

3. ANALISA DAN DESAIN SISTEM

Bab ini membahas desain sistem yang akan digunakan. Analisa, desain dan perancangan sistem merupakan tahap awal sebelum dilakukan tahap pembuatan sistem. Hal yang akan dijelaskan adalah analisis masalah, perencanaan implementasi, desain arsitektur sistem, konfigurasi program, perencanaan metode *scalability* serta penjelasan mengenai PRS online.

3.1. Analisis Masalah

Sistem pendaftaran rencana studi online yang dilakukan saat pergantian semester menyebabkan banyaknya pengguna baik mahasiswa maupun dosen yang melakukan akses ke database *server* PRS. Dengan adanya *login* secara bersamaan dalam jumlah yang sangat banyak, menyebabkan *server* menjadi lambat dan mengalami kendala saat proses berlangsung. Hal ini menyebabkan beberapa pengguna tidak dapat *login* ke dalam aplikasi PRS guna melakukan pendaftaran mata kuliah.

Alternatif yang sekarang sudah dilakukan yaitu dengan membagi hak akses kepada tiap mahasiswa sesuai jurusannya dengan hari dan jam untuk mengakses yang berbeda-beda. Dengan pembagian jadwal tersebut, banyak mahasiswa yang sering lalai dan terkadang masih mengalami kendala lambatnya *server*. Dirasa ada alternatif lain yaitu dengan melakukan *scale* jalur akses ke sistem PRS online sesuai dengan kebutuhan.

3.2. Rencana Implementasi

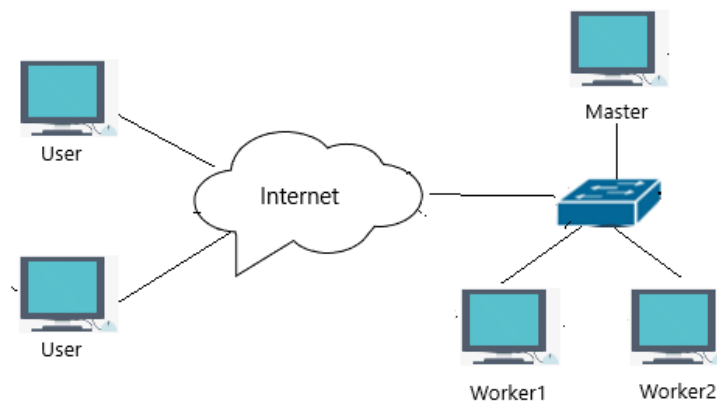
Pada implementasi *container* Kubernetes untuk mendukung *scalability*, maka diperlukan rencana untuk mempersiapkan proses implementasi. Langkah-langkah tersebut yaitu :

- Melakukan instalasi dan konfigurasi Ubuntu baik *desktop* maupun *server*.
- Melakukan instalasi Kubernetes, Docker, serta *konfigurasi* untuk *master node* dan *worker node*.
- Melakukan implementasi aplikasi yang akan di *scalable*.

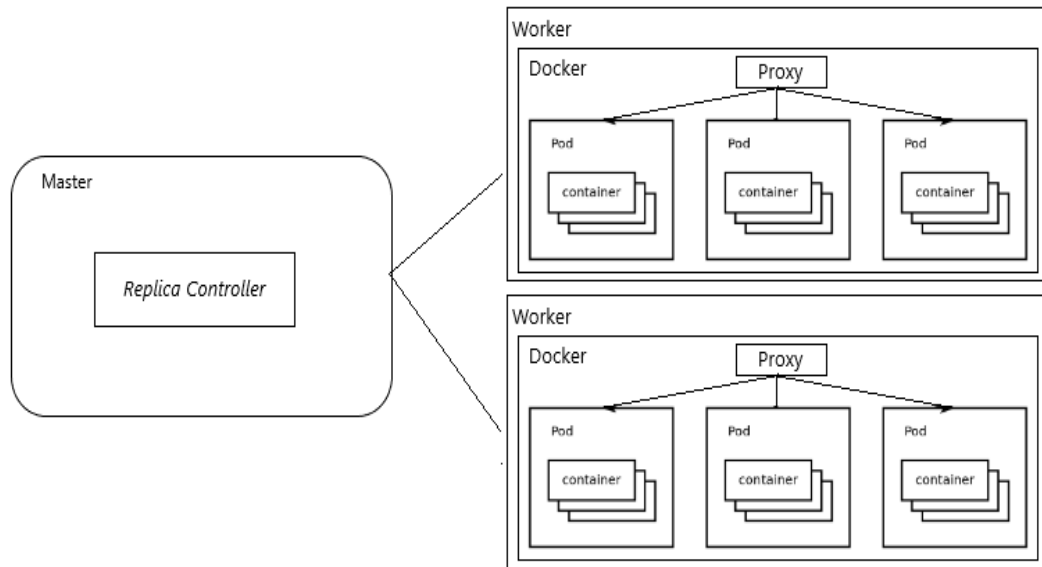
Setelah pembuatan sistem selesai, akan dilakukan implementasi *microservice* PRS online pada Kubernetes. Kemudian dilanjutkan dengan pengujian *concurrent user* dan *response time* yang akan dibandingkan antara *single server* dan *container*.

