Pemetaan Basis Data Ke Dalam Format JSON Menggunakan Query Builder CodeIgniter Untuk Grafik Dashboard Berbasiskan MVC

by 09021381823126 Putri Pebreisnaini

Submission date: 01-Aug-2022 09:00AM (UTC+0700)
Submission ID: 1877443152
File name: 09021381823126_PutriPebreisnaini_-_Putri_Pebreisnaini.docx (89.79K)
Word count: 7576
Character count: 45313

BAB I

PENDAHULUAN

1.1 Pendahuluan

Pada Bab ini membahas uraian dari penelitian yang memuat penjelasan latar belakang masalah, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan serta kesimpulan secara umum pendahuluan ini.

1.2 Latar Belakang

Camp sholawat annur merupakan suatu komunitas sholawat yang berada didaerah palembang sumatera selatan. Camp sholawat memliki tujuan untuk memberikan pencerahan sebuah metode sederhana dan membentuk komunitas sholawat diberbagai negara, dimana trainingnya dilakukan secara offline dan online, hingga lahir ribuan ahli sholawat baru disetiap waktunya. untuk menjangkau anggota komunitas diluar daerah ataupun di berbagai negara camp sholawat sudah memiliki sarana online yang dapat menjangkau semua orang atau komunitas. Camp sholawat memiliki sebuah *website* yang dimana *website* tersebut dibangun menggunakan php native dan hanya berisikan informasi.

Camp sholawat annur menginginkan sebuah peningkatan website mulai dari pemakaian *framework* menggunakan CI (*CodeIgniter*), menambah isi content yang menampilkan data menggunakan grafik dan memudahkan pertukaran data. Dari permasalahan tersebut dibangun sebuah *website* berbasis php menggunakan *framework codeigniter* berbasis *model view controller* (MVC). Yang di dalamnya memiliki content pemetaan basisdata kedalam format JSON yang nantinya JSON

tersebut bertujuan untuk menampilkan sebuah grafik menggunakan *highchart*. *JavaScript Object Notation* (JSON) format pertukaran data yang ringan sehingga memudahkan komputer membuat, membaca, menulis, dan menerjemahkannya (Buwono, 2019). Dapat mempermudahkan camp sholawat annur dalam melakukan memantau kegiatan komunitas sholawat serta memberikan informasi terhadap masyarakat mengenai apa itu camp sholawat annur. Camp sholawat annur dirancang menggunakan bahasa pemrograman PHP dan MySQL sebagai *database* dengan penerapan arsitektur *Model-View-Controller (*MVC).

Aplikasi camp sholawat annur menggunakan teknik MVC, dimana aplikasi dibuat dengan memisahkan data (*Model*) dengan tampilan (*View*) dan yang mengatur cara bagaimana data diproses (*Controller*) (Very, 2017).

Berdasarkan permintaan dari camp sholawat annur maka dibuat sebuah *project* untuk membangun aplikasi. Dari hasil *project* tersebut dituangkan dalam bentuk skripsi yang berjudul Pemetaan Basisdata ke dalam format JSON menggunakan *Query Builder CodeIgniter* untuk grafik *dashboard* berbasiskan MVC.

1.3 Rumusan Masalah

Perumusan masalah yang ada pada Camp sholawat annur berdasarkan identifikasi masalah yang ditemukan oleh penulis, yaitu:

1. Mengubah data MySQL kedalam format JSON.
2. Bagaimana cara memisahkan *model*, *view*, dan *controller* pada aplikasi.
3. Seberapa ringan pertukaran data dengan JSON.

1.4 Tujuan

Berdasarkan rumusan masalah tersebut project ini bertujuan untuk:

1. Dapat mengubah data dari MySQL menjadi bentuk format JSON.
2. Membuat aplikasi menggunakan *framework CodeIgniter* berbasiskan *model-view-controller* (MVC).
3. Dapat mempermudah pertukaran data JSON pada *platform* manapun.

1.5 Manfaat

Berdasarkan Latar belakang manfaat dari project ini adalah:

1. Memudahkan dalam pembentukan *view* dan pertukaran data karena data sudah diubah menjadi format JSON.
2. Aplikasi menjadi lebih terproteksi karena sudah menggunakan *framework*.
3. Aplikasi menjadi lebih tertata karena sudah dipisahkan *model*, *view*, dan *controller*.

1.6 Batasan Masalah

Batasan masalah dalam penelitian ini sebagai berikut:

1. *Project* ini dilakukan pada komunitas camp sholawat annur palembang.
2. Mengkonversi dari basis data ke format JSON.
3. Sistem ini menggunakan *framework CodeIgniter*.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini mengikuti panduan jurusan Teknik Informatika Universitas Sriwijaya sebagai berikut:

BAB I. PENDAHULUAN

Pada bab ini diuraikan mengenai latar belakang, perumusan masalah, tujuan dan manfaat project, batasan masalah, dan sistematika penulisan.

BAB II. KAJIAN LITERATUR

Pada bab ini akan dibahas dasar-dasar teori yang digunakan dalam project, seperti definisi-definisi pemetaan, JSON, pengembangan *website*, *Codeigniter* serta gambaran umum mengenai konsep RUP.

BAB III. METODOLOGI PENELITIAN

Bab ini akan membahas tahapan yang akan dilaksanakan pada dalam penelitian seperti pengumpulan data, analisis data dan perancangan sistem. Di akhir bab ini berisi perancangan manajemen proyek pada pelaksanaan *project*.

BAB IV. PENGEMBANGAN PERANGKAT LUNAK

Bab ini akan membahas proses-proses pengembangan perangkat lunak yang akan dibangun pada *project* ini. Pada pengembangan perangkat lunak metode yang digunakan adalah RUP (*Rational Unified Process*), ada 4 fase pada metode RUP yaitu, insepsi, elaborasi, kontruksi, dan transisi.

BAB V. HASIL DAN ANALISIS PENELITIAN

Pada bab ini menjelaskan hasil dari implementasi perancangan perangkat lunak yang dilakukan sebelumnya dan hasil analisis kesimpulan yang didapat pada *project* yang telah dibuat.

1

BAB VI. KESIMPULAN DAN SARAN

Pada bab VI ini berisikan kesimpulan dan saran, diharapkan dapat membantu dan meningkatkan *project* selanjutnya.

1.8 Kesimpulan

Kesimpulan dari bab 1 yaitu masalah yang harus diselesaikan dalam *project* ini adalah mengembangkan aplikasi yang akan mempermudah komunitas sholawat dalam melihat informasi-informasi mengenai camp sholawat annur.

BAB II

KAJIAN LITERATUR

2.1 Pendahuluan

Pada bab II akan membahas tentang teori dalam menyelesaikan masalah pada *project* ini seperti definisi, Pemetaan, Basis Data, PHP, MySQL, JSON, *CodeIgniter*, *Query Builder* serta RUP sebagai kerangka kerja pengembangan perangkat lunak. Serta menjelaskan penelitian lain yang relevan dengan penelitian ini.

2.2 Landasan Teori

2.2.1 Pemetaan

Menurut (Thomas, 2020) pemetaan merupakan bentuk visualisasi yang memaparkan suatu data ke dalam suatu grafik dengan klasifikasi.

Sedangkan menurut (Latukolan et al., 2019) pemetaan adalah suatu langkah dimana memetakan atau mengubah sesuatu ke dalam bentuk baru dengan elemen yang sama. Sering disebut dengan *mapping*.

2.2.2 Basis Data

Menurut Adi Nugroho (2011;5) dalam jurnal (Wongso, 2015) basis data merupakan kumpulan dari data-data terstruktur yang berhubungan sehingga mudah disimpan, dimanipulasi serta dipanggil oleh pengguna. Istilah hubungan ialah data mendeskripsikan domain (ranah) tertentu

sehingga memudahkan pengguna untuk mendapatkan jawaban atas pertanyaan yang diajukan ke basis data tersebut.

2.2.2.1 MySQL

MySQL merupakan salah satu tipe *database server* yang sangat populer karena MySQL memakai SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL termasuk tipe RDBMS (*Relational Database Management System*). Pada MySQL, suatu *database* mengandung satu atau beberapa tabel. Tabel terdiri atas beberapa baris dan setiap baris memiliki satu atau beberapa kolom untuk mengelola *database* (Maulana, 2016).

2.2.3 PHP (*Hypertext Processor)*

Menurut (Lutfi, 2017) PHP ialah bahasa yang dirancang secara khusus untuk penggunaan pada *website*. PHP merupakan alat untuk pembuatan halaman *website* dinamis. PHP adalah singkatan dari *Hypertext Preprocesso*r, suatu kepanjangan rekursif.

