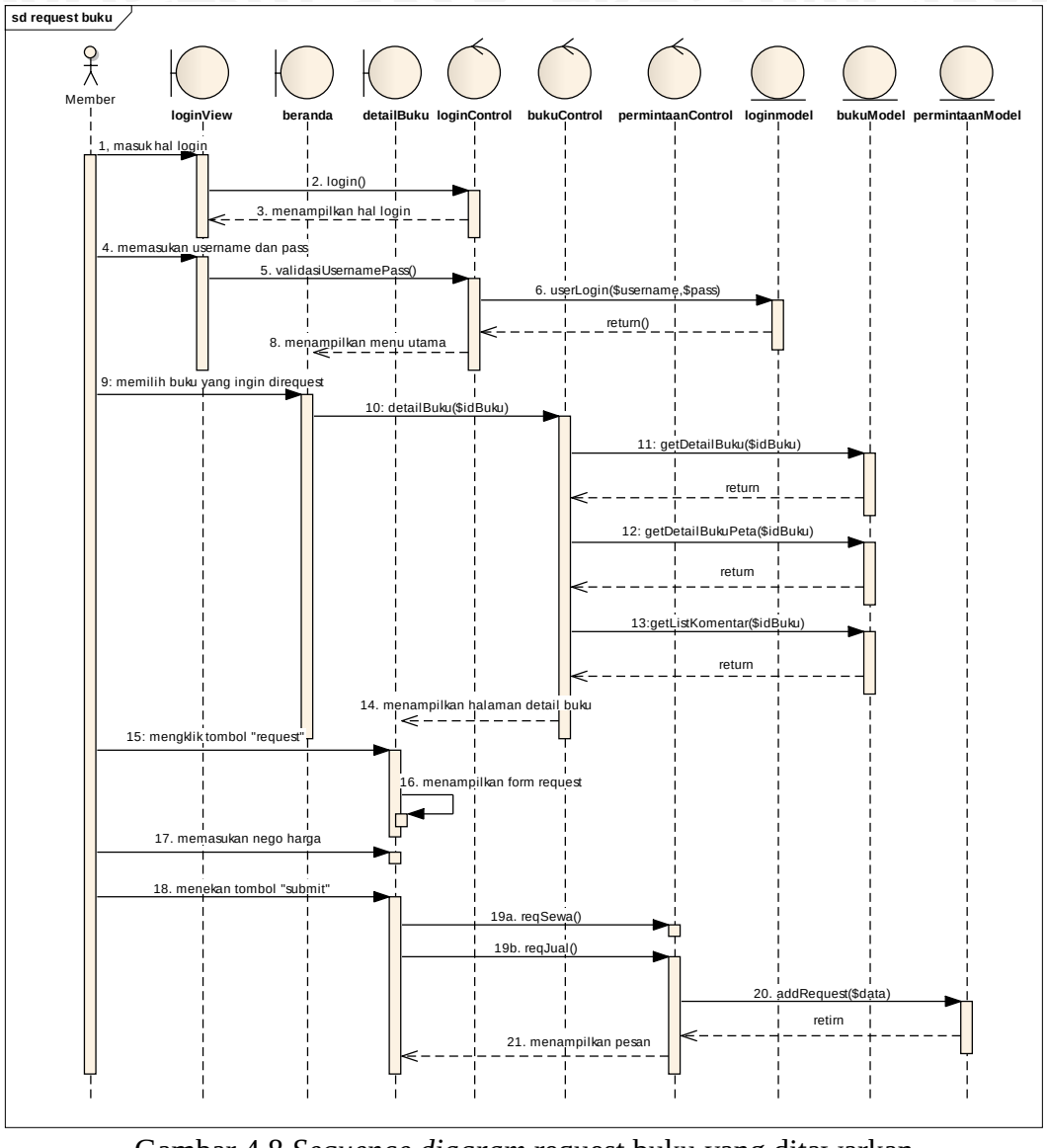
Pada halaman dashboard user memilih tombol "tambah penawaran" kemudian *controller* akan menjalankan method *addBuku()* yang memanggil object *tambah* untuk menampilkan form tambah penawaran buku dan melakukan pengambilan data melalui method *getKategori()* dan *getJenisPenawaran().* Kemudian user mengisi data buku lalu menekan tombol "submit" kemudian

controller akan menjalankan method *prosesAddBuku()* dan menyimpannya kedalam database melalui method *addBuku()* pada class *bukuModel*. Jika berhasil, aplikasi menampilkan pesan bahwa data berhasil disimpan.



Gambar 4.7 *Sequence diagram* memposting iklan penawaran buku

Pada *Sequence diagram* request buku yang ditawarkan menjelaskan aliran pesan antar objek dalam aktivitas request buku yang ditawarkan dimodelkan pada *activity diagram* pada Gambar 4.5. *Sequence diagram* request penawaran buku ditunjukkan pada Gambar 4.8.



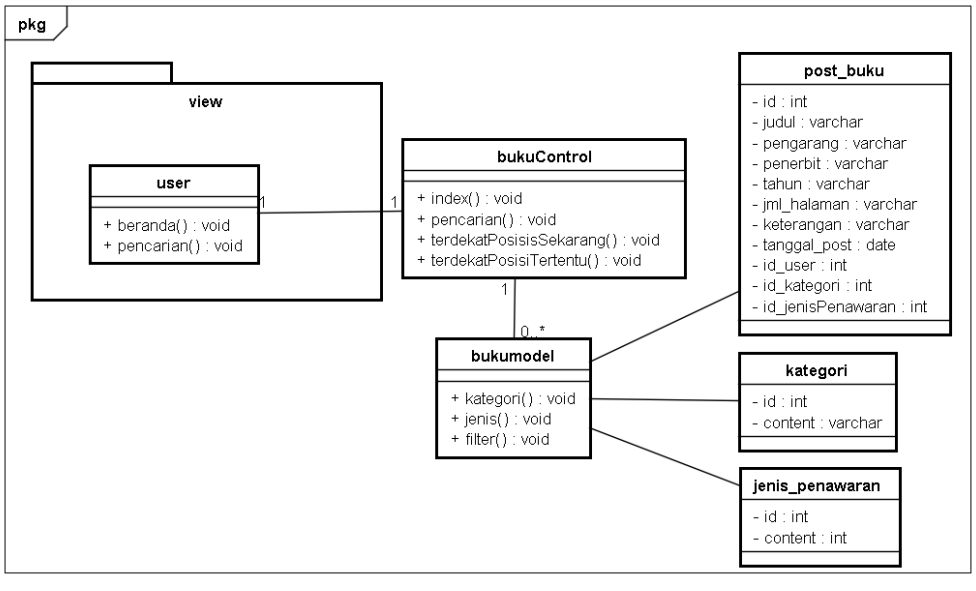
Gambar 4.8 *Sequence diagram* request buku yang ditawarkan

Pada Gambar 4.8 dijelaskan aliran pesan antar object untuk melakukan request penawaran buku baru, user harus melakukan login terlebih dahulu. Pada saat user membuka halaman login aplikasi akan menjalan method *login(),* melalui *controller* akan memanggil object *loginView* yang menampilkan halaman login. Selanjutnya, u*ser* memasukan username dan password kemudian aplikasi menjalankan method *validasiUsernamePassword()* pada *loginControl* untuk mengecek validasi username dan password yang tersimpan pada database melalui method *userLogin($username,$pass)* pada *loginModel.* Jika username dan password benar, aplikasi akan menampilkan halaman dashboard aplikasi berisi data penawaran buku.

Selanjutnya, user memilih menu "home" melalui controller akan memanggil object *beranda*. Pada halaman beranda pengguna memilih buku yang ingin direquest kemudian sistem akan menjalankan method *detailBuku()* pada class *bukuControl* untuk menampilkan detail informasi penawaran buku dan melakukan pemagilan data pada class *bukuModel*. Pengguna menekan tombol "request", sistem akan menampilkan form request. Selanjutnya pengguna menginputkan penawaran nego harga jika status harga buku dapat dinego. Sistem akan menjalankan method *reqSewa()* jika jenis penawaran "disewakan" atau *reqJual()* jika jenis penawaran buku "dijual" dan menyimpan data request buku kedalam database melalu method *addRequest()* pada class *permintaanModel*.

4.2.3 Class Diagram

Class diagram memberikan gambaran pemodelan elemen-elemen *class* yang membentuk sebuah perangkat lunak. *Class-class* ditentukan atau ditemukan dengan cara memeriksa objek-objek dan *method-method* dalam *sequence diagram*. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

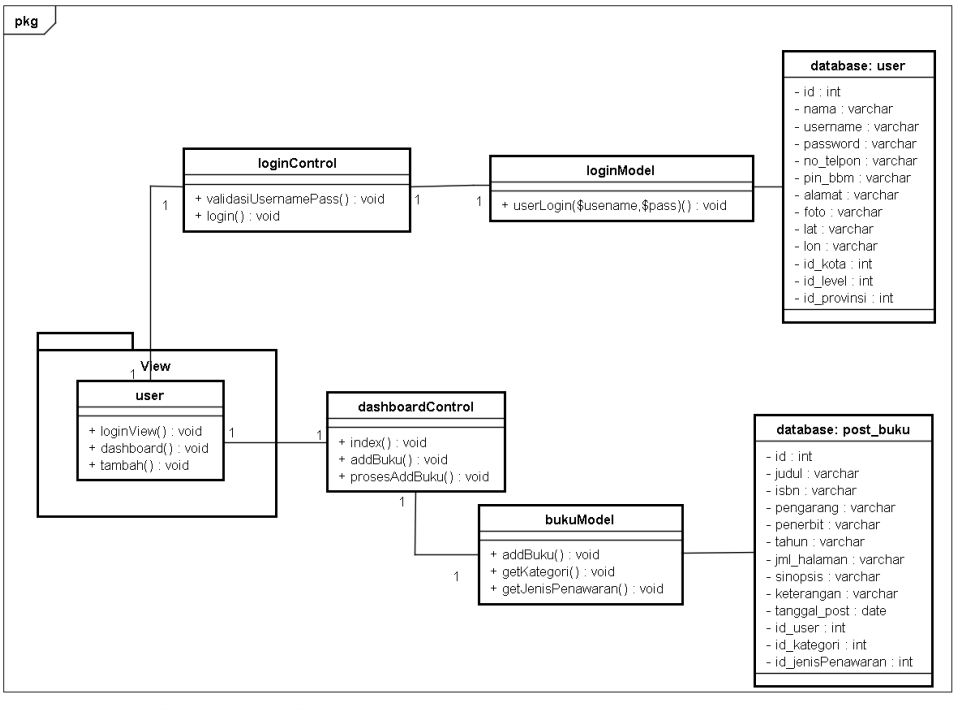


Gambar 4.9 *Class diagram* pencarian buku

menjelaskan tentang *class-class* yang terdapat pada aktivitas pencarian buku terdekat. Struktus *class-class* yang ada pada ditentukan dengan cara memeriksa object-object dan method-method yang terdapat dalam perancangan *sequence diagram* . Pada *class user* merupakan kelas yang menangani tampilan ke pengguna. *Class user* memiliki relasi asosiasi dengan class *bukuControl* yang

berfungsi untuk melakukan proses pembacaan dan pencarian data buku yang kemudian ditampilkan pada *view*. *Class bukuControl* memiliki relasi asosiasi dengan class *bukuModel* yang berfungsi untuk melakukan pengambilan data buku dari database pada tabel post_buku, kategori dan jenis penawaran.

Gambar 4.10 merupakan class-class yang terdapat dalam aktivitas memposting iklan penawaran buku. Struktus *class-class* yang ada pada Gambar 4.10 ditentukan dengan cara memeriksa object-object dan method-method yang terdapat dalam perancangan *sequence diagram* Gambar 4.7.



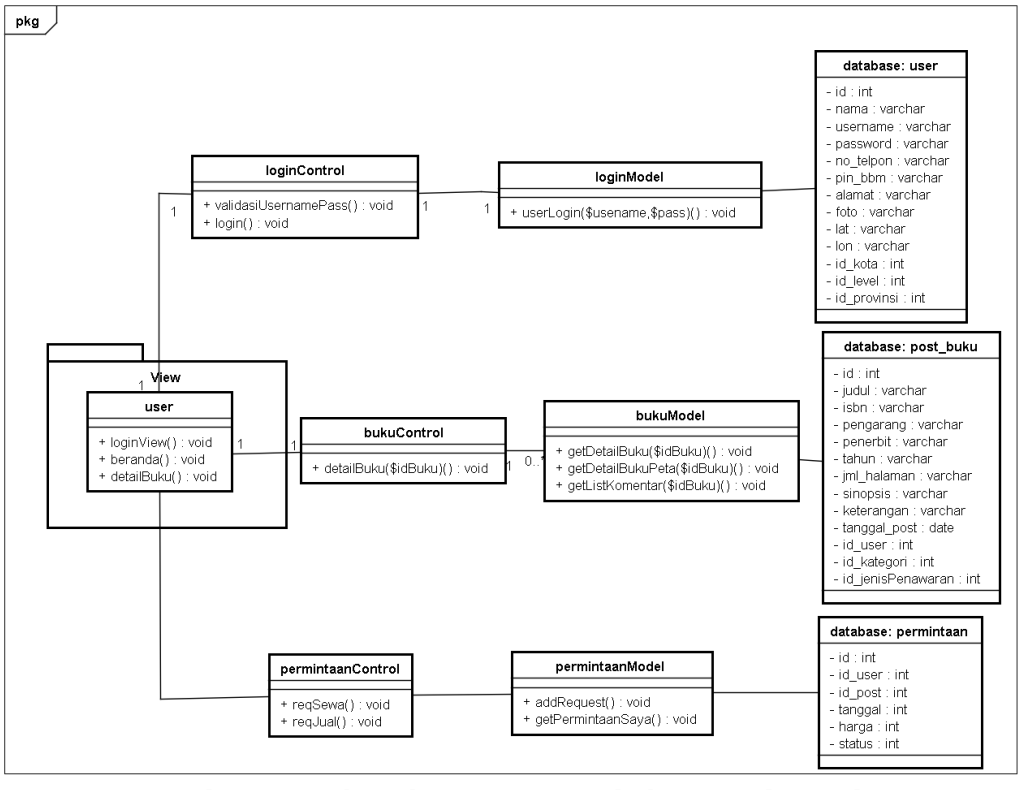
Gambar 4.10 *Class diagram* posting iklan penawaran buku

Pada perancangan *class diagram* posting penawaran buku, terdapat class *user* merupakan kelas yang menangani tampilan ke pengguna. *Class user* memiliki relasi asosiasi dengan *class loginControl* yang berfungsi melakukan validasi username dan password yang tersimpan dalam database. *Class loginControl* memiliki relasi asosiasi dengan *class loginModel* yang berfungsi untuk melakukan pengecekan data login user yang diambil dari database tabel user.

Setelah user berhasil melakukan login, sistem akan menampilkan halaman utama melalui *class user*. *Class user* juga memiliki relasi asosiasi dengan class *dashboardControl* yang berfungsi untuk melakukan proses penambahan data

penawaran buku baru. *Class dashboardControl* memiliki relasi asosiasi dengan *class bukuModel* yang berfungsi untuk menyimpan data ke dalam database.

Gambar 4.11 merupakan class-class yang terdapat dalam aktivitas request atau pemesanan buku yang ditawarkan. Struktur *class-class* yang ada pada Gambar 4.11 ditentukan dengan cara memeriksa object-object dan method-method yang terdapat dalam perancangan *sequence diagram* Gambar 4.8.



Gambar 4.11 *Class diagram* request buku yang ditawarkan

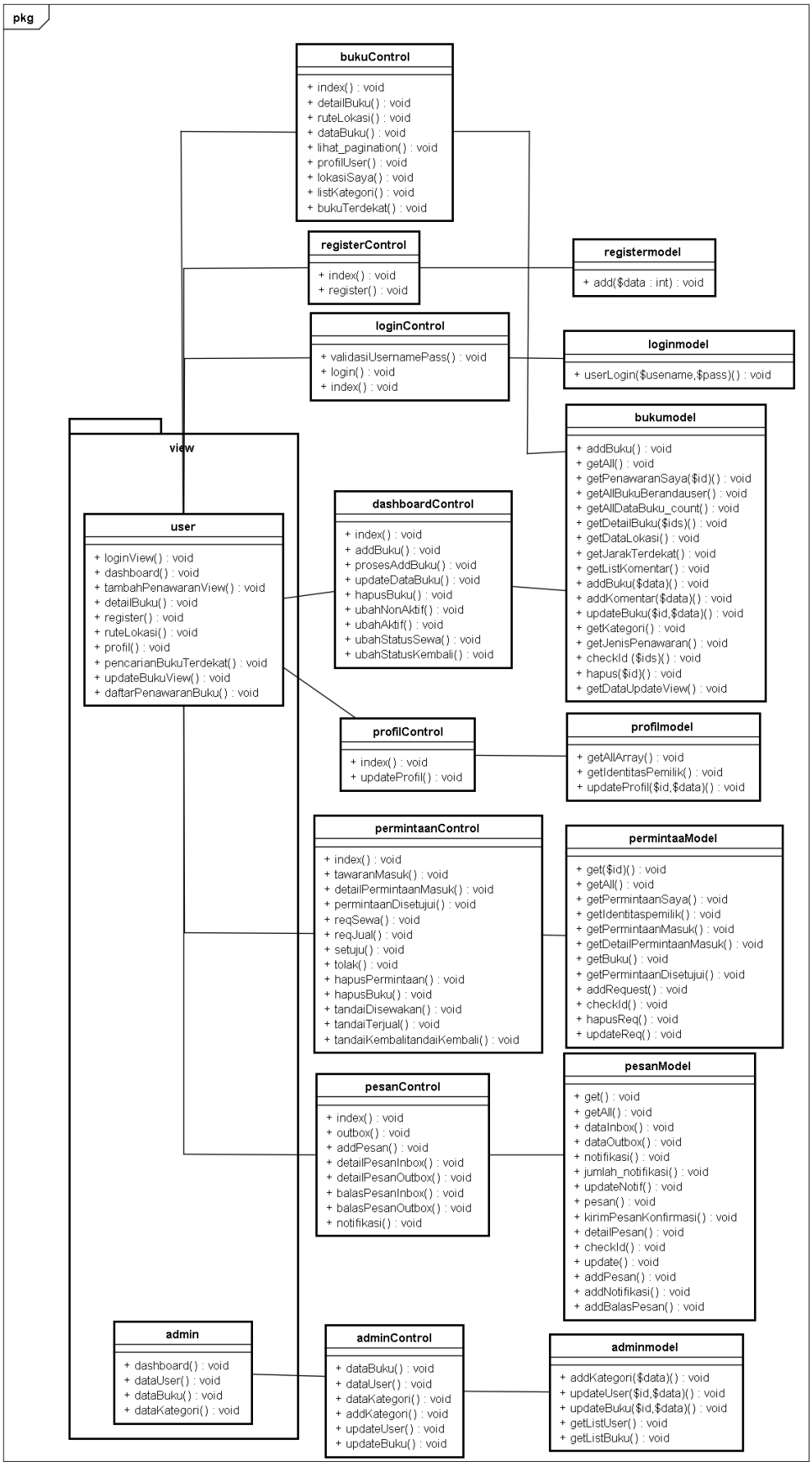
Pada perancangan *class diagram* request penawaran buku pada Gambar 4.11, terdapat class *user* merupakan kelas yang menangani tampilan ke pengguna. *Class user* memiliki relasi asosiasi dengan *class loginControl* yang berfungsi melakukan validasi username dan password yang tersimpan dalam database. *Class loginControl* memiliki relasi asosiasi dengan *class loginModel* yang berfungsi untuk melakukan pengecekan data login user yang diambil dari database tabel user.

Setelah user berhasil melakukan login, sistem akan menampilkan halaman utama melalui *class user*. *Class user* juga memiliki relasi asosiasi dengan class *bukuControl* yang berfungsi untuk melakukan proses pemanggilan halaman detail buku melalui fungsi *detailBuku()*. *Class bukuControl* memiliki relasi asosiasi

dengan *class bukuModel* yang berfungsi untuk mengambil data informasi detail buku dari database pada tabel *post_buku*.

Berdassarkan , Gambar 4.10, Gambar 4.11 dan class-class yang terbentuk dari daftar kebutuhan sistem yang tidak ditampilkan pada dokumen ini, maka didapatkan pemodelan class keseluruhan sistem untuk menggambarkan perancangan struktur-struktur class yang menyusun aplikasi yang ditunjukan pada .

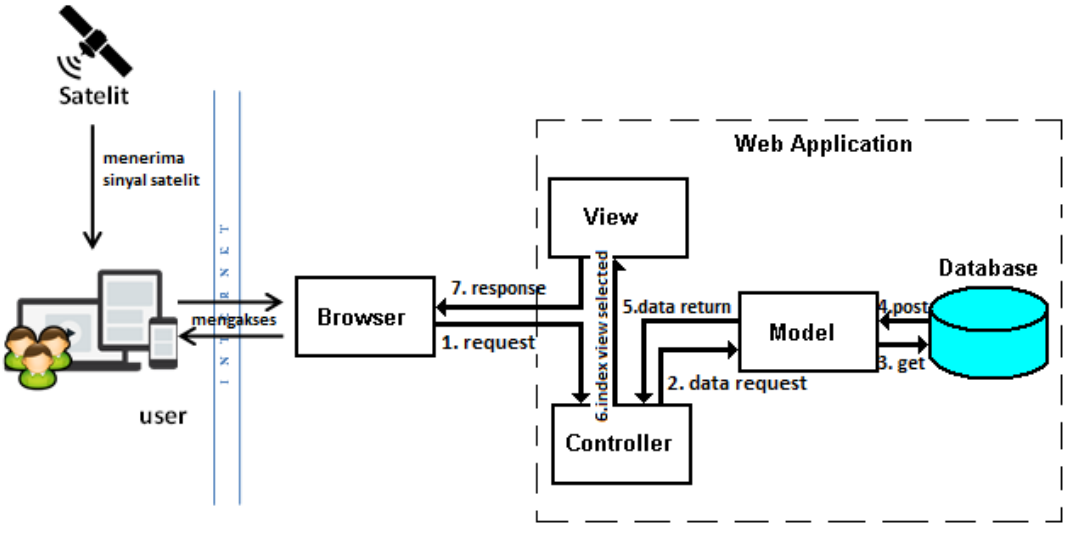
merupakan gabungan class dari seluruh aktivitas yang terdapat pada Aplikasi Pencarian dan Penawaran Buku. Terdapat *class user* merupakan kelas yang menangani tampilan ke pengguna dan class admin yang menangani tampilan pada halaman admin. Class *user* memiliki relasi asosiasi dengan beberapa class controller yaitu class *bukuControl*, class *registerControl*, class *loginControl*, class *bukuControl*, class *profilControl*, *dashboardControl*, *permintaanControl*, *pesanControl*. Class *bukuControl* memiliki relasi asosiasi dengan *class bukuModel* yang berfungsi untuk melakukan pengambilan data buku. Class *registerControl* memiliki relasi asosiasi dengan class *registerModel* yang berfungsi untuk melakukan penyimpanan data user baru kedalam database. Class *dashboardControl* memiliki relasi asosiasi dengan class *bukuModel* yang berfungsi untuk melakukan penyimpanan posting penawaran buku baru. Class *profilControl* memiliki relasi asosiasi dengan class *profilModel* yang berfungsi melakukan pengambilan data user dan data penawaran buku. Class *permintaanControl* memiliki relasi asosiasi dengan class *permintaanModel* yang berfungsi untuk menyimpan data permintaan request buku. Class *pesanControl* memiliki relasi asosiasi dengan class *pesanModel* yang berfungsi melakukan proses pengiriman pesan sebagai komunikasi antar user mengatur transaksi penjualan dan penyewaan buku.