3.3. Desain Arsitektur Sistem

Dari permasalahan yang terjadi, maka diberikan solusi untuk mengatasi masalah banyaknya *user* yang mengakses secara bersamaan. Dengan mengimplementasikan aplikasi kedalam Kubernetes *container*, akan memberikan jalan alternatif kepada *user* yang mengakses. Untuk Kubernetes yang akan digunakan terdiri dari 3 komputer, dimana 1 buah komputer digunakan sebagai *master node*, 2 komputer digunakan sebagai *worker node*. Sedangkan untuk simulasi *multiple user* yang akan dilakukan menggunakan aplikasi untuk menggenerate *multiple user* dan akan mengakses *url* dari *microservice* itu sendiri. Pada *master* dilakukan pengaturan *microservice* yang terdapat di *worker*, dimana pada setiap *worker* nantinya akan terdapat beberapa *container*. Berikut pada Gambar 3.1 dijelaskan mengenai desain arsitektur sistem yang akan dibuat. Sedangkan mengenai desain *container* pada *worker* Kubernetes akan dijelaskan pada Gambar 3.2.



Gambar 3.1 Desain arsitektur sistem



Gambar 3.2 Desain *container* pada *worker* Kubernetes

3.4. Desain Alur Aplikasi

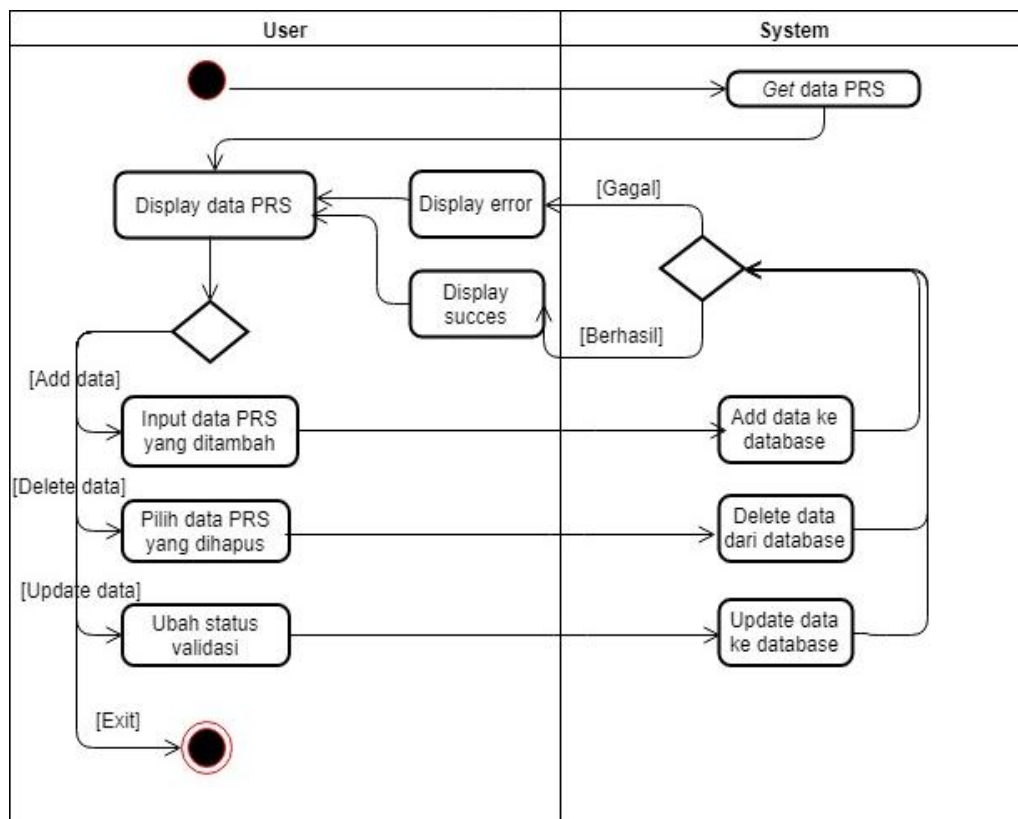
Bagian ini akan menjelaskan mengenai alur tentang gambaran jalannya sistem yang akan dibuat. Untuk melakukan *scalability* terhadap aplikasi yang akan di implementasikan, dibutuhkan langkah-langkah yang harus dijalankan yaitu melakukan *create and deployment* sistem *microservice*, pengaturan konfigurasi untuk *container* Kubernetes, pengaturan *scalability* terhadap aplikasi yang sudah di implementasi, dan yang terakhir yaitu pengujian oleh *multiple user*.

Langkah pertama yang dilakukan yaitu mengatur *setting* pada Kubernetes termasuk melakukan instalasi pada *master* dan *worker* dimana pada *worker* tersebut akan dibuat *container-container* yang akan dipakai sesuai dengan kebutuhan. Langkah kedua yang dilakukan yaitu dengan melakukan *create and deployment* sistem *microservice* yang sudah dibuat. Kemudian langkah selanjutnya yaitu melakukan *setting up scalability* untuk pengaturan *container* yang akan diuji, dengan memperhatikan replikasi *container* yang akan bertambah maupun berkurang sesuai dengan kebutuhan.

Pada proses *setting up* Kubernetes akan dilakukan instalasi *ansible*, *master* dan *worker* untuk Kubernetes. Dimana *ansible* berfungsi untuk melakukan pengelolaan *server* yang lebih mudah dan terstruktur, serta dapat menghemat