Sedangkan menurut (Novendri et al., 2019) PHP merupakan bahasa pemrograman untuk membuat *website* yang bersifat *server-side scripting*. PHP dapat dijalankan pada berbagai macam sistem operasi seperti *Windows*, *Linux*, dan *Mac Os*. Selain Apache, dikarenakan PHP memiliki sifat dinamis. PHP mendukung sebagian *website server* lain, seperti Microsoft ISS, Caudium, dan PWS. PHP dapat memanfaatkan *database* untuk menciptakan halaman *website* yang dinamis. Sistem manajemen *database* yang kerap digunakan bersama PHP ialah MySQL.

Dari pernyataan diatas dapat disimpulkan bahwa PHP yakni bahasa pemrograman yang digunakan pada *website* yang disatukan dengan HTML dan merupakan perintah yang dijalankan dari sisi *server* agar web menjadi dinamis.

2.2.4 JSON

JSON (*JavaScript Object Notation*) yakni format pertukaran data yang ringan, sehingga memudahkan komputer membuat, membaca, menulis, dan menerjemahkannya. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman *JavaScript*. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun sehingga dapat digunakan di bahasa pemrograman *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena itu JSON sangat ideal sebagai bahasa pertukaran data (Buwono, 2019).

2.2.5 *CodeIgniter*

CodeIgniter yaitu suatu *framework* untuk web yang dibuat dalam format PHP. Format yang dibuat ini dapat digunakan untuk membuat sistem aplikasi web yang kompleks. *CodeIgniter* dapat mempercepat proses pembuatan web, karena semua *class* dan modul yang dibutuhkan sudah tersedia dan programmer cukup menggunakannya kembali pada aplikasi web yang akan dibuat (Prabowo, 2015).

2.2.5.1 Model-View-Controller (MVC)

MVC ialah suatu teknik pengembangan perangkat lunak dengan memisahkan data (*Model*) dengan tampilan (*View*) dan yang mengatur cara

bagaimana data diproses (*Controller*). MVC diperkenalkan oleh peneliti di Xerox PARAC yang menggunakan bahasa pemrograman *Smaltalk* yang menggunakan karakteristik dari paradigma bahasa pemrograman berorientasi objek. Penerapan arsitektur MVC bertujuan untuk mempermudah pengembangan aplikasi pada masa yang akan datang (Very, 2017).

2.2.6 *Query Builder*

Query builder adalah kelas pembuat kueri metode pembuat kueri yang memungkinkan untuk membuat kueri secara terprogram menggunakan antarmuka berorientasi objek yang lancar. Ini adalah agnostik basis data, dan nama metodenya sangat mudah untuk diingat. Metode kueri cara umum untuk mengirim data SQL mentah ke *database* dan mendapatkan hasil kembali (Ezell, 2016).

2.3 *Rational Unified Process* (RUP)

2.3.1 Definisi Model *Rational Unified Process* (RUP)

RUP merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practices* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak (Andrian et al., 2014)

RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif. Berikut adalah penjelasan untuk setiap fase pada RUP sebagai berikut (Fitria & Widowati, 2017)

1. *Inception*

   Pada tahap ini penulis menentukan ruang lingkup pengembangan sistem dari hasil wawancara dan observasi yang penulis lakukan.

2. *Elaboration*

   Pada tahap ini dari hasil observasi dan wawancara tersebut penulis dapat melakukan identifikasi masalah pada sistem yang dibuat. Didalam elaboration terdapat dua tahapan yaitu :

   a. Analisis

      Terdapat tiga fase dalam tahapan analisis sistem pada alur pengembangan sistem RUP, yaitu: analisis permasalahan, analisis persyaratan, dan analisis keputusan.

   b. Perancangan

      Pada tahap perancangan terdiri dari: perancangan aplikasi, menggunakan diagram UML meliputi *use-case* diagram, perancangan tampilan, dan menggunakan struktur navigasi.

3. *Construction*

   Pada tahap ini menjelaskan bagaimana mengimplementasi dan melakukan uji coba terhadap aplikasi yang telah dibuat. Dalam tahapan implementasi dijelaskan perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk mengimplementasi aplikasi ini. Sedangkan pada

tahapan uji coba dilakukan *testing*. *Testing* diperlukan untuk menjamin kualitas aplikasi yang telah dibuat apakah telah sesuai dengan yang diharapkan.

4. *Transition*

   Pada tahap *transition* penulis membuat panduan penggunaan dari aplikasi yang telah dibuat.

2.4 Kesimpulan

Kesimpulan bab II berisi tentang metode apa yang akan di lakukan untuk membagun sebuah sistem yaitu dengan metode RUP. Metode ini dianggap cocok untuk menyelesaikan proyek yang akan dikerjakan.

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Pada Bab IV berisi penjelasan proses Pengembangan Perangkat Lunak menggunakan metode RUP (*Rational Unifed Process*). Bab ini akan menjelaskan secara rinci proses pengembangan perangkat lunak yang memilikki beberapa tahapan fase yakni fase insepsi, fase elaborasi, fase kontruksi, dan fase transisi.

4.2 Fase Insepsi

Fase insepsi yaitu tahap pertama yang dilakukan dalam proses pengembangan perangkat lunak yang menggunakan metode RUP. Pada tahapan ini peneliti akan melakukan indentifikasi kebutuhan sistem perangkat lunak yang akan dibuat dan membuat pemodelan diagram *usecase*.

4.2.1 Pemodelan Bisnis

Pemetaan Basis Data adalah suatu proses dimana mengubah sesuatu kedalam bentuk baru seperti merepresentasikan suatu data ke dalam bentuk grafik dengan klasifikasi. Proses klasifikasi mengubah data yang ada pada *database* agar menjadi grafik dapat menggunakan format JSON dan *query builder codeigniter* berbasiskan MVC yang akan digunakan pada penelitian ini. *Query builder* dapat digunakan untuk berkomunikasi dengan data yang ada pada *database*, dengan menggunakan *query builder* ini dapat melakukan perintah seperti, memasukkan data, memilih data, mengubah data, dan menghapus data.

Perangkat lunak yang dibangun ini berbasiskan *website*, perangkat lunak yang dibangun ini akan menampilkan keluaran perangkat lunak yaitu, data seluruh total setoran sholawat, data individu setoran perhari sholawat, sebaran dan rerata setoran berdasarkan *gender*.

4.2.2 Kebutuhan Sistem

Kebutuhan sistem terdapat 2 bagian yaitu, kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional merupakan kebutuhan utama pada perangkat lunak yang harus terpenuhi, sedangkan kebutuhan non-fungsional kebutuhan yang hanya tambahan pada perangkat lunak dan tidak harus terpenuhi. Kebutuhan fungsional dapat dilihat pada tabel IV-1 dan untuk kebutuhan non-fungsional dapat dilihat pada tabel IV-2.

Tabel IV-1. Kebutuhan Fungsional

No	Kebutuhan Fungsional
1.	Perangkat lunak dapat menampilkan data seluruh total sholawat anggota camp sholawat annur berbentuk tabel.
2.	Perangkat lunak dapat menampilkan data individu perhari sholawat anggota camp sholawat annur berbentuk grafik.
3.	Perangkat lunak dapat menampilkan data sebaran dan rerata setoran sholawat anggota camp sholawat berdasarkan *gender* berbentuk grafik.
4.	Perangkat lunak dapat menampilkan persentase anggota camp sholawat berbentuk grafik.

Tabel IV-2. Kebutuhan non-fungsional

No	Kebutuhan Non-Fungsional
1.	Perangkat lunak memiliki antarmuka yang mudah dipahami dan digunakan oleh *user*.

4.2.3 Analisis dan Desain

Analisis dan desain salah satu proses penting yang harus ada pada fase insepsi. Analisis dan desain digunakan untuk menganalisis kebutuhan perangkat lunak dan merancang desain *usecase* agar bisa menyelesaikan suatu permasalahan.

4.2.3.1 Analisis Kebutuhan Perangkat Lunak

Berdasarkan uraian pada pemodelan bisnis, penelitian ini akan membuat perangkat lunak yang dapat mempermudah pengguna untuk mengetahui data-data dan informasi tentang para sholawat berbentuk grafik. Agar data-data sholawat bisa ditampilkan berbentuk grafik data tersebut disimpan terlebih dahulu di dalam *database* berbentuk MySQL, setelah itu akan diubah menjadi format *text* yaitu JSON data yang berada di *database* diambil dan diubah menjadi *syntax* JSON *syntax*-nya terdiri dari dua elemen, yaitu *key* dan *value*. Beberapa langkah yang diperlukan untuk membuat perangkat lunak yang sesuai dengan pemodelan bisnis sebagai berikut;

1. Menyiapkan dataset yang akan dikelolah
2. Mengubah dataset MySQL menjadi format JSON
3. Memasukan format JSON kedalam bentuk grafik

4.2.3.2 Analisis Data

Data yang digunakan oleh perangkat lunak yang dibangun adalah data jamaah camp sholawat annur. Data yang telah dikumpulkan dan datanya telah dijadikan database yang bernama cobacamp.