Gambar 4.12 *Class diagram* aplikasi

4.2.4 Perancangan Arsitektur Sistem

Perancangan perangkat lunak Aplikasi Pencarian dan Penawaran Buku dibangun menggunakan teknologi *Geolocation* API. Teknologi *geolocation* digunakan untuk mengetahui informasi lokasi pengguna. Aplikasi dirancang berbasis web menerapkan konsep MVC *(Model-View-Controller)* yang akan diimplementasikan menggunakan framework *CodeIgniter*. MVC adalah arsitektur yang digunakan dalam rekayasa perangkat lunak atau aplikasi dengan memisahkan antara data (Model) dengan *user interface* atau tampilan (View). Berdarkan pola arsitektur MVC, aplikasi memiliki 3 bagian utama yaitu *View* berfungsi sebagai tampilan antar muka yang digunakan untuk berinteraksi dengan sistem. *Model* yang berfungsi sebagai domain data dari aplikasi. *Model* berperan untuk mengambil data dari database atau menyimpan data ke dalam database. *Controller* merupakan jembatan yang menghubungkan antara *model* dan *view*.



Gambar 4.13 Perancangan arsitektur aplikasi

Gambar 4.13 menjelaskan tentang arsitektur sistem Aplikasi Pencarian dan Penawaran Buku. *User* mengakses aplikasi melalui web browser pada perangkat komputer atau perangkat bergerak yang terhubung dengan GPS untuk menerima sinyal dari satelit. Melalui *web browser*, *user* melakukan *request*, yang akan ditangani oleh *controller*. *Controller* memiliki banyak metode yang akan menangani *request*. *Controller* akan mengambil data pada database melalui *model*. *Model* memberikan data yang relevan kepada controller. Kemudian, *controller* akan mengolah data dan memberikannya kepada *view* (tampilan) yang

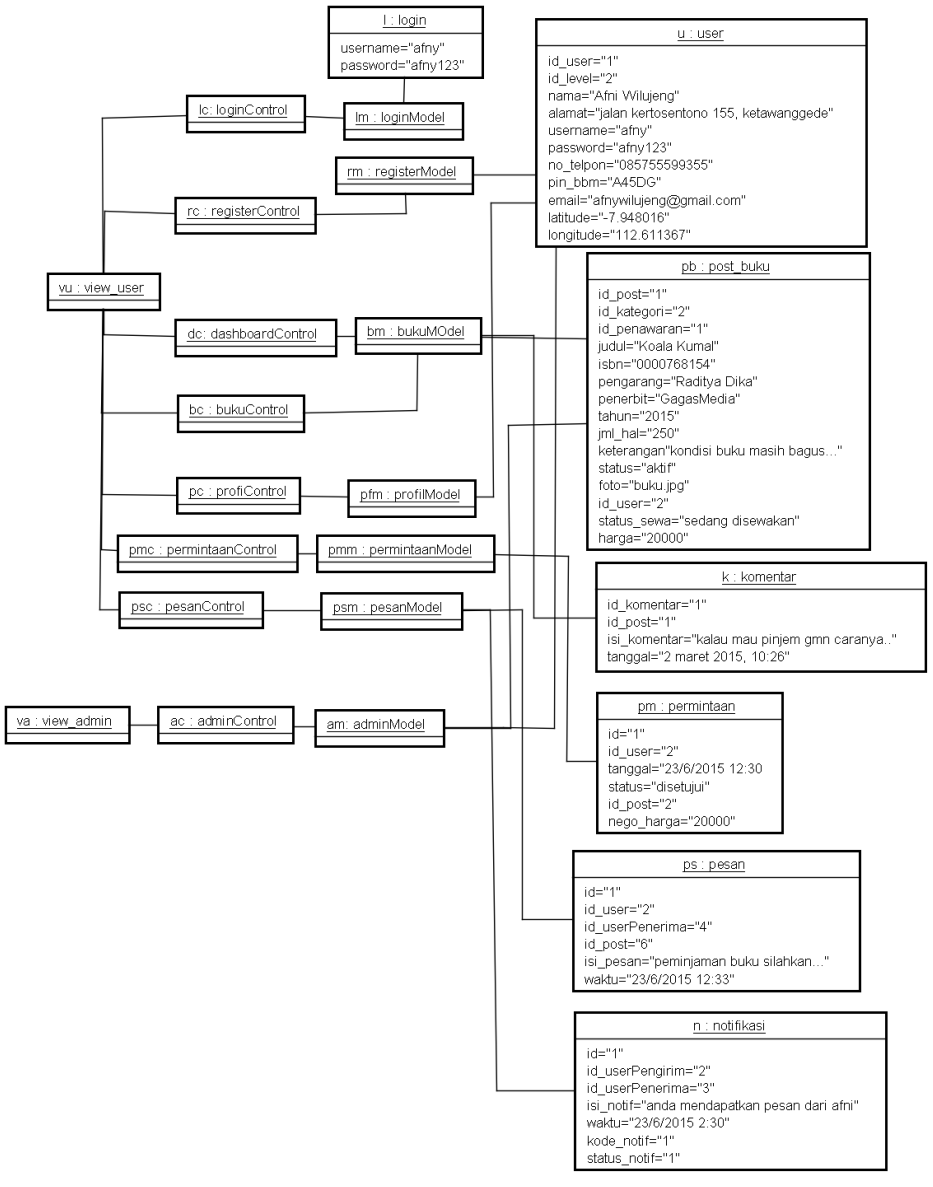
dipilih). Tampilan inilah yang akan dilihat *user* melalui browser yang diakses secara *online* menggunakan jaringan *internet*.

.

4.2.5 Perancangan Basis Data

Perancangan basis data pada Aplikasi Pencarian dan Penawaran Buku berfungsi sebagai media penyimpanan data. Perancangan basis data digunakan untuk merancang data yang akan dibuat agar masukan dan keluaran program sesuai dengan apa yang diharapkan. Perancangan basis data dilakukan dengan cara melakukan proses identifikasi dan analisis kebutuhan-kebutuhan data dari hasil analisis kebutuhan sistem dengan melihat bagian-bagian yang akan berinteraksi dengan sistem, termasuk kebutuhan pengguna dan kebutuhan sistem. Penentuan data disesuaikan dengan kebutuhan sistem yang akan dibangun. Sebagai contoh, untuk melakukan aktivitas posting penanawaran buku, kita dapat menganalisis data yang dibutukan saat pengguna melakukan aktivitas tersebut yaitu data pengguna dan data buku. Data pengguna yang dibutuhkan oleh aplikasi berupa data nama, alamat, no_telpon, pin_bbm, email, foto, username, password, latitude, longitude. Sedangkan, data buku yang dibutuhkan oleh aplikasi berupa data judul, isbn, pengarang, penerbit, tahun, jml_halaman, keterangan, foto.

Perancangan basis pada penelitian ini, direpresentasikan dalam bentuk diagram objek yang ditunjukan pada . merupakan diagram objek yang menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek dipastikan semua *class* sudah didefinisikan pada diagram *class* (Lihat) harus dipakai objeknya, karena jika tidak pendefinisian class tidak dapat dipertanggungjawabkan. Diagram objek berfungsi untuk mendefinisikan contoh nilai atau isi atribut tiap *class*.



Gambar 4.14 Diagram object

Berikut merupakan struktur tabel berserta keterangan masing-masing tabel dan field yang ada pada perancangan database Aplikasi Pencarian dan Penawaran Buku.

Tabel 4.7 Rancangan tabel data user

Data user		
Kolom	**Tipe Data**	**Contoh Data**
Id_user(pk)	Interger	Kode dari user (1,2,3…)
Nama	Varchar	Afni Wilujeng
Alamat	Varchar	Jalan Kertosentono 155, Malang
No_telpon	Varchar	083865411144
Pin_bbm	Varchar	A287AB
Username	Varchar	Afni

Password	Varchar	pass123
Latitude	Varchar	-7.948016
Longitude	Varchar	112.611367
Id_level(fk)	Integer	Kode dari tabel level (1,2,3…)

Tabel 4.7 merupakan perancangan data user yang berfungsi untuk menyimpan data-data user yang melakukan pendanftaran atau regristrasi pada sistem dan menampilkan koordinat *golocation* pengguna. Data user membutuhkan data Id_user, nama, Alamat, No_telpon, Pin_bbm, Foto, Username, Password, Latitude, Longitude yang memiliki tipe data varchar dan integer.

Tabel 4.8 Rancangan tabel data penawaran buku

Data user		
Kolom	**Tipe Data**	**Contoh Data**
Id_post(pk)	Interger	Kode dari buku (1,2,3…)
Id_kategori(fk)	Interger	Kode dari tabel kategori (1,2,3…)
Id_penawaran(fk)	Interger	Kode dari tabel jenis_penawaran (1,2,3…)
Id_user(fk)	Interger	Kode dari tabel user(1,2,3..)
Judul	Varchar	Koala Kumal
Pengarang	Varchar	Raditya Dika
Penerbit	Varchar	GagasMedia
Tahun	Varchar	2015
Jml_halaman	Varchar	250
Keterangan	Varchar	isi dan sampul buku masih dalam keadan bagus dan lengkap
Tgl_post	Varchar	9-03-2015 at 11.12am
Status	Varchar	Aktif atau Tidak Aktif
Foto	Varchar	Foto.jpg
Status Penyewaan	Varchar	Tersedia atau Sedang disewakan

merupakan perancangan data penawaran buku yang berfungsi untuk menyimpan data-data buku yang ditawarkan oleh *user*. Data penawaran buku membutuhkan data Id_post(pk), Judul, Isbn, Pengarang, Penerbit, Tahun, Jml_halaman, Review, Keterangan, Tgl_post, Status, Foto yang memiliki tipe data varchar dan integer. Data buku memiliki relasi dengan data kategori dan data jenis penawaran.

Tabel 4.9 Rancangan tabel komentar

Data komentar		
Kolom	**Tipe Data**	**Contoh Data**

Id_komentar(pk)	Interger	Kode dari komentar (1,2,3…)
nama	Varchar	Berisi nama user
Id_post(fk)	Interger	Kode dari tabel post_buku (1,2,3…)
Isi_komentar	Varchar	Berisi data isi komentar user
Tanggal	date	9-03-2015 at 11.12am

Tabel 4.9 merupakan perancangan data komentar yang berfungsi untuk menyimpan dan menampilkan data-data komentar user. Data komentar membutuhkan dataId_komentar(pk), Id_user (fk), Id_post(fk), Isi_komentar, Tanggal yang memiliki tipe data varchar dan integer. Data komentar memiliki relasi dengan data buku dan data user.

Tabel 4.10 Rancangan tabel permintaan

Data komentar		
Kolom	**Tipe Data**	**Contoh Data**
Id (pk)	Interger	Kode dari komentar (1,2,3…)
Id_user(fk)	Varchar	Kode dari tabel user (1,2,3..)
Id_post(fk)	Interger	Kode dari tabel post_buku (1,2,3…)
Nego_harga	Integer	Berisi nego harga buku yang ditawar
Waktu	date	23/6/2015 2:30
Status	Varchar	“Disetujui”

Tabel 4.10 merupakan perancangan data permintaan request atau pemesanan buku yang berfungsi untuk menyimpan dan menampilkan data-data request permintaan buku oleh pengguna. Data permintaan membutuhkan data Id (pk), Id_user (fk), Id_post(fk), nego_harga, waktu yang memiliki tipe data varchar dan integer. Data permintaan memiliki relasi dengan data buku dan data user.

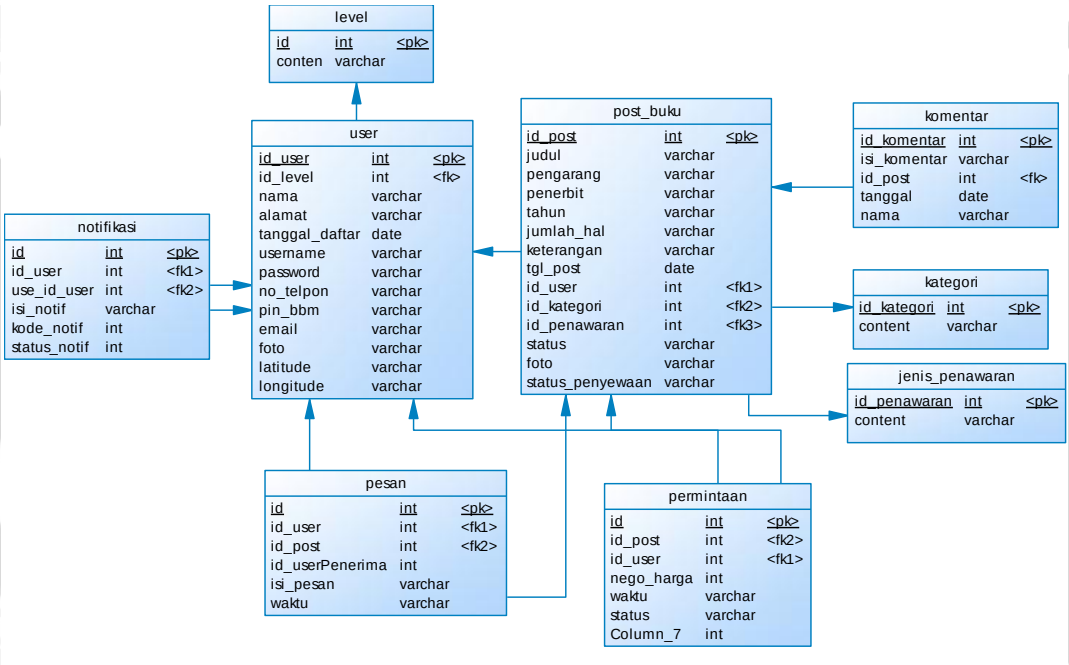
Tabel 4.11 Rancangan tabel pesan

Data komentar		
Kolom	**Tipe Data**	**Contoh Data**
Id (pk)	Interger	Kode dari komentar (1,2,3…)
Id_user(fk)	Interger	Kode dari tabel user (1,2,3..)
Id_userPenerima(fk)	Interger	Kode dari tabel user (1,2,3..)
Id_post(fk)	Interger	Kode dari tabel data penawaran post_buku (1,2,3…)
Isi_pesan	varchar	Berisi pesan “saya pesan buku …”
Waktu	date	23/6/2015 2:30

Tabel 4.11 merupakan perancangan data pesan yang berfungsi untuk menyimpan dan menampilkan data-data pesan untuk melakukan komunikasi pengaturan transaksi penjualan dan penyewaan buku. Data pesan membutuhkan data Id (pk), Id_user (fk), id_userPenerima (fk) Id_post(fk), isi_pesan, waktu yang

memiliki tipe data varchar dan integer. Data pesan memiliki relasi dengan data buku dan data user.

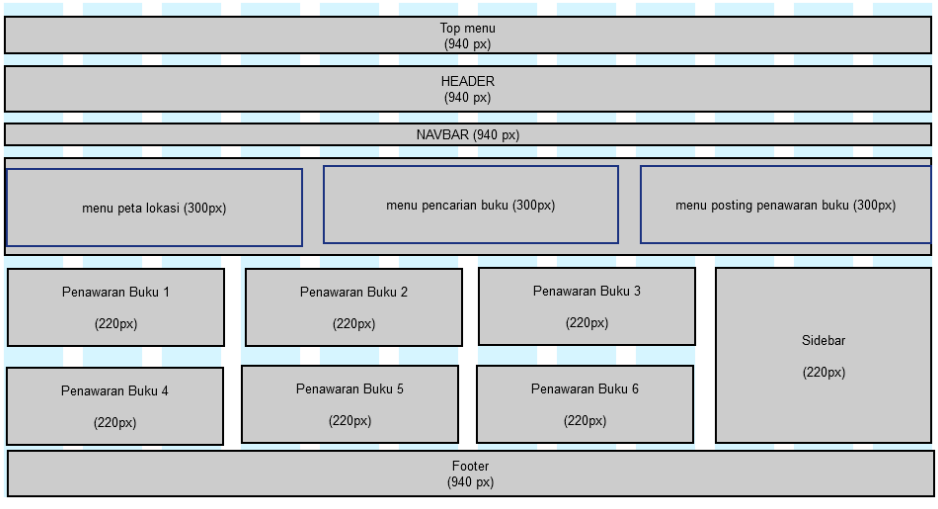
Berdasarkan struktur tabel yang telah didefinisikan pada Tabel 4.7, Tabel 4.8, Tabel 4.9, Tabel 4.10, Tabel 4.11 maka didapatkan pemodelan perancangan basis data direpresentasikan dalam bentuk CDM *(Conseptual Data Model*) yang ditunjukan pada Gambar 4.15. CDM merupakan penjabaran lebih lanjut dari diagram objek pada Gambar 4.14. CDM merupakan bentuk fisik perancangan basis data yang sudah siap diimplementasikan ke dalam DBMS sehingga nama tabel juga sudah merupakan nama asli tabel yang akan diimplementasikan ke dalam DBMS.



Gambar 4.15 *Conceptual Data Model* (CDM)

4.2.6 Perancangan Layout Responsif

Pada perancangan desain layout responsif Aplikasi Pencarian dan Penawaran Buku menggunakan teknik *grid system*. *Grid system* merupakan serangkaian garis bantu secara vertikal yang berguna sebagai panduan mendesain sebuah gambar rancangan agar menjadi proposional dan memiliki keteraturan tiap-tiap elemennya. Dalam mendesain layout aplikasi Pencarian dan Penawaran Buku ini menggunakan grid ukuran 960 pixel dengan 12 kolom, yang mana grid 960 pixel merupakan ukuran standar yang biasa digunakan dalam mendesain suatu web. Berikut merupakan rancangan layout web Aplikasi Pencarian dan Penawaran Buku menggunakan *grid* 960 pixel dengan 12 kolom.



Gambar 4.16 Perancangan layout *desktop* dengan ukuran *fixed*

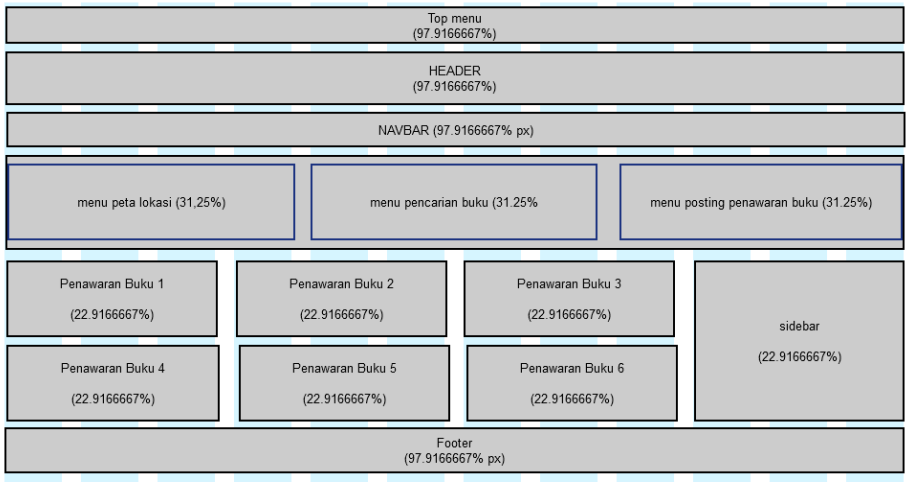
Gambar 4.16 merupakan hasil perancangan desain dengan teknik *grid system* menggunakan ukuran tetap (*fixed).* Pada Gambar 4.16 terdapat grid berukuran 960px dengan 12 kolom, lebar masing-masing kolom adalah 60px, setiap kolom memiliki margin 10 pixel, sehingga ketika dua buah kolom ketemu akan menghasilkan margin sebesar 20px untuk tepi kiri dan kanan. Desain grid di atas merupakan ukuran yang tetap (*fixed*) dalam satuan pixel, maka dalam membuat desain web yang responsive, ukuran *fixed* diatas harus diterjemahkan ke dalam *fluid grid. Fluid grid* merupakan teknik *grid* dengan bentuk dan ukuran yang lebih fleksibel dan dapat menyesuaikan ukuran layar, sehingga layout yang dibuat dengan grid tersebut juga mengikuti ukuran *grid*. Satuan ukuran dalam fluid grid tidak lagi menggunakan pixel akan tetapi menggunakan ukuran dalam persen (%). Untuk mengkonversi ukuran dari satuan pixel ke dalam bentuk yang fleksibel yaitu persen (%) dengan menggunakan rumus sebagai berikut:

(Target/Context) * 100% = Result.

Dimana, *target* merupakan panjang tiap-tiap rancangan layout sedangkan *context* adalah panjang ukuran *grid layout* atau ukuran layar yaitu 960px. Dengan ketentuan bahwa ukuran *target* harus sama dengan ukuran *context* (ukuran layar) dan tidak boleh melebihi ukuran *context.* Hal tersebut agar element content berada di tengah. Untuk menghitung ukuran target menggunakan rumus berikut:

Target = width + (2*border)+(2*padding)+(2*margin)

Dari rumus diatas sehingga didapatkan hasil rancangan desain menggunakan teknik *fluid grid* yang ditunjukan pada .



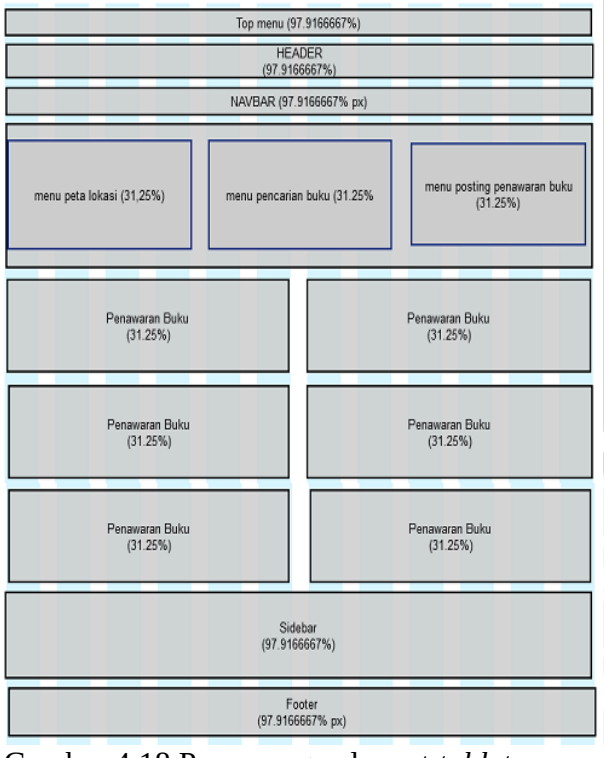
Gambar 4.17 Perancangan layout desktop dengan *fluid grid*

merupakan hasil dari perancangan yang sudah diterjemahkan dengan teknik *fluid grid* dengan menggunakan rumus **(Target/Context) * 100%** , contoh perhitungannya adalah sebagai berikut :

Untuk top menu : (940/960px) * 100% = 97.91666667%

Untuk right bar : (220/960px) * 100% = 22.916666667%.

Selanjutnya, merancang layout pada perangkat *tablet* yang memiliki ukuran layar yang lebih kecil dibandingkan perangkat desktop yang ditunjukan pada Gambar 4.18.

