

Aplikasi Pencarian Guru Privat Berbasis Web

Rahmi Putri¹, Mulyono², Ratna Widyati³

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Jakarta, Rawamangun, Jakarta Timur, 13220

rahmiputri1306@gmail.com¹, mulyono@fmipa.unj.ac.id², rwidtyati@fmipa.unj.ac.id³

Abstrak—Guru privat dibutuhkan untuk sebagian siswa yang ingin lebih memahami materi pelajaran tertentu melalui bimbingan. Skripsi ini bertujuan untuk mengembangkan aplikasi pencarian guru privat berbasis web agar memudahkan siswa dalam mencari guru privat. Pengembangan aplikasi ini menggunakan metode pengembangan perangkat lunak model *Waterfall* yang terdiri dari lima tahapan yaitu tahap *Requirement Analysis*, *System Design*, *Implementation*, *Testing*, dan *Maintenance*. Pengembangan aplikasi pencarian guru privat ini menggunakan *framework CodeIgniter* dengan mengimplementasi arsitektur *Model*, *View*, dan *Controller*. Aplikasi pencarian guru privat menjadi salah satu cara untuk memudahkan siswa dalam mencari dan menjadwalkan privat dengan lebih efisien karena siswa tidak harus ke tempat bimbel. Pengguna aplikasi pencarian guru privat dibagi menjadi tiga yaitu Admin, Guru, dan Siswa. Siswa dapat membuat akun, melakukan penjadwalan privat, serta melakukan absen untuk pencatatan administrasi. Guru dapat menerima atau menolak penjadwalan atau pesanan berdasarkan mata pelajaran yang diampu. Admin dapat melihat rekap mengajar dan rekap honor guru per bulan.

Keywords—Guru Privat; Pencarian Guru Privat; Model *Waterfall*; *CodeIgniter Framework*.

I. PENDAHULUAN

Pengguna internet di Indonesia semakin bertambah dari 88 juta penduduk pada tahun 2014^[1] menjadi 132,7 juta penduduk pada Maret 2017^[2]. Jumlah pengguna tersebut lebih dari setengah dari total jumlah penduduk Indonesia yang diperkirakan 50,4% dari 263.510.146 penduduk pada Maret 2017. Dari data tersebut, Indonesia mendapat peringkat ketiga terbanyak untuk pengguna internet terbanyak di Asia setelah China dan India^[2].

Perangkat yang digunakan oleh pengguna internet adalah *smartphone* dan komputer. Jumlah pengguna internet di Indonesia yang menggunakan *smartphone* saja yaitu sekitar 63,1 juta, hanya menggunakan komputer sekitar 2,2 juta, dan menggunakan kedua perangkat tersebut sekitar 67,2 juta pengguna. Konten yang sering diakses pada saat menggunakan internet adalah media sosial, hiburan, dan berita dengan jumlah masing-masing pengguna sebanyak 129,2 juta, 128,4 juta dan 127,9 juta pengguna internet^[3].

Dengan adanya internet, siswa dapat mencari materi yang kurang dimengerti melalui *smartphone* maupun perangkat komputer. Materi yang telah didapatkan melalui internet memungkinkan sebagian siswa masih belum bisa memahami materi tertentu. Oleh karena itu, siswa yang tidak dapat belajar secara mandiri membutuhkan bimbingan dari orang lain seperti teman sebaya atau guru. Bagi siswa yang kurang memahami materi pelajaran yang dipelajari secara mandiri, siswa tersebut membutuhkan bimbingan di luar kegiatan belajar mengajar di kelas

seperti belajar dengan teman sebaya dengan membentuk sebuah kelompok belajar, atau melalui bimbingan belajar (bimbel) maupun privat. Bimbel maupun privat dapat membantu siswa untuk memenuhi kebutuhan belajar, sebagai contoh dalam memberikan latihan soal yang bervariasi tingkat kesulitannya agar siswa dapat lebih memahami materi pelajaran^[4].

Guru privat dipilih oleh siswa karena suasana belajar yang tenang serta siswa dapat leluasa bertanya kepada guru privat jika siswa malu bertanya pada saat kegiatan belajar mengajar di kelas^[5]. Untuk pencarian guru privat, biasanya siswa atau orang tua siswa ke tempat bimbel yang juga menyediakan jasa guru privat, mencari informasi ke mahasiswa bidang studi terkait, atau bahkan langsung ke kampus untuk mencari informasi mengenai mahasiswa yang telah berpengalaman sebagai guru privat untuk mata pelajaran tertentu.

Untuk lebih mengefisienkan dan memudahkan siswa serta orang tua siswa dalam mencari guru privat, penulis mengusulkan ide untuk membuat sebuah aplikasi pencarian guru privat. Secara umum, aplikasi terbagi dalam dua jenis, yaitu aplikasi *mobile* dan aplikasi *website*. Penulis memilih aplikasi *website* dikarenakan lebih mudah diakses sehingga pengguna tidak diharuskan memasang aplikasi tersebut ke perangkat yang dimiliki. Selain itu juga memudahkan Admin untuk mengatur atau melihat aplikasi jika menggunakan aplikasi *web*. Namun, untuk pengembangan selanjutnya, penulis menyarankan untuk membangun aplikasi *mobile*.

Pengembangan aplikasi guru privat ini menggunakan metode pengembangan Model *Waterfall* karena aplikasi memiliki tujuan dan solusi yang jelas, analisis kebutuhan dapat dinyatakan secara jelas, serta kemajuan pengembangan aplikasi dapat diukur. Selanjutnya, penulis memilih *CodeIgniter* sebagai kerangka pengembangan aplikasi berbasis web dengan menggunakan bahasa PHP. Alasan penulis memilih *CodeIgniter* adalah *framework* yang bersifat *free* dan *open source*, performa atau kinerja *framework* yang cepat, dapat diperluas sesuai kebutuhan, serta memiliki dokumentasi yang baik.