4.2.3.3 Desain Perangkat Lunak

Sub-bab ini akan menjelaskan desain perangkat lunak dalam bentuk diagram *usecase*, tabel definisi aktor, tabel definisi *usecase*, dan skenario *usecase*.

4.2.3.3.1 Diagram *Usecase*

Diagram *usecase* ini digunakan untuk menggambar interaksi aktor dengan perangkat lunak. Aktivitas perangkat lunak yang dilakukan aktor dapat dilihat pada diagram *usecase* pada gambar IV-1.

4.2.3.3.2 Tabel Definisi Aktor

Aktor adalah orang yang dapat berinteraksi dengan perangkat lunak sistem informasi camp sholawat annur ini. Dapat dilihat penjelasan definisi aktor perangkat lunak yang dibuat pada Tabel IV-3.

Tabel IV-3. Definisi Aktor

No	Aktor	Definisi
1.	*User*	*User* adalah seseorang yang dapat berinteraksi secara langsung melihat informasi tentang camp sholawat annur dan dapat mengakses fitur yang ada pada perangkat lunak.

4.2.3.3.3 Definisi *Usecase*

Tabel definisi *usecase* pada sistem perangkat lunak ini dapat di lihat pada Tabel IV-4.

Tabel IV-4. Definisi *Usecase*

No	*Usecase*	Definisi
1.	Melihat data setoran sholawat seluruh jamaah	Menampilkan informasi mengenai total setoran seluruh jamaah camp sholawat annur yang datanya telah diubah menjadi data JSON.
2.	Melihat data progress setoran individu jamaah berbentuk grafik	Menampilkan informasi progress setoran individu dari jamaah camp sholawat annur yang datanya telah diubah menjadi data JSON untuk ditampilkan berbentuk grafik.
3.	Melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik	Menampilkan data sebaran dan rerata setoran sholawat berdasarkan gender jamaah camp sholawat annur yang datanya telah diubah menjadi data JSON untuk ditampilkan berbentuk grafik.

4.2.3.3.4 Skenario *Usecase*

Skenario *usecase* menjelaskan setiap tahapan yang menggambarkan interaksi antara aktor dengan sistem. Skenario *usecase* pada sistem ini terbagi menjadi tiga yaitu skenario melihat data setoran sholawat seluruh jamaah dapat dilihat pada tabel IV-5, skenario melihat data progress setoran individu jamaah berbentuk grafik dapat dilihat pada tabel IV-6, dan skenario melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik dapat dilihat pada tabel IV-7.

Tabel IV-5. Skenario Melihat Data Setoran Sholawat Seluruh Jamaah

No. *Usecase*	001
Nama *Usecase*	Melihat data setoran sholawat seluruh jamaah
Aktor	*User*
Tujuan	Menampilkan informasi jumlah setoran seluruh jamaah sholawat camp sholawat annur
Deskripsi	Menampilkan data setoran sholawat yang dimana datanya telah diubah menjadi data JSON
Kondisi Awal	Sistem keadaan menyala menampilkan beranda *website* yang berisikan menu top setor dan lain-lain
Skenario Normal	
Aktor	**Sistem**
1. Menekan menu “Top Setor”	
	2. Menampilkan data tabel jumlah setoran seluruh jamaah camp sholawat annur

Kondisi Akhir Skenario Normal	Sistem menampilkan seluruh informasi data setoran sholawat jamaah camp sholawat annur berbentuk tabel

Tabel IV-6. Melihat Data Progress Setoran Individu Jamah Berbentuk Grafik

No. *Use Case*	002
Nama *Use Case*	Melihat data progress setoran individu jamaah berbentuk grafik
Aktor	*User*
Tujuan	Menampilkan informasi progress setoran individu jamaah
Deskripsi	Menampilkan data progress setoran jamaah yang dimana datanya telah diubah menjadi format JSON untuk ditampilkan berbentuk grafik
Kondisi Awal	Sistem berada di tampilan data setoran sholawat seluruh jamaah
Skenario Normal	
Aktor	**Sistem**
1. Menekan tombol yang berada di dalam kolom “ID”	

	2. Menampilkan progress setoran individu jamaah berbentuk grafik
Kondisi Akhir Skenario Normal	Sistem menampilkan data progress setoran sholawat individu jamaah camp sholawat annur berbentuk grafik

Tabel IV-7. Melihat Sebaran dan Rerata Setoran Berdasarkan *Gender* Berbentuk Grafik

No. *Use Case*	003
Nama *Use Case*	Melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik
Aktor	*User*
Tujuan	Menampilkan informasi sebaran dan rerata setoran sholawat jamaah camp sholawat annur berdasarkan *gender*-nya
Deskripsi	Menampilkan data sebaran dan rerata setoran sholawat berdasarkan *gender* jamaah yang dimana datanya telah diubah menjadi data JSON untuk ditampilkan berbentuk grafik
Kondisi Awal	Sistem keadaan menyala menampilkan beranda *website* yang berisikan menu top setor dan lain-lain
Skenario Normal	

Aktor	Sistem
1. Menekan menu “dashboard”	
	2. Menampilkan sebaran dan rerata setoran sholawat berdasarkan *gender* berbentuk grafik
Kondisi Akhir Skenario Normal	Sistem menampilkan seluruh data sebaran dan rerata setoran sholawat jamaah camps sholawat annur berdasarkan *gender* berbentuk grafik

4.3 Fase Elaborasi

Fase elaborasi adalah tahapan kedua yang harus dilakukan pada proses pengembangan perangkat lunak menggunakan RUP. Pada fase ini akan memaparkan *database design*, perancangan antarmuka, diagram aktivitas, dan diagram *sequence*.

4.3.1 *Database Design*

Penelitian ini menggunakan *database* ‘cobacamp’ yang memliki empat buah tabel ‘jamaah’, ‘setoran’, ‘komunitas’, dan ‘provinsi’. Data yang digunakan dalam *database* cobacamp adalah data sholawat jamaah camp sholawat annur. Pada tabel jamaah terdapat 6 *fields* yaitu id_jamaah, nama_jamaah, *gender*, id_provinsi, nomor_hp, dan id_komunitas. Pada tabel setoran terdapat 5 *fields* yaitu id_setoran, tanggal_masehi,

tanggal_hijriyah, id_jamaah, dan jumlah. Pada tabel komunitas terdapat 5 *fields* yaitu id_komunitas, nama_komunitas, singkatan_komunitas, kota, dan negara. Pada tabel provinsi terdapat 3 *fields* yaitu id_provinsi, kode_provinsi, dan nama_provinsi. Perancangan *database design* dapat dilihat pada gambar IV-2.

4.3.2 Perancangan Antarmuka

Pada sub-bab ini akan menampilkan rancangan gambar desain antarmuka perangkat lunak yang akan dibangun. Dapat dilihat pada Gambar IV-3, IV-4, IV-5, IV-6, dan IV-7.

Prosedur pengoperasian perangkat lunak pada gambar IV-4 adalah *user* dapat melihat informasi tentang jumlah sholawat jamaah camp sholawat annur, rerata perhari setoran sholawat, dan user dapat melihat progress setoran jamaah per individu dengan menekan tombol yang ada didalam kolom ID. *User* juga dapat melihat informasi setoran sholawat sesuai *gender*, dengan menekan kolom *gender* dan pilih ui *gender* ingin dilihat.

4.3.3 Kebutuhan Sistem

Perangkat lunak yang akan dibangun memerlukan perangkat keras, bahasa pemrograman dan perangkat lunak. Bahasa pemrograman yang digunakan adalah PHP (*Hypertext Preprocessor*), JSON (*JavaScript*

Object Notation) dan MySQL. Perangkat keras yang digunakan pada sistem ini sebagai berikut:

1. *Processor* Intel(R) Core(TM) i3-3217U CPU @1.80GHz
2. RAM 10 GB

Perangkat lunak yang digunakan pada sistem ini sebagai berikut:

1. Sistem Operasi Windows 10 64-bit
2. IDE Sublime
3. XAMPP
4. Microsoft Edge

4.3.4 Diagram Aktivitas

Digram aktivitas adalah menggambarkan aktivitas yang dilakukan oleh aktor. Diagram aktivitas terbagi menjadi tiga yaitu diagram melihat data setoran seluruh jamaah dapat dilihat pada gambar IV-9, diagram melihat data progress setoran individu jamah berbentuk grafik dapat dilihat pada gambar IV-10, dan diagram melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik dapat dilihat pada gambar IV-11.