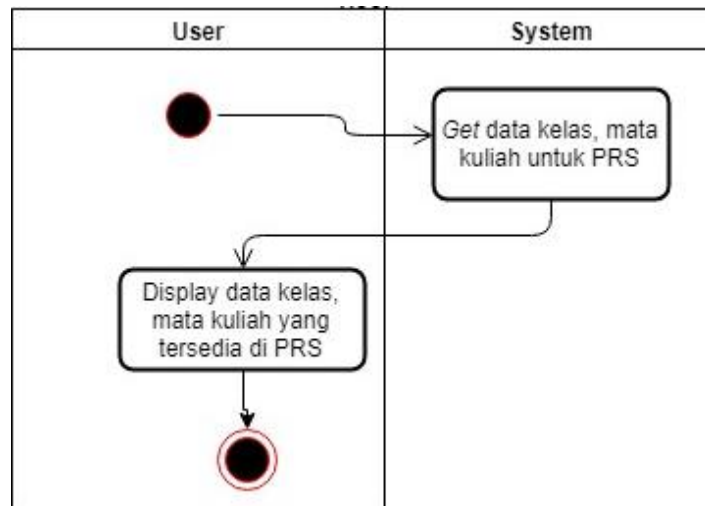
waktu yang ada selagi melakukan instalasi untuk *worker-worker* yang ada. *Master* pada Kubernetes berfungsi untuk mengatur semua *container* yang ada. Dimana *worker* berfungsi sebagai pekerja dan tempat dimana *container* bekerja.

Untuk proses *create and deployment microservice* dimana proses pembuatan *microservice* akan dilakukan terlebih dahulu sebelum proses implementasi. Berikut pada Gambar 3.3 berisi gambar mengenai rancangan proses keseluruhan dari *webservice* yang akan dibuat, dimana nantinya akan dibagi menjadi beberapa *microservice* berdasarkan fungsi dari masing-masing *service*.



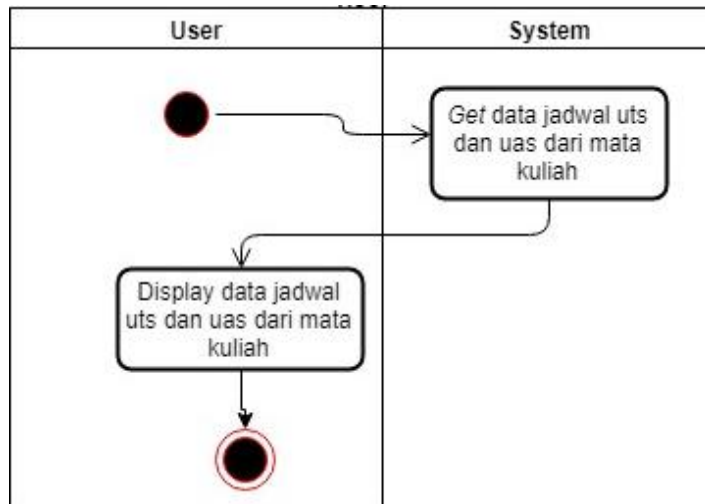
Gambar 3.3 Activity diagram *microservice* secara keseluruhan

Untuk *microservice* yang dibuat akan menggunakan REST dimana berfungsi sebagai API untuk tiap *microservice* yang ada. Serta *slim framework* digunakan untuk proses pembuatan *microservice*. Terdapat 4 fungsi utama yang nantinya akan dibuat pada *microservices* yaitu *get*, *post*, *delete* dan *put*. Dimana tiap fungsi tersebut mempunyai tujuan yang berbeda-beda. Berikut merupakan penjelasan detail tiap *microservice* yang akan dibuat. Yang pertama yaitu *microservice* untuk mendapatkan data kelas dan jadwal mata kuliah yang akan diambil oleh mahasiswa nantinya. Dimana ketika melakukan PRS online mahasiswa cenderung untuk melihat daftar kelas dan mata kuliah yang sedang buka atau wajib diambil, berikut penjelasan lebih lanjut terlihat pada Gambar 3.4



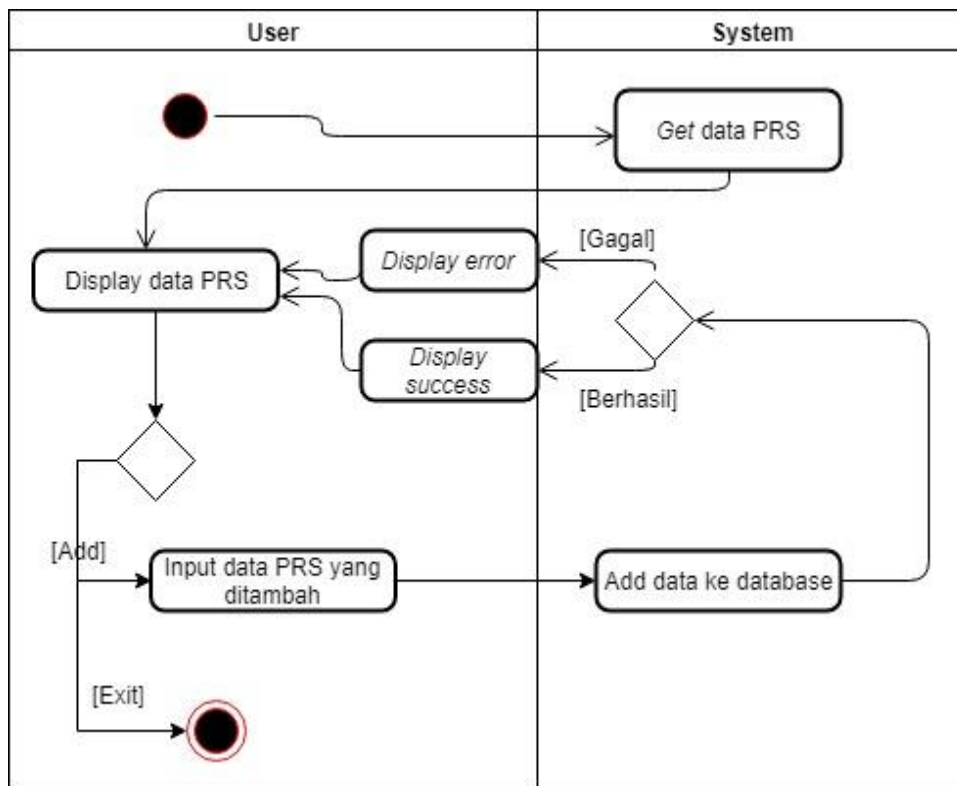
Gambar 3.4 Activity diagram dari *microservice* untuk *get* jadwal kelas dan mata kuliah

