APLIKASI KEHADIRAN MAHASISWA MENGGUNAKAN *BARCODE* BERBASIS *ANDROID* STUDI KASUS STMIK PALANGKARAYA

TUGAS AKHIR

Disusun Sebagai Salah Satu Syarat Kelulusan Program Strata I pada
Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangka Raya

OLEH :

DWI ARIANTO AKBAR
NIM C1455201022
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKA RAYA
2018

APLIKASI KEHADIRAN MAHASISWA MENGGUNAKAN *BARCODE* BERBASIS *ANDROID* STUDI KASUS STMIK PALANGKARAYA

TUGAS AKHIR

Disusun Sebagai Salah Satu Syarat Kelulusan Program Strata I pada
Sekolah Tinggi Manajemen Informatika dan Komputer
(STMIK) Palangka Raya

OLEH

DWI ARIANTO AKBAR
NIM C1455201022
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKA RAYA
2018

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama Mahasiswa : Dwi Arianto Akbar
NIM : C1455201022

menyatakan bahwa tugas akhir dengan judul :

APLIKASI KEHADIRAN MAHASISWA MENGGUNAKAN *BARCODE* BERBASIS *ANDROID* STUDI KASUS STMIK PALANGKARAYA

Adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain, kecuali bagian yang sumber informasi dicantumkan.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggung jawab dan saya bersedia menerima sanksi pembatalan skripsi apabila terbukti melakukan duplikasi terhadap tugas akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, Desember 2018
Yang Membuat Pernyataan,

Matrei
Rp. 6000

Dwi Arianto Akbar

PERSETUJUAN

APLIKASI KEHADIRAN MAHASISWA MENGGUNAKAN *BARCODE* BERBASIS *ANDROID* STUDI KASUS STMIK PALANGKARAYA

Tugas akhir ini telah disetujui untuk diujikan
pada Tanggal 28 November 2018

Pembimbing I,	Pembimbing II,
Sulistyowati, S.Kom., M.Cs	Lili Rusdiana, M.Kom
NIK. 198212162007002	NIK. 198707282011007

Mengetahui
Ketua STMIK Palangkaraya,

Suparno, M.Kom.
NIK. 196901041995105

PENGESAHAN

APLIKASI KEHADIRAN MAHASISWA MENGGUNAKAN *BARCODE* BERBASIS *ANDROID* STUDI KASUS STMIK PALANGKARAYA

Tugas Akhir ini telah Diujikan, Dinilai dan Disahkan
Oleh Tim Penguji pada Tanggal 1 Desember 2018

Tim Penguji Tugas Akhir :

1. Heri Setiawan, M.Kom .....
   Ketua
2. Rudini, M.Pd .....
   Sekretaris
3. Hotmian Sitohang, M.Kom .....
   Anggota
4. Sulistyowati, S.Kom, M.Cs .....
   Anggota
5. Lili Rusdiana, M.Kom .....
   Anggota

MOTO DAN PERSEMBAHAN

"Jika kau merasa hidup ini sulit,
lihatlah betapa kerasnya orang tua
bekerja keras untuk dirimu"

Tugas Akhir ini saya persembahkan untuk:

- *Allah SWT. yang telah memberikan kelancaran dalam pengerjaan program.*
- *Orang tua dan kaka saya.*
- *Bapak ibu dosen STMIK Palangka Raya yang senantiasa ikhlas dalam memberikan ilmu-ilmu baru untuk Saya.*
- *Teman-teman kantin STMIK yang selalu memberikan dukungan maupun motivasi.*
- *Serta, teman-teman dan sahabat-sahabat Saya.*

INTISARI

Dwi Arianto Akbar, C1455201022, 2018. *Aplikasi Kehadiran Mahasiswa Menggunakan Barcode Berbasis Android Studi Kasus STMIK Palangkaraya*, Pembimbing I Sulistyowati, S.Kom, M.Cs., Pembimbing II Lili Rusdiana, M.Kom.

STMIK Palangkaraya memiliki kegiatan perkuliahan yang dilakukan oleh dosen dan mahasiswa, setiap kegiatan perkuliahan tercatat dalam lembar kehadiran. Sebelum mengajar dosen ke akma untuk mengambil lembar kehadiran dan mahasiswa melakukan tanda tangan sebagai tanda mahasiswa hadir pada perkuliahan, serta pihak akma harus menghitung ulang data kehadiran untuk melakukan perekapan dan hal itu akan memperlambat proses rekap data kehadiran. Dengan adanya aplikasi ini kecurangan pada proses pencatatan kehadiran mahasiswa dapat diatasi dan memanfaatkan pengolah data digital agar perekapan data kehadiran lebih mudah.

Metode pengumpulan data yang digunakan yaitu observasi, wawancara, pustaka, dokumentasi dan kuisioner. Model pengembangan yang dilakukan yaitu *prototype*, serta metode analisis sistem menggunakan PIECES (*performance, information, economy, control, eficiency dan service*).

Hasil dari penelitian ini yaitu aplikasi kehadiran mahasiswa menggunakan *barcode* berbasis *android*, yang mampu membantu proses kehadiran mahasiswa menjadi lebih mudah, dosen tidak perlu lagi ke akma untuk mengambil lembar kehadiran serta membantu akma dalam perekapan kehadiran. Aplikasi ini sudah diuji menggunakan metode *black box testing* dan semua komponen dapat berjalan dengan baik. Berdasarkan hasil kuisioner diperoleh nilai 85,3% yang dapat disimpulkan bahwa hasil dari penelitian ini layak untuk digunakan. Maka dari itu dengan memanfaatkan sistem yang terkomputerisasi proses kehadiran dan perekapan data kehadiran dapat dilakukan dengan mudah.

Kata Kunci : Aplikasi, *Android, Barcode*, Kehadiran, *Protoype*, STMIK Palangkaaya.

ABSTRACT

Dwi Arianto Akbar, C1455201022, 2018. *Application The Presence Of Students Use Barcode Based Android Case Study STMIK Palangkaraya,* Advisior I Sulistyowati, S.Kom, M.Cs., Advisior II Lili Rusdiana, M.Kom.

STMIK Palangkaraya having often these institutions granted degrees activities that is carried out by lecturers and students , each activity increasing the number of issuers often these institutions granted degrees in a sheet the presence of .Before teaching he got from his lecturers to akma to take sheets of the presence of and what the students are doing we need you to sign as a sign of what the students are present at often these institutions granted degrees , as well as parties akma shall compute the amount of data they would have to re preferable to have full time to perform recap and that could certainly slowing things down data the presence of followed.

Data collection method used namely observation , interview , library , documentation and of the questionnaire was .Which is executed is the prototype of model of development , as well as the method of analysis system using PIECES (performance, information, economy, control, eficiency dan service).

The result of this is the research application the presence of students use barcode based android case study stmik palangkaraya , which was able to help the process the presence of kids to be easier , he got from his lecturers will no longer need to to akma to take sheets of the presence of akma perekapan as well as help in the presence of .This application is tested uses the method of testing was definitely black box and all sides can work well .Based on the results of the questionnaire was obtained the value of 85.3 % that could be observed that the results of the study is worth used .That is why it is by making use of the system that the process of the presence of and perekapan computerized data the presence of can be conducted easily.

Kata Kunci : Application, Android, Barcode, Presence, Protoype, STMIK Palangkaaya.

KATA PENGANTAR

Dengan memanjatkan puji syukur kepada Allah SWT, atas berkat rahmat dan hidayah-Nya, sehingga penulis dapat menyelsaikan Tugas Akhir dengan judul "Aplikasi Kehadiran Mahasiswa Menggunakan Barcode Berbasis Android Studi Kasus STMIK Palangkaraya".

Tugas Akhir ini disusun untuk memenuhi salah satu syarat kelulusan Strata 1 Program Studi Teknik Informatika pada Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangka Raya.

Pada kesempatan ini penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu dalam penyelesaian skripsi ini terutama kepada:

1. Allah SWT yang telah memberikan kesehatan dan kekuatan kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir.
2. Suparno, M.Kom, selaku Kepala Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya.
3. Sulistyowati, S.Kom, M.Cs., selaku dosen pembimbing I yang telah memberikan bimbingan dalam materi penelitian.
4. Lili Rusdiana, M.Kom., selaku dosen pembimbing II yang telah memberikan bimbingan dalam materi penelitian.
5. Bapak dan Ibu dosen yang telah memberikan bekal ilmu pengetahuan kepada penulis selama di bangku perkuliahan.
6. Kedua orang tua dan kaka saya tercinta yang telah memberikan dukungan, doa, moril dan materil selama ini.

7. Teman-teman kantin STMIK Palangkaraya yang memberikan bantuan berupa dukungan ataupun pemikiran selama penyusunan tugas akhir.
8. Teman-teman Teknik Informatika angkatan 2014 yang telah memberikan bantuan berupa pemikiran selama penyusunan tugas akhir.
9. Semua pihak yang berjasa dalam menyelesaikan penyusunan tugas akhir ini.

Dalam penulisan tugas akhir ini penulis menyadari masih banyak kekurangan yang perlu disempurnakan. Untuk itu kritik dan saran senantiasa diharapkan demi kesempurnaan tugas akhir ini. Akhir kata, penulis ucapkan terima kasih dan semoga tugas akhir ini bermanfaat bagi pembaca.

DAFTAR ISI

HALAMAN SAMPUL DALAM i
LEMBAR PERNYATAAN ii
PERSETUJUAN iii
PENGESAHAN iv
MOTO DAN PERSEMBAHAN v
INTISARI vi
ABSTRACT vii
KATA PENGANTAR viii
DAFTAR ISI x
DAFTAR TABEL xii
DAFTAR GAMBAR xiii

BAB I PENDAHULUAN
A.Latar Belakang Masalah 1
B.Rumusan Masalah 3
C.Batasan Masalah 3
D.Tujuan dan Manfaat 4
1.Tujuan 4
2.Manfaat 4
E.Metode Penelitian 5
F.Sistematika Penulisan 8

BAB II LANDASAN TEORI
A.Kajian Penelitian yang Relevan 10
B.Kajian Teori 12
C.Perangkat yang Digunakan 23
1.*Android Studio* 23
2.*Sublime Text* 23
3.*Xampp* 24

BAB III ANALISIS DAN DESAIN SISTEM
A.Tinjauan Umum 25
B.Analisis 25
1.Anlisis Kelemahan Sistem 26
2.Analisis Kebutuhan Sistem 28
3.Analisis Kelayakan Sistem 30
C.Desain Sistem 31
1.Desain Input 31
2.Desain Proses 34
3.Desain Keluaran (*Output*) 38
4.Rancangan Desain *Interface*/Antarmuka 38
5.Penggunaan Metode Perancangan Sistem 50
6.Desain Basis Data 69

7.Desain Kemanan 75
8.Desain Sumber Daya Manusia (SDM) 76

BAB IV IMPLEMENTASI DAN PEMBAHASAN
A.Implementasi 77
1.Uji Coba Program 77
2.Manual Program 85
3.Manual Instalasi 101
B.Hasil Penelitian dan Pembahasan 105
1.Pembahasan Listing Program 105
2.Pembahasan Basis Data 128
3.Pembahasan Hasil Response Pengguna 130

BAB V PENUTUP
A. Kesimpulan 133
B. Saran 133

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 1. Simbol *Use case* Diagram 16
Tabel 2.Simbol *State Machine* Diagram 19
Tabel 3. *Activity* Diagram 20
Tabel 4. Simbol-Simbol ERD 21
Tabel 5. Analisa PIECES Sistem Lama dan Baru 26
Tabel 6. Desain *Input Login* Dosen 31
Tabel 7. Desain *Input Login* Mahasiswa 31
Tabel 8. Desain *Input Login* Admin 32
Tabel 9. Desain *Input* Data Mahasiswa 32
Tabel 10. Desain *Input* Data Dosen 33
Tabel 11. Desain *Input* Data Matakuliah 33
Tabel 12. Desain *Input* Data KRS Mahasiswa 34
Tabel 13. Desain *Input* Data Mahasiswa Pada Matakuliah 34
Tabel 14. Desain Basis Data Admin 71
Tabel 15. Desain Basis Data Dosen 71
Tabel 16. Desain Basis Data Mahasiswa 72
Tabel 17. Desain Basis Data Matakuliah 73
Tabel 18. Desain Basis Data KRS 74
Tabel 19. Desain Basis Data Kehadiran 75
Tabel 20. Rencana Pengujian *Black Box* 77
Tabel 21. Pengujian Terhadap Mahasiswa 79
Tabel 22. Pengujian Halaman Login 79
Tabel 23. Pengujian Halaman Utama Dosen 80
Tabel 24. Pengujian Halaman Utama Mahasiswa 81
Tabel 25. Pengujian Halaman Mulai Mengajar 81
Tabel 26. Pengujian Halaman Daftar Matakuliah Dosen 82
Tabel 27. Pengujian Halaman Tampilkan *QR Code* 83
Tabel 28. Pengujian Halaman Daftar Matakuliah Mahasiswa 83
Tabel 29. Pengujian Keluar Akun Pengguna Dosen dan Mahasiswa 83
Tabel 30. Pengujian Mengirim Notifikasi Ke Mahasiswa 84
Tabel 31. Pengujian Melihat Laporan Kehadiran Mahasiswa 84
Tabel 32. Pengujian Melihat Histori Mengajar 85
Tabel 33. Hasil Responden Pengguna 130
Tabel 34. Keterangan Bobot Penilaian 131
Tabel 35. Hasil Kuisioner 131

DAFTAR GAMBAR

Gambar 1. *Prototype* 6
Gambar 2. *Activity Login* 39
Gambar 3. *Activity* Utama Dosen 39
Gambar 4. *Activity* Mulai Mengajar 40
Gambar 5. *Activity* Daftar Matakuliah Dosen 40
Gambar 6. *Activity* Pengaturan Dosen 41
Gambar 7. *Activity Login* Mahasiswa 42
Gambar 8. *Activity* Utama Mahasiswa 42
Gambar 9. *Activity* Tampilkan *QR Code* 43
Gambar 10. *Activity* Daftar Matakuliah Mahasiswa 44
Gambar 11. *Activity* Pengaturan Mahasiswa 44
Gambar 12. Halaman *Login* Admin 45
Gambar 13. Halaman Utama Admin 46
Gambar 14. Halaman Atur Data KRS 46
Gambar 15. Halaman Tambah Data KRS 47
Gambar 16. Halaman Lihat Nama Mahasiswa 48
Gambar 17. Halaman Tampil Data Kehadiran 48
Gambar 18. Halaman Laporan Kehadiran 49
Gambar 19. Halaman Kirim Pemberitahuan 50
Gambar 20. *Use case* Diagram Admin 51
Gambar 21. *Use case* Diagram Dosen dan Mahasiswa 51
Gambar 22. *Activity* Diagram *Login* 52
Gambar 23. *Activity* Diagram Kelola Data Mahasiswa 53
Gambar 24. *Activity* Diagram Kelola Data Dosen 54
Gambar 25. *Activity* Diagram Kelola Data Matakuliah 55
Gambar 26. *Activity* Diagram Kelola Data KRS 56
Gambar 27. *Activity* Diagram Data Kehadiran 57
Gambar 28. *Activity* Diagram *Login* Dosen 58
Gambar 29. *Activity* Diagram Mulai Mengajar 59
Gambar 30. *Activity* Diagram Daftar Matakuliah Dosen 60
Gambar 31. *Activity* Diagram Pengaturan Dosen 61
Gambar 32. *Activity* Diagram *Login* Mahasiswa 62
Gambar 33. *Activity* Diagram Tampilkan *QR Code* Mahasiswa 63
Gambar 34. *Activity* Diagram Daftar Matakuliah Mahasiswa 64
Gambar 35. *Activity* Diagram Pengaturan Mahasiswa 65
Gambar 36. *State Machine* Diagram Admin 66
Gambar 37. *State Machine* Diagram Dosen 67
Gambar 38. *State Machine* Diagram Mahasiswa 68
Gambar 39. ERD 70
Gambar 40. Tampilan Login Mahasiswa 86
Gambar 41. Tampilan Menu Utama Mahasiswa 86
Gambar 42. Tampilan Menu Tampilkan QR Code 87

Gambar 43. Tampilan Menu Daftar Matakuliah 88
Gambar 44. Tampilan Toast Daftar Hadir 88
Gambar 45. Tampilan Menu Pengaturan 89
Gambar 46. Tampilan Menu Pengaturan 89
Gambar 47. Tampilan Login Dosen 90
Gambar 48. Tampilan Menu Utama Dosen 91
Gambar 49. Tampilan Menu Mulai Mengajar 91
Gambar 50. Tampilan Matakuliah Mulai Mengajar 92
Gambar 51. Tampilan Menu Daftar Matakuliah 93
Gambar 52. Tampilan Menu Pengaturan 93
Gambar 53. Tampilan Menu Pengaturan 94
Gambar 54. Halaman Login Admin 94
Gambar 55. Halaman Dashboard Admin 95
Gambar 56. Halaman Atur KRS 96
Gambar 57. *Form* Tambah KRS 96
Gambar 58. Halaman Mahasiswa Pada KRS 97
Gambar 59. *Form* Mahasiswa Pada KRS 97
Gambar 60. Halaman History Mengajar 98
Gambar 61. Laporan Rekap Absensi Mengajar 99
Gambar 62. Data Kehadiran Mahasiswa 99
Gambar 63. Pilih Matakuliah Mahasiswa 100
Gambar 64. Lihat Total 101
Gambar 65. Tombol Kirim Pesan 101
Gambar 66. Buka *Android Studio* 102
Gambar 67. *Project* Android Terbuka 102
Gambar 68. Tekan *Build Project* 103
Gambar 69. *Generate Signed* APK 103
Gambar 70. *Build Type* APK 104
Gambar 71. Lokasi File APK 104
Gambar 72. Struktur Tabel Admin 128
Gambar 73. Struktur Tabel Dosen 128
Gambar 74. Struktur Tabel Mahasiswa 129
Gambar 75. Struktur Tabel Matakuliah 129
Gambar 76. Struktur Tabel KRS 129
Gambar 77. Struktur Tabel Kehadiran 130

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Kegiatan perkuliahan dapat berlangsung dengan baik dengan adanya keaktifan mahasiswa dalam menghadiri setiap perkuliahan yang telah dijadwalkan. Pentingnya kehadiran mahasiswa dalam menghadiri perkuliahan, maka dibutuhkan sistem yang baik mengenai absensi kehadiran mahasiswa. Saat ini proses kehadiran mahasiswa masih menggunakan tanda tangan pada lembar kehadiran yang dibagikan setiap menghadiri perkuliahan, metode ini memiliki kelemahan yaitu memungkinan mahasiswa untuk melakukan titip tanda tangan padahal mahasiswa yang bersangkutan tidak hadir pada perkuliahan. Sedangkan apabila absensi dilakukan dengan alat sidik jari akan terkendala masalah biaya alat tersebut, karena harga satu unit alat absensi sidik jari bisa mencapai sekitar 1 juta hingga 3 juta rupiah jika dilihat di toko-toko *online*.

Teknologi *QR barcode* dapat dijadikan solusi untuk masalah harga alat sidik jari yang cukup mahal, karena angka dan huruf bisa dijadikan *Code QR*. Nomor induk mahasiswa (NIM) dapat dijadikan identitas ketika melakukan proses kehadiran pada perkuliahan, ketika NIM telah dirubah ke *Code QR* kode tersebut akan diarahkan ke kamera *smartphone* dosen untuk dilakukan proses *scan* dan status mahasiswa akan dinyatakan hadir oleh sistem pada kegiatan perkuliahan.

Setiap mahasiswa akan memiliki kartu kehadiran yang bentuk dan ukuran nya sama dengan kartu tanda penduduk (KTP), yang berisi biodata mahasiswa serta gambar *Code QR.* Kartu kehadiran mahasiswa harus selalu dibawa oleh mahasiswa ketika menghadiri kegiatan perkuliahan, karena kartu tersebut akan digunakan untuk di *scan* melalui kamera ponsel dosen. Setiap ruangan perkuliahan akan diberikan *tripod/phone holder* untuk tempat meletakkan ponsel dosen. Ketika perkuliahan dimulai oleh dosen mahasiswa hanya perlu maju kedepan dan mengarahkan kartu kehadiran kearah kamera ponsel dosen dan secara otomatis sistem akan menyatakan mahasiswa tersebut hadir pada perkuliahan.

Aplikasi memungkinkan mahasiswa tidak dapat melakukan kecurangan pada status kehadiran mereka pada perkuliahan. Aplikasi ini dibangun dengan basis *android*, hampir setiap perangkat *smartphone* yang beredar dipasaran menggunakan sistem operasi *android*. Hampir semua mahasiswa maupun dosen menggunakan smartphone bersistem operasi *android*.

Bagian AKMA akan terbantu dengan adanya aplikasi ini, karena otomatis akan mendapatkan hasil rekapan total kehadiran mahasiswa dan kehadiran dosen ketika mengajar mata kuliah dalam 1 semester. Oleh karena itu penulis membuat aplikasi kehadiran mahasiswa berbasis *android*, karena *android* merupakan aplikasi *free opensource* yang memudahkan pembuat aplikasi dalam merancang ataupun membangun sebuah aplikasi. Karena permasalahan diatas maka, penulis merasa tertarik untuk merancang serta membahas tentang “Aplikasi Kehadiran Mahasiswa Menggunakan *Barcode*

Berbasis *Android*" yang dapat membuat proses perkuliahan menjadi aman dari kecurangan presensi kehadiran.

B. Rumusan Masalah

Penulis dapat menarik sebuah rumusan masalah dari latar belakang di atas, yaitu "Bagaimana membuat aplikasi kehadiran mahasiswa mengunakan *barcode* berbasis *android* sehingga dapat mengurangi kecurangan pada kehadiran?".

C. Batasan Masalah

Adapun batasan masalah yang dapat penulis sebutkan sebagai pedoman agar penelitian ini tidak melebar luas hingga keluar topik, maka batasan masalah yang akan dibahas adalah sebagai berikut :

1. Aplikasi kehadiran ini dibuat untuk mengurangi kecurangan pada kegiatan perkuliahan.
2. Dalam aplikasi yang dijadikan gambar *QR Code* adalah nomor induk mahasiswa (NIM), karena NIM memiliki kode angka yang berbeda-beda dengan NIM yang lainnya.
3. Untuk dapat menjalankan aplikasi ini, sistem operasi yang digunakan minimal menggunakan versi *Kitkat* (*K*).
4. Aplikasi ini bersifat *online*.
5. Web *backend* yang dibuat oleh penulis hanya untuk membantu proses implementasi program *android*.
6. Mahasiswa yang kehadirannya kurang dari 75% akan mendapatkan pesan pemberitahuan melalui aplikasi kehadiran yang dikirim oleh admin agar

mahasiswa tersebut menghubungi dosen yang bersangkutan agar dapat mengganti kehadiran yang kurang.

7. Kehadiran mahasiswa dan dosen akan direkap secara otomatis oleh sistem pada akhir semester.
8. Kode *barcode* yang digunakan penulis adalah kode *QR* dan module dlazaro66 sebagai *qr code reader view.*

D. Tujuan dan Manfaat

1. Tujuan

   Adapun tujuan penulis adalah membangun sebuah aplikasi untuk mendukung kegiatan perkuliahan menjadi lebih baik dan mengurangi kecurangan pada kehadiran mahasiswa.

2. Manfaat

   a. Bagi Pengguna

   Adapun manfaat bagi pengguna yaitu dapat membuat mahasiswa menjadi lebih rajin untuk menghadiri perkuliahan dan dosen tidak perlu lagi ke akma untuk mengambil lembar kehadiran mahasiswa.

   b. Bagi Penulis

   Sebagai penerapan ilmu yang telah didapatkan selama masa kuliah dan membangun Aplikasi Kehadiran Mahasiswa Menggunakan Barcode Berbasis *Android.*

   c. Bagi STMIK Palangkaraya

   Penelitian ini dapat memperbaharui sistem kehadiran STMIK yang dulu menjadi lebih baik dan memanfaatkan teknologi yang sudah ada.

E. Metode Penelitian

Secara umum metode yang digunakan dalam menyusun proposal tugas akhir ini penulis menggunakan beberapa tahapan metode penelitian, antara lain :

1. Tahap Pengumpulan Data
    a. Metode Observasi

        Merupakan metode pengumpulan data dengan cara terjun langsung kelapangan untuk mendapatkan informasi yang dibutuhkan. Dalam hal ini penulis mencari data yang akurat dengan melihat kegiatan perkuliahan selama ini yang sering terjadi adalah kecurangan titip tanda tangan kehadiran.

    b. Metode Wawancara

        Merupakan metode pengumpulan data yang dilakukan dengan jalan wawancara, yaitu mendapatkan informasi lebih dengan bertanya secara langsung kepada beberapa dosen dan pegawai sehingga mendapatkan hasil yang lebih akurat.

    c. Studi Literatur

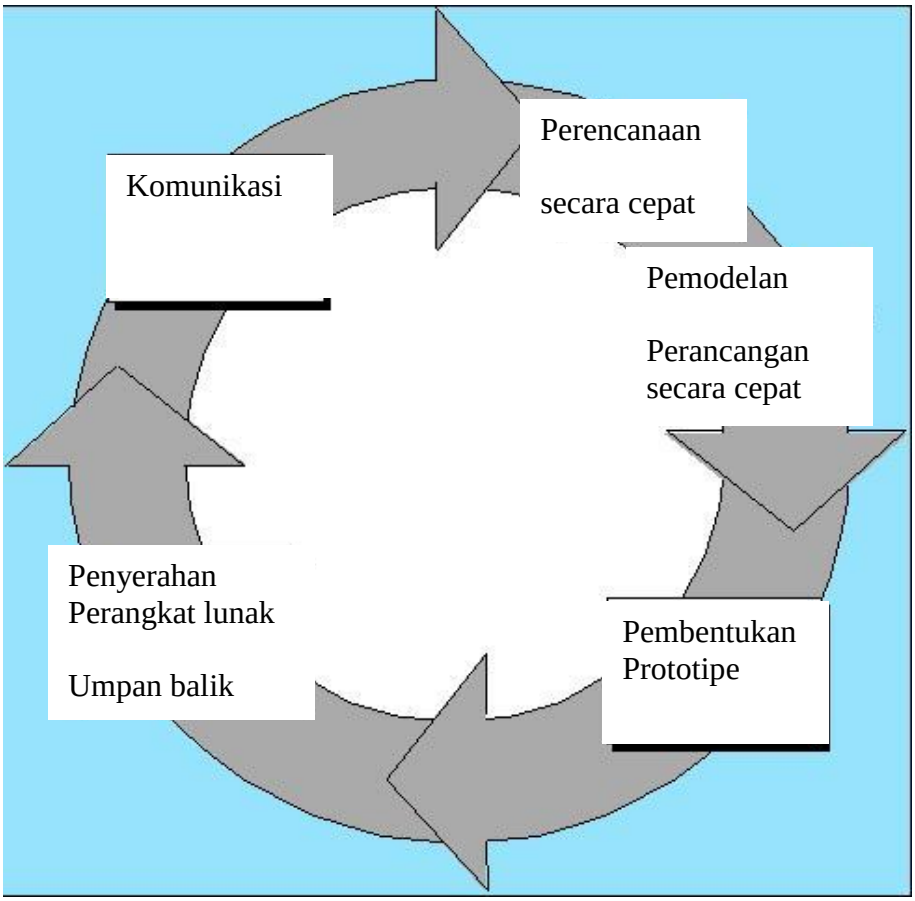
        Pengumpulan data dengan cara mengumpulkan literatur, jurnal, browsing internet dan bacaan yang ada kaitannya dengan topik penelitian baik berupa *textbook* atau *paper*.

    d. Kepustakaan

        Metode ini adalah metode pengumpulan data dengan mempelajari beberapa buku yang berhubungan dengan topik aplikasi kehadiran

ataupun dokumen–dokumen yang berhubungan dengan permasalahan yang dihadapi.

2. Tahap Pengembangan Perangkat Lunak



Gambar 1. *Prototype*

(sumber : Pressman, Roger. *Rekayasa Perangkat Lunak*, 2012)

Dalam penyusunan laporan proposal skripsi ini penulis menggunakan metode *prototype* untuk tahapan pengembangan perangkat lunaknya, seperti yang dapat dilihat pada gambar 1. Adapun beberapa tahapan diantaranya adalah :

1. Komunikasi

   Pada tahap ini penulis melakukan penelitian mengenai kebutuhan-kebutuhan yang diperlukan untuk membangun sistem nantinya dengan cara observasi dan wawancara ke STMIK Palangkaraya yaitu melihat kegiatan perkuliahan dan mengajukan pertanyaan kepada karyawan ataupun dosen agar mendapatkan data yang akurat.

2. Perancanaan secara cepat

   Pada tahap ini penulis melakukan perancanaan sistem dengan mengidentifikasi keseluruhan perangkat lunak ataupun perangkat keras yang akan digunakan untuk mengembangkan sistem. Perangkat lunak yang digunakan adalah *Android Studio*, *Sublime Text*, dan *Browser (Google Chrome/Mozila FireFox)* sedangkan perangkat keras yang digunakan adalah Laptop untuk melakukan pengodingan dan *Smartphone* untuk melakukan pengujian sistem nantinya.

3. Pemodelan

   Pada tahap ini penulis melakukan pemodelan alur sistem baru yang akan digunakan pada aplikasi nanti nya.

4. Perancangan secara cepat

   Pada tahap ini penulis merepresentasikan semua aspek perangkat lunak yang akan terlihat oleh pengguna, seperti rancangan antarmuka pengguna (*user interface*) atau format tampilan.

5. Pembentukan Prototipe

   Pada tahap ini penulis akan menerjemahkan *prototyping* yang sudah disepakati kedalam bahasa pemrograman *Android Native* yaitu *Java*.

6. Penyerahan sistem/perangkat lunak kepada pelanggan/pengguna

   Pada tahap ini Prototipe akan diserahkan kepada pengguna dan kemudian pengguna akan melakukan pengujian kelayakan sistem yang telah dibuat.

7. Umpan balik

   Ditahap ini pengguna akan memberikan umpan balik atau respon penilaian apakah sistem yang baru sudah sesuai dengan yang diharapkan atau tidak, jika masih belum sesuai maka akan dilakukan lagi ke proses pemodelan dan perancangan agar sistem bisa berjalan lebih baik.

F. Sistematika Penulisan

Untuk memudahkan dalam memahami laporan tugas akhir, penulis mengemukakan sistematika penulisan yang terdiri dari 5 (lima) bab, yaitu :

BAB I PENDAHULUAN

Berisi latar belakang yang menjelaskan tentang pentingnya penelitian yang dilakukan, latar belakang masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi yang dipakai selama penelitian, dan sistematika penulisan yang digunakan dalam penulisan laporan skripsi ini.

BAB II LANDASAN TEORI

Pada bab ini akan dijelaskan landasan teori – teori yang akan mendukung pada proses pembuatan sistem. Teori – teori tersebut diantaranya adalah sebagai berikut : Tinjauan pustaka tentang hasil penelitian orang lain dengan penelitian yang sedang diteliti, Pengertian tentang Aplikasi, Pengertian tentang Kehadiran, Pengertian tentang Barcode, Pengertian tentang android.