4.3.5 Diagram *Sequence*

Diagram *sequence* pada perangkat lunak ini terbagi menjadi 3 buah diagram *sequence*. Diagram *sequence* melihat data setoran seluruh jamaah, dapat digambarkan pada gambar IV-12. Pada gambar IV-13, diagram *sequence* melihat data progress setoran individu jamaah berbentuk grafik. Dan pada

gambar IV-14, diagram *sequence* melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik.

4.4 Fase Kontruksi

Fase kontruksi adalah tahap ketiga dalam pengembangan perangkat lunak menggunakan RUP dalam project ini. Pada pembangunan perangkat lunak fase kontruksi ini di analisis dan desain yang telah dilakukan sebelumnya mulai dibangun. Pada fase kontruksi berisikan perancangan diagram kelas, implementasi kelas, implementasi *query* dan implementasi antarmuka perangkat lunak yang akan dibangun.

4.4.1 Diagram Kelas

Pada sub-bab diagram kelas akan mendeskripsikan kelas *method* dan atribut. Diagram kelas perangkat lunak dapat dilihat pada gambar IV-15.

4.4.2 Implementasi Kelas

Pada implementasi kelas ini perangkat lunak akan dibuat menggunakan bahasa pemrograman PHP. Implementasi kelas akan menjelaskan mengenai diagram kelas yang telah dirancang pada tahap sebelumnya. Pada tabel IV-8 akan mendeskripsikan daftar kelas tersebut.

Tabel IV-8. Implementasi Kelas

No.	Nama Kelas	Nama File	Keterangan

1.	*Database*	database.php	Kelas *database* adalah kelas yang berfungsi untuk menghubungkan basis data ke *website*
2.	Setoran	Setoran.php	Kelas Setoran adalah kelas *controller* yang berfungsi menghubungan *model* yang diambil dari *database* yang berproses untuk menjadi data format JSON
3.	*MY_Controller*	MY_Controller.php	Kelas *MY_Controller* adalah kelas yang berasal dari *codeigniter* untuk membedakan antara *controller* dan *model*.
4.	*MY_Model*	MY_Model.php	*MY_Model* adalah kelas yang berasal dari *codeigniter* untuk membedakan antara *controller* dan *model*
5.	P_*Model*	P_Model.php	Kelas P_*Model* adalah kelas *model* yang dimana semua fungsi di *extends* ke kelas *MY_Model*
6.	P_Setoran	P_Setoran.php	Kelas P_Setoran adalah model yang berisikan proses tabel jamaah, tabel komunitas, tabel

			provinsi, dan tabel setoran untuk digunakan oleh *controller* setoran

4.4.3 Implementasi *Query*

Pada implementasi *query* ini peneliti akan menjelaskan *query* apa saja di setiap kelas yang digunakan perangkat lunak. Pada tabel IV-9 akan mendeskripsikan *query* MySQL. Pada tabel IV-10 mendeskripsikan *query codeigniter*.

Tabel IV-9. Implementasi *Query* MySQL

Nama Query	Keterangan
select()	*Query* "select()" yang ada pada kelas P_Setoran digunakan untuk memilih data yang ada pada tabel *database*.
from()	*Query* "from()" yang ada pada kelas P_Setoran yang artinya memilih data dari tabel jamaah yang ada pada *database*.
get()	*Query* "get()" digunakan untuk mengambil suatu data yang dimana data tersebut diambil dari *database*.
where()	*Query* "where()" digunakan untuk memfilter data yang ada pada perintah *select*.

join()	*Query* "join()" yang ada pada kelas P_Setoran digunakan untuk menggabungkan tabel yang ada pada *database*.
group_by()	*Query* "group_by()" yang ada pada kelas P_Setoran digunakan untuk mengelompokkan data dengan kriteria tertentu yang ada di *database*.
order_by()	*Query* "order_by()" yang ada pada kelas P_setoran digunakan untuk mengurutkan data berdasarkan *fields* tertentu. Dan data tersebut secara *default* yang datanya tersusun dari urutan kecil ke besar.
SELECT * FROM	*Query* "SELECT * FROM *$this*->tableName" pada kelas MY_Model digunakan untuk memilih data-data tabel *database* yaitu data *tableName*
count()	*Query* "count(setoran.jumlah),0)" pada kelas P_Setoran digunakan untuk menghitung jumlah (bilangan cacah) pada suatu data.
avg()	*Query* "avg(setoran.jumlah),0)" pada kelas P_Setoran digunakan untuk menghitung nilai rata-rat suatu data.

sum()	*Query* "sum(setoran.jumlah),0)" pada kelas P_Setoran digunakan untuk mendapatkan nilai total penjumlahan data-data yang ada.
get_where()	*Query* "get_where('provinsi')" pada kelas P_Provinsi digunakan untuk memilih data dan memfilternya dari tabel provinsi yang berada di *database*.

Tabel IV-10. Implementasi *Query Codeigniter*

Nama Query	Keterangan
row_array()	*Query* "row_array()" yang berada dikelas *MY_Model* dan kelas P_Provinsi tersebut digunakan untuk mengembalikan data dengan hanya mengirimkan nomor "ID" sebagai digit paramater
result_array()	*Query* "result_array()" yang berada dikelas P_Setoran digunakan untuk mengembalikan hasil kueri sebagai array murni atau array kosong saat tidak ada hasil yang dihasilkan.
query()	"query()" berfungsi untuk mengembalikan objek dari hasil *database* saat kueri bertipe "baca" dijalankan dan juga ketika kueri jenis "tulis"

	dijalankan itu hanya berfungsi untuk mengembalikan *TRUE* atau *FALSE* tergantung pada keberhasilan atau kegagalan saat mengambil data.