Kemudian yang kedua yaitu *microservice* untuk melakukan *get* pada jadwal UTS dan UAS dari mata kuliah yang ada. Dimana mahasiswa perlu memastikan agar kelas dari mata kuliah yang mereka ambil tidak memiliki jadwal UTS maupun UAS yang bertabrakkan. Berikut pada Gambar 3.5 merupakan *activity diagram* dari *microservice* untuk melakukan *get* jadwal UTS dan UAS.



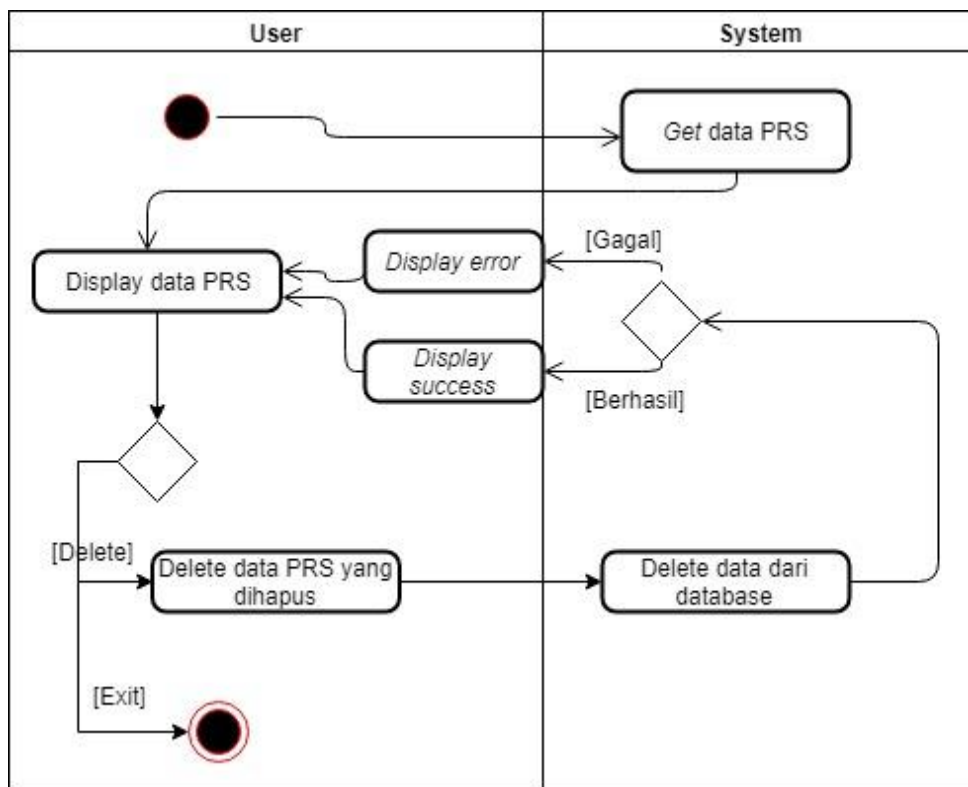
Gambar 3.5 *Activity diagram* dari *microservice* untuk *get* jadwal UTS dan UAS dari mata kuliah

Kemudian selanjutnya *microservice* yang akan dibuat yaitu fungsi *post* untuk melakukan input ke dalam database mengenai mata kuliah yang sudah dipilih oleh mahasiswa. Berikut pada Gambar 3.6 merupakan *activity diagram* dari *microservice* untuk fungsi *post*.



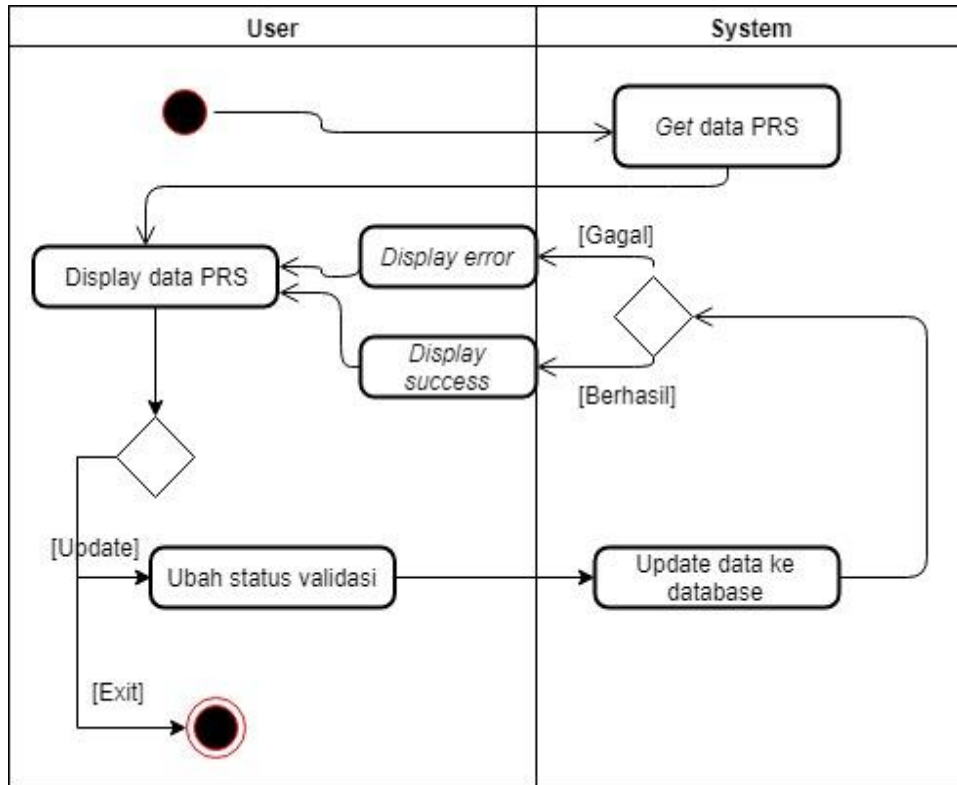
Gambar 3.6 *Activity diagram* dari *microservice* untuk fungsi *post*