Dalam penelitian ini, penulis akan mengembangkan sebuah aplikasi pencarian guru privat berbasis web untuk memudahkan siswa mencari guru privat sesuai dengan kebutuhan seperti dapat memilih mata pelajaran, jadwal belajar, berapa lama waktu belajar, berapa banyak pertemuan dalam satu pesanan, dan menentukan berapa banyak siswa yang ikut dalam belajar.

Aplikasi ini dibatasi hanya untuk mata pelajaran Matematika dan IPA untuk siswa SD kelas I sampai dengan kelas VI serta siswa SMP kelas VII, VIII, dan IX. Mata pelajaran Matematika, Fisika, Kimia, dan Biologi untuk siswa tingkat SMA kelas X, XI, dan XII. Dalam setiap pesanan, siswa hanya bisa menjadwalkan satu mata pelajaran, satu pertemuan dalam satu minggu, empat

sampai delapan pertemuan. Lokasi siswa hanya untuk siswa yang berdomisili di Jakarta Timur. Penulis berkerja sama dengan bimbel Mafika sebagai bentuk dukungan dalam penyedia tenaga kerja guru serta aplikasi ini akan digunakan untuk media pencarian guru bagi bimbel Mafika.

Adapun rumusan masalah berdasarkan pada latar belakang yang sudah dipaparkan adalah bagaimana desain tampilan halaman *website* dan cara mengimplementasikan aplikasi pencarian guru privat? Tujuan penelitian aplikasi pencarian guru privat berbasis web pada skripsi ini adalah untuk mengetahui bagaimana rancangan tampilan serta untuk mengetahui cara mengembangkan aplikasi pencarian guru privat berbasis *website* dengan menggunakan *framework CodeIgniter*.

Penelitian ini diharapkan dapat bermanfaat bagi penulis dapat memahami dan mengembangkan aplikasi berbasis *web* mulai dari analisis kebutuhan untuk *website* tersebut hingga membuat aplikasi pencarian guru privat. Penelitian ini diharapkan dapat memberikan informasi mengenai cara pengembangan aplikasi berbasis *web* kepada mahasiswa Program Studi Ilmu Komputer Universitas Negeri Jakarta. Penelitian ini diharapkan dapat memudahkan siswa atau pun orang tua siswa untuk mencari guru privat sesuai dengan kebutuhan. Tidak hanya itu, aplikasi ini juga membuka lapangan kerja untuk pendidik.

II. KAJIAN PUSTAKA

A. Sistem Informasi

Sistem informasi dapat didefinisikan sebagai seperangkat komponen yang saling terkait dengan mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan pengendalian dalam sebuah organisasi. Selain untuk mendukung pengambilan keputusan, koordinasi, dan pengendalian, sistem informasi juga dapat membantu manajer dan tim pengembang menganalisis masalah, memvisualisasikan subjek yang kompleks, dan menciptakan produk baru [6].

B. System Develop Life Cycle (SDLC)

System Develop Life Cycle adalah metodologi untuk merancang, membangun, dan memelihara informasi dan sistem industri [7]. Salah satu metodologi pengembangan sistem belum tentu sesuai untuk digunakan oleh semua proyek. Masing-masing metodologi yang tersedia paling sesuai untuk jenis proyek tertentu, berdasarkan pertimbangan teknis, organisasi, proyek dan tim pengembang sistem [8]. SDLC digunakan pada beberapa bidang teknik dan industri, seperti rekayasa sistem, rekayasa perangkat lunak, teknik mesin, ilmu komputer, ilmu komputasi, dan teknik terapan.

1. Waterfall Model

Model *Waterfall* adalah proses pengembangan perangkat lunak sekuensial atau berurutan yang dimulai dari spesifikasi persyaratan dan kemajuan dari pihak *client* [9]. Model *Waterfall* terdiri dari lima tahapan yaitu: *Requirement Analysis* (Analisis Kebutuhan), *System Design* (Desain Sistem), *Implementation*, *Testing* (Uji Coba), dan *Maintenance* (Pemeliharaan).

a. Requirement Analysis

Requirement Analysis adalah deskripsi mengenai persyaratan kebutuhan sistem mengenai perilaku perangkat lunak yang akan dikembangkan. *Requirement* ini didefinisikan dan didokumentasikan dalam bentuk *Use Case Diagram* yang dapat menggambarkan interaksi antara pengguna dengan perangkat lunak.

b. System Design

Spesifikasi kebutuhan dari tahap pertama dipelajari pada tahap ini dan tim pengembang membuat sistem desain. Desain sistem membantu menentukan persyaratan perangkat keras maupun sistem dan juga membantu dalam mendefinisikan keseluruhan arsitektur sistem.

c. Implementation

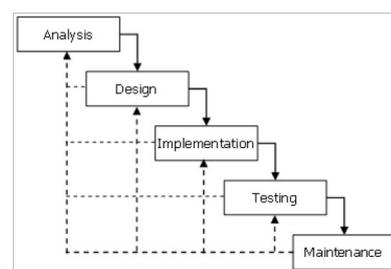
Dengan masukan dari tahapan sebelumnya yaitu perancangan sistem, sistem ini pertama kali dikembangkan pada program kecil yang disebut unit, yang akan terintegrasi pada tahap berikutnya. Setiap unit dikembangkan dan diuji untuk fungsinya yang disebut *Unit Testing*.

d. Integration and Testing

Semua unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem dan melakukan pengujian masing-masing unit. Integrasi posting seluruh sistem diuji untuk setiap kesalahan dan kegagalan.

e. Maintenance

Pada tahap ini, pengembang memastikan sistem berjalan dengan baik. Ada beberapa masalah yang muncul di lingkungan *client*. Dalam tahap ini, tim pengembang dapat memperbaiki masalah tersebut. Selain itu, tim pengembang melakukan *maintenance* untuk meningkatkan sistem. Pemeliharaan dilakukan untuk menyampaikan perubahan lingkungan pelanggan.



Gambar 2.1: Waterfall Model

C. Unified Modelling Language (UML)

Unified Modeling Language (UML) berfungsi untuk memodelkan aplikasi komputer yang menjadi bahasa pemodelan standar dalam pengembangan suatu sistem. Diagram UML juga dapat mempermudah pengembang lain untuk bergabung dalam suatu proyek agar alur pengembangan proyek tersebut mudah dimengerti. Standarisasi ini dapat mengurangi kesalahpahaman bagi tim pengembang [10]. Diagram UML yang akan dibahas yaitu *Use Case Diagram*, *Class Diagram*, serta *Activity Diagram*.