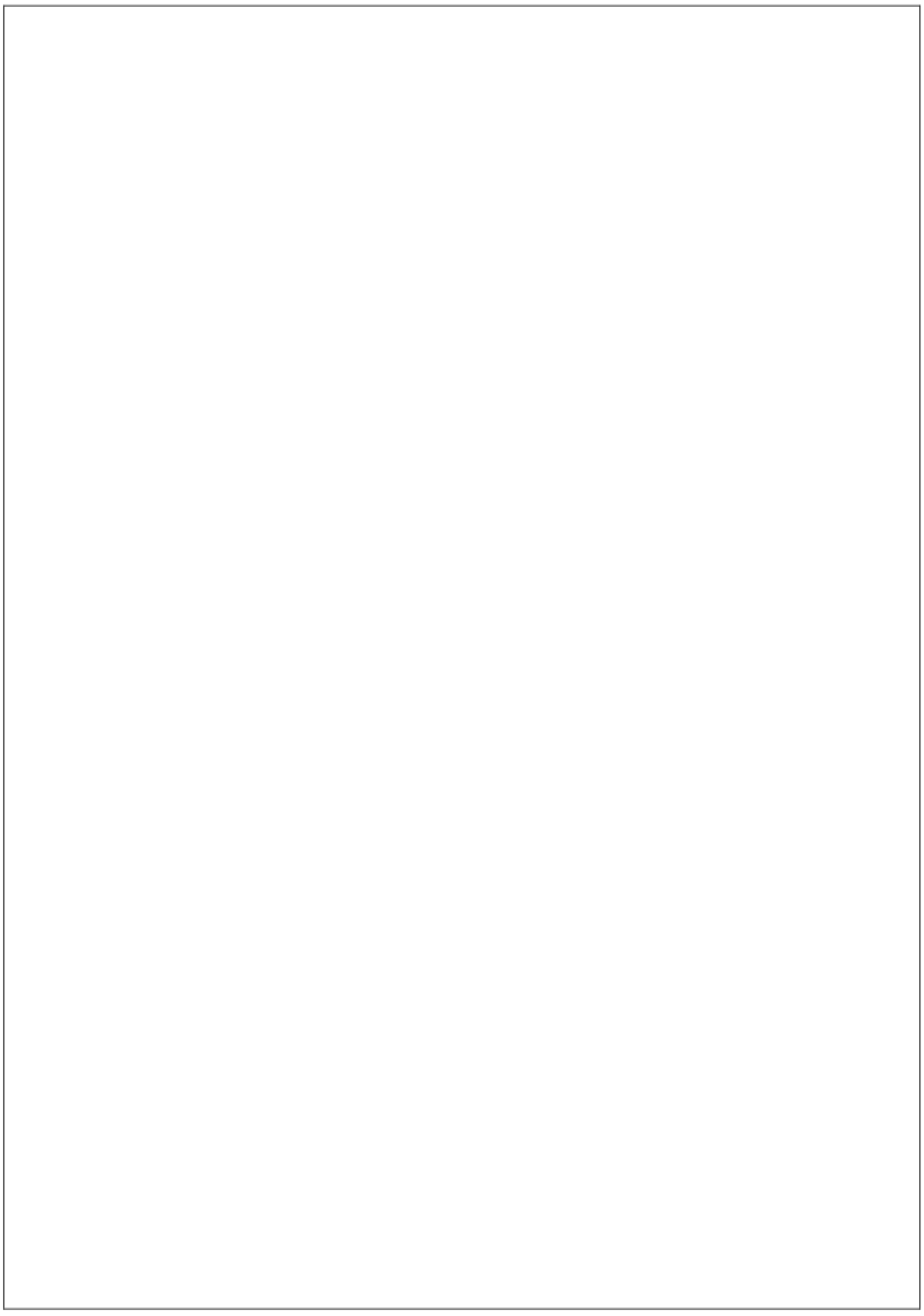
4.4.4 Implementasi Program

Pada implementasi program ini akan menjelaskan langkah-langkah perangkat lunak dari hasil desain sistem yang telah di buat pada tahap sebelumnya. Bahasa pemrograman yang digunakan adalah PHP dan JSON.

Penjelasan program pada tabel IV-11 ketika *user* masuk ke halaman camp sholawat annur pada *website* maka akan langsung muncul beranda *website* camp sholawat annur yang menampilkan data setoran berbentuk tabel dan akan muncul data berbentuk grafik. Dapat dilihat pada tabel IV-11 dijelaskan cara menampilkan halaman *website* yang berisikan data setoran seluruh jamaah dengan menggunakan ajax, pada halaman tersebut akan ditampilkan beranda *website* yang terdapat 3 buah *navbar* atau menu yaitu top setor, *dashboard* dan kontak, saat menekan menu top setor akan menampilkan data setoran seluruh jamaah berbentuk tabel.

Tabel IV-11. Implementasi Program Melihat Data Setoran Seluruh Jamaah

Implementasi Program Melihat Data Setoran Seluruh jamaah



Pada tabel IV-12 menjelaskan bahwa fungsi getListDataV3 adalah data menggunakan DBMS yang digunakan adalah MySQL, dimana data diambil dari *database* yang berasal dari tabel jamaah, provinsi, dan komunitas dan tersimpan didalam *variabel res array*.

Tabel IV-12. Implementasi *Query* Data Setoran Jamaah

Implementasi Query Data Setoran Jamaah

Pada tabel IV-13 menjelaskan bagaimana cara agar menampilkan setoran seluruh jamaah yang dimana menggunakan MySQL dikonversi menjadi format JSON. Data ditampung didalam fungsi array echo json_encode ($retval) untuk mengubah data array menjadi JSON.

Tabel IV-13. Implementasi Proses Query Menggunakan JSON

Implementasi Proses Query Menggunakan JSON

Dapat dilihat pada tabel dibawah untuk melihat progress setoran individu sholawat data diambil menggunakan ajax akan muncul tampilan halaman progress setoran sholawat individu jamaah berbentuk grafik saat menekan id masing-masing jamaah.

Tabel IV-14. Implementasi Progress Setoran Individu Jamaah Berbentuk Grafik

Implementasi Melihat Progress Setoran Individu Jamaah Berbentuk Grafik

Pada tabel IV-15 diatas merupakan proses cara untuk mendapatkan data ajax. Data dari progress setoran yang berasal dari *database* di inputkan data dari setoran seluruh jamaah yang telah ditampung di dalam *variabel array res* yang mengubah menjadi data json dan menggunakan paramater id jamaah untuk di tampilkan datanya menjadi grafik progress setoran sholawat individu jamaah.

Tabel IV-15. Implementasi Proses *Query* Progress Setoran Menggunakan JSON

Implementasi Proses *Query* Progress Setoran Menggunakan JSON

Dapat dilihat pada tabel IV-16 sebaran dan rerata setoran berdasarkan *gender* data diambil menggunakan ajax. Yang dimana ketika ketika *user* masuk ke halaman camp sholawat annur pada *website* maka akan langsung muncul beranda *website* camp sholawat annur dan menekan *navbar* atau menu *dashboard* yang akan muncul data sebaran dan rerata setoran berdasarkan gender berbentuk grafik.

Tabel IV-16. Implementasi Melihat Sebaran dan Rerata Setoran Berdasarakan *Gender*

Implementasi Melihat Sebaran dan Rerata setoran berdasarakan *Gender*

Pada tabel IV-7 merupakan proses untuk mendapat data ajax. Data di proses dari *database* yang sudah ada yang akan di konversi ke format JSON menggunakan *syntax* echo json_encode($retval) sebagai variabel penampung data yang di proses.

Tabel IV-17. Implementasi *Query* Sebaran dan Rerata Setoran Berdasarkan *Gender* Menggunakan JSON

Implementasi Query Sebaran dan Rerata Setoran Berdasarkan *Gender* Menggunakan JSON

4.4.5 Implementasi Antarmuka

Pada sub-bab ini proses implementasi antarmuka perangkat lunak yang dibangun sesuai dengan perancangan antarmuka yang telah dibuat pada tahap sebelumnya. Implementasi antarmuka terdiri dari 5 bagian yaitu antarmuka beranda *website* dapat dilihat pada gambar IV-16, antarmuka melihat data setoran sholawat seluruh jamaah dapat dilihat pada gambar IV-17, antarmuka melihat data progress setoran individu jamaah berbentuk grafik dapat dilihat pada gambar IV-18, antarmuka Melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik dapat dilihat pada gambar IV-19 dan gambar IV-20, antarmuka tampilan format JSON dapat dilihat pada gambar IV-21.

4.5 Fase Transisi

Pada fase transisi ini merupakan fase tahap terakhir dalam pengembangan RUP pada perangkat lunak dalam penelitian ini. Pada fase ini dilakukan pengujian terhadap perangkat lunak yang dibangun pada fase sebelumnya. Fase transisi meliputi pengujian perangkat lunak yang tediri dari rencana pengujian dan kasus uji.

4.5.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak pada peneltian ini menggunakan metode *white box* dan *black box* pada perancangan perangkat lunak dan hasil implementasi perangkat lunak.

4.5.2 Rencana Pengujian

Pengujian dilakukan untuk menguji perangkat lunak yang telah dibangun menggunakan metode *white box* dan *black box*. Rencana pengujian perangkat lunak dideskripsikan sebagai berikut:

1. Rencana pengujian *usecase* melihat data setoran seluruh jamaah

Tabel IV-18. Rencana Pengujian *Usecase* Melihat Data Setoran Seluruh Jamaah

No.	ID	Pengujian	Tingkat Pengujian
1.	U-1-1	Menekan menu "Top Setor" untuk menampilkan seluruh data setoran para jamaah berbentuk tabel	Pengujian Unit

2. Rencana pengujian *usecase* melihat data progress setoran individu jamaah berbentuk grafik

Tabel IV-19. Rencana Pengujian Melihat Data Progress Setoran Individu Jamaah Berbentuk Grafik

No.	ID	Pengujian	Tingkat Pengujian
1.	U-2-1	Menekan tombol yang berada pada kolom "ID" untuk Menampilkan data progress setoran individu jamaah	Pengujian Unit

3. Rencana pengujian melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik

Tabel IV-20. Rencana Pengujian Melihat Sebaran dan Rerata Setoran Berdasarkan *Gender* Berbentuk Grafik

No.	ID	Pengujian	Tingkat Pengujian
1.	U-3-1	Menekan menu "dashboard" untuk Menampilkan data sebaran dan rerata setoran berdasarkan *gender*	Pengujian Unit

4.5.3 Implementasi Pengujian

Sub-bab ini akan membahas mengenai hasil implementasi rencana pengujian *white box* dan *black box* yang telah dibuat.

4.5.3.1 Implementasi Pengujian *White Box*

Pada implementasi pengujian *white box* perangkat lunak ini peneliti menggunakan metode basis *path*. Basis *path* merupakan teknik uji yang digunakan untuk mengukur kompleksitas logis dari desain *procedural* dan menggunakan sebagai pedoman untuk menetapkan himpunan basis dari semua jalur eksekusi. Untuk mendapatkan *test case* digunakan untuk mengerjakan basis set pengerjaan setiap perintah min 1 kali selama uji coba.