Microservice yang selanjutnya yaitu fungsi *delete* untuk melakukan *delete* dari mata kuliah yang sudah dipilih jika terjadi pembatalan pilihan. Selama prs belum di validasi, mahasiswa dapat melakukan proses *delete* mata kuliah yang batal dipilih, namun tidak berlaku untuk mahasiswa yang sudah melakukan validasi. Berikut pada Gambar 3.7 merupakan *activity diagram* dari *microservice* untuk fungsi *delete*.



Gambar 3.7 Activity diagram dari *microservice* untuk fungsi *delete*

Kemudian *microservice* untuk fungsi *put* pada *microservice* yang bertujuan untuk mengubah status *validasi* mahasiswa agar tidak dapat melakukan fungsi *post* atau *delete* lagi. Berikut pada Gambar 3.8 merupakan *activity diagram* dari *microservice* fungsi *put* untuk proses *update*.



Gambar 3.8 Activity diagram untuk *microservice* dengan fungsi *update* status validasi.

Pada proses selanjutnya yaitu konfigurasi *scalability* pada Kubernetes akan dilakukan *setting horizontal pod autoscale* yang berisi *setting cpu percent* serta *min* dan *max pod* untuk menentukan *replica container* yang akan ter-*create* sesuai dengan kebutuhan. Dimana nantinya akan di lakukan pengaturan batas *cpu* atau *memory load* untuk tiap *pod microservice* yang tersedia. Ketika *user request* yang mengakses *microservice* melebihi batas *cpu load* maka otomatis akan terbentuk *replica* dari *pod* tersebut. Dan apabila *user request* sudah dibawah batas *cpu load*, maka otomatis *replica* yang sudah tidak dibutuhkan akan melakukan *scaling down container*.

3.5. Aplikasi Sistem Informasi Manajemen

Pada bagian ini akan menjelaskan mengenai gambaran dari PRS online serta gambaran *web service* dari PRS online serta contoh *website* dari PRS online, dan *design database system* yang akan dibuat.

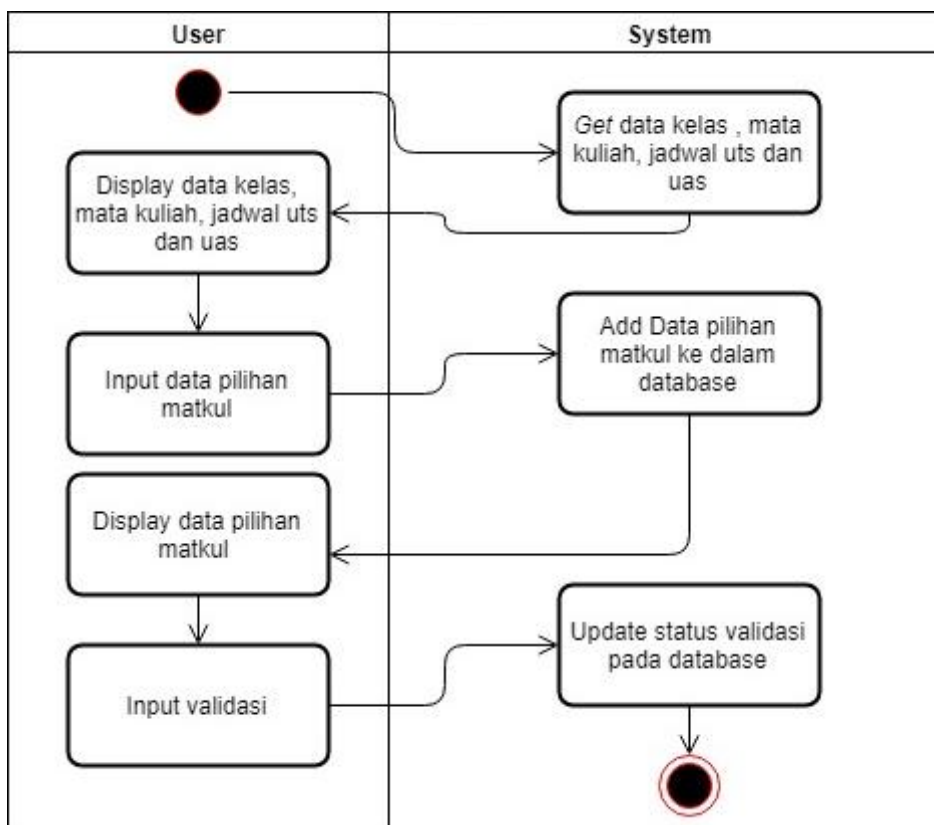
3.5.1. Gambaran PRS Online

Aplikasi yang tersedia pada Universitas Kristen Petra tentunya bermacam-macam. Salah satunya adalah Sistem Informasi Manajemen, dimana SIM merupakan wadah bagi mahasiswa maupun dosen mengenai kegiatan yang berada di lingkungan kampus Universitas Kristen Petra. Fokus gambaran besar skripsi ini kepada SIM Akademik yang berisi tentang kegiatan proses perkuliahan serta dapat diakses oleh mahasiswa maupun dosen. Kegiatan tentang jadwal kuliah, PRS online, status mahasiswa, juga nilai mahasiswa semuanya tertampung disini.