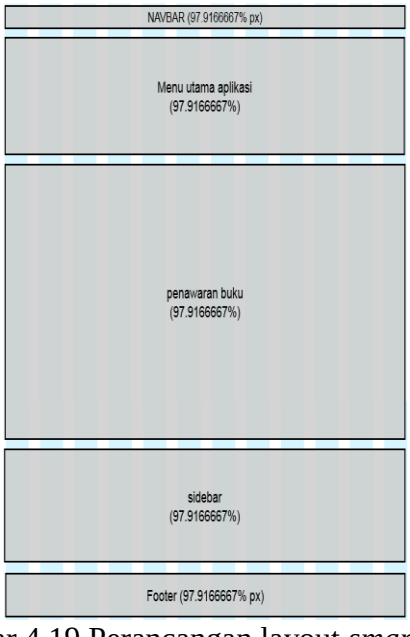


Gambar 4.18 Perancangan layout *tablet*

Gambar 4.18 merupakan perancangan layout untuk perangkat *tablet* dengan ukuran maksimum 1024px. Pada rancangan layoutnya terdapat beberapa penyesuaian yaitu kolom penawaran buku, *rightbar* pencarian, *rightbar* kategori, dan tombol next. Untuk ukuran kolom tetap menyesuaikan jumlah grid dengan 12 kolom. Agar desain *layout* dapat menyesuaikan layar *tablet* dengan ukuran maksimum 1024px maka dibutuhkan teknik *Media Queries. Media Queries* adalah salah satu fitur dalam CSS3 yang memungkinkan untuk menentukan kapan aturan CSS tertentu harus diterapkan, yaitu dengan kemampuannya yang bisa mengenali ukuran layar perangkat yang digunakan, Untuk menginialisasi penggunaan *media queries* pada perangkat *tablet* menggunakan kode CSS seperti dibawah ini:

```
@media screen and (max-width: 1024px) {
/* CSS Styles */
}
```

Selanjutnya, merancang layout pada perangkat *smartphone* yang ditunjukan pada .



Gambar 4.19 Perancangan layout *smartphone*

merupakan rancangan layout yang dibuat untuk perangkat. Perangkat *smartphone* memiliki ukuran layar jauh lebih kecil dibandingkan perangkat desktop maupun *tablet*. Sehingga, untuk menampilkan isi *website* secara optimal dan tanpa kehilangan isinya, cara terbaik adalah menyusun semua layout

memanjang kebawah dan merampingkan layout sehingga letak kolomnya akan disesuaikan seperti yang ditunjukan pada .

Agar desain layout dapat menyesuaikan layar *smartphone* dengan ukuran maksimum 768px maka dibutuhkan teknik *Media Queries. Media Queries* adalah salah satu fitur dalam CSS3 yang memungkinkan untuk menentukan kapan aturan CSS tertentu harus diterapkan, yaitu dengan kemampuannya yang bisa mengenali ukuran layar perangkat yang digunakan. Untuk menginialisasi penggunaan *media queries* pada perangkat *smartphone* dengan ukuran layar maksimum 768px menggunakan kode CSS seperti dibawah ini:

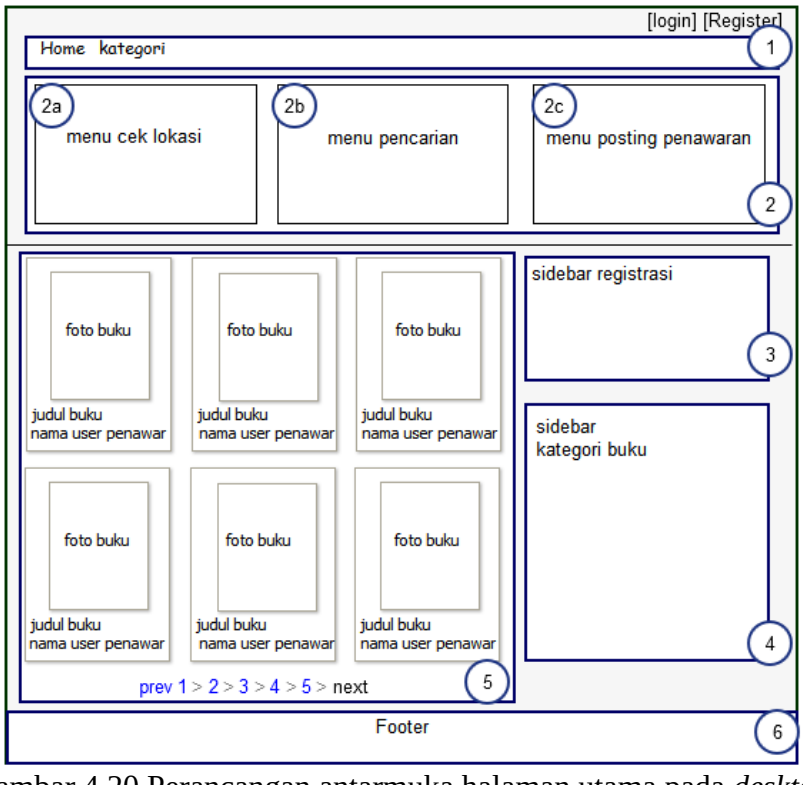
```
@media screen and (max-width: 768px) {
/* CSS Styles */
}
```

4.2.7 Perancangan Antarmuka

Perancangan antarmuka merupakan perancangan awal yang akan digunakan dalam implementasi antarmuka pengguna pada Aplikasi Pencarian dan Penawaran Buku. Antarmuka perangkat lunak ini akan digunakan oleh pengguna untuk untuk berinteraksi dengan sistem. Berikut merupakan perancangan antar muka aplikasi untuk pengguna.

4.2.7.1 Perancangan Antarmuka Halaman Utama

Halaman utama merupakan perancangan antarmuka tampilan awal ketika pengguna mengakses sistem. Halaman utama terdapat tampilan daftar penawaran buku dan menu-menu utama aplikasi yang terdiri dari menu cek lokasi, menu pencarian, menu posting penawaran. Menu cek lokasi berfungsi untuk menampilkan lokasi pengguna saat itu menggunakan teknologi *geolocation.* Menu pencarian berfungsi menampilkan hasil pencarian buku. Menu posting penawaran menampilkan form untuk menambah data penawaran buku baru ke dalam sistem. Untuk melakukan tambah penawaran buku, aktor harus sudah memiliki akun dan melakukan login terlebih dahulu.



Gambar 4.20 Perancangan antarmuka halaman utama pada *desktop*

merupakan perancangan antarmuka halaman utama aplikasi ketika diakses perangkat desktop dengan keterangan label sebagai berikut:

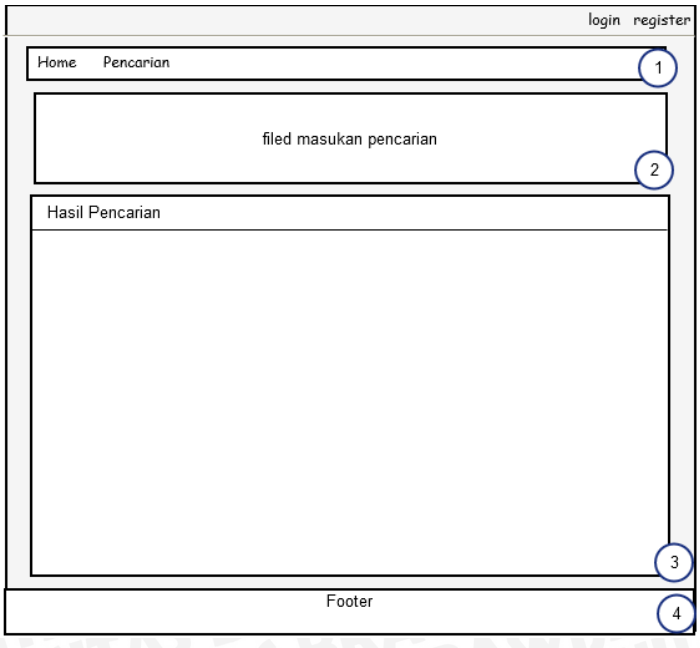
1. Label 1 merupakan elemen *navbar* menu ketika diakses pada perangkat dektop
2. Label 2 merupakan elemen menu utama aplikasi menu cek lokasi, menu pencarian dan menu posting penawaran
3. Label 3 merupakan elemen sidebar yang berisi informasi untuk pendaftaran akun untuk melakukan penawaran buku.
4. Label 4 merupakan elemen sidebar kategori yang menampilkan daftar kategori buku
5. Label 5 merupakan sidebar daftar penawaran buku yang ditawarkan
6. Label 6 merupakan footer yang menampilkan informasi pembuatan aplikasi

Gambar 4.21 Perancangan antarmuka halaman utama pada *smartphone* dan *tablet*

Gambar 4.21 merupakan perancangan antarmuka halaman utama aplikasi ketika diakses perangkat *smartphone dan tablet*. Pada saat halaman website diakses pada perangkat *tablet* terdapat beberapan penyesuaian elemen label 3 dan label 4 yang ditunjukan pada Gambar 4.20 tampilan akan menyesuaikan secara vertikal. Sedangkan saat diakses menggunakan perangkat smartphone terdapat penyesuaian konten web pada label 2 dan label 5 menjadi vertikal ke bawah.

4.2.7.2 Perancangan Antarmuka Halaman Pencarian

Halaman pencarian buku merupakan tampilan yang berisi *field* atau masukan untuk memfilter buku yang dicari. Pada halaman pencarian ditampilkan data hasil pencarian buku beserta jarak lokasi terdekat dari lokasi pengguna.

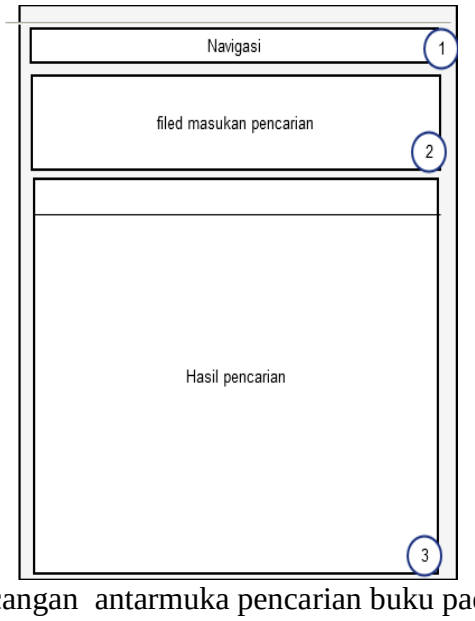


Gambar 4.22 Perancangan antarmuka pencarian buku pada *desktop*

Gambar 4.22 merupakan perancangan antarmuka halaman pencarian buku ketika diakses perangkat desktop dengan keterangan label sebagai berikut:

1. Label 1 merupakan elemen *navbar* menu
2. Label 2 merupakan field untuk melakukan filter pencarian buku
3. Label 3 merupakan hasil pencarian buku sesuai kriteria yang dimasukan
4. Label 5 merupakan footer yang menampilkan informasi pembuatan aplikasi

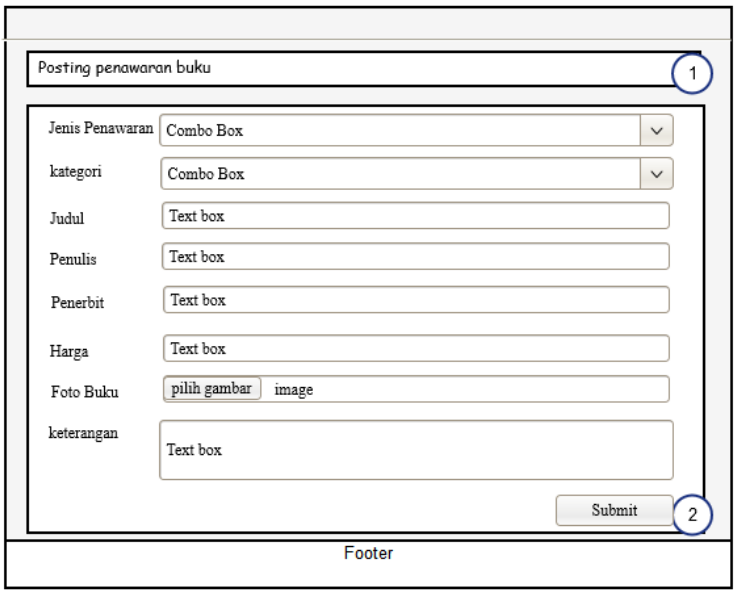
Gambar 4.23 merupakan perancangan perangkat lunak halaman pencarian buku ketika diakses perangkat *smartphone dan tablet*.



Gambar 4.23 Perancangan antarmuka pencarian buku pada *smartphone* atau *tablet*

4.2.7.3 Peracangan Antarmuka Posting Penawaran Buku

Halaman posting penawaran buku merupakan tampilan form untuk memasukan data penawaran buku baru. Pada form posting penawaran buku pertama kali user diminta untuk memilik jenis penawaran buku yang akan ditawarkan selanjutkan sistem akan menampilkan isian form sesuai dengan jenis penawaran buku yang dipilih.

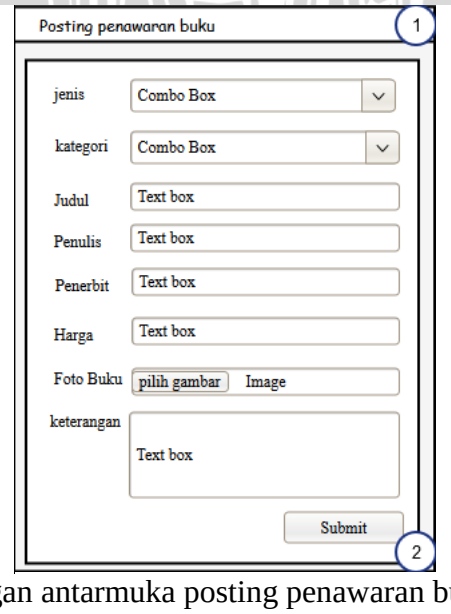


Gambar 4.24 Perancangan antarmuka posting penawaran buku pada *desktop*

Gambar 4.24 merupakan perancangan antarmuka halaman posting penawaran buku ketika diakses perangkat desktop dengan keterangan label sebagai berikut:

1. Label 1 merupakan *tab pane* pengisian form
2. Label 2 merupakan *field* data buku yang akan diposting yang terdiri dari jenis penawaran, kategori, judul, penulis, penerbit, keterangan penawaran.

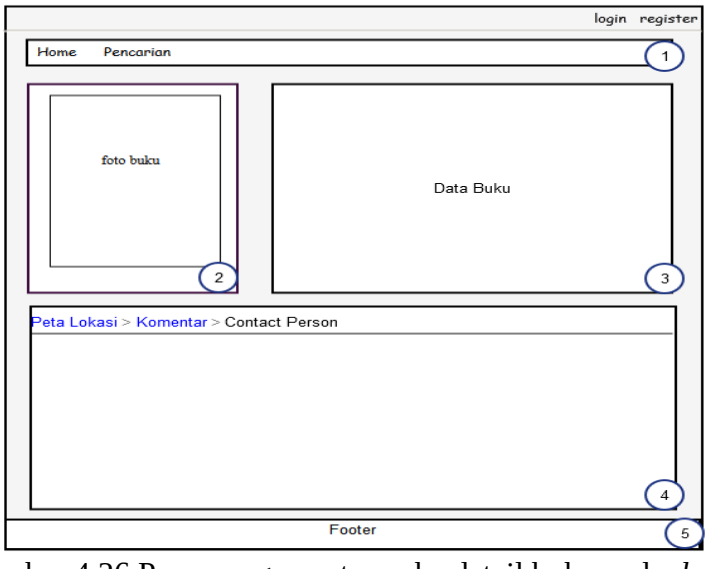
Gambar 4.25 merupakan perancangan perangkat lunak halaman posting penawaran buku ketika diakses perangkat *smartphone dan tablet*.



Gambar 4.25 Perancangan antarmuka posting penawaran buku pada *tablet* atau *smartphone*

4.2.7.5 Antarmuka Detail Buku

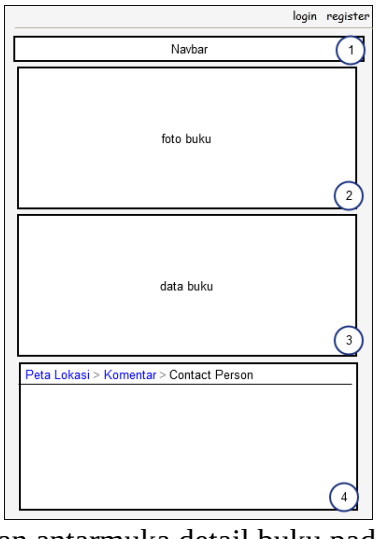
Halaman detail buku merupakan tampilan yang berisi informasi penawaran buku secara lengkap. Pada halaman detail buku ditampilkan informasi penawaran berupa foto buku, data buku, status penawaran buku, informasi pemilik, map dan komentar.



Gambar 4.26 Perancangan antarmuka detail buku pada *desktop*

Gambar 4.26 merupakan perancangan antarmuka halaman detail buku ketika diakses perangkat desktop dengan keterangan label sebagai berikut:

1. Label 1 merupakan elemen *navbar* menu
2. Label 2 merupakan elemen foto buku yang ditawarkan
3. Label 3 merupakan elemen informasi data buku.
4. Label 4 merupakan elemen *tap-pane* yang berisi peta lokasi, komentar dan contact person pemilik buku

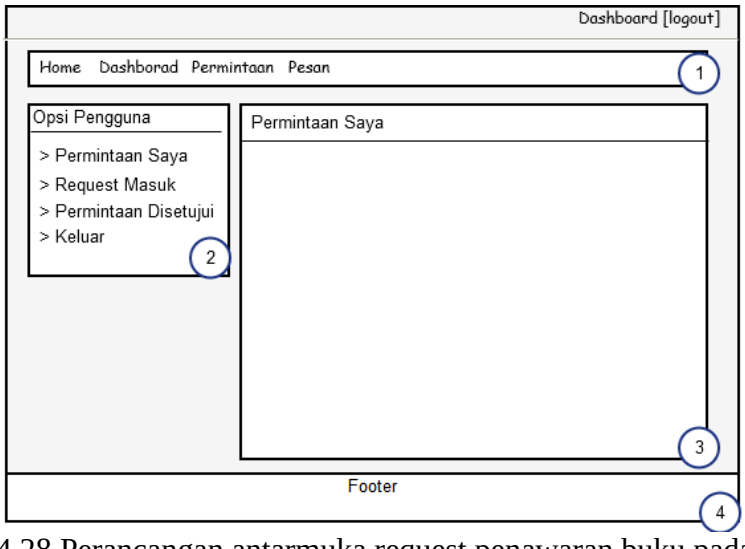


Gambar 4.27 Perancangan antarmuka detail buku pada smartphone atau tablet

Gambar 4.27 merupakan perancangan antarmuka halaman detail buku ketika diakses perangkat *smartphone dan tablet*. Pada saat halaman website diakses pada perangkat *smartphone* atau *tablet* elemen layout akan ditampilkan memanjang kebawah.

4.2.7.6 Antarmuka Halaman Request Penawaran Buku

Perancangan antarmuka request penawaran buku merupakan tampilan berisi history atau daftar penawaran buku yang direquest oleh user untuk menunggu konfirmasi dari user pemilik buku yang direquest.

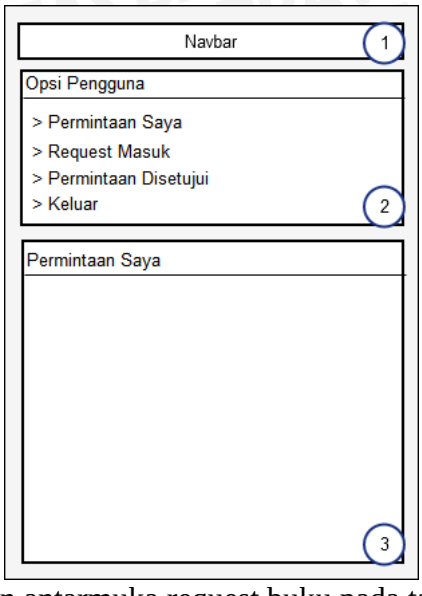


Gambar 4.28 Perancangan antarmuka request penawaran buku pada *desktop*

Gambar 4.28 merupakan perancangan antarmuka request terhadap buku yang ditawarkan ketika diakses perangkat desktop dengan keterangan label sebagai berikut:

1. Label 1 merupakan elemen *navbar* menu
2. Label 2 merupakan elemen opsi pengguna
3. Label 3 berisi daftar penwaran buku yang direquest
4. Label 4 merupakan footer aplikasi

Gambar 4.29 merupakan perancangan perangkat lunak halaman penawaran buku yang direquest ketika diakses perangkat *smartphone dan tablet*.



Gambar 4.29 Perancangan antarmuka request buku pada tablet atau smartphone

4.3 Perancangan Pengujian

Perancangan pengujian atau *test case* merupakan suatu tes yang dibutuhkan berdasarkan pada suatu inisialisasi, masukan, kondisi, ataupun hasil yang telah ditentukan sebelumnya. Perancangan pengujian dilakukan bertujuan untuk mengetahui fungsi-fungsi yang ada pada sistem telah dapat berjalan sesuai dengan apa yang diharapkan.

4.3.1 Lingkungan Pengujian Perangkat Lunak

Dalam melakukan pengujian Aplikasi Pencarian dan Penawaran Buku dibutuhkan perangkat lunak pengujian dan perangkat keras pengujian sebagai berikut:

4.3.1.1 Perangkat Lunak Pengujian

Perangkat lunak pengujian menggunakan browser yang terinstall pada perangkat keras yang akan digunakan sebagai pengujian. Beberapa browser yang digunakan pengujian meliputi Chorme, IE, Firefox, Safari dan Android Browser.

4.3.1.2 Perangkat Keras Pengujian

Perangkat keras yang dibutuhkan dalam melakukan pengujian aplikasi antara lain :

1. Notebook, dengan spesifikasi ukuran layar 14 inches (1366 x 768 pixel).
2. *Tablet Android* dengan spesifikasi ukuran layar 7 inches (1024 x 768 pixel)
3. *Smartphone* Android dengan spesifikasi ukuran layar 4 inches (480 x 768 pixel).

4.3.2 Identifikasi dan Rencana Pengujian

Tabel 4.12 Identifikasi dan rencana pengujian

No	Aspek Pengujian	Keterangan	Kriteria Pengujian	Jenis pengujian
1	Pengujian Validasi	Pengujian validasi digunakan untuk menguji kebutuhan fungsional apakah aplikasi yang dibangun sudah berjalan seperti yang diharapkan sesuai perancangan sistem	Pengujian validasi dikatakan sukses ketika sudah dilakukan pengujian terhadap kebutuhan fungsional yang didefinisikan di daftar kebutuhan pada Tabel 4.2 sudah sesuai dengan apa yang diharapkan	Black Box
2	Pengujian Layout Responsive	Pengujian *layout web responsive* dilakukan untuk mengetahui apakah antar muka aplikasi sudah dapat berjalan secara responsive saat diakses dalam lingkungan perangkat dengan ukuran layar yang berbeda.	Pengujian layout dikatakan berhasil ketika antarmuka aplikasi dapat menyesuaikan ukuran layar perangkat yang sudah dijelaskan pada 4.1.3.2 sesuai dengan peracangan antarmuka sistem yang sudah didefinikasikan pada 4.2.7	*Responsive Test*
3	Pengujian penentuan lokasi terdekat berdasarkan jarak	Pengujian jarak bertujuan untuk memastikan ketepatan jarak yang ditampilkan pada aplikasi dengan jarak sebenarnya untuk melakukan pencarian buku sehingga pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya..	Pengujian dikatakan berhasil apabila aplikasi dapat merepresentasikan lokasi terdekat dengan posisi pengguna berdasarkan jarak lokasi	Pengujian Jarak
4	Pengujian kompatibilitas sistem	Pengujian *compatibility* dilakukan untuk mengetahui kompatibilitas sistem untuk dapat menjalankan fungsi yang ada pada kebutuhan fungsional sistem dengan baik pada jenis browser yang berbeda pada perangkat	Pengujian dikatakan berhasil jika kebutuhan fungsional sistem dapat berjalan dengan baik pada beberapa jenis browser yang dijelaskan pada 4.3.1.1	Pengujian Compatibi-lity

Tabel 4.12 merupakan identifikasi dan rencana pengujian yang akan dilakukan pada tahap pengujian. Rencana pengujian yang akan dilakukan untuk pengujian sistem antara lain pengujian validasi untuk mengetahui apakah aplikasi

yang dibangun sudah berjalan seperti yang diharapkan, pengujian *responsive test* untuk menguji layout web responsive terhadap ukuran layar dari perangkat yang digunakan, pengujian perhitungan jarak untuk menguji akuransi jarak antar lokasi dan pengujian non-fungsional pada aplikasi Pencarian dan Penawaran Buku.