BAB III ANALISA DAN DESAIN SISTEM

Pada bab ini akan membahas tentang analisis terhadap permasalahan yang terdapat pada kasus yang sedang diteliti. Meliputi analisis terhadap masalah sistem yang sedang berjalan, analisis hasil solusinya, analisis kebutuhan terhadap sistem yang diusulkan, dan analisis kelayakan sistem yang diusulkan. Kemudian juga akan membahas tentang desain sistem yang akan di olah menjadi suatu aplikasi.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan diuraikan hasil penelitian beserta pembahasannya.

BAB V PENUTUP

Bab ini merupakan bagian terakhir dari laporan skripsi yang berisi kesimpulan dari uji coba dan saran – saran yang diambil dari kelemahan aplikasi untuk pengembangan lebih lanjut dari sistem yang telah dibuat oleh penulis.

BAB II

LANDASAN TEORI

A. Kajian Penelitian yang Relevan

Penelitian yang dilakukan Raka Onny Diar Danur, dkk, (2016). Penelitian ini menghasilkan aplikasi absensi berbasis platform android, metode yang digunakan adalah *prototype.* Untuk menggunakan fitur absensi mahasiswa, pertama-tama dosen harus login agar modul perkuliahan muncul dan mahasiswa bisa melakukan absensi dengan menekan tombol absen. Sedangkan penelitian yang dilakukan penulis mahasiswa hanya cukup mengarahkan kartu kehadiran atau gambar *QR Code* dari aplikasi ke kamera ponsel dosen agar dinyatakan hadir oleh sistem.

Selanjutnya penelitian yang dilakukan Qurotul, dkk, (2017). Penerapan penelitian absensi ini menggunakan web yang berbasis *Framework Yii*, metode yang digunakan adalah *prototype*. Karena abensi ini menggunakan web, akan ada penarikan data yang lumayan besar. Karena *resource* yang dipanggil berasal dari alamat hosting, tampilan akan bagus jika digunakan pada PC/Laptop dan perangkat yang memiliki layar besar. Sistem yang dibuat penulis lebih efektif, karena *resource* telah ada didalam file aplikasi ketika di instal pada *smartphone.*

Selanjutnya penelitian yang dilakukan Jashinta, dkk, (2017). Penelitian ini menggunakan *Reporting Service*, metode yang digunakan adalah *Rapid Application Development* (RAD). Aplikasi ini hanya memanfaatkan penggunaan *SQL Server Reporting Service* (SSRS) untuk

melakukan penyimpanan ataupun pengarsipan data. Sedangkan aplikasi yang dibuat penulis tidak hanya berfokus pada pengarsipan data kehadiran tapi juga pada proses pengambilan kehadiran itu sendiri.

Selanjutnya penelitian yang dilakukan Rachman Arief (2017). Penelitian ini memanfaatkan *Google Forms*, *Sheet*, *Sites*, *Awesome Tabl*e dan *Gmail*. Aplikasi ini berbasis web dan menggunakan tempat penyimpanan data dari google yang telah menggunakan *ajax*, tetapi untuk proses menyimpan kehadiran masih manual. Yaitu menggunakan seorang pengurus kelas untuk menginput data tiap kelas kedalam *form*, sedangkan aplikasi yang dibuat penulis proses penginputan data kehadiran akan otomatis ketika mahasiswa telah melakukan *scan* kode *barcode* pada kartu kehadiran mahasiswa.

Selanjutnya penelitian yang dilakukan Mohammad Aziz Fikri, dkk, (2017). Penelitian ini menghasilkan aplikasi kehadiran yang berbasis web dan sistem nya menggunakan *Barcode Scanning*, metode yang digunakan adalah *Prototype* dengan tahapan analisis. Proses penginputan kehadiran pada aplikasi ini menggunakan media kartu pegawai yang nantinya berisi gambar *barcode* dan akan di *scan* menggunakan alat *scanning*, Sedangkan penelitian yang dilakukan penulis berbasis *android* dan *scan* nya menggunakan kamera ponsel tanpa harus membeli alat *barcode scan*.

B. Kajian Teori

1. Aplikasi

Menurut Sutabri (2012), Aplikasi adalah alat terapanyang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya. Aplikasi sangat diperlukan dalam berbagai pengolahan data mengenai suatu informasi atau suatu kegiatan administrasi sebuah organisasi.

2. Kehadiran

Kehadiran memiliki makna yang sama dengan absensi yaitu suatu pendataan kehadiran yang merupakan bagian dari aktifitas pelaporan yang ada dalam sebuah institusi. Absensi disusun dan diatur sehingga mudah untuk dicari dan dipergunakan ketika diperlukan oleh pihak yang berkepentingan.

Secara umum, jenis-jenis absensi menurut cara penggunaannya dapat dikelompokkan menjadi dua, yaitu :

a) Absensi Manual, yang merupakan cara penulisan kehadiran dengan cara menggunakan pena untuk melakukan tanda tangan.
b) Absensi non manual, yang merupakan cara penulisan kehadiran dengan alat yang terkomputerisasi, bisa menggunakan RFID ataupun *fingerpint*, dll.

3. Mahasiswa

Menurut Yusuf (2012) seorang mahasiswa dikategorikan pada tahap perkembangan yang usianya 18 sampai 25 tahun. Tahap ini dapat digolongkan pada masa remaja akhir sampai dewasa awal dan dilihat dari segi perkembangan, tugas perkembangan pada usia mahasiswa ini ialah pemantapan pendirian hidup.

Berdasarkan uraian tersebut datap disimpulkan bahwa mahasiswa ialah seorang peserta didik berusia 18 sampai 25 tahun yang terdaftar dan menjalani pendidikannya di perguruan tinggi baik dari akademik, politeknik, sekolah tinggi, institut dan unvitersitas.

4. *Barcode*

Barcode merupakan gambar garis tegak yang biasanya ditempelkan pada item toko ritel, kartu identitas, dan surat pos untuk mengidentifikasi sejumlah produk tertentu. Secara umum *barcode* digunakan sebagai UPC (*Universal Price Code*) atau pembaca harga barang secara otomatis. Kode tersebut menggunakan urutan bar vertikal dan spasi untuk mewakili angka dan simbol lainnya. Simbol *barcode* biasanya terdiri dari lima bagian: zona tenang (*quite zone*), karakter awal, data karakter, karakter berhenti, dan *quite zone* lain (Rouse, 2009).

5. *Android*

Android adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh *Android* Inc., dengan dukungan

finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Android Versi 4.4 *Kitkat* dirilis tanggal 31.10.2013. Fokus terhadap peningkatan *user experience* dan performa khususnya pada perangkat kelas bawah. Android 4.4 Kitkat memberikan batas minimal RAM tersedia untuk Android sebesar 512 MB sehingga perangkat yang memiliki RAM di bawah itu digolongkan sebagai perangkat RAM rendah. Fitur terbarunya adalah:

a) *Screen recording*.
b) *New* Translucent *system UI*.
c) Peningkatan akses notifikasi.
d) *System-wide* settings untuk *closed captioning*.
e) Peningkatan kinerja.
f) Mengaktifkan Sprint Spark band 26 dan band 41.
g) Perbaikan kerentanan pada *Heartbleed* / *OpenSSL*.

6. *Android* SDK

Android Software Development Kit (SDK) merupakan perangkat lunak untuk membuat dan mengembangkan aplikasi Android. Di

dalamnya terdapat *library*, *debugger*, *Android Emulator*, serta perangkat lunak lainnya yang dibutuhkan untuk membuat sebuah aplikasi *Android*.

Pengembangan aplikasi *Android* umumnya dilakukan dengan bahasa pemrograman java, meski menggunakan bahasa lain seperti bahasa C juga bisa dilakukan menggunakan *Android Native Development Kit*. *Android* SDK akan selalu di-*update* menyesuaikan dengan versi sistem operasi *android* yang sudah resmi dirilis.

7. *Unifield Modeling Language* (UML)

Menurut Menurut Rosa (2013), UML (*Unified Modeling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodelogi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada *metodelogi* berorientasi objek.

UML mendefinisikan beberapa jenis diagram-diagram diantaranya sebagai berikut:

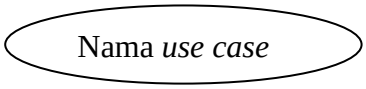
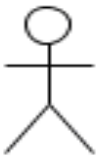
a. *Use Case Diagram*

Use case adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata *user*.

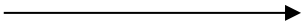
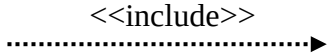
Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.

Adapun penjelasan *use case* diagram dapat dilihat pada tabel 1 berikut :

Tabel 1. Simbol *Use case* Diagram
(Sumber : Rosa 2013)

Simbol	Deskripsi
1	2
Use case  	Fungsionalitas yang disediakan sistem unit-unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase name *use case*
Aktor/actor  Nama Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi biasanya dinyatakan menggunakan kata bnda di awal frase nama *actor*.

1	2
Asosiasi/association 	Komunikasi antara actor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor
Ekstensi/extend <<extend>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu ; mirip dengan prinsip inheritance pada pemerograman berorientasi object; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambah.

1	2
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Include 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat di jalankan *use case* ini.

b. *State Maching* Diagram

State Machine diagram atau *statechart* diagram atau dalam bahasa Indonesia disebut diagram mesin status atau sering juga disebut diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau objek.

Komponen-komponen dasar yang ada dalam *state maching diagram* dapat dilihat pada tabel 2 berikut :

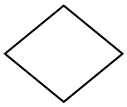
Tabel 2.Simbol *State Machine* Diagram
(Sumber : Rosa 2013)

Simbol	Deskripsi
1	2
Start/Status Awal (*initial State*) 	*Start* atau *initial state* adalah state atau keadaan awal pada saat sistem mulai hidup.
End/ Status Akhir (*Final State*) 	*End* atau *final state* adalah state keadaan akhir dari daur hidup suatu sistem.
Event	*Event* adalah kegiatan yang menyebabkan berubahan status mesin.
State 	*State* atau status adalah keadaan sistem pada waktu tertentu. *State* dapat berubah jika ada event tertentu yang memicu perubahan tersebut.

c. *Activity* Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use cas e*atau interaksi. Simbol-simbol dan deskripsi *activity* diagram dapat dilihat pada tabel 3 berikut ini :

Tabel 3. *Activity* Diagram
(Sumber : Rosa 2013)

Simbol	Deskripsi
Status Awal 	Status awal aktivasi sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivasi biasanya diawali dengan kata kerja
Pencabangan/ *decision* 	*Asosiasi* pencabangan dimana jika ada pilihan aktivitas lebih dari satu

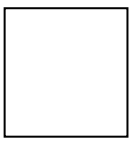
Penggabungan/join 	*Asosiasi* penggabungan dimana lebih dari satu aktivitas digabungkan menjadi asatu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

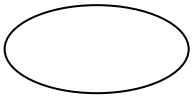
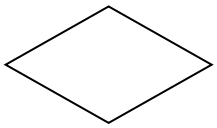
8. ERD (*Entity Relationship Diagram)*

Menurut Utami dan Hartanto (2012) ERD (*Entity Relationship Diagram*) adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data *relasional*.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Berikut adalah simbol-simbol ERD yang digunakan :

Tabel 4. Simbol-Simbol ERD

No	Simbol	Keterangan
1	2	3
1		Menunjukan entitas (*entity*) yaitu suatu objek yang mewakili sesuatu yang nyta dan dapat dibedakan dari sesuatu yang lain.

1	2	3
2		Menunjukan atribut (*atrribute*) yaitu entity yang mempunyai atribut dan berfungsi untuk mendeskripsikan karaketeristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain.
3		Menunjukkan hubungan atau relasi (*relationship*) yaitu hubungan antara suatu himpunan entitas lainnya.
4		Menunjukan alat penghubung, yang digunakan untuk menghubungkan entitas dengan entitas ataupun entitas dengan dengan atribut.

9. *MySQL*

Menurut Madcoms (2010) *MySQL* adalah bahasa pemrograman dan database yang sangat fleksibel dan mudah dipahami. Database sendiri berfungsi sebagai penampung data yang di inputkan melalui *form* website. *MySQL* menggunakan *SQL* yang bersifat grafis, selain itu *MySQL* dapat berjalan diberbagai platform, antara lain linux, windows, dan sebagainya.

*Database MySQ*L dapat dibuat dengan menggunakan tampilan jendela phpAdmin atau menggunakan sebuah *script PHP*.

MySQL (*My Structure language*) adalah sebuah program pembuat database yang bersifat *open source*. MySQL sebenarnya produk yang berjalan pada platform linux. Kerena sifatnya *open source*, maka *MySQL* dapat dijalankan pada semua platform baik Windows maupun Linux.

C. Perangkat yang Digunakan

1. *Android Studio*

Android Studio merupakan sebuah *software tools Integrated Development Environment* (*IDE*) untuk *platform Android*. *Android Studio* ini diluncurkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers.

Android Studio ini bersifat *free* di bawah Apache License 2.0. *Android Studio* awalnya dimulai dengan versi 0.1 pada bulan mei 2013, kemudian dibuat versi beta 0.8 yang dirilis pada bulan juni 2014. Yang paling terbaru dirilis v.3 pada bulan Oktober 2017. Berbasiskan JetBrainns IntelliJ IDEA, Studio didesain khusus untuk *Android Development*.

2. *Sublime Text*

Sublime text adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal dikalangan *developer* (pengembang), penulis dan desainer. Para programmer biasanya menggunakan *sublime text* untuk menyunting *source*

code yang sedang ia kerjakan. Sampai saat ini *sublime text* sudah mencapai versi 3.

3. *Xampp*

Xampp merupakan suatu *software* yang bersifat *open source* yang merupakan pengembangan dari LAMP (*Linux, Apache, MySQL, PHP, and Perl*). *Xampp* ini merupakan proyek non-*profit* yang dikembangkan oleh *Apache Friends* yang didirikan Kai Oswalad Seidler dan Kay Vogelgesang pada tahun 2002. Proyek mereka ini bertujuan untuk mempromosikan penggunaan *Apache web server* (Purbadian, 2016).

BAB III

ANALISIS DAN DESAIN SISTEM

A. Tinjauan Umum

Kehadiran perkuliahan pada Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya merupakan hal yang sangat penting, karena tanpa adanya kehadiran mahasiswa dan dosen maka tidak akan ada kegiatan perkuliahan yang dapat dilaksanakan. Sebab dalam perkuliahan harus ada dosen yang memiliki peran menyampaikan materi dan mahasiswa yang memiliki peran untuk menerima materi.

Perkembangan teknologi informasi saat ini telah mengubah pola pengolahan data dan publikasi data, yang dulunya masih secara manual dengan menggunakan media kertas dan harus datang ke tempat dimana data dipublikasikan untuk dapat melihat data tersebut. Sehingga kini pola pengolah data menjadi digital dan publikasi data menjadi *online.*

Fenomena seperti ini membuat sistem kehadiran yang dulunya menggunakan lembar kehadiran menjadi data digital dan publikasi data akan bersifat *online* sehingga data dapat dilihat dimana saja selama data tersimpan pada jaringan. Agar lebih efisien, kehadiran mahasiswa dibuat dalam bentuk aplikasi berbasis *android* dengan memanfaatkan teknologi yang berkembang saat ini.

B. Analisis

Pada tahap ini, penulis akan melakukan analisa sistem yang akan dibangun nanti. Adapun analisa tersebut sebagai berikut :

1. Anlisis Kelemahan Sistem

Sistem kehadiran pada STMIK Palangkaraya masih bisa dikatakan kurang efisien karena masih menggunakan media lembar kertas kehadiran, sehingga dosen harus mengambil lembar kehadiran terlebih dahulu ke bagian AKMA agar mahasiswa dapat mengisi lembar kehadiran. Pihak akma masih harus melakukan rekap kehadiran seluruh mahasiswa secara manual setiap mendekati akhir semester dengan melihat daftar lembar kehadiran yang biasanya selalu dikumpulkan di akma, sehingga akan memakan banyak waktu untuk proses tersebut.

Dari hasil menganalisa sistem yang telah dilakukan maka metode analisa yang digunakan adalah PIECES sehingga diperolehlah beberapa data yang dapat dilihat pada tabel berikut :

Tabel 5. Analisa PIECES Sistem Lama dan Baru

Data	Sistem	Sistem Baru
1	2	3
Performance (Kinerja)	Untuk dapat melihat data kehadiran mahasiswa atau dosen harus melihat lembar kehadiran terlebih dahulu	Dengan menggunakan aplikasi kehadiran, maka mahasiswa ataupun dosen dapat melihat total kehadiran tanpa mencari lembar kehadiran
Information (Informasi)	Pihak kampus harus menghubungi mahasiswa secara manual dengan mengubungi mahasiswa yang bersangkutan jika kehadirannya kurang dari peraturan kampus untuk mengikuti ujian akhir semester.	Dengan adanya fitur notifikasi, pihak admin hanya perlu mengirimkan notifikasi ke mahasiswa yang kehadirannya kurang dan secara otomatis mahasiswa yang bersangkutan akan menerima pemberitahuan.

1	2	3
Economy (Ekonomi)	Sebelum memulai kehadiran, dosen diharuskan mengambil lembar kehadiran diruangan akma terlebih dahulu.	Dosen hanya perlu menggunakan *smartphone android* untuk login ke aplikasi kehadiran dan memilih matakuliah maka proses absensi bisa dilakukan.
Control (Kontrol)	Proses kehadiran yang lama masih bisa dimanipulasi dengan melakukan titip absen dan melakukan banyak kehadiran dalam satu hari untuk menutupi kekurangan jam mengajar dosen.	Kecurangan pada kehadiran dapat dikurangi dengan menggunakan aplikasi ini, karena mahasiswa diharuskan maju kedepan satu persatu sehingga tidak bisa melakukan titip absen dan setiap dosen melakukan jam mengajar histori akan tersimpan di aplikasi.
Effeciency (Efisiensi/daya guna)	Proses rekapan kehadiran masih sangat lambat, karena pihak akma harus menghitung total kehadiran mahasiswa dan jam mengajar dosen secara manual dari lembar kehadiran.	Pada aplikasi web admin, rekapan kehadiran dosen dan mahasiswa secara otomatis telah terhitung dan ditampilkan dalam bentuk pdf.
Service (Layanan)	Proses kehadiran menggunakan media kertas dan melakukan tanda tangan menggunakan pulpen	Proses kehadiran menggunakan aplikasi yang memiliki user *interface* mudah dipahami dan digunakan, data yang digunakan adalah *QR Code* yang berasal dari NIM mahasiswa

2. Analisis Kebutuhan Sistem

a. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan untuk pengembangan sistem ini terdiri dari 2 bagian yaitu kebutuhan untuk mengembangkan sistem dan kebutuhan untuk implementasi sistem. Berikut perangkat yang digunakan :

1). Kebutuhan perangkat keras untuk mengembangkan sistem

Berikut merupakan detail perangkat keras yang digunakan untuk mengembangkan sistem :

a). Laptop Lenovo G40-70

(1). Processor Intel Core i7 4510u 2,40 Ghz

(2). RAM 8GB DDR3L

(3). SSD 250GB

b). Perangkat Selular XIAOMI MI3W

(1). Chipset Qualcomm MSM8274AB Snapdragon 800

(2). CPU Quad-Core 2.3 Ghz Krait 400

(3). RAM 2GB

(4). Memori 16GB

(5). USB MicroUSB 2.0

2). Kebutuhan perangkat keras untuk implementasi sistem

Perangkat keras yang dibutuhkan untuk implementasi aplikasi kehadiran ini adalah semua jenis *smartphone* dengan sistem operasi *android* versi 4.1 *Jelly Bean* sampai versi terbaru. Sedangkan

komputer yang diperlukan untuk peng inputan data oleh akma adalah komputer yang memiliki spesifikasi diatas Pentium 4, RAM 4GB, HDD 120GB, Browser Chrome/FireFox terbaru dan terkoneksi dengan internet.

b. Kebutuhan perangkat lunak

Adapun perangkat lunak yang digunakan dalam pengembangan sistem sebagai berikut :

1). Android Studio 64 Bit (versi terbaru)

2). Windows 10 64 Bit

3). Android versi 4.1 (Jelly Bean)

c. Kebutuhan informasi

Sistem yang dibangun akan memiliki fitur untuk memulai proses pencatatan/absensi kegiatan perkuliahan, menampilkan daftar mata kuliah yang diambil mahasiswa, daftar mata kuliah yang dosen ajarkan dan notifikasi jika kehadiran mahasiswa kurang dari 75%. Dalam kasus ini penulis menggunakan *smartphone* penulis sendiri sebagai media *testing* awal, sedangkan ketika program telah memasuki tahap *testing* terakhir penulis akan menggunakan *smartphone* lain atau menggunakan *smartphone* orang lain.

d. Kebutuhan pengguna

Sistem yang dibangun ini nantinya dapat digunakan oleh mahasiswa dan dosen STMIK Palangkaraya sebagai aplikasi untuk menjalankan proses pencatatan kehadiran perkuliahan. Untuk dapat

menggunakan sistem nantinya, pengguna membutuhkan *smartphone* dengan sistem operasi android 4.1 (Jelly Bean) ataupu versi diatasnya.

Bagian akma juga mengelola data aplikasi ini dengan menggunakan web backend yang nantinya dapat dijalankan pada browser chrome ataupun firefox.

3. Analisis Kelayakan Sistem

Pada bagian ini penulis membahas analisis tentang kelayakan teknologi dan kelayakan operasional.

a. Kelayakan teknologi

Dari kelayakan teknologi, aplikasi ini dapat dinyatakan layak karena untuk menjalankan aplikasi ini pengguna hanya memerlukan *smartphone android* untuk menampilkan *qr code* atau membawa kartu kehadiran yang berisikan gambar *qr code*.

b. Kelayakan operasional

Dari segi operasional aplikasi ini dikatakan layak, karena sebagian besar mahasiswa dan dosen STMIK Palangkaraya menggunakan *smartphone android* dan mampu mengoperasikan perangkat tersebut. Dengan menggunakan aplikasi ini pengguna dapat menghemat waktu dan tenaga, karena dosen tidak perlu lagi membawa lembar kehadiran setiap mengajar, adanya fitur notifikasi jika kehadiran mahasiswa kurang dari 75% dan bagian akma akan dengan mudah merekap kehadiran setiap akhir semester.

C. Desain Sistem

1. Desain Input

Desain *input* yang menjadi tahap perencanaan sebelum di*implementasi*kan ke dalam kode program adalah sebagai berikut :

a) Desain *Input Login* Dosen

Desain *input login* dosen adalah rancangan data yang digunakan sebagai data masukkan pada sistem saat dosen akan *login* kedalam aplikasi *android* sebagai pengguna.

Tabel 6. Desain *Input Login* Dosen

No.	Data *Input*	Keterangan
1.	NIK	NIK Dosen untuk masuk ke *Activity* dosen.
2.	*Password*	*Password* untuk keamanan saat dosen akan melakukan *login*.

b) Desain *Input Login* Mahasiswa

Desain *input login* mahasiswa adalah rancangan data yang digunakan sebagai data masukkan pada sistem saat mahasiswa akan *login* kedalam aplikasi *android* sebagai pengguna.

Tabel 7. Desain *Input Login* Mahasiswa

No.	Data *Input*	Keterangan
1.	NIM	NIM Mahasiswa untuk masuk ke *Activity* mahasiswa.
2.	*Password*	*Password* untuk keamanan saat mahasiswa akan melakukan *login*.

c) Desain *Input Login* ADMIN

Desain *input login* admin adalah rancangan data yang digunakan sebagai data masukkan pada sistem saat admin akan *login* kedalam web admin.

Tabel 8. Desain *Input Login* Admin

No.	Data *Input*	Keterangan
1.	Username	Username AKMA untuk masuk ke halaman ADMIN.
2.	Password	Password untuk keamanan saat admin AKMA akan *login*.

d) Desain *Input* Data Mahasiswa

Desain *input* data mahasiswa adalah data yang dimasukkan oleh admin ke dalam sistem dan akan tersimpan ke database.

Tabel 9. Desain *Input* Data Mahasiswa

No.	Data *Input*	Keterangan
1.	NIM	NIM Mahasiswa.
2.	Nama	Nama Mahasiswa.
3.	Jurusan	Jurusan Mahasiswa.
4.	Password	Password yang digunakan mahasiswa untuk login

e) Desain *Input* Data Dosen

Desain *input* data dosen adalah data yang dimasukkan oleh admin ke dalam sistem dan akan tersimpan ke database.

Tabel 10. Desain *Input* Data Dosen

No.	Data *Input*	Keterangan
1.	NIK	NIM Dosen.
2.	Nama	Nama Dosen.
3.	Jurusan	Jurusan Dosen.
4.	Password	Password yang digunakan dosen untuk *login*

f) Desain *Input* Data Matakuliah

Desain *input* data matakuliah adalah data yang dimasukkan oleh admin ke dalam sistem dan akan tersimpan ke database.

Tabel 11. Desain *Input* Data Matakuliah

No.	Data *Input*	Keterangan
1.	Kode MK	Kode Matakuliah.
2.	Nama MK	Nama Matakuliah.
3.	Jurusan	Jurusan Matakuliah.

g) Desain *Input* Data KRS Mahasiswa

Desain *input* data krs mahasiswa adalah data rencana studi yang akan dimasukkan oleh admin ke dalam sistem dan akan tersimpan ke database.

Tabel 12. Desain *Input* Data KRS Mahasiswa

No.	Data *Input*	Keterangan
1.	Nama Matakuliah	Nama Matakuliah
2.	Nama Dosen	Dosen yang mengajar matakuliah
3.	Kelas	Kelas matakuliah dan dosen
4.	Tahun	Tahun Akademik

h) Desain *Input* Data Mahasiswa Pada KRS Matakuliah

Desain *input* data krs mahasiswa adalah data nama mahasiswa yang akan dimasukkan oleh admin ke dalam sistem dan akan tersimpan ke database.

Tabel 13. Desain *Input* Data Mahasiswa Pada Matakuliah

No.	Data *Input*	Keterangan
1.	Nama Mahasiswa	Nama Mahasiswa

2. Desain Proses

Desain proses adalah suatu rangkaian kegiatan penggunaan saat menggunakan sistem untuk menghasilkan sebuah informasi adalah sebagai berikut:

a) Proses Dosen

1) Proses pertama dosen ketika akan menjalankan proses perkuliahan adalah dengan *login* terlebih dahulu pada aplikasi *android*.
2) Proses *Login*, pada proses ini dosen mengisi NIK dan Password pada *form Login*. Jika data yang dimasukkan benar maka akan diarahkan ke *activity* utama dosen.
3) Proses mengaktifkan status perkuliahan, pada proses ini dosen membuka menu mulai mengajar. Sehingga akan muncul 2 pilihan radio button yaitu mulai dan akhiri, jika dosen memilih *radio button* mulai dan menekan tombol proses maka secara otomatis akan tampil list beberapa matakuliah yang dapat dosen ajarkan.
4) Proses mengakifkan fitur *scan* kamera, dalam proses ini dosen harus memilih salah satu matakuliah yang akan diajarkan. Setelah dosen menekan salah satu matakuliah yang akan diajarkan, maka kamera ponsel dosen otomatis akan terbuka dan kamera sudah siap untuk melakukan *scan* kehadiran mahasiswa.
5) Proses mengkahiri perkuliahan, dalam proses ini dosen hanya menekan tombol kembali maka otomatis kamera akan tertutup dan pilih *radio button* akhiri menonaktifkan lalu pilih tombol proses maka secara otomatis akan teralihkan ke *activity* utama dosen.

b) Proses Mahasiswa

1) Pada proses absensi kehadiran, mahasiswa dapat menggunakan kartu kehadiran yang telah berisikan *QR Code* dan langsung mengarahkan kartu tersebut ke kamera dosen maka akan otomatis absen akan tersimpan kedalam sistem.
2) Jika mahasiswa lupa membawa kartu kehadiran, maka mahasiswa dapat menggunakan *QR Code* yang ada pada aplikasi. Dengan cara *login* terlebih dulu kedalam aplikasi dan membuka menu tampilkan *QR Code*, maka secara otomatis gambar *QR Code* dari nim mahasiswa akan tampil.
3) Proses selanjutnya setelah dosen mengaktifkan perkuliahan, mahasiswa diharuskan maju kedepan dan mengarahkan kartu kehadiran yang telah ada gambar *QR Code* ke kamera *smartphone* dosen. Jika muncul pesan sukses maka status mahasiswa dinyatakan telah hadir pada perkuliahan tersebut.

c) Proses Admin Akma

1) Proses Login Admin Akma, pada proses ini admin mengisi *username* dan *password*. Jika data yang dimasukka benar maka akan diarahkan ke halaman utama admin.
2) Proses input data Mahasiswa, pada proses ini data mahasiswa yang di *input*kan diantaranya NIM, nama mahasiswa, jurusan mahasiswa dan *password* mahasiswa.

3) Proses *input* data Dosen, pada proses ini data dosen yang di *input*kan diantaranya NIK, nama dosen, jurusan dosen dan *password* dosen.
4) Proses *input* data Matakuliah, pada proses ini data matakuliah yang di inputkan diantaranya kode matakuliah, nama matakuliah dan jurusan.
5) Proses *input* data KRS, pada proses ini data KRS yang di *input*kan adalah nama matakuliah, nama dosen yang mengajarkan matakuliah tersebut, kelas untuk matakuliah dosen dan tahun akademik.
6) Proses *input* data nama mahasiswa pada matakuliah yang ada pada krs, pada proses ini data yang di *input*kan adalah nama mahasiswa yang akan mengambil kelas matakuliah yang telah dipilih oleh mahasiswa.
7) Proses mengirimkan notifikasi, pada proses ini admin masuk ke menu data kehadiran. Memilih mahasiswa yang ingin dikirimkan notifikasi dan menekan tombol lihat kehadiran. Pilih menu lihat total, tekan tombol kirim pesan. Maka akan tampil halaman yang berisikan *form* untuk mengisi judul dan pesan, setelah selesai maka tekan tombol kirim. Secara otomatis mahasiswa yang bersangkutan akan menerima pemberitahuan pada *smartphone* yang digunakan oleh mahasiswa tersebut.

3. Desain Keluaran (*Output*)

Desain *output* yang akan menjadi keluaran dari sistem ini adalah berupa dialog layar terminal dari media perangkat lunak aplikasi *android* dan web browser.

Berikut adalah output dari sistem ini :

a) Informasi daftar matakuliah yang diajarkan dosen dalam 1 semester.
b) Informasi daftar matakuliah yang diambil oleh mahasiswa pada 1 semester.
c) Informasi total kehadiran mahasiswa pada setiap mata kuliah.
d) Notifikasi jika kehadiran mahasiswa kurang dari total aturan yang ditentukan.
e) Laporan yang berupa *print out* total kehadiran pada akhir semester.