Skripsi ini lebih difokuskan lagi hanya ke bagian PRS online. PRS online merupakan wadah bagi mahasiswa untuk melakukan pendaftaran mata kuliah yang akan diambilnya setiap semester. Pendaftaran Rencana Studi ini hanya dibuka saat awal tiap semester. PRS online terbagi menjadi tiga bagian yaitu PRS 1, PRS 2, dan PRS 3 PRS 1 dimana semua mempunyai tujuan yang berbeda. PRS 1 merupakan Pendaftaran Rencana Studi yang dibuka pada tahap pertama. Para mahasiswa (selain mahasiswa baru) diharuskan melakukan pendaftaran mata kuliah yang akan mereka pilih baik itu mata kuliah wajib maupun pilihan.

Sebelum melakukan PRS 1, tahap perwalian terhadap dosen wali perlu dilalui oleh mahasiswa. Setelah dosen wali membuka hak akses mahasiswa untuk melakukan PRS 1, barulah mahasiswa dapat mendaftarkan mata kuliah yang dipilihnya. Kemudian akan ada selang waktu untuk Perubahan Rencana Studi (PRS 2), dimana mahasiswa dapat melakukan perubahan rencana studi. Dan tahap akhir adalah Pembatalan Rencana Studi (PRS 3), dimana mahasiswa dapat membatalkan mata kuliah yang sudah dipilihnya.

Namun pada proses skripsi ini, hanya difokuskan untuk simulasi PRS 1 saja. Dimana pada awal mahasiswa melakukan pendaftaran mata kuliah dengan memilih mata kuliah yang diinginkan, kemudian melakukan proses validasi. Berikut pada Gambar 3.9 akan dijelaskan mengenai gambaran alur pendaftaran rencana studi yang pertama.



Gambar 3.9 Activity diagram Pendaftaran PRS 1 online

Proses pendaftaran rencana studi tidak hanya melibatkan mahasiswa, namun juga para dosen wali. Oleh karena itu waktu yang bersamaan untuk proses pengaksesan sistem PRS melibatkan banyak *user* sehingga membuat sistem SIM menjadi lambat dan proses *login* menjadi terhalang. Dengan kebiasaan mahasiswa yang memilih waktu *login* secara bersama-sama yang tidak dapat dihindari membuat *server* mengalami *overload*. Ditambah banyak mahasiswa yang sering merasa tidak yakin akan PRS online, membuat mereka harus melakukan login berulang-ulang untuk memastikan mata kuliah yang mereka ambil sebelum melakukan validasi. Banyaknya daftar mata kuliah serta jurusan yang ada, bisa menjadi salah satu kemungkinan pengaksesan database memakan waktu yang cukup lama juga. Akibat yang dihasilkan, banyak dari mahasiswa yang tidak dapat secara langsung melakukan pendaftaran, namun harus menunggu waktu tertentu dan hari yang khusus.

Pembagian waktu ini dirasa kurang fleksibel, dimana para mahasiswa harus memperhatikan jadwal-jadwal tertentu yang sudah dibagi. Hal ini dapat membuat mahasiswa terkadang lalai dan harus mengingat kapan mereka dapat melakukan pendaftaran. Dan dirasa susah untuk membagi jadwal secara adil dan merata kepada tiap mahasiswa.

3.5.2. Desain Web Service

Service yang akan diimplementasikan berupa aplikasi PRS online. Berikut merupakan tampilan PRS *online* secara keseluruhan pada Gambar 3.10 di bawah ini.

Pendaftaran Rencana Studi Mahasiswa (PRS 1)
Semester Genap 2018/2019

Nama : _____
NRP : 26415xxx
Dosen Wali : _____

Fakultas : TEKNOLOGI INDUSTRI
Jurusan : INFORMATIKA
SKS Lulus : 144 IPK : 4.00

Anda Belum VALIDASI

Semester : Semua Semester
Mata Kuliah : -- Pilih Mata Kuliah --
Pilihan 1 : -- Kelas untuk Mata Kuliah ini tidak terprogram --
Pilihan 2 : -- Kelas untuk Mata Kuliah ini tidak terprogram --
Pilihan 3 : -- Kelas untuk Mata Kuliah ini tidak terprogram --

No	Kode	Mata Kuliah	Kelas	SKS	Jadwal	
PRS 1						
2	TF4442 *	TEKNOLOGI VIRTUALISASI	A	3	Selasa, 13:30 - 16:30	[delete]
3	TF4442 *	TEKNOLOGI VIRTUALISASI	B	3	Rabu, 13:30 - 16:30	[delete]
1	TF4999 *	TUGAS AKHIR	A	6	Sabtu, 16:30 - 22:30	[delete]

Jumlah SKS : 12
*) Merupakan mata kuliah prioritas (maksimal 20 sks). Untuk mengambil mata kuliah prioritas, pilih pilihan 1. Bila anda ingin mengganti prioritas, hapus pilihan 1 mata kuliah prioritas yang lama dan tambahkan mata kuliah prioritas yang baru.