4.3.3 Perancangan Kasus Uji

4.3.3.1 Kasus Uji Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah aplikasi yang dibangun sudah berjalan seperti yang diharapkan sesuai dengan skeneario kebutuhan fungsional sistem yang sudah dirancang sebelumnya. Pengujian validasi ini dilakukan berdasarkan daftar kebutuhan fungsional pada Tabel 4.2.. Pengujian ini menggunakan metode *black-box testing*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan kecocokan antara kinerja sistem dengan daftar kebutuhan (*use case) yang* ditunjukan pada Tabel 4.2. Berikut merupakan *Test Case* atau kasus uji dari pengujian validasi:

1. Kasus Uji Pencarian Buku

Tabel 4.13 Kasus uji pencarian buku

Nomor Kasus Uji	UVL_01
Nama Kasus Uji	Kasus Uji Pencarian Buku
Objek Uji	Kebutuhan Fungsional SRS_03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat melakukan pencarian buku yang menampilkan jarak lokasi sehingga aktor dapat menemukan penawaran buku yang dekat dengan lokasinya
Data Masukan	1. Koordinat latitude dan longitude pengguna yang diperoleh melalui browser dengan menggunakan teknologi Geolocation. 2. Koordinat lokasi alamat pemilik buku
Prosedur Uji	1. Aktor masuk mengakses sistem 2. Sistem menampilkan halaman menu utama aplikasi

	3. Aktor Memilih menu “cari buku” 4. Sistem menampilkan halaman pencarian buku 5. Aktor menginputkan data pencarian buku yang berjudul “koala kumal” (studi kasus buku yang dicari) 6. Sistem menampilkan hasil pencarian buku yang berjudul “koala kumal” sesuai masukan pengguna berserta informasi lokasi dan jarak menuju lokasi.
Hasil yang Diharapkan	Aplikasi dapat manampilkan daftar pencarian buku yang berjudul “koala kumal” beserta informasi lokasi dan jarak menuju lokasi berdasarkan hasil perhitungan koordinat perangkat user dan koordinat alamat user penawar menggunakan rumus Haversine Formula.

2. Kasus Uji Memposting Penawaran Buku

Tabel 4.14 Kasus uji memposting penawaran buku

Nomor Kasus Uji	UVL_02
Nama Kasus Uji	Kasus Uji Memposting Penawaran Buku
Objek Uji	Kebutuhan Fungsional SRS_06
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat melakukan posting penawaran buku baru ke dalam sistem
Data Masukan	Data penawaran buku yang berjudul “koala kumal” (data yang dimasukan: judul, penerbit, penulis, jenis penawaran, kategori, harga, dll)
Prosedur Uji	1. Aktor melakukan login dan masuk ke halaman dashboard sistem 2. Aktor memilih tombol “tambah penawaran” 3. Sistem menampilkan form tambah penawaran buku 4. Aktor memasukan data penawaran buku sesuai isian *field* yang diminta (jenis penawaran, foto, judul, penulis, penebit, jumlah halaman, kategori buku, harga) 5. Aktor menekan tombol post penawan buku 6. Sistem memeriksa kelengkapan isian data

	7. Jika semua data sudah diisi maka sistem akan menyimpan data penawaran buku kedalam database
Hasil yang Diharapkan	Data penawaran buku baru dengan judul "koala kumal" yang dimasukan oleh aktor berhasil tersimpan ke dalam database dan tampil pada halaman daftar penawaran buku sesuai dengan data yang dimasukan

3. Kasus Uji Request Buku yang ditawarkan

Tabel 4.15 Kasus uji request buku yang ditawarkan

Nomor Kasus Uji	UVL_03
Nama Kasus Uji	Kasus Uji Request Buku yang Ditawarkan
Objek Uji	Kebutuhan Fungsional SRS_08
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat mengirim data request buku kepada user pemilik yang menawarkan buku tersebut dan menyimpan data request pada history "permintaan saya"
Data Masukan	Data permintaan buku yang berjudul "koala kumal" dengan user pemilik bernama "afni"
Prosedur Uji	1. Aktor melakukan login dan masuk ke halaman dashboard sistem 2. Aktor masuk pada halaman utaman sistem 3. Sistem menampilkan halaman utama aplikasi 4. Aktor memilih salah satu buku yang ingin direquest (buku yang dipilih yaitu judul buku: koala kumal dengan pemilik bernama afni) 5. Sistem menampilkan detail informasi buku yang dipilih 6. Aktor menekan tombol "request" 7. Sistem menampilkan form request 8. User memasukan nego harga penawaran jika status harga dapat dinego dan menekan tombol "submit" 9. Sistem menyimpan data permintaan buku ke dalam history

	"permintaan saya" dan mengirimkan permintaan request buku kepada user pemilik buku tersebut.
Hasil yang Diharapkan	Data penawaran buku yang direquest atau dipesan dengan judul "koala kumal" tersimpan pada history daftar buku yang direquest dan permintaan request berhasil dikirimkan kepada user pemilik buku yang direquest bernama "afni"

4.3.3.2 Kasus Uji Pengujian Layout Web Responsif

Pengujian *layout web responsive* dilakukan untuk mengetahui apakah antar muka aplikasi sudah dapat berjalan secara responsive saat diakses dalam lingkungan perangkat dengan ukuran layar yang berbeda. Pengujian dilakukan dengan melakukan akses terhadap aplikasi melalui perangkat dengan ukuran layar yang berbeda. Perangkat yang digunakan meliputi *smartphone*, *tablet* dan *desktop.*

1. Kasus Uji Pengujian Layout Pada Ukuran Layar *Smartphone*

Pengujian *layout web responsive* pada layar *smartphone* dilakukan dengan mengakses aplikasi Pencarian dan Penawaran Buku dengan menggunakan perangkat *smartphone*. Spesifikasi perngkat lunak yang digunakan adalah *smartphone platform* Android dengan ukuran layar 4.0 *inches* yang memiliki (css width ≤768 pixel) menggunakan browser chrome android. Tabel 4.16 merupakan desain pengujian layout web responsive pada perangkat *smartphone*.

Tabel 4.16 Kasus uji pengujian aplikasi pada layar *smartphone*

Aspek Pengujian	Pengujian layout pada ukuran layar *smartphone*
Tujuan Pengujian	Pengujian dilakukan untuk memastikan kebutuhan fungsional aplikasi sudah dapat berjalan secara responsive saat diakses pada ukuran layar perangkat *smartphone* sesuai perancangan antarmuka aplikasi
Masukan	Antamuka aplikasi dijalankan pada perangkat *smartphone*
Prosedur Uji	Aplikasi dijalankan pada perangkat *smartphone* dengan ukuran layar 4.0 inches (480 x 768 pixel) browser chrome android

Hasil yang Diharapkan	Konten website pada tampilan antarmuka yang diuji dapat menyesuaikan ukuran layar pada perangkat *smartphone* dengan sesuai dengan perancangan antarmuka aplikasi pada 4.2.7

2. Kasus Uji Pengujian Layout Pada Ukuran Layar *Tablet*

Pengujian layout web responsive pada layar *tablet* dilakukan dengan mengakses aplikasi Pencarian dan Penawaran Buku dengan menggunakan perangkat *tablet*. Spesifikasi perngkat lunak yang digunakan adalah *Tablet* Android dengan ukuran layar 7.0 inches menggunakan browser Android. Tabel 4.17 merupakan desain pengujian layout web responsive pada perangkat *tablet*.

Tabel 4.17 Kasus Uji pengujian aplikasi pada layar *tablet*

Aspek Pengujian	Pengujian layout pada ukuran layar *tablet*
Tujuan Pengujian	Pengujian dilakukan untuk memastikan kebutuhan fungsional aplikasi sudah dapat berjalan secara responsive saat diakses pada ukuran layar perangkat *tablet* sesuai perancangan antarmuka aplikasi
Masukan	Antarmuka Aplikasi dijalankan pada perangkat *tablet*
Prosedur Uji	Aplikasi dijalankan pada perangkat *tablet* dengan ukuran layar 7.0 inches (1024 x 768 pixel) menggunakan browser Android.
Hasil yang Diharapkan	Konten website pada tampilan antarmuka yang diuji dapat menyesuaikan ukuran layar pada perangkat *tablet* dengan baik sesuai dengan perancangan antarmuka aplikasi sesuai dengan perancangan antarmuka aplikasi pada 4.2.7

3. Kasus Uji Pengujian Layout pada Ukuran Layar Desktop

Pengujian layout web responsive pada layar desktop dilakukan dengan mengakses aplikasi Pencarian dan Penawaran Buku dengan menggunakan perangkat desktop. Spesifikasi perngkat lunak yang digunakan adalah Notebook atau laptop dengan ukuran layar 14 inches menggunakan browser Chrome. Tabel 4.18 merupakan test case pengujian layout web responsive pada perangkat desktop.

Tabel 4.18 Kasus uji pengujian aplikasi pada layar laptop

Aspek	Pengujian layout pada ukuran layar laptop

Pengujian	
Tujuan Pengujian	Pengujian dilakukan untuk memastikan kebutuhan fungsional aplikasi sudah dapat berjalan secara responsive saat diakses pada ukuran layar perangkat *tablet* sesuai perancangan antarmuka aplikasi
Masukan	Antarmuka aplikasi dijalankan pda perangkat laptop
Prosedur Pengujian	Aplikasi dijalankan pada perangkat laptop dengan ukuran layar 14 inches (1366 x 768 pixel) menggunakan browser Chrome
Hasil yang Diharapkan	Konten website pada tampilan antarmuka yang diuji dapat menyesuaikan ukuran layar pada perangkat laptop sesuai dengan perancangan antarmuka aplikasi pada 4.2.7

4.3.3.3 Kasus Uji Pengujian Jarak Lokasi

Pengujian jarak dilakukan untuk menguji ketepatan hasil penentuan lokasi terdekat berdasakan jarak antar lokasi yang diperoleh menggunakan persamaan Haversine Formula. Pengujian ketepatan jarak yang ditampilkan pada aplikasi bertujuan untuk melakukan pencarian buku yang menampilkan jarak lokasi sehingga pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya sehingga mempermudah transaksi peminjaman dan pengembalian buku. Sesuai rancangan aplikasi, persamaan Haversine Formula mampu menentukan lokasi terdekat dengan posisi user dan memiliki akuransi kesalahan sebesar 0.5%,

Pengujian akuransi jarak dilakukan dengan cara membandingan perhitungan jarak pada aplikasi dengan perhitungan jarak sebenarnya menggunakan *speedometer*. Tabel 4.19 merupakan test case dari pengujian perhitungan jarak menggunakan rumus Haversine Formula

Tabel 4.19 Kasus uji pengujian akuransi jarak

Aspek Pengujian	Pengujian Haversine Formula
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat menentukan lokasi terdekat berdasarkan jarak lokasi dan memperoleh akuransi dengan membandingkan selisih jarak

	sebenarnya dengan perhitungan jarak menggunakan metode Haversine Formula
Masukan	1. Lokasi awal dan koordinat latitude dan longitude pengguna yang diperoleh melalui browser dengan menggunakan teknologi Geolocation. 2. Lokasi tujuan dan koordinat alamat lokasi pemilik buku yang dipilih
Hasil yang Diharapkan	Penerapan Haversine Formula dapat menentukan lokasi terdekat berdasarkan jarak dengan membandingan jarak asli dengan hasil perhitungan jarak rumus (berdasarkan koordinat perangkat pengguna dan koordinat alamat lokasi user penawar) dan didapatkan prosentase selisih akurasi kesalahan terhadap jarak sebenarnya

4.3.3.4 Kasus Uji Pengujian Compatibility

Pengujian compatibility dilakukan untuk mengetahui kompatibilitas antarmuka sistem dapat menjalankan fungsi yang ada pada kebutuhan fungsional sistem dengan baik pada jenis browser yang berbeda pada perangkat desktop maupun perangkat bergerak. Browser yang digunakan untuk melakukan pengujian kompatibilitas sitem menggunakan browser Chorme, Internet Explorer, Firefox, Safari dan Android Browser. Pengujian dilakukan pada kelima browser tersebut dengan alasan browser tersebut sudah mendukung teknologi *responsive web design* dan teknologi Geolocation API serta berdasarkan *survey StatCounter* pada tahun 2013, kelima browser tersebut merupakan browser paling banyak digunakan di dunia (Buckler, 2013). Tabel 4.20 merupakan test case dari pengujian kompatibilitas sistem.

Tabel 4.20 Kasus uji pengujian compatibilty

Aspek Pengujian	Pengujian kompatibilitas
Tujuan Pengujian	Pengujian dilakukan untuk mengetahui kompatibilitas antarmuka sistem dapat menjalankan fungsi yang ada pada kebutuhan fungsional sistem dengan baik pada jenis browser yang berbeda pada perangkat desktop maupun perangkat bergerak

Masukan	Kebutuhan fungsional aplikasi dijalankan jenis browser yang berbeda (jenis browser yang digunakan yaitu : Chorme, Internet Explorer, Firefox, Safari dan Android Browser)
Prosedur Pengujian	Menjalakan setiap halaman sesuai dengan spesifikasi kebutuhan sistem pada jenis browser yang berbeda (Chorme, Internet Explorer, Firefox, Safari dan Android Browser)
Hasil yang Diharapkan	Aplikasi dapat menjalankan kebutuhan fungsional sistem dengan baik pada jenis browser yang berbeda (Chorme, Internet Explorer, Firefox, Safari dan Android Browser)

BAB V
IMPLEMENTASI

Pada bab ini dibahas mengenai spesifikasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan perancangan perangkat lunak. Pada pembahasan bab implementasi terdiri dari spesifikasi sistem, batasan-batasan dalam implementasi, implementasi class dan assets, implementasi algoritma dan kode program, implementasi *layout responsive* dan implementasi antarmuka aplikasi.

9.1 Spesifikasi Sistem

Aplikasi Pencarian dan Penawaran Buku dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak sebagai berikut:

1. Spesifikasi Perangkat Keras

   Spesifikasi perangkat keras yang dipakai dalam proses pembuatan aplikasi Pencarian dan Penawaran Buku menggunakan sebuah komputer notebook dengan sistem operasi windows 7 dan prosesor Intel Core i3.

2. Spesifikasi Perangkat Lunak

   Spesifikasi perangkat lunak yang dipakai dalam proses pembuatan aplikasi Pencarian dan Penawaran Buku dijelaskan pada tabel 5.1

Tabel 5.1 Spesifikasi perangkat lunak

Operating System	Microsoft Windows 7
Programming Language	PHP, JavaScript
Programming Enviroment	Apache 2.4.7
Application Programming Interface	Google Maps API v3, Geolocation API, jQuery
Database Management System	MySQL 5.6.16

5.2 Batasan-batasan Implementasi

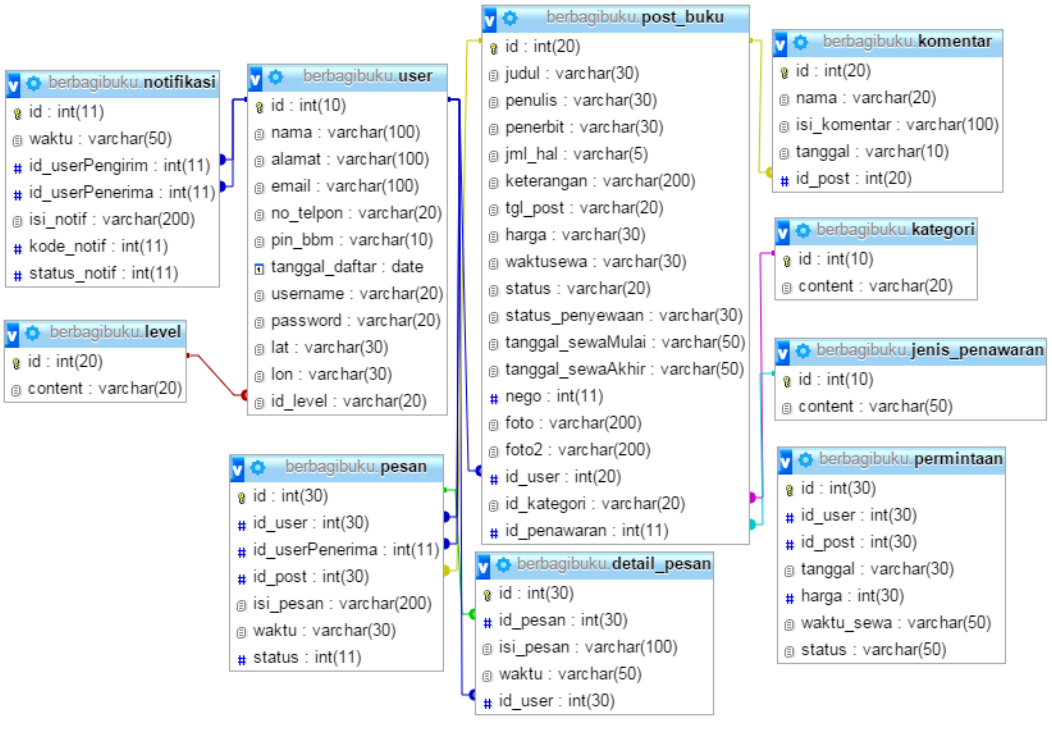
Pada Implementasi perangkat lunak aplikasi Pencarian dan Penawaran Buku terdapat batasan-batasan dalam proses yaitu sebagai berikut:

1. Aplikasi Pencarian dan Penawaran Buku dibangun berbasis *web* menggunakan bahasa pemograman PHP dan *Javascript.*

2. Implementasi layout menggunakan tenik *Responsive Web Design* berupa *grid system*, *fluid system* dan *media queries.*
3. Implementasi antarmuka layout web responsive menggunakan HTML5 dan CSS3 dengan rentang ukuran layar perangkat *smartphone* (≤768 px), *tablet (*768-1024 px), dan *desktop* atau *laptop* (>1024 px)
4. Implementasi penyajian data spasial pemetaan menggunakan Google Maps API.
5. Perhitungan lokasi terdekat berdasarkan jarak menggunakan rumus Haversine Formula.

5.3 Implementasi Basis Data

Implementasi basis data dibuat berdasarkan perancangan database yang digambarkan dalam diagram *Conseptual Data Model* (CDM) Gambar 4.15 yang telah dijabarkan pada naskah dokumen 4.2.5. Implementasi pada penyimpanan data dilakukan dengan database management system MySQL. Hasil implementasi penyimpanan database dimodelkan dalam diagram *phisycal entity relationship* dari aplikasi Pencarian dan Penawaran Buku. Gambar 5.1 merupakan implementasi basis data aplikasi Pencarian dan Penawaran Buku pada database MySQL yang menyimpan data user dan data penawaran buku.



Gambar 5.1 Diagram *phisycal entity relationship*

Gambar 5.1 merupakan implementasi database sistem, terdapat tabel user merupakan tabel yang berfungsi untuk menyimpan data pengguna yang menawarkan bukunya. Atribut tabel user dibuat sesuai dengan perancangan data pada Tabel 4.7. Tabel user memiliki relasi dengan tabel level yang berfungsi untuk membedakan akses sistem antara admin dengan pengguna. Tabel post_buku merupakan tabel untuk menyimpan data penawaran buku dan memiliki atribut sesuai dengan perancangan data pada tabel 4.8. Tabel post_buku memiliki relasi dengan tabel jenis_penawaran dan tabel kategori. Tabel komentar merupakan implementasi basisdata dari perancangan Tabel 4.9 yang berfungsi untuk menyimpan data komentar penawaran buku yang memiliki relasi dengan tabel post_buku. Tabel permintaan merupakan implementasi basisdata dari perancangan Tabel 4.10 yang berfungsi untuk menyimpan request buku yang ditawarkan dan memiliki relasi dengan tabel post_buku dan tabel user. Tabel pesan merupakan implementasi basisdata dari perancangan Tabel 4.11 yang berfungsi untuk mengirimkan data pesan dan memiliki relasi dengan tabel post_buku dan tabel user.

5.4 Implementasi Class

Implementasi class diimplementasikan berdasarkan perancangan diagram class pada Gambar 4.12. Implementasi *Class* direalisasikan kedalam bentuk file program, diantaranya adalah implementasi *class default* dengan ekstensi (.php) dan implementasi asset menggunakan CSS dengan ekstensi (.css).

Tabel 5.2 merupakan hasil implementasi dari perancangan *class diagram* (Lihat Gambar 4.12) yang direalisasikan pada file program. Terdapat *class controller* yang berfungsi menjalan proses sesuai perintah masukan pengguna. Class controller memiliki relasi dengan class View yang berfungsi sebagai tampilan antarmuka sistem dan berelasi dengan Class Model yang berfungsi sebagai pemanggilan data dari database sistem.

Tabel 5.2 Implementasi class pada file program

No	Package	Nama Class	Nama File Program
1	Controller	loginControl	loginControl.php
2	Controller	registerControl	registerControl.php
3	Controller	bukuControl	bukuControl.php

4	Controller	dashboardControl	halAwalControl.php
5	Controller	profilControl	profilControl.php
6	Controller	permintaanControl	permintaanControl.php
7	Controller	pesanControl	pesanControl.php
8	Controller	adminControl	adminControl.php
9	View	user	– menuUser.php – footerUser.php – beranda.php – dashboard.php – pencarian.php – pencarianPosisiSekarang.php – pencarianPosisiTertentu.php – detailBuku.php – permintaanSayap_view.php – permintaanMasuk.php – pesanInbox.php – pesanOutbox.php – notifikasi.php – ruteLokasi.php – profil.php
10	View	admin	– menuAdmin.php – footerAdmin.php – dashboard.php – dataUser.php – dataBuku.php – dataKategori.php
10	Model	bukuModel	– bukuModel.php
11	Model	userModel	– userModel.php
13	Model	profilModel	– profilModel.php
14	Model	loginModel	– loginModel.php
15	Model	permintaanModel	– permintaanModel.php
16	Model	pesanModel	– pesanModel.php

17	Model	adminModel	– adminModel.php

5.5 Implementasi Kode Program

Pada implementasi kode program akan dijelaskan hasil implementasi aplikasi sesuai dari perancangan yang telah dijelaskan pada bab sebelumnya. Aplikasi Pencarian dan Penawaran Buku mempunyai beberapa proses atau fungsi utama yang terbagi dalam beberapa file program. Penulisan tugas akhir ini hanya menyantumkan implementasi dari beberapa proses saja sesuai perancangan daftar kebutuhan fungsional pada tabel 4.2. Implementasi proses yang dicantumkan antara lain adalah implementasi kode pencarian buku, memposting penawaran buku, request atau pemesanan buku yang ditawarkan.