4. Rancangan Desain *Interface*/Antarmuka

Pada tahapan ini akan dijelaskan rancangan desain *interface* pada Aplikasi Kehadiran Mahasiswa STMIK Palangkaraya Berbasis *Android*. Adapun rancangan desain *interface*-nya adalah sebagai berikut :

a) *Activity* Dosen

1) *Activity Login* Dosen

Activity Login Dosen adalah halaman yang digunakan untuk menampilkan *activity* utama dosen, dosen memasukkan *input*an seperti NIK dan *password* kemudian menekan tombol *Login* yang ada pada *Login Activity*. Berikut ini adalah rancangan desain *interface*-nya yang dapat dilihat pada gambar 2 :



Gambar 2. *Activity Login*

2) *Activity* Utama Dosen

Activity Utama Dosen adalah halaman yang muncul ketika dosen berhasil *Login*. Berikut ini adalah rancangan desain *interface* halaman beranda untuk dosen yang dapat dilihat pada gambar 3 :



Gambar 3. *Activity* Utama Dosen

3) *Activity* Mulai Mengajar

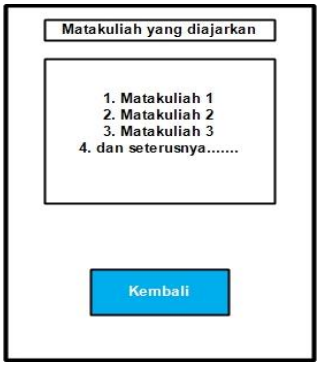
Activity Mulai Mengajar adalah halaman yang digunakan untuk mengaktifkan menu perkuliahan. Berikut ini adalah rancangan desain *interface* halaman mulai mengajar untuk dosen yang dapat dilihat pada gambar 4 :



Gambar 4. *Activity* Mulai Mengajar

4) *Activity* Daftar Matakuliah Dosen

Activity Daftar Matakuliah adalah halaman yang digunakan untuk menampilkan daftar mata kuliah yang akan diajarkan dosen pada satu semester. Berikut ini adalah rancangan desain *interface* halaman daftar matakuliah yang dapat dilihat pada gambar 5 :



Gambar 5. *Activity* Daftar Matakuliah Dosen

5) *Activity* Pengaturan Dosen

Activity Pengaturan Dosen adalah halaman yang digunakan untuk mengubah *password* akun dosen. Berikut ini adalah rancangan desain *interface* halaman pengaturan yang dapat dilihat pada gambar 6 :



Gambar 6. *Activity* Pengaturan Dosen

b) *Activity* Mahasiswa

1) *Activity Login* Mahasiswa

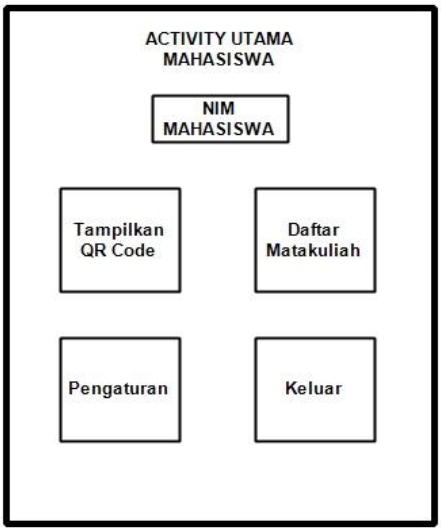
Activity Login Mahasiswa adalah halaman yang digunakan untuk menampilkan *activity* utama mahasiswa, mahasiswa memasukkan inputan seperti NIM dan *password* kemudian menekan tombol *Login* yang ada pada *Login Activity*. Berikut ini adalah rancangan desain *interface*-nya yang dapat dilihat pada gambar 7 :



Gambar 7. *Activity Login* Mahasiswa

2) *Activity* Utama Mahasiswa

Activity Utama Mahasiswa adalah halaman yang digunakan ketika mahasiswa berhasil *login*, berikut ini adalah rancangan desain *interface* halaman utama yang dapat dilihat pada gambar 8 :



Gambar 8. *Activity* Utama Mahasiswa

3) *Activity* Tampilkan *QR Code*

Activity Tampilkan *QR Code* adalah halaman yang digunakan untuk menampilkan NIM mahasiswa yang telah degenerate menjadi gambar *QR Code*. Berikut ini adalah rancangan desain *interface* halaman daftar mata kuliah yang dapat dilihat pada gambar 9 :



Gambar 9. *Activity* Tampilkan *QR Code*

4) *Activity* Daftar Matakuliah Mahasiswa

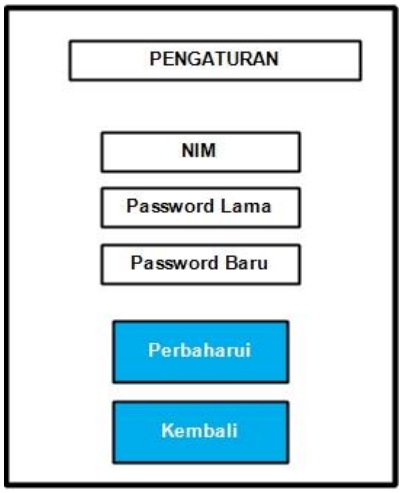
Activity Daftar Mata Kuliah adalah halaman yang digunakan untuk menampilkan daftar mata kuliah yang diambil mahasiswa pada satu semester. Berikut ini adalah rancangan desain *interface* halaman daftar mata kuliah yang dapat dilihat pada gambar 10:



Gambar 10. *Activity* Daftar Matakuliah Mahasiswa

5) *Activity* Pengaturan Mahasiswa

Activity Pengaturan adalah halaman yang digunakan untuk mengubah *password* akun mahasiswa. Berikut ini adalah rancangan desain *interface* halaman pengaturan yang dapat dilihat pada gambar 11 :

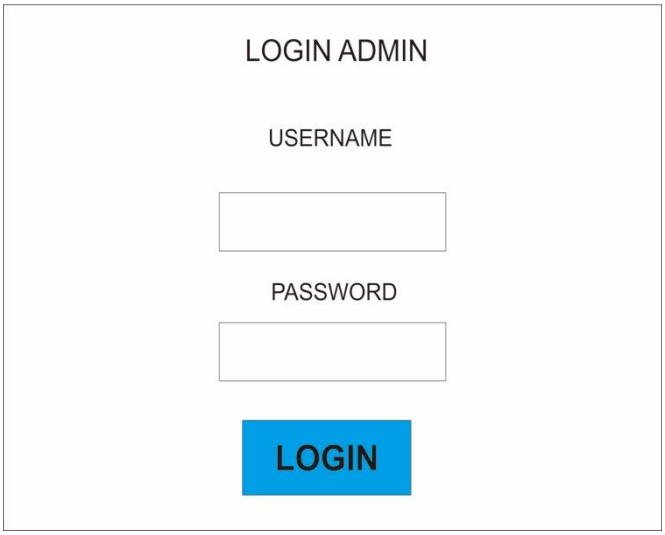


Gambar 11. *Activity* Pengaturan Mahasiswa

c) Halaman Admin

1) Halaman *Login* Admin

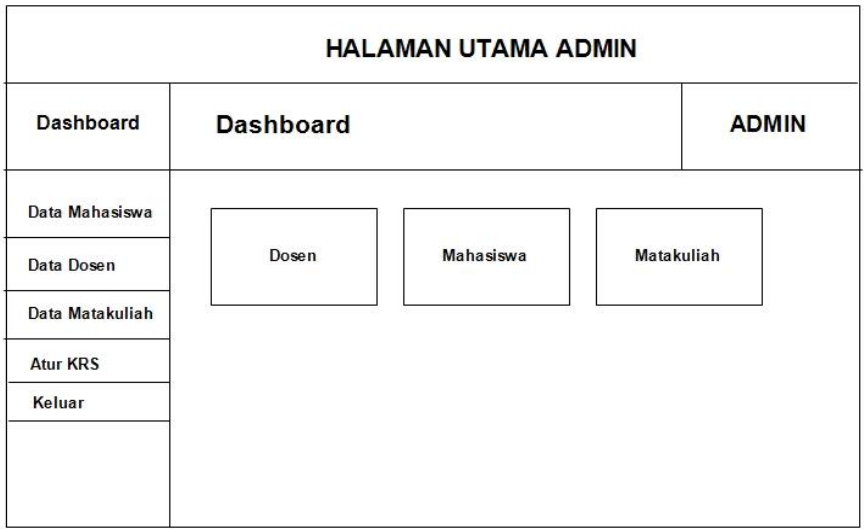
Halaman *Login* admin adalah halaman yang digunakan untuk masuk ke halaman admin, admin memasukkan *input*-an seperti *username* dan *password* kemudian menekan tombol *Login*. Berikut ini adalah rancangan desain interface halaman login admin yang dapat dilihat pada gambar 12 :



Gambar 12. Halaman *Login* Admin

2) Halaman *Dashboard* Admin

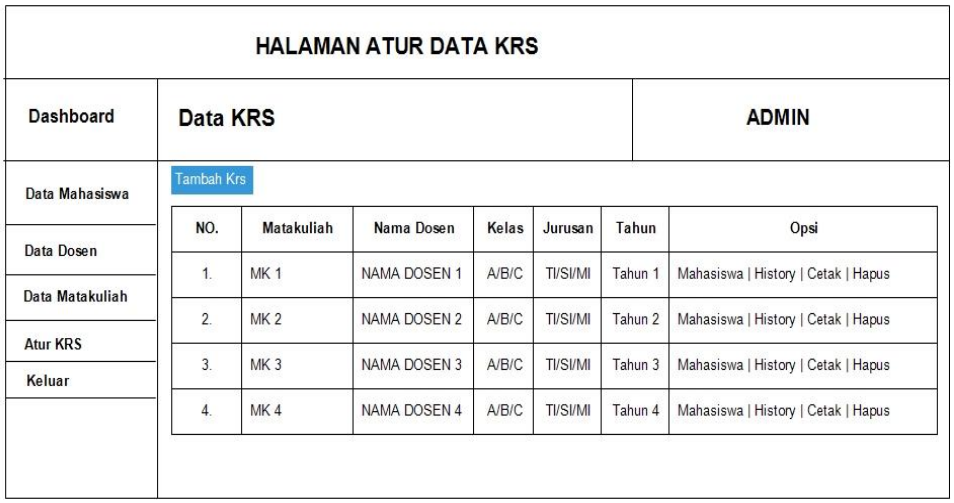
Halaman *Dashboard* Admin adalah halaman yang muncul ketika admin berhasil *Login*. Berikut ini adalah rancangan desain *interface* halaman dashboard untuk admin yang dapat dilihat pada gambar 13 :



Gambar 13. Halaman Utama Admin

3) Halaman Atur Data KRS

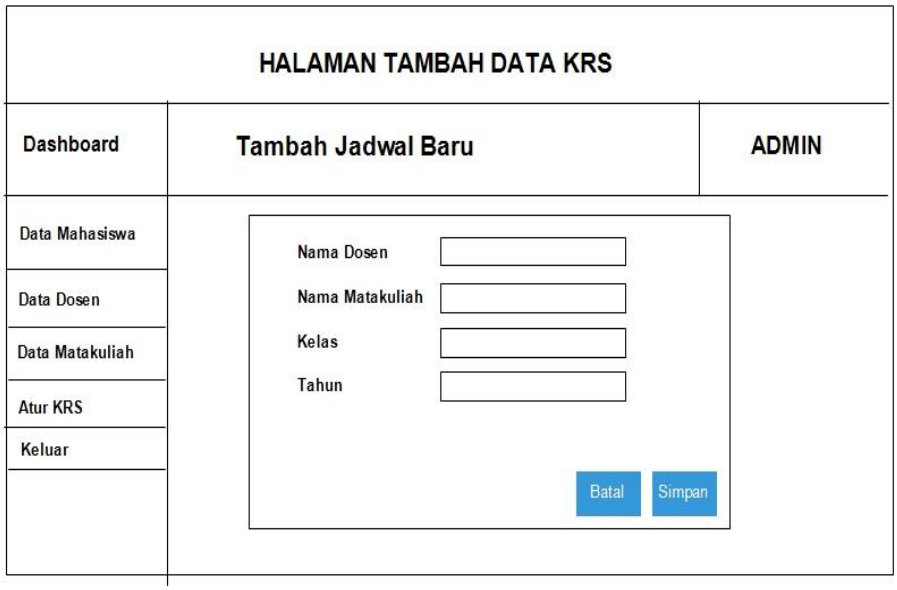
Halaman Atur KRS adalah halaman yang digunakan untuk menambahkan data matakuliah dan menampilkan data matakuliah yang diambil setiap mahasiswa dan melihat dosen pengampu pada matakuliah tersebut. Berikut ini adalah rancangan desain *interface* halaman untuk atur data KRS yang dapat dilihat pada gambar 14 :



Gambar 14. Halaman Atur Data KRS

4) Halaman Tambah Data KRS

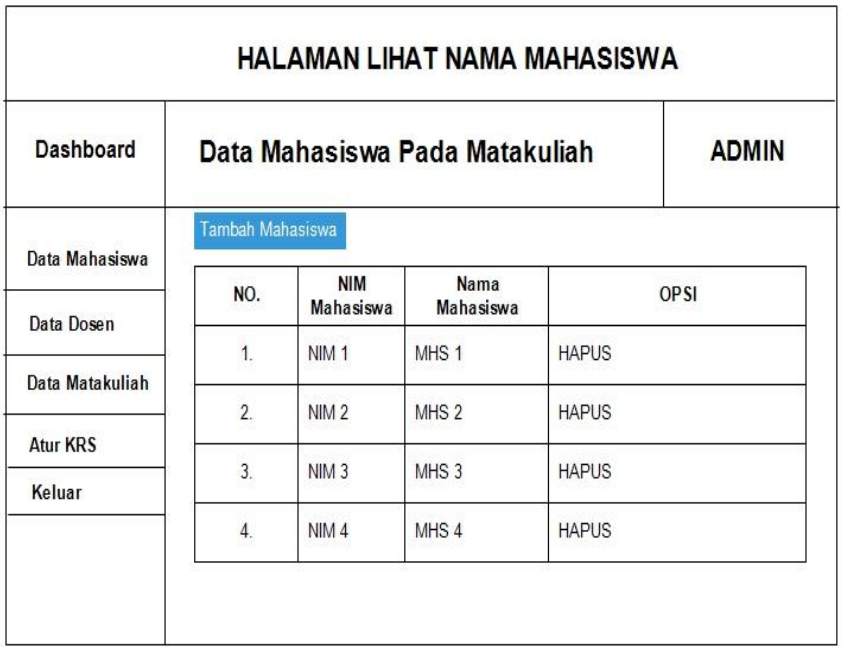
Halaman Tambah Data KRS adalah halaman yang digunakan untuk menambahkan data matakuliah yang diambil setiap mahasiswa dan menentukan dosen pengampu pada matakuliah tersebut. Berikut ini adalah rancangan desain *interface* halaman untuk menambah data KRS yang dapat dilihat pada gambar 15 :



Gambar 15. Halaman Tambah Data KRS

5) Halaman Mahasiswa

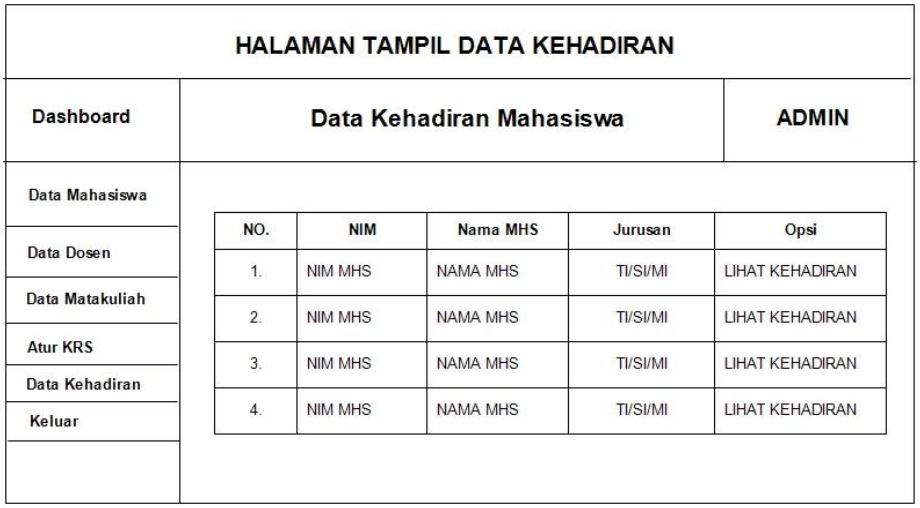
Halaman Mahasiswa adalah halaman yang digunakan untuk menampilkan data nama-nama mahasiswa yang mengambil matakuliah pada 1 semester. Berikut ini adalah rancangan desain *interface* halaman untuk lihat mahasiswa yang dapat dilihat pada gambar 16 :



Gambar 16. Halaman Lihat Nama Mahasiswa

6) Halaman Tampil Data Kehadiran

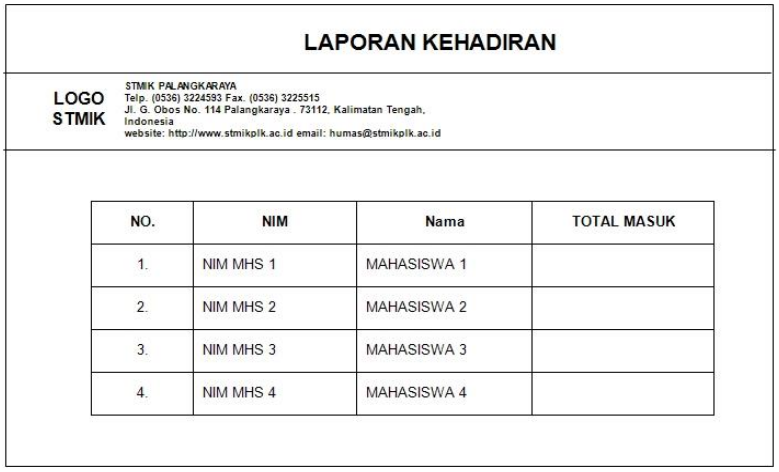
Halaman tampil data kehadiran adalah halaman yang digunakan untuk menampilkan data kehadiran setiap mahasiswa pada tiap matakuliah yang diambil. Berikut ini adalah rancangan desain *interface* halaman untuk menampilkan data kehadiran yang dapat dilihat pada gambar 17 :



Gambar 17. Halaman Tampil Data Kehadiran

7) Halaman Laporan Kehadiran

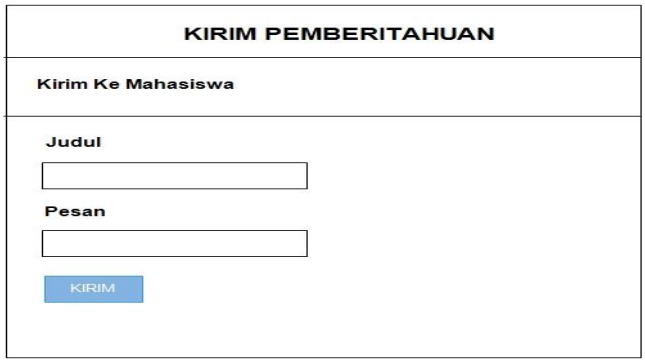
Halaman laporan kehadiran adalah halaman yang digunakan untuk menampilkan rekapan total kehadiran mahasiswa pada matakuliah. Berikut ini adalah rancangan desain interface halaman laporan kehadiran yang dapat dilihat pada gambar 18 :



Gambar 18. Halaman Laporan Kehadiran

8) Halaman Kirim Pesan

Halaman kirim pesan adalah halaman yang digunakan untuk mengirim pesan pemberitahuan kepada mahasiswa, dan pesan akan terkirim ke ponsel yang telah di instalkan aplikasi kehadiran berbasis android. Berikut ini adalah rancangan desain interface halaman kirim pesan yang dapat dilihat pada gambar 19 :



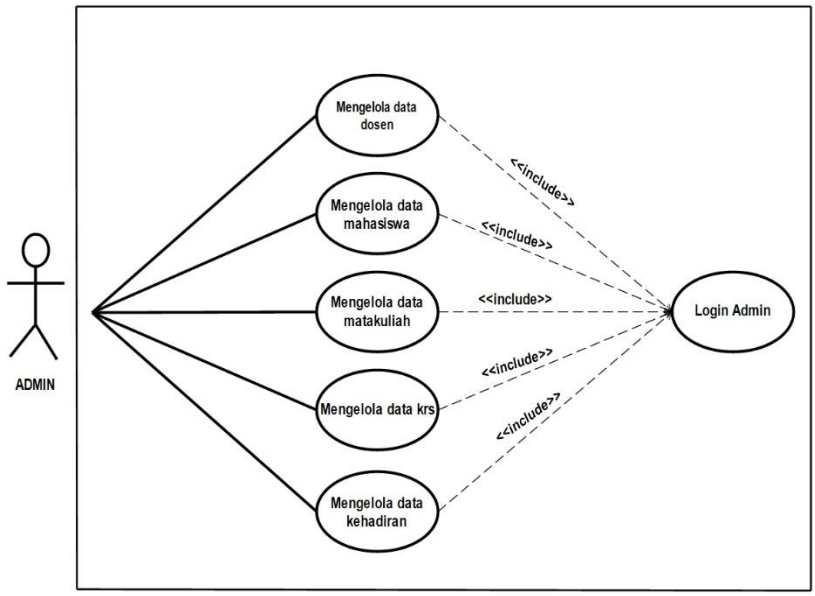
Gambar 19. Halaman Kirim Pemberitahuan

5. Penggunaan Metode Perancangan Sistem

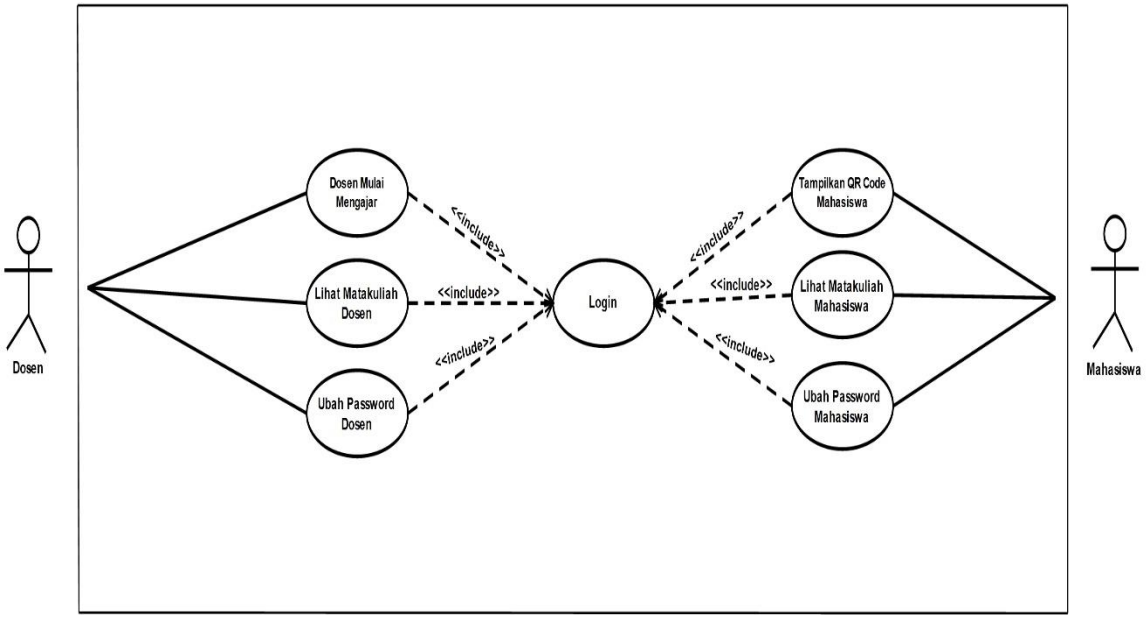
a) *Use case* Diagram

Sebuah diagram *use case* mengidentifikasi aktor yang terlibat dalam kegiatan yang dapat dilakukan terhadap sistem. Adapun dalam sistem ini terdapat 3 aktor yaitu admin (Staff Akma), Dosen dan mahasiswa. Admin merupakan

aktor yang mengelola seluruh data yang ada pada sistem, sedangkan dosen dan mahasiswa adalah aktor yang menggunakan layanan pada sistem. Berikut ini adalah *use case* diagram admin, dosen dan mahasiswa pada sistem yang dapat dilihat pada gambar 20 dan 21 :



Gambar 20. *Use case* Diagram Admin



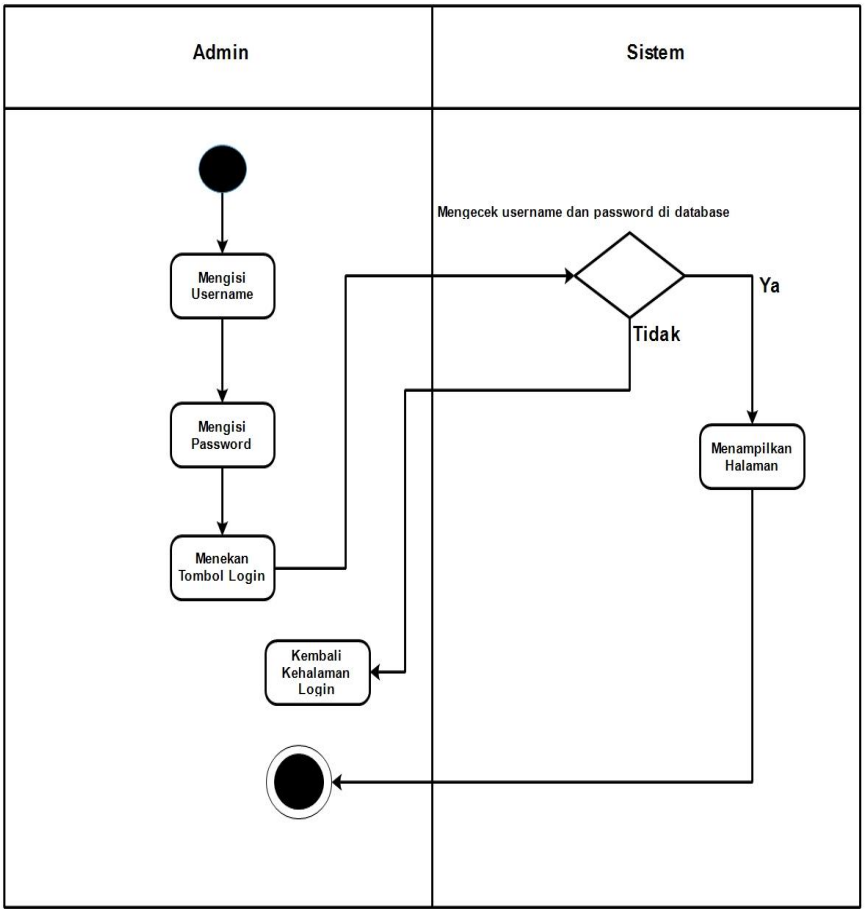
Gambar 21. *Use case* Diagram Dosen dan Mahasiswa

b) *Activity* Diagram Admin

1) *Activity* Diagram Login

Admin melakukan login dengan cara memasukkan *username* dan *password* pada *form login* lalu menekan tombol *login*. Sistem akan mengecek ke dalam database apakah data yang

dimasukkan tersebut sesuai dengan yang ada di database. Jika data sesuai, maka sistem akan menampilkan halaman utama admin. Jika data tidak sesuai maka sistem akan menampilkan halaman *Login*. Berikut ini *activity* diagram *login* admin yang dapat dilihat pada gambar 22 :

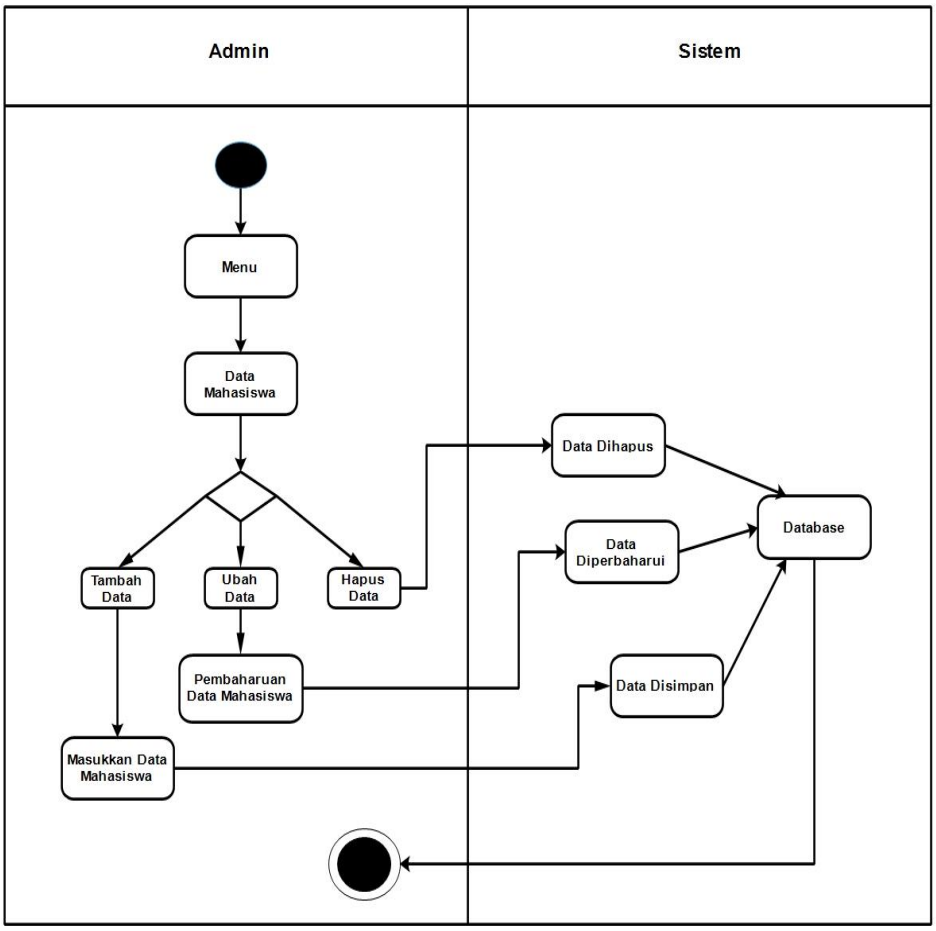


Gambar 22. *Activity* Diagram *Login*

2) *Activity* Diagram Kelola Data Mahasiswa

Setelah admin berhasil melakukan *login* maka admin dapat melakukan beberapa aktifitas di dalam sistem, seperti kelola data mahasiswa, diantaranya tambah data, ubah data dan hapus data mahasiswa yang kemudian disimpan ke dalam database. Berikut