1. Use Case Diagram

Use Case Diagram bertugas untuk menggambarkan satu unit fungsi yang diberikan oleh sistem dengan tujuan

utamanya adalah untuk membantu tim pengembang memvisualisasikan kebutuhan fungsionalitas suatu sistem. *Use Case Diagram* juga menjelaskan hubungan antara aktor (pengguna) dan sistem untuk mencapai suatu tujuan. Diagram *Use Case* biasanya digunakan untuk mengkomunikasikan fungsi tingkat tinggi sistem dan cakupan sistem [11].

2. Class Diagram

Class Diagram menunjukkan relasi antar entitas yang berbeda, baik orang benda maupun data. Diagram ini dapat digunakan untuk menampilkan kelas logis dan kelas implementasi. *Class Diagram* merupakan bagian utama pada pemodelan berorientasi objek dan juga memiliki struktur statis dari aplikasi komputer. Diagram ini menunjukkan kelas sistem, atribut, operasi, dan hubungan antar kelas [12].

3. Activity Diagram

Diagram ini menunjukkan aliran prosedural kontrol antara dua atau lebih objek kelas saat melakukan suatu proses aktivitas. *Activity Diagram* dapat digunakan pada model proses bisnis tingkat tinggi maupun model tindakan internal kelas tingkat rendah [11]. *Activity Diagram* adalah representasi grafis alur kerja dari aktivitas dan tindakan bertahap dengan dukungan untuk pilihan, iterasi dan konkurensi. *Activity Diagram* dapat digunakan untuk alur kerja dan operasional langkah demi langkah komponen dalam suatu sistem. Diagram ini juga menunjukkan keseluruhan aliran kontrol sistem [12].

D. Sistem Manajemen Basis Data

Basis data merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya yang diaorganisasikan sesuai struktur tertentu dan disimpan dengan baik. Untuk mendapatkan informasi yang berguna dari kumpulan data maka diperlukan suatu perangkat lunak (*software*) untuk memanipulasi data sehingga mendapatkan informasi yang berguna. *Database Management System* (DBMS) merupakan software yang digunakan untuk membangun sebuah sistem basis data yang berbasis komputerisasi. DBMS membantu dalam pemeliharaan dan pengolahan kumpulan data dalam jumlah besar. Sehingga dengan menggunakan DBMS tidak menimbulkan kekacauan dan dapat digunakan oleh pengguna sesuai dengan kebutuhan [13].

E. Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah jenis diagram struktural untuk digunakan dalam desain *database*. ERD berisi simbol dan konektor yang berbeda yang memvisualisasikan dua informasi penting yaitu: entitas (*entity*) utama dalam lingkup sistem, dan hubungan (*relation*) antar entitas [14].

III. IMPLEMENTASI PROGRAM

A. Tahap Penelitian

Pada penelitian ini, penulis membagi tahap-tahap penelitian menjadi tiga bagian, yaitu perencanaan, pelaksanaan, dan penulisan. Berikut adalah penjelasan pada tahapan penelitian:

1. Perencanaan

Sebelum melakukan penelitian, peneliti memilih tema atau topik yang akan dilakukan, merumuskan masalah apa saja yang terkait mengenai topik yang dipilih, melakukan

studi pendahuluan dengan menentukan kerangka teoritis mengenai pemecahan masalah, menentukan sampel atau objek penelitian, menyusun rencana penelitian serta menyusun apa tujuan dan manfaat penelitian.

2. Tahap Pelaksanaan

Setelah melakukan tahap perencanaan, peneliti melakukan pelaksanaan penelitian sesuai dengan model pengembangan perangkat lunak yang dipilih yaitu model *Waterfall*. Tahapan yang dilaksanakan yaitu peneliti menganalisis kebutuhan apa saja yang diperlukan untuk pengembangan sistem, menerjemahkan kebutuhan melalui merancang sistem maupun antarmuka, melakukan implementasi, uji coba sistem, serta pemeliharaan sistem.

3. Tahap Penulisan

Penulis membuat laporan penelitian berdasarkan pedoman penulisan dengan jenis penelitian aplikasi produk.

B. Requirement Analysis

Tahap ini, penulis menganalisis kebutuhan utama apa saja yang harus dipenuhi oleh sistem agar dapat beroperasi secara baik. Pada fase ini, penulis mewawancarai pihak bimbel mafika hal apa saja yang dibutuhkan pada website pencarian guru privat. Seperti yang telah disebutkan pada batasan masalah, guru privat yang dimaksud ialah guru yang telah terdaftar di bimbel mafika. Untuk akun aplikasi web ini, guru privat didaftarkan oleh admin, dan selanjutnya dapat dipakai oleh guru untuk melakukan konfirmasi jika pesanan yang ada.

Aplikasi ini diharapkan dapat melakukan hal-hal berikut:

- Siswa dapat memesan guru privat dengan membuat akun terlebih dahulu.
- Siswa dapat menentukan jadwal belajar dengan minimal lima jam sebelum kegiatan belajar mengajar dimulai.
- Siswa dapat menentukan jumlah siswa yang dapat mengikuti privat maksimal empat orang siswa.
- Siswa dapat memilih metode pembayaran dengan cara *transfer* dan *cash*.
- Siswa dapat mengunggah foto soal yang akan dibahas saat pembelajaran berlangsung.
- Setelah data siswa seperti nama, nomer hp, *e-mail*, alamat, mata pelajaran, kelas, tanggal belajar, waktu belajar, foto soal yang akan dibahas, serta metode pembayaran, pesanan akan dikonfirmasi oleh guru.
- Siswa melakukan pembayaran setelah guru mengkonfirmasi pesanan.
- Setelah siswa membayar total pembayaran yang telah dikalkulasi oleh sistem, siswa dapat mengunggah bukti pembayaran.
- Siswa dapat absen setelah guru datang.
- Siswa dapat mengusulkan ganti hari pada pertemuan tertentu.