Kode 5.1 merupakan script PHP pada class "bukuControl" yang digunakan untuk melakukan proses pencarian penawaran buku disekitar area lokasi user berdasarkan jarak terdekat. Implementasi kode program pencarian buku dibuat berdasarkan perancangan *class diagram* pada Gambar 4.9.

```
1  class bukuControl extends CI_Controller{
2  .
3  .
4  public function pencarian(){
5    $this->data['listKategori']= $this->bukuModel-
6  >getKategori();
7    $this->load->view('templates/menuUser',$this->data);
8    $this->load->view('user/pencarianView,$this->data);
9    $this->load->view('templates/footerUser',$this->data);
10 }
11 public function terdekatPosisiSekarang(){
12   $data['listKategori']= $this->bukumodel->kategori();
13   $data['jenis']= $this->bukumodel->jenis();
14   $this->load->view('templates/menuUser',$this->data);
15   $this->load->view('user/pencarianPosisiSekarang',$data);
16   $this->load->view('templates/footerUser',$this->data);
17 }
18
19 public function terdekatPosisiTertentu(){
20   $data['listKategori']= $this->bukumodel->kategori();
21   $data['jenis']= $this->bukumodel->jenis();
22   $this->load->view('templates/menuUser',$this->data);
23   $this->load->view('user/pencarianPosisiTertentu',$data);
24   $this->load->view('templates/footerUser',$this->data);
25 }
26
27 public function prosesCariTerdekat(){
28   $lat = $this->input->post('getlat');
     $lon = $this->input->post('getlon');
29   $kategori = $this->input->post('kategori');
30   $jenis = $this->input->post('jenis');
31   $judul= $this->input->post('judul');
32   $this->data["bukuTerdekat"]= $this->bukumodel->filter($lat,
```

```
33 $lon, $judul, $jenis, $kategori);
34   $this->load->view('user/pencarianBukuTerdekat',$this-
35 >data);
36 }
37 .
38 .
39 }
```

Kode 5.1 Script PHP pada class bukuControl untuk proses pencarian buku

Kode 5.1 merupakan script PHP pada *class* bukuControl berdasarkan perancangan *class* yang dibuat pada Gambar 4.6. Pada *class bukucontrol* terdapat fungsi *pencarian()* akan menampilkan halaman pencarian. Fungsi *terdekatPosisiSekarang()* merupakan fungsi untuk melakukan proses pencarian buku berdarsakan posisi terdekat saat itu. Fungsi *terdekatPosisiTertentu()* merupakan fungsi untuk melakukan proses pencarian buku terdekat berdasarkan lokasi yang ditentukan oleh user. Pencarian buku terdekat dilakukan dengan menghitung jarak berdasarkan koordinat *latitude* dan *longitude* lokasi awal dan koordinat *latitude* dan *longitude* alamat lokasi pengguna yang menawarkan bukunya menggunakan persamaan Haversine Formula.

```
1  class bukuModel extends CI_Model{
2  .
3  .
4  public function filter($lat, $lon, $judul, $jenis, $kategori){
5  if($jenis=="-" && $kategori=="-"){
6  $query=$this->db->query("SELECT post_buku.id AS idPost,
7  post_buku.foto AS foto1, post_buku.tgl_post, post_buku.harga,
8  post_buku.waktusewa, jenis_penawaran.content AS penawaran,
9  post_buku.judul AS judul, post_buku.penulis AS pengarang,
10 post_buku.penerbit, user.id AS id_user,post_buku.keterangan,
11 user.username, user.nama, user.alamat, user.pin_bbm,
12 user.no_telpon, user.email, kategori.id, kategori.content AS
13 kategori, ROUND( ( 6371 * ACOS( COS( RADIANS( $lat ) ) * COS(
14 RADIANS( user.lat ) ) * COS( RADIANS( $lon ) - RADIANS(
15 user.lon ) ) + SIN( RADIANS( $lat ) ) * SIN( RADIANS( user.lat
16 ) ) ) ) , 2 ) AS jarak
17 FROM post_buku
18 INNER JOIN user ON user.id = post_buku.id_user
19 INNER JOIN kategori ON kategori.id = post_buku.id_kategori
20 INNER JOIN jenis_penawaran ON jenis_penawaran.id =
21 post_buku.id_penawaran
22 WHERE post buku.status = 'Aktif'
23 AND post_buku.judul LIKE '%".$judul."%'
24 order by jarak asc");
25
26 }else if ($judul==null && $jenis!="-" && $kategori!="-"){
27 $query=$this->db->query("SELECT post_buku.id AS idPost,
28 post_buku.foto AS foto1, post_buku.tgl_post, post_buku.harga,
29 post_buku.waktusewa, jenis_penawaran.content AS penawaran,
30 post_buku.judul AS judul, post_buku.penulis AS pengarang,
31 post_buku.penerbit, user.id AS id_user,post_buku.keterangan,
32 user.username, user.nama, user.alamat, user.pin_bbm,
```

```
user.no_telpon, user.email, kategori.id, kategori.content AS
kategori, ROUND( ( 6371 * ACOS( COS( RADIANS( $lat ) ) * COS(
RADIANS( user.lat ) ) * COS( RADIANS( $lon ) - RADIANS(
user.lon ) ) + SIN( RADIANS( $lat ) ) * SIN( RADIANS( user.lat
) ) ) ) , 2 ) AS jarak
FROM post_buku
INNER JOIN user ON user.id = post_buku.id_user
INNER JOIN kategori ON kategori.id = post_buku.id_kategori
INNER JOIN jenis_penawaran ON jenis_penawaran.id =
post_buku.id_penawaran
WHERE post_buku.status = 'Aktif'
AND jenis_penawaran.id = $jenis
AND kategori.id=$kategori
order by jarak asc");
.
.
return $q;
}
}
```

Kode 5.2 Script PHP pada class bukuModel untuk hasil pencarian buku

Kode 5.2 merupakan *script* PHP pada *class* bukuModel berdasarkan class bukuModel yang terdapat pada perancangan diagram class Gambar 4.9. Pada diagram class bukuModel Gambar 4.9 terdapat fungsi *filter*(). Fungsi *filter()* yang diimplementasikan pada kode 5.2 sesuai dengan perancangan *class* Gambar 4.9. Fungsi *filter()* berfungsi untuk melakukan query database dan melakukan pencarian buku berserta perhitungan jarak berdasarkan koordinat latitude dan longitude lokasi pengguna dengan koordinat *latitude* dan *longitude* alamat lokasi pengguna yang menawarkan buku tersebut menggunakan Haversine Formula.

```
if (navigator.geolocation) {

      navigator.geolocation.getCurrentPosition(showPosition);
                  }

function showPosition(position) {
document.getElementById("getlat").value
=position.coords.latitude;
document.getElementById("getlon").value
position.coords.longitude;
//document.getElementById("idKategori").value=<?php ?>;
}

function fireAjax(id){
$.ajax({url:"<?php echo
base_url()?>halAwal_control/bukuTerdekat/"+id,
      type:"POST",
      async:false,

data:{getlat:$("#getlat").val(),getlon:$("#getlon").val()},
            success: function (response){
            $('#response').html(response);
            }
     });
```

```
25  }
```

Kode 5.3 Script Javascript untuk melakukan pencarian buku

Kode 5.3 merupakan *script javascript* yang digunakan untuk mencari lokasi penawaran buku terdekat berdasarkan lokasi pengguna saat itu dengan menggunakan teknonologi *Geolocation.* Untuk melakukan pencarian lokasi terdekat dimulai dengan mencari koordinat pengguna saat itu menggunakan *Geolocation API.* Apabila *Geolocation API* berhasil dijalankan maka akan didapat koordinat latitude dan longitude pengguna selanjutnya akan dikirim ke fungsi *prosesCariTerdekat()* yang berada pada *class* bukuControl Kode 5.1.

```
1   class dashboardControl extends CI_Controller{
2   .
3   .
4   public function addBuku($sukses=0){
5   //get list kategori
6   $this->data['listkategori']= $this->bukumodel-
7   >getKategori('id');
8
9   // list Jenis Penawaran
10  $this->data['listJenisPenawaran']= $this->bukumodel-
11  >getJenisPenawaran('id');
12
13  // get detail sukses
14  $this->data['sukses']= $sukses;
15  // ------------------------------------------------------------
16  ---------------------------------------
17  // return data to view
18  $this->load->view('templates/menuUser',$this->data);
19  $this->load->view('user/tambahPenawaranView',$this->data);
20  $this->load->view('templates/footerUser',$this->data);
21  }
22
23  public function prosesAddBuku(){
24
25  $config['upload_path'] =  './upload/';
26  $config['allowed_types'] = 'gif|jpg|png';
27  $config['max_width']  = '5000';
28  $config['max_height']  = '5000';
29  $config['overwrite']  = TRUE;
30  $this->load->library('upload', $config);
31  $dataup = $this->upload->data();
32  $dataGambar=$dataup['orig_name'];
33
34  if (isset($_FILES['foto'])) {
35  $this->upload->do_upload("foto");
36  // $this->upload->do_upload('foto2');
37  $dataup = $this->upload->data();
38  $dataGambar=$dataup['orig_name'];
39  $datas['id_user']= $this->input->post('id_user');
40  $datas['id']=$this->input->post('idBuku');
41  $datas['tgl_post']=$this->input->post('tgl_post');
42  $datas['judul']=$this->input->post('judul');
43  $datas['penulis']=$this->input->post('pengarang');
```

```
$datas['penerbit']=$this->input->post('penerbit');
$datas['jml_hal']=$this->input->post('jml_hal');
$datas['harga']=$this->input->post('harga');
$datas['keterangan']=$this->input->post('keterangan');
$datas['tgl_post']=$this->input->post('tgl_post');
$datas['id_kategori']=$this->input->post('kategori');
$datas['id_penawaran']=$this->input->post('jenisPenawaran');
$datas['foto']=$dataGambar;
$datas['status']="Aktif";
$datas['status_penyewaan'] = "Tersedia";
/* $username=$this->session->userdata('username');
$id=$this->input->post('idBuku');*/
// $idPelajaran=$this->input->post('id_pelajaran');
//============================
if ($this->bukumodel->addBuku($datas)) {
$sukses=1;
}
else
$sukses=2;
redirect('dashboardControl');
 }
}
.
.
}
```

Kode 5.4 Script PHP pada class dashboardControl untuk memposting penawaran

Kode 5.4 merupakan implementasi dari perancangam *class diagram dashboardControl()* yang terdapat pada perancangan *class diagram* Gambar 4.10. Kode 5.4 merupakan *controller* dari kode program PHP pada *class dashboardControl* yang digunakan untuk melakukan posting penawaran buku baru. Terdapat fungsi *addBuku()* yang merupakan sebuah fungsi untuk memanggil halaman form penawaran buku dan mengambil data pada *class bukuModel()*. Fungsi *prosesAddBuku*() merupakan sebuah fungsi untuk melakukan proses posting penawaran buku kemudian disimpan kedalam database.

```
class bukuModel extends CI_Model{
public function getKategori($orderby){
$this->db->order_by($orderby);
$Q = $this->db->get('kategori');
return $Q;
}

public function getJenisPenawaran($orderby){
$this->db->order_by($orderby);
$Q = $this->db->get('jenis_penawaran');
return $Q;
}

public function addBuku($data){
$this->db->insert('post_buku',$data);
return TRUE;
)
.
.
```

```
20  }
```

Kode 5.5 Script PHP pada class bukuModel untuk menyimpan data penawaran

Kode 5.5 merupakan implementasi dari perancangam class bukuModel() yang yang terdapat pada perancangan *class diagram* Gambar 4.10. Kode 5.5 merupakan *script* PHP untuk melakukan penyimpanan data penawaran buku kedalam database sistem. Pada class *bukuModel* terdapat fungsi *getJenisPenawaran()* untuk melakukan pengambilan data jenis penawaran dan fungsi *getKategori()* untuk melakukan pengambilan data kategori buku pada database yang ditampilkan pada form penawaran buku. Fungsi *addBuku*() untuk melakukan *insert* data posting penawaran buku.

```
1  Class permintaanControl extends CI_Controller{
2  .
3  .
4  public function index() {
5    $id=$this->session->userdata('id');
6    $this->data['permintaanSaya']= $this->permintaanModel-
7  >getPermintaanSaya($id);
8    $this->load->view('templates/menuUser',$this->data);
9    $this->load->view('user/permintaanSaya_view',$this->data);
10   $this->load->view('templates/footerUser',$this->data);
11 }
12
13 public function reqSewa(){
14    $data['id_user']=$this->input->post('id_user');
15    $data['tanggal']=$this->input->post('tanggal');
16    $data['harga']=$this->input->post('harga');
17    $data['waktu_sewa']=$this->input->post('waktu_sewa');
18    $data['status']="belum dikonfirmasi";
19    $data['id_post']=$this->input->post('idBuku');
20
21    $notif['id_userPengirim']=$this->input->post('id_user');
22    $notif['id_userPenerima']=$this->input-
23 ('id_userPenerima');
24    $notif['isi_notif'] = 'Buku anda direquest oleh';
25    $notif['waktu']=$this->input->post('tanggal');
26    $notif['kode_notif']=3; //menuju ke tawaran masuk
27    $this->pesanModel->addNotifikasi($notif);
28
      $this->permintaanModel->addRequest($data);
29    redirect('permintaanControl/');
30 }
31
32 public function reqJual(){
33     $data['id_user']=$this->input->post('id_user');
34     $data['tanggal']=$this->input->post('tanggal');
35     $data['harga']=$this->input->post('harga');
36     $data['status']="belum dikonfirmasi";
37     $data['id_post']=$this->input->post('idBuku');
38
39     $notif['id_userPengirim']=$this->session->userdata('id');
       $notif['id_userPenerima']=$this->input-
st('id_userPenerima');
```

```
        $notif['isi_notif'] = 'Buku anda direquest oleh';
        $notif['waktu']=$this->input->post('tanggal');
        $notif['kode_notif']=3; //menuju ke tawaran masuk
        $this->pesanModel->addNotifikasi($notif);

        $this->permintaanModel->addRequest($data);
        redirect('permintaanControl/');
    }
    .
    .
    }
```

Kode 5.6 Script PHP pada class *permintaanControl* untuk request buku

Kode 5.6 merupakan implementasi dari perancangam *class diagram permintaanControl()* yang terdapat pada perancangan *class diagram* Gambar 4.11. Kode 5.6 merupakan *controller* dari kode program PHP pada *class permintaanControl* yang digunakan untuk melakukan proses request buku yang ditawarkan. Terdapat fungsi *index()* yang merupakan sebuah fungsi untuk menampilkan history buku yang direquest. Fungsi *reqSewa()* merupakan sebuah fungsi untuk melakukan proses request buku yang disewakan sedangkan fungsi *reqJual()* merupakan fungsi untuk melakukan request buku yang dijual.

```
1  class permintaanModel extends CI_Model{
2
3  public function getPermintaanSaya($ids){
4  $q = $this->db->query("SELECT DISTINCT abc.idPermintaan,
5  abc.tanggal, abc.idBuku, abc.idUserReq, abc.idPemilik,
6  abc.foto, abc.judul, abc.idKategori, abc.idPenawaran,
7  abc.harga, abc.waktu_sewa, abc.tukar, abc.status, abc.nama,
8  abc.no_telpon, abc.email, abc.alamat, d.content AS kategori
9  FROM kategori d
10 INNER JOIN (
11 SELECT DISTINCT ab.idPermintaan,ab.tanggal, ab.idBuku,
12 ab.idUserReq, ab.idPemilik, ab.foto, ab.judul, ab.idKategori,
13 ab.idPenawaran, ab.harga, ab.waktu_sewa, ab.tukar, ab.status,
14 f.nama, f.no telpon, f.email, f.alamat
15 FROM user f
16 INNER JOIN (
17 SELECT DISTINCT a.id AS idPermintaan, a.tanggal, b.id AS
18 idBuku, a.id_user AS idUserReq, c.nama, c.no_telpon, c.email,
19 c.alamat, b.id_user AS idPemilik, b.foto AS foto, b.judul AS
20 judul, b.id_kategori AS idKategori, b.id_penawaran AS
21 idPenawaran, a.harga, a.waktu_sewa, a.tukar, a.status
22 FROM permintaan a, post_buku b, user c
23 WHERE a.id_user =$ids
24 AND b.id = a.id_post
25 AND b.id_user != a.id_user
26 )ab ON f.id = ab.idPemilik
27 )abc ON d.id = abc.idKategori
28 ORDER BY abc.tanggal DESC
29 LIMIT 0 , 30");
30 return $q;
31 }
32
33 public function addRequest($data){
```

```
$this->db->insert('permintaan',$data);
return TRUE;
}
.
.
```

Kode 5.7 Script PHP pada class *permintaanModel()*

Kode 5.7 merupakan implementasi dari perancangam class permintaanModel yang yang terdapat pada perancangan *class diagram* Gambar 4.11. Kode 5.7 merupakan *script* PHP untuk melakukan menyimpan data request buku yang ditawarkan kedalam database sistem. Pada class *permintaanModel* terdapat fungsi *addRequest()* untuk melakukan *insert* data request dan fungsi *getPermintaanSaya()* untuk menampilkan data request buku yang dikirimkan oleh pengguna kepada user pemilik buku yang direquest.

5.6 Implementasi Layout Responsif

Pada bagian ini akan dijelaskan implementasi layout web *responsive* pada aplikasi Pencarian dan Penawaran Buku. Implementasi layout web *responsive* berupa implementasi kode program menggunakan CSS3. Implementasi layout web responsive mengacu pada parancangan layout *responsive* pada naskah dokumen 4.2.6. Implementasi grid layout web responsive menggunakan ukuran layout 960px kemudian dikonversi kedalam ukuran persen (%) seperti yang ditunjukan pada Kode 5.8.

```
/* Sistem grid dengan ukuran yang diasumsikan 960px kemudian
di konversi kedalam ukuran % */
#gridtemplate{

      width:93.75%;
      margin:auto;
}
.column1{
      width:6.25%; /*60px;*/
}
.column2{
      width:14.58333333333333%;/*140px;*/
}
.column3{
      width:22.91666666666667%;/*220px;*/
}

.colomn4{
      width:28.125%; /*300px;*/
}

.column4{
      width:31.25%; /*300px;*/
}
```

```
25
26  .column5{
27        width:39.58333333333333%; /*380px;*/
28  }
29  .column6{
30        width:47.91666666666667%; /*460px;*/
31  }
32  .column7{
33        width:56.25%; /*540px;*/
34  }
35  .column8{
36        width:64.58333333333333%;/*620px;*/
37  }
38  .column9{
39        width:72.91666666666667%; /*700px;*/
40  }
41  .column10{
42        width:81.25%; /*780px;*/
43  }
44  .column11{
45        width:89.58333333333333%; /*860px;*/
46  }
47  .column12{
48        width:97.91666666666667%;/*940px;*/
49  }
50  .break{
51        clear:both;
52        display:block;
53        overflow:hidden;
54  }
55  .column1, .column2, .column3, .column4, .column5, .column6,
56  .column7, .column8, .column9, .column10, .column11,
57  .column12{
58        background-color: transparent;
59        height:auto;
60        margin-left:1.041666666666667%;/*10px/960;*/
61        margin-right:1.041666666666667%;/*10px;*/
62        margin-bottom:0.5208333333333333%;
63        float:left;
64        text-align: left;
65  }
```

Kode 5.8 Kode CSS untuk mengatur ukuran *fluid grid*

Kode 5.8 merupakan implementasi kode CSS3 berdasarkan perancangan layout responsive pada Gambar 4.16. Implementasi kode program 5.8 berfungsi untuk mengatur ukuran grid dengan menerapkan teknik grid system. Dalam implementasi layout aplikasi Pencarian dan Penawaran Buku ini menggunakan grid ukuran 960 pixel dengan 12 kolom yang sudah dirancang pada Gambar 4.16 menggunakan teknik *fluid grid*. Proses perhitungan *grid system* menjadi satuan *fluid grid* dijelaskan pada naskah dokumen 4.2.6.

Setelah melakukan implementasi *layout grid* menggunakan teknik *grid system*, langkah selanjutnya adalah implementasi layout pada perangkat desktop

menggunakan *grid*. Kode 5.9 merupakan potongan kode program dari implementasi antamuka aplikasi pada perangkat desktop. Implementasi kode 5.7 berdasarkan perancangan layout pada Gambar 4.17.

```
@charset "utf-8";
/* CSS Document */
body{
	font-family: Arial, Helvetica, sans-serif;
	background-color: #000;
	background-repeat: no-repeat;
	background-position: center 40px;
	background-attachment: fixed;
	font-size: 14px;
}

#top_menu{
	background:url(../images/bg_black.gif);
	height:auto;
	color:#FFF;
	position:fixed;
	width:100%;
	margin:1.04166667%;
	font-size:1.125em;

}
#header{
	padding-left: 1%;
	margin-top: 8.5%;

}

#header img {
	width: 50%;
	max-width: 100%;
	height: auto;
}

#navbar{
	background:url(../images/bg_yellow.gif);
	height:auto;
	text-align:center;
}
#leftbar{
	font-size:1.125em;
	background-color:#FFF;
}

#sidemenu{
	font-size:1em;
	background-color:#FFF;
	margin:1.0416667%;
}
#footer{
	background-color:#151515;
	width:100%;
	margin:1.667%;
	text-align:center;
}
```

```
55
56  .
57  .
58  }
```

Kode 5.9 Kode CSS untuk mengatur antarmuka aplikasi pada perangkat *desktop*.

Setelah menyelesaikan layout untuk tampilan desktop, tahap selanjutnya yaitu membuat tampilan layout menjadi *responsive* pada perangkat *tablet* atau *smartphone*. Untuk membuat tampilan layout menjadi pada perangkat yang digunakan adalah dengan mengatur kode-kode CSS berdasarkan ukuran layar perangkat atau jendela browser. Implementasi Kode 5.10 dibuat berdasarkan perancangan layout pada Gambar 4.18 yang mana terdapat penyesuaian beberapa elemen yang menyesuaikan ukuran dari perangkat *tablet*.

```
1   @charset "utf-8";
2   /* CSS Document */
3   @media screen and (max-width:1024px){
4   #leftbar{
5         width:97.91667%;
6   }
7   #leftbar .news{
8         width:45.83333%; /* 440px/960 */
9   }
10  #barkanan{
11        width:97.91667%;
12        font-size:1.125em;
13        padding: 1.041667%;
14
15  }
16
17  #sidemenu .konten{
18        text-align:left;
19        width:45.91667%;
20        font-size:1.125em;
21        padding: 1.041667%;
22  }
23
24  #navbar ul{
25        display:none;
26  }
27  .slide_button{
28        display:block;
29  }
30
31  .
32  .
33  }
```

Kode 5.10 Kode CSS untuk mengatur antarmuka aplikasi pada perangkat *tablet*.

Kode 5.10 merupakan implementasi antarmuka aplikasi pada tampilan perangkat *tablet*. Dalam membuat aturan-aturan CSS pada perangkat *tablet* dengan menentukan terlebih dahulu range ukuran perangkat layarnya yaitu ukuran

minimal dan ukuran maksimal. Pada aplikasi Pencarian dan Penawaran Buku untuk implementasi anatrmuka pada perangkat *tablet* menggunakan ukuran maksimal 1024px yang diimplementasikan menggunakan teknik *media queries.* Ukuran tersebut kemudian didefinikan pada baris awal CSS menggunakan *media queries* dengan kode `@media screen and (max-width:1024px)` sehingga tampilan dapat menyesuaikan resolusi layar pada perangkat. Selanjutnya, mengatur ulang elemen-elemen menggunakan *layout grid* agar dapat menyesuaikan ukuran layar pada perangkat *tablet*.

```
1  @charset "utf-8";
2  /* CSS Document */
3  @media screen and (max-width:768px){
4  #header {
5  display:none;
6  }
7  #leftbar{
8        width:97.91667%;
9  }
10
11 #leftbar .news{
12       width:97.91667%;
13 }
14 #sidemenu{
15       width:97.91667%;
16 }
17 #sidemenu .konten{
18       text-align:left;
19       width:97.91667%;
20       font-size: 1.125cm;
21       padding: 1.041667%;
22 }
23
24 #navbar ul{
25       display:none;
26 }
27 .slide_button{
28       display:block;
29 }
30 .
31 .
32 }
```

Kode 5.11 Kode CSS untuk mengatur antarmuka pada *smartphone*.

Kode 5.11 merupakan implementasi antarmuka pada tampilan perangkat *smartphone* berdasarkan perancangan layout Gambar 4.19. Pada aplikasi Pencarian dan Penawaran Buku untuk implementasi anatrmuka pada perangkat *smartphone* menggunakan ukuran maksimal 768 pixel. Ukuran tersebut kemudian didefinikan pada baris awal CSS menggunakan *media queries* sehingga tampilan

dapat menyesuaikan resolusi layar pada perangkat. Selanjutnya, mengatur ulang elemen-elemen dan menyusun semua layout memanjang kebawah untuk perangkat *smartphone* menggunakan *layout grid* sesuai dengan perancangan layout pada Gambar 4.19 kemudian diimplementasikan menjadi kode program yang ditunjukan pada Kode 5.11.