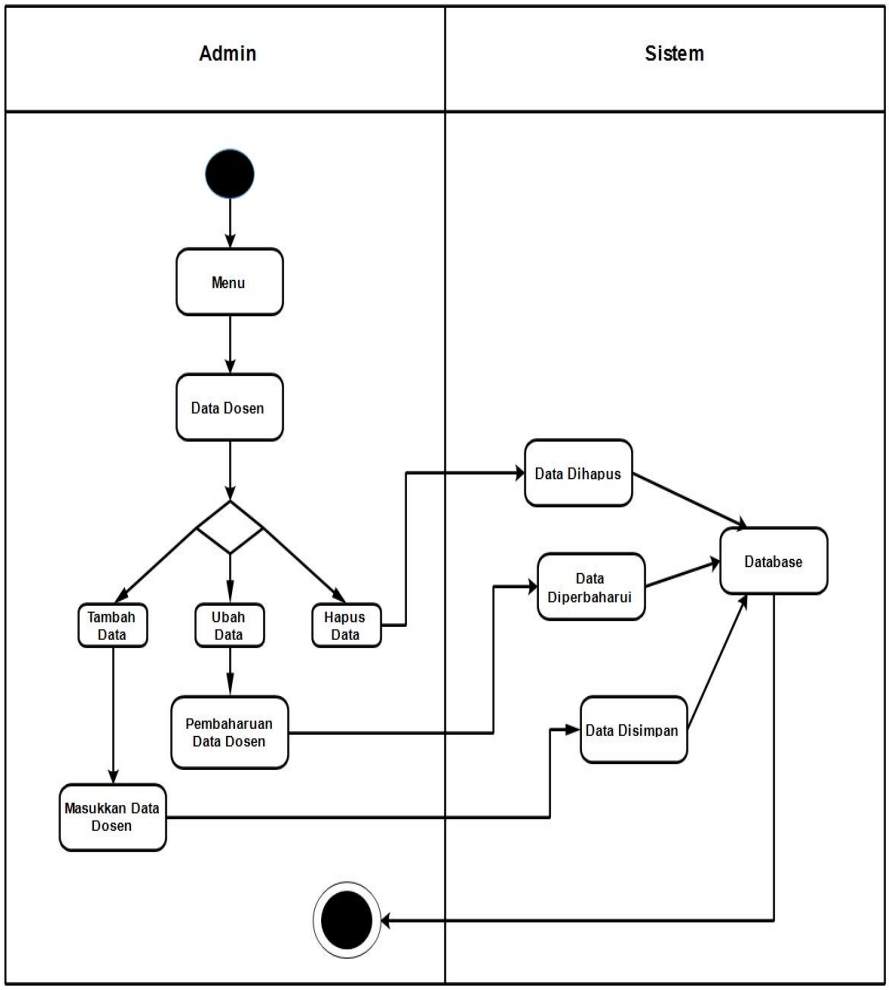
ini *activity* diagram kelola data mahasiswa yang dapat dilihat pada gambar 23 :



Gambar 23. *Activity* Diagram Kelola Data Mahasiswa

3) *Activity* Diagram Kelola Data Dosen

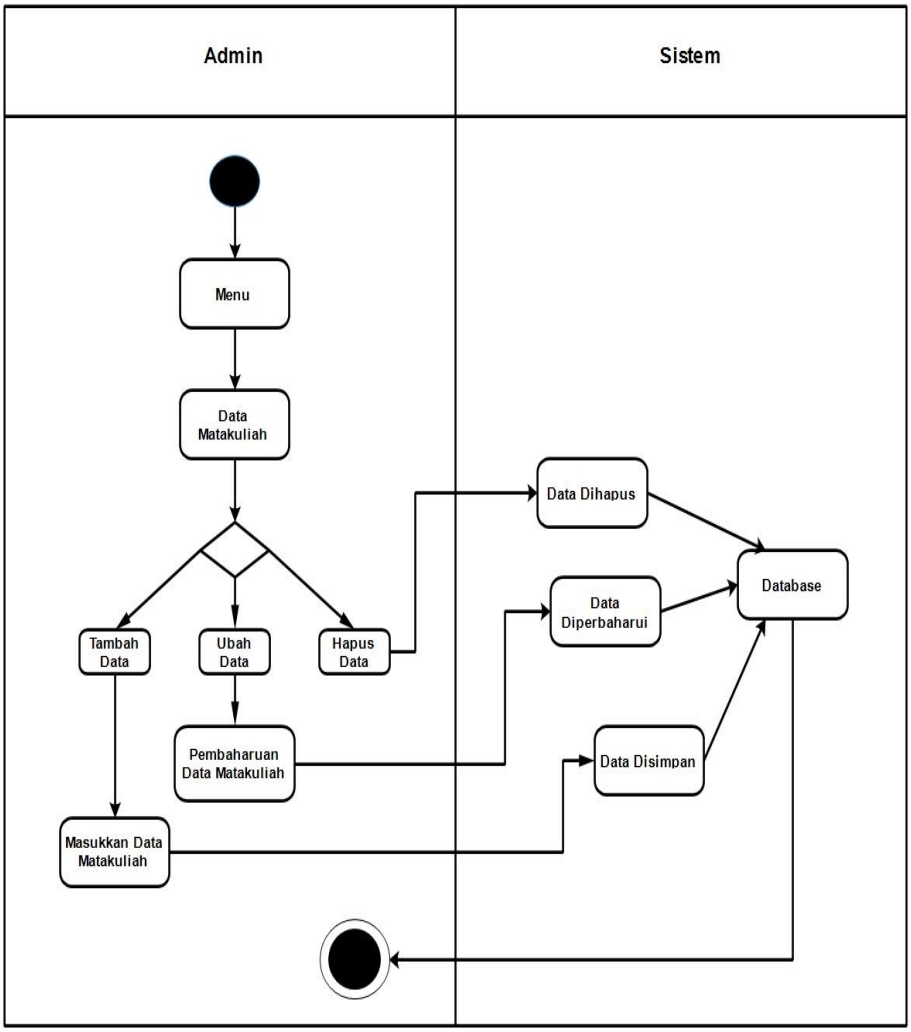
Setelah admin berhasil melakukan *login* maka admin dapat melakukan beberapa aktifitas di dalam sistem, seperti kelola data dosen, diantaranya tambah data, ubah data dan hapus data dosen yang kemudian disimpan ke dalam database. Berikut ini *activity* diagram kelola data dosen yang dapat dilihat pada gambar 24 :



Gambar 24. *Activity* Diagram Kelola Data Dosen

4) *Activity* Diagram Kelola Data Matakuliah

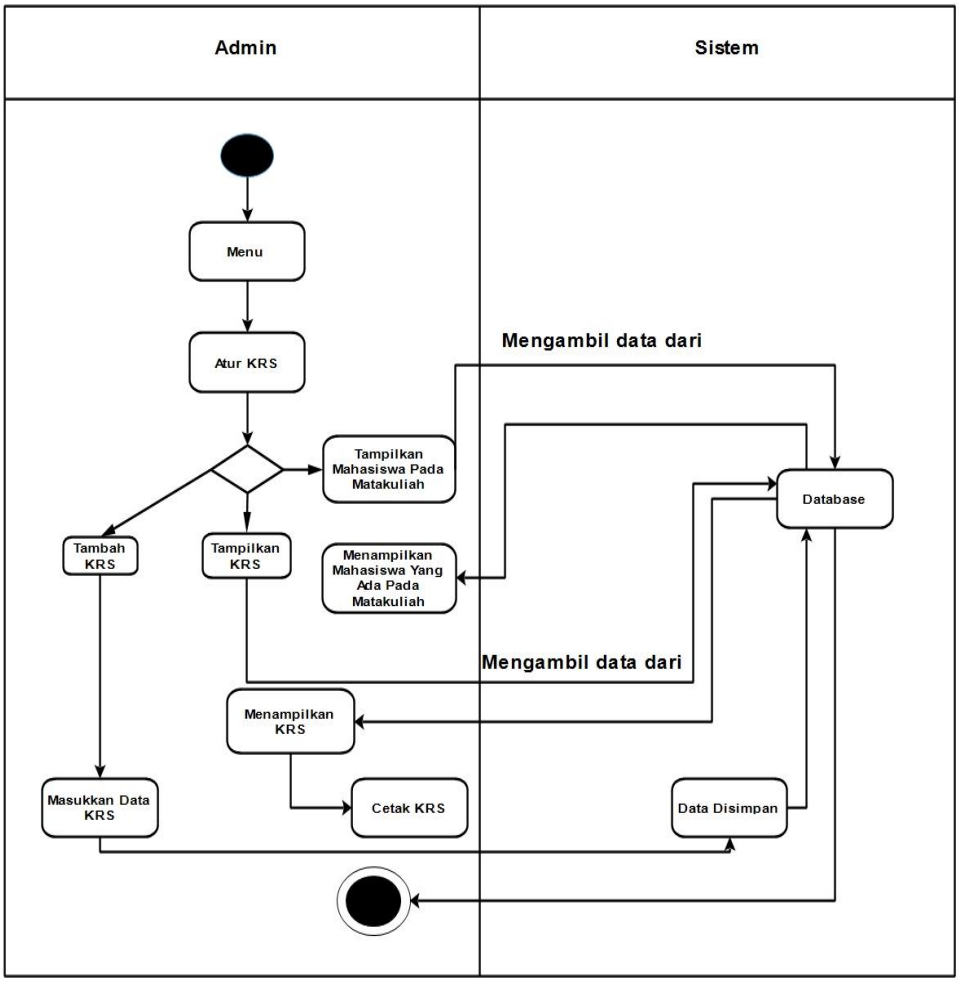
Setelah admin berhasil melakukan *login* maka admin dapat melakukan beberapa aktifitas di dalam sistem, seperti kelola data matakuliah, diantaranya tambah data, ubah data dan hapus data matakuliah yang kemudian disimpan ke dalam database. Berikut ini *activity* diagram kelola data matakuliah yang dapat dilihat pada gambar 25 :



Gambar 25. *Activity* Diagram Kelola Data Matakuliah

5) *Activity* Diagram Kelola Data KRS

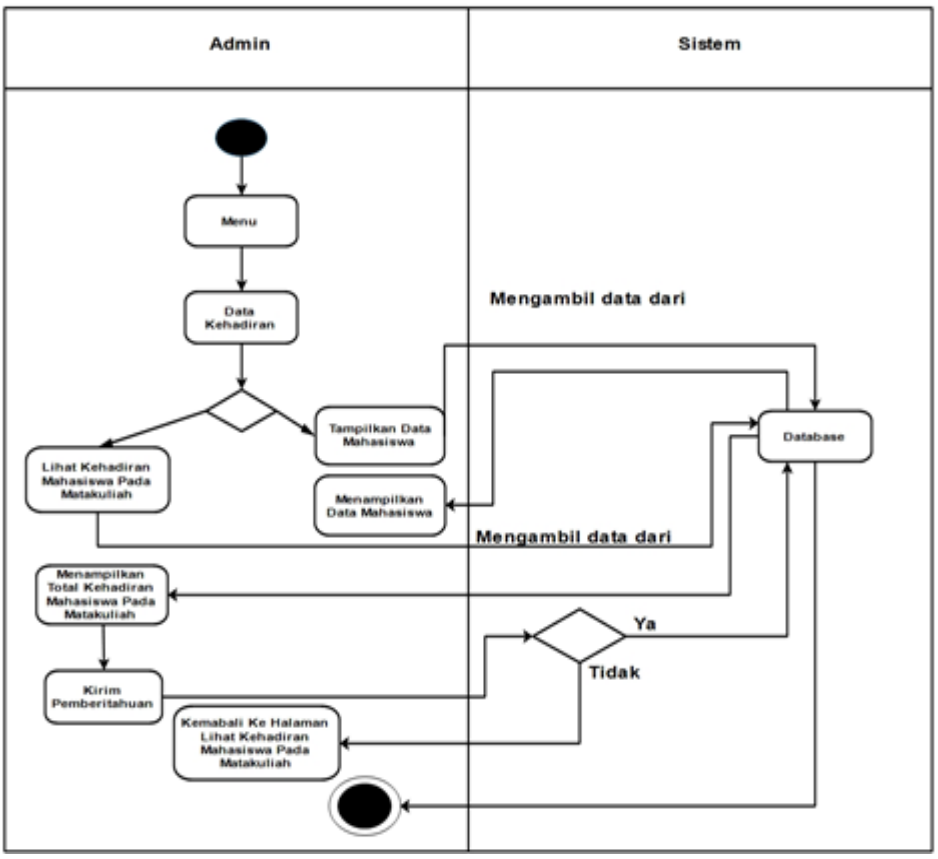
Activity diagram kelola data krs merupakan aktifitas admin mengelola data krs, dimana setelah admin *login* dan memilih menu atur data krs. Di dalam halaman atur data krs admin dapat menambahkan data, menampilkan data krs, menampilkan data mahasiswa yang mengambil matakuliah dan mencetak rekapan kehadiran mahasiswa pada 1 semester. Berikut ini *activity* diagram kelola data krs yang dapat dilihat pada gambar 26 :



Gambar 26. *Activity* Diagram Kelola Data KRS

6) *Activity* Diagram Data Kehadiran

Activity diagram data kehadiran merupakan aktifitas admin mengelola data kehadiran mahasiswa, dimana setelah admin *login* dan memilih menu data kehadiran. Admin dapat melihat data dari total kehadiran mahasiswa pada setiap matakuliah yang diambil oleh mahasiswa dan mengirimkan notifikasi/pemberitahuan ke salah satu mahasiswa atapun ke semua mahasiswa. Berikut ini *activity* diagram data kehadiran yang dapat dilihat pada gambar 27 :

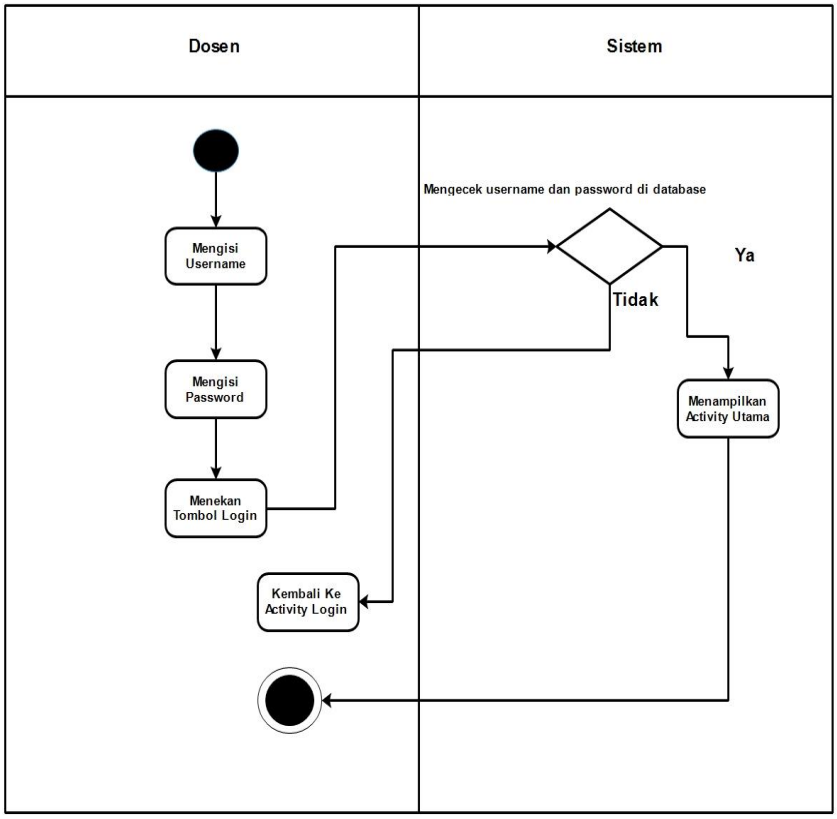


Gambar 27. *Activity* Diagram Data Kehadiran

c) *Activity* Diagram Dosen

1) *Activity* Diagram *Login* Dosen

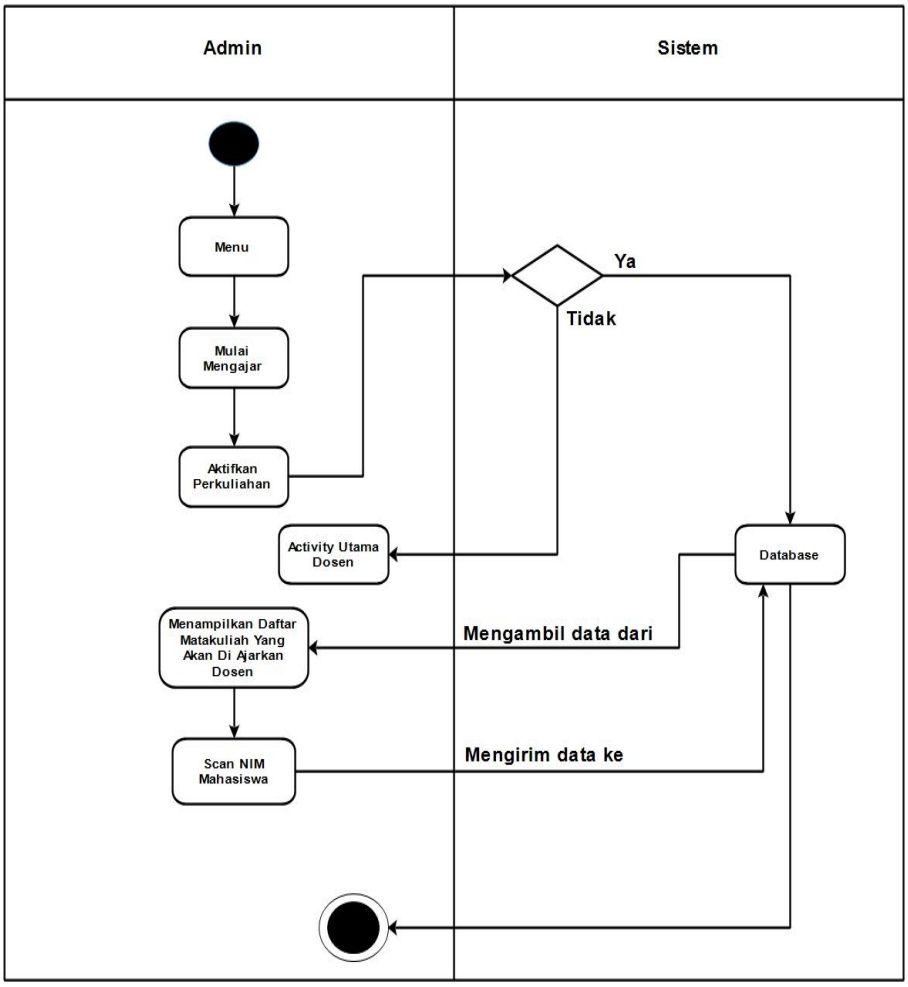
Dosen melakukan *login* dengan cara memasukkan *username* dan *password* pada *form login* lalu menekan tombol *login*. Sistem akan mengecek ke dalam database apakah data yang dimasukkan tersebut sesuai dengan yang ada di database. Jika data sesuai, maka sistem akan menampilkan *activity* utama dosen. Jika data tidak sesuai maka sistem akan kembali menampilkan *activity Login*. Berikut ini *activity* diagram *login* dosen yang dapat dilihat pada gambar 28 :



Gambar 28. *Activity* Diagram *Login* Dosen

2) *Activity* Diagram Mulai Mengajar

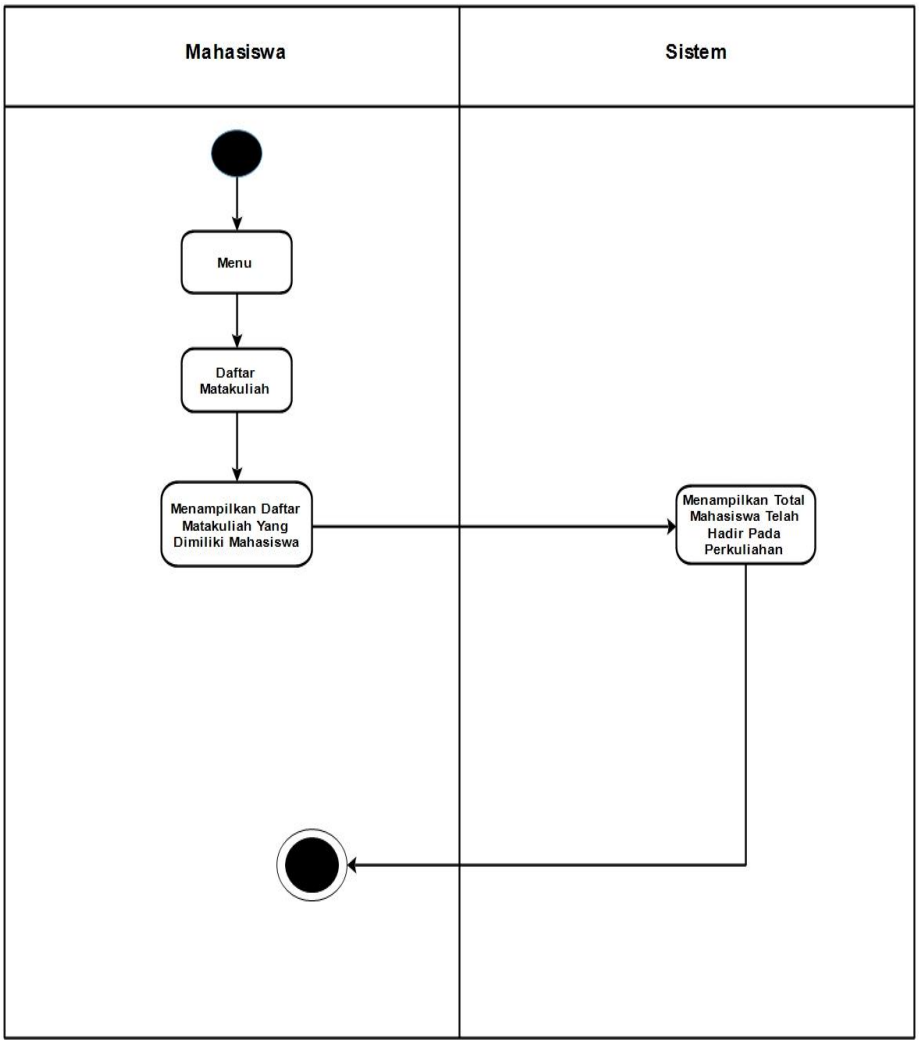
Setelah dosen berhasil melakukan *login* maka dosen dapat melakukan beberapa aktifitas di dalam sistem, seperti mulai mengajar, diantaranya mengaktifkan opsi mengajar, memilih matakuliah dan melakukan *scan* nim mahasiswa. Jika proses *scan* berhasil maka data akan disimpan ke dalam database. Berikut ini *activity* diagram mulai mengajar dosen yang dapat dilihat pada gambar 29 :



Gambar 29. *Activity* Diagram Mulai Mengajar

3) *Activity* Diagram Daftar Matakuliah Dosen

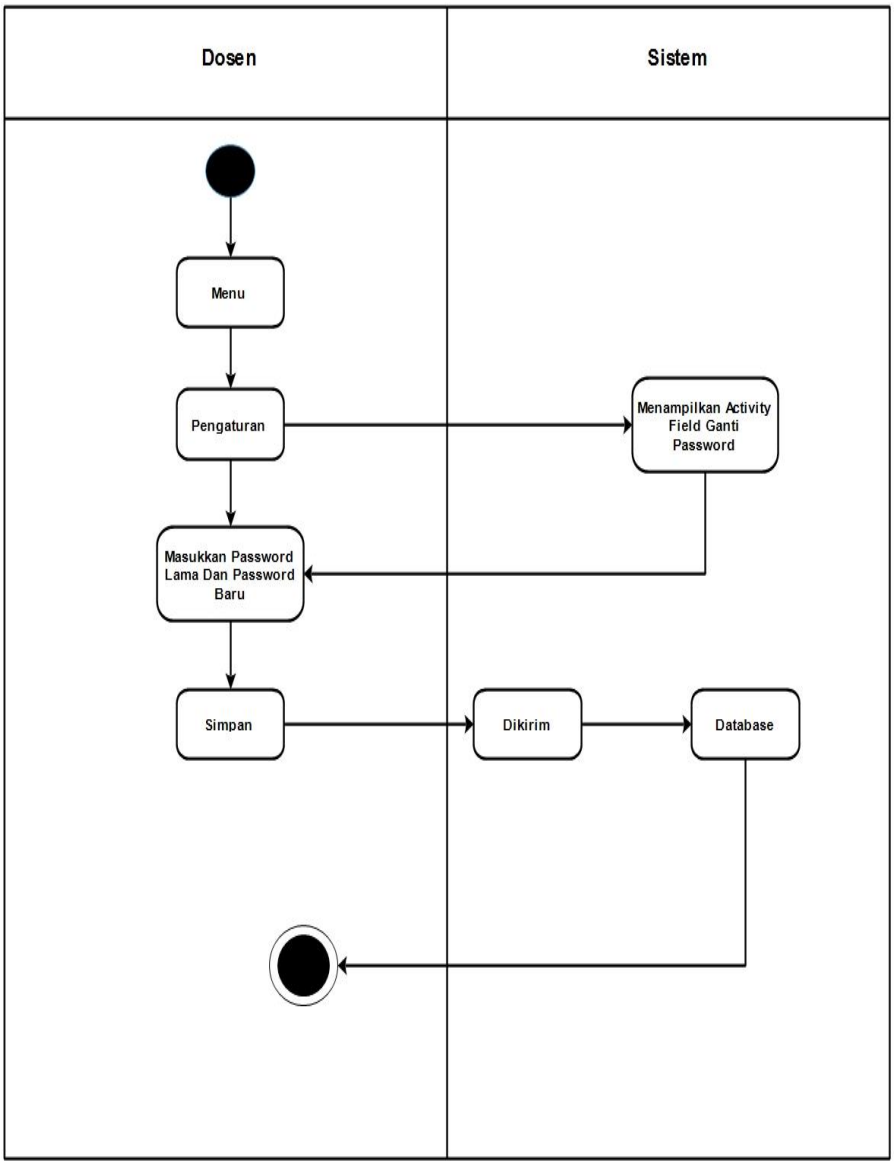
Activity diagram daftar matakuliah adalah aktifitas dosen untuk melihat daftar matakuliah yang akan diajarkan dosen selama 1 semester dan dapat menampilkan total dosen mengajar serta menampilkan nama-nama mahasiswa yang mengambil matakuliah dosen. Berikut ini *activity* diagram daftar matakuliah dosen yang dapat dilihat pada gambar 30 :



Gambar 30. *Activity* Diagram Daftar Matakuliah Dosen

4) *Activity* Diagram Pengaturan dosen

Activity diagram pengaturan adalah aktifitas dosen untuk mengganti *password* akun pengguna dosen yang akan digunakan ketika *login*. Berikut ini *activity* diagram pengaturan dosen yang dapat dilihat pada gambar 31 :



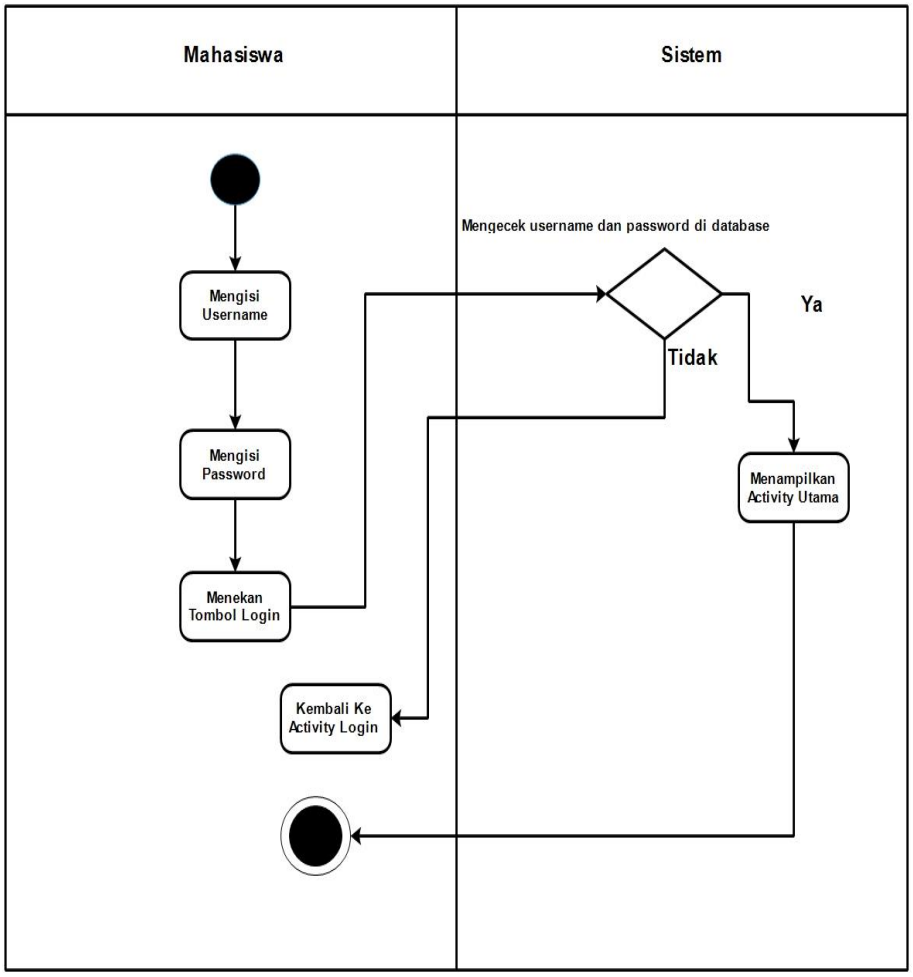
Gambar 31. *Activity* Diagram Pengaturan Dosen

d) *Activity* Diagram Mahasiswa

1) *Activity* Diagram *Login* Mahasiswa

Mahasiswa melakukan *login* dengan cara memasukkan *username* dan *password* pada form login lalu menekan tombol *login*. Sistem akan mengecek ke dalam database apakah data yang dimasukkan tersebut sesuai dengan yang ada di database. Jika data sesuai, maka sistem akan menampilkan *activity* utama

mahasiswa. Jika data tidak sesuai maka sistem akan kembali menampilkan *activity Login*. Berikut ini *activity* diagram *login* mahasiswa yang dapat dilihat pada gambar 32 :

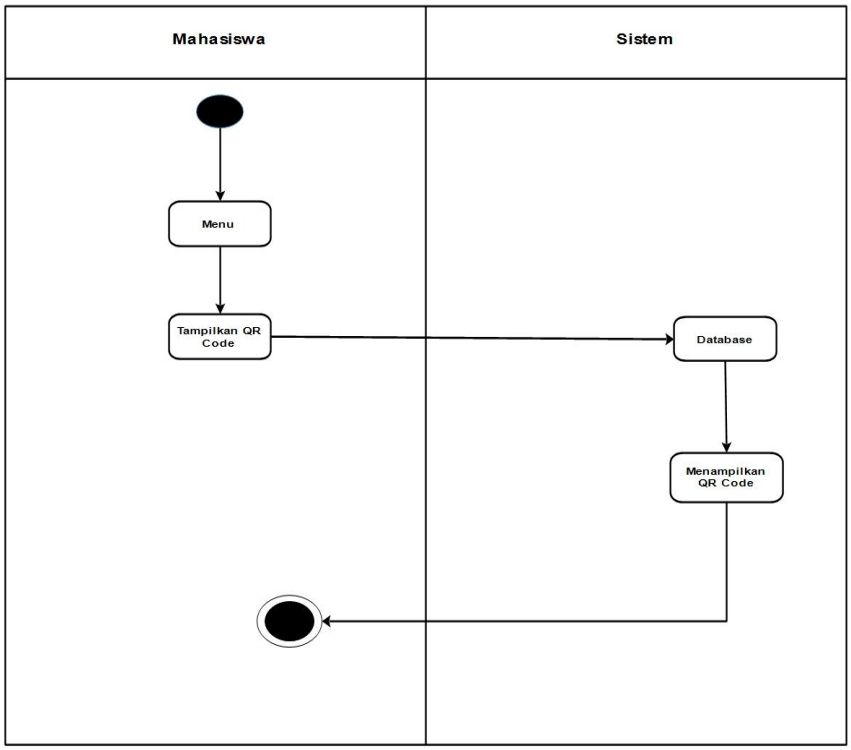


Gambar 32. *Activity* Diagram *Login* Mahasiswa

2) *Activity* Diagram Tampilkan *QR Code* Mahasiswa

Setelah Mahasiswa *Login*, mahasiswa menggunakan menu tampilkan *QR Code* untuk menampilkan gambar yang berasal dari kode nim mahasiswa. Penggunaan menu ini agar mahasiswa dapat melakukan proses kehadiran pada setiap matakuliah.