C. Desain Sistem

Pada tahap desain sistem, penulis menerjemahkan kebutuhan sistem yang telah didapatkan kedalam bentuk visual berupa *Use Case Diagram*, *Class Diagram*, *Activity*

Diagram, Entity Relationship Diagram serta tampilan antarmuka sistem atau *user interface*.

1. Use Case Diagram

Penulis menggunakan *use case diagram* untuk menggambarkan aktor-aktor yang terdapat dalam sistem beserta peranannya di dalam sistem yang sedang dikembangkan. Pada sistem ini, terbagi menjadi 3 aktor yaitu Siswa, Tutor dan Admin. Hal yang dapat dilakukan Siswa pada sistem ini adalah:

- Siswa dapat membuat akun dengan mengisi biodata pada *form* yang telah disediakan.
- Siswa dapat masuk (*login*) ke sistem apabila telah membuat akun.
- Siswa dapat mengubah biodata profil jika ada perubahan data.
- Siswa dapat memesan guru privat dengan mengisi *form* pesanan.
- Setelah form pesanan diisi dan pesanan juga telah dikonfirmasi dan diambil oleh guru, siswa diharuskan mengkonfirmasi pembayaran dengan mengunggah bukti pembayaran.
- Siswa dapat melihat jadwal pada pesanan yang telah dilakukan oleh siswa.
- Siswa dapat melakukan absen jika guru telah datang ke rumah.
- Siswa dapat mengusulkan ganti jadwal pada pertemuan tertentu.
- Siswa dapat keluar (*logout*) dari akun.

Berikut adalah hal yang dapat dilakukan guru pada aplikasi pencarian guru privat ini:

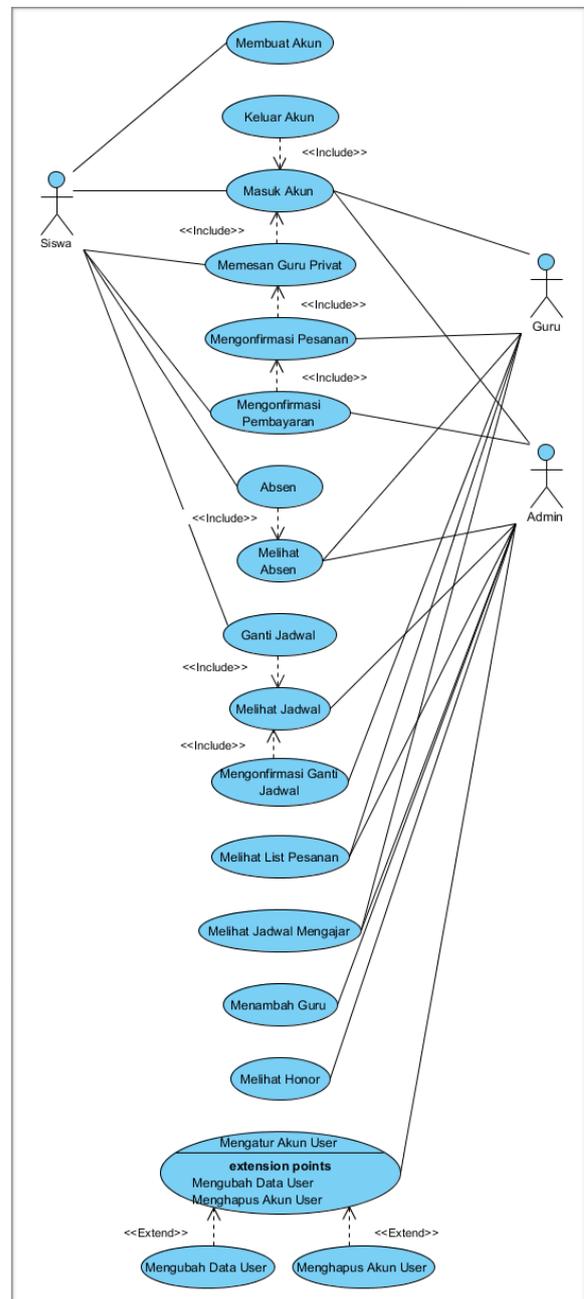
- Guru dapat masuk (*login*) ke akun jika Guru tersebut telah memiliki akun yang ditambahkan oleh Admin.
- Guru dapat mengubah biodata profil jika ada perubahan data.
- Guru dapat melihat *list* pesanan jika ada siswa yang memesan guru privat sesuai dengan mata pelajaran yang diampu oleh guru yang bersangkutan.
- Guru bisa mengkonfirmasi pesanan siswa dengan mengambil pesanan atau tidak mengambil pesanan.
- Guru bisa melihat rekap mengajar guru yang bersangkutan.
- Guru dapat melihat jadwal setiap pesanan yang telah diterima oleh guru tersebut.
- Guru bisa melakukan konfirmasi menerima atau menolak usulan ganti hari jika siswa mengusulkan ganti hari.
- Guru dapat keluar (*logout*) dari akun.

Untuk Admin, hal yang dapat dilakukan pada sistem adalah sebagai berikut:

- Admin dapat masuk (*login*) ke sistem.
- Admin dapat mengubah profil diri jika ada perubahan.

- Admin dapat mengatur akun Guru dengan menambah, mengubah data, serta menghapus akun Guru.
- Admin juga dapat mengatur akun Siswa dengan mengubah data dan menghapus akun Siswa.
- Admin dapat melihat semua pesanan baik yang sudah diterima atau belum dikonfirmasi oleh guru.
- Admin juga dapat memeriksa status pembayaran Siswa apakah siswa tersebut telah melakukan pembayaran atau belum.
- Admin dapat melakukan verifikasi bahwa Siswa telah melakukan pembayaran.
- Admin dapat melihat honor guru berdasarkan absen yang telah dilakukan oleh siswa.
- Admin dapat keluar (*logout*) dari akun.