5.7 Implementasi Antarmuka Aplikasi

Pada bagian ini menjelaskan mengenai implementasi antarmuka berdasarkan perancangan antarmuka yang telah dilakukan sebelumnya pada dokumen nomor 4.2.7. Berikut adalah implementasi antarmuka halaman menu utama, halaman pencarian, halaman posting penawaran buku dan halaman request penawaran buku.

5.7.1 Implementasi Antarmuka halaman utama

Tampilan antarmuka halaman utama merupakan implementasi dari perancangan antarmuka pada Gambar 4.20. Antarmuka halaman utama merupakan tampilan awal ketika pengguna mengakses aplikasi. Pada halaman utama berisi pilihan menu utama dan daftar penawaran buku. Pilihan menu utama terdiri dari menu peta lokasi yang berfungsi untuk menampilkan informasi lokasi pengguna saat itu dan lokasi pengguna yang menawarkan bukunya. Terdapat menu cari buku berfungsi untuk melakukan pencarian buku. Terdapat menu posting penawaran buku yang berfungsi untuk menambah atau memposting data penawaran buku baru kedalam sistem. Pada halaman utama sistem juga ditampilkan daftar penawaran buku, user dapat menampilkan detail informasi data penawaran buku yang dipilih. Kode 5.12 merupakan bagian kode implementasi antarmuka dari perancangan antarmukan halaman utama.

```
div class="column12  menu_utama">
<div class="menutama" style="text-align:center; padding-top: 10px;
">
<div class="column4 text-center" id="menu">
<a href="<?php echo base_url(); ?>bukuControl/petaLokasi"><img
src="<?php echo base_url(); ?>images/marker.png" alt="Dashboard"
class="img-circle"></a>
<a href="<?php echo base_url(); ?>bukuControl/petaLokasi
"><h4>Peta Lokasi</h4></a>
</div>

<div class="column4 text-center" id="menu">
<a href="<?php echo base_url(); ?>bukuControl/listKategori"><img
src="<?php echo base_url(); ?>images/caribuku.png" alt="cari buku"
class="img-circle"></a>
```

```
<a href="<?php echo base_url();
?>bukuControl/listKategori"><h4>Cari Buku</h4></a>
</div>

<div class="column4 text-center" id="menu">
<a href="<?php echo base_url(); ?>dashboardControl/addBuku"><img
src="<?php echo base_url(); ?>images/iklansaya.png" alt="iklan
saya" class="img-circle" ></a>
<a href="<?php echo base_url(); ?>bukuControl/addBuku"><h4>Posting
Penawaran Buku</h4></a>
</div>
</div> <!-- end menu utama -->

<div id="leftbar" class="column9">
<div class="column12 judul">
<p>Iklan Penawaran Buku</p>
</div> <!-- end judul -->
<div id="data-paging">
</div>
.
.
```

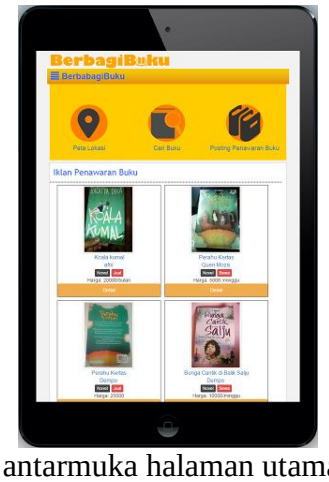
Kode 5.12 Implementasi antarmuka halaman utama

Gambar 5.2 Tampilan antarmuka halaman utama pada desktop

Gambar 5.2 merupakan tampilan antarmuka halaman utama aplikasi ketika diakses pada perangkat *desktop* dengan ukuran layar 14 inches (width >1024px). Gambar 5.2 merupakan tampilan antamuka halaman utama dari hasil implementasi antarmuka Kode 5.12. Implementasi kode program yang ditunjukan pada Kode 5.12 dibuat berdasarkan perancangan antarmuka Gambar 4.20

Gambar 5.3 merupakan tampilan antarmuka halaman utama aplikasi ketika diakses pada perangkat *tablet* dengan ukuran layar 7 inches (width ≤1024px).

Gambar 5.3 merupakan tampilan antamuka halaman utama dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.21



Gambar 5.3 Tampilan antarmuka halaman utama pada perangkat *tablet*

Gambar 5.4 merupakan tampilan antarmuka halaman utama aplikasi ketika diakses pada perangkat *smartphone* dengan ukuran layar 4 inches (width ≤768px). Gambar 5.4 merupakan tampilan antamuka halaman utama dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.21. Terdapat penyesuai beberapa elemen sehingga tampilan dapat menyesuaikan ukuran layar dari perangkat *smartphone.*

Gambar 5.4 Tampilan antarmuka halaman utama pada *smartphone*

5.7.2 Implementasi Antarmuka Pencarian

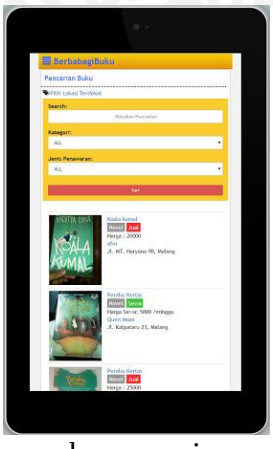
Tampilan antarmuka pencarian penawaran buku merupakan implementasi dari perancangan antarmuka pada Gambar 4.22. Halaman pencarian buku merupakan tampilan antarmuka yang berisi daftar pencarian buku berserta alamat lokasi user yang menawarkan buku dan jarak menuju lokasi. Pada halaman pencarian user dapat melakukan filter pencarian buku yang dibutuhkan sesuai kriteria dari masukan user. Kode 5.13 merupakan bagian kode implementasi antarmuka dari perancangan antarmuka pencarian penawaran buku pada Gambar 4.22.

```
<tbody>
<?php foreach ($bukuTerdekat->result() as $key ): ?>
<tr>
<td><a href="<?php echo
base_url().'bukuControl/detailBuku/'.$key->idPost ?>"><img
class="img-listpencarian" src="<?php echo
base_url().'upload/'.$key->foto1 ?>"></a></td>
<td><a href="#"><?php echo $key->judul; ?></a></br>
<a data-value="1"><span class="status-metro status-disabled"
title="kategori"><?php echo $key->kategori; ?></span></a>
<?php if($key->penawaran =="Sewa") { ?>
<a data-value="1"><span class="status-metro status-active"
title="sewa">Sewa</span></a>
<?php }if($key->penawaran =="Jual"){ ?>
<a data-value="2"><span class="status-metro status-suspended"
title="jual">Jual</span></a>
<?php } ?>
<br> <a href="<?php echo
base_url().'bukuControl/profilUser/'.$key->id_user ?>"> <?php echo
$key->nama; ?></a><br>
<?php echo $key->alamat; ?>
<br> ± <?php echo $key->jarak; ?> km
</td>
</tr>
<?php endforeach ?>
</tbody> <!-- end tbody -->.
```

Kode 5.13 Implementasi antarmuka halaman pencarian buku

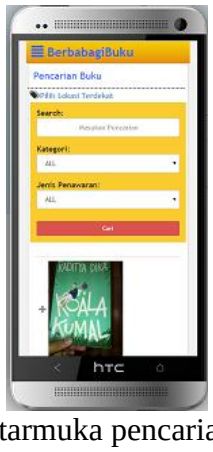
Gambar 5.5 Tampilan antarmuka halaman pencarian buku pada perangkat *desktop*

Gambar 5.5 merupakan tampilan antarmuka pencarian penawaran buku ketika diakses pada perangkat *desktop* dengan ukuran layar 14 inches (width >1024px). Gambar 5.5 merupakan tampilan antamuka pencarian penawarn buku dari hasil implementasi antarmuka Kode 5.13. Implementasi kode program yang ditunjukan pada Kode 5.13 dibuat berdasarkan perancangan antarmuka Gambar 4.22.



Gambar 5.6 Tampilan antarmuka pencarian penawaran buku pada *tablet*

Gambar 5.6 merupakan tampilan perangkat lunak halaman pencarian buku ketika diakses perangkat *tablet* (width<1024px). Gambar 5.6 merupakan tampilan antamuka pencarian buku dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.23. Terdapat penyesuai beberapa elemen sehingga tampilan dapat menyesuaikan ukuran layar dari perangkat *tablet.*



Gambar 5.7 Tampilan antarmuka pencarian buku pada *smartphone*

Gambar 5.7 merupakan tampilan antarmuka pencarian penawaran buku ketika diakses pada perangkat *smartphone* dengan ukuran layar 4 inches (width ≤768px). Gambar 5.7 merupakan tampilan antamuka pencarian penawaran buku dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.23. Terdapat penyesuai beberapa elemen sehingga tampilan dapat menyesuaikan ukuran layar dari perangkat *smartphone.*

5.7.3 Implementasi Antarmuka Posting Penawaran Buku

Antarmuka posting penawaran buku merupakan implementasi dari perancangan antarmuka pada Gambar 4.24. Antarmuka posting penawaran buku merupakan tampilan form untuk memasukan data penawaran buku baru. Pada

form posting penawaran buku pertama kali user diminta untuk memilih jenis penawaran buku yang akan ditawarkan selanjutkan sistem akan menampilkan isian form sesuai dengan jenis penawaran buku yang dipilih. Kode 5.14 merupakan bagian kode implementasi antarmuka dari perancangan antarmuka posting penawaran buku

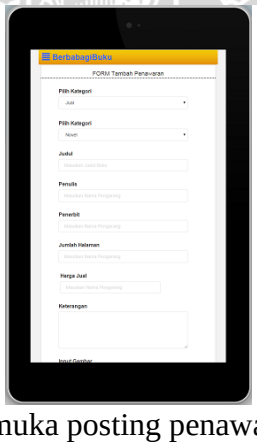
```
<div class="form-group">
<label for="exampleInputEmail1">Judul</label>
<input type="text" name="judul" class="form-control"
id="exampleInputEmail1" placeholder="Masukan Judul Buku">
</div>

<div class="form-group">
<label for="exampleInputEmail1">Pengarang</label>
<input type="text" name="pengarang" class="form-control"
id="exampleInputEmail1" placeholder="penulis...">
</div>
<div class="form-group">
<label for="exampleInputEmail1">Penerbit</label>
<input type="text" name="penerbit" class="form-control"
id="exampleInputEmail1" placeholder="penerbit...">
</div>
<div class="form-group">
<label for="exampleInputEmail1" >Jumlah Halaman</label>
<input type="text" name="jml_hal" class="form-control"
id="exampleInputEmail1" placeholder="masukan jumlah halaman">
</div>
<div class="form-group">
<label for="exampleInputEmail1" >Kategori</label>
<select name="kategori" class="form-control">
  <?php foreach ($listkategori->result() as $key) { ?>
<option  value="<?php echo $key->id; ?>"> <?php echo $key-
>content; ?></option>
<?php } ?>
</select>
</div>
<div class="form-group">
<label for="exampleInputEmail1" >Harga</label>
<input type="text" name="harga" class="form-control"
id="exampleInputEmail1" placeholder="masukan harga buku">
</div>
```

Kode 5.14 Implementasi antarmuka posting penawaran buku

Gambar 5.8 merupakan tampilan antarmuka pencarian penawaran buku ketika diakses pada perangkat *desktop* dengan ukuran layar 14 inches (width >1024px). Gambar 5.8 merupakan tampilan antamuka posting penawarn buku dari hasil implementasi antarmuka Kode 5.14. Implementasi kode program yang ditunjukan pada Kode 5.14 dibuat berdasarkan perancangan antarmuka Gambar 4.24

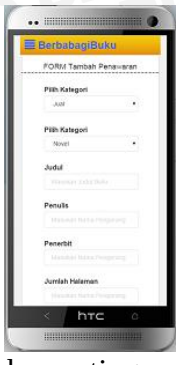
Gambar 5.8 Tampilan antarmuka posting penawaran buku pada *desktop*

Gambar 5.9 merupakan tampilan antamuka posting penawaran buku ketika diakses perangkat *tablet* (width=<1024px). Gambar 5.9 merupakan tampilan antamuka posting penawaran dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.25. Terdapat penyesuai beberapa elemen sehingga tampilan dapat menyesuaikan ukuran layar dari perangkat *tablet.*



Gambar 5.9 Tampilan antarmuka posting penawaran buku pada tablet

Gambar 5.10 merupakan tampilan antarmuka posting penawaran buku ketika diakses pada perangkat *smartphone* dengan ukuran layar 4 inches (width ≤768px). Gambar 5.10 merupakan tampilan antamuka posting penawaran buku dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.25. Terdapat penyesuai beberapa elemen sehingga tampilan dapat menyesuaikan ukuran layar dari perangkat *smartphone.*



Gambar 5.10 Tampilan antarmuka posting penawaran buku pada *smartphone*

5.7.4 Implementasi Antarmuka Request Penawaran Buku

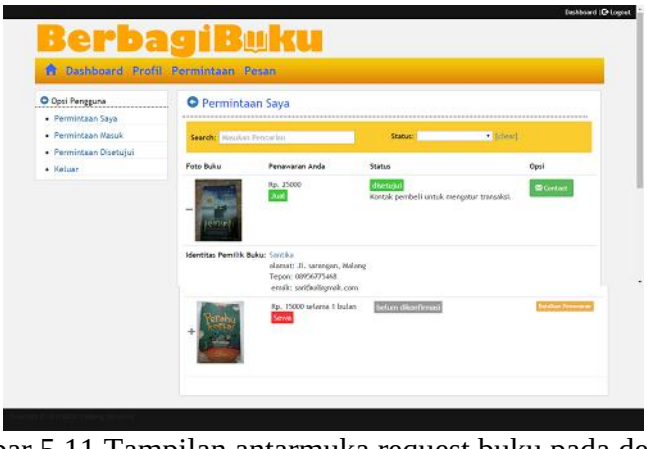
Antarmuka request penawaran buku merupakan implementasi dari perancangan antarmuka pada Gambar 4.27. Antarmuka request penawaran buku merupakan tampilan history atau daftar penawaran buku yang direquest untuk menunggu konfirmasi dari user pemilik buku yang direquest. Kode 5.15 merupakan bagian kode implementasi antarmuka dari perancangan antarmuka daftar buku yang direquest

```
<tbody>
<?php foreach ($permintaanSaya->result() as $key ): ?>
<tr>
<td><a href="<?php echo
base_url().'bukuControl/detailBuku/'.$key->idBuku ?>"><img
class="img-listpencarian" src="<?php echo
base_url().'upload/'.$key->foto ?>"></a></td>
<td><input type="hidden" name="idPemilik" value="<?php echo $key-
>idPemilik; ?>"></input>
<a href="#"><?php echo $key->nama; ?></a> <br>
            alamat: <?php echo $key->alamat; ?><br>
            Tepon: <?php echo $key->no_telpon; ?> <br>
            email: <?php echo $key->email; ?>
</td>
<td>Rp. <?php echo $key->harga; ?>
<?php if ($key->idPenawaran==1){ ?>
<br><span class="status-metro status-active"
title="jual">Jual</span>
<?php } elseif ($key->idPenawaran==2){ ?> selama <?php echo $key-
>waktu_sewa; ?> <br> <span class="status-metro status-suspended"
title="sewa">Sewa</span>
<?php } ?>
</td>

<?php if ($key->status=="disetujui"){ ?>
<td data-value="1"><span class="status-metro status-active"
title="Active"><?php echo $key->status; ?></span>
<p>Kontak pembeli untuk mengatur transaksi.</p>
</td>
<?php } elseif ($key->status=="belum dikonfirmasi"){ ?>
<td data-value="2"><span class="status-metro status-disabled"
title="Disabled"><?php echo $key->status; ?></span></td>
```

```
<?php } else {?>
<td data-value="3"><span class="status-metro status-suspended"
title="Suspended"><?php echo $key->status; ?></span></td>
<?php } ?>
<?php if ($key->status=="disetujui"){ ?>
<td data-value="1"><a data-toggle="modal" data-id="<?php echo
$key->idBuku; ?>" data-idd="<?php echo $key->idPemilik; ?>"
title="Add this item" class="open-AddBookDialog btn btn-success"
href="#addBookDialog"><span class="glyphicon glyphicon-
envelope"></span> Contact</a></td>
<?php } elseif ($key->status=="belum dikonfirmasi"){ ?>
<td><a href="<?php echo
base_url().'permintaanControl/hapusPermintaan/'.$key->idPermintaan
?>" onclick="return confirm('Anda yakin ingin membatalkan
penawaran anda?')"><button type="button" class="btn btn-warning
btn-xs" title="hapus">Batalkan Penawaran</button></a></td>
<?php } else {?>
<td><a href="<?php echo
base_url().'permintaanControl/hapusPermintaan/'.$key->idPermintaan
?>" onclick="return confirm('Anda yakin ingin menghapus penawaran
anda?')"><button type="button" class="btn btn-danger btn-xs"
title="hapus"><span class="glyphicon glyphicon-
remove"></span>Hapus</button></Hapus></span></td>
 <?php } ?>
</tr>
<?php endforeach ?>
</tbody>
```

Kode 5.15 Implementasi antarmuka posting penawaran buku

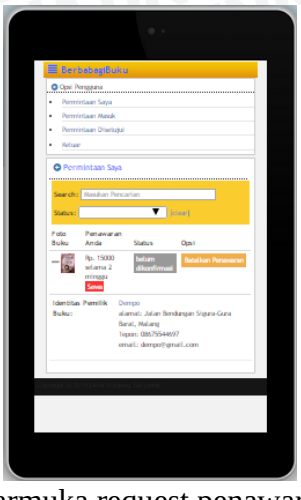
Gambar 5.11 merupakan tampilan antarmuka daftar buku yang direquest ketika diakses pada perangkat *desktop* dengan ukuran layar 14 inches (width >1024px). Gambar 5.1 merupakan tampilan antamuka request penawarn buku dari hasil implementasi antarmuka Kode 5.15. Implementasi kode program yang ditunjukan pada Kode 5.15 dibuat berdasarkan perancangan antarmuka Gambar 4.28



Gambar 5.11 Tampilan antarmuka request buku pada desktop

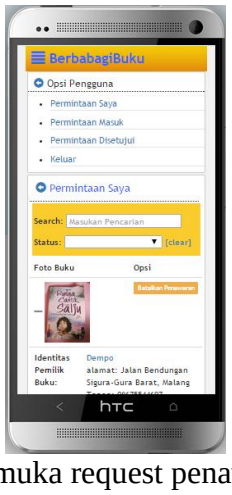
Gambar 5.12 merupakan tampilan antamuka daftar penawaran buku yang direquest ketika diakses perangkat *tablet* (width<1024px). Gambar 5.12

merupakan tampilan antamuka request penawaran dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.29. Terdapat penyesuai beberapa elemen sehingga tampilan dapat menyesuaikan ukuran layar dari perangkat *tablet.*



Gambar 5.12 Tampilan antarmuka request penawaran buku pada tablet

Gambar 5.13 merupakan tampilan antarmuka daftar penawaran buku yang direquest ketika diakses pada perangkat *smartphone* dengan ukuran layar 4 inches (width ≤768px). Gambar 5.13 merupakan tampilan antamuka posting penawaran buku dari hasil implementasi antarmuka berdasarkan perancangan antarmuka Gambar 4.29. Terdapat penyesuai beberapa elemen sehingga tampilan dapat menyesuaikan ukuran layar dari perangkat *smartphone.*



Gambar 5.13 Tampilan antarmuka request penawaran buku pada *smartphone*

BAB VI
PENGUJIAN DAN ANALISIS

Pada Bab ini berisi tentang pengujian dan analisis pada aplikasi yang telah dibangun. Pengujian dilakukan antara lain pengujian validasi untuk mengetahui apakah aplikasi yang dibangun sudah berjalan seperti yang diharapkan, pengujian *responsive test* untuk menguji layout web responsive terhadap ukuran layar dari perangkat yang digunakan, pengujian perhitungan jarak untuk menguji akuransi jarak antar lokasi dan pengujian non-fungsional pada aplikasi Pencarian dan Penawaran Buku.

6.1 Hasil Pengujian

Hasil pengujian merupakan hasil uji coba aplikasi yang dilakukan pada tahap pengujian sistem berdasarkan perancangan pengujian yang sudah dibuat sebelumnya. Pengujian aplikasi dilakukan untuk mengetahui apakah aplikasi yang dibangun sudah berjalan sesuai dengan yang diharapkan pada tahap perancangan pengujian pada 4.3.

6.1.1 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui kesesuaian aplikasi yang dibangun sudah berjalan seperti yang diharapkan. Pengujian validasi ini dilakukan sesuai perancangan pengujian atau desain uji yang sudah dirancang pada bab sebelumnya.

Tabel 6.1 merupakan hasil pengujian validasi yang telah dilakukan dengan melihat kesesuaian antara hasil pengujian fungsional sistem dengan hasil yang diharapkan pada tahap perancangan. Kebutuhan fungsional yang diuji berdasarkan test case yang terdapat pada Tabel 4.13, Tabel 4.14, Tabel 4.15, Lampiran 7, Lampiran 8 dan Lampiran 9.