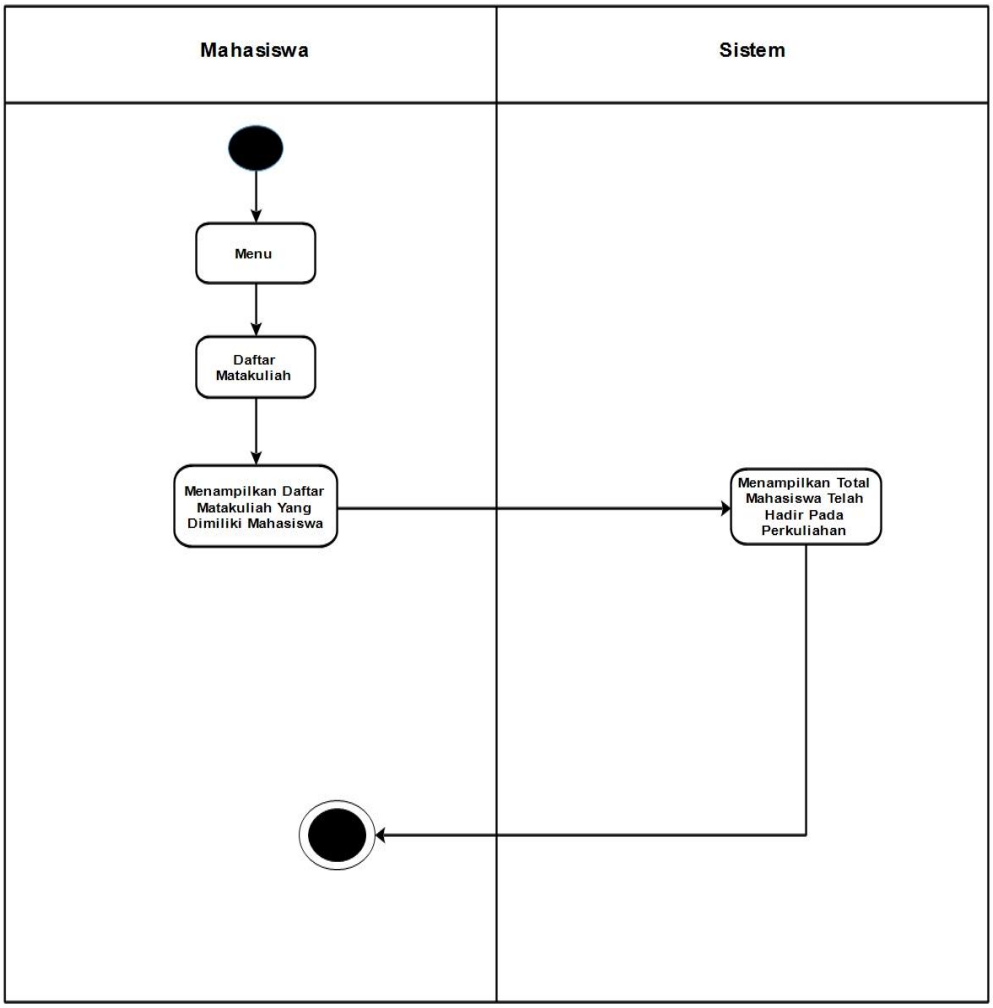
Berikut ini *activity* diagram tampilkan *QR Code* mahasiswa yang dapat dilihat pada gambar 33 :



Gambar 33. *Activity* Diagram Tampilkan *QR Code* Mahasiswa

3) *Activity* Diagram Daftar Matakuliah Mahasiswa

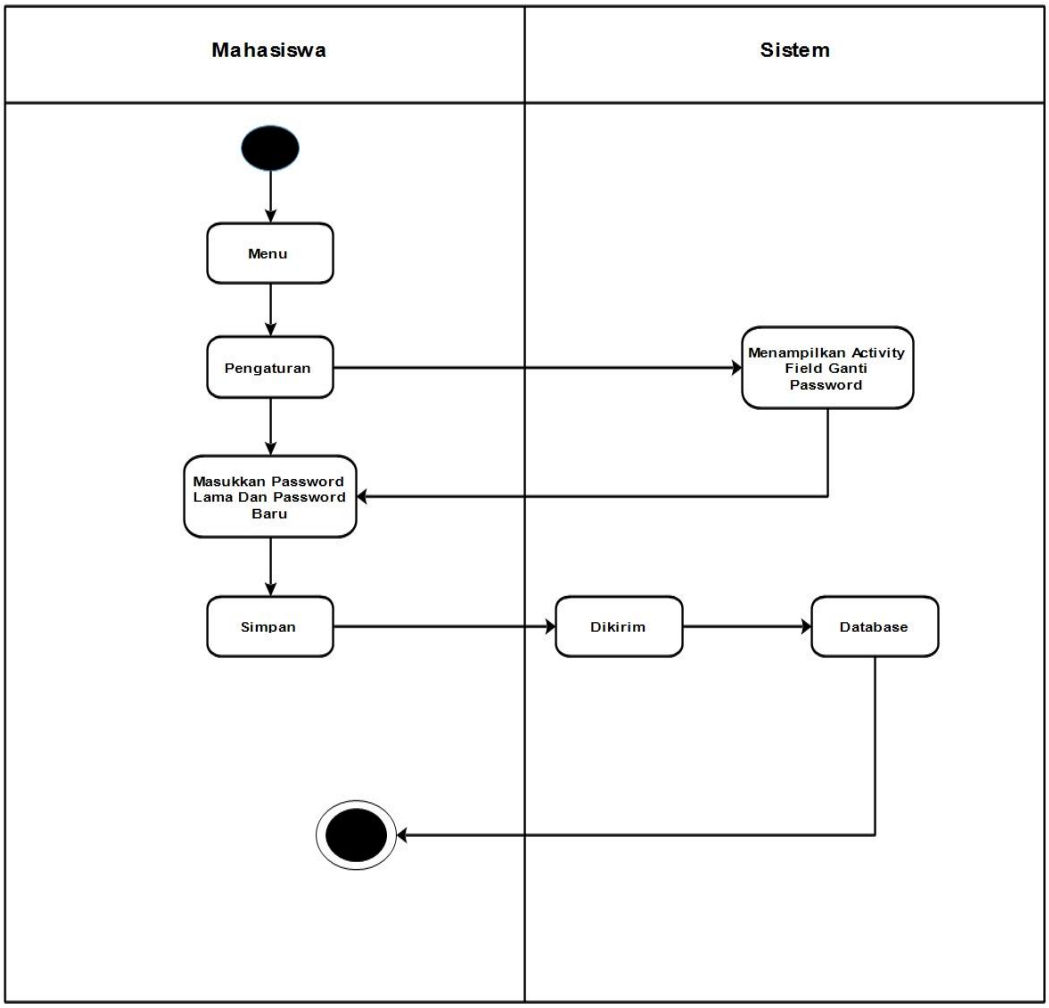
Activity diagram daftar matakuliah adalah aktifitas mahasiswa untuk melihat daftar matakuliah yang telah disusun mahasiswa pada krs untuk menghadiri perkuliahan selama 1 semester dan dapat menampilkan total mahasiswa telah hadir pada perkuliahan dari masing-masing matakuliah. Berikut ini *activity* diagram daftar matakuliah mahasiswa yang dapat dilihat pada gambar 34 :



Gambar 34. *Activity* Diagram Daftar Matakuliah Mahasiswa

4) *Activity* Diagram Pengaturan Mahasiswa

Activity diagram pengaturan adalah aktifitas mahasiswa untuk mengganti *password* akun pengguna mahasiswa yang akan digunakan ketika *login*. Berikut ini *activity* diagram pengaturan mahasiswa yang dapat dilihat pada gambar 35 :

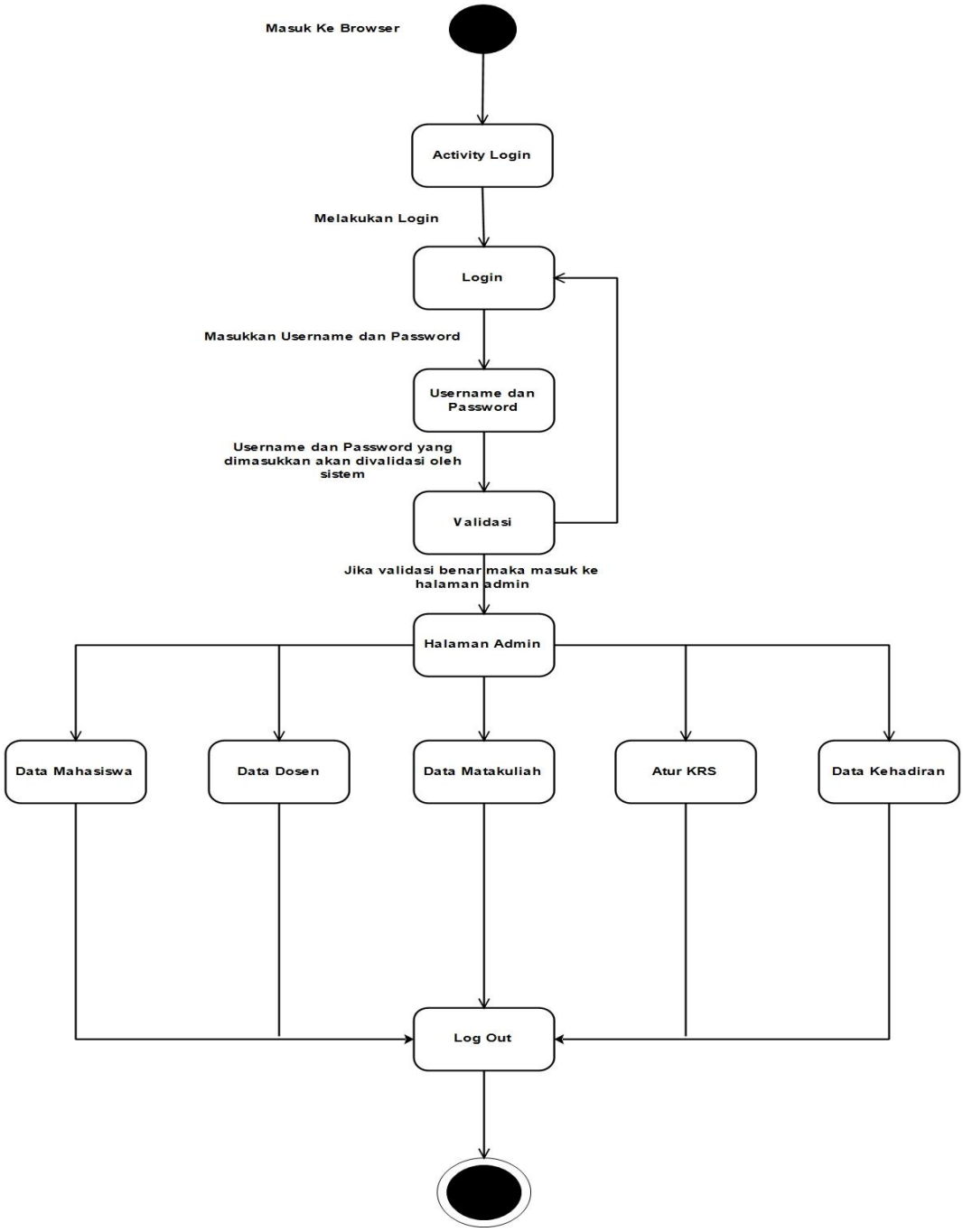


Gambar 35. *Activity* Diagram Pengaturan Mahasiswa

e) *State Machine* Diagram

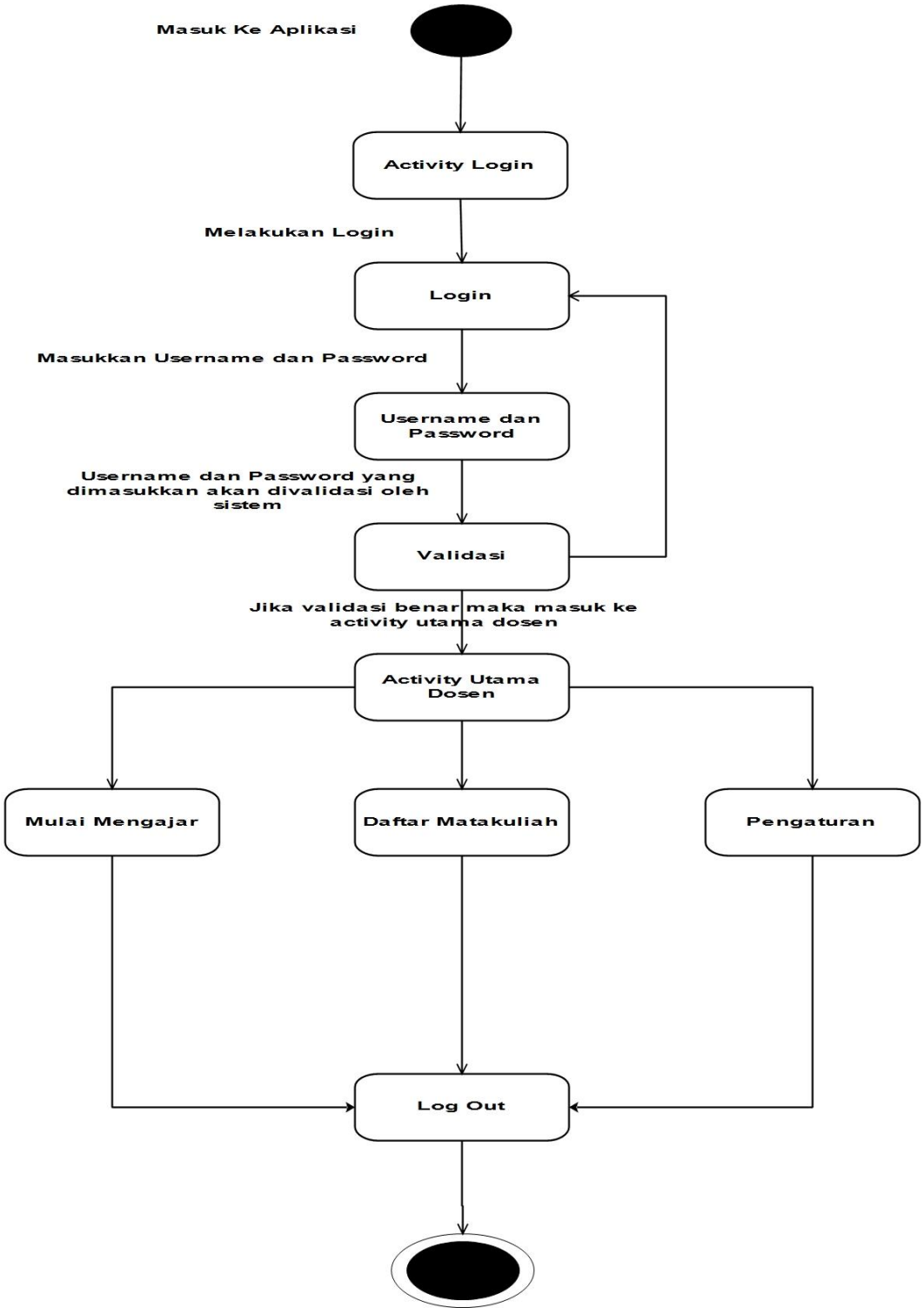
State diagram digunakan untuk mendeskripsikan prilaku sistem. *State* diagram mendeskripsikan semua kondisi yang mungkin muncul sebagai sebuah *object* begitu pula *event*. *State Machine* diagram sistem yang diusulkan dapat dilihat pada gambar di bawah ini:

1) *State Machine* Diagram Admin



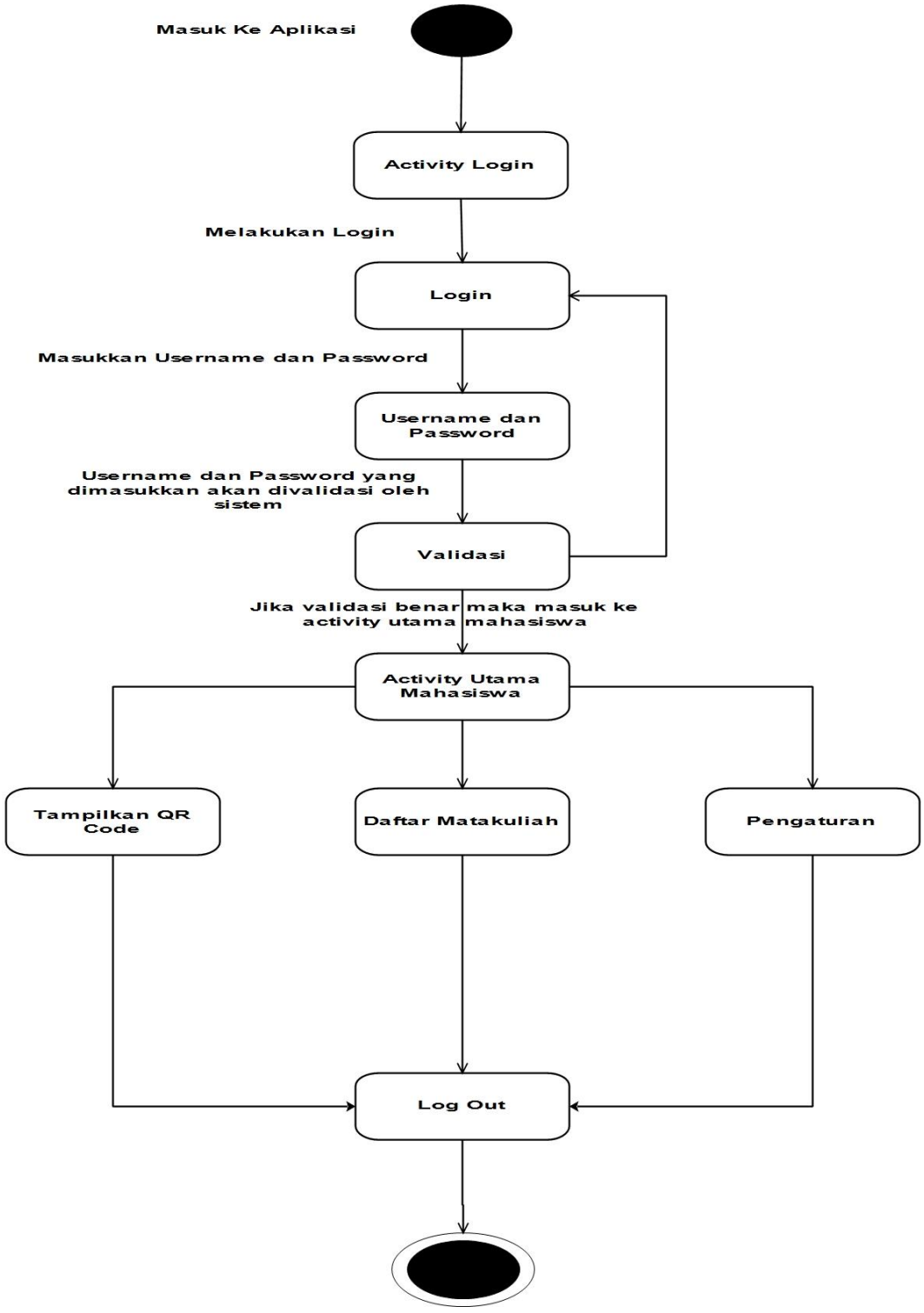
Gambar 36. *State Machine* Diagram Admin

2) *State Machine* Diagram Dosen



Gambar 37. *State Machine* Diagram Dosen

3) *State Machine* Diagram Mahasiswa

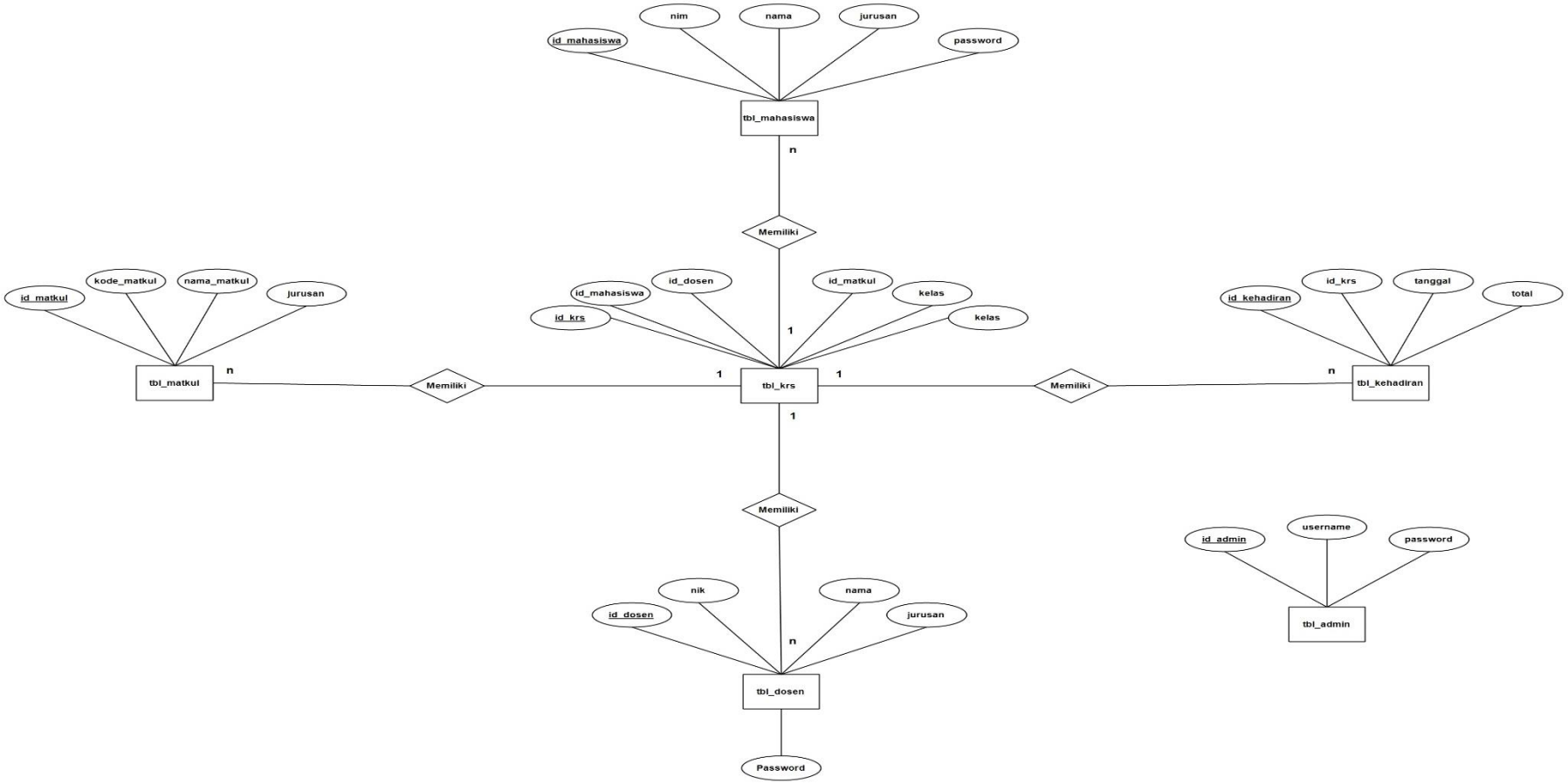


Gambar 38. *State Machine* Diagram Mahasiswa

6. Desain Basis Data

a) Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan notasi yang digunakan dalam melakukan aktivitas pemodelan data. Dalam perancangan desain basis data diperlukannya *Entity Relationship diagram* (ERD) untuk memudahkan dalam membuat struktur tabel basis data. Berikut adalah gambar dari ERD Aplikasi Kehadiran Mahasiswa Menggunakan *Barcode* Bebasis *Android* yang dapar dilihat pada 39:



Gambar 39. ERD

b) Desain Tabel

Di dalam perancangan aplikasi kehadiran pada STMIK Palangkaraya berbasis *android* terdapat beberapa struktur tabel basis data yang akan digunakan sebagai tempat menyimpan data, tabel tersebut diantaranya dapat dilihat pada tabel berikut :

1) Tabel Admin

Nama Tabel : tbl_admin

Primary Key : id_dosen

Tabel 14. Desain Basis Data Admin

No	Field	Type	Size	Ket
1	id_admin	int	20	Id pembeda admin
2	username	varchar	20	Nama pengguna ketika admin login
3	password	varchar	20	Password pengguna yang digunakan ketika admin login

2) Tabel dosen

Nama Tabel : tbl_dosen

Primary Key : id_dosen

Tabel 15. Desain Basis Data Dosen

No	Field	Type	Size	Ket
1	2	3	4	5
1	id_dosen	int	225	Id pembeda dosen

1	2	3	4	5
2	nik	varchar	20	Nomor induk kepegawaian yang dimiliki setiap dosen
3	nama	varchar	50	Nama dosen
4	jurusan	varchar	20	Jurusan dosen
5	password	varchar	50	Password dosen yang digunakan ketika login

3) Tabel Mahasiswa

Nama Tabel : tbl_mahasiswa

Primary Key : id_mahasiswa

Tabel 16. Desain Basis Data Mahasiswa

No	Field	Type	Size	Ket
1	2	3	4	5
1	id_mahasiswa	int	225	Id pembeda dosen
2	nim	varchar	100	Nomor induk mahasiswa yang dimiliki setiap mahasiswa
3	nama	varchar	50	Nama mahasiswa
4	jurusan	varchar	20	Jurusan mahasiswa

1	2	3	4	5
5	password	varchar	50	Password mahasiswa yang digunakan ketika login
6	regId	varchar	500	Id yang dimiliki mahasiswa ketika berhasil login pada aplikasi android, yang berfungsi sebagai pengenal perangkat agar dapat menerima notifikasi

4) Tabel Matakuliah

Nama Tabel : tbl_matkul

Primary Key : id_matkul

Tabel 17. Desain Basis Data Matakuliah

No	Field	Type	Size	Ket
1	2	3	4	5
1	id_matkul	int	20	Id pembeda matakuliah

1	2	3	4	5
2	kode_matkul	varchar	20	Kode yang dimiliki matakuliah
3	nama_matkul	varchar	50	Nama mahasiswa
4	jurusan	varchar	20	Jurusan matakuliah

5) Tabel KRS

Nama Tabel : tbl_krs

Primary Key : id_krs

Tabel 18. Desain Basis Data KRS

No	Field	Type	Size	Ket
1	id_krs	int	20	Id pembeda krs
2	id_mahasiswa	varchar	20	Id mahasiswa
3	id_dosen	varchar	50	Id mahasiswa
4	id_matkul	varchar	20	Id matakuliah
5	kelas	varchar	5	Kelas matakuliah
6	tahun	varchar	12	Tahun akademik matakuliah

6) Tabel Kehadiran

Nama Tabel : tbl_kehadiran

Primary Key : id_kehadiran

Tabel 19. Desain Basis Data Kehadiran

No	Field	Type	Size	Ket
1	id_kehadiran	int	20	Id kehadiran
2	nim	varchar	50	Nim mahasiswa
3	nik	varchar	50	Nik mahasiswa
4	id_matkul	varchar	20	Id matakuliah
5	id_krs	varchar	20	Id krs
6	kelas	varchar	5	Kelas matakuliah
7	tahun	varchar	5	tahun
8	tanggal	timestamp		Tanggal melakukan *scan* kehadiran
9	total	int	20	Total mahasiswa hadir

7. Desain Kemanan

Keamanan basis data harus diperhatikan dalam sistem ini, maka diperlukan penyedia layanan hosting yang dapat dipercaya dalam hal keamanan data dan aplikasi. Salah satu penyedia layanan hosting yang penulis gunakan adalah hosteko.com, penyedia jasa hosting ini memiliki server yang stabil. Sehingga ketika server tidak mudah mengalami *down* ketika digunakan meload data.

Keamanan proses absensi ketika mengajar juga harus dijaga karena dengan mahasiswa maju kedepan ketika proses absensi, dosen dapat melihat apakah mahasiswa benar-benar melakukan absensi sendiri atau titip absen. Jika dosen melakukan banyak mengajar dalam 1 hari untuk menutup kekurangan jam mengajar, maka admin dapat melakukan pengecekan history dosen mengajar pada web admin.

8. Desain Sumber Daya Manusia (SDM)

Aplikasi ini dibuat untuk memudahkan proses absensi mahasiswa dan dosen, serta memudahkan admin untuk melakukan perekapan data kehadiran mahasiswa pada 1 semester. Maka dari itu sumber daya manusia yang menggunakan aplikasi ini adalah dosen, mahasiswa dan admin. Dimana admin memiliki hak akses penuh terhadap sistem aplikasi ini, mulai dari mengelola data hingga mengirimkan notifikasi ke mahasiswa.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

A. Implementasi

Implementasi merupakan tahap pengembangan rancangan menjadi progam dan program dapat berjalan sesuai rancangan. Berikut adalah penjelasan untuk beberapa proses implementasi yang penulis lakukan :

1. Uji Coba Program

Uji coba aplikasi dan *source code* perangkat lunak dari aplikasi kehadiran mahasiswa menggunakan *barcode* berbasis *android*.

a. Rencana Pengujian *Black Box*

Dalam pengujian sistem ini, pengujian dilakukan dengan metode *black box* testing dimana tahap pengujian ini berfokus pada kebutuhan fungsional dari program. Pengujian sistem pada program ini menggunakan data uji berupa data masukkan. Berikut rencana pengujian *Black Box* bisa di lihat pada Tabel.

Tabel 20. Rencana Pengujian *Black Box*

Pengujian	Kriteria Evaluasi Hasil
1	2
Pengujian membuka dan menutup aplikasi	Saat pengguna membuka aplikasi, aplikasi akan terbuka serta menampilkan halaman login Dan aplikasi akan tertutup saat menyentuh tombol kembali.
Pengujian terhadap *login* aplikasi android	Saat pengguna membuka aplikasi *android* akan menampilkan *login* yaitu halaman yang digunakan pengguna untuk *login*.