Keterangan
Jadwal mata kuliah jurusan
Jadwal PRS I: s.d.
Agenda tidak mengikuti PRS 1 diizinkan mengikuti PRS 2 dan dilaksanakan sesuai ketentuan yang berlaku.
Jadwal PRS II: s.d.
Jadwal PRS III: s.d.

Keterangan Status :
F : Pendaftaran diterima
F : Diterima di kelas lain
PS : Tolak karena prasyarat sedang kuliah
PK : Tolak karena prasyarat sudah kuliah
PL : Tolak karena prasyarat lulus kuliah
PD : Tolak karena prasyarat kredit lulus/Tolak karena prasyarat kredit lulus
SK : Tolak karena seleksi kelas
BK : Jadwal bentrok

Gambar 3.10 Web service PRS online

Dimana terdapat berbagai jurusan untuk dipilih, juga mata kuliah yang akan diambil, serta pilihan kelas dan waktu kelas tersebut diadakan. Berikut pada Gambar 3.11 merupakan gambar *web service* untuk pemilihan mata kuliah pada PRS online.

Semester : Semua Semester
Mata Kuliah : -- Pilih Mata Kuliah --
Pilihan 1 : -- Kelas untuk Mata Kuliah ini tidak terprogram --
Pilihan 2 : -- Kelas untuk Mata Kuliah ini tidak terprogram --
Pilihan 3 : -- Kelas untuk Mata Kuliah ini tidak terprogram --

Tambah

Gambar 3.11 Pemilihan mata kuliah untuk PRS

Kemudian akan muncul daftar pilihan mata kuliah yang telah dipilih oleh *user* pada halaman tersebut. Dimana terdapat kode mata kuliah, nama mata kuliah, kelas, jumlah sks, serta jadwal kelas yang tersedia untuk mata kuliah tersebut. Berikut contoh tampilan mata kuliah yang telah dipilih terlihat pada Gambar 3.12.

No	Kode	Mata Kuliah	Kelas	SKS	Jadwal	
PRS 1						
2	TF4442 *	TEKNOLOGI VIRTUALISASI	A	3	Selasa, 13:30 - 16:30	[delete]
3	TF4442 *	TEKNOLOGI VIRTUALISASI	B	3	Rabu, 13:30 - 16:30	[delete]
1	TF4999 *	TUGAS AKHIR	A	6	Sabtu, 16:30 - 22:30	[delete]
Validasi						

Gambar 3.12 Daftar mata kuliah yang dipilih

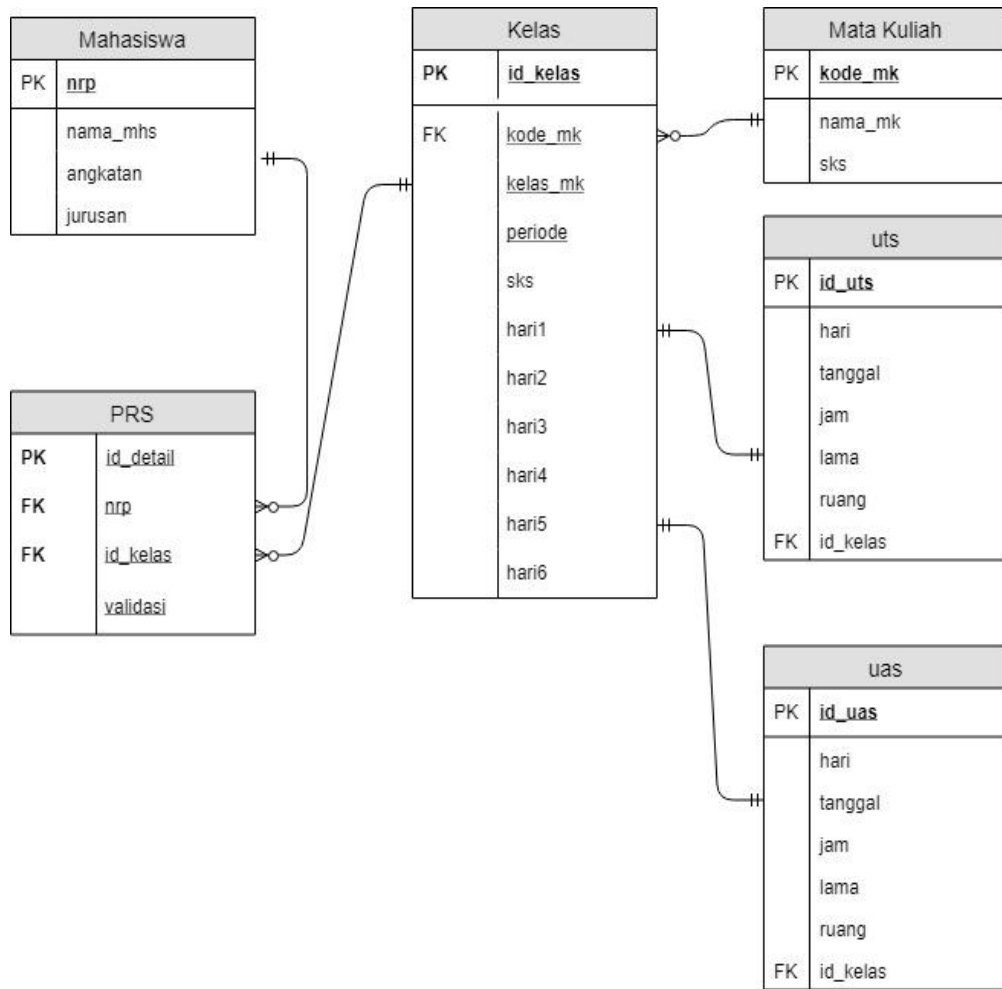
Ketika *user* sudah yakin dengan pilihan mereka, maka langkah selanjutnya yang dilakukan oleh *user* ialah menekan tombol validasi. Ketika data sudah terproses, *user* dapat menunggu proses pengumuman PRS selama kurang lebih satu minggu.