Tabel 6.1 Hasil Pengujian Validasi

Test Case	**Hasil yang diharapkan**	**Hasil yang didapatkan**	**Status Validasi**
UVL_01 (Tabel 4.13)	Aplikasi dapat manampilkan daftar pencarian buku yang berjudul “koala kumal” beserta informasi lokasi dan	Aplikasi dapat manampilkan daftar pencarian buku yang berjudul “koala kumal” beserta informasi lokasi dan	VALID

	jarak menuju lokasi berdasarkan hasil perhitungan koordinat perangkat user dan koordinat alamat user penawar menggunakan rumus Haversine Formula.	jarak menuju lokasi berdasarkan hasil perhitungan koordinat perangkat user dan koordinat alamat user penawar menggunakan rumus Haversine Formula.	
UVL_02 (Tabel 4.14)	Data penawaran buku baru dengan judul "koala kumal" yang dimasukan oleh aktor berhasil tersimpan ke dalam database dan tampil pada halaman daftar penawaran buku sesuai dengan data yang dimasukan	Data penawaran buku baru dengan judul "koala kumal" yang dimasukan oleh aktor berhasil tersimpan ke dalam database dan tampil pada halaman daftar penawaran buku sesuai dengan data yang dimasukan	VALID
UVL_03 (Tabel 4.15)	Data penawaran buku yang direquest atau dipesan dengan judul "koala kumal" tersimpan pada history daftar buku yang direquest dan permintaan request berhasil dikirimkan kepada user pemilik buku yang direquest bernama "afni"	Data penawaran buku yang direquest atau dipesan dengan judul "koala kumal" tersimpan pada history daftar buku yang direquest dan permintaan request berhasil dikirimkan kepada user pemilik buku yang direquest bernama "afni"	
UVL_04 (lampiran 7)	Status permintaan request berhasil diubah dan menyimpanya pada history "permintaan yang disetujui" jika permintaan tersebut disetujui	Status permintaan request berhasil diubah dan menyimpanya pada history "permintaan yang disetujui" jika permintaan tersebut disetujui	
UVL_05 (lampiran 8)	Data pesan berhasil dikirimkan kepada user pemilik buku yang dipilih sesuai dengan masukan user.	Data pesan berhasil dikirimkan kepada user pemilik buku yang dipilih sesuai dengan masukan user.	
UVL_06 (lampiran 9)	1. Aplikasi dapat mengubah status penawaran buku berjudul "koala kumal" yang ditandai oleh aktor "sedang disewakan" maka status penawaran buku berubah menjadi "sedang disewakan" berserta waktu peminjamanserta dapat mengubah statusnya kembali jika buku sudah dikembalikan	1. Aplikasi dapat mengubah status penawaran buku berjudul "koala kumal" yang ditandai oleh aktor "sedang disewakan" maka status penawaran buku berubah menjadi "sedang disewakan" berserta waktu peminjaman serta dapat mengubah statusnya kembali jika buku sudah dikembalikan	VALID

	2. Aplikasi dapat mengubah status buku yang berjudul "perahu kertas" yang ditandai aktor menjadi tidak aktif pada jenis penawaran "jual" dan penawaran buku tersebut tidak lagi ditampilkan pada sistem	2. Aplikasi dapat mengubah status buku yang berjudul "perahu kertas" yang ditandai aktor menjadi tidak aktif pada jenis penawaran "jual" dan penawaran buku tersebut tidak lagi ditampilkan pada sistem	

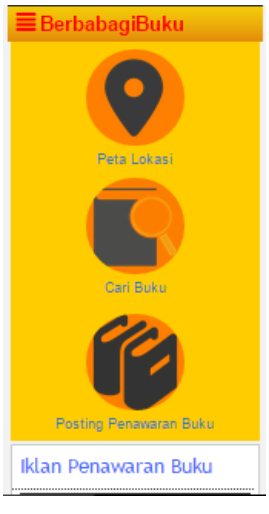
6.1.2 Pengujian Layout Web Responsif

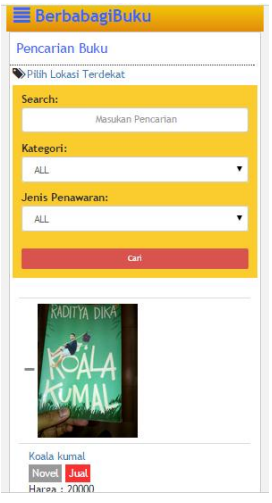
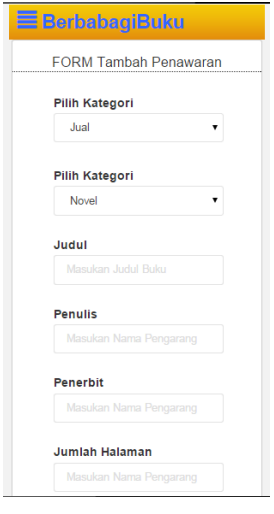
Pengujian *layout web responsive* dilakukan untuk mengetahui apakah antar muka aplikasi sudah dapat berjalan secara responsive saat diakses dalam lingkungan perangkat dengan ukuran layar yang berbeda. Pengujian dilakukan dengan melakukan akses terhadap aplikasi melalui perangkat dengan ukuran layar yang berbeda. Perangkat yang digunakan meliputi *smartphone*, *tablet* dan *desktop.*

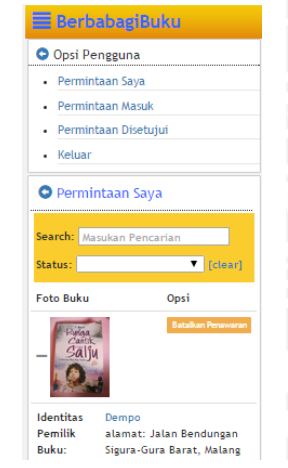
1. Pengujian Layout pada Ukuran Layar *Smartphone*

Pengujian *layout web responsive* pada layar *smartphone* dilakukan sesuai dengan *test case* pengujian pada perancangan pengujian Tabel 4.16. Tabel 6.2 merupakan hasil pengujian layout web responsive pada perangkat *smartphone*.

Tabel 6.2 Hasil pengujian antarmuka aplikasi pada layar *smartphone*

Antarmuka Aplikasi	**Hasil yang diharapkan**	**Hasil yang didapatkan**	**Status**
Halaman Utama pada Gambar 4.21	Konten dan layout halaman utama mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.21	Konten dan layout halaman utama mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.21	VALID 

Halaman Pencarian pada Gambar 4.23	Konten dan layout halaman pencarian mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.23	Konten dan layout halaman pencarian mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.23	VALID 
Halaman Posting Penawaran Buku pada Gambar 4.25	Konten dan layout halaman dashboard penawaran mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.25	Konten dan layout halaman dashboard penawaran mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.25	VALID 
Halaman Detail Buku pada Gambar 4.27	Konten dan layout halaman detail buku mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.27	Konten dan layout halaman detail buku mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.27	VALID

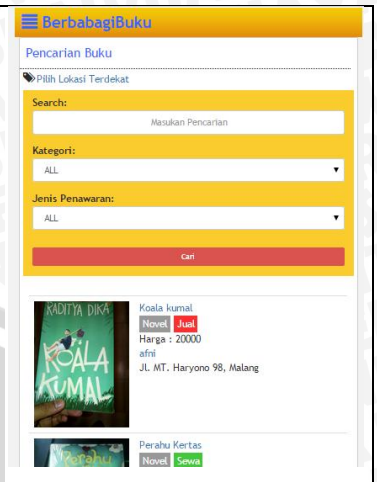
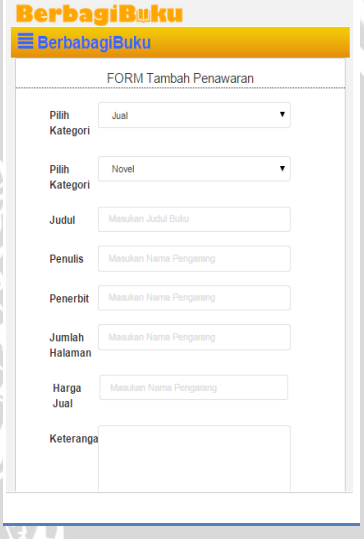
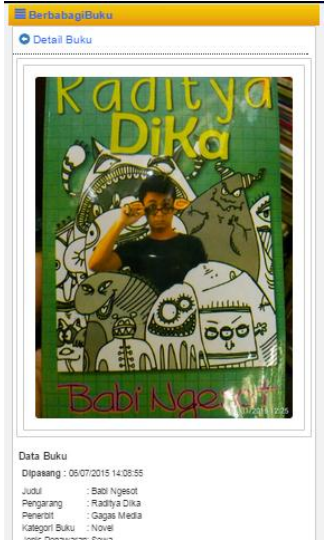
Halaman request penawaran buku pada Gambar 4.29	Kontent dan layout halaman request penawaran buku mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.29	Kontent dan layout halaman request penawaran buku mampu menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan perancangan antarmuka Gambar 4.29	VALID 

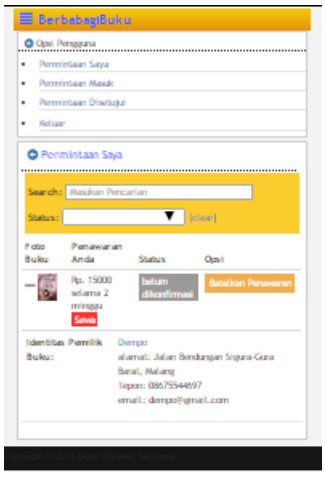
2. Pengujian Layout pada Ukuran Layar *Tablet*

Pengujian layout web responsive pada layar *tablet* dilakukan sesuai dengan test case pengujian pada perancangan pengujian Tabel 4.17. Tabel 6.3 merupakan hasil pengujian layout web responsive pada perangkat *tablet*.

Tabel 6.3 Hasil pengujian antarmuka aplikasi pada layar *tablet*

Antarmuka aplikasi	**Hasil yang diharapkan**	**Hasil yang didapatkan**	**Status**
Halaman Utama pada Gambar 4.21	Konten dan layout halaman utama mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.21	Konten dan layout halaman utama mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.21	VALID
Halaman Pencarian pada Gambar 4.23	Konten dan layout halaman pencarian mampu menyesuaikan ukuran layar *tablet* dengan baik	Konten dan layout halaman pencarian mampu menyesuaikan ukuran layar *tablet* dengan baik	VALID

	sesuai dengan perancangan antarmuka Gambar 4.23	sesuai dengan perancangan antarmuka Gambar 4.23	
Halaman Posting Penawaran pada Gambar 4.25	Konten dan layout halaman dashboard penawaran mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.25	Konten dan layout halaman dashboard penawaran mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.25	VALID 
Halaman Detail Buku pada Gambar 2.47	Konten dan layout halaman detail buku mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.27	Konten dan layout halaman detail buku mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.27	VALID 

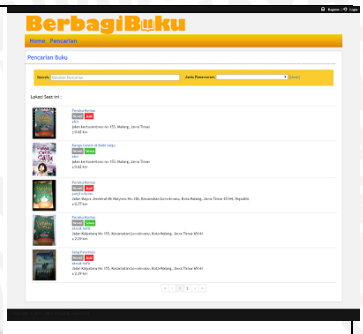
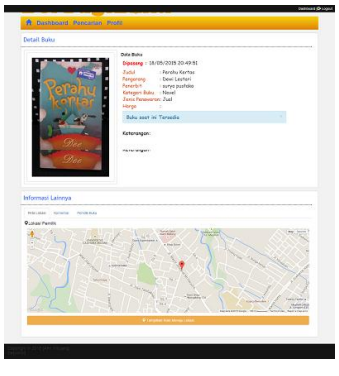
Halaman Rute Lokasi pada Gambar 4.29	Kontent dan layout halaman rute lokasi mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.29	Kontent dan layout halaman rute lokasi mampu menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan perancangan antarmuka Gambar 4.29	VALID 

3. Pengujian Layout pada Ukuran Layar Desktop

Pengujian layout web responsive pada layar desktop dilakukan sesuai dengan test case pengujian pada perancangan pengujian Tabel 4.18. Tabel 6.4 merupakan hasil pengujian layout web responsive pada perangkat *laptop*.

Tabel 6.4 Hasil pengujian antarmuka aplikasi pada layar *laptop*

Antarmuka Aplikasi	**Hasil yang diharapkan**	**Hasil yang didapatkan**	**Status**
Halaman Utama pada Gambar 4.20	Konten dan layout halaman utama mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.20	Konten dan layout halaman utama mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.20	VALID
Halaman Pencarian pada Gambar	Konten dan layout halaman pencarian mampu menyesuaikan	Konten dan layout halaman pencarian mampu menyesuaikan	VALID

4.22	ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.22	ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.22	
Halaman Posting Penawaran pada Gambar 4.24	Konten dan layout halaman dashboard penawaran mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.24	Konten dan layout halaman dashboard penawaran mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.24	VALID
Halaman Detail Buku pada Gambar 4.26	Konten dan layout halaman detail buku mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.26	Konten dan layout halaman detail buku mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar 4.26	VALID 
Halaman request penawaran buku pada Gambar 2.28	Kontent dan layout halaman request buku mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan antarmuka Gambar	Kontent dan layout halaman request buku mampu menyesuaikan ukuran layar laptop dengan baik sesuai dengan perancangan	VALID

	4.28	antarmuka Gambar 4.28	

6.1.3 Pengujian Haversine Formula

Pengujian berikutnya yang dilakukan adalah uji coba pada haversine formula. Pengujian jarak dilakukan untuk menguji penentuan lokasi terdekat berdasarkan jarak dengan membandingan jarak asli menggunakan *speedometer* dengan hasil perhitungan jarak rumus Haversine Formula. Pengujian ketepatan jarak yang ditampilkan pada aplikasi bertujuan untuk melakukan pencarian buku dengan ditampilkan jarak lokasi supaya pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya sehingga mempermudah transaksi peminjaman dan pengembalian buku. Dalam contoh kasus uji coba ini diketahui koordinat posisi user yang ditunjukan pada Tabel 6.5.

Tabel 6.5 Studi kasus posisi perangkat user

Lokasi perangkat user	**Latitude**	**longitude**	**Masukan**
Jalan Kertosentono, Lowokwaru, Malang	-7.9487086	112.6097988	Pengguna mencari buku yang berjudul "perahu kertas"

Dalam kasus uji coba ini pengguna mencari buku yang berjudul "perahu kertas" dari hasil pencarian terdapat 5 judul data penawaran buku yang sama (studi kasus buku yang dicari berjudul "perahu kertas") dengan pemilik buku yang berbeda sehingga dari kasus ini memiliki lokasi alamat pemilik buku yang berbeda. Tabel 6.6 merupakan contoh kasus terdapat 5 user penawar yang menawarkan buku yang berjudul "perahu kertas" dengan alamat lokasi yang berbeda. Dari hasil pengujian jarak diharapkan pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya sehingga mempermudah transaksi peminjaman dan pengembalian buku.

Tabel 6.6 Studi kasus lokasi user penawaran buku

Pengguna yang menawarkan buku	**Alamat**	**Latitude**	**longitude**
User 1	Jl. MT. Haryono	-7.949394363066064	112.61494747252198

(Afny)	98, Malang		
User 2 (Dempo)	Jalan Bendungan Sigura-Gura Barat, Malang	-7.957633387480265	112.60751344509117
User 3 (Quen Moza)	Jl. Kalpataru 23, Malang	-7.950780696966493	112.63145384571612
User 4 (Santika)	Jl. Sarangan 27, Malang	-7.959704421311055	112.63212473226315
User 5 (Tebe)	Jl. Terusan Borobudur 33, Mlang	-7.940563128792238	112.63964559549254

Dalam uji coba ini pengguna diletakan pada posisi pada latitude -7.9487086 dan longitude 112.6097988 (Lihat tabel 6.5). Dengan menggunakan rumus pada Persamaan 2.2 untuk menghitung nilai hasil perhitungan dari Haversine Formula untuk tiap-tiap lokasi user penawar. Tabel 6.7 merupakan hasil perhitungan jarak lokasi dengan Haversine Formula.

Tabel 6.7 Hasil perhitungan jarak lokasi dengan Haversine Formula

Pengguna yang menawarkan buku	**Jarak (dalam km)**			**Persentase kesalahan (selisih/jarak sebenarnay) %**
	Jarak aplikasi	**Jarak sebenarnya**	**selisih**	
User_1	0.57	0.7	0.13	0.18%
User_2	1.03	1.2	0.27	0.20%
User_3	2.39	2.9	0.51	0.17%
User_4	2.75	3.4	0.65	0.19%
User_5	3.41	4.2	0.79	0.18%
Kesalahan rata-rata			0.47	

6.1.4 Pengujian Compatibility

Pengujian compatibility dilakukan untuk mengetahui kompatibilitas antarmuka sistem dapat menjalankan fungsi yang ada pada kebutuhan fungsional sistem dengan baik pada jenis browser yang berbeda pada perangkat desktop maupun perangkat bergerak. Dalam uji coba ini dilakukan menggunakan browser Chorme, Internet Explorer, Firefox, Safari dan Android Browser dilakukan berdasarkan perancangan pengujian Tabel 4.20. Tabel 6.8 merupakan hasil dari pengujian kompatibilitas sistem.

Tabel 6.8 Hasil pengujian kompatibilitas sistem

Kebutuhan Fungsional	Status Validitas Hasil Pengujian Kompatibilitas Sistem				
	Chrome	Firefox	IE	Safari	Android Browser
SRS_01	VALID	VALID	VALID	VALID	VALID
SRS_02	VALID	VALID	VALID	VALID	VALID
SRS_03	VALID	VALID	VALID	VALID	VALID
SRS_04	VALID	VALID	VALID	VALID	VALID
SRS_05	VALID	VALID	VALID	VALID	VALID
SRS_06	VALID	VALID	VALID	VALID	VALID
SRS_07	VALID	VALID	VALID	VALID	VALID
SRS_08	VALID	VALID	VALID	VALID	VALID
SRS_09	VALID	VALID	VALID	VALID	VALID
SRS_10	VALID	VALID	VALID	VALID	VALID
SRS_11	VALID	VALID	VALID	VALID	VALID
SRS_12	VALID	VALID	VALID	VALID	VALID

5.2 Analisis Pengujian

Proses analisis terhadap hasil pengujian dilakukan untuk mendapatkan kesimpulan dari hasil pengjian sistem yang sudah dilakukan. Proses analisis terhadap pengujian sistem dilakukan dengan melihat kecocokan antara hasil yang diharapkan pada perancangan pengujian 4.3 dengan hasil uji coba yang sudah dilakukan pada tahap pengujian. Proses analisis yang dilakukan meliputi analisis pengujian validasi, analisis pengujian layout web responsive, analisis pengujian Haversine Formula, analisis pengujian kompatibitas sistem.

5.2.1 Analisis Pengujian Validasi

Proses analisis terhadap pengujian validasi dilakukan dengan melihat kesesuaian antara hasil pengujian fungsional sistem apakah sudah sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian sistem yang ada pada naskah dokumen 4.3.3.1. Berdasarkan hasil pengujian validasi yang ditunjukan pada Tabel 6.1 dapat disimpulkan bahwa Aplikasi Pencarian dan Penawaran Buku berhasil memenuhi kebutuhan fungsional yang telah dijabarkan pada tahap perancangan sistem. Hasil yang didapat pada tahap pengujian validasi didapatkan

validitas sebesar 100% sudah sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian sistem pada naskah dokumen 4.3.3.1

5.2.2 Analisis Pengujian Layout Responsive

Proses analisis terhadap pengujian layout responsive dilakukan dengan melihat kesesuaian antara hasil pengujian layout sistem apakah sudah berjalan secara responsive pada ukuran layar yang berbeda sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian sistem pada naskah dokumen 4.3.3.2. Berdasarkan hasil pengujian antarmuka layout web responsive terhadap perangkat *smartphone* yang ditunjukan pada Tabel 6.2 dapat disimpulkan bahwa antarmuka Aplikasi Pencarian dan Penawaran Buku dapat berjalan secara renponsive menyesuaikan ukuran layar *smartphone* dengan baik sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian (Lihat Tabel 4.16).

Berdasarkan hasil pengujian layout web responsive terhadap perangkat *tablet* yang ditunjukan pada Tabel 6.3 dapat disimpulkan bahwa antarmuka Aplikasi Pencarian dan Penawaran Buku dapat berjalan secara renponsive menyesuaikan ukuran layar *tablet* dengan baik sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian (Lihat Tabel 4.17).

Berdasarkan hasil pengujian layout web responsive terhadap perangkat desktop (*laptop)* yang ditunjukan pada Tabel 6.4 dapat disimpulkan bahwa antarmuka Aplikasi Pencarian dan Penawaran Buku dapat berjalan secara renponsive menyesuaikan ukuran layar desktop dengan baik sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian (Lihat Tabel 4.18).

Validitas hasil yang didapat pada tahap pengujian layout responsive sebesar 100%. Aplikasi Pencarian dan Penawaran Buku telah dapat menyesuaikan ukuran layar pada perangkat yang berbeda (*smartphone*, *tablet*, desktop) dengan baik sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian sistem yang ada pada naskah dokumen 4.3.3.2

5.2.3 Analisis Pengujian Jarak Lokasi

Analisis pengujian jarak lokasi dilakukan untuk memastikan ketepatan hasil dari perhitungan aplikasi untuk menentukan lokasi terdekat. Pengujian ketepatan jarak yang ditampilkan pada aplikasi bertujuan untuk melakukan

pencarian buku yang menampilkan jarak lokasi sehingga pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya. Sesuai rancangan aplikasi, persamaan Haversine Formula mampu menentukan lokasi terdekat dengan posisi user dan memiliki akuransi kesalahan hanya mencapai 0.5%,

Proses analisis terhadap pengujian jarak dilakukan dengan membandingkan hasil dari perhitungan jarak yang ditampilkan aplikasi menggunakan persamaan Haversine Formula dengan jarak sebernarnya yang didapat dari pengukuran jarak sesuai jalur yang ditampilkan pada Google Maps menggunakan *speedometer*.

Berdasarkan hasil pengujian jarak pada tabel, dapat disimpulkan bahwa persamaan Haversine Formula mampu menentukan lokasi terdekat dengan perbandingan jarak aplikasi dengan jarak sebenarnya memiliki kesalahan rata-rata 0,47 kilometer atau 470 meter dengan presentase akuransi kesalahan masing-masing kasus uji masih kurang dari 0.5%. Hal ini dapat disebabkan oleh berbagai hal seperti kondisi sinyal provider (GPS, *WiFi*, *Cell ID)* yang tidak stabil di lokasi, sehingga membuat pembacaan lokasi melalui perangkat menjadi tidak akurat dan berbeda dengan kondisi sebenarnya dilapangan. Dari hasil analisis pengujian jarak terhadap fungsi aplikasi, persamaan Haversine Formula mampu merepresentasikan lokasi terdekat berdasarkan jarak lokasi sehingga pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya

5.2.4 Analisis Pengujian *Compatibility*

Proses analisis terhadap pengujian kompatibilitas sistem dilakukan dengan melihat kesesuaian antara hasil pengujian kompatibitas sistem, apakah aplikasi sudah dapat menjalankan kebutuhan fungsional sistem dengan baik pada browser yang berbeda sesuai dengan hasil yang diharapkan pada tahap perancangan pengujian sistem pada Tabel 4.20. Berdasarkan Tabel 6.8, hasil pengujian kompatibitas sistem terhadap jenis browser yang berbeda dapat disimpulkan bahwa Aplikasi Pencarian dan Penawaran Buku dapat menjalankan fungsional sistem dengan baik pada jenis browser yang berbeda (Chorme, IE, Firefox, Safari dan Android Browser). Validitas hasil yang didapat pada tahap pengujian kompatibitas sistem sebesar 100% sudah memenuhi hasil yang diharapkan pada tahap perancangan pengujian (Lihat Tabel 4.20).

BAB VII
PENUTUP

7.1 Kesimpulan

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

1. Pemanfaatan teknologi *Geolocation* dan Haversine Formula mampu menentukan lokasi terdekat dengan perbandingan jarak aplikasi dengan jarak sebenarnya memiliki kesalahan rata-rata 0,47 kilometer atau 470 meter dan masing-masing kasus uji memiliki akuransi kesalahan kurang dari 0.5% berdasarkan hasil pengujian yang didapat. Penerapan teknologi Geolocation dan Haversine Formula pada aplikasi dapat membantu pengguna melakukan pencarian buku dengan ditampilkan jarak lokasi sehingga pengguna dapat menemukan penawaran buku yang dekat dengan lokasinya. Pada aplikasi Pencarian dan Penawaran Buku juga ditampilkan informasi petunjuk lokasi pada peta menggunakan *Google Direction Service* sehingga pengguna dapat mengetahui jalur yang bisa ditempuh untuk mencapai lokasi tersebut.
2. Penerapan *responsive web design* pada aplikasi pencarian dan penawaran buku dirancang menggunakan teknik grid system, fluid grid dan media queries. Grid system dirancang menggunakan grid ukuran 960 pixel dengan 12 kolom yang berfungsi sebagai panduan dalam mendesain web agar tampak proposinal. *Fluid grid* merupakan teknik *grid* yang berfungsi untuk mengkonversi ukuran satuan pixel grid kedalam bentuk yang lebih fleksibel yaitu persen (%). Sedangkan, media queries memungkinkan untuk menentukan kapan aturan CSS tertentu harus diterapkan dengan kemampuannya yang bisa mengenali ukuran layar perangkat yang digunakan. Implementasi antarmuka aplikasi menggunakan HTML5 dan CSS3.
3. Berdasarkan hasil pengujian *layout responsive*, penerapan teknik *Responsive Web Design* pada Aplikasi Pencarian dan Penawaran Buku dapat beradaptasi secara optimal dalam beberapa jenis perangkat (*smartphone, tablet, desktop*), berupa respon secara otomatis atau penyesuaian antarmuka web terhadap ukuran layar perangkat yang digunakan sehingga memberikan kenyamanan pengguna pada saat mengakses aplikasi tersebut.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi tahap selanjutnya adalah sebagai berikut:

1. Untuk pengembangan tahap selanjutnya, aplikasi ini dapat diimplementasikan ke tahap lebih lanjut dengan menambahkan beberapa fitur baru dan data kota maupun provinsi secara luas sehingga aplikasi Pencarian dan Penawaran Buku ini bisa dipublikasikan ke masyarakat luas.
2. Di dalam pembuatan aplikasi Pencarian dan Penawaran Buku ini masih belum ada fungsi untuk melakukan pengecekan kevalidan dari masukan data pengguna dan data penawaran buku oleh karena itu harus dilakukan tahap pengembangan lebih lanjut dan diperlukan adanya prosedur pemeriksaan kevalidan data pengguna dan data penawaran buku.
3. Dapat dilakukan penyempurnaan desain antarmuka agar semakin menarik dan *user friendly* sehingga memudahkan pengguna saat mengakses aplikasi.