Berikut adalah *Use Case Diagram* dari aplikasi pencarian guru privat :



Gambar 3.1: Use Case Diagram pada Aplikasi Pencarian Guru Privat

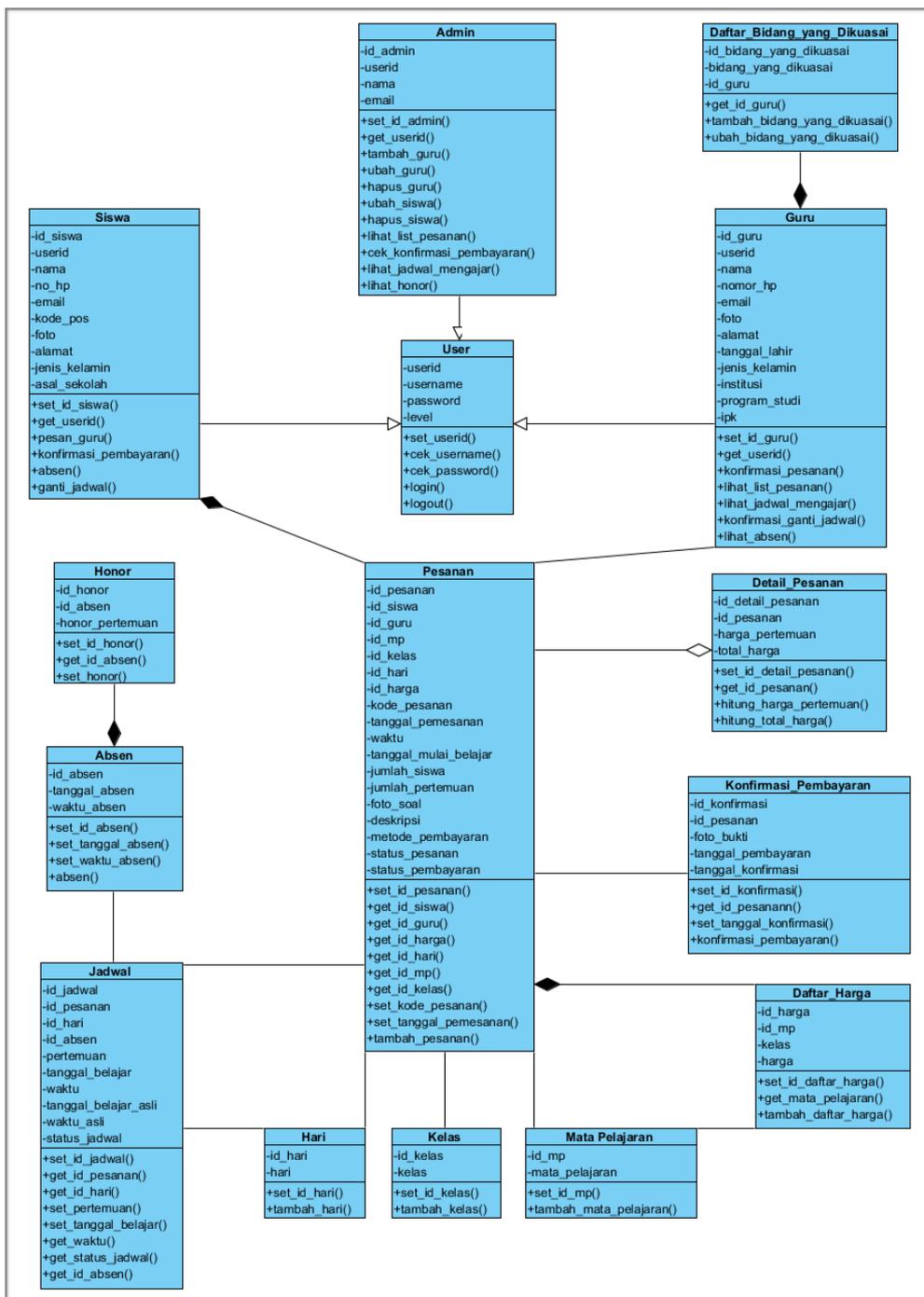
2. Class Diagram

Desain *Class Diagram* pada aplikasi pencarian guru privat ini memiliki sepuluh *Class* seperti yang dapat dilihat pada gambar sebelumnya. *Class* user berfungsi untuk masuk atau *login* ke dalam sistem. *Class* admin, siswa dan guru merupakan biodata diri masing-masing user. *Class* pesanan berfungsi untuk menyimpan data pesanan siswa. *Class* daftar harga untuk *list* harga tiap pertemuan berdasarkan mata pelajaran dan kelas. *Class* detail pesanan untuk menyimpan data harga, jumlah pertemuan dan total harga. *Class* konfirmasi pembayaran untuk menyimpan bukti pembayaran siswa.

3. Activity Diagram

Siswa yang ingin memesan guru privat harus memiliki akun terlebih dahulu. Setelah itu, siswa dapat memesan guru privat dengan mata pelajaran Matematika, Fisika, Kimia, atau Biologi pada hari dan waktu yang ditentukan oleh siswa. Siswa dapat memesan pada hari yang sama namun dengan batasan lima jam sebelumnya. Misalnya siswa memesan pada pukul 12.00 dengan jadwal mulai dari pukul 17.00.

Setelah memilih mata pelajaran, kelas, tanggal, waktu, dan lama belajar, serta mengisi alamat, banyak siswa, jumlah pertemuan, siswa mengajukan pemesanan dan menunggu guru untuk mengkonfirmasi pesanan tersebut. Kemudian, setelah seorang guru mengkonfirmasi bisa



Gambar 3.2: Class Diagram pada Aplikasi Pencarian

Penulis menggunakan ERD untuk menggambarkan bagaimana alur penyimpanan data pada sistem yang dikembangkan. ERD pada penelitian ini memiliki lima belas entitas, yaitu User, Admin, Guru, Siswa, Pesanan, Detail Pesanan, Konfirmasi Pembayaran, Jadwal, Absen, Honor, Hari, Mata Pelajaran, Kelas, Daftar Harga, dan Daftar Bidang yang dikuasai.