Mahasiswa dapat memastikan pilihan mata kuliahnya dengan melihat jadwal UTS dan UAS dari mata kuliah tersebut. Pada Gambar 3.13 ditunjukkan contoh jadwal UTS dan UAS dari semua mata kuliah yang ada, mahasiswa dapat memastikan apakah jadwal ujian tidak bentrok dari antara pilihan mata kuliah yang diambil.

Jam	Lama	Ruang	Mata Kuliah	Kelas
07:30	120	P.617	INTERNET OF EVERYTHING	D
08:30	120	P.614	BAHASA INGGRIS	E
09:30	120	LAB.INFOR	PEMROGRAMAN BERORIENTASI OBYEK	D
13:30	120	I.208	BAHASA INGGRIS	A
13:30	120	P.708	BAHASA INGGRIS	C
13:30	180	I.308B	MATEMATIKA DASAR	A
13:30	180	LAB.INFOR	TEKNOLOGI OPEN SOURCE	A
13:30	180	LAB.INFOR	TEKNOLOGI WEB	B
16:30	180	P.612	MATEMATIKA DISKRIT	C

Gambar 3.13 Jadwal UTS dan UAS dari mata kuliah

3.5.3. Database Service



Gambar 3.14 ERD PRS online

Berdasarkan *Entity Relationship Diagram (ERD)* yang telah didesain pada Gambar 3.14, dibuat sebuah *database* yang terdiri dari tabel-tabel berikut :

Tabel 3.1 Mata Kuliah

Nama Field	Keys	Tipe Data	Deskripsi
kode_mk	<i>Primary Key</i>	<i>text</i>	Identifier kode mata kuliah
nama_mk		<i>text</i>	Identifier nama mata kuliah
sks		<i>integer</i>	Identifer jumlah sks tiap mata kuliah

Tabel 3.2 Kelas

Nama <i>Field</i>	<i>Keys</i>	Tipe Data	Deskripsi
id_kelas	<i>Primary Key</i>	<i>integer</i>	Identifier kode kelas
periode		<i>text</i>	Identifier periode yang sedang berlangsung
kode_mk	<i>Foreign Key</i>	<i>text</i>	Identifier kode mata kuliah
kelas_mk	<i>Foreign Key</i>	<i>text</i>	Identifer kelas mata kuliah
sks		<i>integer</i>	Identifier jumlah sks tiap mata kuliah
hari1		<i>text</i>	Identifier hari pertama dari mata kuliah yang bersangkutan
hari2		<i>text</i>	Identifier hari kedua dari mata kuliah yang bersangkutan
hari3		<i>text</i>	Identifier hari ketiga dari mata kuliah yang bersangkutan
hari4		<i>text</i>	Identifier hari keempat dari mata kuliah yang bersangkutan
hari5		<i>text</i>	Identifier hari kelima dari mata kuliah yang bersangkutan
hari6		<i>text</i>	Identifier hari keenam dari mata kuliah yang bersangkutan

Tabel 3.3 Mahasiswa

Nama <i>Field</i>	<i>Keys</i>	Tipe Data	Deskripsi
nrp	<i>Primary Key</i>	<i>text</i>	Identifier dari id mahasiswa
nama_mhs		<i>text</i>	Identifier dari nama mahasiswa
angkatan		<i>integer</i>	Identifer dari angkatan mahasiswa yang mengambil mata kuliah

jurusan		<i>text</i>	Identifier dari jurusan yang diambil oleh mahasiswa
password		<i>text</i>	Identifier password tiap mahasiswa

Tabel 3.4 PRS

Nama <i>Field</i>	<i>Keys</i>	Tipe Data	Deskripsi
id_detail	<i>Primary Key</i>	<i>serial</i>	Identifier id_detail
nrp	<i>Foreign Key</i>	<i>text</i>	Identifier dari id mahasiswa
id_kelas	<i>Foreign Key</i>	<i>integer</i>	Identifier id kelas
validasi		<i>integer</i>	Identifier status validasi

Tabel 3.5 UTS

Nama <i>Field</i>	<i>Keys</i>	Tipe Data	Deskripsi
id_uts	<i>Primary Key</i>	<i>serial</i>	Identifier dari id UTS
hari		<i>text</i>	Identifier dari hari UTS berlangsung
tanggal		<i>date</i>	Identifier dari tanggal UTS berlangsung
jam		<i>time</i>	Identifier dari tanggal UTS berlangsung
lama		<i>integer</i>	Identifier dari lama durasi UTS berlangsung
ruang		<i>text</i>	Identifier dari ruang UTS
id_kelas	<i>Foreign Key</i>	<i>integer</i>	Identifier dari id_kelas

Tabel 3.6 UAS

Nama <i>Field</i>	<i>Keys</i>	Tipe Data	Deskripsi
id_uas	<i>Primary Key</i>	<i>serial</i>	Identifier dari id UAS
hari		<i>text</i>	Identifier dari hari UAS

			berlangsung
tanggal		<i>date</i>	Identifier dari tanggal UAS berlangsung
jam		<i>time</i>	Identifier dari tanggal UAS berlangsung
lama		<i>integer</i>	Identifier dari lama durasi UAS berlangsung
ruang		<i>text</i>	Identifier dari ruang UAS
id_kelas		<i>integer</i>	Identifier dari id_kelas