DAFTAR PUSTAKA

Bhawiyuga, A. (2011). Sistem Pelaporan dan Informasi Posisi Kereta Api Berbasis Global Position System Berbasis Android. *Tugas Akhir ITS*, 1-9.

Gardjito. (2005). Kebijakan Pengelolaan Perpustakaan Berbasis Teknologi Informasi (TI). *VII*(2).

Guswid. (2010, September 3). *Perpustakaan Mojokerto*. Retrieved November 10, 2014, from perpuskotamojokerto.org: http://www.perpuskotamojokerto.org/arsip/detilartikel.php?id=12

Longitudestore. (2012). *The Haversine Formula*. Retrieved Desember 17, 2014, from The Longitude Store loves trigonometry!: http://www.longitudestore.com/haversine-formula.html

Marcotte, E. (2010, May 25). *Responsive Web Design*. Retrieved Desember 16, 2014, from Alist Apart: http://alistapart.com/article/responsive-web-design

Pressman, R. S. (2001). *Software Engineering : A Practitioner's* . Singapore: McGraw Hill.

Putra, C. A. (2012, September 27). *Pengantar Google Maps API*. Retrieved Desember 17, 2014, from CandraLab Studio: http://www.candra.web.id/2012/09/27/pengantar-google-maps-api/

Putri, S. A. (2010). *P\eningkatan Minat dan Budaya Baca Masyarakat.* Jakarta: Universitas Indonesia.

Ramadhani, A. (2013, September 13). *Rahasia Responsive Web Design (RWD) dengan HTML5 dan CSS3*. Retrieved Desember 16, 2014, from o-om.com: http://www.o-om.com/2013/09/responsive-web-design-valid-html5-css3.html

Riyadi, S. (2014). *Kupas Tuntas Web Responsive.* Bogor: Gudang Pustaka.

Rizkia, C. (2015, April 27). *Penggunaan Tablet untuk Internet di Indonesia Hanya 1.9 Persen*. Retrieved Juni 4, 2015, from Selullar ID: http://selular.id/news/2015/04/penggunaan-tablet-untuk-internet-di-indonesia-hanya-1-9-persen/

Rossa, & Shalahudin, M. (2013, Desember 19). Rekayasa Perangkat Lunak. Bandung: Informatika.

rumahdot. (2012, April 15). *HTML5 Geolocation*. Retrieved Desember 16, 2014, from rumahdot.com: http://rumahdot.com/web-design/html5-geolocation/

Syachbana, & Akib, Z. (2013). Perancangan Website Menggunakan Responsive Web Design. *Jurnal Sigmata*, 23.

w3schools. (2011). *Google Maps Basic*. Retrieved Desember 16, 2014, from w3schools.com: http://www.w3schools.com/googleAPI/google_maps_basic.asp

Wibowo, H. (2014). Sistem Informasi Potensi Industri di Kabupaten Bantul Berbasis Geographic Informastion System dan Location Based Service. *Jurnal Script*, 120-129.

Widyatmmoko, N. (2012). Pemanfaatan Geolocation dan Haversine Formulla dalam Perancangan Sistem Informasi Geografis (GIS) (Studi Kasus : Pariwisata Kabupaten Semarang). *T1 - Information Engineering*.

LAMPIRAN

Lampiran 1. Skenario Use Case Kofirmasi Request Masuk

Identifikasi	
Nomor use case	SRS_09
Nama	Konfirmasi Request Masuk
Tujuan	Aplikasi dapat melakukan konfirmasi request dari pengguna lain terhadap buku yang kita tawarkan.
Deskripsi	Use case ini memungkinkan pengguna untuk melakukan konfirmasi request masuk dari buku yang kita tawarkan
Aktor	*Member*
Kondisi awal	Aktor harus masuk halaman login
Kondisi akhir	Aplikasi dapat mengubah status permintaan request dan menyimpanya pada history "permintaan yang disetujui" jika permintaan tersebut disetujui

Skenario utama (*basic flow)*	
Aksi aktor	Reaksi sistem
	1. Menampilkan halaman login
2. Memasukan username dan password	
	3. Mengecek validasi username dan password
	4. Sistem menampilkan halaman dashboard
5. Aktor memilih menu "permintaan"	6. Sistem menampilkan halaman permintaan
7. Aktor memilih menu "request masuk"	8. Sistem menampilkan halaman request masuk
9. Aktor memilih buku yang direquest oleh user lain terhadap buku yang kita ditawarkan	10. Sistem menampilkan data user yang mengirimkan request terhadap buku yang kita tawarkan
11. Aktor menyetujui atau menolak permintaan request	12. Sistem mengubah status permintaan request dan menyimpanya pada history "permintaan yang disetujui" jika permintaan tersebut disetujui

Lampiran 2. Skenario Use Case Mengirimkan Pesan

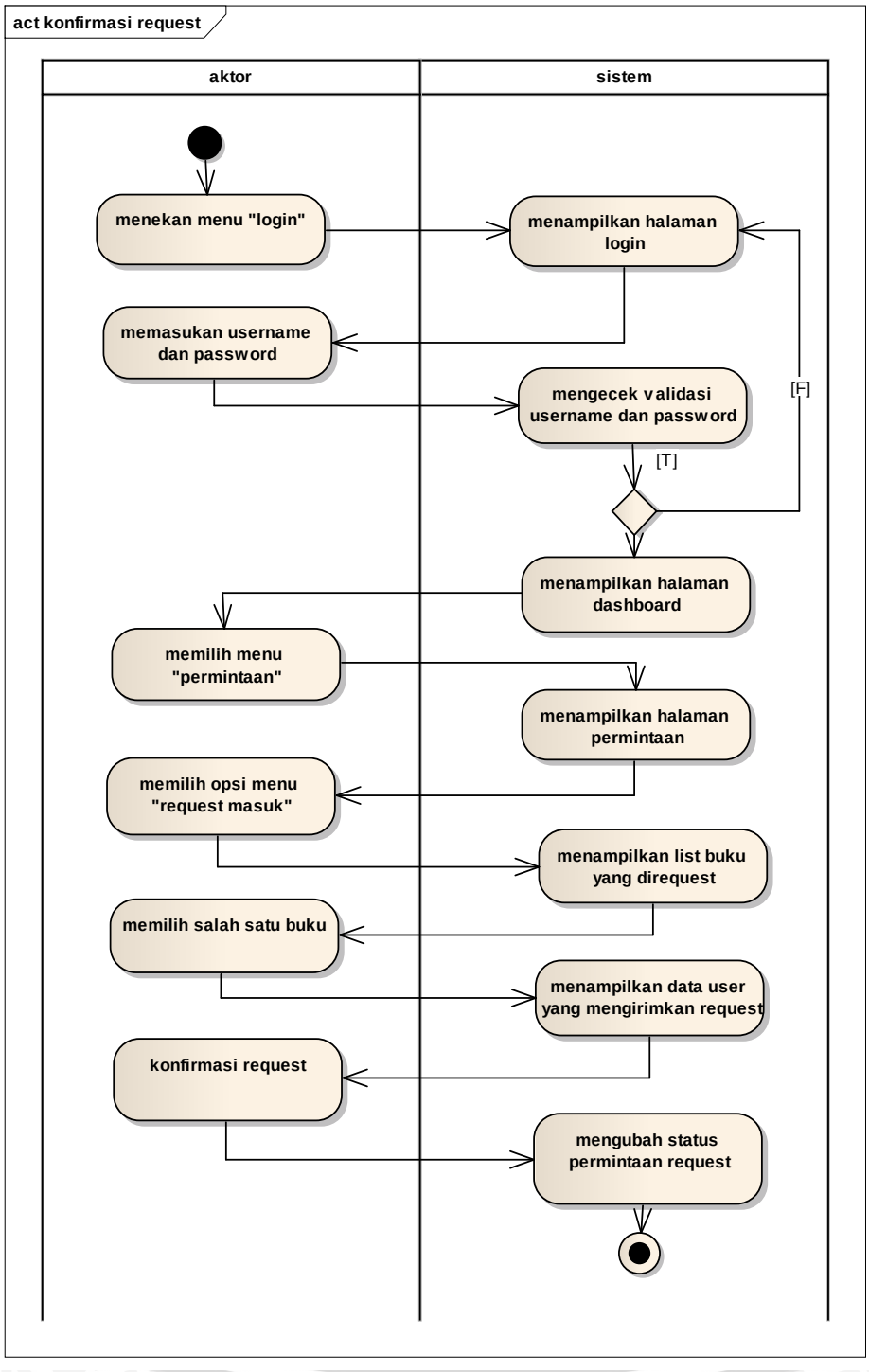
Identifikasi	
Nomor use case	SRS_11
Nama	Mengirimkan Pesan
Tujuan	Aplikasi data melakukan pengiriman pesan untuk mengatur transaksi penjualan atau penyewaan buku
Deskripsi	Use case ini memungkinkan pengguna untuk mengirimkan pesan antar pengguna
Aktor	*Member*
Kondisi awal	Aktor harus mengakses halaman utama sistem
Kondisi akhir	Aplikasi menyimpan data pesan dan mengirimkannya kepada user yang dituju
Skenario utama (*basic flow*)	
Aksi aktor	Reaksi sistem
	1. Menampilkan halaman login
2. Memasukan username dan password	
	2. Mengecek validasi username dan password
	3. Jika username dan password benar, Sistem menampilkan halaman dashboard
4. Aktor memilih menu home	5. Menampilkan halaman utama
6. Aktor memilih salah satu buku	7. Menampilkan informasi detail buku
8. Aktor menekan tombol “kontak”	9. Sistem menampilkan form untuk mengirimkan pesan
10. Aktor memasukan isi pesan dan menekan tombol kirim	11. Sistem menyimpan data pesan dan mengirimkannya kepada user yang dituju
Skenario Alternative	
Aksi aktor	Reaksi sistem
	1. Menampilkan halaman dashboard
2. Aktor memilih menu “permintaan”	3. Sistem menampilkan halaman permintaan
4. Aktor menekan tombol “kontak” pada opsi permintaan	5. Sistem menampilkan form untuk mengirimkan pesan
6. Aktor memasukan isi pesan dan menekan tombol kirim	7. Sistem menyimpan data pesan dan mengirimkannya kepada user yang dituju

Lampiran 3. Skenario use case mengubah status penawaran buku

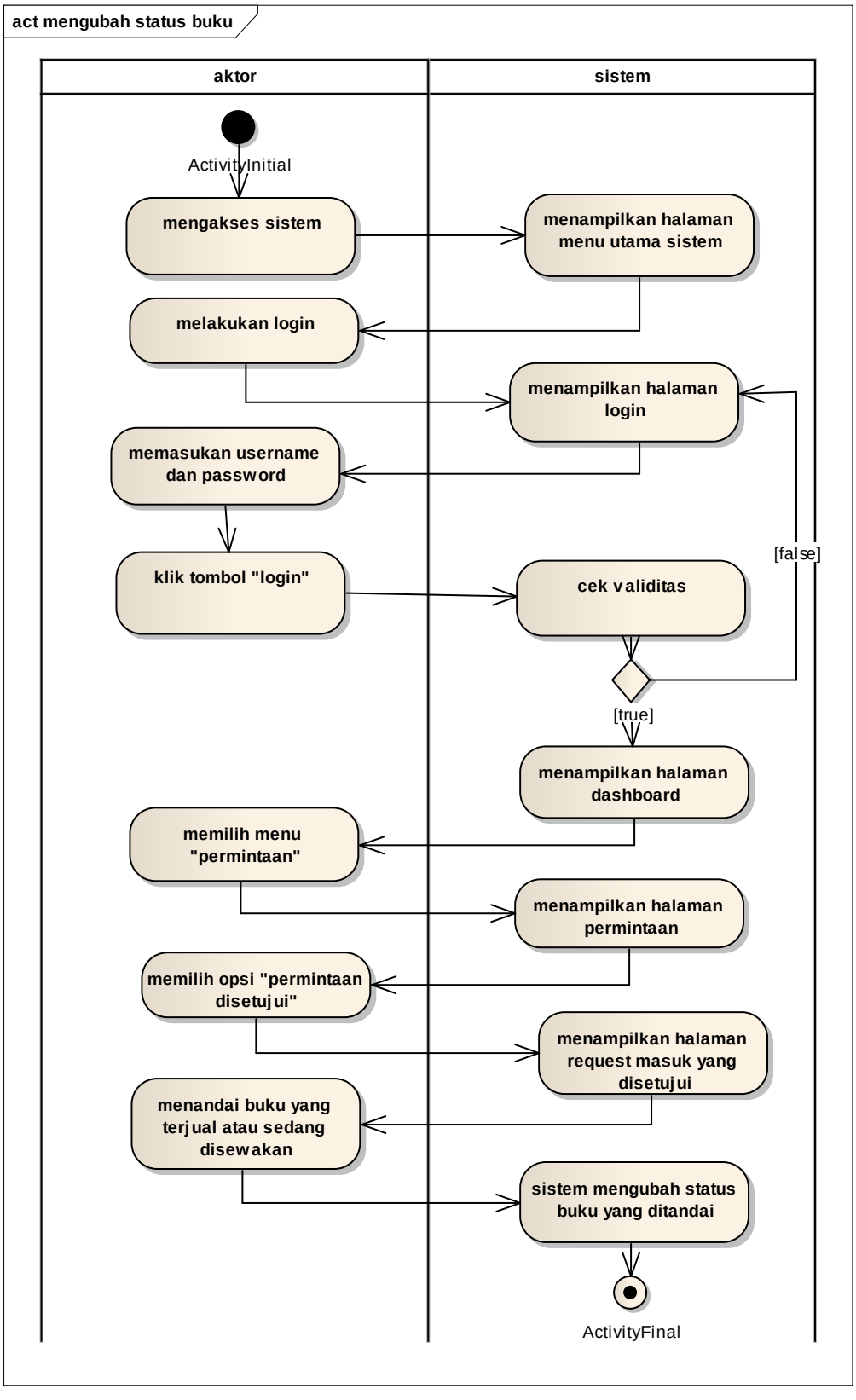
Identifikasi	
Nomor use case	SRS_10
Nama	Request Permintaan Buku
Tujuan	Aplikasi dapat mengubah status penawaran buku jika penawaran buku yang ditawarkan pada sistem sudah laku terjual atau sedang disewakan.
Deskripsi	Use case ini memungkinkan pengguna untuk mengubah status penawaran buku berdasarkan jenis penawaran buku jika penawaran buku yang ditawarkan pada sistem sudah laku terjual atau sedang disewakan.
Aktor	*Member*
Kondisi awal	Aktor harus login dan masuk halaman utama sistem
Kondisi akhir	Aplikasi mengubah status penawaran berdasarkan jenis penawaran buku yang ditawarkan

Skenario utama (*basic flow*)	
Aksi aktor	Reaksi sistem
	1. Menampilkan halaman login
2. Memasukan username dan password	
	3. Mengecek validasi username dan password
	4. Sistem menampilkan halaman dashboard sistem
5. Aktor masuk ke halaman “permintaan yang disetujui”	6. Sistem menampilkan daftar permintaan yang disetujui
7. Aktor manandai buku (“tandai terjual” atau tandai sedang disewakan”) berdarkan jenis penawaran	8. Sitem mengubah status penawaran buku yang ditandai oleh user berdasarkan jenis penawaran

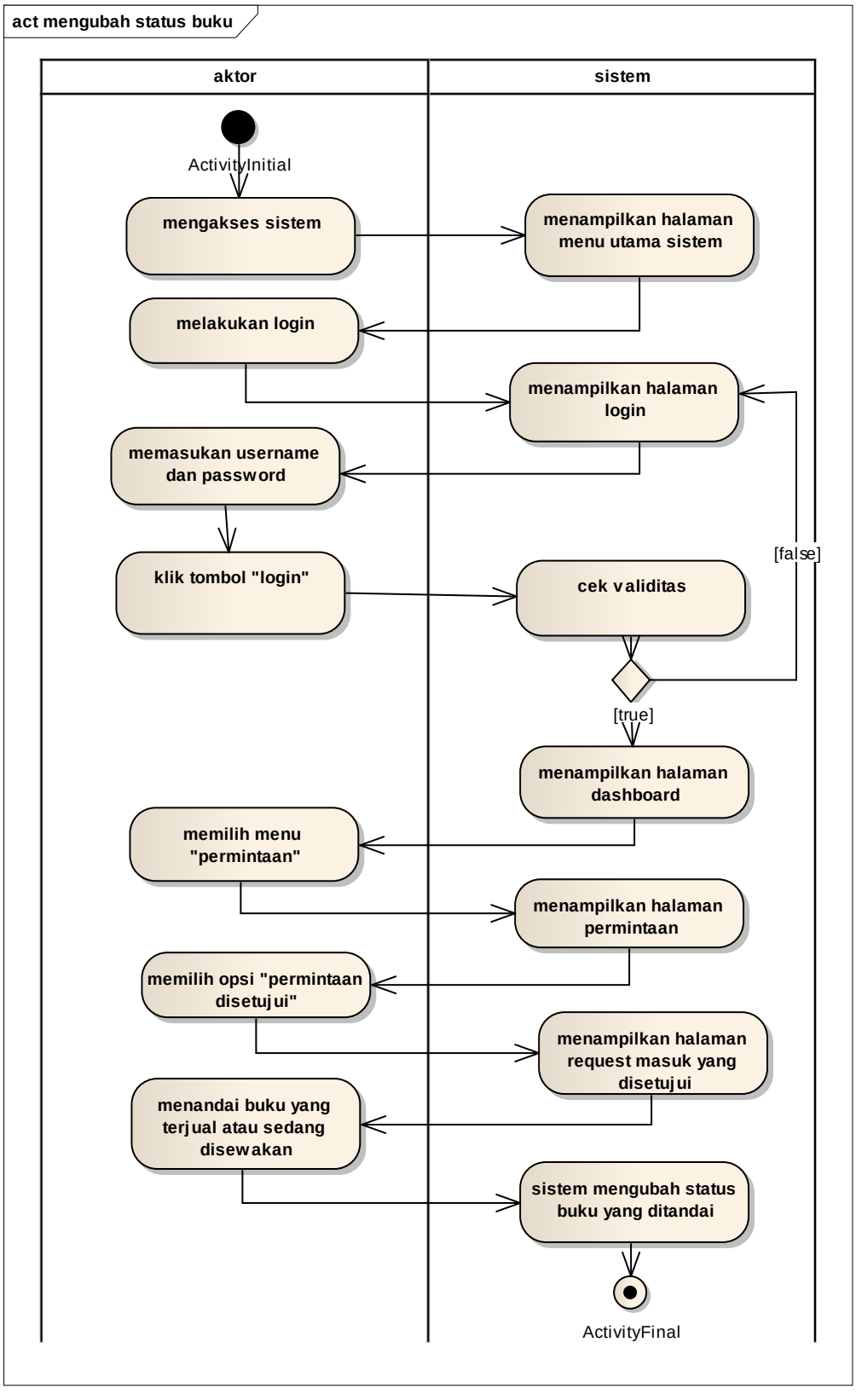
Lampiran 4. Activity Diagram konfirmasi Request Masuk



Lampiran 5. Activity Diagram Mengirimkan Pesan



Lampiran 6. Activity Diagram Mengubah Status Penawaran Buku



Lampiran 7 Kasus Uji Validasi Konfirmasi Request Masuk

Nomor Kasus Uji	UVL_04
Nama Kasus Uji	Kasus Uji Konfirmasi Request Masuk
Objek Uji	Kebutuhan Fungsional SRS_09
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat melakukan konfirmasi terhadap permintaan request masuk dari buku yang kita tawarkan
Data Masukan	Data status request buku (disetujui atau ditolak)
Prosedur Uji	1. Aktor melakukan login dan masuk ke halaman dashboard sistem 2. Aktor memilih menu “permintaan” 3. Sistem menampilkan halaman permintaan 4. Aktor memilih menu “request masuk” 5. Sistem menampilkan halaman daftar request masuk 6. Aktor memilih buku yang direquest oleh user lain terhadap buku yang kita ditawarkan Sistem menampilkan form request 7. Sistem menampilkan data user yang mengirimkan request terhadap buku yang kita tawarkan 8. Aktor menyetujui atau menolak permintaan request 9. Sistem mengubah status permintaan request dan menyimpanya pada history “permintaan yang disetujui” jika permintaan tersebut disetujui
Hasil yang Diharapkan	Status permintaan request berhasil diubah dan menyimpanya pada history “permintaan yang disetujui” jika permintaan tersebut disetujui

Lampiran 8. Kasus Uji Validasi Kirim Pesan

Nomor Kasus Uji	UVL_04
Nama Kasus Uji	Kasus Uji Mengirimkan Pesan
Objek Uji	Kebutuhan Fungsional SRS_11
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat melakukan melakukan pengiriman pesan kepada user yang dituju
Data Masukan	Data penawaran buku dan isi pesan (judul : koala kumal, nama pemilik: santika, isi pesan: terimakasih)
Prosedur Uji	1. Aktor melakukan login dan masuk ke halaman dashboard sistem 2. Aktor memilih menu “home” 3. Sistem menampilkan halaman utama 4. Aktor memilih salah satu buku yang berjudul “koala kumal” 5. Sistem menampilkan detail informasi penawaran buku 6. Aktor menekan tombol “contact” 7. Sistem menampilkan form untuk mengirimkan pesan 8. Aktor memasukan isi pesan dan menekan tombol kirim 9. Aktor menyetujui atau menolak permintaan request 10. Sistem menyimpan data pesan dan mengirimkannya kepada user pemilik buku tersebut
Hasil yang Diharapkan	Data pesan berhasil dikirimkan kepada user pemilik buku yang dipilih sesuai dengan masukan user.

Lampiran 9. Kasus Uji Validasi Mengubah Status Penawaran Buku

Nomor Kasus Uji	UVL_06
Nama Kasus Uji	Kasus Uji Mengubah Status Penawaran Buku
Objek Uji	Kebutuhan Fungsional SRS_10
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat mengubah status penawaran buku jika penawaran buku yang ditawarkan pada sistem sudah laku terjual atau sedang disewakan.
Data Masukan	data penawaran buku yang sudah terjual, sedang disewakan atau sudah dikembalikan sesuai dengan jenis penawaran Studi kasus: 1. Penawaran buku dengan jenis penawaran "sewa" yang berjudul "koala kumal"yang ditawarkan pada sistem sedang disewakan 2. Penawaran buku dengan jenis penawaran "jual" yang berjudul "perahu kertas sudah laku terjual
Prosedur Uji	1. Aktor melakukan login dan masuk ke dashboard aplikasi 2. Aktor memilih menu "permintaan" lalu menekan opsi "permintaan yang disetujui" 3. Sistem akan menampilkan daftar permintaan buku yang disetujui oleh aktor tersebut. 4. Aktor menandai buku "sedang disewakan" pada buku yang berjudul "koala kumal" serta aktor dapat mengubah kembali status penawaran buku "sudah kembali" jika penyewaan buku sudah yang berjudul "koala kumal" sudah dikembalikan.
Hasil yang Diharapkan	1. Aplikasi dapat mengubah status penawaran buku berjudul "koala kumal" yang ditandai oleh aktor "sedang disewakan" maka status penawaran buku berubah menjadi "sedang disewakan" berserta waktu peminjaman serta dapat mengubah statusnya kembali jika buku sudah dikembalikan. 2. Aplikasi dapat mengubah status buku yang berjudul "perahu kertas" yang ditandai aktor menjadi tidak aktif pada jenis penawaran "jual" dan penawaran buku tersebut tidak lagi ditampilkan pada sistem.

UNIVERSITAS BRAWIJAYA