Pengujian *usecase* melihat data setoran seluruh jamaah sebagai berikut:

1. Alur komponen yang akan diuji

   Alur komponen yang akan di uji pada metode basis *path* titik awal untuk *path testing* adalah suatu program *flowchar*t dan *flowgraph* yang menunjukkan *node-node* yang mempresentasikan satu atau lebih *procedural* menyatakan *decision* (*if-then-else*,*condition*) dan panah busur yang disebut *edge* menyatakan alur kontrol suatu *node*. Pada gambar IV-22 ada *flowchart* dan *flowgraph* menggambarkan bahwa simpul mulai *node* 1 diarahkan masuk ke dalam program, lalu pada *node* 2 menyatakan bahwa ada *decision* (*if-then-else*) jika programnya berhasil masuk akan menghantarkan masuk ke beranda *website* yang diarahkan pada *node* 3, jika programnya tidak berhasil masuk program akan *error* atau tidak tampil yang ditunjukkan pada *node* 4.

2. Perhitungan independen kompleksitas siklomatis (v)

   Pada gambar IV-22 terdapat *flowgraph* yang telah tersedia indepeden kompleksitas siklomatis dari sebuah program dapat dihitung dengan rumus sebagai berikut:

   - V(G) = E - N + 2
   - V(G) = 4 – 4 + 2 = 2
   - Jumlah *Region* = 2
   - Jumlah *Path* = 2

   Hasil independen *path* dapat dijabarkan sebagai berikut:

   - *Path* 1 = 1 – 2 – 3
   - *Path* 2 = 1 – 2 – 4

   Jalur dasar 1-2-3 bahwa sebuah program dapat menunjukkan semua pemrosesan *statement* telah dilalui dengan berhasil tidak ada pemrosesan *error*. Pada jalur dasar 1-2-4 bahwa sebuah program menunjukkan semua pemrosesan *statement* yang telah dilalui mendapatkan pemrosesan *error* yang berarti program tidak berjalan.

Pengujian *usecase* melihat data progress setoran individu jamaah berbentuk grafik

1. Alur komponen yang akan diuji

   Alur komponen pengujian *usecase* melihat data progress setoran individu jamah berbentuk grafik dapat di lihat pada gambar IV-

23. Pada gambar IV-23 terdapat *flowchart* dan *flowgraph* yang menggambarkan bahwa simpul mulai ditunjukkan pada *node* 1 masuk ke dalam program dan diarahkan pada node 2 yang dimana *node* 2 menyatakan (*if-then-else*) jika program berhasil masuk akan diarahkan pada *node* 3 dan jika program tidak berhasil akan diarahkan pada *node* 4 yang dimana program *error* atau tidak tampil. Lalu jika berhasil program akan diarahkan pada *node* 5 dan akan ada pernyataan (*if-then-else*) pada node 6, jika program berhasil akan menghantarkan pada *node* 7 dan 9. Jika program tidak berhasil programnya tampil akan tetapi datanya dan grafiknya tidak tampil yang diarahkan pada *node* 8 dan 9.

2. Perhitungan independen kompleksitas siklomatis (v)

   Pada gambar IV-23 *flowgraph* yang tersedia indepeden kompleksitas siklomatis dari sebuah program dapat dihitung dengan rumus sebagai berikut:

   - V(G) = E - N + 2
   - V(G) = 10 – 9 + 2 = 3
   - Jumlah *Region* = 3
   - Jumlah *Path* = 3

   Hasil independen *path* dijabarkan sebagai berikut:

   - *Path* 1 = 1 – 2 – 3 – 5 – 6 – 8 – 9

- *Path* 2 = 1 – 2 – 4
- *Path* 3 = 1 – 2 – 3 – 5 – 6 – 7 – 9

Jalur dasar 1-2-3-5-6-8-9 yang didapatkan bahwa sebuah program menunjukkan semua pemrosesan *statement* telah dilalui dengan berhasil tidak ada pemrosesan *error*. Pada jalur dasar 1-2-4 bahwa sebuah program menunjukkan semua pemrosesan *statement* yang telah dilalui mendapatkan pemrosesan *error* yang berarti program tidak berjalan. Dan pada jalur dasar 1-2-3-5-6-7-9 sebuah program menunjukkan bahwa semua pemrosesan yang dilalui ada satu pemrosesan *statement* yang belum dilalui mendapatkan pemrosesan *erorr* yang berarti program berhasil berjalan akan tetapi pada saat program yang diinginkan program tidak tampil. Jalur dasar yang lebih pendek kemungkinan tidak kompleks dibanding dengan jalur dasar yang lebih panjang.

Pengujian *usecase* melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik

1. Alur komponen yang akan diuji

   Alur komponen pengujian melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik dapat dilihat pada gambar IV-24. Pada gambar IV-24 *flowchart* dan *flowgraph* menggambarkan

titik mulai program masuk ditunjukkan pada *node* 1 dan diarahkan pada *node* 2 menyatakan (*if-then-else*). Jika program berhasil masuk akan diarahkan pada *node* 3 dan *node* 5 menyatakan (*if-then-else*) jika berhasil masuk akan menampilkan data grafik diarahkan pada *node* 6, jika program tidak berhasil masuk program akan *error* dan data grafiknya tidak muncul diarahkan pada node 7.

2. Perhitungan independen kompleksitas siklomatis (v)

   Flowgraph pada gambar IV-23 telah tersedia maka independen kompleksitas siklomatis dari sebuah program dapat dihitung dengan rumus sebagai berikut:

   - V(G) = E - N + 2
   - V(G) = 7 – 6 + 2 = 3
   - Jumlah *Region* = 3
   - Jumlah *Path* = 3

   Hasil independen path dapat dijabarkan sebagai berikut:

   - *Path* 1 = 1 – 2 – 3,5 – 6
   - *Path* 2 = 1 – 2 – 4
   - *Path* 3 = 1 – 2 – 3,5 – 7

   Jalur dasar 1-2-3,5-6 yang didapatkan bahwa sebuah program menunjukkan semua pemrosesan *statement* telah dilalui dan ada dua *statement* sekaligus dilalui dengan berhasil tidak ada pemrosesan *error*. Pada jalur dasar 1-2-4

bahwa sebuah program menunjukkan semua pemrosesan *statement* yang telah dilalui mendapatkan pemrosesan *error* yang berarti program tidak berjalan. Dan pada jalur dasar 1-2-3,5-6 sebuah program menunjukkan bahwa semua pemrosesan yang dilalui ada dua pemrosesan *statement* dilalui dengan sekaligus dan ada statement yang belum dilalui mendapatkan pemrosesan *erorr* yang berarti program berhasil berjalan akan tetapi pada saat program yang diinginkan program tidak tampil. Jalur dasar yang lebih pendek dan jalur dengan dua statement sekaligus kemungkinan tidak kompleks.

4.5.3.2 Implementasi Pengujian *Black Box*

Pengujian *usecase* melihat data setoran seluruh jamaah

Table IV-21. Pengujian *Usecase* Melihat Data Setoran Seluruh Jamaah

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
1.	Menekan menu "Top Setor" untuk Menampilkan seluruh data setoran para jamaah berbentuk tabel	Menekan menu " Top Setor"	Tidak Ada	Data yang ingin ditampilkan akan menampilkan seluruh data setoran berbentuk tabel	Perangkat lunak dapat menampilkan seluruh data setoran berbentuk tabel	Terpenuhi

Pengujian *usecase* melihat data progress setoran individu jamaah berbentuk grafik.

Tabel IV-22. Pengujian *Usecase* Melihat Data Progress Setoran Individu Jamaah Berbentuk Grafik

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
1.	Menekan tombol yang berada didalam "ID" untuk Menampilkan progress setoran individu jamaah berbentuk grafik	Menekan tombol yang berada pada kolom "ID"	Tidak Ada	Data yang ingin ditampilkan akan menampilkan progress setoran individu jamaah berbentuk grafik	Perangkat lunak dapat menampilkan progress setoran individu jamaah berbentuk grafik	Terpenuhi

Pengujian *usecase* melihat sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik

Tabel IV-23. Pengujian *Usecase* Melihat Sebaran dan Rerata Setoran Berdasarkan *Gender* Berbentuk Grafik

ID	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
1.	Menekan menu "dashboard" untuk Menampilkan data sebaran dan rerata setoran berdasarkan	Menekan menu "dashboard"	Tidak Ada	Data yang ingin ditampilkan akan menampilkan data sebaran dan rerata setoran berdasarkan *gender* berbentuk grafik	Perangkat lunak dapat menampilkan data sebaran dan rerata setoran berdasarkan *gender*	Terpenuhi

	gender berbentuk grafik				berbentuk grafik	

4.6 Kesimpulan

Pada bab 4 ini proses pengembangan perangkat lunak menggunakan metode RUP dapat disimpulkan bahwa kegiatan analisis, perancangan, implementasi, dan pengujian perangkat lunak pemetaan basis data kedalam format JSON menggunakan *query builder codeigniter* untuk grafik dashboard berbasiskan MVC ini telah dilakukan dengan baik dan perangkat lunak memenuhi pengujian dengan menggunakan metode *white box* dan *black box*.