D. Tahap Penelitian

Tahap implementasi ini untuk menerjemahkan ke dalam bahasa komputer berdasarkan kebutuhan pada aplikasi pencarian guru privat. Tahapan implementasi yang dilakukan oleh penulis yaitu:

1. Membuat basis data

Basis data dikembangkan menggunakan MySQL berdasarkan ERD yang telah dibuat sebelumnya pada tahap pengembangan desain sistem. Dalam basis data pencarian guru privat ini, terdapat lima belas tabel yaitu tabel Absen, Admin, Daftar Harga, Daftar Bidang yang dikuasai, Detail Pesanan, Guru, Hari, Honor, Jadwal, Kelas, Konfirmasi Pembayaran, Mata Pelajaran, Pesanan, Siswa dan User. Berikut adalah gambar basis data yang dibuat menggunakan aplikasi phpMyAdmin.

Table	Action	Rows	Type	Collation	Size	Overhead
absen	8	InnoDB	latin1_swedish_ci	16 KiB	-	
admin	1	InnoDB	latin1_swedish_ci	32 KiB	-	
daftar_bidang_yang_dikuasai	0	InnoDB	latin1_swedish_ci	48 KiB	-	
daftar_harga	33	InnoDB	latin1_swedish_ci	48 KiB	-	
detail_pesanan	7	InnoDB	latin1_swedish_ci	32 KiB	-	
guru	5	InnoDB	latin1_swedish_ci	64 KiB	-	
hari	7	InnoDB	latin1_swedish_ci	16 KiB	-	
honor	5	InnoDB	latin1_swedish_ci	32 KiB	-	
jadwal	26	InnoDB	latin1_swedish_ci	64 KiB	-	
kelas	15	InnoDB	latin1_swedish_ci	16 KiB	-	
konfirmasi_pembayaran	7	InnoDB	latin1_swedish_ci	32 KiB	-	
mata_pelajaran	5	InnoDB	latin1_swedish_ci	16 KiB	-	
pesanan	7	InnoDB	latin1_swedish_ci	176 KiB	-	
siswa	2	InnoDB	latin1_swedish_ci	32 KiB	-	
user	8	InnoDB	latin1_swedish_ci	32 KiB	-	
15 tables	Sum	136	InnoDB	latin1_swedish_ci	656 KiB	0 B

Gambar 3.6: Basis Data Aplikasi Pencarian Guru Privat

2. Implementasi Desain Tampilan Sistem

Pada pengimplementasian desain tampilan sistem, penulis menggunakan *framework CodeIgniter* dan *Materialize* berdasarkan desain tampilan awal pada tahap desain sistem. Halaman pertama pada aplikasi pencarian guru privat terdapat menu Mafika atau Home untuk menuju ke halaman utama, menu Tentang Kami untuk memberikan gambaran atau profil mengenai Mafika, menu Daftar Akun untuk membuat akun siswa, menu Masuk untuk login ke Akun. Berikut adalah Tampilan Halaman Utama Mafika Privat:

IV. PERCOBAAN DAN HASIL PERCOBAAN

A. Percobaan

Tahap percobaan atau uji coba merupakan tahap keempat setelah analisis kebutuhan, desain sistem, dan implementasi pada metode *Waterfall*. Pada tahap ini, sistem yang dikembangkan diuji apakah sudah berjalan dengan baik serta sesuai dengan kebutuhan.

Adapun langkah-langkah pengujian aplikasi pencarian guru privat adalah sebagai berikut:

- Admin membuat akun guru privat dengan mengisi formulir biodata guru.
- Guru dapat mengakses akun yang telah dibuat oleh admin.
- Siswa membuat akun dengan mengisi biodata diri.
- Siswa melakukan pesan akun dengan mengisi formulir yang telah disediakan.
- Pesanan akan muncul pada halaman guru dan admin.
- Guru mengkonfirmasi pesanan dengan menerima

maupun menolak pesanan.

- Jika guru menerima pesanan, Siswa dapat melihat jadwal dengan mengakses halaman jadwal belajar.
- Siswa dapat mengunggah bukti pembayaran jika siswa memilih metode pembayaran transfer.
- Siswa dapat mengusulkan ganti jadwal jika pesanan telah diterima oleh guru.
- Guru dapat mengkonfirmasi ganti jadwal dengan menerima dan menolak ganti jadwal.
- Siswa dapat melakukan absen pada setiap pertemuan di halaman jadwal belajar.
- Admin dapat mengatur akun guru dan siswa dengan mengubah data, mengubah password.
- Admin juga dapat menghapus akun guru dan siswa.

- Admin dapat melihat *list* pesanan.
- Admin dapat melakukan verifikasi pembayaran siswa yang dapat diakses pada halaman *list* pesanan.
- Admin dapat melihat rekap mengajar guru berdasarkan pesanan yang telah diterima oleh guru.
- Admin dapat melihat total honor guru dan honor per bulan guru.

Pengujian sistem ini dilakukan melalui metode *black box* dengan menguji sistem dari sisi fungsionalitas. Pengujian ini bertujuan untuk mengetahui apakah sistem dapat berfungsi dengan baik. Hal yang diuji pada aplikasi pencarian guru privat adalah:

1. Admin

- Manajemen akun Guru (tambah, edit dan hapus akun).
- Manajemen akun Siswa (edit dan hapus akun).
 - *List* pesanan (lihat pesanan, lihat detail pesanan, lihat status pembayaran).
- *List* jadwal mengajar (lihat jadwal mengajar).
- *List* honor guru (lihat honor guru).

2. Guru

- Manajemen akun (edit).
- *List* pesanan (lihat pesanan, lihat detail pesanan, terima atau tolak pesanan).
- *List* jadwal mengajar (lihat jadwal mengajar).
- *List* jadwal belajar (lihat jadwal, lihat absen, terima atau tolak ganti jadwal).

3. Siswa

- Manajemen akun (buat dan ubah).
- Pesan Guru (tambah).
- *List* riwayat pemesanan (lihat).
- Pembayaran (konfirmasi pembayaran).
- *List* jadwal belajar (lihat, ganti, absen).