1	2
Pengujian terhadap menu tampilkan *QR Code* mahasiswa	Saat mahasiswa telah *login* dan menekan menu tampilkan *QR Code* maka akan menampilkan *QR Code* yang telah degenerate otomatis dari NIM mahasiswa.
Pengujian terhadap menu daftar mata kuliah mahasiswa	Saat mahasiswa menekan menu daftar matakuliah maka di alihkan ke *activity* daftar matakuliah dan menampilkan daftar matakuliah yang di ambil oleh mahasiswa.
Pengujian terhadap menu pengaturan mahasiswa	Saat mahasiswa menekan menu pengaturan maka akan dialihkan ke *activity* pengaturan yang digunakan untuk mengganti *password* mahasiswa.
Pengujian terhadap *logout* mahasiswa	Saat mahasiswa menekan *logout* maka akan dialihkan ke *activity login*.
Pengujian terhadap menu mulai mengajar dosen	Saat dosen menekan menu mulai mengajar maka akan dialihkan ke *activity* aktifkan matakuliah dan setelah memilih mulai maka akan terpindah ke *activity* pilih matakuliah yang digunakan untuk memilih matakuliah yang akan diajarkan serta untuk melakukan *scan* nim mahasiswa.
Pengujian terhadap menu daftar matakuliah dosen	Saat dosen menekan menu daftar matakuliah maka di alihkan ke *activity* daftar matakuliah dan menampilkan daftar matakuliah yang akan diajarkan oleh dosen.
Pengujian terhadap menu pengaturan dosen	Saat dosen menekan menu pengaturan maka akan dialihkan ke *activity* pengaturan yang digunakan untuk mengganti password dosen.
Pengujian terhadap *logout* dosen	Saat dosen menekan *logout* maka akan dialihkan ke *activity login*.
Pengujian mengirim notifikasi ke mahasiswa	Saat admin masuk ke dalam web *backend*, admin dapat melakukan pengiriman notifikasi ke salah satu mahasiswa atapun semua mahasiswa.
Pengujian melihat laporan absensi/kehadiran mahasiswa	Saat admin masuk ke dalam web *backend*, admin dapat melihat, menyimpan ataupun melakukan pencetakan laporan kehadiran mahasiswa.

1	2
Pengujian melihat histori dosen mengajar	Saat admin masuk ke dalam web *backend*, admin dapat melihat histori dosen mengahjar. Melihat nim siapa saja yang telah di scan serta tanggal dan jam melakukan mengajar akan tampil.

b. Pengujian Sistem

Tabel 21. Pengujian Terhadap Mahasiswa

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka dan memulai aplikasi	Buka aplikasi kehadiran pada ponsel yang telah di instal aplikasi kehadiran	Aplikasi akan terbuka dan menampilkan halaman login	
Menutup aplikasi	Sentuh tombol *back* padaa ponsel	Aplikasi akan tertutup	

Tabel 22. Pengujian Halaman Login

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Masukkan Id pengguna	Sentuh *edit text* pada halaman *login* aplikasi	*Edit text* dapat di isi dengan id pengguna	
Masukkan *password*	Sentuh *edit text* pada halaman *login* aplikasi	*Edit text* dapat di isi dengan *password* pengguna	

1	2	3	4
Tombol *login*	Sentuh tombol *login* pada halaman *login*	Jika id pengguna dan *password* yang dimasukkan benar maka akan dialihkan ke halaman utama dosen/mahasiswa sedangkan jika *password* salah maka akan ada pemberitahuan jika *password* yang dimasukkan salah	

Tabel 23. Pengujian Halaman Utama Dosen

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Menu mulai mengajar	Sentuh menu mulai mengajar pada halaman utama dosen	Halaman akan dialihkan ke halaman mulai mengajar	
Menu daftar matakuliah	Sentuh menu daftar matakuliah pada halaman utama dosen	Maka akan menampilkan beberapa matakuliah yang diajarkan dosen selama 1 semester	
Menu pengaturan	Sentuh menu pengaturan pada halaman utama dosen	Halaman akan dialihkan ke halaman pengaturan	
Menu keluar	Sentuh menu keluar pada halaman utama dosen	Maka akan dialihkan ke halaman login	

Tabel 24. Pengujian Halaman Utama Mahasiswa

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Menu tampilkan *qr code*	Sentuh menu tampilkan qr code pada halaman utama mahasiswa	Halaman akan dialihkan ke halaman tampilkan *qr code*	
Menu daftar matakuliah	Sentuh menu daftar matakuliah pada halaman utama mahasiswa	Maka akan menampilkan beberapa matakuliah yang diprogramkan oleh mahasiswa selama 1 semester	
Menu pengaturan	Sentuh menu pengaturan pada halaman utama mahasiswa	Halaman akan dialihkan ke halaman pengaturan	
Menu keluar	Sentuh menu keluar pada halaman utama mahasiswa	Maka akan dialihkan ke halaman login	

Tabel 25. Pengujian Halaman Mulai Mengajar

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka menu mulai mengajar	Sentuh menu mulai mengajar pada halaman menu utama dosen	Halaman akan dialihkan ke halaman mulai mengajar	
Pilih *radio button*	Sentuh salah satu *radio button* (Mulai/Akhiri)	Salah satu *radio button* akan terpilih	

1	2	3	4
Tombol proses	Sentuh tombol proses	Jika yang dipilih mulai maka akan menampilkan beberapa daftar matakuliah, sedangkan jika yang dipilih akhiri maka akan kembali ke menu utama dosen.	
Pilih matakuliah	Sentuh salah satu matakuliah	Kamera ponsel otomatis akan terbuka	
Melakukan *scan qr code*	Arahkan kamera ponsel ke gambar nim yang telah menjadi *qr code*	Kamera ponsel akan menyimpan nim yang telah di *scan* ke *database*	

Tabel 26. Pengujian Halaman Daftar Matakuliah Dosen

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka menu daftar matakuliah	Sentuh menu daftar matakuliah pada halaman menu utama dosen	Maka akan menampilkan beberapa matakuliah yang di ajarkan dosen selama 1 semester	
Menampilkan total mengajar dan nama mahasiswa pada matakuliah	Sentuh salah satu matakuliah yang ada	Maka akan menampilkan beberapa nama mahasiswa yang mengambil matakuliah tersebut dan total dosen telah mengajar	

Tabel 27. Pengujian Halaman Tampilkan *QR Code*

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka menu tampilkan *qr code*	Sentuh menu tampilkan *qr code* pada halaman menu utama mahasiswa	Maka akan menampilkan gambar *qr code* yang berasal dari nim mahasiswa	

Tabel 28. Pengujian Halaman Daftar Matakuliah Mahasiswa

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka menu daftar matakuliah	Sentuh menu daftar matakuliah pada halaman menu utama mahasiswa	Maka akan menampilkan beberapa matakuliah yang diambil mahasiswa selama 1 semester	
Melihat total hadir mahasiswa dan nama dosen pengampu pada matakuliah	Sentuh salah satu matakuliah yang ada	Maka akan menampilkan total kehadiran mahasiswa dan nama dosen yang mengajar pada matakuliah tersebut	

Tabel 29. Pengujian Keluar Akun Pengguna Dosen dan Mahasiswa

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka menu keluar	Sentuh menu keluar pada halaman menu utama mahasiswa/dosen	Maka akan menampilkan halaman *login*	

Tabel 30. Pengujian Mengirim Notifikasi Ke Mahasiswa

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka halaman web back end	Login kedalam halaman web back end admin	Maka akan dapat mengakses sebagai admin untuk mengelola data	
Membuka menu data kehadiran	Klik menu data kehadiran	Maka akan menampilkan daftar nama mahasiswa	
Membuka lihat kehadiran	Klik tombol lihat kehadiran pada salah satu mahasiswa	Akan menampilkan daftar matakuliah yang diambil oleh mahasiswa	
Membuka lihat total	Klik tombol lihat total pada salah satu matakuliah	Total hadir mahasiswa pada matakuliah tersebut akan tampil	
Tombol kirim pesan	Klik tombol kirim pesan	Maka akan menampilkan halaman untuk mengirim notifikasi	
Tombol kirim	Isi judul dan pesan lalu klik kirim	Notifikasi akan terkirim ke mahasiswa yang telah dipilih	

Tabel 31. Pengujian Melihat Laporan Kehadiran Mahasiswa

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka menu atur krs	Klik menu atur krs	Maka akan menampilkan data krs	
Tombol cetak	Klik tombol cetak pada tabel krs	Laporan absensi akan tampil	

Tabel 32. Pengujian Melihat Histori Mengajar

Pengujian	Sekenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	2	3	4
Membuka menu atur krs	Klik menu atur krs	Maka akan menampilkan data krs	
Tombol hostori	Klik tombol histori pada tabel krs	Riwayat dosen mengajar akan tampil dalam bentuk tabel	

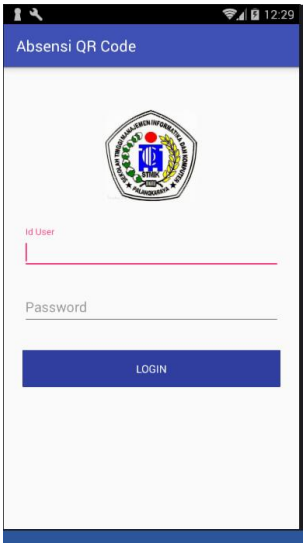
2. Manual Program

Manual program merupakan tahapan untuk menjelaskan cara penggunaan atau pengoperasian aplikasi kehadiran mahasiswa menggunakan *barcode* berbasis android. Tahapan yang akan dilakukan dalam proses manual program ini dibagi menjadi tiga kategori yaitu mahasiswa, dosen dan admin antara lain :

a. Penjelasan Manual Program Mahasiswa

1) Tampilan Login Mahasiswa

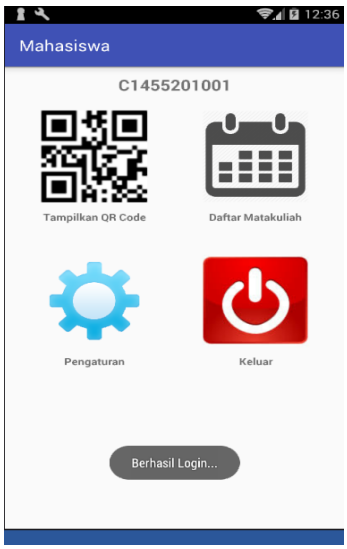
Tampilan Login Mahasiswa berisi memiliki gambar, field dan tombol seperti logo STMIK, Id User, Password dan Login. Seperti yang terlihat pada gambar 40 :



Gambar 40. Tampilan Login Mahasiswa

2) Tampilan Menu Utama Mahasiswa

Tampilan Menu Utama Mahasiswa akan tampil jika mahasiswa telah berhasil melakukan login, tampilan ini berisikan field dan beberapa menu pilihan seperti NIM mahasiswa yang melakukan login, Tampilkan QR Code, Daftar Matakuliah, Pengaturan dan Keluar. Seperti yang terlihat pada gambar 41 :



Gambar 41. Tampilan Menu Utama Mahasiswa

3) Tampilan Menu Tampilkan QR Code

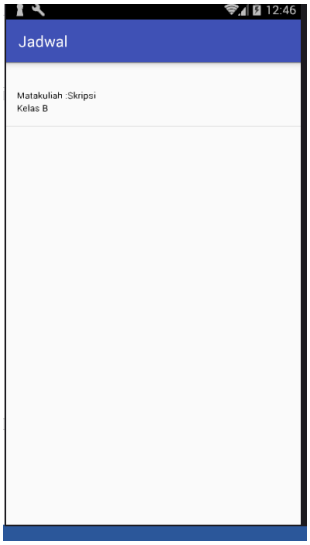
Tampilan Menu Tampilkan QR Code akan tampil jika mahasiswa telah membuka menu Tampilkan QR Code, tampilan ini akan menampilkan gambar qr code yang di generate secara otomatis dari nim mahasiswa yang digunakan untuk login. Seperti yang terlihat pada gambar 42 :



Gambar 42. Tampilan Menu Tampilkan QR Code

4) Tampilan Menu Daftar Matakuliah

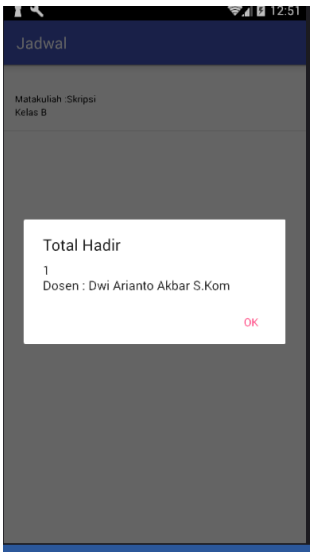
Tampilan Menu Daftar Matakuliah akan tampil jika mahasiswa telah membuka menu Daftar Matakuliah, tampilan ini akan menampilkan nama matakuliah dan kelas yang diambil oleh mahasiswa. Seperti yang terlihat pada gambar 43 :



Gambar 43. Tampilan Menu Daftar Matakuliah

5) Tampilan Toast Total Hadir

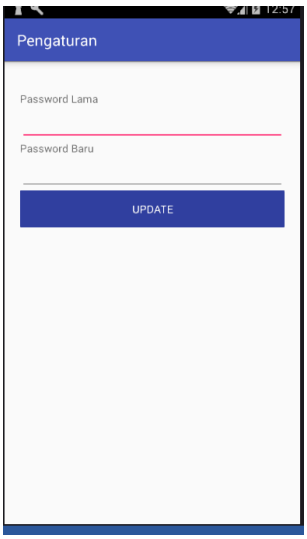
Tampilan Toast Total Hadir akan tampil jika mahasiswa menyentuh salah satu matakuliah yang dipilih, tampilan ini akan menampilkan total hadir mahasiswa dan nama dosen pengampu dari matakuliah yang dimiliki oleh mahasiswa. Seperti yang terlihat pada gambar 44 :



Gambar 44. Tampilan Toast Daftar Hadir

6) Tampilan Menu Pengaturan

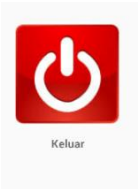
Tampilan Menu Pengaturan akan tampil jika mahasiswa membuka menu pengaturan, tampilan ini akan menampilkan beberapa edit text dan tombol. Tampilan menu pengaturan memiliki edit text password lama, password baru dan tombol update. Seperti yang terlihat pada gambar 45 :



Gambar 45. Tampilan Menu Pengaturan

7) Tampilan Menu Keluar

Tampilan Menu Keluar akan membuat mahasiswa keluar dari akun mahasiswa dan kembali ke tampilan login mahasiswa. Tampilan menu keluar memiliki gambar dan teks. Seperti yang terlihat pada gambar 46 :

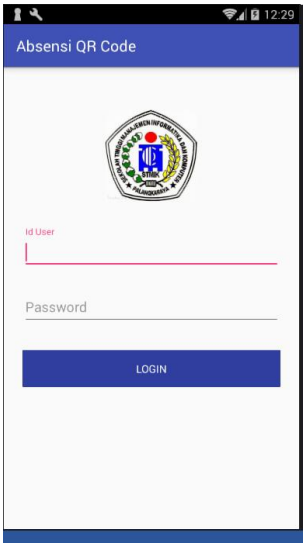


Gambar 46. Tampilan Menu Pengaturan

b. Penjelasan Manual Program Dosen

1) Tampilan Login Dosen

Tampilan Login Dosen berisi memiliki gambar, field dan tombol seperti logo STMIK, Id User, Password dan Login. Seperti yang terlihat pada gambar 47 :



Gambar 47. Tampilan Login Dosen

2) Tampilan Menu Utama Dosen

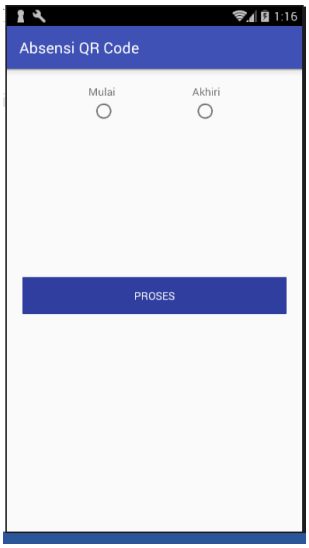
Tampilan Menu Utama Dosen akan tampil jika dosen telah berhasil melakukan login, tampilan ini berisikan field dan beberapa menu pilihan seperti NIK dosen yang melakukan login, Mulai Mengajar, Daftar Matakuliah, Pengaturan dan Keluar. Seperti yang terlihat pada gambar 48 :



Gambar 48. Tampilan Menu Utama Dosen

3) Tampilan Menu Mulai Mengajar

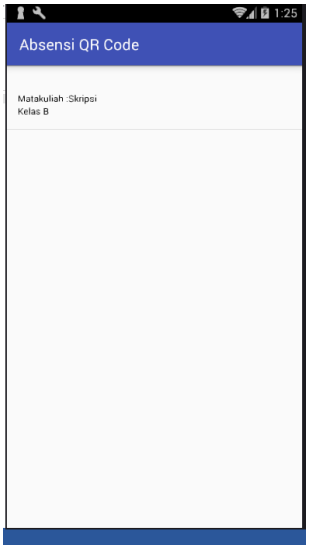
Tampilan Menu Mulai Mengajar akan tampil jika dosen telah membuka menu Mulai Mengajar, tampilan ini akan menampilkan radio button dan tombol. Seperti yang terlihat pada gambar 49 :



Gambar 49. Tampilan Menu Mulai Mengajar

4) Tampilan Matakuliah Mulai Mengajar

Tampilan Matakuliah Mulai Mengajar akan tampil jika dosen telah menekan tombol proses, tampilan ini akan menampilkan daftar matakuliah yang dimiliki dosen. Jika matakuliah dipilih akan langsung membuka kamera. Seperti yang terlihat pada gambar 50 :



Gambar 50. Tampilan Matakuliah Mulai Mengajar

5) Tampilan Menu Daftar Matakuliah

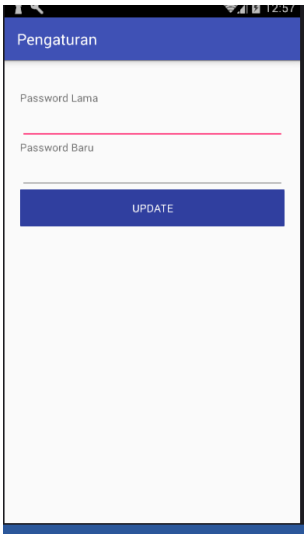
Tampilan Menu Daftar Matakuliah akan tampil jika dosen telah membuka menu Daftar Matakuliah, tampilan ini akan menampilkan nama matakuliah dan kelas yang dimiliki oleh dosen. Seperti yang terlihat pada gambar 51 :



Gambar 51. Tampilan Menu Daftar Matakuliah

6) Tampilan Pengaturan

Tampilan Menu Pengaturan akan tampil jika dosen membuka menu pengaturan, tampilan ini akan menampilkan beberapa edit text dan tombol. Tampilan menu pengaturan memiliki edit text password lama, password baru dan tombol update. Seperti yang terlihat pada gambar 52 :



Gambar 52. Tampilan Menu Pengaturan

7) Tampilan Menu Keluar

Tampilan Menu Keluar akan membuat dosen keluar dari akun dosen dan kembali ke tampilan login dosen. Tampilan menu keluar memiliki gambar dan teks. Seperti yang terlihat pada gambar 53 :

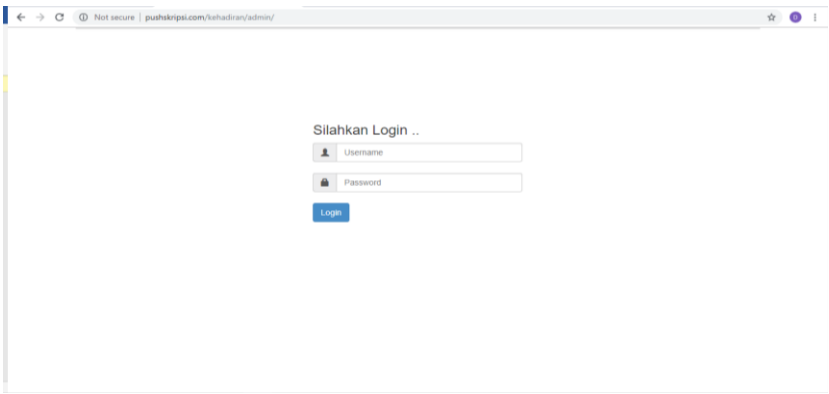


Gambar 53. Tampilan Menu Pengaturan

c. Penjelasan Manual Program Admin

1) Halaman *Login* Admin

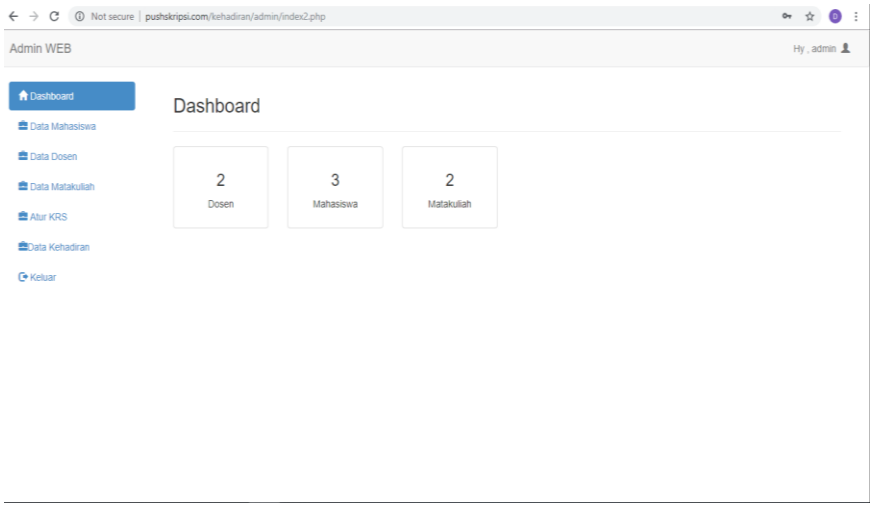
Halaman ini berguna agar admin dapat masuk kedalam sistem serta mengelola data mahasiswa, dosen, matakuliah, krs, dll. Admin diharuskan memasukkan username dan password, jika *username* atau *password* yang dimasukkan benar maka akan dialihkan ke halaman dashboard admin namun jika salah maka akan ada pemberitahuan jika *login* gagal. Seperti yang terlihat pada gambar 54 :



Gambar 54. Halaman Login Admin

2) Halaman Dashboard

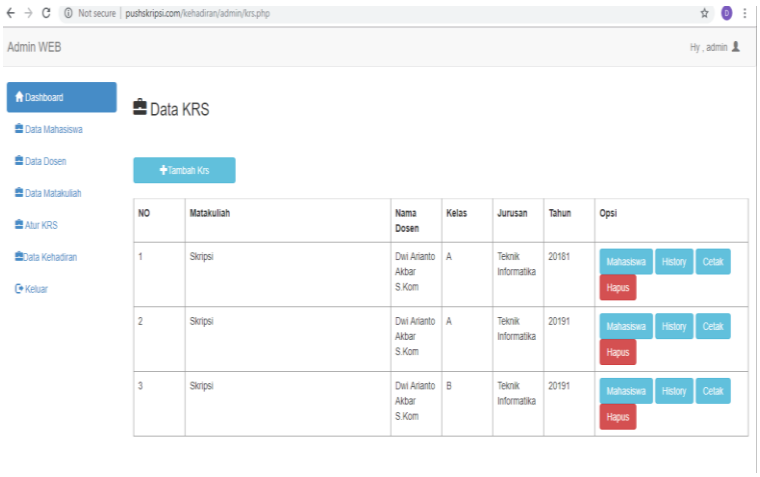
Halaman dashboard akan tampil jika admin berhasil masuk kedalam sistem. Pada halaman ini berisikan menu dashboard, data mahasiswa, data dosen, data matakuliah, atur krs, data kehadiran dan keluar. Seperti yang terlihat pada gambar 55 :



Gambar 55. Halaman Dashboard Admin

3) Halaman Atur KRS

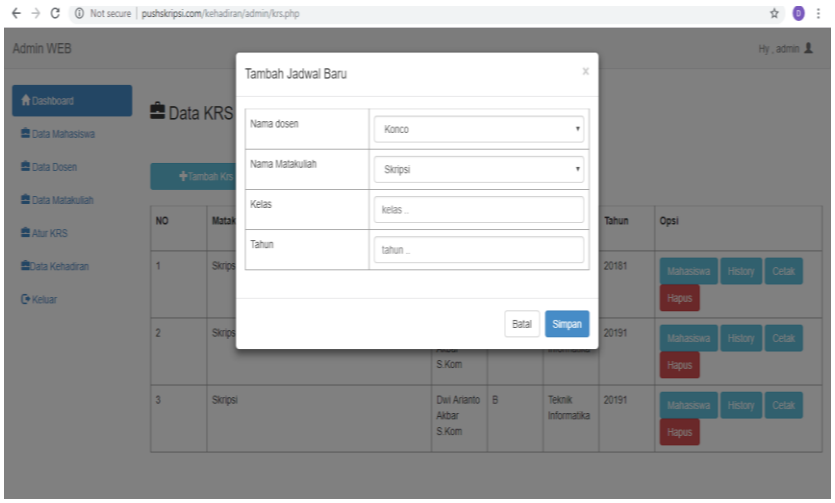
Halaman ini digunakan admin untuk membuat data krs agar kegiatan absensi dapat berjalan pada aplikasi. Data yang ditampilkan pada halaman ini adalah matakuliah, nama dosen, kelas, jurusan, tahun serta beberapa tombol seperti tambah krs, mahasiswa history cetak dan hapus. Seperti yang terlihat pada gambar 56 :



Gambar 56. Halaman Atur KRS

4) *Form* Tambah KRS

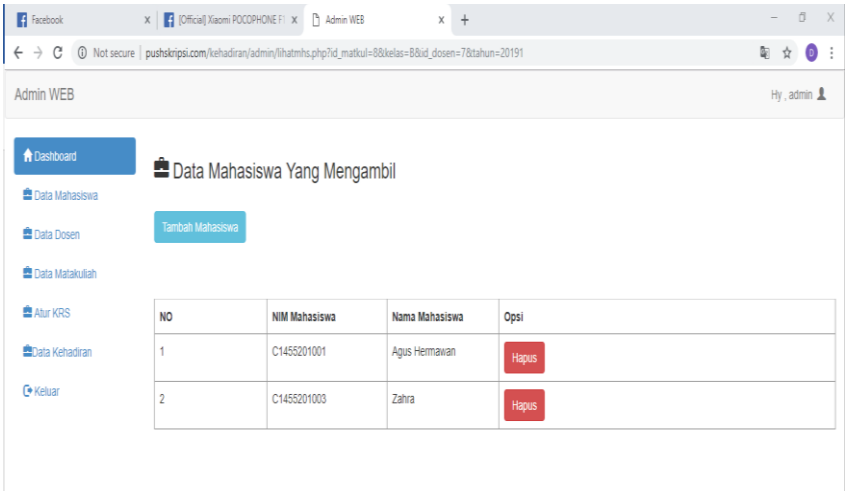
Form ini memiliki fungsi inputan berupa nama dosen, nama matakuliah, kelas dan tahun. Jika data telah dimasukkan maka tekan tombol simpan maka data akan masuk ke database, sedangkan jika tidak ingin data tersimpan bisa dengan menekan tombol batal. Seperti yang terlihat pada gambar 57 :



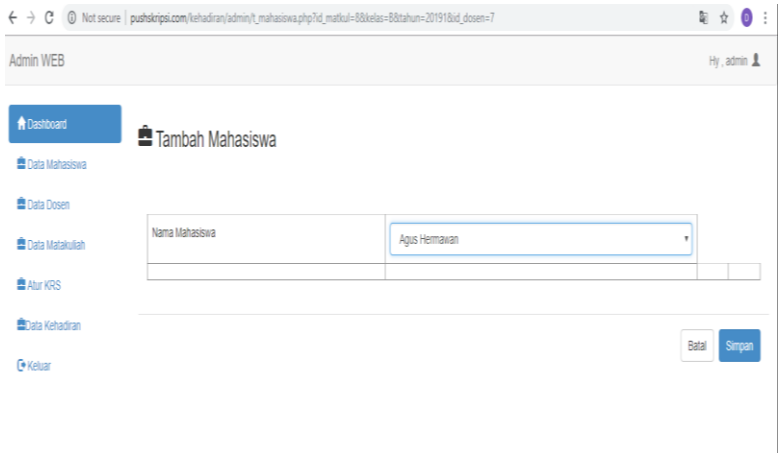
Gambar 57. *Form* Tambah KRS

5) Tombol Mahasiswa

Tombol mahasiswa memiliki fungsi untuk memasukkan data mahasiswa pada matakuliah, agar mahasiswa memiliki matakuliah dan dapat melakukan proses kehadiran perkuliahan. Seperti yang terlihat pada gambar 58 dan 59 :



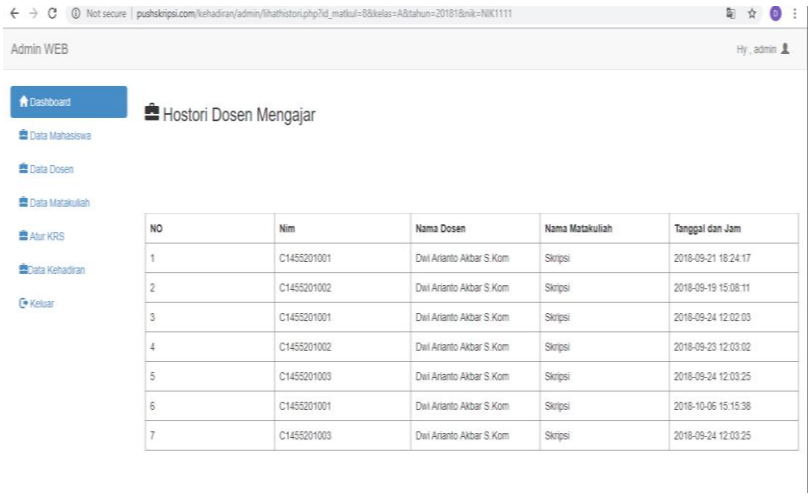
Gambar 58. Halaman Mahasiswa Pada KRS



Gambar 59. *Form* Mahasiswa Pada KRS

6) Tombol History

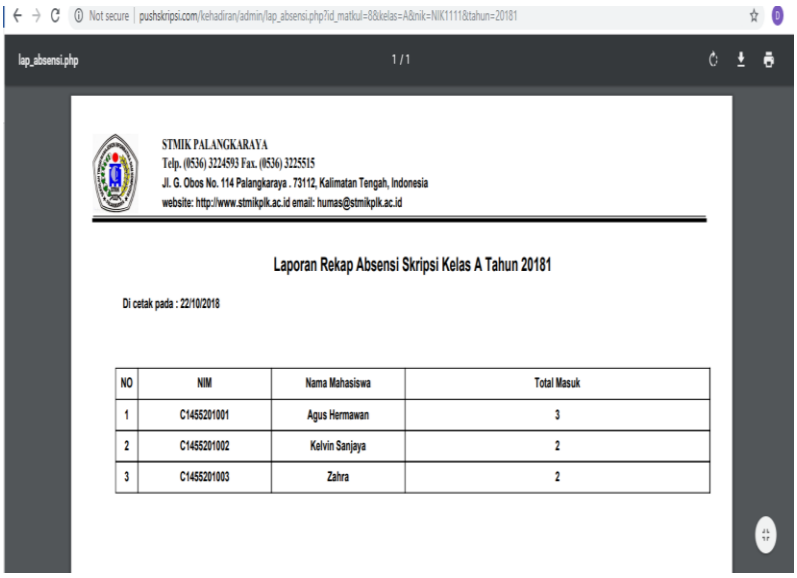
Tombol ini akan menampilkan halaman histori dosen mengajar dan menampilkan data ketika dosen melakukan absensi mahasiswa pada perkuliahan. Data yang tampil adalah nim, nama dosen, nama matakuliah, tanggal dan jam. Seperti yang terlihat pada gambar 60 :