BAB V

HASIL DAN ANALISIS PENELITIAN

5.1 Pendahuluan

Pada bab ini akan membahas mengenai hasil dan analisis *project* yang telah diperoleh dari hasil pengembangan dan pengujian perangkat lunak akan dibahas lebih rinci.

5.2 Hasil Penelitian

5.2.1 Konfigurasi Percobaan

Pengujian pemetaan basis data ke dalam format JSON ini dilakukan uji coba dengan *tools postman* untuk mengukur waktu kecepatannya, status, dan berapa besar data. *postman* adalah aplikasi untuk melakukan pengujian *web server* atau API. Uji coba yang dilakukan adalah mengakses *localhost* dengan 2 perangkat yang berbeda dan terhubung dalam satu jaringan WLAN yang sama.

5.2.2 Hasil Uji Percobaan

A. Uji coba pada *localhost*

Uji coba yang pertama dilakukan pada *localhost*, pada tabel V-1 hasil percobaan uji coba ini dilakukan sebanyak 10 kali percobaan pada *tools postman*. Yang dimana mendapatkan hasil kecepatan waktu yang berbeda-beda dari 10 kali uji coba. Kecepatan waktu didapatkan tidak tentu karena berdasarkan jaringan yang digunakan. Besar data JSON data setoran yang didapatkan pada uji coba dalam 10 kali dilakukan selalu sama dengan besar data 39.31 KB, akan tetapi ada satu besar

data yang berbeda pada percobaan pertama yaitu 39.43 KB. Semakin besar data berarti data pada JSON yang digunakan banyak data tabel dari *database* yang digunakan besar data juga muncul dengan tidak tentu, akan mendapatkan besar data didalam salah satu percobaan tersebut. Status yang didapatkan menunjukkan 200 ok bahwa JSON yang di uji coba berhasil.

Tabel V-1. Hasil Percobaan Uji Coba JSON Data Setoran pada *Localhost*

No.	Estimasi Waktu	Size Data	Status
1.	602 ms	39.43 KB	200 ok
2.	510 ms	39.31 KB	200 ok
3.	511 ms	39.31 KB	200 ok
4.	419 ms	39.31 KB	200 ok
5.	476 ms	39.31 KB	200 ok
6.	457 ms	39.31 KB	200 ok
7.	442 ms	39.31 KB	200 ok
8.	424 ms	39.31 KB	200 ok
9.	426 ms	39.31 KB	200 ok
10.	436 ms	39.31 KB	200 ok

Pada tabel V-2 menjelaskan bahwa hasil percobaan uji coba pada *localhost* dilakukan sebanyak 10 kali dengan *tools postman*, Kecepatan waktu yang didapatkan pada uji coba tidak tentu. Besar data JSON pada *session id* jamaah yang didapatkan kecil bahwa besar data JSON tersebut tidak banyak data tabel dari

database yang digunakan, uji coba pada 10 kali percobaan mendapatkan besar data yang selalu sama yaitu 1.53, akan tetapi ada satu besar data yang berbeda pada 10 kali percobaan uji coba tersebut dengan besar data 1.66 KB pada percobaan pertama. Status 200 ok pada uji coba pada tabel dibawah ini menunjukkan bahwa JSON yang di uji coba tersebut berhasil.

Tabel V-2. Hasil Percobaan Uji Coba JSON *Session Id* Jamaah pada *Localhost*

No.	Estimasi Waktu	Size Data	Status
1.	365 ms	1.66 KB	200 ok
2.	263 ms	1.53 KB	200 ok
3.	322 ms	1.53 KB	200 ok
4.	331 ms	1.53 KB	200 ok
5.	336 ms	1.53 KB	200 ok
6.	484 ms	1.53 KB	200 ok
7.	293 ms	1.53 KB	200 ok
8.	273 ms	1.53 KB	200 ok
9.	263 ms	1.53 KB	200 ok
10.	306 ms	1.53 KB	200 ok

Tabel di bawah ini menjelaskan bahwa uji coba pada *localhost* dilakukan sebanyak 10 kali percobaan dengan waktu yang didapatkan berbeda-beda yang dimana waktu tersebut tidak tentu karena jaringan yang digunakan pada uji coba.

Dan besar data yang didapatkan pada uji coba data JSON sebaran setoran berdasarkan *gender* tidak terlalu besar yang berarti data yang digunakan dari tabel *database* tidak terlalu banyak sehingga membuat data JSON saat di uji coba menghasilkan hanya 2.42 KB, dari 10 kali percobaan uji coba besar datanya selalu sama, akan tetapi pada pertengahan percobaan ada besar data yang berbeda dengan besar data 2.54 KB. Status 200 ok menunjukkan bahwa JSON yang diuji coba berhasil.

Tabel V-3. Hasil Percobaan Uji Coba JSON Sebaran Setoran Berdasarkan *Gender* pada *Localhost*

No.	Estimasi Waktu	Size Data	Status
1.	263 ms	2.42 KB	200 ok
2.	310 ms	2.42 KB	200 ok
3.	291 ms	2.42 KB	200 ok
4.	237 ms	2.42 KB	200 ok
5.	217 ms	2.42 KB	200 ok
6.	364 ms	2.54 KB	200 ok
7.	246 ms	2.42 KB	200 ok
8.	250 ms	2.42 KB	200 ok
9.	210 ms	2.42 KB	200 ok
10.	229 ms	2.42 KB	200 ok

Pada gambar V-1, gambar V-2 dan gambar V-3 adalah salah satu hasil percobaan uji coba kecepatan dengan *localhost*. Pada gambar tersebut format json yang diuji coba menampilkan status, *time*, dan *size*. Jika data JSON berhasil atau berjalan saat diuji data JSON akan muncul pada tampilan di *tools postman* tersebut, yang dimana waktu dan besar data akan berbeda-beda setiap kali kita kirim untuk uji coba data tersebut seperti yang didapatkan pada tabel V-1, tabel V-2, dan tabel V-3.

B. Uji Coba pada WLAN

Uji coba WLAN dilakukkan pada perangkat yang terhubung 1 jaringan dengan *localhost* dapat dilihat pada tabel V-4 kecepatan waktu yang didapat pada saat uji coba dengan WLAN menghasilkan kecepatan yang berbeda-beda juga. Besar data pada uji coba JSON data setoran pada WLAN menghasilkan besar data yang sama dengan pengujian pada localhost sebelumnya yang besarnya selalu sama yaitu 39.31 KB yang berarti isi JSON data setoran pada uji coba WLAN sama. Status 200 ok menunjukkan bahwa JSON yang diuji berhasil.

Tabel V-4. Hasil Percobaan Uji Coba JSON Data Setoran pada WLAN

No.	Estimasi Waktu	Size Data	Status
1.	508 ms	39.31 KB	200 ok
2.	397 ms	39.31 KB	200 ok
3.	359 ms	39.31 KB	200 ok

4.	382 ms	39.31 KB	200 ok
5.	484 ms	39.31 KB	200 ok
6.	356 ms	39.31 KB	200 ok
7.	398 ms	39.31 KB	200 ok
8.	505 ms	39.31 KB	200 ok
9.	490 ms	39.31 KB	200 ok
10.	413 ms	39.31 KB	200 ok

Pada tabel V-5 hasil percobaan uji coba JSON *session id* jamaah pada WLAN sebanyak 10 kali percobaan mendapatkan kecepatan waktu tidak tentu. Besar data yang didapatkan sama besarnya dengan besar data pada saat uji coba pada *localhost* sebelumnya yaitu 1.53 KB dan mendapatkan ada satu besar data yang berbeda yaitu 1.66 KB, yang berarti isi JSON tersebut sama tidak ada yang berubah dengan pengujian pada *localhost*. Status 200 ok menunjukkan bahwa JSON tersebut berhasil.