B. Hasil Percobaan

Berdasarkan uji coba yang telah dilakukan untuk menguji setiap level akun yaitu user admin, guru, dan siswa, berikut adalah hasil percobaan yang telah dilakukan: Halaman akun admin, admin dapat melakukan manajemen akun guru baik dalam tambah, ubah, serta hapus akun. Admin juga dapat melakukan manajemen akun siswa yaitu untuk ubah dan hapus akun siswa. Admin dapat melihat semua *list* pesanan siswa.

Halaman manajemen akun guru menampilkan data guru seperti username, nama guru, bidang yang dikuasai, no hp. Admin dapat melakukan tambah akun, hapus akun, dan hapus akun guru tertentu. Pada halaman tambah akun, admin memasukkan nama guru, username, password, nomor hp, email, unggah foto guru, alamat, tanggal lahir, jenis kelamin, institusi, program studi, bidang yang dikuasai. Username harus unik dan tipe foto guru adalah jpg atau png. Pada setiap akun guru, data guru dapat diubah namun username tidak dapat diubah. Akun guru dapat dihapus melalui tampilan manajemen akun guru.

Halaman manajemen akun siswa menampilkan semua akun siswa seperti username, nama siswa, dan nomor hp.

Data siswa dapat diubah dengan memilih tombol ubah pada halaman manajemen akun siswa. Data yang dapat diubah yaitu nama siswa, nomor hp, email, alamat, jenis kelamin, asal sekolah. Akun siswa juga dapat dihapus oleh Admin.

Halaman *list* pesanan menampilkan semua pesanan baik pesanan yang belum diterima, yang sudah diterima, maupun yang ditolak oleh guru. Data yang ditampilkan yaitu nama siswa, mata pelajaran, kelas, tanggal pemesanan, hari belajar, waktu belajar, tanggal belajar, nama guru, metode pembayaran, dan status pembayaran. Pada *list* pesanan terdapat tombol konfirmasi pembayaran untuk mengubah status pembayaran yang awalnya sudah konfirmasi menjadi sudah bayar.

Halaman rekap mengajar untuk menampilkan pesanan yang sudah diterima oleh guru dan pembayaran sudah dilakukan oleh siswa. Data yang ditampilkan yaitu nama siswa, mata pelajaran, kelas, hari belajar, nama guru, jumlah pertemuan, dan jumlah kehadiran. Admin dapat melihat absen guru dengan menekan tombol *view* absen jika siswa telah melakukan absen. Data yang dapat dilihat pada halaman *view* absen adalah pertemuan ke, tanggal belajar, mata pelajaran, kelas, materi, dan nama guru.

Admin dapat melihat daftar honor pada halaman rekap honor. Data dikelompokkan berdasarkan nama guru. Daftar honor berdasarkan absen yang telah dilakukan oleh siswa jika guru telah datang ke rumah. Honor guru dapat juga dilihat pada setiap bulannya. Data yang ditampilkan pada tabel honor adalah nama guru, jumlah pertemuan, dan total honor guru.

Tabel 4.1: Komponen Uji Coba Halaman Akun Admin

No.	Menu	Detail	Keterangan
1.	Manajemen Akun Guru	Tambah, Ubah, Hapus	[x] Sesuai
2.	Manajemen Akun Siswa	Ubah, Hapus	[x] Sesuai
3.	<i>List</i> Pesanan	Lihat	[x] Sesuai
4.	Rekap Mengajar	Lihat	[x] Sesuai
5.	Absen	Lihat	[x] Sesuai
6.	Rekap Honor Guru	Lihat	[x] Sesuai

Untuk halaman akun guru, guru dapat melihat *list* pesanan dan *detail* pesanan, serta melakukan konfirmasi pesanan siswa apakah guru terima atau tolak pesanan. Pada halaman *list* pesanan, guru dapat melihat data semua pesanan berdasarkan bidang ilmu yang dikuasai guru. Data yang ditampilkan yaitu nama siswa, mata pelajaran, kelas, tanggal mulai belajar, hari dan waktu belajar, alamat siswa, jumlah siswa, jumlah pertemuan, jumlah honor guru, deskripsi materi yang akan diajarkan, foto soal, serta status pesanan. Pada setiap pesanan terdapat tombol aksi detail pesanan untuk melihat detail pesanan, tombol terima pesanan untuk guru menerima pesanan, dan tombol tolak untuk guru menolak pesanan tersebut. Tombol terima atau tolak tidak bisa berfungsi lagi jika pesanan sudah diterima atau ditolak oleh guru tersebut.

Pada menu rekap mengajar, guru dapat melihat semua pesanan yang diterima oleh guru. Data yang dapat dilihat yaitu nama siswa, mata pelajaran, kelas, tanggal mulai belajar, hari dan, jumlah pertemuan, alamat siswa, jumlah honor guru, deskripsi materi, serta foto soal. Pada halaman rekap mengajar, terdapat tombol lihat jadwal pada setiap pesanan. Halaman jadwal menampilkan data nama siswa, pertemuan ke, hari dan waktu, tanggal belajar, status

jadwal, status absen, dan dua tombol aksi yaitu terima atau tolak untuk setiap ganti jadwal jika siswa mengusulkan ganti jadwal.

Tabel 4.2: Komponen Uji Coba Halaman Akun Guru

No.	Menu	Detail	Keterangan
1.	List Pesanan	Lihat	[x] Sesuai
2.	Detail Pesanan	Lihat	[x] Sesuai
3.	Manajemen Akun Guru	Tambah, Ubah, Hapus	[x] Sesuai
4.	Konfirmasi Pesanan	Terima, Tolak	[x] Sesuai
5.	List Mengajar	Lihat	[x] Sesuai
6.	Absen	Lihat	[x] Sesuai

Untuk *user* siswa, siswa dapat membuat akun, pesan guru privat, lihat *list* pesanan apakah pesanan sudah diterima atau ditolak, lihat detail *detail* pesanan, serta melakukan konfirmasi pembayaran.

Siswa bisa membuat akun pada menu Daftar akun. Siswa mengisi semua data pada formulir pendaftaran. Data yang diisi yaitu username, password, nama, email, nomor hp, kode pos, alamat, jenis kelamin, dan asal sekolah. Username harus unik, tidak ada duplikasi data username, jika username telah dipakai, siswa tidak dapat membuat akun dan memilih username lain.