Gambar 60. Halaman History Mengajar

7) Tombol Cetak

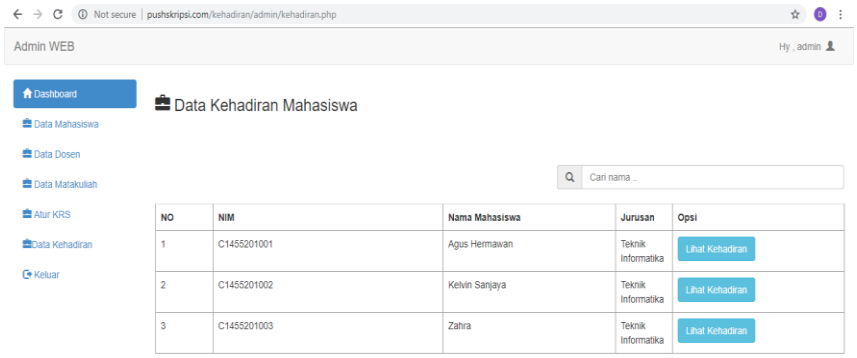
Tombol ini digunakan admin untuk menampilkan laporan kehadiran mahasiswa selama 1 semester dalam bentuk keluaran pdf, data yang ditampilkan adalah nama matakuliah, kelas, tahun, nim, nama mahasiswa dan total masuk. Seperti yang terlihat pada gambar 61 :



Gambar 61. Laporan Rekap Absensi Mengajar

8) Halaman Data Kehadiran

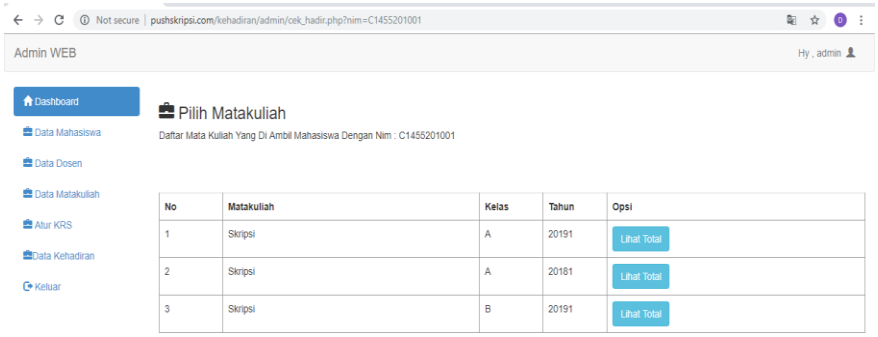
Halaman ini digunakan admin untuk melihat total mahasiswa hadir pada perkuliahan. Data yang ditampilkan pada halaman ini adalah nim, nama mahasiswa, jurusan dan tombol lihat kehadiran. Seperti yang terlihat pada gambar 62 :



Gambar 62. Data Kehadiran Mahasiswa

9) Tombol Lihat Kehadiran

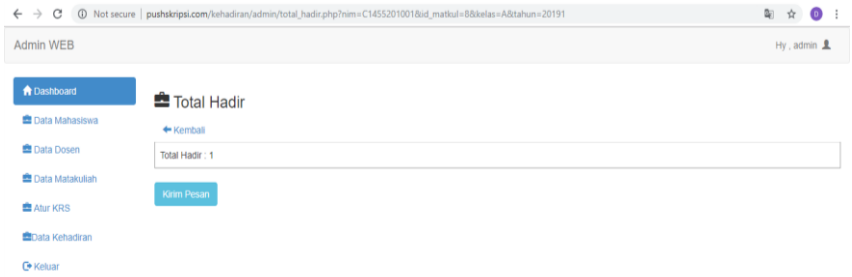
Ketika admin menekan tombol lihat kehadiran, maka akan tampil halaman yang akan menampilkan matakuliah yang dimiliki oleh mahasiswa. Data yang ditampilkan berupa nama matakuliah, kelas, tahun dan tombol lihat total. Seperti yang terlihat pada gambar 63 :



Gambar 63. Pilih Matakuliah Mahasiswa

10) Tombol Lihat Total

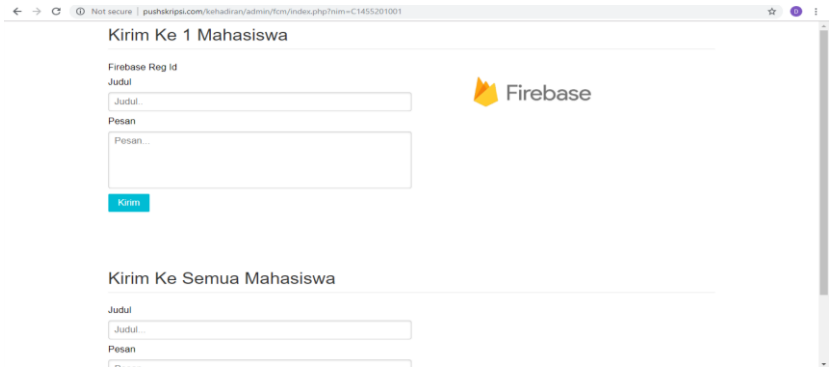
Ketika admin menekan tombol lihat total pada salah satu matakuliah, maka akan tampil halaman total hadir yang menampilkan total kehadiran mahasiswa pada matakuliah tersebut. Seperti yang terlihat pada gambar 64 :



Gambar 64. Lihat Total

11) Tombol Kirim Pesan

Tombol kirim pesan digunakan admin untuk melakukan pengiriman pemberitahuan ke mahasiswa yang bersangkutan atau kesemua mahasiswa. Karena saat admin menekan tombol kirim pesan maka akan menampilkan halaman firebase yang berisikan inputan judul dan pesan. Seperti yang terlihat pada gambar 65 :

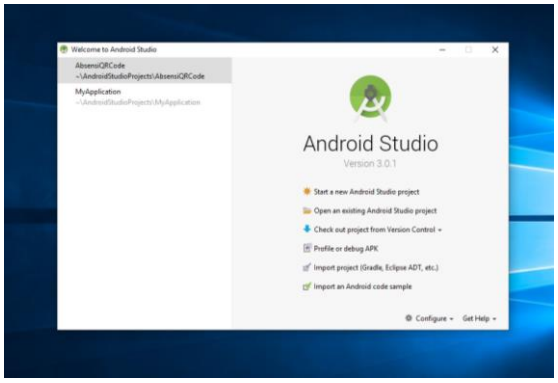


Gambar 65. Tombol Kirim Pesan

3. Manual Instalasi

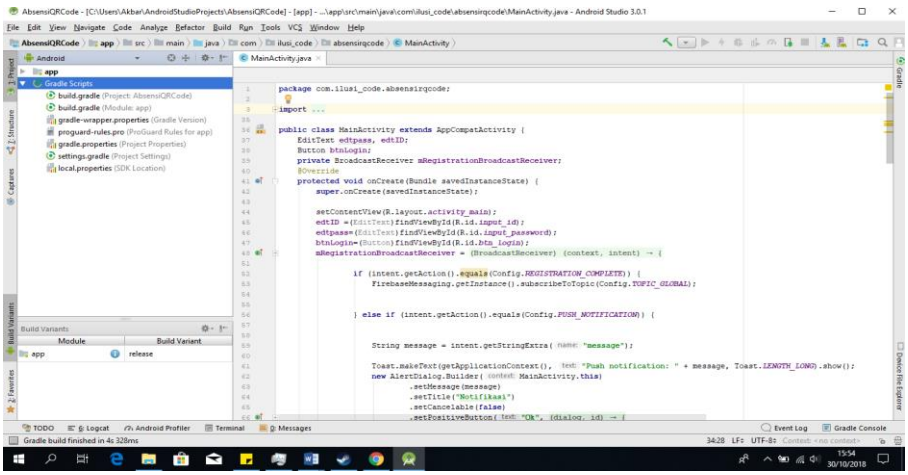
Tahapan ini merupakan langkah demi langkah instalasi Aplikasi Kehadiran Mahasiswa Menggunakan *Barcode Berbasis Android* seperti melakukan *build apk* pada *android* studio agar aplikasi bisa digunakan pada *smartphone.* Dalam tahap ini penulis akan menjelaskan langkah – langkah melakukan *build* apk melalui *android studio*.

a) Buka *android studio*



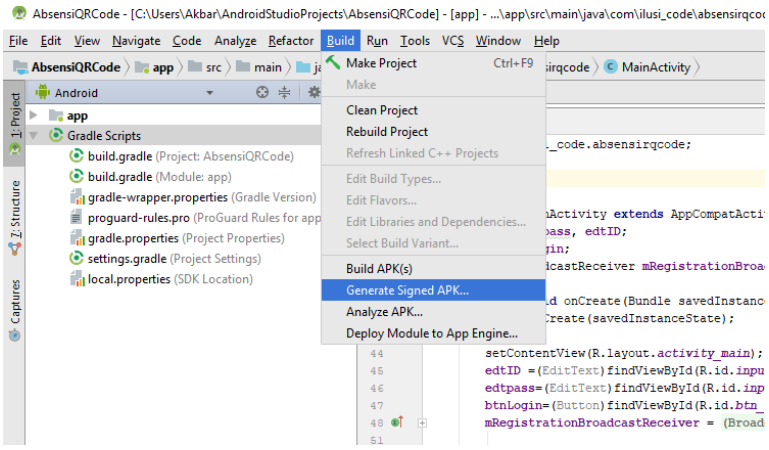
Gambar 66. Buka *Android Studio*

b) Pilih *Project* Aplikasi yang akan dibuka, maka akan tampil seperti berikut



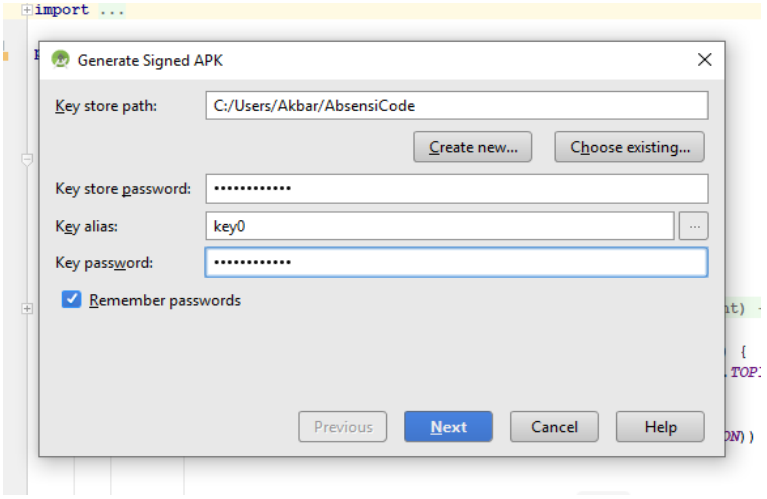
Gambar 67. *Project* Android Terbuka

c) Pilih menu *build* dan tekan *Generate Signed* APK



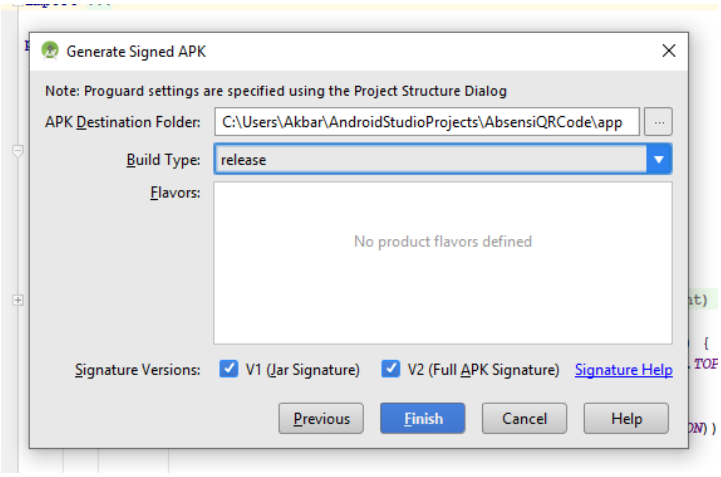
Gambar 68. Tekan *Build Project*

d) Pilih keystore path jika suda ada, jika belum silahkan pilih create new. lalu isi key store password dan key password pilih next.



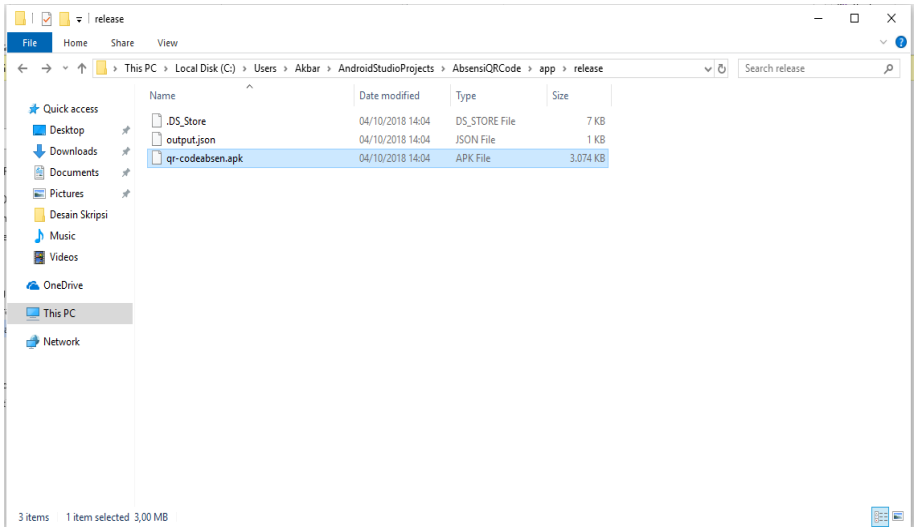
Gambar 69. *Generate Signed* APK

e) Pilih lokasi tempat aplikasi selesai dibuat menjadi apk, *build* type gunakan *release*, centang V1 dan V2 (optional). Pilih *Finish* dan tunggu beberapa saat hingga aplikasi selesai di *build* menjadi apk.



Gambar 70. *Build Type* APK

f) Setelah selesai di *build*, maka buka folder tempat file yang telah ditentukan dan aplikasi sudah bisa untuk di instal pada *smartphone*.



Gambar 71. Lokasi File APK

B. Hasil Penelitian dan Pembahasan

1. Pembahasan Listing Program

Pada tahap ini penulis menggunakan ketetapan 3 bahasa pemrograman yaitu JAVA, PHP, HTML dan MySQL sebagai *database server* program.

a. Build Gradle

Pada listing Gradle memiliki fungsi sebagai pengatur target sdk, minimum sdk dan penambahan library untuk menambahan plugin tambahan agar mempermudah proses pembuatan aplikasi android.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 27
    defaultConfig {
        applicationId "com.ilusi_code.absensirqcode"
        minSdkVersion 16
        targetSdkVersion 27
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support:support-v4:27.1.1'
    //noinspection GradleCompatible
    implementation 'com.google.firebase:firebase-messaging:17.3.0'
    implementation 'com.google.firebase:firebase-core:16.0.3'
    implementation 'com.android.volley:volley:1.0.0'
    implementation 'com.android.support:design:27.1.1'
    implementation 'com.dlazaro66.qrcodereaderview:qrcodereaderview:2.0.3'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
}
apply plugin: 'com.google.gms.google-services'
```

b. TampilkanQrcode.java

Listing ini memiliki fungsi untuk merubah data nim menjadi gambar *qr code* secara otomatis melalui jaringan internet.

```
package com.ilusi_code.absensirqcode.mahasiswa;

import android.annotation.SuppressLint;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.widget.Toast;

import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.NetworkImageView;
import com.google.firebase.messaging.FirebaseMessaging;
import com.ilusi_code.absensirqcode.MenuMhs;
import com.ilusi_code.absensirqcode.R;
import com.ilusi_code.absensirqcode.app.Config;
import com.ilusi_code.absensirqcode.util.CustomVolleyRequest;



public class TampilkanQrcode  extends AppCompatActivity {


    private NetworkImageView imageView;
    private ImageLoader imageLoader;
    private BroadcastReceiver mRegistrationBroadcastReceiver;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tampilkan_qrcode);

        imageView = (NetworkImageView) findViewById(R.id.qrcode);

        Intent intent =getIntent();

loadImage("https://chart.googleapis.com/chart?chs=300x300&cht=qr&chl="+intent.getStr
ingExtra("id_mahasiswa")+"&choe=UTF-8");
        mRegistrationBroadcastReceiver = new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {

                if (intent.getAction().equals(Config.REGISTRATION_COMPLETE)) {

FirebaseMessaging.getInstance().subscribeToTopic(Config.TOPIC_GLOBAL);


                } else if (intent.getAction().equals(Config.PUSH_NOTIFICATION)) {

                    String message = intent.getStringExtra("message");

                    Toast.makeText(getApplicationContext(), "Push notification: " + message,
Toast.LENGTH_LONG).show();
```

```
                new AlertDialog.Builder(TampilkanQrcode.this)
                        .setMessage(message)
                        .setTitle("Notifikasi")
                        .setCancelable(false)
                        .setPositiveButton("Ok", new DialogInterface.OnClickListener() {
                            @SuppressLint("NewApi")
                            public void onClick(DialogInterface dialog, int id) {
                                dialog.cancel();

                            }
                        })


                        .show();

                }
            }
        };
    }

    private void loadImage(String qrcode){


        imageLoader = CustomVolleyRequest.getInstance(this.getApplicationContext())
                .getImageLoader();
        imageLoader.get(qrcode, ImageLoader.getImageListener(imageView,
                R.drawable.logo, android.R.drawable
                        .ic_dialog_alert));



                imageView.setImageUrl(qrcode, imageLoader);
    }

}
```

c. ListJadwal.java

Listing ini berfungsi untuk menampilkan daftar jadwal yang dimiliki mahasiswa.

```
package com.ilusi_code.absensirqcode.mahasiswa;

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
```

```
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.ilusi_code.absensirqcode.R;
import com.ilusi_code.absensirqcode.jadwal.Jadwal;
import com.ilusi_code.absensirqcode.jadwal.JadwalAdapter;
import com.ilusi_code.absensirqcode.jadwal.JadwalJson;
import com.ilusi_code.absensirqcode.pdModel.pdModel;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

/**
 *
 */

public class ListJadwal  extends AppCompatActivity implements
ListView.OnItemClickListener {
    List<Jadwal> jadwalList;
    ListView lv;
    TextView tvNim,tvNim2;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);


        final Intent intent = getIntent();
        lv = (ListView) findViewById(R.id.list);
        lv.setOnItemClickListener(this);
        tvNim =(TextView)findViewById(R.id.tvidmahasiswa);
        tvNim.setText(intent.getStringExtra("id_mahasiswa"));
        tvNim2 =(TextView)findViewById(R.id.tvidmahasiswa2);
        tvNim2.setText(intent.getStringExtra("nim"));
        requestData("http://pushskripsi.com/kehadiran/listjadwal.php?id_mahasiswa="+
tvNim.getText().toString());


    }
    public void requestData(String uri) {

        StringRequest request = new StringRequest(uri,

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        jadwalList = JadwalJson.parseData(response);
                        JadwalAdapter adapter = new JadwalAdapter(ListJadwal.this, jadwalList);
                        lv.setAdapter(adapter);
                        pdModel.hideProgressDialog();

                    }
                },

                new Response.ErrorListener() {
                    @Override
```

```
                public void onErrorResponse(VolleyError error) {

                    Toast.makeText(ListJadwal.this, error.getMessage().toString(),
Toast.LENGTH_SHORT).show();
                    pdModel.hideProgressDialog();
                }
            });

        RequestQueue queue = Volley.newRequestQueue(this);
        queue.add(request);
    }



    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

        Jadwal jadwal = jadwalList.get(i);
        getAbsen(tvNim2.getText().toString(),jadwal.getId_matkul(),jadwal.getKelas());


    }

   public void getAbsen(String nim,String id_matkul,String kelas) {
        pdModel.pdLogin(ListJadwal.this);
        final String url
="http://pushskripsi.com/kehadiran/totalabsen.php?nim="+nim+"&id_matkul="+id_matkul
+"&kelas="+kelas;
        StringRequest stringRequest2 = new StringRequest(url , new
Response.Listener<String>() {
            @Override
            public void onResponse(String response2) {
                String nama_dosen = "";
                String nama = "";
                try {
                    JSONObject jsonObject = new JSONObject(response2);
                    JSONArray result = jsonObject.getJSONArray("total");
                    JSONObject collegeData = result.getJSONObject(0);
                    nama = collegeData.getString("total_hadir");
                    nama_dosen = collegeData.getString("nama");



                } catch (JSONException e) {
                    e.printStackTrace();
                }
                 new AlertDialog.Builder(ListJadwal.this)
                        .setMessage(nama+"\n"+"Dosen : "+nama_dosen)
                        .setTitle("Total Hadir")
                        .setCancelable(false)
                        .setPositiveButton("Ok", new DialogInterface.OnClickListener() {
                            @SuppressLint("NewApi")
                            public void onClick(DialogInterface dialog, int id) {
                                dialog.dismiss();

                            }
                        })


                        .show();
                pdModel.hideProgressDialog();
```

```
            }
        },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(ListJadwal.this, "Not Connections",
Toast.LENGTH_LONG).show();
                        pdModel.hideProgressDialog();

                    }
                });

        RequestQueue requestQueue2 = Volley.newRequestQueue(ListJadwal.this);
        requestQueue2.add(stringRequest2);
    }

}
```

d. PilihMatkul.java

Listing ini berfungsi untuk menampilkan pilihan matakuliah yang akan diajarkan oleh dosen.

```
package com.ilusi_code.absensirqcode.dosen;

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.ilusi_code.absensirqcode.R;
import com.ilusi_code.absensirqcode.jadwal.Jadwal;
import com.ilusi_code.absensirqcode.jadwal.JadwalAdapter;
import com.ilusi_code.absensirqcode.jadwal.JadwalJson;
import com.ilusi_code.absensirqcode.mahasiswa.ListJadwal;
import com.ilusi_code.absensirqcode.pdModel.pdModel;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

/**
 *.
 */
```

```
public class PilihMatkul extends AppCompatActivity implements
ListView.OnItemClickListener {
    List<Jadwal> jadwalList;
    ListView lv;
    TextView tvnik,tvnik2;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);


        final Intent intent = getIntent();
        lv = (ListView) findViewById(R.id.list);
        lv.setOnItemClickListener(this);
        tvnik =(TextView)findViewById(R.id.tvidmahasiswa);
        tvnik.setText(intent.getStringExtra("id_dosen"));
        tvnik2 =(TextView)findViewById(R.id.tvidmahasiswa2);
        tvnik2.setText(intent.getStringExtra("nik"));
        requestData("http://pushskripsi.com/kehadiran/listjadwaldosen.php?id_dosen="+
tvnik.getText().toString());


    }
    public void requestData(String uri) {

        StringRequest request = new StringRequest(uri,

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        jadwalList = JadwalJson.parseData(response);
                        JadwalAdapter adapter = new JadwalAdapter(PilihMatkul.this,
jadwalList);
                        lv.setAdapter(adapter);
                        pdModel.hideProgressDialog();

                    }
                },

                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {

                        Toast.makeText(PilihMatkul.this, error.getMessage().toString(),
Toast.LENGTH_SHORT).show();
                        pdModel.hideProgressDialog();
                    }
                });

        RequestQueue queue = Volley.newRequestQueue(this);
        queue.add(request);
    }



    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

        Jadwal jadwal = jadwalList.get(i);
        Intent intent =new
```

```
Intent(PilihMatkul.this,BacaQrcode.class);
        intent.putExtra("id_matkul",jadwal.getId_matkul());
        intent.putExtra("id_krs",jadwal.getId_krs());
        intent.putExtra("kelas",jadwal.getKelas());
        intent.putExtra("id_dosen",tvnik.getText().toString());
        intent.putExtra("nik",tvnik2.getText().toString());
        startActivity(intent);

    }

}
```

e. BacaQrcode.java

Listing ini berfungsi agar kamera dapat melakukan *scan* nim mahasiswa yang sudah menjadi *qr code*.

```
package com.ilusi_code.absensirqcode.dosen;

import android.content.Intent;
import android.graphics.PointF;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.dlazaro66.qrcodereaderview.QRCodeReaderView;
import com.ilusi_code.absensirqcode.R;

import java.util.Hashtable;
import java.util.Map;

/**
 *
 */

public class BacaQrcode extends AppCompatActivity implements
QRCodeReaderView.OnQRCodeReadListener{

    QRCodeReaderView qrCodeReaderView;
    TextView tvnik,tvidmatkul,tvkelas,tvidkrs;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.scan_qrcode);
        tvnik=(TextView)findViewById(R.id.tvnik);
        tvkelas=(TextView)findViewById(R.id.tvkelas);
        tvidmatkul=(TextView)findViewById(R.id.tvidmatkul);
        tvidkrs=(TextView)findViewById(R.id.tvidkrs);
        Intent intent=getIntent();
        tvnik.setText(intent.getStringExtra("nik"));
```

```
        tvidmatkul.setText(intent.getStringExtra("id_matkul"));
        tvkelas.setText(intent.getStringExtra("kelas"));
        tvidkrs.setText(intent.getStringExtra("id_krs"));
        qrCodeReaderView=(QRCodeReaderView)findViewById(R.id.bacaqrcode);
        qrCodeReaderView.setOnQRCodeReadListener(this);
        qrCodeReaderView.setQRDecodingEnabled(true);
        qrCodeReaderView.setAutofocusInterval(2000L);
        qrCodeReaderView.setTorchEnabled(true);
        qrCodeReaderView.setFrontCamera();
        qrCodeReaderView.setBackCamera();

    }

    @Override
    public void onQRCodeRead(String text, PointF[] points) {
        Toast.makeText(BacaQrcode.this,text,Toast.LENGTH_LONG).show();
        absensi(text);
        qrCodeReaderView.stopCamera();
    }

    @Override
    protected void onResume() {
        super.onResume();
        qrCodeReaderView.startCamera();
    }

    @Override
    protected void onPause() {
        super.onPause();
        qrCodeReaderView.stopCamera();
    }
    private void absensi(final String nim){

        StringRequest stringRequest = new StringRequest(Request.Method.POST,
"http://pushskripsi.com/kehadiran/insertAbsensi.php",

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String s) {

Toast.makeText(BacaQrcode.this,"Sukses",Toast.LENGTH_SHORT).show();

                        finish();


                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError volleyError) {

                        Toast.makeText(BacaQrcode.this, "Gagal ",
Toast.LENGTH_LONG).show();


                    }
                }){
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {


                Map<String,String> params = new Hashtable<String, String>();
```

```
                params.put("nik",tvnik.getText().toString());
                params.put("nim",nim);
                params.put("kelas",tvkelas.getText().toString());
                params.put("id_matkul", tvidmatkul.getText().toString());
                params.put("id_krs",tvidkrs.getText().toString());


                return params;
            }
        };

        RequestQueue requestQueue = Volley.newRequestQueue(this);

        requestQueue.add(stringRequest);
    }
}
```

f. JadwalDosen.java

Listing ini berfungsi untuk menampilkan daftar matakuliah yang dimiliki oleh dosen.

```
package com.ilusi_code.absensirqcode.dosen;

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.ilusi_code.absensirqcode.R;
import com.ilusi_code.absensirqcode.jadwal.Jadwal;
import com.ilusi_code.absensirqcode.jadwal.JadwalAdapter;
import com.ilusi_code.absensirqcode.jadwal.JadwalJson;
import com.ilusi_code.absensirqcode.mahasiswa.ListJadwal;
import com.ilusi_code.absensirqcode.pdModel.pdModel;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

/**
 *
 */
```

```
public class JadwalDosen extends AppCompatActivity implements
ListView.OnItemClickListener {
    List<Jadwal> jadwalList;
    ListView lv;
    TextView tvnik,tvnik2;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);


        final Intent intent = getIntent();
        lv = (ListView) findViewById(R.id.list);
        lv.setOnItemClickListener(this);
        tvnik =(TextView)findViewById(R.id.tvidmahasiswa);
        tvnik.setText(intent.getStringExtra("id_dosen"));
        tvnik2 =(TextView)findViewById(R.id.tvidmahasiswa2);
        tvnik2.setText(intent.getStringExtra("nik"));
        requestData("http://pushskripsi.com/kehadiran/listjadwaldosen.php?id_dosen="+
tvnik.getText().toString());


    }
    public void requestData(String uri) {

        StringRequest request = new StringRequest(uri,

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        jadwalList = JadwalJson.parseData(response);
                        JadwalAdapter adapter = new JadwalAdapter(JadwalDosen.this,
jadwalList);
                        lv.setAdapter(adapter);
                        pdModel.hideProgressDialog();

                    }
                },

                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {

                        Toast.makeText(JadwalDosen.this, error.getMessage().toString(),
Toast.LENGTH_SHORT).show();
                        pdModel.hideProgressDialog();
                    }
                });

        RequestQueue queue = Volley.newRequestQueue(this);
        queue.add(request);
    }



    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

        Jadwal jadwal = jadwalList.get(i);
        Intent intent =new Intent(JadwalDosen.this,LihatTotdalDosen.class);
```

```
        intent.putExtra("id_dosen",tvnik.getText().toString());
        intent.putExtra("nik",tvnik2.getText().toString());

        intent.putExtra("id_matkul",jadwal.getId_matkul());
        intent.putExtra("tahun",jadwal.getTahun());
        intent.putExtra("kelas",jadwal.getKelas());


        startActivity(intent);


    }

    public void getAbsen(String nik,String id_matkul,String kelas) {
        pdModel.pdLogin(JadwalDosen.this);
        final String url
="http://pushskripsi.com/kehadiran/totaldosen.php?nik="+nik+"&id_matkul="+id_matkul+
"&kelas="+kelas;
        StringRequest stringRequest2 = new StringRequest(url , new
Response.Listener<String>() {
            @Override
            public void onResponse(String response2) {
                String nama = "";



                try {
                    JSONObject jsonObject = new JSONObject(response2);
                    JSONArray result = jsonObject.getJSONArray("total");
                    JSONObject collegeData = result.getJSONObject(0);
                    nama = collegeData.getString("total_hadir");



                } catch (JSONException e) {
                    e.printStackTrace();
                }
                 new AlertDialog.Builder(JadwalDosen.this)
                        .setMessage(nama)
                        .setTitle("Total Hadir")
                        .setCancelable(false)
                        .setPositiveButton("Lihat Mahasiswa", new
DialogInterface.OnClickListener() {
                            @SuppressLint("NewApi")
                            public void onClick(DialogInterface dialog, int id) {
                                dialog.dismiss();

                            }
                        })



                        .show();
                pdModel.hideProgressDialog();



            }
        },
```

```
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(JadwalDosen.this, "Not Connections",
Toast.LENGTH_LONG).show();
                    pdModel.hideProgressDialog();

                }
            });

        RequestQueue requestQueue2 = Volley.newRequestQueue(JadwalDosen.this);
        requestQueue2.add(stringRequest2);
    }
}
```

g. LihatTodalDosen.java

Listing ini berfungsi untuk menampilkan total dosen mengajar pada tiap matakuliah selama 1 semester.