Tabel V-5. Hasil Percobaan Uji Coba JSON *Session Id* Jamaah pada WLAN

No.	Estimasi Waktu	Size Data	Status
1.	274 ms	1.66 KB	200 ok
2.	319 ms	1.53 KB	200 ok
3.	375 ms	1.53 KB	200 ok
4.	403 ms	1.53 KB	200 ok
5.	318 ms	1.53 KB	200 ok

6.	220 ms	1.53 KB	200 ok
7.	316 ms	1.53 KB	200 ok
8.	300 ms	1.53 KB	200 ok
9.	337 ms	1.53 KB	200 ok
10.	247 ms	1.53 KB	200 ok

Pada tabel V-6 di bawah ini hasil percobaan uji coba JSON data sebaran setoran berdasarkan *gender* pada WLAN mendapatkan kecepatan yang tidak tentu. Besar data yang didapatkan pada uji coba WLAN hasilnya sama dengan uji coba pada *localhost* sebelumnya yaitu 2.42 KB dan ada satu besar data yang berbeda juga pada percobaan pertama mendapatkan besar data 2.54 KB, yang berarti isi JSON yang diuji tidak ada yang berubah. Status 200 ok menandakan bahwa JSON yang diuji berhasil.

Tabel V-6. Hasil Percobaan Uji Coba JSON Sebaran Setoran Berdasarkan *Gender* pada WLAN

No.	Estimasi Waktu	Size Data	Status
1.	360 ms	2.54 KB	200 ok
2.	209 ms	2.42 KB	200 ok
3.	240 ms	2.42 KB	200 ok
4.	209 ms	2.42 KB	200 ok

5.	210 ms	2.42 KB	200 ok
6.	205 ms	2.42 KB	200 ok
7.	217 ms	2.42 KB	200 ok
8.	322 ms	2.42 KB	200 ok
9.	217 ms	2.42 KB	200 ok
10.	228 ms	2.42 KB	200 ok

Pada gambar V-4, gambar V-5, dan gambar V-6 merupakan salah satu tampilan hasil dari percobaan uji coba JSON data setoran, *session id* jamaah sebaran setoran berdasarkan *gender* pada WLAN. Pada gambar tersebut uji coba kecepatan yang dilakukan menghasilkan status, *time*, dan *size*, dan jika JSON berjalan dengan baik JSON tersebut akan tampil pada halaman *postman* dengan status 200 ok yang menunjukkan bahwa JSON tersebut berhasil.

5.3 Analisis Hasil Penelitian

Hasil Percobaan menggunakan postman pada *website* yang dibangun ini terangkum pada garis grafik. Perbandingan waktu kecepatan dengan *localhost* dan WLAN dapat dilihat pada gambar V-7, gambar V-8, dan gambar V-9.

Pada gambar V-7 menunjukkan bahwa pengujian pemetaan basis data ke format JSON pada data setoran yang dilakukan 10 kali pengujian dengan besar *size* yang hampir sama sekitar 39.31 KB – 39.43 KB. Pengujian dilakukan dengan *localhost* dimana waktu pengujian pertama

kecepatan mencapai 602 ms, selanjutnya pengujian kedua turun sekitar 510 ms – 511 ms, seterusnya sampai 10 kali pengujian kecepatan dengan *localhost* waktunya stabil sekitar 429 ms – 479 ms. Sedangkan dengan WLAN kecepatan pengujian pertama 508 ms, selanjutnya pengujian kedua turun sekitar 397 ms. Pada pengujian seterusnya kecepatan dengan WLAN naik turun dari sekitar 400-an turun 300-an ms dan naik hingga 500-an ms dan pengujian ke 10 turun menjadi 413 ms.

Pada gambar V-8 menunjukkan bahwa perbandingan kecepatan pemetaan basis data ke dalam JSON data *session id* jamaah yang dilakukan sebanyak 10 kali dengan *size* sama sekitar 1.53 KB – 1.66 KB. Uji coba pada *localhost* mendapatkan 365 ms pada percobaan pertama, pada percobaan kedua dan seterusnya kecepatannya menurun sekitar 263 ms – 336 ms, akan tetapi di pertengahan percobaan uji coba pada *localhost* mendapatkan 484 ms. Sedangkan dengan uji coba WLAN pada percobaan pertama kecepatan pada 274 ms, percobaan kedua dan seterusnya mendapatkan kecepatan 220 ms – 375 ms, sama seperti uji coba pada localhost pada uji coba WLAN pertengahan percobaan mendapatkan kecepatan sekitar 403 ms.

Gambar diatas menunjukkan perbandingan kecepatan pemetaan basis data ke dalam JSON data sebaran setoran berdasarkan gender yang dilakukan 10 kali pengujian dengan *size* sama sekitar 2.42 KB – 2.54 KB. Pengujian dilakukan dengan *localhost* dan WLAN yang dimana kecepatan keduanya sama dan stabil sekitar 205-362 ms, akan tetapi pengujian pertama dengan *localhost* dengan kecepatan 263 ms, sedangkan dengan WLAN

dengan kecepatan sekitar 360 ms. Pada pengujian ke 10 *localhost* dan WLAN sama-sama berada di kecepatan yang hampir sama *localhost* dengan kecepatan 229 ms dan WLAN kecepatan 228 ms.

Grafik pada percobaan uji coba kecepatan pemetaan basis data ke dalam format JSON dengan *tools postman* menggunakan 2 perangkat dan terhubung dalam 1 jaringan yang sama dari 10 kali uji coba menghasilkan persentase sekitar 70% pada localhost dan 65% pada WLAN.

5.4 Kesimpulan

Hasil Pengujian Kecepatan pemetaan basis data ke dalam format JSON menggunakan 2 perangkat dan terhubung dalam 1 jaringan yang sama menghasilkan kecepatan yang berbeda. dapat disimpulkan bahwa pengujian dengan *localhost* memiliki kecepatan lebih stabil dibandingkan dengan perangkat yang terhubung melalui WLAN, akan tetapi kecepatan diantara kedua perangkat tersebut tidak jauh berbeda dengan persentase sekitar 70% - 65%.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang didapat setelah melakukan penelitian *project* mengenai pemetaan basis data ke dalam format JSON menggunakan *query builder codeigniter* untuk grafik *dashboard* berbasiskan MVC adalah sebagai berikut:

1. Pemetaan basis data ke dalam format JSON menggunakan *query builder codeigniter* berhasil di implementasikan.
2. Pengembangan perangkat lunak pada *projec*t ini untuk grafik *dashboard* berbasiskan MVC berhasil di implementasikan.
3. Pengujian kecepatan dilakukan menggunakan *tools postman*. Hasil Pengujian Kecepatan pemetaan basis data ke dalam format JSON menggunakan dua perangkat dan terhubung dalam satu jaringan yang sama menghasilkan kecepatan yang berbeda, akan tetapi kecepatan diantara kedua perangkat tersebut tidak jauh berbeda yang menghasilkan persentase sekitar 70% - 65%.

6.2 Saran

Adapun saran yang dapat digunakan pada penelitian selanjutnya, sebagi berikut:

1. Melakukan proses join tabel langsung dari tabel *view database* tidak melalui pemrosesan pada halaman php agar saat menggabung tabel-tabel yang ada pada *database* hanya menganggil satu tabel saja.
2. Dapat menggunakan *framework* lain seperti *laravel*.
3. Selain menggunakan format JSON dapat menggunakan format lain seperti XML.

Pemetaan Basis Data Ke Dalam Format JSON Menggunakan Query Builder CodeIgniter Untuk Grafik Dashboard Berbasiskan MVC

ORIGINALITY REPORT

14%	11%	2%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sriwijaya University Student Paper	10%
2	ejournal.amikdumai.ac.id Internet Source	1%
3	jurnal.stikomcki.ac.id Internet Source	1%
4	repository.usd.ac.id Internet Source	1%
5	docplayer.info Internet Source	1%
6	123dok.com Internet Source	1%
7	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur Student Paper	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		

SURAT KETERANGAN PENGECEKAN SIMILARITY

Saya yang bertanda tangan di bawah ini

Nama	: Putri Pebreisnaini
Nim	: 09021381823126
Prodi	: Teknik Informatika Bilingual
Fakultas	: Ilmu Komputer

Menyatakan bahwa benar hasil pengecekan similarity Skripsi/Tesis/Disertasi/Lap. Penelitian yang berjudul Pemetaan Basis Data Ke Dalam Format JSON Menggunakan Query Builder CodeIgniter Untuk Grafik Dashboard Berbasiskan MVC adalah 14%. Dicek oleh operator *:

1. Dosen Pembimbing
2. UPT Perpustakaan
3. Operatur Fakultas

Demikianlah surat keterangan ini saya buat dengan sebenarnya dan dapat saya pertanggung jawabkan.

Indralaya, 1 September 2022

Menyetujui
Dosen pembimbing,

Dr. M. Fachrurrozi, M.T
NIP. 198005222008121002

Yang menyatakan,

Putri Pebreisnaini
NIM. 09021381823126