Untuk pesan guru, siswa memasukkan kelas, mata pelajaran, hari, tanggal mulai belajar, waktu belajar, jumlah siswa, jumlah pertemuan, unggah foto soal, metode pembayaran. Jika siswa telah memesan, siswa dapat melihat pesanan di menu *list* pesanan. Pada *list* pesanan siswa dapat melihat data nama siswa, mata pelajaran, kelas, jumlah siswa, jumlah pertemuan, total pembayaran, status pesanan. Setiap pesanan bisa dilihat dan siswa dapat mengkonfirmasi pembayaran, serta melihat jadwal belajar pada setiap pesanan.

Halaman jadwal terdapat tabel hari dan waktu, tanggal belajar, status jadwal, status kehadiran dan tombol aksi ganti jadwal dan absen. Pada halaman ganti jadwal, siswa memilih tanggal dan waktu untuk mengusulkan kepada guru bahwa pada pertemuan tertentu akan diubah hari belajar. Untuk melakukan absensi, menekan tombol absen pada halaman jadwal belajar setiap kali pertemuan.

Tabel 4.3: Komponen Uji Coba Halaman Akun Siswa

No.	Menu	Detail	Keterangan
1.	Daftar Akun	Tambah	[x] Sesuai
2.	Pesan Guru Privat	Tambah	[x] Sesuai
3.	List Pesanan	Lihat	[x] Sesuai
4.	Detail Pesanan	Lihat	[x] Sesuai
5.	Konfirmasi Pembayaran	Tambah	[x] Sesuai
6.	Riwayat Pemesanan	Lihat	[x] Sesuai
7.	Jadwal Belajar	Lihat, Ubah	[x] Sesuai
8.	Absen	Tambah	[x] Sesuai

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil analisis dan pengujian fungsional aplikasi ini, didapat kesimpulan sebagai berikut:

1. Aplikasi pencarian guru privat dapat menampilkan data sesuai dengan analisis kebutuhan. Tampilan aplikasi ini telah *responsive* atau dapat menyesuaikan tampilan *smartphone* jika diakses melalui *smartphone*.
2. Aplikasi pencarian guru privat diimplementasi sesuai analisis kebutuhan dan dapat berjalan dengan baik berdasarkan hasil percobaan yang telah dilakukan. Siswa dapat melakukan registrasi atau daftar akun, pesan guru, konfirmasi pembayaran, lihat *list* pesanan siswa, melakukan absen jika guru telah datang ke rumah. Guru dapat melihat semua pesanan berdasarkan bidang yang dikuasai oleh guru, dapat menerima atau menolak pesanan, melihat *list* pesanan yang diterima oleh guru. Admin dapat mengatur akun guru baik menambah, mengubah, serta menghapus akun, dapat mengatur akun siswa dalam mengubah atau menghapus akun, melihat semua pesanan, melihat semua pesanan yang diterima.

B. Saran

Adapun saran penulis dalam pengembangan aplikasi pencarian guru untuk penelitian selanjutnya:

1. Pada pengembangan selanjutnya diharapkan dapat menampilkan *list* guru yang sedang tidak mengajar pada jadwal yang ditentukan oleh siswa dan siswa dapat memilih guru yang diinginkan berdasarkan kebutuhan siswa.
2. Berdasarkan saran pada poin sebelumnya, diharapkan pada pengembangan selanjutnya, pengembang dapat menambahkan fitur *rating* yang diberikan siswa untuk guru agar dapat meyakinkan siswa saat memilih guru.
3. Mengembangkan metode pemesanan dengan siswa dapat melakukan pencarian atau penjadwalan lebih dari satu mata pelajaran dalam satu kali transaksi.
4. Menambahkan fitur pembayaran yang telah bekerja sama dengan Bank tertentu sehingga dapat memudahkan siswa untuk konfirmasi pembayaran dan admin dalam hal pemeriksaan pembayaran.

DAFTAR PUSTAKA

1. Widiartanto, Y. H. (2016) "2016, Pengguna Internet di Indonesia Capai 132 Juta." [Online] <http://tekno.kompas.com/read/2016/10/24/15064727/2016.pengguna.internet.di.indonesia.capai.132.juta/>
2. Tim, Internet World Stats. (2017) "Asia Internet Use, Population Data and Facebook Statistics." [Online] <http://www.internetworldstats.com/stats3.htm/>
3. Gracivia, L. (2016) "Statistik Pengguna Internet di Indonesia." [Online] <http://www.cnnindonesia.com/teknologi/20161025185725-188-167920/statistik-pengguna-internet-di-indonesia/>
4. Inggried. (2011) "Bimbingan Belajar, Perlukah?" [Online] <http://edukasi.kompas.com/read/2011/10/07/12390757/bimbingan.belajar.perluah/>
5. Aulia, M. (2011) "Pilih Les Privat atau Bimbingan Belajar." [Online] <http://edukasi.kompas.com/read/2011/05/01/04311459/Pilih.Les.Privat.atau.Bimbingan.Belajar/>
6. Laudon, K. C. *Manajemen Information Systems*. London: Prentice Hall, 2009.
7. Bassil, Y., "A Simulation Model for the Waterfall Software Development Life Cycle," 2012.

8. Department, Health & Human Services USA. "Selecting A Development Approach," 2008.
9. Pressman, R. S., *Software Engineering*. London: Mac Graw Hill, 2010.
10. Williams, L., "An Introduction to the Unified Modeling Language," 2004.
11. Bell, D. (2003) "An Introduction to the Unified Modeling Language." [Online] <https://www.ibm.com/developerworks/rational/library/769.html>
12. Lee, S., "Unified Modeling Language (UML) for Database Systems and Computer Applications," 2012.
13. Connolly, T., *Database Systems: A Practical Approach to Design, Implementation and Management* (6th Edition). London: Pearson, 2014.
14. Tim, Visual Paradigm. "What is What is Entity Relationship Diagram (ERD)?" [Online] <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>