```
package com.ilusi_code.absensirqcode.dosen;

import android.annotation.SuppressLint;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.ilusi_code.absensirqcode.R;
import com.ilusi_code.absensirqcode.jadwal.Jadwal;
import com.ilusi_code.absensirqcode.jadwal.JadwalAdapter;
import com.ilusi_code.absensirqcode.jadwal.JadwalJson;
import com.ilusi_code.absensirqcode.mhs.Mhs;
import com.ilusi_code.absensirqcode.mhs.MhsAdapter;
import com.ilusi_code.absensirqcode.mhs.MhsJson;
import com.ilusi_code.absensirqcode.pdModel.pdModel;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.List;

/**
 *
 */
```

```
public class LihatTotdalDosen extends AppCompatActivity implements
ListView.OnItemClickListener {
    List<Mhs> jadwalList;
    ListView lv;
    TextView tvnik,tvnik2, tvidmatkul,tvkelas,tvtotal;


    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list);


        final Intent intent = getIntent();
        lv = (ListView) findViewById(R.id.list);
        lv.setOnItemClickListener(this);
        tvnik = (TextView) findViewById(R.id.tvnik);
        tvnik2 = (TextView) findViewById(R.id.tvnik2);
        tvtotal = (TextView) findViewById(R.id.tvtotal);
        tvnik.setText(intent.getStringExtra("id_dosen"));
        tvnik2.setText(intent.getStringExtra("nik"));
        tvidmatkul = (TextView) findViewById(R.id.tvidjadwal);
        tvidmatkul.setText(intent.getStringExtra("id_matkul"));
        tvkelas = (TextView) findViewById(R.id.tvkelas);
        tvkelas.setText(intent.getStringExtra("kelas"));
        requestData("http://pushskripsi.com/kehadiran/listMhs.php?id_dosen=" +
tvnik.getText().toString()+"&id_matkul="+tvidmatkul.getText().toString()+"&kelas="+tvkel
as.getText().toString());

getAbsen(tvnik2.getText().toString(),tvidmatkul.getText().toString(),tvkelas.getText().toStr
ing());

    }

    public void requestData(String uri) {

        StringRequest request = new StringRequest(uri,

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        jadwalList = MhsJson.parseData(response);
                        MhsAdapter adapter = new MhsAdapter(LihatTotdalDosen.this,
jadwalList);
                        lv.setAdapter(adapter);
                        pdModel.hideProgressDialog();

                    }
                },

                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {

                        Toast.makeText(LihatTotdalDosen.this, error.getMessage().toString(),
Toast.LENGTH_SHORT).show();
                        pdModel.hideProgressDialog();
                    }
                });

        RequestQueue queue = Volley.newRequestQueue(this);
```

```
        queue.add(request);
    }


    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

    }

    public void getAbsen(String nik, String id_matkul, String kelas) {
        pdModel.pdLogin(LihatTotdalDosen.this);
        final String url = "http://pushskripsi.com/kehadiran/totaldosen.php?nik=" + nik +
"&id_matkul=" + id_matkul + "&kelas=" + kelas;
        StringRequest stringRequest2 = new StringRequest(url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response2) {
                String nama = "";

                try {
                    JSONObject jsonObject = new JSONObject(response2);
                    JSONArray result = jsonObject.getJSONArray("total");
                    JSONObject collegeData = result.getJSONObject(0);
                    nama = collegeData.getString("total_hadir");



                } catch (JSONException e) {
                    e.printStackTrace();
                }
               tvtotal.setText("Total Absensi : "+nama);
                pdModel.hideProgressDialog();


            }
        },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(LihatTotdalDosen.this, "Not Connections",
Toast.LENGTH_LONG).show();
                        pdModel.hideProgressDialog();

                    }
                });

        RequestQueue requestQueue2 =
Volley.newRequestQueue(LihatTotdalDosen.this);
        requestQueue2.add(stringRequest2);
    }

}
```

h. Mhs.java

Listing ini berfungsi agar data variable mahasiswa tipe data nya dapat ditetapkan.

```
package com.ilusi_code.absensirqcode.mhs;
```

```
/**
 */

public class Mhs {
    private String nama;
    private String nim;

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNim() {
        return nim;
    }

    public void setNim(String nim) {
        this.nim = nim;
    }
}
```

i. MhsAdapter.java

Listing ini berfungsi agar data mahasiswa dapat ditampilkan dalam bentuk list.

```
package com.ilusi_code.absensirqcode.mhs;

import android.content.Context;
import android.graphics.Bitmap;
import android.util.LruCache;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;
import com.ilusi_code.absensirqcode.R;

import java.util.List;

/**
 *.
 */

public class MhsAdapter extends BaseAdapter {
    private Context context;
    private List<Mhs> pList;
    private LayoutInflater inflater = null;
    private LruCache<Integer,Bitmap> imageCache;
    private RequestQueue queue;
    public MhsAdapter(Context context, List<Mhs> list) {
```

```
        this.context = context;
        this.pList = list;
        inflater = LayoutInflater.from(context);
        final int maxMemory = (int) (Runtime.getRuntime().maxMemory() / 1024);
        final int cacheSize = maxMemory / 8;
        imageCache = new LruCache<>(cacheSize);
        queue = Volley.newRequestQueue(context);
    }

    public class ViewHolder {

        TextView _d1;
        TextView _d2;
        TextView _d3;


    }

    @Override
    public int getCount() {
        return pList.size();
    }

    @Override
    public Object getItem(int position) {

        return pList.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup viewGroup) {


        final Mhs p = pList.get(position);
        final MhsAdapter.ViewHolder holder;
        if(convertView == null) {

            convertView = inflater.inflate(R.layout.template_list,null);
            holder = new MhsAdapter.ViewHolder();

            holder._d1 = (TextView) convertView.findViewById(R.id.tv1);
            holder._d2 = (TextView) convertView.findViewById(R.id.tv2);
            holder._d3 = (TextView) convertView.findViewById(R.id.tv3);


            convertView.setTag(holder);
        }
        else {

            holder = (MhsAdapter.ViewHolder) convertView.getTag();
        }

        holder._d2.setText(""+p.getNama().toString());
        holder._d3.setText(""+p.getNim().toString());
       return convertView;
    }
}
```

j. MhsJson.java

Listing ini memiliki fungsi untuk menampilkan data berupa nama dan nim mahasiswa dalam bentuk list.

```
package com.ilusi_code.absensirqcode.mhs;

import com.ilusi_code.absensirqcode.jadwal.Jadwal;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

/**
 *
 */

public class MhsJson {
    static List<Mhs> pList;

    public static List<Mhs> parseData(String content) {

        JSONArray p_arry = null;
        Mhs p = null;
        try {

            p_arry = new JSONArray(content);
            pList = new ArrayList<>();

            for (int i = 0; i < p_arry.length(); i++) {

                JSONObject obj = p_arry.getJSONObject(i);
                p = new Mhs();
                p.setNama(obj.getString("nama"));
                p.setNim(obj.getString("nim"));



                pList.add(p);
            }
            return pList;

        }
        catch (JSONException ex) {
            ex.printStackTrace();
            return null;
        }
    }
}
```

k. Koneksi Android

Listing ini berfungsi untuk menghubungkan aplikasi android dengan database agar android dapat mengakses data pada server.

```
<?php
     define('HOST','localhost');
     define('USER','pushskri_user');
     define('PASS','nvgeforceGtx9600');
     define('DB','pushskri_kehadiran');

     $con = mysqli_connect(HOST,USER,PASS,DB) or
die('Unable to Connect');
?>
```

l. insertAbsensi.php

Listing ini berfungsi agar ketika proses scan qr code nim mahasiswa data akan masuk ke database.

```
<?php
require_once 'config.php';
       $nik = $_POST["nik"];
       $id_matkul = $_POST["id_matkul"];
       $id_krs = $_POST["id_krs"];
$nim = $_POST["nim"];
$kelas = $_POST["kelas"];
$tahun2 = $_POST["tahun"];
$bulan=date('m', strtotime('-0 month'));
$tgl=date('d', strtotime('-0 day'));
$tahun=date('Y');
$query2= "SELECT * FROM tbl_kehadiran WHERE nik='$nik'
and id_matkul='$id_matkul' and nim='$nim' and
kelas='$kelas' and MONTH(tanggal)='$bulan' and
DAY(tanggal)='$tgl' and YEAR(tanggal)='$tahun'";

$result = mysqli_query($con,$query2);
if (mysqli_num_rows($result)>0)
{ //proses pengecekan data
    echo "<script>
                alert('Data Sudah Ada silahkan ulangi data
yang anda daftarkan');

          </script>";
}
else
{ //tuliskan disini perintah untuk insert data

       $query = "INSERT tbl_kehadiran SET nik=
'$nik',id_matkul='$id_matkul',id_krs='$id_krs',nim='$nim'
,kelas='$kelas',tahun='$tahun2', total='1'";
```

```
        mysqli_query($con, $query) or
 die(mysqli_error($con));
        mysqli_close($con);
 }
 ?>
```

m. listjadwal.php

Listing ini berfungsi untuk menampilkan jadwal mahasiswa yang ada pada database ke aplikasi.

```
<?php
require_once 'config.php';
 $nik = $_POST["nik"];
 $id_matkul = $_POST["id_matkul"];
 $id_krs = $_POST["id_krs"];
$nim = $_POST["nim"];
$kelas = $_POST["kelas"];
$tahun2 = $_POST["tahun"];
$bulan=date('m', strtotime('-0 month'));
$tgl=date('d', strtotime('-0 day'));
$tahun=date('Y');
$query2= "SELECT * FROM tbl_kehadiran WHERE nik='$nik' and
id_matkul='$id_matkul' and nim='$nim' and kelas='$kelas'
and MONTH(tanggal)='$bulan' and DAY(tanggal)='$tgl' and
YEAR(tanggal)='$tahun'";

$result = mysqli_query($con,$query2);
if (mysqli_num_rows($result)>0)
{ //proses pengecekan data
    echo "<script>
                alert('Data Sudah Ada silahkan ulangi data
yang anda daftarkan');

          </script>";
}
else
{ //tuliskan disini perintah untuk insert data

        $query = "INSERT tbl_kehadiran SET nik=
'$nik',id_matkul='$id_matkul',id_krs='$id_krs',nim='$nim',k
elas='$kelas',tahun='$tahun2', total='1'";

 mysqli_query($con, $query) or die(mysqli_error($con));
 mysqli_close($con);
}
?>
```

n. updateRegId.php

Listing ini berfungsi untuk memberikan data id ke setiap mahasiswa agar mahasiswa dapat menerima pesan pemberitahuan melalui aplikasi.

```
<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {
 require 'config.php';
 createNilai();
}

function createNilai() {
 global $con;
 $nim = mysqli_real_escape_string($con, $_POST["nim"]);
        $regId = mysqli_real_escape_string($con,
$_POST["regId"]);


 $query = "UPDATE tbl_mahasiswa SET regId='$regId' WHERE
nim = '$nim'";

 mysqli_query($con, $query) or die(mysqli_error($con));
 mysqli_close($con);

}
?>
```

o. getIdmhs.php

Listing yang memiliki fungsi untuk mendapatkan nim mahasiswa dari id mahasiswa yang ada pada database.

```
<?php
require_once 'config.php';
$nim = mysqli_real_escape_string($con,$_GET['nim']);

$sql = "SELECT id_mahasiswa FROM tbl_mahasiswa WHERE
nim='$nim' ";
$result = mysqli_query($con, $sql);
$response = array("id_" => array());
while ($row = mysqli_fetch_array($result)) {
 $temp = array("id_mahasiswa" => $row["id_mahasiswa"]);
 array_push($response["id_"], $temp);
}

$data = json_encode($response);
$data=stripslashes($data);
echo "$data";
?>
```

p. getIddosen.php

Listing yang memiliki fungsi untuk mendapatkan nik dosen dari id dosen yang ada pada database.

```
<?php
require_once 'config.php';
$nik = mysqli_real_escape_string($con,$_GET['nik']);

$sql = "SELECT id_dosen FROM tbl_dosen WHERE nik='$nik' ";
$result = mysqli_query($con, $sql);
$response = array("id_" => array());
while ($row = mysqli_fetch_array($result)) {
 $temp = array("id_dosen" => $row["id_dosen"]);
 array_push($response["id_"], $temp);
}



$data = json_encode($response);
$data=stripslashes($data);
echo "$data";
?>
```

q. lap_absensi.php

Listing ini berfungsi untuk menampilkan laporan kehadiran mahasiswa pada tiap matakuliah selama 1 semester.

```
<?php

include 'config.php';
require('../assets/pdf/fpdf.php');

$pdf = new FPDF("L","cm","A4");

$pdf->SetMargins(2,1,1);
$pdf->AliasNbPages();
$pdf->AddPage();
$pdf->SetFont('Times','B',11);
$pdf->Image('logo.jpg',1,1,2,2);
$pdf->SetX(4);
$pdf->MultiCell(19.5,0.5,'STMIK PALANGKARAYA',0,'L');
$pdf->SetX(4);
$pdf->MultiCell(19.5,0.5,'Telp. (0536) 3224593 Fax. (0536)
3225515
',0,'L');
$pdf->SetFont('Arial','B',10);
$pdf->SetX(4);
```

```
$pdf->MultiCell(19.5,0.5,'Jl. G. Obos No. 114 Palangkaraya
. 73112, Kalimatan Tengah, Indonesia',0,'L');
$pdf->SetX(4);
$pdf->MultiCell(19.5,0.5,'website:
http://www.stmikplk.ac.id email:
humas@stmikplk.ac.id',0,'L');
$pdf->Line(1,3.1,28.5,3.1);
$pdf->SetLineWidth(0.1);
$pdf->Line(1,3.2,28.5,3.2);
$pdf->SetLineWidth(0);
$pdf->ln(1);
$pdf->SetFont('Arial','B',14);
$id_matkul=$_GET['id_matkul'];
$kelas=$_GET['kelas'];
$tahun=$_GET['tahun'];
$query2=mysql_query("SELECT * from tbl_matkul where
id_matkul='$id_matkul'");
while($lihat2=mysql_fetch_array($query2)){
$pdf->Cell(0,0.7,'Laporan Rekap Absensi
'.$lihat2['nama_matkul'].' Kelas '.$kelas.' Tahun
'.$tahun,0,0,'C');
}
$pdf->ln(1);
$pdf->SetFont('Arial','B',10);
$pdf->Cell(5,0.7,"Di cetak pada : ".date("d/m/Y"),0,0,'C');
$pdf->ln(1);
$nik=$_GET['nik'];
//$query3=mysql_query("SELECT SUM(DISTINCT(tanggal)) AS
total_hadir from tbl_kehadiran where id_matkul='$id_matkul'
AND kelas='$kelas' AND tahun='$tahun'  GROUP BY
YEAR(tanggal) ");
//while($lihat3=mysql_fetch_array($query3)){
//$pdf->Cell(5,0.7,"Total Masuk :
".$lihat3['total_hadir'],0,0,'C');
//}
$pdf->Cell(6,0.7,"",0,0,'C');
$pdf->ln(1);
$pdf->Cell(1, 0.8, 'NO', 1, 0, 'C');
$pdf->Cell(6, 0.8, 'NIM', 1, 0, 'C');
$pdf->Cell(6, 0.8, 'Nama Mahasiswa', 1, 0, 'C');
$pdf->Cell(0, 0.8, 'Total Masuk', 1, 0, 'C');
$pdf->Cell(0, 0.8, '', 1, 0, 'C');
$pdf->Cell(0, 0.8, '', 1, 0, 'C');
$pdf->Cell(0, 0.8, '', 1, 1, 'C');

$no=1;
$id_matkul=$_GET['id_matkul'];
$query=mysql_query("SELECT
tbl_mahasiswa.nama,tbl_mahasiswa.nim ,
SUM(tbl_kehadiran.total) as total_hadir FROM tbl_mahasiswa
JOIN tbl_kehadiran ON tbl_mahasiswa.nim=tbl_kehadiran.nim
WHERE tbl_kehadiran.id_matkul='$id_matkul' and
tbl_kehadiran.kelas='$kelas' and
tbl_kehadiran.tahun='$tahun' GROUP BY tbl_kehadiran.nim,
'$tahun' ");
while($lihat=mysql_fetch_array($query)){
```

```
$pdf->Cell(1, 0.8, $no , 1, 0, 'C');
$pdf->Cell(6, 0.8, $lihat['nim'],1, 0, 'C');
$pdf->Cell(6, 0.8, $lihat['nama'],1, 0, 'C');
$pdf->Cell(0, 0.8, $lihat['total_hadir'], 1, 0,'C');
$pdf->Cell(0, 0.8, "",1, 0, 'C');
$pdf->Cell(0, 0.8, "" , 1, 1,'C');
$no++;
}

$pdf->Output("laporan_buku.pdf","I");

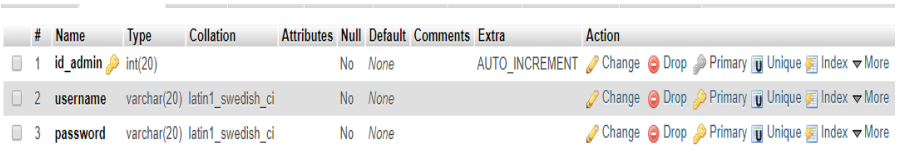
?>
```

2. Pembahasan Basis Data

Basis data yang digunakan pada aplikasi kehadiran mahasiswa menggunakan barcode berbasis android terdiri dari beberapa tabel, yaitu sebagai berikut :

a. Tabel Admin

Pada gambar di bawah ini merupakan tampilan tabel pada database yang digunakan untuk penyimpanan data admin.

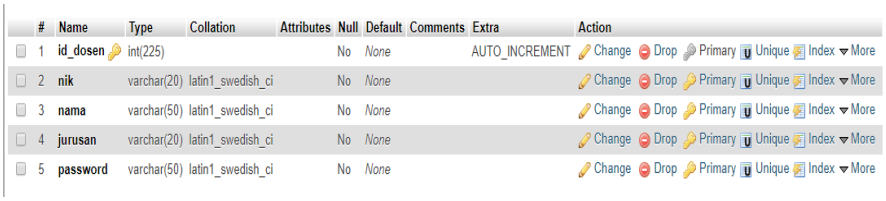


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_admin	int(20)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	username	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
3	password	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More

Gambar 72. Struktur Tabel Admin

b. Tabel Dosen

Pada gambar di bawah ini merupakan tampilan tabel pada database yang digunakan untuk penyimpanan data dosen.

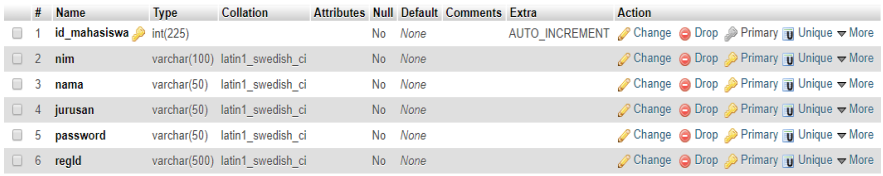


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_dosen	int(225)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	nik	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
3	nama	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
4	jurusan	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
5	password	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More

Gambar 73. Struktur Tabel Dosen

c. Tabel Mahasiswa

Pada gambar di bawah ini merupakan tampilan tabel pada database yang digunakan untik penyimpanan data mahasiswa.

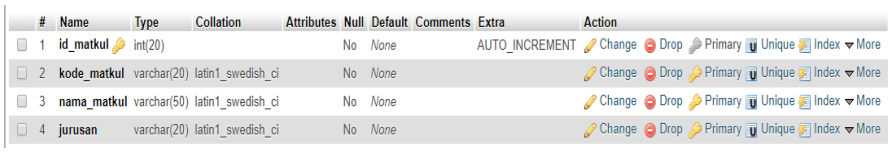


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_mahasiswa	int(225)			No	None		AUTO_INCREMENT	Change Drop Primary Unique More
2	nim	varchar(100)	latin1_swedish_ci		No	None			Change Drop Primary Unique More
3	nama	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique More
4	jurusan	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique More
5	password	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique More
6	regId	varchar(500)	latin1_swedish_ci		No	None			Change Drop Primary Unique More

Gambar 74. Struktur Tabel Mahasiswa

d. Tabel Matakuliah

Pada gambar di bawah ini merupakan tampilan tabel pada database yang digunakan untuk penyimpanan data matakuliah.

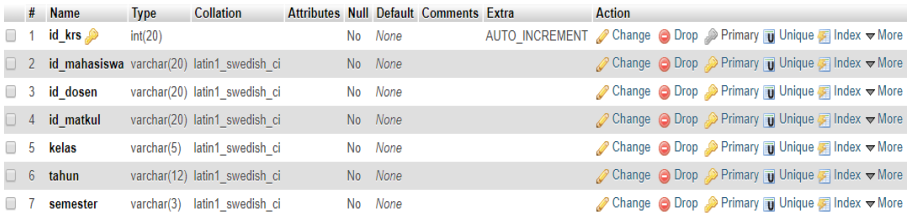


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_matkul	int(20)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	kode_matkul	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
3	nama_matkul	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
4	jurusan	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More

Gambar 75. Struktur Tabel Matakuliah

e. Tabel KRS

Pada gambar di bawah ini merupakan tampilan tabel pada database yang digunakan untuk penyimpanan data krs.

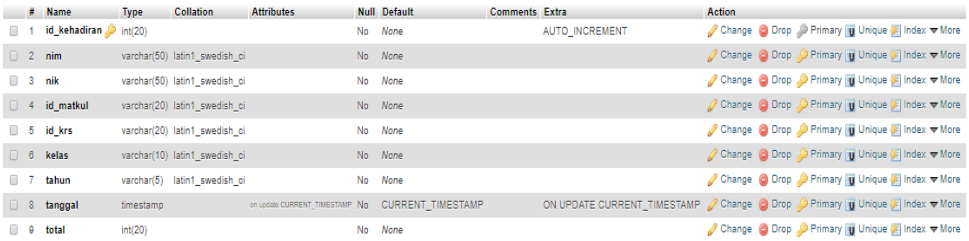


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_krs	int(20)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	id_mahasiswa	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
3	id_dosen	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
4	id_matkul	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
5	kelas	varchar(5)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
6	tahun	varchar(12)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
7	semester	varchar(3)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More

Gambar 76. Struktur Tabel KRS

f. Tabel Kehadiran

Pada gambar di bawah ini merupakan tampilan tabel pada database yang digunakan untuk penyimpanan data kehadiran.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_kehadiran	int(20)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	nim	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
3	nik	varchar(50)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
4	id_matkul	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
5	id_krs	varchar(20)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
6	kelas	varchar(10)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
7	tahun	varchar(5)	latin1_swedish_ci		No	None			Change Drop Primary Unique Index More
8	tanggal	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP		ON UPDATE CURRENT_TIMESTAMP	Change Drop Primary Unique Index More
9	total	int(20)			No	None			Change Drop Primary Unique Index More

Gambar 77. Struktur Tabel Kehadiran

3. Pembahasan Hasil Response Pengguna

Pada tabel hasil kuisioner merupakan perhitungan dan penilaian responden terhadap aplikasi kehadiran mahasiswa menggunakan *barcode* berbasis android.

Tabel 33. Hasil Responden Pengguna

No.	Pernyataan	Penilaian				
		Sangat Baik	Baik	Cukup Baik	Kurang Baik	Tidak Baik
1.	Aplikasi dapat diinstal pada *smartphone android*					
2.	Aplikasi kehadiran dapat dijalankan pada *smartphone* dan tidak mengalami *crash* atau kegagalan saat dimuat.					
3.	Tampilan menu-menu utama jelas dan mudah dimengerti bagi pengguna saat dioperasikan.					
4.	Aplikasi dapat terhubung dengan kamera *smartphone*					
5.	Setiap fitur pada aplikasi kehadiran ini dapat berfungsi dengan benar dan mudah dimengerti penggunaan nya.					
6.	Informasi yang disajikan aplikasi lengkap dan akurat					

Keterangan bobot penilaian dapat dilihat pada tabel 34.

Tabel 34. Keterangan Bobot Penilaian

Keterangan	Nilai
Sangat Baik	5
Baik	4
Cukup Baik	3
Kurang Baik	2
Tidak Baik	1

Hasil kuisioner merupakan hasil perhitungan dari penilaian responden terhadap aplikasi kehadiran mahasiswa menggunakan *barcode* berbasis android. Hasil Kuesioner dapat dilihat dapat dilihat pada Tabel 29.

Tabel 35. Hasil Kuisioner

Jawaban Pertanyaan	Responden										Total Nilai Per Point
	1	2	3	4	5	6	7	8	9	10	
1	5	5	5	4	4	4	5	5	5	5	47
2	4	5	4	4	4	4	4	5	4	4	42
3	4	4	4	3	4	4	5	4	4	4	40
4	4	5	5	4	4	5	4	5	4	5	45
5	4	4	4	4	4	4	4	5	4	4	41
6	4	4	4	3	4	5	4	5	4	4	41
											256

Untuk mengetahui jumlah skor berdasarkan tabel diatas dapat dihitung menggunakan rumus sebagai berikut :

Jumlah Skor Kriteria = (skor tertinggi tiap item) x (jumlah pertanyaan) x(jumlah responden)

Jumlah Skor Kriteria = 5 x 6 x 10

= 300

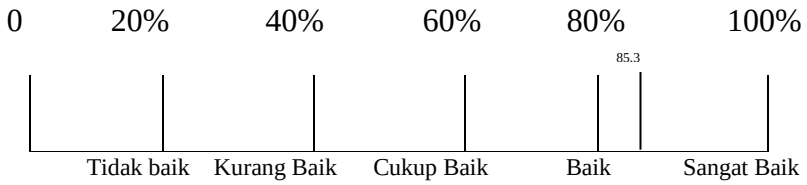
Berdasarkan tabel 29 untuk total jawaban responden adalah 256 dengan demikian aplikasi ini menurut pendapat 10 responden yaitu :

Total hasil responden = 256/300*100%

= 0.853 x 100%

= 85,3 %

Nilai total hasil responden tersebut dapat dilihat dapat dilihat pada kurva *skala likert* berikut :



Gambar 67 . Kurva *Skala Likert*
(Sumber : Sudaryono, 2015)

Jadi, menurut hasil kuisioner dari 10 responden yaitu 85,3 % menunjukan bahwa skala pada aplikasi kehadiran mahasiswa menggunakan *barcode* berbasis *android* berdasarkan gambar 67 termasuk dalam kategori baik.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan dari hasil penelitian dan pembahasan sebelumnya maka diperoleh kesimpulan, yaitu sebagai berikut:

1. Aplikasi kehadiran mahasiswa menggunakan *barcode* berbasis *android* telah berhasil
2. dibuat dengan menggunakan beberapa aplikasi pendukung seperti *android studio*, *sublime*, *xampp* dan *google chrome*.
3. Dari hasil uji coba *blackbox testing* dapat disimpulkan bahwa aplikasi yang dibuat telah sesuai dengan harapan.
4. Dari perhitungan respon pengguna menggunakan metode *skala likert* yang menghasilkan nilai sebesar 85,3 %, maka dapat disimpulkan bahwa aplikasi yang dibangun berguna untuk digunakan.

B. Saran

Adapun beberapa saran dalam penggunaan maupun pengembangan sistem ini dikemudian hari:

1. Untuk saat ini aplikasi ini hanya bisa digunakan pada perangkat android, diharapkan pada pengembangan selanjutnya aplikasi ini bisa digunakan pada perangkat ios.
2. Notifikasi aplikasi ini berfokus pada matahasisa, diharapkan kedepannya dosen juga bisa menerima notifikasi.

3. Admin melakukan pencadangan database secara berkala untuk mencegah kehilangan data.
4. Domain dan server yang digunakan pada aplikasi ini memiliki bandwith yang terbatas, jika aplikasi akan digunakan pada instansi maka silahkan menggunakan domain dan server yang memiliki bandwith lebih besar.
5. Notifikasi pada aplikasi maksimal hingga android nugget, jika ada pengembangan lebih lanjut agar notifikasi lebih maksimal pada android Oreo dan Pie.

DAFTAR PUSTAKA

Agung, Gregorius. 2015. *Mengenal Dasar-Dasar Pemrograman Android Buku Paling mudah menjelaskan cara membuat program Android yang efektif*, Jubilee Enterprise, Jakarta.

Arief, Rachman, 2017, Aplikasi Presensi Siswa Online Menggunakan Google Forms, Sheets, Sites, Awesome Table dan Gmail, *Jurnal Seminar Nasional Sains dan Teknologi Terapan*. Vol. V. Surabaya

Danur, Raka Onny Diar, I Nyoman Piarsa, A. A. Kompiang Oka Sudana, 2016, *Rancang Bangun Aplikasi Absensi Mahasiswa Pada Platform Android, Merpati Vol. 4 No. 1*. Bali.

Faridl, Miftah. *Fitur Dahsyat Sublime Text 3*, Linux User Group, Surabaya.

Febrian Wahyutama, Febriliyan Samopa dan Hatma Suryotrisongko, 2013, Penggunaan Teknologi Augmented Reality Berbasis Barcode Sebagai Sarana Penyampaian Informasi Spesifikasi dan Harga Barang yang Interaktif Berbasis Android, Studi Kasus pada Toko Elektronik ABC Surabaya, *Jurnal Teknik POMITS.* Vol. 2 No. 3.

Fikri, Mohammad Aziz, Mochammad Yusman dan Eko Subyantaro, Aplikasi Absensi Menggunakan Barcode Scanning, *Karya Ilmiah Mahasiswa Manajemen Informatika*. Bandung.

Irawan. 2012. *Membuat Aplikasi Android untuk Orang Awam*, Maxikom, Palembang.

Joni Mauliandar, S.Kom., M.Eng, Alfan Aria Mustofa, 2016, Perancangan Aplikasi Tagihan SPP Berbasis Android, *Jurnal Sains Tech Politeknik Indonusa Surakarta.* Vol. 1 Nomor 5.

Madcoms, 2010. *Kupas Tuntas Adobe Dreamweaver CS6 dengan Pemrograman PHP & MySQL*, Andy, Yogyakarta.

Pressman, Roger S. 2012. *Rekayasa Perangkat Lunak*, Andi, Yogyakarta

Suhariyaningsih, 2016, *Perancangan Computer Assisted Instruction, Jurnal Riset Komputer (JURIKOM).* Vol. 3 No. 4.

Sutabri, Tata. 2012. *Konsep Sistem Informasi*. Andi, Yogyakarta.

Triandini, Evi dan I Gede Sardika, 2012. Step by Step Desain Proyek Menggunakan UML, Andy, Yogyakarta.

Warkim, Hafiz Novanda Ichwan dan Husnul Kamal Z, 2015, Analisa dan l Sistem Kehadiran Pegawai Pada Pusat Penelitian Perkembangan l Lembaga Ilmu Pengetahuan Indonesia, *Jurnal Sistem Informasi*. Volume 1-12.

Widodo, Prabowo Pudjo. 2011. *Menggunakan UML*. Informatika, Bandung.

Yudhanto, Yudha. 2017. *Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio*, PT Gramedia, Jakarta.

Yusuf, Syamsu. 2012. *Psikologi Perkembangan Anak dan Remaja*, Remaja Rosdakarya, Bandung.

LAMPIRAN